TractorEYE: Vision-based Real-time Detection for Autonomous Vehicles in Agriculture

Peter Christiansen

SOLIDUM PETIT IN PROFUNDIS · UNIVERSITAS ARHUSIENSIS ·

PhD
Aarhus University
2017

ISBN: 978-87-7507-426-6
DOI. 10.7146/aul.277.192

Abstract

Agricultural vehicles such as tractors and harvesters have for decades been able to navigate automatically and more efficiently using commercially available products such as auto-steering and tractor-guidance systems. However, a human operator is still required inside the vehicle to ensure the safety of vehicle and especially surroundings such as humans and animals. To get fully autonomous vehicles certified for farming, computer vision algorithms and sensor technologies must detect obstacles with equivalent or better than human-level performance. Furthermore, detections must run in real-time to allow vehicles to actuate and avoid collision.

This thesis proposes a detection system (TractorEYE), a dataset (FieldSAFE), and procedures to fuse information from multiple sensor technologies to improve detection of obstacles and to generate a map.

TractorEYE is a multi-sensor detection system for autonomous vehicles in agriculture. The multi-sensor system consists of three hardware synchronized and registered sensors (stereo camera, thermal camera and multi-beam lidar) mounted on/in a ruggedized and water-resistant casing. Algorithms have been developed to run a total of six detection algorithms (four for rgb camera, one for thermal camera and one for a Multi-beam lidar) and fuse detection information in a common format using either 3D positions or Inverse Sensor Models. A GPU powered computational platform is able to run detection algorithms online. For the rgb camera, a deep learning algorithm is proposed *DeepAnomaly* to perform real-time anomaly detection of distant, heavy occluded and unknown obstacles in agriculture. DeepAnomaly is – compared to a state-of-the-art object detector *Faster R-CNN* – for an agricultural use-case able to detect humans better and at longer ranges (45-90m) using a smaller memory footprint and 7.3-times faster processing. Low memory footprint and fast processing makes DeepAnomaly suitable for real-time applications running on an embedded GPU.

FieldSAFE is a multi-modal dataset for detection of static and moving obstacles in agriculture. The dataset includes synchronized recordings from a rgb camera, stereo camera, thermal camera, 360-degree camera, lidar and radar. Precise localization and pose is provided using IMU and GPS. Ground truth of static and moving obstacles (humans, mannequin dolls, barrels, buildings, vehicles, and vegetation) are available as an annotated orthophoto and GPS coordinates for

moving obstacles.

Fusion and Mapping. Detection information from multiple detection algorithms and sensors are fused into a map using Inverse Sensor Models and occupancy grid maps.

This thesis presented many scientific contribution and state-of-the-art within perception for autonomous tractors, this includes a dataset, sensor platform, detection algorithms and procedures to perform multi-sensor fusion. Furthermore, important engineering contributions to autonomous farming vehicles are presented such as easily applicable, open-source software packages and algorithms that have been demonstrated in an end-to-end real-time detection system. The contributions of this thesis have demonstrated, addressed and solved critical issues to utilize camera-based perception systems that are essential to make autonomous vehicles in agriculture a reality.

Resumé

Landbrugsmaskiner, såsom traktorer og mejetærskere, har i årtier navigeret automatisk og effektivt med kommercielt tilgængelige produkter (auto-steering og tractor-guidance). Det kræver dog en operatør for at sikre såvel landbrugsmaskinen og specielt sikkerheden af omgivelserne såsom mennesker og dyr. For at få autonome køretøjer certificeret til landbrug, skal computeralgoritmer og sensorteknologier udføre detektion af forhindringer på et niveau der er tilsvarende eller bedre end menneskers. Endvidere, skal detektionsalgoritmer afvikles i realtid, så køretøjet kan handle og undgå kollision.

Bidraget af denne afhandlingen er et system til detektion af forhindringer (TractorEYE), et datasæt (FieldSAFE), og procedurer til at sammenkoble og udnytte information fra forskellige sensorteknologier/algoritmer for at forbedre både detektion og genereringen af kort.

TractorEYE er et system bestående af tre sensorer (stereokamera, termisk kamera og multibeam lidar) til detektion af forhindringer for autonome køretøjer i landbrug. TractorEYE består af synkroniserede og registrerede sensorer monteret i en mekanisk robust og vandafvisende kasse. Enheden består af seks algoritmer til detektion (fire til rgb kamera, én til termisk kamera og én til multi-beam lidar) og omdanner information fra forskellige algoritmer til et fælles format med enten 3D positioner eller med *Inverse Sensor Models*. En GPU accelereret beregningsplatform er i stand til at afvikle algoritmerne i realtid. Et bidrag i denne afhandling er en kamerabaseret deep learning algoritme, *DeepAnomaly*, til at udføre realtidsdetektion af anormaliteter, der i en landbrugssammenhæng ofte er fjerne, tildækkede eller ukendte forhindring eller elementer. DeepAnomaly er – sammenlignet med den førende algoritme til detektion af objekter *Faster R-CNN* – i et landbrugsscenario i stand til at detektere mennesker bedre og på længere afstand (45-90m) med et lille hukommelsesforbrug og 7.3 gange hurtigere behandling. Det lave hukommelsesforbrug og den korte processeringstid gør DeepAnomaly egnet som et realtidssystem på en indlejret GPU til landbrug.

FieldSAFE er et multi-sensor datasæt til detektion af statiske og dynamiske forhindringer i landbrug. Datasættet består af optagelser fra webcam, stereo kamera, 360-graders kamera, lidar og en radar. Præcis lokalisering og orientering er beregnet med IMU og GPS. For at kunne evaluere algoritmer til detektion er den sande placering af alle statiske elementer i marken (menne-

sker, mannequindukker, tønder, bygninger, veje, køretøjer, læhegn og vegetation) tilgængelige som et annoteret ortofoto, og forhindringer i bevægelse er angivet med GPS koordinater over tid.

Fusion and Mapping Detektionsinformation fra mange algoritmer og sensorer fusioneres til et kort med Inverse Sensor Models.

Denne afhandling præsenterer mange videnskabelige bidrag og førende forskning inden for perception til autonome køretøjer i landbrug. Dette inkluderer et datasæt, en sensorplatform, algoritmer til detektion og procedurer til at sammenkoble og udnytte information fra et multisensorsystem. Ydermere, præsenteres mange ingeniørrelaterede og realiserbare bidrag til autonome landbrugsmaskiner, såsom nemt anvendelig og frit tilgængelig software-pakker og algoritmer der er blevet demonstreret i et komplet perceptionssystem til detektion af forhindringer. I denne afhandling er der demonstreret, adresseret og løst kritiske problemer for kamerabaserede perceptionssystemer der er afgørende for at autonome køretøjer i landbrug kan blive en realitet.

Thesis Details

Thesis title	TractorEYE: Vision-based Real-time Detection for Autonomous Vehicles in Agriculture
PhD candidate	Peter Christiansen
Supervisors	Henrik Karstoft Rasmus Nyholm Jørgensen

Main contributions of this thesis are the first ten publications in Table 1. Software contributions are listed in Table 2.

Publications
Publications from this thesis are marked by a P-prefix – P[X] – to make them easily distinctable from other publications when used in the main text. Publications are divided in *Primary*, *SAFE related* and *Other*. Only *Primary* papers are address and attached in back of the dissertation.

GitHub/ROS Packages
A list of GitHub repositories developed in the project.

Table 1: List of publications

ID	Title	Type	Author	State
	Primary			
P[1]	DeepAnomaly: Combining Background Subtraction and Deep Learning for Detecting Obstacles and Anomalies in an Agricultural Field	Journal: *Sensors*	First	Published
P[2]	Automated Detection and Recognition of Wildlife Using Thermal Cameras	Journal: *Sensors*	First	Published
P[3]	FieldSAFE: Dataset for Obstacle Detection in Agriculture	*ArXiv*	First *joint*	Published *preprint*
P[4]	(Draft) Multi-modal Detection of Static and Dynamic Obstacles in Agriculture for Process Evaluation	Journal: *Multi-modal Sensor Fusion*	First *joint*	ToBeSub. 13-10-07
P[5]	Using Deep Learning to Challenge Safety Standard for Highly Autonomous Machines in Agriculture	Journal: *Journal of imaging*	First *joint*	Published
P[6]	Platform for Evaluating Sensors and Human Detection in Autonomous Mowing Operations	Journal: *Precision Agriculture*	First	Published
P[7]	Towards Autonomous Plant Production using Fully Convolutional Neural Networks	Conference: *ICPA*	First	Published
P[8]	Advanced Sensor Platform for Human Detection and Protection in Autonomous Farming	Conference: *ECPA*	First	Published
P[9]	Towards Inverse Sensor Mapping in Agriculture	Conference: *AGROB*	Third	Accepted
P[10]	Multi-modal Obstacle Detection and Evaluation of Evidence Grid Mapping in Agriculture	Conference: *ICPA*	Second	Published
	SAFE related			
P[11]	Stereo and Active-Sensor Data Fusion for Improved Stereo Block Matching	Conference: *ICIAR*	Fourth	Published
P[12]	Towards a DSL for Perception-Based Safety Systems	Conference: *DSLRob-15*	Fourth	Published
P[13]	Udviklingen inden for Præcisionsjordbrug og Tilknyttet Udstyr	Conference: *Plantekongres*	Fifth	Published
	Other			
P[14]	Thesis: Automated Classification of Seedlings Using Computer Vision	Technical report	First *joint*	-
P[15]	Estimation of Plant Species by Classifying Plants and Leaves in Combination	Journal: *Field Robotics*	Second	Published
P[16]	Field Trial Design using Semi-automated Conventional Machinery and Aerial Drone Imaging for Outlier identification	Conference *ECPA*	Last	Published

Table 2: Software Contributions

ID	Description
GH1	**ped_detector_ros** Robot Operating System (ROS) package for pedestrian detection using Local Decorrelated Channel Features [17] *https://github.com/PeteHeine/pedestrian_detector_ros*
GH2	**yolo_v2_ros** ROS package for YOLOv2 [18] , a fast Convolutional Neural Network for object detection *https://github.com/PeteHeine/yolo_v2_ros*
GH3	**fcn8_ros** ROS package for a Fully Convolutional Network for Semantic Segmentation [19] *https://github.com/PeteHeine/fcn8_ros*
GH4	**deepanomaly (private)** ROS package for DeepAnomaly P[1] a Convolutional Neural Network based anomaly detector. *https://github.com/PeteHeine/deepanomaly*
GH5	**dynamic_heat_detection** Thermal camera ROS package for detecting hot elements using a dynamic threshold. Used in P[4] *https://github.com/PeteHeine/dynamic_heat_detection*
GH6	**image_boundingbox_to_3d** ROS package for converting 2D bounding boxes to 3D. Used in P[4] *https://github.com/PeteHeine/image_boundingbox_to_3d*
GH7	**image_inverse_sensor_model2** ROS package for converting image detection to image Inverse Sensor Models. Used in P[4], P[10] *https://github.com/PeteHeine/image_inverse_sensor_model2*
GH8	**safe_sensor_msgs** ROS package defined by the SAFE protocol and functions to convert between SAFE protocol and MarkerArrays *https://github.com/PeteHeine/safe_sensor_msgs*

Preface

The PhD project is part of a 29 million kroner research project funded by Innovation Fond Denmark (Project No. 16-2014-0) called Safer Autonomous Farming Equipment (SAFE) with the goal of improving safety for both traditional and autonomous vehicles in the agricultural domain (Figure 1). The project is a collaboration between AgroIntelli, Claas Agrosystems, Conpleks Innovation, Key Research, Aarhus University (AU) and University of Southern Denmark (SDU).

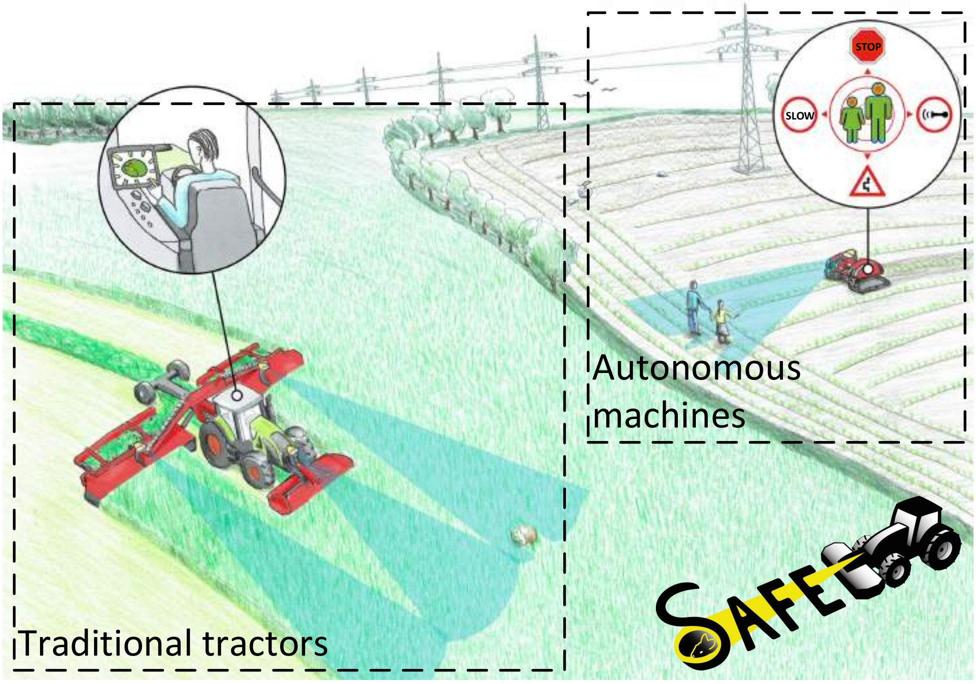


Figure 1: Illustration of the SAFE project. Traditional tractors: Perception to warn the farmer of potential hazards. Autonomous machines: Perception to enable the machine to act according to surroundings.

I would like to thank everyone in the Signal Processing Group for technical discussions, the slightly too long dinner breaks and for making school more fun. This includes Morten *Gadget* Laursen for knowing everything there is to know about hardware, sensors & interfaces and Mads Dyrmann for many years of collaborations and his valuable comments and corrections of this dissertation. Special thanks to my supervisors. Karstoft, for pushing my engineering into actual research, giving involved reviews and for providing qualified supervision on both specific theoretical/mathematical subjects and on the broader lines (what to do? and when to do it?). Rasmus, for persuading me into Academia, standing up for the Minions and making stuff happen in real life. Special thanks to my brilliant colleague and friend Mikkel Kragh, for a great collaboration in the SAFE project.

Special thanks to my girlfriend Cecilie for her support and listening to all the deep learning talk.

Aarhus, September 2017 P. Christiansen

Contents

1 Introduction

Our senses are essential in our daily lives. They allow us to perceive our surroundings and act accordingly. By working together, our senses enable us to do advanced tasks like driving a car, doing sport, or being social. Similarly, autonomous systems are required to sense and perceive surroundings to perform intelligent tasks and complete a given purpose.

Sensing is a basic ability of humans, and we are able to interpret visual information and sound in the blink of the eye [20]. Researchers have for decades struggled to give similar abilities to computers. However, a recent breakthrough – defined as Deep learning – have given computers human-level capabilities for interpreting visual information [21–25], sound/sequential data [26, 27] and learning to control a system from previous experiments (reinforcement learning) [28]. The sudden improvement of computers to interpret and act on sensor information has created a new potential for autonomous systems to perform more complicated actions - such as driving a vehicle.

Large funds are invested into self-driving cars by automotive companies, mega-corporations and startups[1], among which many have demonstrated autonomous vehicles as both prototypes and commercial products. Recently, agricultural companies have also demonstrated autonomous farming vehicles [29–31].

Agricultural vehicles such as tractors and harvesters have for decades been able to navigate automatically and more efficiently using commercially available auto-steering and tractor-guidance systems. The crucial deficiency of these systems is that a human operator is still required inside the vehicle to ensure the safety of the vehicle itself, humans, animals, and other surroundings. In order for an autonomous vehicle to operate safely and to be certified for unsupervised operation, it must perform high-accuracy real-time risk assessment and accidence avoidance in the field with high reliability.

[1] Tesla, Google/Waymo, Uber/Otto, Apple, Nvidia, Ford, Audi, Mercedes, Mapillary and Intel/Mobileye

Autonomous vehicles in the automotive industry and agriculture share a set of similar disciplines such as obstacle detection, sensor fusion, localization, mapping, planning, control and navigation. Specific to a self-driving car is higher speeds, which require lower latency processing and detection at longer range. Additionally, interacting with other cars, pedestrian and bicycles makes planning and navigation far more challenging.

However, obstacle detection in agriculture introduces a set of specific challenges. In the automotive industry, a depth-based sensor such as a lidar can exploit that all obstacles protrude or reside on the road surface. This simple heuristic enables depth-based algorithms to detect both known and unknown obstacles. In agriculture, obstacles may not protrude the crop surface and may reside below or just above an uneven surface. This introduces heavy occlusion of obstacles and depth-based sensors are less reliable for detecting both known and unknown obstacles. Furthermore, self-driving cars are also able to detect obstacles by finding elements that do not conform to pre-generated and detailed 3D map. Maps are useful in the context of agriculture, but a detail 3D map cannot similarly be used for obstacle detection, simply because crops and fields are constantly changing in appearance and extend. High performance real-time obstacle detection algorithms for agriculture are therefore an important criteria for improving safety and ultimately realizing autonomous farming machines.

The automotive industry can easily exploit the vast amount of existing and labelled datasets such as Kitti [32] and Cityscapes [33] to perform evaluation and push forward the development of detection algorithms. Similar high-quality datasets are not available in agriculture. A multimodal dataset for object detection in agriculture is therefore important to evaluate detection algorithms and to push forward development of autonomous vehicles in agriculture.

1.1 Contributions

This thesis proposes a multi-modal data set for agriculture *FieldSAFE*, a multi-sensor obstacle detection system *TractorEYE* and procedures for fusing detection information into a map. Research contributions are presented in Figure 1.1 and divided in five subjects; *Sensor Platform*, *Data Collection*, *Obstacle Detection*, *Sensor Registration & Detection Alignment* and *Localization, Fusion and Mapping*. The dissertation and sections complete the pipeline and flow of information for a multi-modal perception system. As represented in Figure 1.1, sensor data or images are processed by a detection algorithm. Detections from various algorithms and sensors are transformed into a common format as either 3D detections or a local grid maps defined as Inverse Sensor Models. Finally, various detections algorithms and sensors can with – an ISM representation – be fused in a map.

Sensor Platform The sensor platform is a multi-sensor tractor-mountable platform with six exteroceptive sensors (rgb camera, rgb 360-degree, stereo camera, thermal camera, radar, and lidar) and two localization/odometry sensors (RTK GPS, and IMU). Procedures for calibrating, synchronizing and registering sensors are performed. A thermal calibration panel is proposed to calibrate and determine extrinsic parameters of thermal and rgb cameras.
Data Collection Data has been gathered from a tractor to get realistic data in terms of mechanical vibrations and to capture natural elements for an agricultural field (field, houses, roads, shelterbelts, trees and vehicles). To also simulate hazard situations mannequins, barrels,

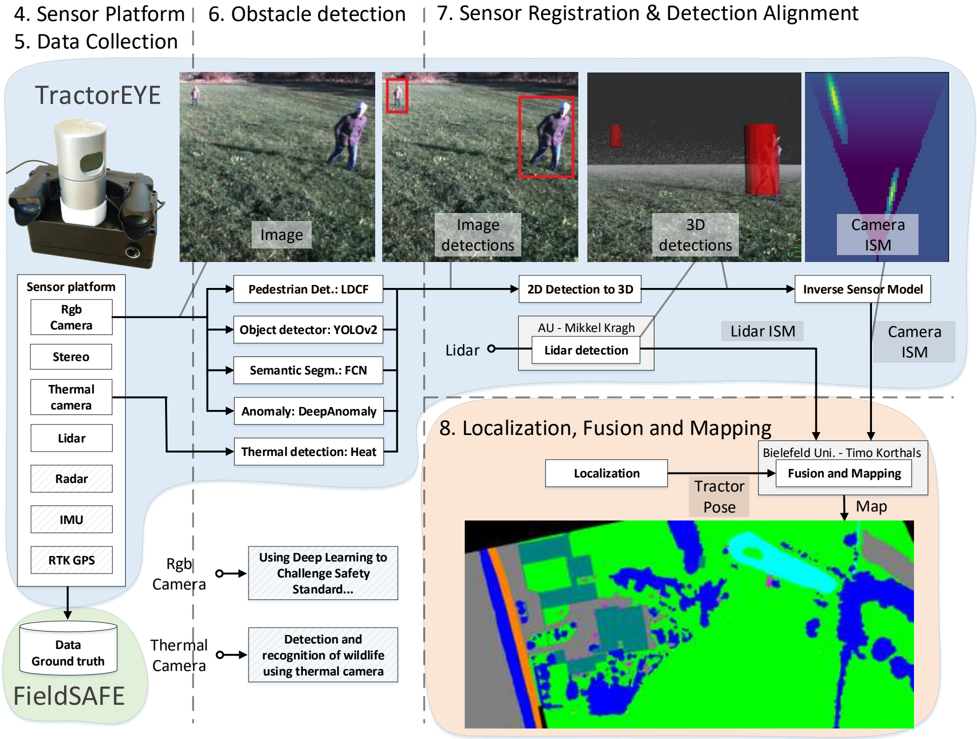


Figure 1.1: Summarizing contributions of the thesis. FieldSAFE is the outcome of *Sensor Platform* and *Data Collection*. **TractorEYE** is the outcome of *Sensor Platform, Data Collection, Obstacle Detection, Sensor Registration & Detection Alignment*. Images are captured by the sensor platform. Obstacle detections are performed on image and other sensor data. These detections are represented in a common format (3D detections or inverse sensor models). Detection information is finally fused in a map.

hydrants and wells are placed and humans are lying, sitting, standing and walking in the field. Drones are used before, under and after data collection to gather ground truth of static elements and moving humans in the field .

Obstacle Detection State-of-the-art detection algorithms have been investigated for autonomous vehicles in agriculture. Namely, a pedestrian detector "*Local Decorrelation For Improved Detection*" (LDCF), three deep learning object detectors (YOLO, YOLOv2 and faster R-CNN), a fully convolutional neural network for semantic segmentation (FCN) and *DeepAnomaly*. DeepAnomaly is real-time deep learning-based anomaly detector for detecting distant, heavy occluded and unknown obstacles in the field. For a human use case in agriculture, it performs better than state-of-the-art (YOLO, Faster R-CNN, FCN). Additional two algorithms have been proposed. One for detecting a specific ISO-specified barrel "Using Deep Learning to Challenge Safety Standard" and one for detecting wildlife from an UAV "Detection and Recognition of Wildlife using Thermal Camera".

Sensor Registration & Detection Alignment The purpose is twofold. Sensor registration involves determining extrinsic and intrinsic parameters for all sensors. A thermal calibration panel is proposed to calibrate thermal camera and rgb camera. Detection alignment is procedures to map detection information from various sensors and algorithms into a common format. This is done either by representing detections as 3D coordinates – defined by the safe protocol GH8 – or Inverse Sensor Models to later enable detections to be fused into a map

Localization, Fusion and Mapping Localization, fusion and mapping is a set of procedures to fuse information from seven detection algorithms – across four sensors – into a map[2]

FieldSAFE is the outcome of contributions in *Sensor Platform* and *Data Collection*. FieldSAFE is a multi-modal dataset for detection of static and moving obstacles in agriculture. Ground truth of static and moving obstacles are available as an annotated orthophoto and GPS coordinates through time for moving obstacles.

TractorEYE is the outcome of contributions in *Sensor Platform, Data Collection, Obstacle Detection, Sensor Registration & Detection Alignment*. TractorEYE is a multi-sensor detection system for autonomous vehicles in agricultural. ROS-packages have been developed to run a total of six detection algorithms (LDCF, YOLOv2, FCN, DeepAnomaly, dynamic heat detection and a SVM classifier for lidar[34][3]) and additional algorithms to transform detection information into a common format by either mapping detections to 3D positions or with an Inverse Sensor Model.

1.2 Organisation of Dissertation

The thesis is organized into three parts.

Part I: A Survey of deep learning algorithms for image recognition. Survey is divided into five headings; *Convolutional Neural Networks (CNN) for Image Classification, Object Detection, Semantic Segmentation* and *Efficient Deep Learning*

Part II: Summary of thesis work and contributions presented in Figure 1.1. Research contributions are divided in five subjects; *Sensor Platform, Data Collection, Obstacle Detection, Sensor Registration & Detection Alignment* and *Localization, Fusion and Mapping*

Part III: Attachment of 10 selected papers.

[2] Fusion and mapping have been developed in collaboration with Timo Korthals from Bielefeld University

[3] This was developed by Mikkel Fly Kragh and is not covered in thesis summary.

Survey Part I

2 Deep Learning for Image Recognition

Deep learning and image recognition have been a main research area of this thesis. This section is dedicated to a survey on deep learning for image recognition.

2.1 Deep Learning

Deep learning have dramatically improved multiple state-of-the-art tasks [35] in image data (such as image classification [36], object detection [37], semantic segmentation [19], face detection [38], face recognition [25]) and sequential data (speech recognition [39], detection Cardiac Arrhythmia [27]). Deep learning methods consist of multiple layers of simple and non-linear operations that for each layer, transform the current layer into a higher representation/abstraction. Using typically a batch of samples, the backpropagation algorithm specifies how to update the internal model parameters to optimize for a given task [35]. Ideally, the model will iterate and converge to some generalizable, local minimum solution.

Multiple aspects separate deep learning from many previous conventional/classic machine learning methods.

One important aspect is that Deep learning models are able to be trained end-to-end. This powerful concept, allows a single model to map high dimensional (and potentially raw) data to a desired output. This avoids the time consuming task of engineering specific handcrafted features and combining subsequent steps of processing to solve a specific case. The model is simply optimized to automatically create optimal processing steps and high abstraction features [40] to solve a given task. This is illustrated in Figure 2.1.

The modular structure and end-to-end training, makes deep learning generic/general purpose, allowing the same language, modules, optimization procedures and frameworks to be shared

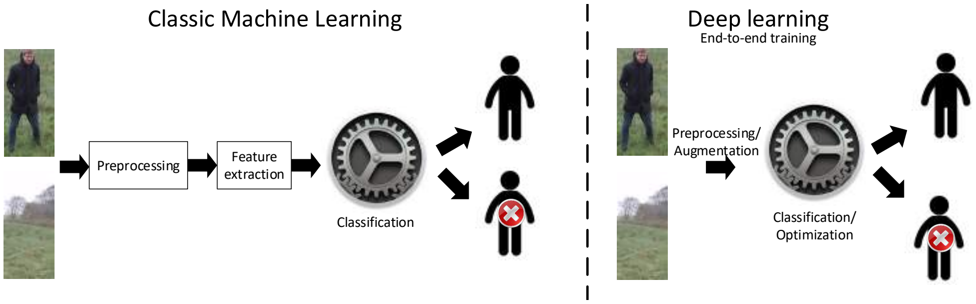


Figure 2.1: Classic machine learning vs Deep learning. Classic machine learning requires a preprocessing and a feature extraction step before optimizing a classifier. Deep learning is trained end-to-end.

across multiple machine learning discipline (image, time-series/audio and natural language processing) in many domains (agriculture, medicine, automotive, economics).

Another key aspect of Deep learning models is the high capacity and that a model may contain millions of parameters without overfitting [41] - even above 100 million parameters [42]. This enables them to consistently learn from new data as demonstrated in Figure 2.2. This assumption has recently been confirmed by Google using 300M images to train a model [43].

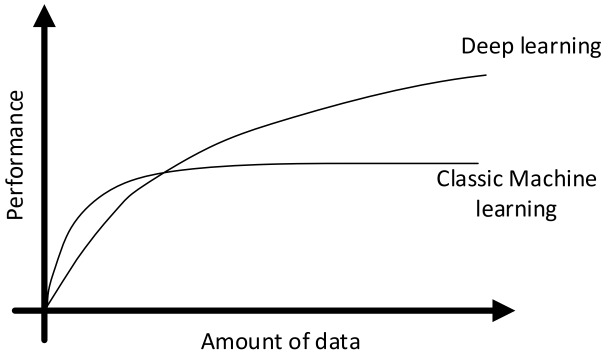


Figure 2.2: Deep learning learns from more data. Inspired by Andrew Ng

In image detection, high capacity enables models to classify many object types and to model intra-class variation of a specific class caused by occlusion, deformation, camera viewpoint variation, illumination variation and scale variation. Most importantly, deep learning based methods achieve state-of-the-art performance in many benchmarks [32, 33, 44–47] and deliver human-level or professional-level performance in a range of new areas such as traffic sign classification [24], image classification [23], speech recognition [48], conversational speech recognition [26], lip reading [21], skin cancer classification [22], cardiac arrhythmia detection [27].

A drawback of CNN-based models is the requirement of enough data to generalize for a given task. Transfer learning [49, 50] is the concept of fine-tuning an already trained model to a new task with significantly less samples. However, the task and data of the pre-trained model should for especially high abstraction layers be similar to the new task. Another drawback is the hierarchical structure of deep learning models that causes the processing load to grow. Utilization of GPU, low cost of computing, deep learning accelerated libraries (cuDNN) and model improvements have reduced the processing time.

Especially two powerful neural network structures have pushed improvements of deep learning. Recurrent Nets (LSTM [51] and GRU [52]) for processing of sequential data and natural language processing and Convolutional Nets [53, 54] in images. Especially a combination of the network structures is able to shape powerful applications for speech recognition [39, 48], video activity recognition, image captioning, video description [55], object tracking in video [56], lip reading [21], head pose, facial landmark localization [57]. Finally, Generative Adversarial Networks (GAN) [58] and reinforcement learning [28] have presented interesting results for deep learning based methods.

Convolutional Neural Network (CNN) have demonstrated great performance for supervised image recognition. The following section covers, CNN-based networks for supervised image recognition with the following headings; image classification, object detection, semantic segmentation and efficient deep learning.

2.2 Convolutional Neural Networks (CNN) for Image Classification

Image classification is the task of providing object class or a set of object classes that are presented in an image. ImageNet [44] includes an image classification benchmark with a large dataset of more than 1 million images and 1000 object classes. The benchmark has played a key role for CNN-based methods and became a prestige competition for both universities and mega-corporations (Google, Microsoft and Baidu) to publish and improve on existing results/methods.

The breakthrough of deep learning is often credited to AlexNet [36] in 2012 for winning ImageNet by a large margin using a CNN consisting of convolutional layers [53, 54], a Rectified Linear unit (ReLU) [59] as activation function, max-pooling for subsampling and two fully connected layers with dropout [60] before the final softmax layer. The network is trained using backpropagation [61–64] and Stochastic Gradient Descent (SGD).

Core concepts of AlexNet [36] such as convolutional layers, SGD, and backpropagation have been introduced decades ago [**?**] and used in LeNet [54, 65]. The sudden improvement of CNN in 2012 can be explained by the existence of large annotated datasets, utilization of GPU hardware (with sufficient memory) and network improvements such as dropout and ReLU that enables a model to converge on natural images.

In 2013 the image classification competition was won with a ZFNet [40] network by adding only minor adjustments to AlexNet. More interestingly, they managed to visualize high abstraction features and to demonstrate the power of an ImageNet pre-trained model on other small image datasets. In 2014, the best single model performance was achieved with a VGG network [42]. VGG uses a simple network structure with only 3×3 kernels in 16 convolutional layers. Though the best single model was VGG, the winning team of 2014 used an ensemble of GoogLeNet [66] networks. The main component of GoogLeNet is an inception module using 1×1 convolutions [67] to reduced computations and parameters of the network. Global average pooling [67] is used prior to the first FC-layer to dramatically reduce the number of parameters in the first FC-layer compared to e.g. AlexNet and VGG. GoogLeNet has improved multiple times [68–70]. In following years, network improvements were derived from the concept, that more layers are better – also stated by GoogLeNet "We need to go deeper". However, training a deeper network is not as simple as adding more layers [71, 72], and training becomes difficult for more than five layers [73]. In GoogLeNet more layers are successfully stacked by adding *auxiliary* classifiers in intermediate layers. In VGG, depth is increased by first training a moderate model and then iteratively concatenate more layers and retrain.

The degradation problem [72], states that the accuracy performance of deep CNNs will eventually saturate and degrade rapidly if more layers are constantly added. In training, the input and gradients must flow through the whole network without vanishing or exploding. Careful initialization has proven to avoid the issue of vanishing gradients. He Kaiming demonstrated the importance of weights initialization in [23] to match ReLU / PReLU activation functions, which enabled networks with up to 30 weight layers to be trained from scratch. Further research was presented in [74] to demonstrate the consequence of too small or too large weights for an increasing number of layers. Batch normalization [68] makes weight initialization less critical by normalizing the output of each convolutional layers before activation. Batch normalization is now commonly used in CNNs to improve accuracy and train using fewer epochs. A remaining problem of traditionally stacked networks is that low-level features and output gradients are forced to propagate through all intermediate layers in the network.

To get beyond 100 layers in a network, information must be able to flow unobstructed both forward and backwards through the network. Forward to allow useful low level features to be used in higher layers or directly in the classification layer. Backwards to have gradients train directly on low level layers and to avoid vanishing gradients. Highway networks [71] are first to address the information flow problem using gating units to stack 100 layers. Gating units allow information to jump multiple layers using information highways. The information problem is also handled in ResNet network [72]. ResNet is the winner of the ImageNet competition in 2015 with 152 stacked layers. A parametric free identity mapping between layers allows information to always be passed through the network. In Highway network information is potentially gated from the "highway". ResNet is improved in [75] by moving the ReLU activation outside the skip connection. This ensures that the flow of information is not constrained to only positive values and 200 layers are successfully stacked for image classification on ImageNet.

DenseNet [76] improves information flow by densely connecting all layers of the network by concatenating feature maps of all previous layers. This would presumably cause the number of feature maps, parameters and computations to explode for an increasing number of layers. Somehow counter-intuitively, DenseNet introduces only a small set of kernels for each layer and requires in practice less parameters and computations than a ResNet-based architecture with similar classification accuracy. The reason for this is that low level features / states are concatenated in a separate lane in DenseNet. The number of newly introduced kernels for each layer should only represent new features and are not required to also represent features from all previous layers.

The pursuit of extreme depth (>1000 layers) is computationally inefficient and designs with wider architectures are starting to emerge such as ResNeXt [77] and Wide ResNets [78]. Merging calculation into fewer large layers runs more efficiently and provides similar results as extremely deep networks.

2.3 Object Detection

An object detection algorithm must both recognize and localize instances of one or more specific object classes in an image or a video.

The first real-time applicable object detection algorithm was introduced by Viola and Jones [79, 80] using haar features, sliding window and AdaBoost [81]. Accuracy is improved in [82] using Histogram of Oriented Gradients (HOG) and a SVM classifier for pedestrian detection. A better and faster pedestrian detection was later introduced by Dollar [83] and improved in multiple publications [17, 83, 84]. All of the above methods uses a fixed aspect ratio bounding box to detect only one object type. Various aspect ratios and object types are detectable by Deformable Parts Model (DPM) [85, 86]. DPM have been popular in the late 2000s for object classification tasks such as the PASCAL VOC object detection competition with 20 object classes. The performance is, however, not near the performance of recent deep learning based object detection methods.

Deep learning algorithms have dramatically improved state-of-the-art on object detection [35]. Deep learning object detection was – similar to image classification – kick-started by ImageNet in 2013, when the benchmark was extended with an object detection task with 200 object classes across 400,000 images. The possibility to constantly evaluate and compare results has shifted the research community to other benchmarks. Especially the older PASCAL VOC [45] benchmark with 20 object classes in 11,000 images and MS COCO [87] with more than 300.000 images across 80 object classes is frequently updated with the recent state-of-the-art. Many automotive-related dataset with associated online benchmarks are popular such as Kitti [32] and CityScapes [33].

CNN-based object detectors consist of typically a fully convolutional network (feature extractor) followed by a task-specific recognition part used for e.g. image classification, object detection, semantic segmentation or instance segmentation. Adapting the terminology from [88], the feature extractor is defined as the *backbone* part of the network and the task specific recognition is defined as the *head* of a network. In the survey of object detection, we will only describe the network head as this can be used with any backbone architecture (LeNet, AlexNet, VGG, GoogLeNet, ResNet, DenseNet, ResNeXt).

A naive CNN-based object detector can use an image classification network to perform sliding window across the image at multiple scales. This can be implemented more efficiently by reshaping fully connected layers to convolutions as described in [19, 38, 89]. *Sermanet* won with OverFeat [89] the ImageNet localization challenge in 2013 using a sliding window approach combined with four regression outputs to improve localization.

Ross Girshick proposes in [37] to combine Regional proposals with a CNN (R-CNN) to win the 2014 ImageNet detection benchmark with a large margin over OverFeat. A region proposal algorithm such as *Selective Search* [90] or later the faster *EdgeBox* [91] is used for generating many potential object regions. Each region is squeezed to a fixed size, forwarded through a network and classified using a Support Vector Machine (SVM). Region proposal methods have improved in several iterations [88, 92–94] to a more unified end-to-end system that delivers state-of-the-art accuracy for object detector [88, 95]. Kaiming He proposes in [92] to use *Selective Search* in combination with a Spatial Pyramid Pooling (SPP)-module. Unlike R-CNN, the image is only forwarded through the convolutional layers once, and the SPP-module extracts features from a region and performs classification. A Region-Of-Interest (ROI) module is introduced in Fast R-CNN [93] to pool features for each region to a feature vector. The feature vector is used in a multi-task loss function to classify objects and improve localization of region using regression. A Region Proposal Network (RPN) was introduced in Faster R-CNN [94] to make a complete end-to-end system of close to real-time performance. The RPN is a convolutional module that provides k bounding boxes (potential objects) at each position. Each bounding box is defined by an anchor – a prior bounding box shape with predefined scale and aspect ratio – and two outputs; a box-classification output of $2 \times k$ values (box or not box) and a box-regression output of $4 \times k$ to specify the location relative to an anchor. Faster R-CNN is in Mask R-CNN [88] extended with a new instance segmentation branch to win the MS COCO 2016 challenge in object detection and instance segmentation.

Region-based detectors such as Faster R-CNN provide state-of-the-art accuracy performance and are in fact also fast. However, the underlying concept of Region-based methods is that the detectors require a second stage to do per-proposal classification [95]. Another branch of CNN-based detectors seek to improve efficiency of networks by only running a single forward pass [18, 96–100] as stated in papers titled “You Only Look Once” (YOLO) [18] and “Single Shot MultiBox Detectors” (SSD) [100]. Recently, single forward or fully convolutional networks have adapted anchor boxes [99–101]. Demonstrating that the concept of anchor boxes currently is preferable for object detection. A Feature Pyramid Network (FPN) is a generic concept adapted by state-of-the-art detectors to scales up feature maps to better detect small instances [102] and to become more invariant to scale. FPN upscales intermediate feature maps and merges them using addition. This enables the same object detector to be used at multiple scales of the feature pyramid.

2.4 Semantic Segmentation

Semantic segmentation provides richer information than object detection by classifying all pixels in an image.

Similar to image classification and object detection, the performance of semantic segmentation networks have been pushed forward by public benchmarks such as Pascal VOC [45]. New and interesting datasets for semantic segmentation are MS COCO [87] with instance segmentations and CityScapes [33] in the automotive domain. Pascal Context [103], and MIT Scene Parsing Benchmark [47] with whole scene annotations.

CNN-based semantic segmentation networks can be separated in *backbone* and *head* - also defined as encoder and decoder for semantic segmentation. The backbone uses a CNN-based model that potentially has been trained for image classification, and the head network is responsible for upsampling feature maps to provide more fine-grained predictions and better boundaries.

A Fully Convolutional Network (FCN) is presented in [19] for semantic segmentation. A CNN network (VGG [42]) is first trained for image classification using ImageNet. The CNN is converted to a fully convolutional network by discarding the final classification layer and reshaping fully connected layers to convolutional layers. The output feature maps are upsampled using deconvolutions or backward convolutions and merged with intermediate feature maps to provide fine-grained predictions.

DeepLap [104–106] has demonstrated state-of-the-art detection performance on Pascal VOC by combining a CNN for semantic segmentation with Fully Connected Conditional Random Fields (CRF) to better classify object boundaries. DeepLap uses *atrous* convolutions (dilation factor > 1) instead of upsampling. In [107] a CRF is modelled with a Recurrent Neural Network to enable end-to-end training. Recently, the CRF post-processing step has been removed by state-of-art algorithms - also by DeepLap [106].

Another semantic segmentation architecture is SegNet [108]. SegNet is symmetric in the sense that each encoder layer has a corresponding decoder layer. SegNet uses indices from encoder max-pooling layers in the decoder upsampling layers. A symmetric architecture has also been used in U-Net [109] for Biomedical Image Segmentation. As for U-Net, intermediate feature maps are forwarded to similar level feature maps in the decoder. Also interesting for U-Net is that only valid convolutions are used to avoid artefacts from zero-padding.

2.5 Efficient Deep Learning

Deep learning methods are computationally expensive and procedures are required to make systems more cost efficient. Especially in the automotive industry, the requirement for real-time operation on embedded platforms have made power consumption, memory usage and processing time important network parameters. Efficient algorithms can be optimized with hardware, software libraries, and network architecture.

Concepts for providing more efficient deep learning models in compute, power and memory have often been introduced or used by state-of-the-art accuracy performance networks. In AlexNet [36], the highly compute efficient ReLU [59, 110] activation function is able to execute and converge models faster than the sigmoid function commonly used previously in neural networks. GoogLeNet [66] uses two important concepts initially introduced in [67]; global average pooling and 1×1 kernel convolutions. In initial CNN models (LeNet, AlexNet, ZFNet, VGG), the final convolutional layer is connected to an FC-layer. In e.g. VGG the first FC layer uses 103 million parameters, corresponding to 74% of all weight in the network. Global average pooling will average across each channel in the feature map before connecting to an FC-layer and reduce the number of parameters dramatically in the first FC layer e.g. by a factor of 49 for VGG. The use of 1×1 convolutional kernels became popular with GoogLeNet to reduce the number of computation and have been a key component in many efficient models to – depending on publication – flatten, factorize, compress, branch or decompose feature maps [66, 72, 76, 77, 111, 112]. In GoogLeNet, feature maps are compressed to especially reduce computations of expensive 3×3 and 5×5 convolutions. In ResNet, the bottleneck architecture is introduced by substituting a pair of 3×3 convolutions to a module of 1×1, 3×3 and 1×1 convolutions. The first 1×1 convolution reduces the number of channels and the final 1×1 restores the number of channels. This allows a small number of 3×3 kernels to be run on a feature map with only a few channels while still maintaining a high number of input and output features / channels. In ResNeXt [77], 1×1 convolutions create up to 32 branches of 3×3 convolutions that are concatenated and again expanded using 1×1 convolutions. DenseNet uses 1×1 to implement both bottleneck and compression. However, the most important concept to both improve convergence, accuracy and computations is that initial feature states can be passed directly to any layer. Batch normalization makes inference slower and requires more parameters, but is important to train models more efficiently. Models are able to converge more easily, require far less iterations and obtain better accuracy.

Models have also been developed to improve efficiency with similar or insignificant drop in accuracy [113]. SqueezeNet [114] combines fire-modules and global average pooling with DeepCompression techniques – this is covered in the next text section. The Fire-module is related to the Inception-module by using 1x1 kernels to squeeze feature map into two branches (1×1 and 3×3 kernel convolutions) that are afterwards concatenated. Compared to AlexNet, SqueezeNet is able to obtain similar classification accuracy using 50× fewer model parameters without DeepCompression and 510× less with DeepCompression. Results are remarkable. However, a comparison to a network with inception-modules and global average pooling is more comparable to state-of-the-art in terms of accuracy. Furthermore, the aim of the paper is to reduce the number of model parameters. However, the number of model parameters should not be confused with the actual model memory usage. Memory usage is a more critical GPU hardware constraint for state-of-the-art networks. An improved SqeezeNet model has been released (1.1) with 2.4× fewer computations and provides also a speedup compared to AlexNet. In MobileNet [112] an efficient architecture is presented by exchanging traditional 3×3 convolutions with a micro-architecture of depth-wise separable filters [115] and a 1×1 convolution. This simple concept is able to improve processing speed and reduce the number of model parameters. MobileNet-based networks are able to obtain similar classification accuracies as AlexNet, SqueezeNet, VGG and GoogLeNet with respectively a factor of 9.5×, 22.3× ,26.9× and 2.7× less Mult-Add operations.

A set of tools defined as DeepCompression have been presented by Song Han [116] which includes network pruning [117], quantization, and Huffman coding. DeepCompression demonstrates that networks are highly overrepresented and many connections are in fact redundant. In [117] it is demonstrated that 92.5% of all weights or connections are redundant. Quantization and huffman coding reduces the required bits to represents weights thereby increasing the compression rate from 19 to 49. The overall drawback of DeepCompression is that pruning, quantization and huffman coding are hardly parallelizable on a GPU. Claims of 3× to 4× layers-wise speed up is slightly misleading as this is only true for fully connected layers with a batch sizes of 1. Similar improvements are not to be expected for convolutional layers where “only” roughly 1/3 of all weights are redundant and sparsity is harder to parallelize for convolutional layers. Furthermore FC-layers runs very efficient for large batches and typically only a small fraction of processing time is used on FC [118] - especially for networks using global average pooling after the last convolutional layer. The work of Song Han is theoretically very interesting and even an ASIC chip have been developed in [119] by Song Han to demonstrate incredible speed performance and power savings. However, the concepts presented in DeepCompression are to my knowledge not efficiently applicable on GPU using common software libraries.

In hardware, the reduction of cost per flop and utilization of GPU for deep learning have been essential in the breakthrough of deep learning in general. Traditionally, GPUs used in research have required double precision floating-point format (64 bit). Deep learning models are less dependent on bit precision and similar accuracy can be obtained by using 32 bits or 16 bits. The Pascal GPU architecture by Nvidia allows 16-bit floating point operations (half precision) to be executed twice the rate of single precision. The software package TensorRT by Nvidia also provides 8 bit integer precision. Even binary networks have been demonstrated in [120] to theoretically reduce memory usage by roughly a factor of 32 and computations by a factor of 58. Lately, more dedicated deep learning hardware has been introduced in the very fast and energy efficient Tensor Processing Unit (TPU) by Google [121] and a Deep Learning Accelerator (DLA) by Nvidia in the Volta architecture. Nvidia have been the key hardware supplier to deep learning due to high performance, fast adaption and because many deep learning frameworks primarily support software libraries that runs on Nvidia GPUs (cuda and cuDNN). However, hardware competitors are starting to challenge the monopoly of Nvidia. Looking strictly on hardware, AMD has recently become competitive to Nvidia. The new AMD Radeon Vega GPU is cheaper with slightly higher brute force performance than Nvidia Titan Xp and the new Ryzen Threadripper CPU with 64 PCI 3.0 lanes can be connected to four Vega or Titan GPUs.

Summary Part II

3 Overview of Summary

Figure 3.1 presents thesis contributions divided in chapters and as processes/modules. The output of each module is visualized by an example, showing the raw sensor data (image), detections in the image (image detections), detections in world coordinates (3D detections) and registered detections using a Camera Inverse Sensor Model (ISM). Finally, fusing and mapping is performed to generate a final output map (Map).

Chapter 4: Multi-sensor platform with exteroceptive sensors (rgb, thermal, stereo, radar and lidar) and proprioceptive sensors (IMU and GPS).

Chapter 5: Overview of all field trials and generation of ground truth data.

Chapter 6: Multiple detection algorithms for RGB and thermal cameras. Five of the seven algorithms are applicable for TractorEYE. The remaining two modules; "*Using Deep Learning to Challenge Safety Standards*" and "*Detection and Recognition of Wildlife using Thermal Camera*" are standalone contributions not used by TractorEYE.

Chapter 7: Sensor registration and detection alignment. Calibration and registration of rgb/stereo camera, thermal camera and lidar. Detections information is aligned by mapping detections from multiple algorithms and sensors to either 3D position or inverse sensor models (ISM)

Chapter 8: Describes how ISMs from TractorEYE are fused. Localization is used for estimating tractor pose and fusing information into maps.

Chapter 9: Presents a concise overview of all contributions.

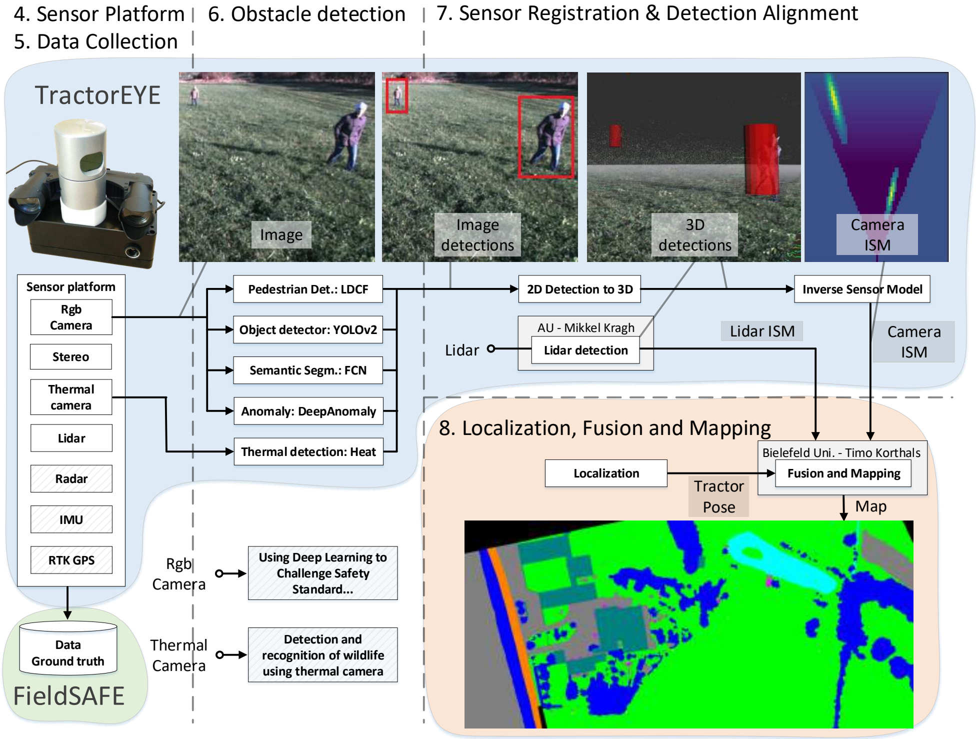


Figure 3.1: Summarizing contributions of the thesis. FieldSAFE is the outcome of *Sensor Platform* and *Data Collection*. TractorEYE is the outcome of *Sensor Platform, Data Collection, Obstacle Detection, Sensor Registration & Detection Alignment*. Images are captured by the sensor platform. Obstacle detections are performed on image and other sensor data. These detections are represented in a common format (3D detections or inverse sensor models). Detection information is finally fused in a map.

4 Sensor Platform

A single sensor technology is unlikely to guarantee safe operation of autonomous vehicles in agriculture. Depth-based sensors such as stereo cameras and lidars are popular in robotics to map surroundings and detect obstacles. Depth-based sensors may use generalizable detection heuristics by e.g. simply avoiding elements that obstruct the field-of-view of the sensor or by detecting obstacles that protrude the ground surface. Similar heuristics are also useful in agriculture to detect large elements that protrude either ground or crop surface. However, in agriculture a vehicle must be able to traverse areas with crops and obstacles may reside inside crops. This will make depth-bases sensors less reliable for detecting especially small obstacles that resides below or just above an uneven crop surface such as a kid, an animal or a sitting / an unconscious human. A depth-based sensor such as a multi-beam Velodyne lidar is very reliable and provide excellent perception capabilities when obstacles are protruding. A stereo camera provides both depth and color point clouds but are less reliable.

Humans rely mainly on visual information to perceive surroundings and navigate a vehicle. Rgb cameras record visual information and should in principle provide sufficient information to perceive surrounding. However, problems for rgb cameras are the limited accuracy in localizing obstacles precisely, high vulnerability to weather/lighting conditions and they will hardly recognize animals that by nature are visually camouflaged. Thermal cameras have a potential for detecting animals in the field. Though, a thermal camera is very dependent on weather conditions or temperatures of obstacles and surroundings.

Advantages and disadvantages of each sensor modality is further addressed in P[6, 8]. This includes a comparison between sensors in terms of range, resolution, cost, robustness to light & weather changes and the ability to detect camouflaged, protruding/non-protruding and various obstacles.

Multiple sensor technologies must be combined to increase detection performance and to introduce redundancy. This chapter proposes a multi-sensor platform for capturing such data. The sensor platform is presented in P[3, 6, 8] and used for data acquisition in P[1, 4, 5, 7, 10, 11]. The platform has been under constant development in the thesis and iterated into two sensor platforms: The SuperSensorKit for large field trials and the more compact design MiniSensorKit used in TractorEYE. The platform runs on Linux (Ubuntu) as operating system and Robot Operating System (ROS) as middleware to record and connect sensors. Contributions are presented in Figure 4.1.

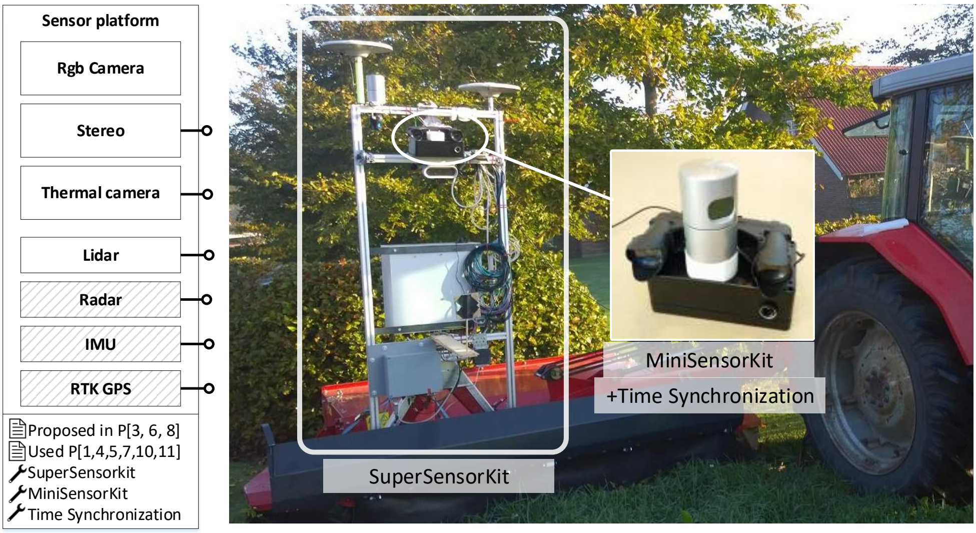
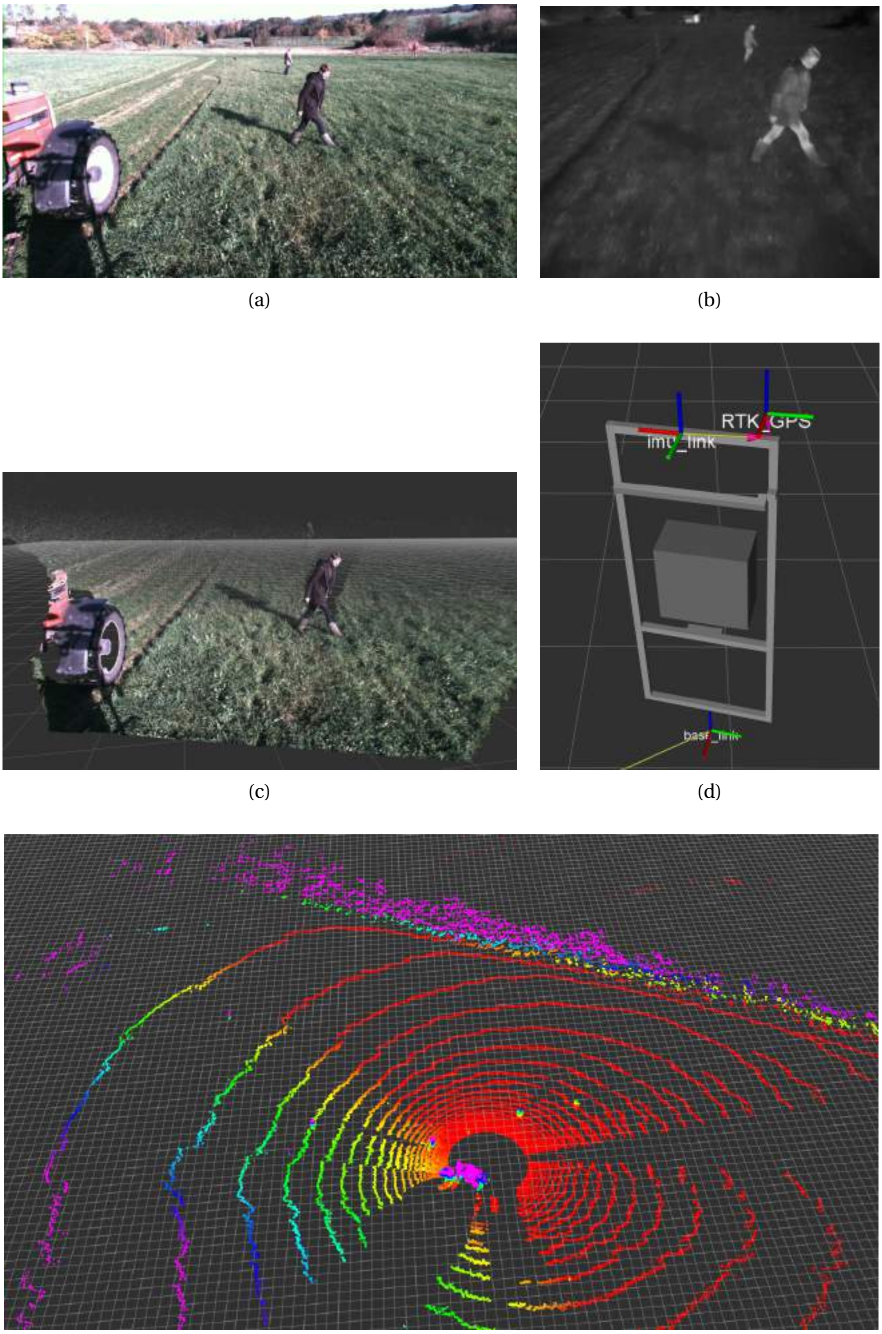


Figure 4.1: Contributions of Chapter 4. A sensor platform with multiple sensor technologies have been proposed in P[3, 6, 8] and used in P[1, 4, 5, 7, 10, 11]. The sensor platform is arranged in two platforms; SuperSensorKit and MiniSensorKit. The MiniSensorKit uses a micro controller to synchronize stereo camera, thermal camera and lidar.

4.1 Sensors

We have selected a broad range of sensor modalities including rgb, stereo, thermal, multi-beam lidar and radar (Figure 4.2) to compare and evaluate the performance of each modality for autonomous vehicles in agriculture. The platform has been described in P[3, 6, 8], and modalities are evaluated qualitatively. In P[4, 10] sensor modalities have been evaluated and compared individually and in pairs. Exteroceptive sensors (GPS and IMU) are used for localization to estimate pose and position of the sensor platform.



(a)

(b)

(c)

(d)

(e)

Figure 4.2: Selected modalities (a) rgb camera (color image) (b) thermal camera (thermal image) (c) stereo matching (color point cloud) (d) visualization of robot pose (GPS and IMU) (e) multi-beam lidar (point cloud)

The modalities have remained constant in the SAFE project, however the actual sensor models have changed as new experience and sensors have been acquired. The sensors that have remained since P[6] is presented in Figure 4.3 (a)-(d) and listed below:

a) Rgb camera: HD Pro C920 Webcam from Logitech (Silicon Valley, USA) with 1920 x 1080 pixels at 30fps

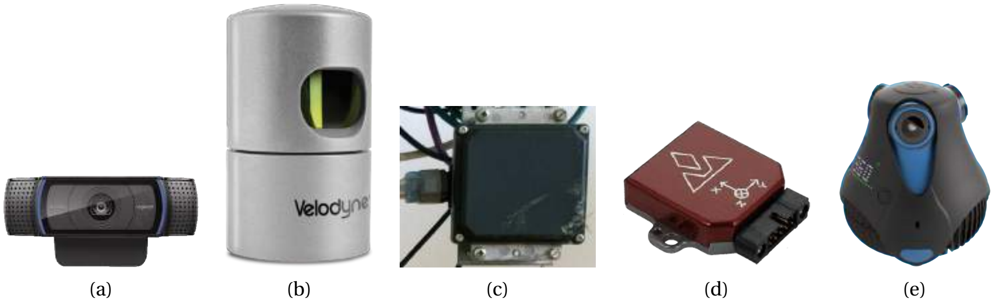
b) Multi-beam lidar: HDL-32E lidar from Velodyne (Morgan Hill, USA) a 32-beam laser scanner providing 70,000 points at 10 Hz with 1–100 m range.

c) Radar: Automotive Delphi ESR 64-target radar from Delphi (Washington, DC, USA)

d) IMU: VN-100 from Vectornav (Dallas, USA) providing synchronized three-axis accelerometers, gyros, magnetometers and a barometric pressure sensor

Later a 360-degree rgb camera was added to the sensor kit. Which is presented in Figure 4.3 (e)

e) 360 degree rgb camera: HD Giroptic 360-degrees from Giroptic (San Francisco, USA) providing 2048 x 1024 at 30 fps.



(a) (b) (c) (d) (e)

Figure 4.3: Sensors used throughout the SAFE project. (a) Logitech Webcam, (b) Velodyne Lidar, (c) Delphi Radar, (d) Vectornav IMU, (e) Giroptic 360 degree camera

4.1.1 Thermal Camera

Three different thermal cameras have been tested on the platform. The cameras are presented in Figure 4.4 and listed below and denoted T1, T2 and T3:

T1: FLIR A320 by FLIR systems (Wilsonville, USA) with a resolution of 380 x 240 pixels at 9 fps.

T2: HawkVision by Tonbo Imaging Inc (East Palo Alto, USA) analog IR camera providing 640 x 480 pixels at 25 fps. A 3D printed casing provided by DAS (Copenhagen, Denmark) to hold thermal camera and a analog-to-GigE converter by Pleora (Ottawa, Canada).

T3: FLIR A65 by FLIR systems (Wilsonville, USA) with a resolution of 640 x 512 pixels at 30 fps.

The first thermal camera (FLIR A320) was used in P[2] – published prior to the sensor platform – and also in the initial sensor setup in P[8]. The camera requires an additional computer with windows installed (software was developed in C# and .NET) and provided low resolution and framerate. Secondly, a higher framerate is desired to reduce the latency of a detection and better resolution to detect objects at further distances and to get a more fine-grained representation of obstacles. The second camera (HawkVision) provides higher resolution and framerate. Pleora analog-to-GigE is used for connecting to ROS using existing ROS packages. The output format of HawkVision is intensities and not absolute temperatures like FLIR A320. Additional funding enabled us to finally get FLIR A65. The FLIR A65 provides high resolution, high framerate, GigE for connecting to ROS and absolute temperatures.

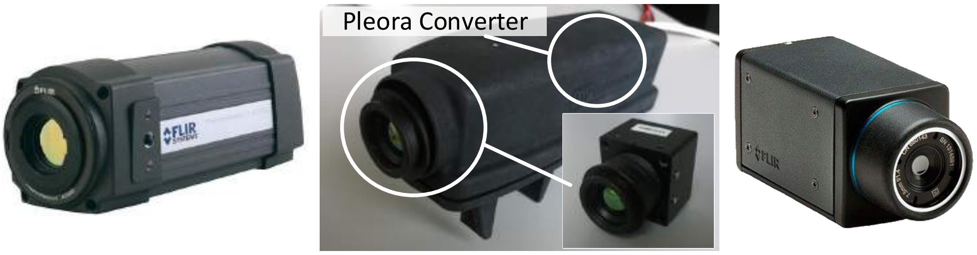


Figure 4.4: Thermal cameras used in the SAFE project. FLIR A320, HawkVision and FLIR A65

4.1.2 Stereo Camera

The three stereo cameras are presented in Figure 4.5 and listed below:

S1: New Imaging Technology (Paris, France) using NSC1003 CMOS sensors and providing 1280×1024 pixels at 25 fps. The camera uses global shutter and sensors are dynamic range (logarithmic). The sensors are separated by a narrow baseline of 5 cm.

S2: Two Flea3/FL3-GE-28S4C-C cameras from Point Grey (Richmond, Canada) are mounted to a solid metal frame with a baseline of 24 cm. The camera uses global shutter and provides 1928 x 1448 pixels at 15 fps. A small circuit board ensures that the cameras are hardware synchronized

S3: MultiSense S21 is a global shutter camera from Carnegie Robotics (Pittsburgh, USA) with a baseline of 21 cm. Stereo-matching is performed online and provides disparity maps at 2, 1 or 0.5 megapixels for 7.5, 15 or 30 fps respectively.

The first camera by New Imaging Technology is a logarithmic dynamic range camera used in P[8]. Dynamic range – is in principle – favorable for autonomous vehicles to capture the large intensity differences. Unfortunately, the image quality is too noisy and the narrow baseline of 5 cm makes distance estimations imprecise at far distances. These issues are handled with the Point Grey-based stereo camera with a much wider baseline and better image quality. Point Grey is used in P[6] and formed a satisfying solution for research, where stereo matching was performed offline after data collection. The third camera MultiSense S21 performs stereo matching online using an FPGA. The camera is plug-and-play and comes with good documentation and a well-functioning ROS package. MultiSense S21 is used in P[4].

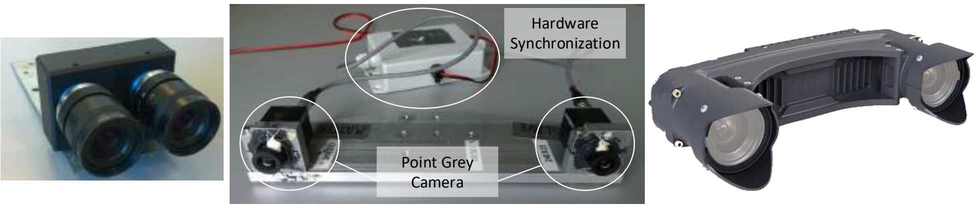


Figure 4.5: Stereo cameras used on the platform. New Imaging Technology, Point Grey and MultiSense S21

4.1.3 RTK GPS

Two Real Time Kinematic (RTK) GPS solutions have been used on the platform and are listed below:

- Single RTK GPS (AG GPS361) from Trimble (Sunnyvale, California, USA). Enhancing precision of GPS with up to centimeter-level accuracy
- Differential RTK GPS: Dual Antennas system from Trimble (Sunnyvale, California, USA). A BD982 receiver for determining heading vector between two antennas.

A single RTK GPS provides high accuracy localization information and have been used in a majority of publications of this thesis. A differential RTK GPS with dual antenna was used in FieldSAFE to also provide a heading vector.

4.2 SuperSensorKit

The multi-sensor platform defined as the SuperSensorKit is presented in Figure 4.6. The SuperSensorKit is a metal rack with two proprioceptive sensors (double RTK GPS and IMU) and the six exteroceptive sensors (webcam, 360 degrees camera, stereo camera, thermal camera, lidar and radar). Adjustable angle-parts allow sensors to be tilted for a desired angling. Sensors and computer are powered by a generator in the experiments. A large metal cabinet capsulates adaptors, a ruggedized computer *Robotech Controller 701* (Struer, Danmark) and GPS modules from the harsh environment. An A-frame allows the SuperSensorKit to be mounted to multiple setups as presented in Figure 4.7. To avoid mechanical noise from disturbing sensor measurements, rubber parts connect A-frame and rack to physically separate tractor and sensors by rubber.

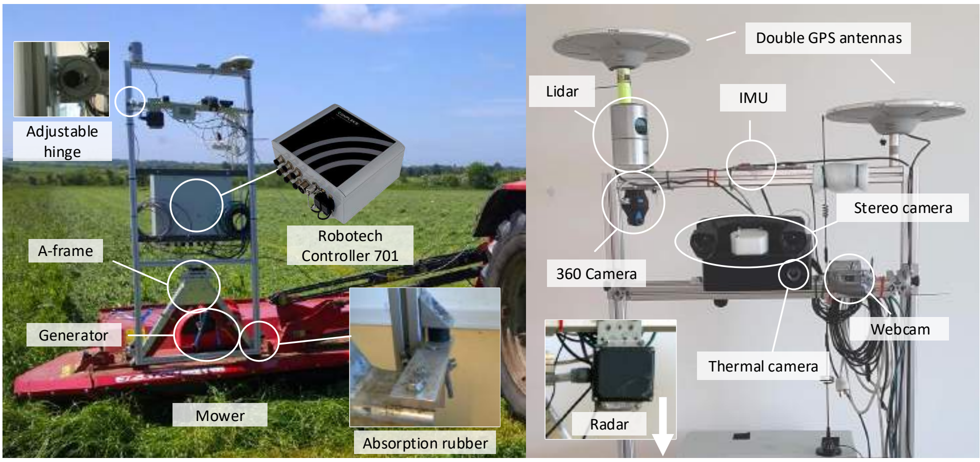


Figure 4.6: SuperSensorKit

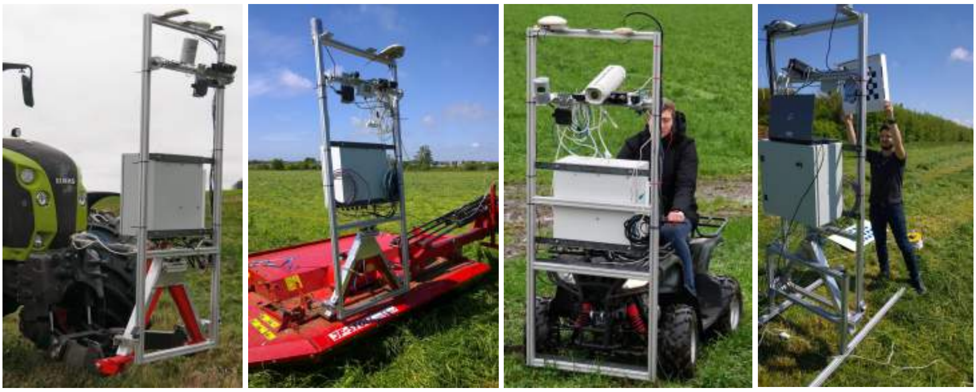
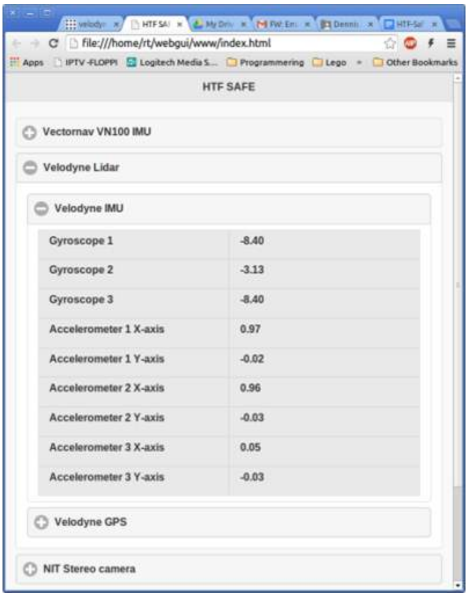


Figure 4.7: SuperSensorKit mounted on a tractor, implement, ATV and our lab mount.

4.2.1 Web GUI

A JavaScript-based web-GUI was been developed[1] to through WiFi easily monitor status of sensors, start/stop recordings and watch live camera feeds using e.g. a smartphone or computer. Though similar features are possible using ROS, the interfaces have proven to be very useful and convenient when running experiments and recordings in the field. The web-GUI is presented in Figure 4.8.



HTF SAFE

Vectornav VN100 IMU

Velodyne Lidar

Velodyne IMU

Gyroscope 1	-8.40
Gyroscope 2	-3.13
Gyroscope 3	-8.40
Accelerometer 1 X-axis	0.97
Accelerometer 1 Y-axis	-0.02
Accelerometer 2 X-axis	0.96
Accelerometer 2 Y-axis	-0.03
Accelerometer 3 X-axis	0.05
Accelerometer 3 Y-axis	-0.03

Velodyne GPS

NIT Stereo camera

Figure 4.8: Web GUI for data collection

[1] Web-GUI was developed by grad students and Mikkel Fly Kragh

4.3 MiniSensorKit

The SuperSensorKit has proven to be useful for data collection in large experiments. For small experiments, the SuperSensorKit is costly and bulky as it requires a large truck and at least two people to lift/mount it. A new sensor kit was developed with three sensors (FLIR A65, MultiSense S21 and Velodyne lidar). Stereo and lidar is mounted to the metal casing. The metal casing encapsulates the thermal camera, stereo camera and lidar wiring, a microcontroller for synchronizing sensors, a Velodyne lidar module and a DC-DC converter. Synchronization of sensors – meaning that data is captured in the same position in time – is important when sensors are calibrated / registered and in sensor fusion. The micro controller is responsible for signaling the stereo camera and the thermal camera simultaneously. The length of a second is precisely defined from a PPS signal provided by the Velodyne GPS. The design provides a ruggedized, portable and more water resistant setup for smaller experiments and robots. The sensors are firmly fixed to better maintain calibration. The MiniSensorKit is easy to mount and easily connected with three Ethernet ports – one for each sensor – and requires only one power source of 9V to 36V.

A CAD model was created in SolidWorks, see Figure 4.9. For data collection in field experiments, the MiniSensorKit is mounted to the SuperSensorKit frame as in P[4], see Figure 4.10. In Figure 4.11, the MiniSensorKit (aka TractorEYE) has been used on an actual robot called BallBot for online obstacle detection. View YouTube demo[2]

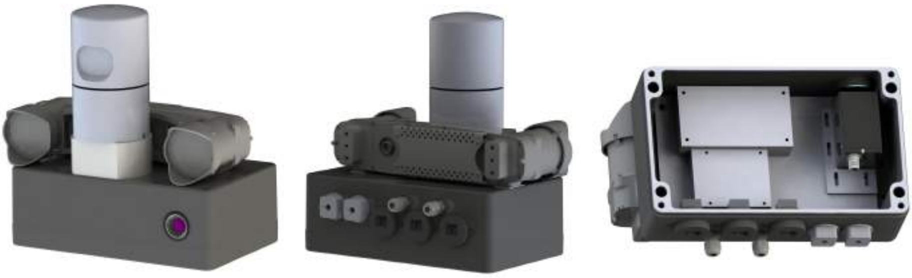
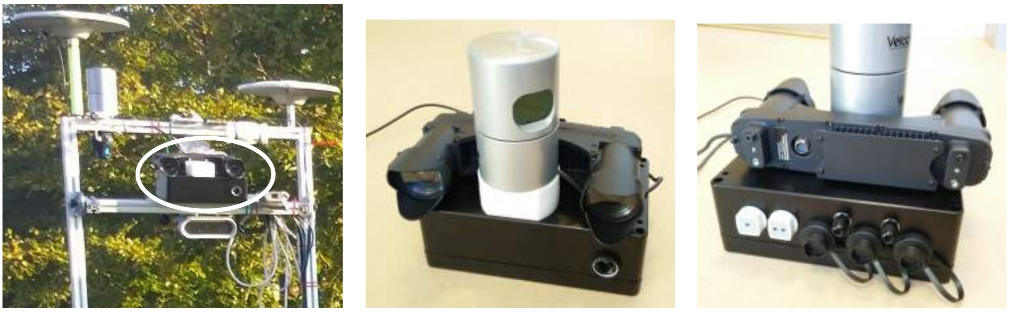


Figure 4.9: CAD model of front, back and inside MiniSensorKit

4.4 Concluding Remarks

A broad selection of modalities and sensors have been used. This includes mechanically mounting and setting up interfaces to one webcam, one 360-degrees camera, three stereo cameras, three thermal cameras, one multi-beam lidar, one radar, one imu and two gps'. The MiniSensorKit is a ruggedized and water-resistant casing which includes a thermal camera (FLIR A65), a stereo camera (MultiSense S21) and a multi-beam lidar (Velodyne 32E). A micro controller is used for hardware synchronizing sensors in time. The MiniSensorKit addresses important issues for a sensor platform to be used for autonomous farming vehicles. The platform is able to handle the rough environment in agriculture and mountable to relatively small autonomous vehicles. This have been demonstrated with BallBot (Figure 4.11) in a video demo[2].

[2] Video demo is available at YouTube *https://youtu.be/KDa_y-RfkhM?t=109*



(a) (b) (c)

Figure 4.10: MiniSensorKit. (a) MiniSensorKit mounted on SuperSensorKit rack, (b) MiniSensorKit front, (c) MiniSensorKit back

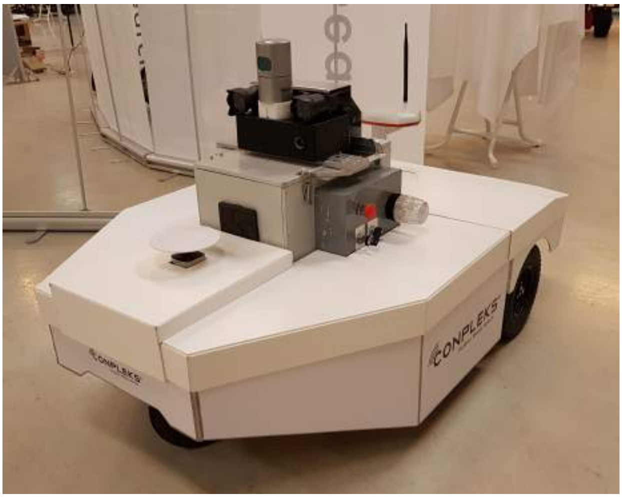


Figure 4.11: MiniSensorKit mounted on BallBot

The MiniSensorKit is the sensor platform used by TractorEYE. The SuperSensorKit is developed for research and includes the MiniSensorKit, a radar, a 360-degrees camera, localization and pose sensors (GPS and IMU). Sensors are mounted on a tractor mountable metal rack. The SuperSensorKit is used for creating the FieldSAFE dataset.

5 Data Collection

High quality and publicly available multi-modal datasets for detection in agriculture is needed to evaluate detection algorithm and push forward the development of autonomous vehicles in agriculture. This chapter describes field trials used for data collection in the thesis. A total of six field trials have been conducted and used in P[1, 3–11]. The six field trials are denoted D1-D6. The main contribution of this chapter is the final field trial D6 called FarmSAFE P[3] - A Agricultural Dataset with Static and Moving Obstacles.

5.1 Field Trials - Overview

Table 5.1 presents an overview of the area covered, publications, sensors and obstacles of each field trail. Figure 5.1 presents locations, sizes and shapes of fields.

5.2 Field Trials

5.2.1 Field Trial 1 - Børnebondegården

The aim of the first field trial is to test an early version of the SuperSensorKit and to get a first impression of sensor outputs in an agricultural context. Data was used in P[8].

Data were recorded on a small grass area at Børnebondegården near Horsens in November 2014. The SuperSensorKit was mounted to an ATV and most data was recorded statically or by slowly moving the ATV around. Data samples of animals and humans were collected in a broad set of scenarios including different walk patterns, age, clothing and posture.

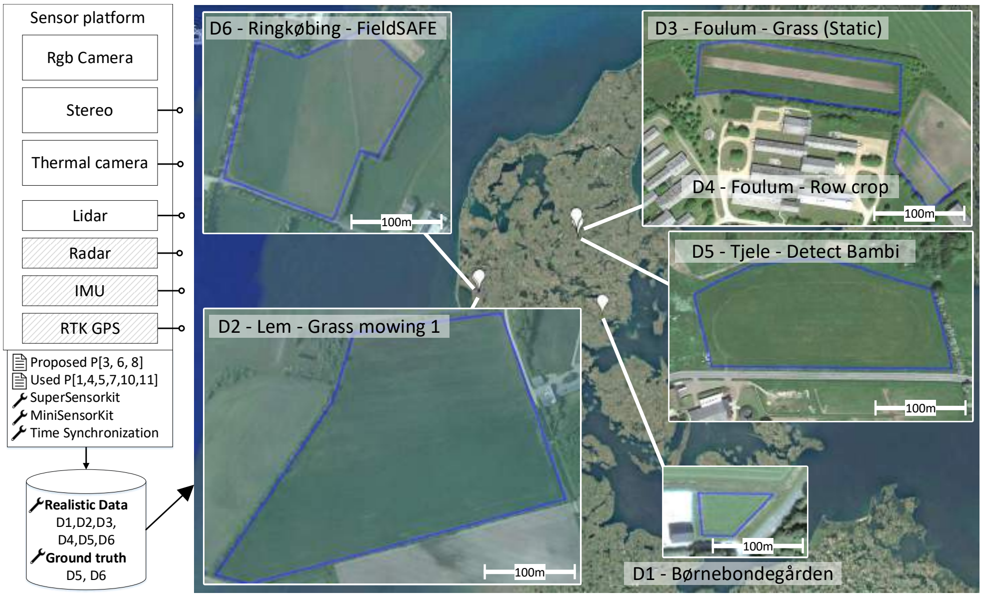


Figure 5.1: Contributions of chapter 5. Data collection through a total of six field trials (D1-D6) including ground truth for field trail D5 and D6. Location and size of each field is illustrated with field boundaries. Sensor platform and data have been presented in P[3, 6, 8] and used in P[1, 4, 5, 7, 10, 11].

Table 5.1: Overview of data including area, publications, used vehicle, used sensors, if the MiniSensorKit (MSK) is used and used obstacles

					Sensors			Obstacles		
ID	**Date: Place**	**Area (ha)**	**Publi- cations**	**Vehicle**	**Stereo Camera**	**Thermal Camera**	**MSK**	**Human**	**Doll + barrel**	**Ground truth**
D1	2014-11: Børnebondegården	0.1	P[6, 8]	ATV	S1	T1, T2		**X**		
D2	2015-06: Lem - Grass mowing	7.5	P[1, 5], P[6, 7] P[11]	Tractor + Implement	S2	T2		**X**	**X**	
D3	2015-06: Foulum - Grass (Static)	1.1	P[10]	Tractor	S2	T2		**X**	**X**	**X**
D4	2015-09: Foulum - Row crop	0.2	[5, 7]	Tractor	S2	T2			**X**	
D5	2016-06: Tjele - Detect Bambi	3.5		Tractor + Implement	S3	T3	**X**			
D6	2016-10: Ringkøbing FieldSAFE	3.1	[3, 4]	Tractor + Implement	S3	T3	**X**	**X**	**X**	**X++**

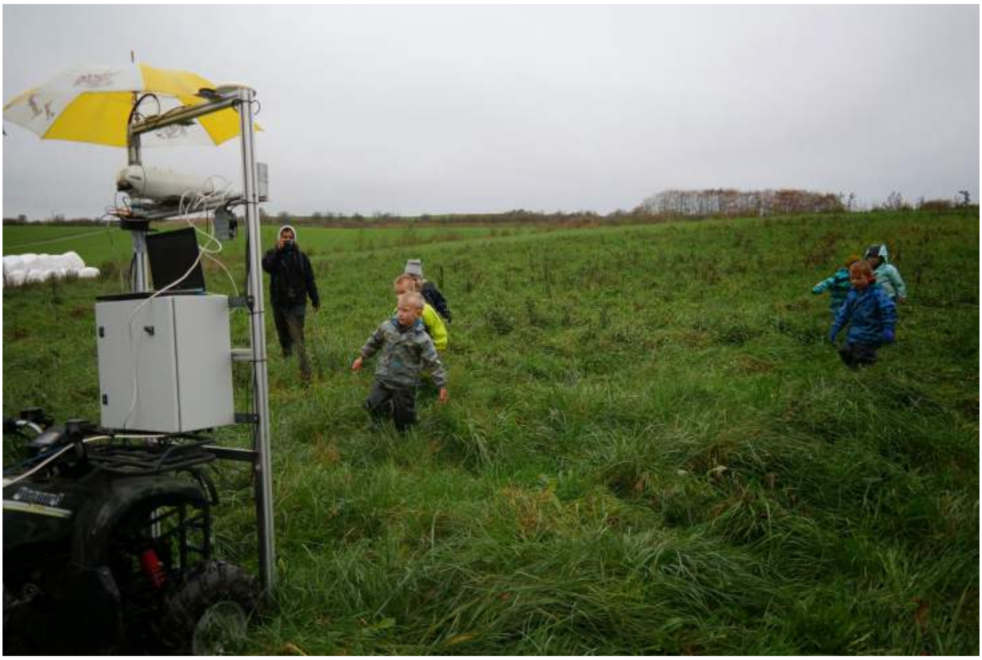


Figure 5.2: (Not so water-resistant, early edition) SuperSensorKit mounted on an ATV facing a kindergarten

A few issues made the platform inconvenient for data collection in an agricultural environment. The ATV and the SuperSensorKit was hardly able to traverse high grass, run at high speeds or suited for wet conditions, see Figure 5.2. The thermal camera (T1) also required an extra computer running on windows.

5.2.2 Field Trial 2 - Grass mowing

The aim is to capture realistic data of natural obstacles and obstacles in dangerous situations for a grass-harvesting use case. Data is used in five publications P[1, 5–7, 11].

The second field trial was recorded in a 7.5ha grass field near Lem in the beginning of June 2015. The SuperSensorKit was mounted to the implement of a grass harvesting tractor, which is shown in Figure 5.3. Data samples of natural elements in and around a field (shelterbelts, grass, ground, houses and wells) are captured. To also simulate hazard situations, obstacles were placed in the tractor trajectory. For each obstacle, the tractor breaks just before colliding with obstacles. Adult and child mannequins were used instead of real humans to ensure that no humans were harmed in the experiments. The ISO-barrel was also introduced to incorporate safety standards – the ISO-barrel is covered in *6.6 Using Deep Learning to Challenge Safety Standards*. To also get more authentic data, the mower was turned off for a few laps to capture “real” human samples. Obstacles used in the trial are presented in Figure 5.4. Field, tractor trajectories and obstacle positions are presented in Figure 5.5.

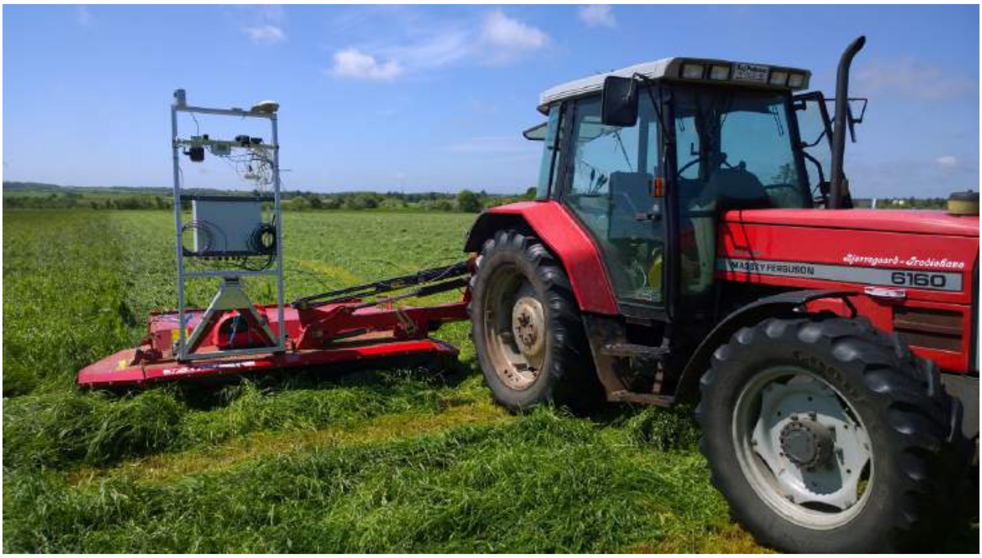


Figure 5.3: Tractor with implement and SuperSensorKit. From P[5]

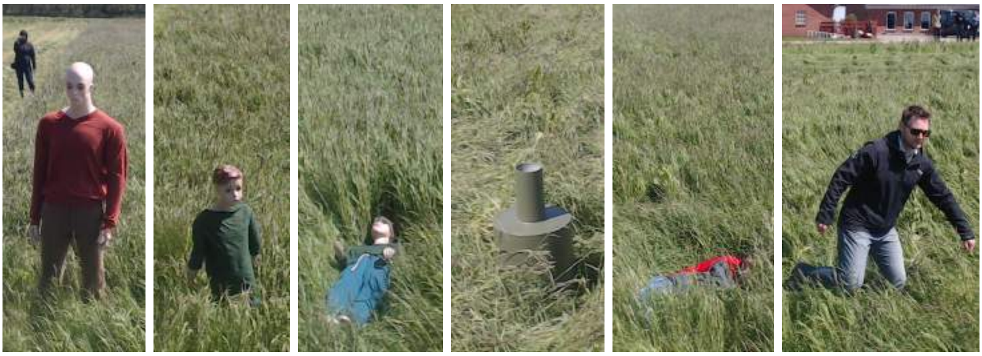


Figure 5.4: Obstacles. Mannequins, barrel and humans. From P[6]

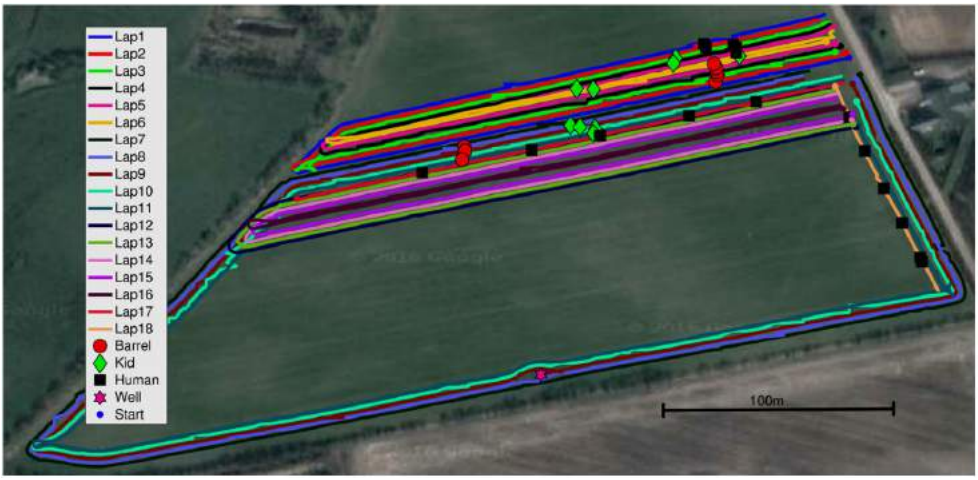


Figure 5.5: Obstacle position and lap trajectories. From P[6]

A set of improvements to the SuperSensorKit was implemented since the first field trial, D1.

- Angle-parts allow sensors to be titled in a desired pitch.
- The A-frame for easy mounting to tractor or implement.
- Rubber parts are separating A-frame and sensor rack to absorb mechanical noise from tractor.
- The Hawkvision thermal camera (T2) is placed in a 3D printed housing.
- The stereo camera has been upgraded from S1 to S2.

The second field trial, D2, provides realistic data for a grass-harvesting tractor running at real speeds. Obstacles are placed in hazard situations with various postures (lying, sitting and standing) and occluded by the high grass.

5.2.3 Field Trial 3 - Grass (Static obstacles)

The aim is to capture samples of a static environment and provide the ground truth as a static map. Data is used in P[10].

The third field trial D3 was recorded in a grass field of 1.1ha near Research Center Foulum in June 2015. The SuperSensorKit was mounted on the front-mount of a tractor traversing the field. The data contains natural elements in a field (shelterbelt, grass, ground and wells) and static obstacle (a car, barrels, and adult and kid mannequin dolls) were placed and positions were measured using RTK GPS, see Figure 5.6.

An orthophoto was generated using a Phantom 2 drone by DJI (Shenzhen, China). Ground truth was obtained by manually labelling the whole orthophoto, see Figure 5.7.

5.2.4 Field Trial 4 - Row crop (Static obstacles)

The aim is to capture samples of static obstacles in also row crops and to cover the same field under varying lighting conditions. Data is used in P[5, 7].

The fourth field trial D4 was recorded in a small row crop maize field of 0.2 ha near Research Center Foulum in September 2015. The SuperSensorKit was mounted on the front-mount of a tractor and static obstacles are placed in the field, see Figure 5.8. At roughly 9:30, 11:00 and 12:30, the tractor travels back-forth once to capture different light conditions across the same day.

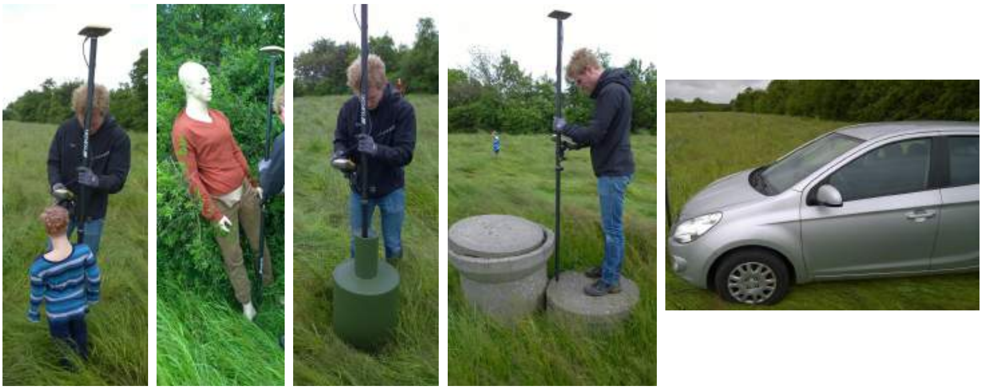


Figure 5.6: Static obstacles used.

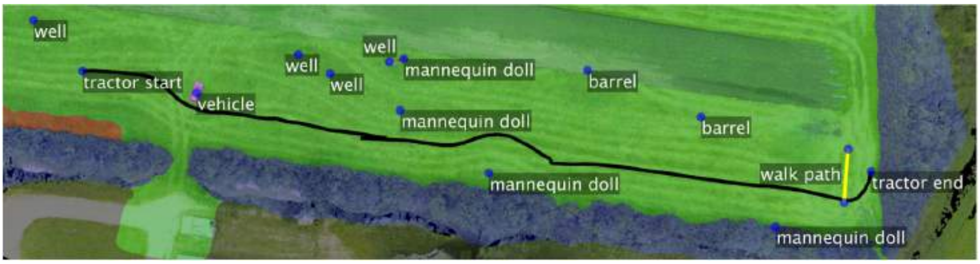


Figure 5.7: Orthophoto with static objects, tractor trajectory (black line) and human walk path (yellow line). An overlay shows the ground truth of shelterbelts (blue), ground (green) and non-traversable ground (red). From P[10]

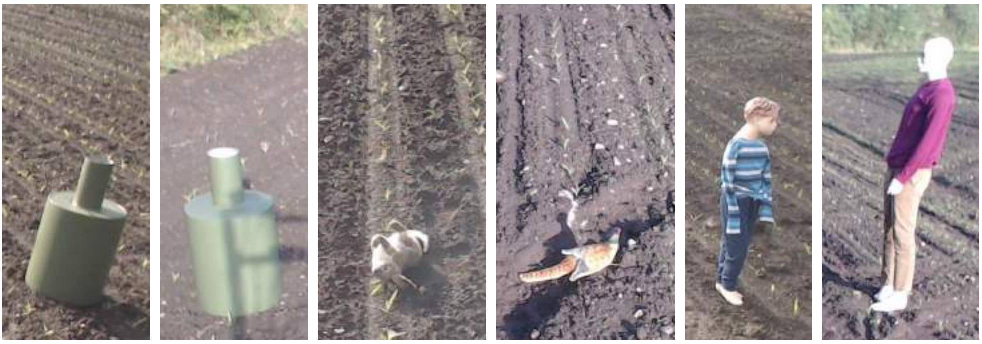


Figure 5.8: Static obstacles (barrels, teddy hare, teddy pheasant and mannequins) in row crops.

5.2.5 Field Trial 5 - Detect Bambi

The aim is to test the new MiniSensorKit mounted on the SuperSensorKit to get multi-modal samples of roe deer fawns. Data has not been used in any publications.

The fifth field trial was recorded in a grass field of 3.5ha in Tjele in June 2016. The SuperSensorKit was mounted to the implement of a grass-harvesting tractor. The field trial failed to deliver a complete and useful dataset. No roe deer fawns were detected and stereo camera, lidar, IMU and webcam failed to record at all time. However, the field trial became a valuable test for next field trial (D6 - FieldSAFE).

5.2.6 Field Trial 6 - FieldSAFE

The aim is to capture realistic data in a grass-harvesting use case and to get samples of both static and moving obstacles. A dynamic map is generated to provide ground truth of both static and dynamic obstacle. The dataset is published in P[3] and made publicly available.

Recordings
The field trial was recorded in a grass field of 3.1ha near Ringkøbing in October 2016. The SuperSensorKit and MiniSensorKit were mounted to the implement of a grass-harvesting tractor. Natural elements in and around the field are grass, ground, shelterbelts, trees, houses and roads. Static objects (barrel, mannequin kid and adult and GPS markers) were placed in the field. See Figure 5.9.

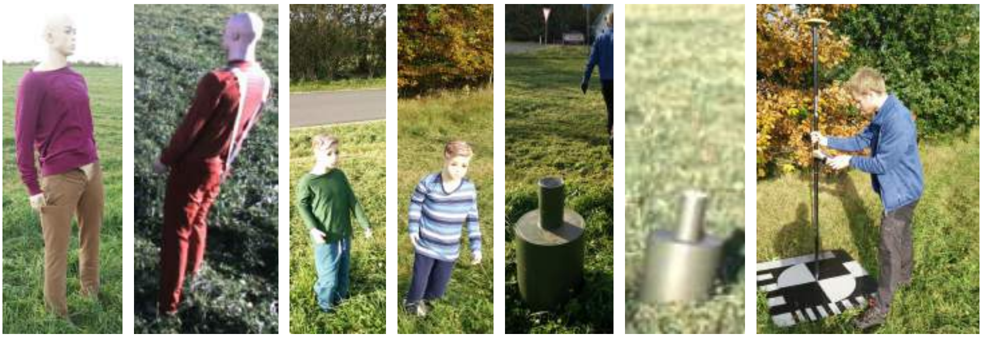


Figure 5.9: Static obstacles in the field. The position of GPS markers are measured using RTK GPS.

The data is divided in two field trials; a static and a dynamic field trial. The tractor performs harvesting for both static and dynamic recordings. In static recordings, humans are not allowed in or around the field. In dynamic recordings, the tractor and a group of up to seven people move in a pre-specified area. Figure 5.10 (left) presents tractor trajectory for static and dynamic field trials.

Ground truth - Static
The ground truth map of all static elements in the field is generated using drone recordings covering the field. An orthophoto was generated from the recording using structure-from-motion software by Pix4D (Lausanne, Switzerland). The positions of GPS markers are measured

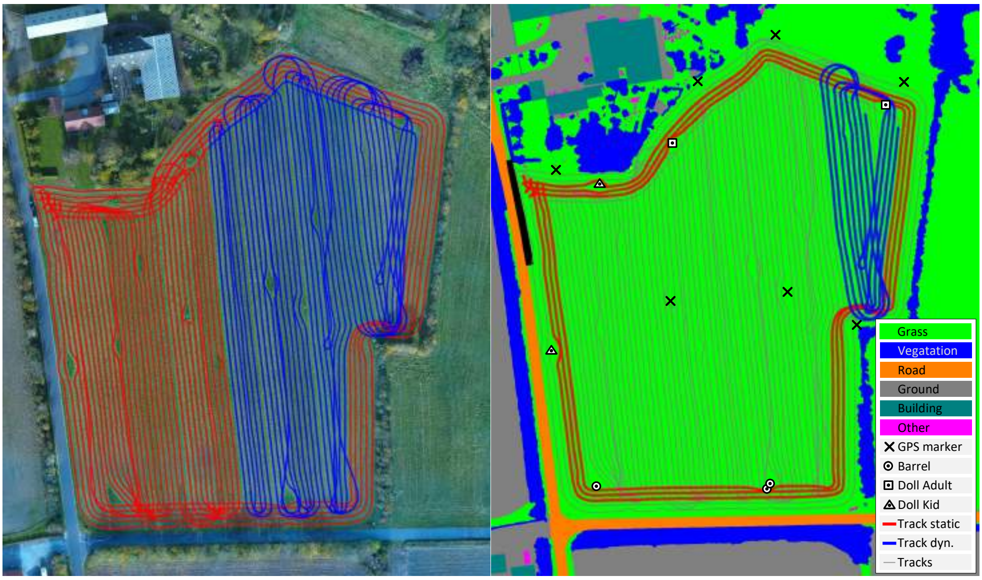


Figure 5.10: (left) Orthophoto shows static and dynamic areas of the field (right) Ground truth map showing static obstacles, tractor trajectory and field labels.

using RTK GPS to improve and align orthophoto with world coordinates. The ground truth map of static elements is obtained by manually labelling the orthophoto. Figure 5.10 b) presents ground truth of static obstacles (markers) and elements in the field (color). Not all data is used or accessible in P[3, 4]. Figure 5.10 b) shows unused data (gray), static (red) and dynamic (blue) tractor trajectories.

Ground truth - Dynamic

In the dynamic field trial used in P[3, 4], four humans (Figure 5.11) act as dynamic obstacles moving in areas close to the tractor. The challenge is obtaining the ground truth trajectory of dynamic obstacles in the orthophoto as presented in Figure 5.12.

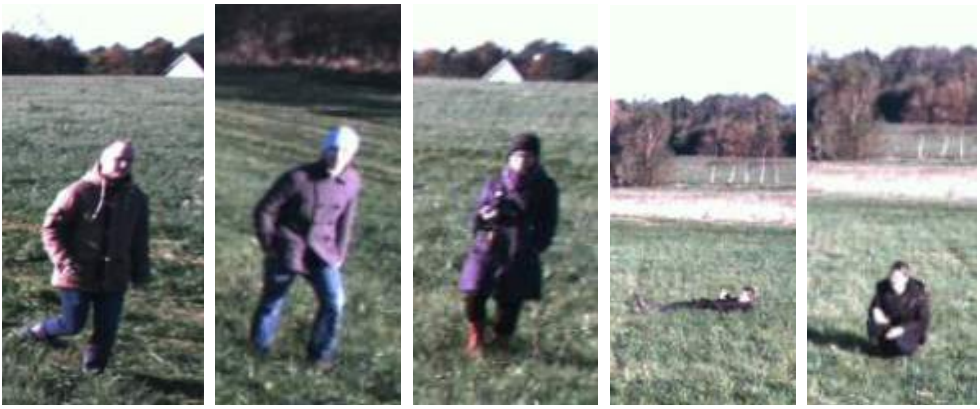


Figure 5.11: Dynamic obstacles from stereo camera. Person 1, 2, 3 and 4 (lying and sitting)

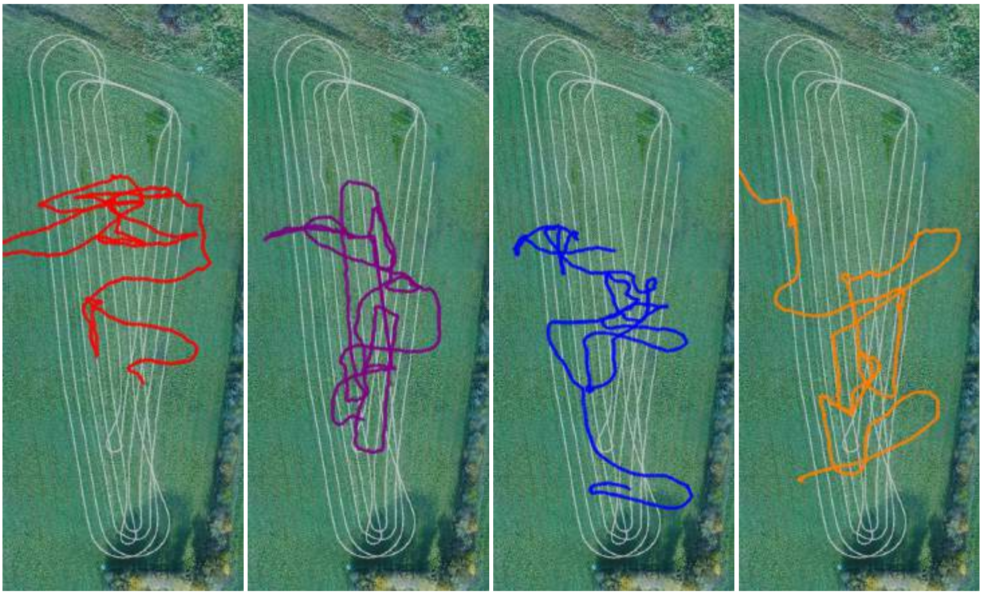


Figure 5.12: Human trajectories in individual plots

Ground truth of dynamic obstacles was estimated from gimbal stabilized footage recorded by a hovering drone. Assuming a flat surface and a pinhole camera model, there is a perspective transformation that maps drone images to the orthophoto as illustrated in Figure 5.13. First the drone camera is calibrated and images are rectified. Secondly, the perspective transformation is determined by matching static points that are recognizable in both the orthophoto and the drone image (GPS markers and mannequins). A small script was developed to automatically track the recognizable points in the drone recording using simple template matching. A user will only need to point out recognizable points in the first frame and these points are tracked throughout the whole recording.

The vatic annotation tool [122] was used for annotating dynamic obstacles in the transformed video. This enables us to draw human trajectories in the orthophoto as presented in the previous Figure 5.12. Drone and sensor platform was synchronized in time by a human clap visible to cameras on both drone and sensor kit, which allowed us to estimate the position of dynamic obstacles at a specific point in time.

Improvements of D6 FieldSAFE compared to previous field trials (D2-D4) are:

- Synchronization of thermal and stereo data
- Thermal camera provides absolute temperatures
- Stereo camera provides disparity maps online
- Better registration using thermal calibration panel – the thermal calibration panel is described in *7.1.1 Thermal-visual Calibration Panel*
- Localization and heading have improved using a differential GPS.
- Static and dynamic ground truth

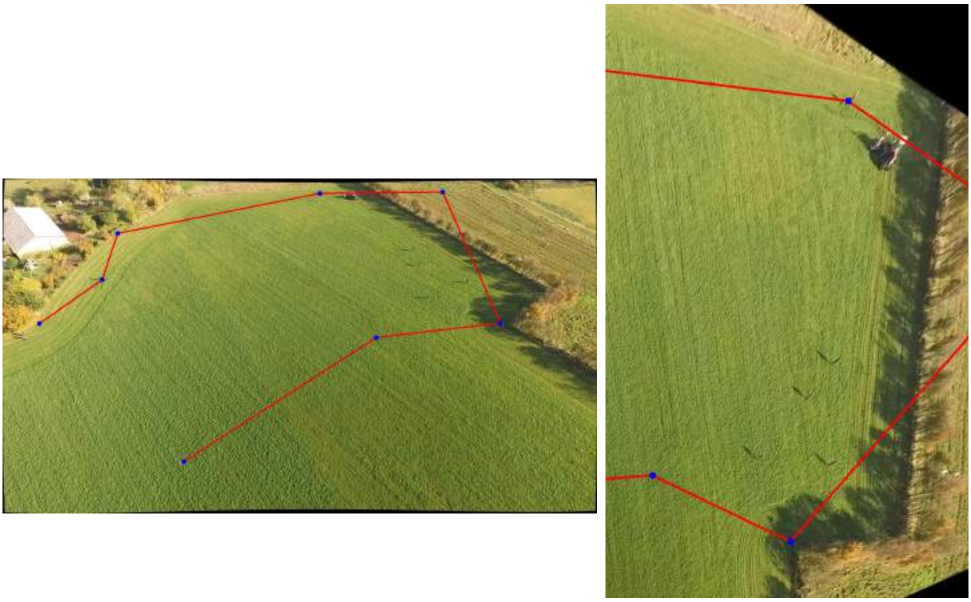


Figure 5.13: (left) Drone images (right) Drone image after transformation

5.3 Concluding Remarks

A total of six large scale field trials have been conducted with a multi-sensor system. The dataset includes many obstacles that naturally resides in or around an agricultural field such as shelterbelt/trees, building, road, grass and the field itself. Furthermore, hazard situations are simulated by placing static obstacle and instructing humans to act moving obstacles in and around the trajectory of the tractor. To comply with recent safety standards within agriculture, an olive-green barrel have been used and created according to the ISO/FDIS 18497 standard *"Agricultural machinery and tractors — Safety of Highly Automated Agricultural Machines"*. Drones are used before, under and after the final field trial *FieldSAFE* to capture ground truth data of static and moving obstacles in the field. Datasets such as FieldSAFE are an important step to improve and evaluate detection algorithms and to ultimately realize autonomous farming vehicles.

5.3.1 Challenges and Future Work

Providing data of sufficient quality in the agricultural domain is costly and time-consuming. One thing is the development of a robust sensor platform that is suited for agricultural vehicles. The other is planning realistic field trials and to acquire agricultural vehicles, grass mowing implements, people to operate them, volunteers to act moving obstacles and patient farmers to make their field available. Furthermore, static obstacles must be acquired, transported and placed and move in the field. Finally data collection must match the routines of the farmer and can only be conducted on harvesting days. To improve errors or bad recordings you will either need to find a new field or wait another season.

Originally, data was recorded for training multi-modal detection algorithms in the agricultural domain. However, especially for camera-based detection algorithms, this requires training data of a broad set of scenarios with multiple objects in various postures and weather conditions to avoid overfitting. Compared to e.g. the Kitti benchmark in a suburban environment, the incidence of obstacles are low in an agricultural field, making the process comprehensive in terms of data and the subsequent annotation of data. The large and high-quality data available for image recognitions in other domains have in this thesis been used to demonstrate that such data is – to some extend – able to generalize to agriculture. The purpose of field trials have, therefore, become more a dataset for testing detection algorithms, and the training of algorithms is done using datasets from other domains.

A benchmark for multi-modal object recognition with labels in sensor frame would eventually be valuable for autonomous vehicles in agriculture and the MiniSensorKit was originally developed to enable a single person to capture multi-modal data in environments and of objects related to agriculture. This task was eventually dropped because of the limited time frame of this thesis. Future work is the generation of sensor frame annotation of FieldSAFE and new data to provide training data in agricultural context.

6 Obstacle Detection

To get fully autonomous vehicles certified for farming, computer vision algorithms and sensor technologies must detect obstacles with equivalent or better than human-level performance. Furthermore, detections must run in real-time to enable vehicles to actuate and avoid collision. This section describes detection algorithms for the thermal and rgb camera. Figure 6.1 illustrates contributions and information flow - input image and output detections.

The five detection modules used in TractorEYE are LDCF [17], YOLOv2 [99], FCN [19], DeepAnomaly P[1] and DynamicHeat P[4]. All five algorithms have been implemented in this thesis as ROS package GH1, GH2, GH3, GH4 and GH5 to be used in TractorEYE. DeepAnomaly P[1], DynamicHeat P[4] are proposed in this thesis. LDFC, YOLOv2 and FCN have been investigated and evaluated in an agricultural context P[1, 4, 7, 10]. Finally, the algorithms in P[2, 5] have been proposed but not used in TractorEYE.

6.1 Pedestrian Detector: LDCF

The Local Decorrelated Channel Features (LDCF) [17] by Piotr Dollar as the main contributor have been evaluated for agriculture in the following publications P[1, 4, 10]. MATLAB code was converted to C++ and a ROS package, GH1, to comply with SAFE deliverables. The used model is trained on the INRIA Person Dataset [123]. Initial publications of LCDF were proposed in 2009 [83] and 2010 [124] and were – prior to deep learning methods – state-of-the-art for fast pedestrian detector. LDCF is public available in a detector framework [125] by Piotr Dollar. The detector is first trained on pedestrians (positives) and non-pedestrians (negatives) using AdaBoost as classifier and aggregated decorrelated channel features.

In the detection phase, channel features are calculated on the whole image and pedestrians are detected at multiple scales and positions in the image with a sliding window approach. The

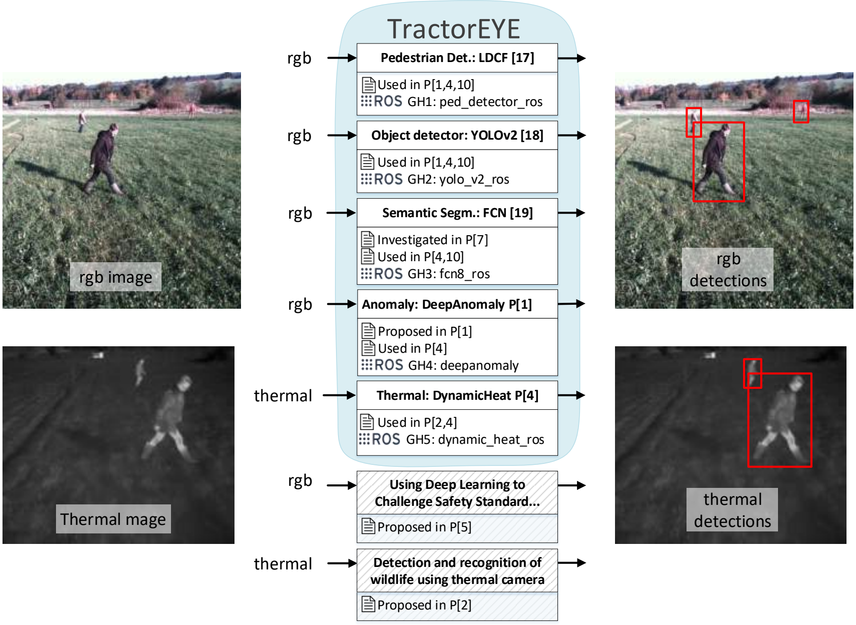


Figure 6.1: Contributions of chapter 6. Five detection algoritms are used in TractorEYE (LDCF [17], YOLOv2 [99], FCN [19], DeepAnomaly P[1], DynamicHeat P[4]) and implemented as ROS package GH1, GH2, GH3, GH4 and GH5. LDFC, YOLOv2 and FCN have been evaluated in an agricultural context P[1, 4, 7, 10]. DeepAnomaly P[1], DynamicHeat and P[2, 5] are proposed in this thesis.

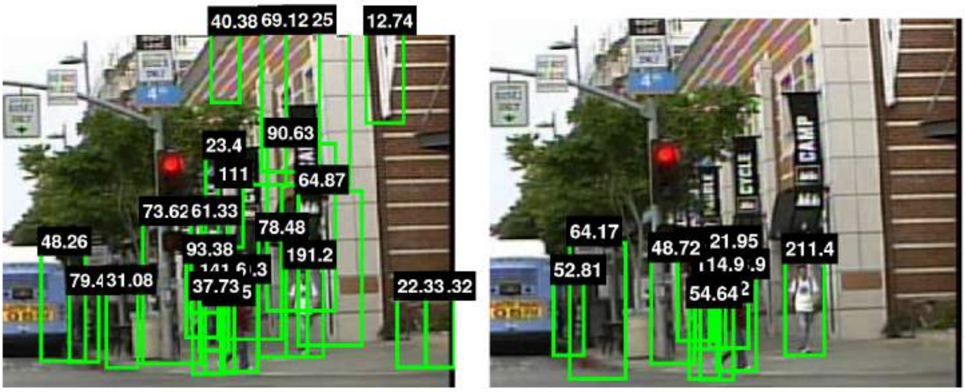


Figure 6.2: (Left) ACF without ground plane assumption (Right) ACF with ground plane assumption.

algorithm is optimized for speed by estimating features at multiple scales instead of calculating them explicitly. The detector has improved through multiple publications [17, 83, 84, 126, 127].

I have an unpublished paper using Aggregated Channel Feature [126] (ACF) combined with ground plane assumption. Figure 6.2 presents the algorithm with and without a ground plane assumption.

LDCF and Viola Jones based detectors have for a decade [128] been developed and used for assisted driver/safety systems in the automotive industry. Mobileye was founded for solving this issue (hardware and software) and is now a $15.3 billion company, though their systems have become far more intelligent. Viola Jones based algorithms are fast and applicable for up-right pedestrians and faces, where a bounding box of fixed aspect ratio is able to capture the object of interest. However, the limited capacity of Viola Jones based detectors and the fixed aspect ratio is conceptually not suited for detecting multiple object types or a single object class with variable postures. In agricultural and urban environments, people will often be in an upright posture as a pedestrian. However, detection of other human postures are crucial to autonomous vehicles as especially a lying, sitting or unconscious person have reduced mobility to actively avoid a dangerous situation. This was also experienced in the D1 data using the LDCF detector, see Figure 6.3. Furthermore, a single detector should preferably detect multiple object classes such as e.g. humans, animals and agricultural vehicles.

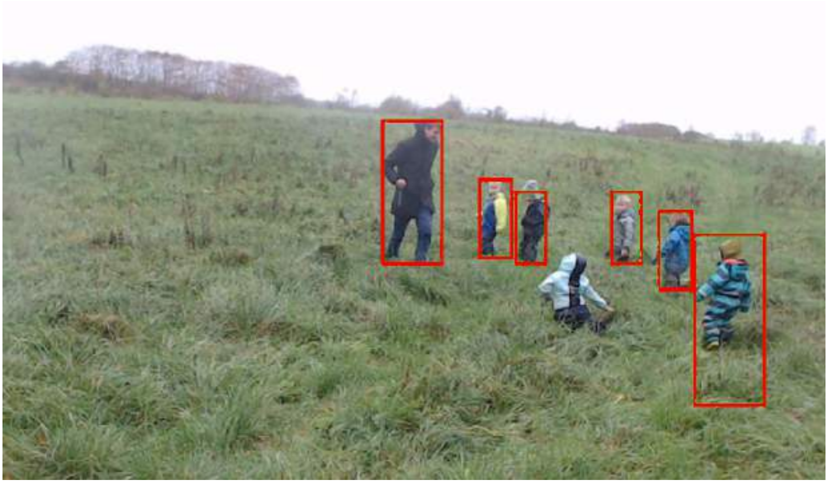


Figure 6.3: Hazard situation, showing that LDCF detects all humans but the failing kid

Object detection is broader defined and is conceptually able to detect multiple object classes and variable postures. Deformable parts model (DPM) [85, 86] have been popular in the late 2000s for object classification tasks such as in the PASCAL VOC object detection competition with 20 object classes. The performance is however not near the performance of recent CNN-based models.

6.2 Object detection: YOLOv2

The object detection module uses a CNN-based single forward pass object detector called YOLO [18, 99] by Joseph Redmon as the main contributor. YOLO is implemented in Darknet [129] – a deep learning framework written in C – also maintained by Joseph Redmon. Thesis contributions are two ROS packages GH2 to enable YOLO [18] and YOLOv2 [99] to be used in P[1, 4, 10]. Compared to LDCF, YOLOv2 is able to detect multiple object classes with variable postures of much higher accuracy and is able to run real-time using a middle range GPU.

The same hazard situation from Figure 6.3 using the LDCF detector is presented again in Figure 6.4 using YOLOv2 as detector. YOLOv2 is able to detect all persons including the falling kid.

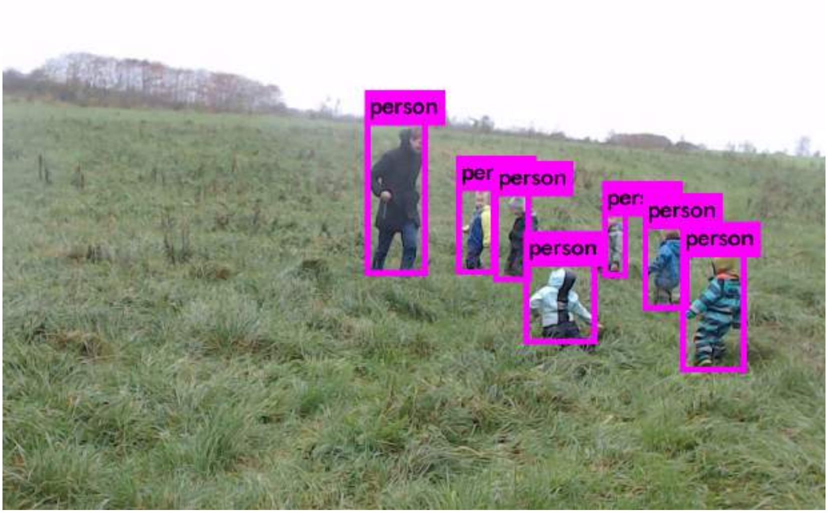


Figure 6.4: Hazard situation with YOLOv2. YOLOv2 is able to capable of detecting all persons including a falling kid.

For autonomous vehicles in the agriculture, many obstacles or elements are imprecisely localized with a bounding box such as roads, building, shelterbelts and fencing. This have been demonstrated in Figure 6.5 with simulated annotations. Detections of such elements are critical in agriculture and to operate the vehicle safely.

Traditionally, multiple detection algorithms or sensor modalities have been used for detecting such semantics. A rapidly growing image recognition task defined as semantic segmentation enables the detection and precise delimitation of both obstacles and elements in the image by classifying each pixel.

6.3 Semantic Segmentation: FCN

The semantic segmentation module uses a Fully Convolutional Network (FCN) for semantic segmentation [19] developed by Jonathan Long using the Caffe framework [130]. The contribution of this thesis is a preliminary study of FCN for autonomous tractors in a standalone paper P[7]. Additionally FCN is evaluated, combined and compared to other CNN-based rgb

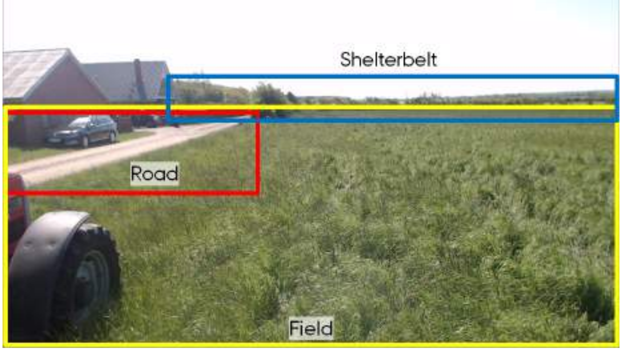


Figure 6.5: Demonstrations of how annotations for field, road and shelterbelt are imprecisely delimited with a bounding box

algorithms (YOLO [18], LDCF [17]) and sensor modalities for mapping static obstacles in P[10] and in P[4] for mapping both static and dynamic obstacles. A ROS package (GH3) is made for executing a pre-trained model.

The contribution of P[7] is a preliminary study of FCN for autonomous farming vehicles in two use cases; grass mowing and row crop operation. The study uses predictions from a model [19] trained for whole scene per-pixel classification on the 59 most frequent classes of Pascal Context [103]. Pascal Context provides per-pixel annotations of whole scenes in 10,103 images of 407 object classes. Not to be confused with Pascal VOC, where only 20 object classes have been annotated. Predictions for the 59 most frequent classes are remapped to 11 agricultural super-categories; animal, building, field, ground, obstacle, person, shelterbelt, sky, vehicle, water, and unknown. An image example for grass and row crops is presented in Figure 6.6 and Figure 6.7, respectively.

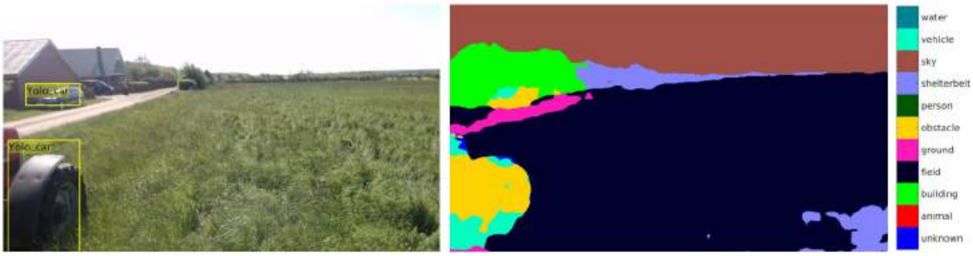


Figure 6.6: Semantic segmentation of remapped classes in grass. From P[7]

Ground truth is annotated for five grass and five row crop images and evaluated to a classification accuracy of 95.25% and 70.54%, respectively. The simple remapping of a pre-trained model demonstrates the potential of semantic segmentation in an agricultural use case. Still, better data is required to better classify a class like row crops, as it is not uniquely defined in the Pascal Context data set.

Semantic segmentation provides powerful detection capabilities to a regular rgb camera, allowing it to detect objects and elements that are not precisely delimited by a bounding box.

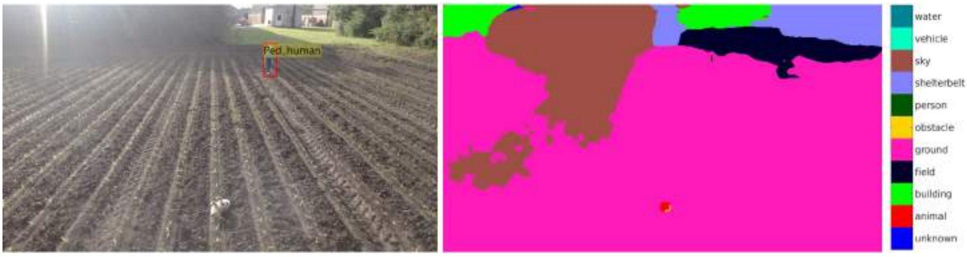


Figure 6.7: Semantic segmentation of remapped classes in row crop. From P[7]

Conceptually, supervised algorithms such as semantic segmentation and object detection are only able to detect a set of predefined classes and require images samples for each object class. Secondly, agricultural productions fields are covered by single crop species making a field visually homogeneous in texture and color. Elements that do not conform to the visual appearance of the field are potential obstacles. Figure 6.8 presents a set of obstacles that are difficult for a supervised algorithm. Because objects are heavy occluded, unexpected or rare in an agricultural field.

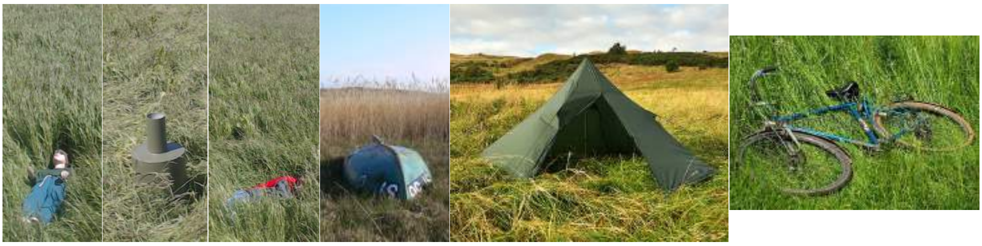


Figure 6.8: Image examples of heavy occluded or rare obstacles in an agricultural context.

6.4 Anomaly detection: DeepAnomaly

The DeepAnomaly module in TractorEye uses the anomaly detector proposed in P[1], entitled "DeepAnomaly: Combining Background Subtraction and Deep Learning for Detecting Obstacles and Anomalies in an Agricultural Field". Additionally, DeepAnomaly is evaluated, combined and compared to other CNN-based rgb algorithms and sensor modalities in P[4] for mapping both static and dynamic obstacles. A ROS package has been developed but is currently not publically available (GH4). The algorithm is demonstrated in Figure 6.9 and Figure 6.10.

DeepAnomaly uses high level features from a pre-trained CNN-model to detect anomalies that do not conform to an agricultural context, see Figure 6.11.

A broad set of configurations (460 settings) have been evaluated for detection accuracy and compute time. A single setting with a high detection accuracy and low compute time is selected and denoted DeepAnomaly. DeepAnomaly uses the output feature map generated by the last convolutional layer of a modified AlexNet-model trained for image classification on ImageNet. A set of images of an agricultural field with no obstacles is propagated through the network.

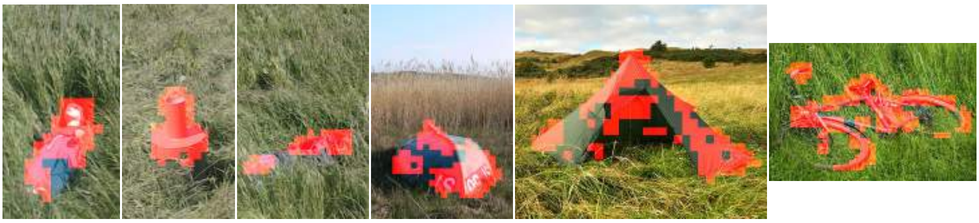


Figure 6.9: DeepAnomaly on anomaly image examples (from Figure 6.8)

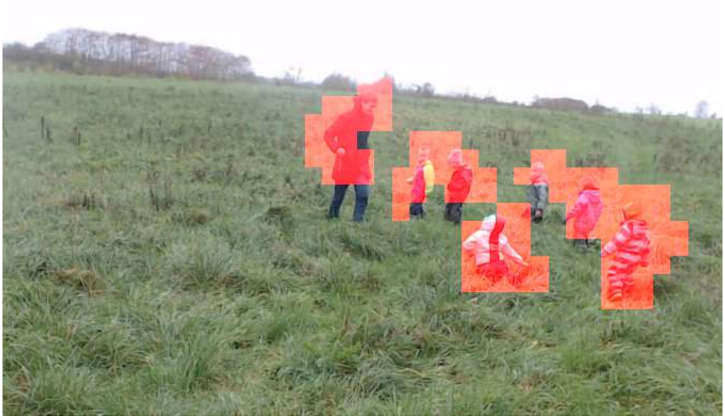


Figure 6.10: DeepAnomaly on kindergarden image. Same image have also been tested with LDCF in Figure 6.3 and YOLOv2 in Figure 6.4

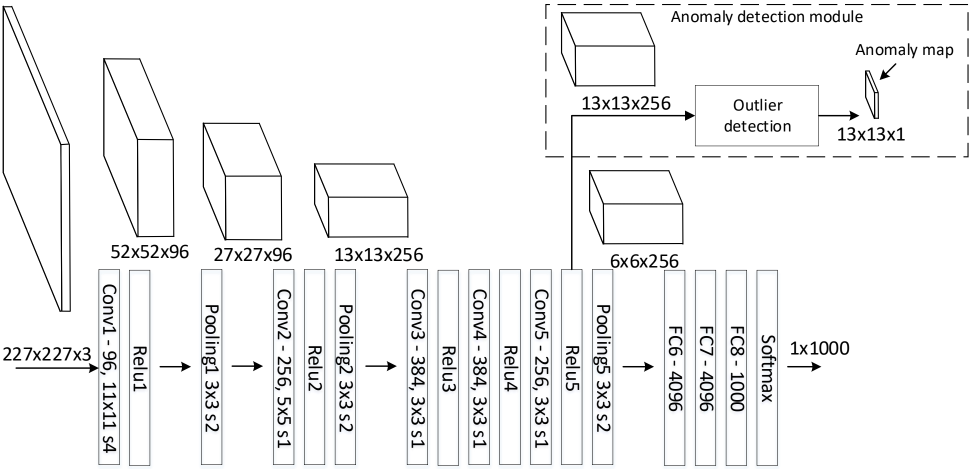


Figure 6.11: Feature maps of intermediate layers of AlexNet for image classification of 1000 classes. DeepAnomaly uses high abstraction features from AlexNet to create a low resolution anomaly map. Figure taken from P[1]

An outlier detector is generated by modelling the normal statistics of the output feature maps (Figure 6.11) using a single variate Gaussian distribution model. The outlier detector, measures the Mahalanobis distance between the normal statistics and a new feature map entry. A feature map entry is defined as an anomaly if the Mahalanobis distance excessed a pre-defined threshold. Anomaly detections are presented in Figure 6.13.

A human detection accuracy metric is used to compared Deep anomaly to four other detection algorithms YOLO (version 1) [18], Faster R-CNN [94], FCN [19] and LDCF [17]. A human detection use case is defined to quantitatively demonstrate that DeepAnomaly is a state-of-the-art real-time detector in an agricultural context for detecting distant, heavily occluded and unknown obstacles (anomalies). Figure 6.12 (a) demonstrates that DeepAnomaly provides better detection accuracy (F1-score) and Figure 6.12 (b) shows that DeepAnomaly is generally better – especially for longer distance intervals.

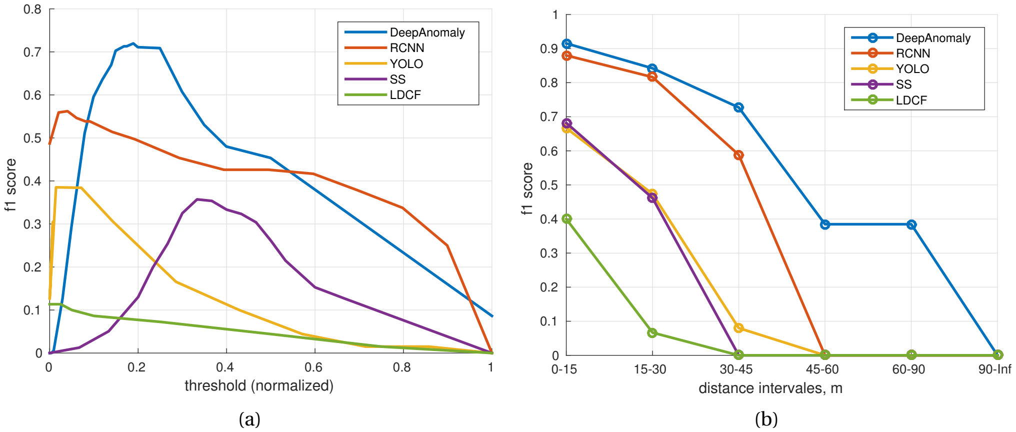


Figure 6.12: (a) Detection accuracy (F1-score) for variable thresholds. (b) Detection accuracy (F1-score) for a variable distance intervals P[1].

Image examples present detections for all algorithms in Figure 6.14.

A large and slightly edited section is taken from P[1] "*Deep learning-based object detection and semantic segmentation have recently showed state-of-the-art results in detecting specific objects. However, in an agricultural context, they have difficulty in detecting heavily occluded and distant objects, and methods are, by definition, trained to recognize a predefined set of object types. DeepAnomaly can exploit the very homogeneous characteristics of an agricultural field to detect distant, heavy occluded and unknown objects. Qualitatively, this is illustrated in Figure 6.13, where DeepAnomaly detects a distant and occluded mannequin kid, a human showing only his arm, a heavy occluded olive-green barrel (with similar color as the field), a well cover and detections of obstacles with a size of less than 16 × 16 pixels. By using DeepAnomaly in junction with other deep learning algorithms, it can save computations by using convolutional features from other networks. DeepAnomaly also spares the time-consuming task of providing domain- or algorithm-specific annotated data. An evaluation metric for detecting humans is defined to compare DeepAnomaly with four state-of-the-art algorithms. The comparison shows that DeepAnomaly is better at detecting humans at longer ranges (45–90 m). R-CNN has similar performance at short range (0–30 m). However, with much fewer model parameters and*

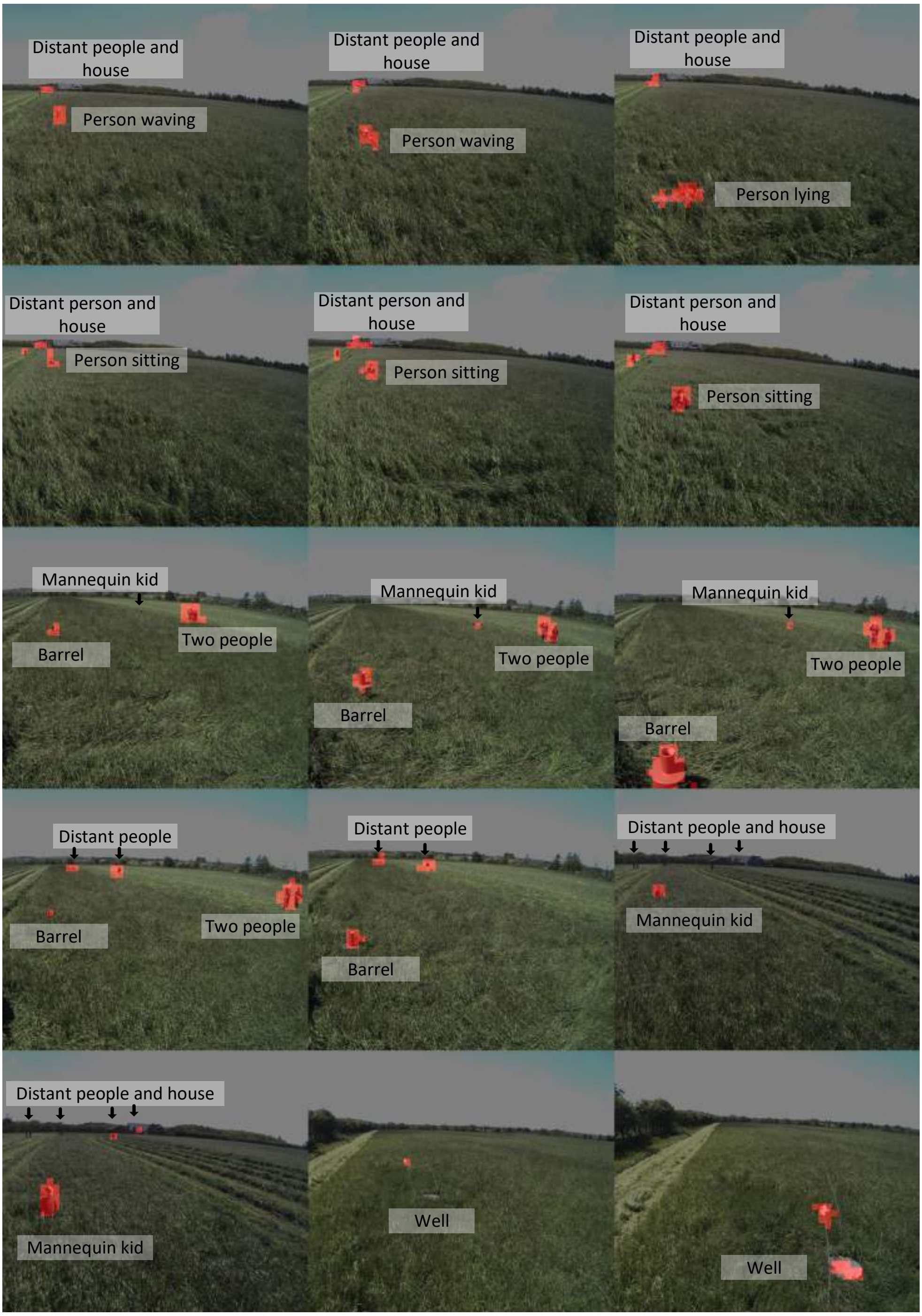


Figure 6.13: Anomaly detection by DeepAnomaly. From P[1]

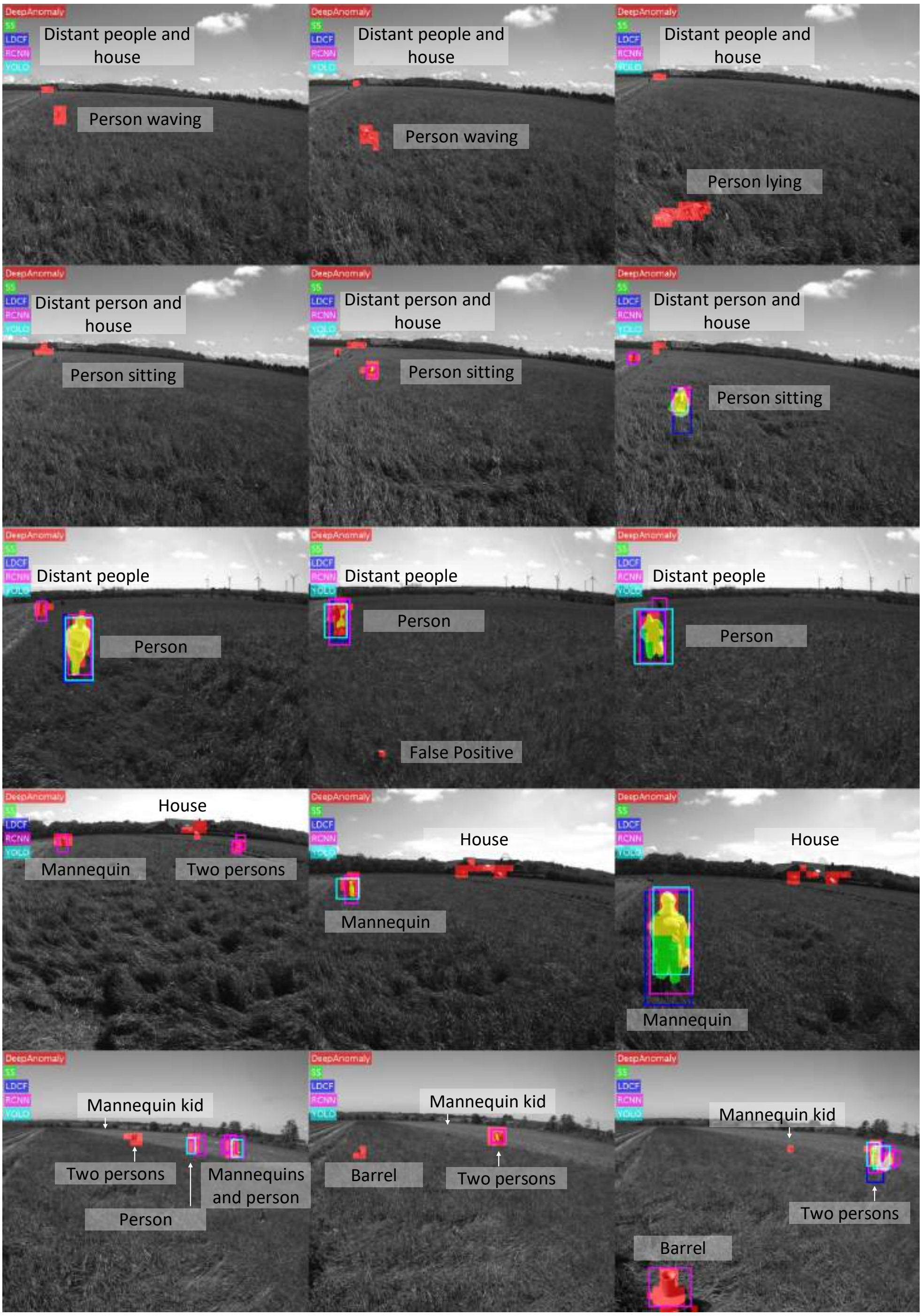
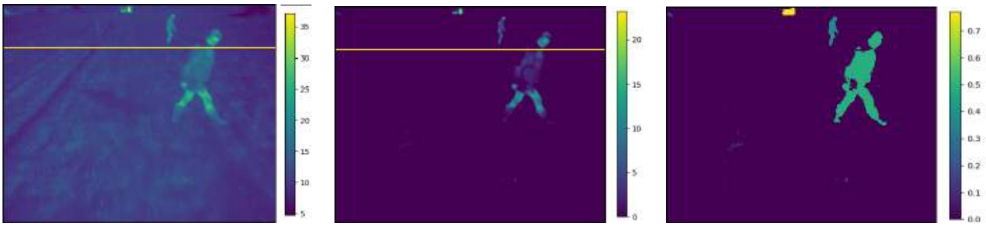


Figure 6.14: Detection of DeepAnomaly, SS (FCN), Faster R-CNN and LDCF. Figure taken from appendix in from P[1]

a $(182ms/25ms =)$ *7.28-times faster processing time per image, DeepAnomaly is more suitable for real-time applications running on an embedded GPU. The used detection metric copes with dissimilar outputs of the evaluated algorithm and will not favor a precise localization/position of a detection. However, in the context of autonomous vehicles in agriculture, the exact bounding box position or semantic segmentation at pixel-level precision is not of critical importance. Rough localization markings (±12 pixel) are sufficient and more important are the detector's ability to, in real time, detect obstacles even when they are heavily occluded, distant and potentially unknown. However, it is important to state that the object type is unknown to DeepAnomaly, and it requires specific conditions in terms of visually-homogenous surrounding and a low incidence of anomalies. YOLO, R-CNN, FCN and LDCF provide also object type and are more generalizable."*

6.5 Thermal heat detection: DynamicHeat

A simple heat detection module is developed in P[2] and modified and tested in P[4]. The implementation is available on GitHub (GH5). Hot elements are detected using a slightly modified dynamic threshold P[2]. The median temperature is determined for all image pixels in a bottom region of the image. The bottom region is visualized by a yellow line crossing the image in Figure 6.15 (a). The median temperature and a constant value is subtracted from the image and all negative values are set to zero as in Figure 6.15 (b). A connected components-algorithm is used for merging elements in the image and assigning them to the maximum value of each component as presented in Figure 6.15 (c). Values are normalized according to a maximum value to represent hot elements with some pseudo probability measure.



(a) (b) (c)

Figure 6.15: (a) Thermal image. (b) Thermal image subtracted by the median temperature. (c) Connected components are set to the maximum value of each component and normalized according to a specified value.

6.6 Using Deep Learning to Challenge Safety Standards

The publication "Using Deep Learning to Challenge Safety Standard for Highly Autonomous Machines in Agriculture" P[5] is a response to the emerging Safety Standard ISO/DIS 18497 called "*Agricultural machinery and tractors – Safety of highly automated agricultural machines*". A section in the ISO describes how to meet requirements for obstacle detection. A standardized object is defined to mimic a human seated with a visible torso and head, see Figure 6.16.

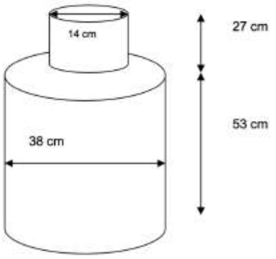


Figure 6.16: Standardized obstacle

For distance and depth sensors (ultrasonic, lidar and Time-of-Flight) such an obstacle or barrel would resemble a kid or a sitting human. However, for an imaging rgb camera, a barrel will obviously not resembles a human and an rgb-based object detector trained on barrel images would not generalize to humans.

A simple use case and detector was designed to demonstrate that a CNN-based detector would with high detection accuracy be able to detect barrels and no mannequins, humans or (stuffed) animals.

Firstly, 437 barrel images were extracted from five small video sequences, see Figure 6.17.

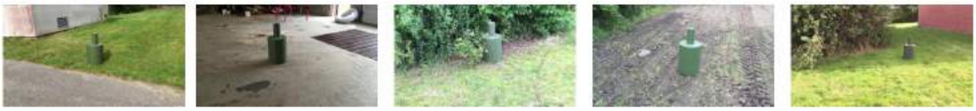


Figure 6.17: Barrel images are extracted from 5 video sequences. From P[5]

The evaluation is performed in a grass field (D2) and row crops (D4) as presented in Figure 6.18.

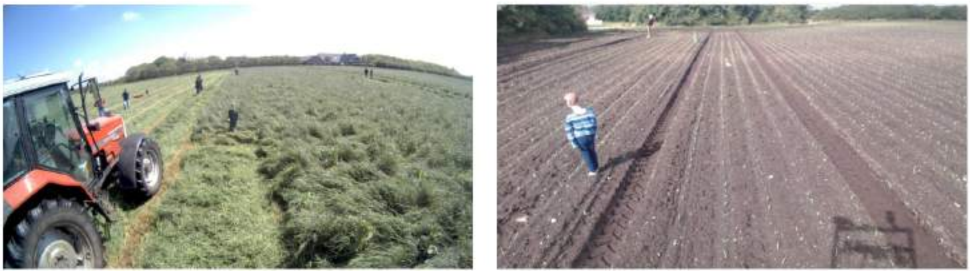


Figure 6.18: Test data is from grass and row crops. From P[5]

Following the steps in [38], a pre-trained AlexNet model was fine-tuned for image classification of barrel images. The fully connected layers are reshaped to convolutional layers allowing the model to be process larger images at multiple scales to create a heat map for each scale, see Figure 6.19.

Heat maps are then converted to bounding boxes and non-maximum suppression removes bounding box duplicates. The tractor drives by the barrel 14 times and is able to detect it for each run. More on procedure and results can be found in the paper.

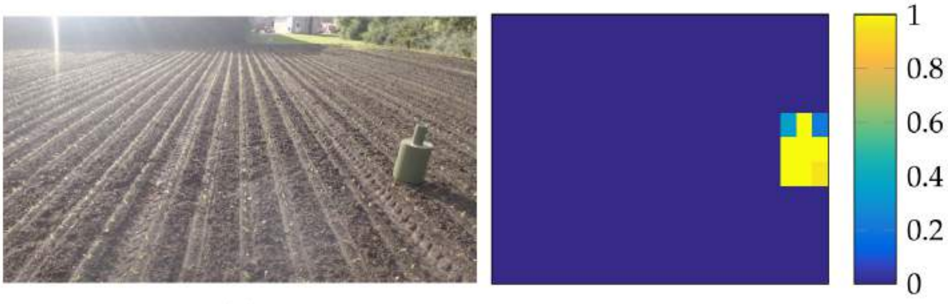


Figure 6.19: Presents heat map of the barrel class for a particular scaled input image. From P[5]

6.7 Wildlife Detection using Thermal Camera

The publication "*Automated Detection and Recognition of Wildlife Using Thermal Cameras*" P[2] is an early standalone publication to be used by an Unmanned Aerial Vehicle (UAV). The data is recorded from a telescopic boom and includes a molehill, a rabbit and a chicken to simulate data from an UAV. Elements are detected by thresholding objects warmer than a dynamic temperature, which is defined as a small constant (2 degrees) above the median temperature of all pixels in the current image. The contribution of this paper is a translational, rotational and partly scale invariant feature descriptor - defined as the thermal signature. The thermal signature measures the temperature from boundary to object center by iteratively shrinking the object by its own boundary, see Figure 6.20. For each iteration the mean contour temperature is measured and stored in a vector as presented in Figure 6.21.

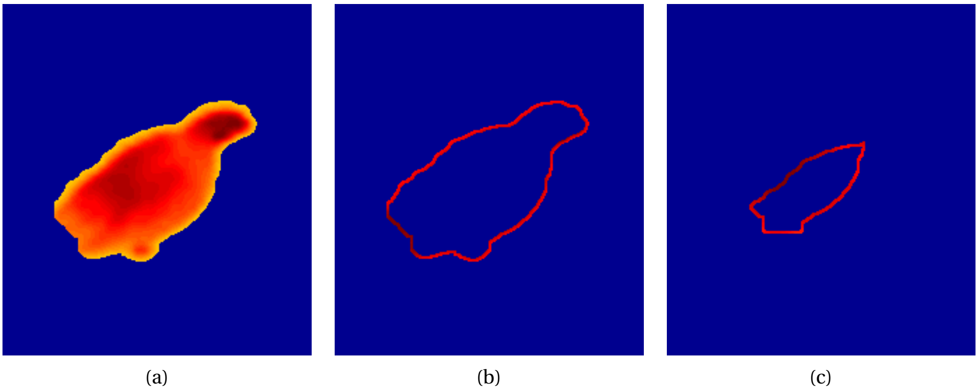


Figure 6.20: Detected object is iterative shrunk by the contour. From P[2]

To normalize and to use a fixed sized vector, vectors are subtracted by the first contour value and approximated by a Discrete Cosine Transform with a fixed number of coefficients. Seven coefficients are used as these are able to describe 95% of the signature information for 95% of the provided data. Temporal information was incorporated using a simple tracker and the Bayes rule to obtain a balanced classification accuracy of 93.5% in the altitude range 3-10m and 77.7% in the altitude range of 10-20m. More on procedure and results can be found in the paper P[2].

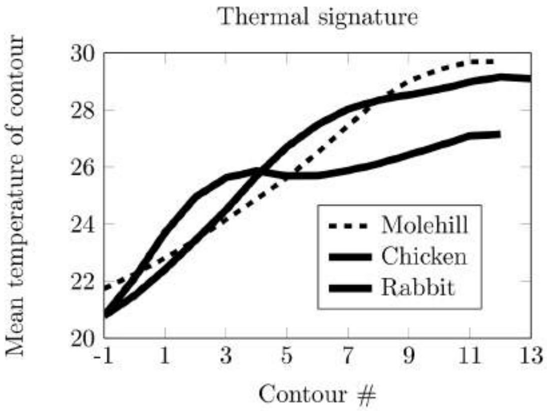


Figure 6.21: Thermal signature for molehill, chicken and rabbit. From P[2]

6.8 Concluding Remarks

A broad set of state-of-the-art real-time detection algorithms have been investigated such as LDCF, Faster R-CNN, YOLO, YOLOv2, FCN and DeepAnomaly to detect obstacles in agriculture. A total of five algorithms (LDCF, YOLO/YOLOv2, FCN, DeepAnomaly and HeatDetection) have been implemented as ROS-packages to enable autonomous farming machines and TractorEYE to execute algorithms in an actual application. DeepAnomaly, HeatDetection and P[2, 4] have been proposed in this thesis. DeepAnomaly is especially a valuable contribution to autonomous vehicle in agriculture to detect distant, heavy occluded and unknown obstacles. For an agricultural use-case it is – compared to a state-of-the-art object detector *Faster R-CNN* – able to detect humans better and at longer ranges (45-90m) using a smaller memory footprint and 7.3-times faster processing. Low memory footprint and real-time processing makes DeepAnomaly suitable for an embedded GPU and for autonomous farming vehicle. Another advantage of DeepAnomaly is its application for potentially other domains and how easily it can be adapted for a new use case. Deep Learning algorithms are often dependent on a large amount of annotated data. Using a pre-trained network, DeepAnomaly have in P[1] achieved state-of-the-art performs using only 56 images. The selected images are only weakly supervised in the sense that selected images should simply not contain obstacles or anomalies.

6.8.1 Challenges and Future Work

State-of-the-art image recognition have dramatically improved through the course of this thesis. The development of hardware, libraries, frameworks and algorithms for deep learning have been a great advantage, but also required my work to rapidly adapt according to new methods and research. The rapid development have also introduced many new concepts to run algorithms more efficiently, obtain better accuracy and turn predictions into probabilities.

Bayesian Deep Learning A disadvantage of deep learning model predictions is that values do not represent actual probabilities. Bayesian deep learning seeks to quantify and predict uncertainties to basically understand what a model does not know [131–133]. This is especially important for detection algorithms in robotics and autonomous systems where the environment and measurements are modeled using probabilistic robotics [134]. Bayesian beep learning is applied to represent two types of uncertainties for both regression and classification networks in [132]; epistemic for describing model uncertainties and aleatoric describing uncertainty of predictions because such information is simply not represented in the data e.g. areas that are overexposed in an image.

Multi-task Learning Recently, many deep learning networks are trained for multiple tasks (multi-task learning) [135] to predict e.g. semantic segmentation, instance segmentation and depth/disparity image [133] using a single backbone network. The advantage is shared computations (one model instead of three) and that the accuracy of each tasks is potentially improved compared to a single task learning setup [133, 135]. Distance to objects is for TractorEYE estimated using a stereo camera/stereo matching and detection algorithms are using individual models. Using multi-task learning to execute obstacle detection, semantic segmentation and estimate the distance to objects would reduce the computational requirements of TractorEYE and avoid the cost of a stereo camera.

Improvements in Accuracy and Compute Compute efficiency is expected to improve using the concepts of MobileNet [112] for both object detection, semantic segmentation and anomaly detection. Improvements in accuracy are also expected by using concepts from other network architectures such as Wide ResNet, ResNeXt and DenseNet. Object detection are expected to improved using Feature Pyramid Networks. Recently, datasets for whole scene semantic segmentation have been introduced such as MIT Scene Parsing Benchmark with 20,000 images and a new *stuff segmentation* challenge in MS COCO with 40,000 images. Such data is expected to improve on the less common challenge of whole scene annotations. Whole scene semantic segmentation is especially important in agriculture to also detect shelterbelts, grass, field, crop, roads and buildings that are imprecisely delimited with a bounding box.

Training algorithms in an agricultural context TractorEYE detection algorithms have been trained mostly using existing datasets in general context images. In an actual application, detection algorithms need to be fine-tuned on data from an agricultural environment to exploit context.

Object tracking High-level fusion of temporal information have been incorporated using occupancy grids and mapping – more on fusion and mapping in *8.2 Fusion and Mapping.* However, more low level multi-object tracking procedures are yet to be explored for moving obstacles in either image frame (2D) [136] / world (3D) coordinates using visual cues [137]. RNN based networks have also been combined to track facial landmarks [57].

7 Sensor Registration & Detection Alignment

Multiple sensor technologies and detection algorithms will output information in various formats. To utilize the advantages of a multi-modal system in agriculture, the output format should be mapped to a common representation/format that is interpretable by a robot. The purpose of this chapter is to represent detection information from cameras in a common representation as either 3D position or as Inverse Sensor Models. The chapter is divided in sensor registration and detection alignment. Chapter contributions are presented in Figure 7.1

7.1 Sensor Registration

7.1.1 Thermal-visual Calibration Panel

A thermal calibration panel is proposed in P[6] for calibrating both thermal and rgb cameras (intrinsic parameters) and determining the registration between them (extrinsic parameters) using a single calibration panel. Two thermal calibration panels were tested for calibration and registration. The core property of a calibration element is that similar points are detectable in both modalities.

Figure 7.2 shows the first thermal calibration panel version used in D2. A heat pillow was used for heating a blue metal plate and a white reflective surface with carved squares is placed on top. Heat from the heat pillow ensures that blue squares are distinguishable from the white front panel both in thermal and rgb images. A script was developed to automatically detect corner candidates using color transformations and a Harris corner detector. An annotation tool was developed to manual adjust incorrect or missing corners. The calibration board created an inefficient and imprecise solution. The manual adjustment of square corners was tedious and poor carvings made annotations imprecise.

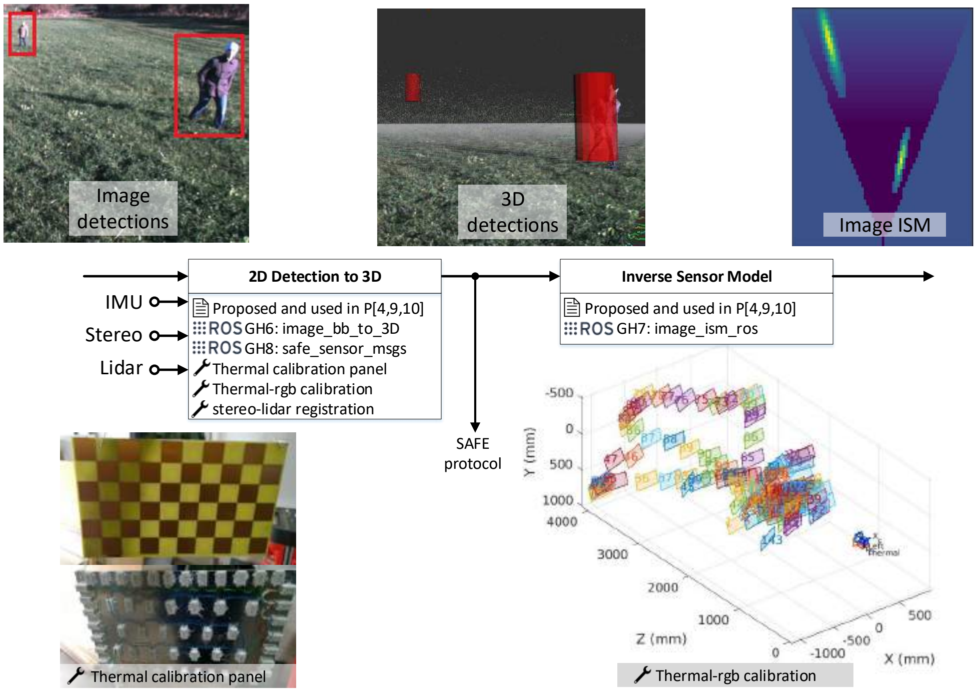


Figure 7.1: Contributions of Chapter 7 is divided in two subsections Sensor Registration and Detection Alignment. In Sensor registration, stereo camera, thermal camera and lidar are registrated. A thermal calibration panel is proposed to do rgb-thermal calibration and registration. In Detection Alignment, detections are transformed to a common format as either 3D detection or image ISM. Software contributions is GH6 for transforming detection in to 3D positions, GH7 for generating ISMs and GH8 is the safe protocol.

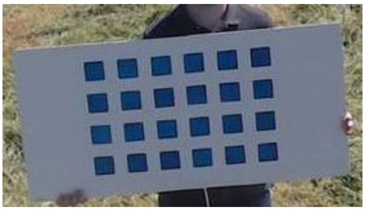
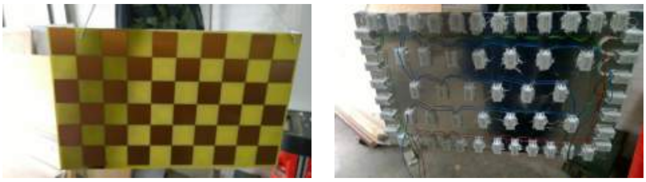


Figure 7.2: First version of the thermal calibration panel

A second thermal calibration panel was proposed. Figure 7.3 a) and b) presents front and back.

An A4 sized circuit board was printed to precisely create cobber squares in a checkerboard pattern. The low emissivity coefficient of the cobber coding makes cobber squares act as reflectors for the thermal camera. A metal plate was mounted on the back with 60 power resistors to deliver 216 W using a 12V car battery.

Non-cobber squares will emit heat from power resistors and cobber squares will reflect the (colder) surrounding environment to create a distinct transition between the two surface materials. The advantage of a checkerboard pattern is that existing toolboxes are able to



(a) (b)

Figure 7.3: Front and back of thermal calibration panel. From P[6]

automatically detect checkerboards in the images. Unfortunately, the checkerboard detection function from MATLAB was unable to detect the not so distinct color transitions between squares in the rgb images. A procedure to emphasize checkerboards are described in P[6] and improved in P[4]. A script is developed to manually segment an area inside the checkerboard. Thermal images are normalized (shifted and scaled) according to the selected area. The MATLAB checkerboard detection algorithm is evaluated for four color transformations of the image to improve chances of detection. First, the input image is transformed to the LAB-color space. Four image transformations are generated from the L- and A-channels by both normalizing and histogram equalizing according to the selected area. Figure 7.4 presents a raw rgb image, a processed rgb image and the normalized thermal image.

Extrinsic parameters (the displacement between thermal and rgb cameras) are estimated as for a stereo camera using hardware synchronized thermal and rgb images. Registration results are presented in Figure 7.5 showing detected checkerboards and displacement between the thermal and left rgb camera.

7.1.2 Stereo to lidar Registration

To complete registration, the extrinsic parameters between the lidar and the cameras (thermal and stereo) needs to be determined. Extrinsic parameters are found using the Iterative Closest Point (ICP) [138] algorithm by determining the transformation that aligns stereo and lidar point clouds. To improve the registration, a static scene with distinct 3D structures and many “3D corners” should be used to ensure a unique/convex solution. Scenes of only a flat surface or an edge between two surfaces have an infinite number of solutions. Figure 7.6 (b) presents registration between lidar and stereo camera.

7.2 Detection Alignment

The purpose of detection alignment is to map detection information from multiple detection algorithms and sensors with different “output formats” into a common format / representation by either estimating the 3D bounding boxes of 2D image detections or by generating Inverse Sensor Models for each algorithm and sensor.

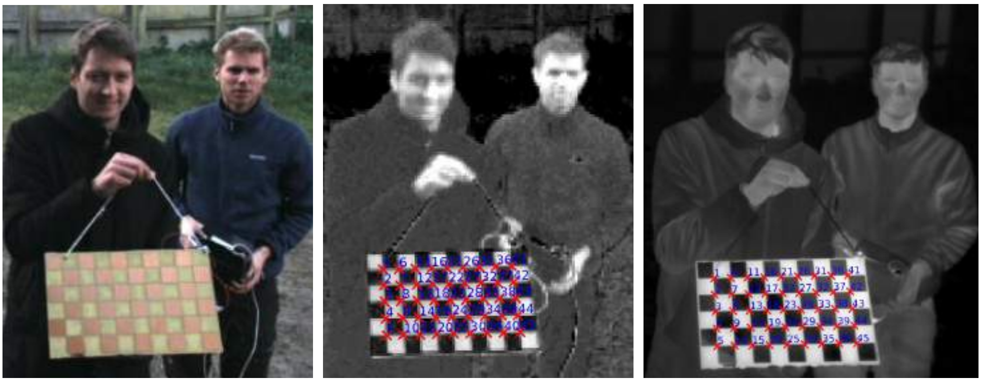


Figure 7.4: Thermal calibration panel as rgb, processed rgb and normalized thermal

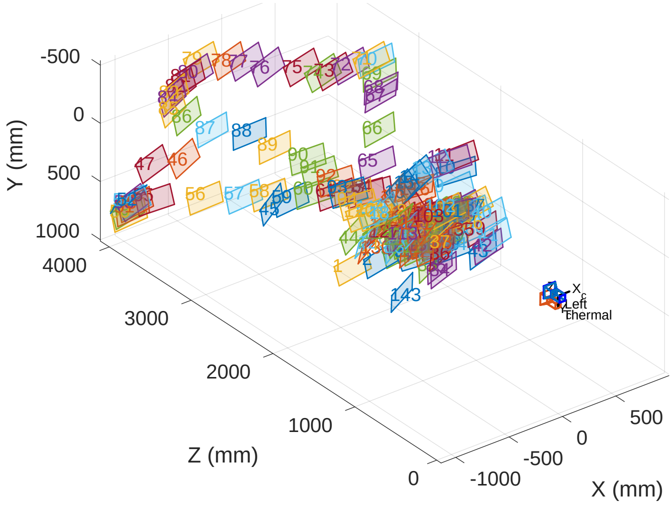
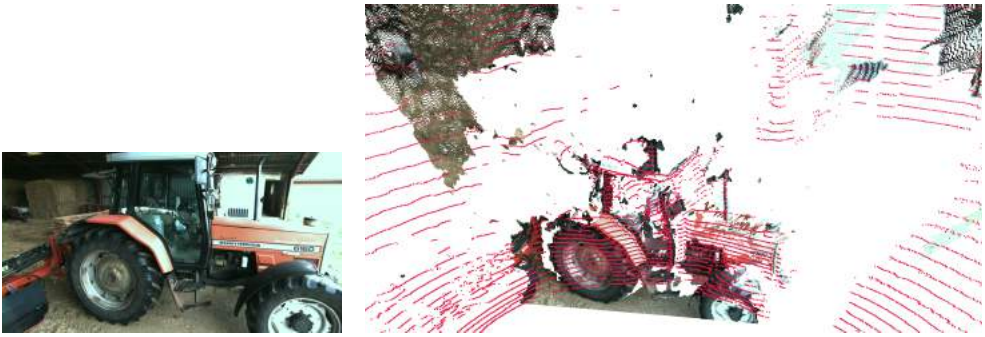


Figure 7.5: Detected checkerboards and displacement of thermal and left rgb camera.



(a)

(b)

Figure 7.6: (a) Image from the stereo camera (b) Registration between the lidar and the stereo point cloud.

7.2.1 Mapping of image detections to 3D bounding boxes

An image detection can be mapped to a 3D bounding box by first estimating the distance to an object (Figure 7.7). A ROS package (GH6) have been developed for converting an image 2D bounding box to 3D by estimating the distance of a detection using either information of the static camera position (tf-tree) or stereo disparity.

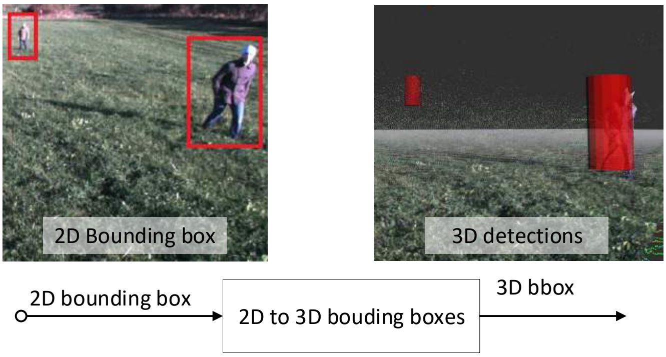


Figure 7.7: Estimating 3D bounding box from 2D image detection.

The distance is used for mapping the four 2D bounding box corners to world coordinates. The bounding box position and extend is derived in 3D and represented as a cylinder (position, height and width) as shown in Figure 7.7.

Heuristics to estimating the distance of a bounding box is listed below.

- **Using a static ground plane assumption** The camera is mounted statically above a flat surface with a fixed angling. The row position of a detection (lower bar) can be mapped to a distance on the ground plane, see Figure 7.8.
- **Using a dynamic ground plane assumption** is similar to a static ground plane assumption. However, the camera angling is constantly updated using an IMU. A tf-tree is specified in the ROS-package.
- **Using stereo matching** In a stereo setup where the disparity map is aligned with the bounding box detections, the disparity map can be used for estimating the distance. The average, median or percentile distance inside the bounding box can be used. The median or the closest 5-10% percentile will avoid being influenced by very distant depth measurements or a few very close points.
- **Using other depth sensors** Depth measurements from sensors that are not directly aligned with the image such as the lidar. The intrinsic camera parameters and the extrinsic (displacement between sensors) can be used for projecting depth measurements from a depth sensor to the image frame. After projecting depth information to the camera image frame, the distance to a detection can be estimated as for a stereo camera setup. Furthermore, the stereo point cloud or the lidar depth measurements can be projected onto the thermal camera image frame to estimate distances for the thermal camera. Unfortunately, projection of depth measurements between sensors has not yet been implemented in (GH6).

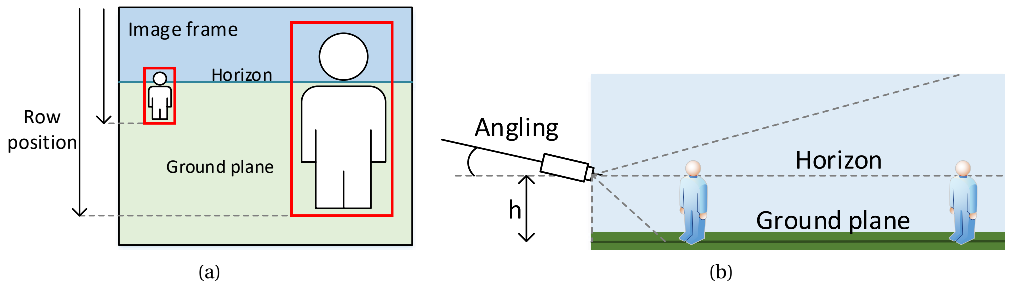


Figure 7.8: A row position in an image maps to a specific distance for a statically mounted camera on a flat surface a) Image frame with detection b) Static camera angling to flat ground surface

7.2.2 Inverse Sensor Models

The purpose of Inverse Sensor Models (ISM) is to convert detections from multiple sensors and algorithms with different "output formats" into a common grid-based format. The format allows information from multiple sensors and algorithms to be registered and fused.

Contributions related to the Image inverse sensor model (ISM) module is

- One publication P[9] describing heuristics for generating ISMs.
- In two publications P[4, 10], ISMs have been used for fusing detection information from multiple algorithms in a multi-modal system.
- A ROS package (GH7) for mapping a detection into an ISM. One function converts 3D detection to an ISM. The second function converts images to an ISM using an Inverse Perspective Mapping (IPM).

Occupancy grid maps are evenly spaced grid maps commonly used in probabilistic robotics [134]. The purpose of an occupancy grid maps is to generate a map by estimating the posterior probability over maps given a sensor measurement. Each grid cell contains a probabilistic measure of occupancy using a real valued number in the interval [0, 1]. A value of 0 represents unoccupied, a value of 1 represents occupied and 0.5 represents unknown.

An Inverse Sensor Model (ISM) is a local grid map representation generated from typically a range sensor as presented in Figure 7.9 (a). This local map is then merged with a global occupancy grid map. The image inverse sensor model comprises both a detection algorithm and the generation of a local grid map from detections, see Figure 7.9 (b). This section only deals with the step of converting detections to local grid maps.

7.2.3 Image ISM - Inverse Perspective Mapping

Inverse Perspective Mapping (IPM) is a geometrical transformation that projects image to ground plane surface [139, 140]. For a flat surface, the perspective effect is removed by transforming the viewpoint from camera to bird's eye view. In GH7, the camera configuration is specified explicitly (height and angle relative to ground surface) or by a tf-tree. The homography for mapping image coordinates to surface is defined by intrinsic camera parameters, camera

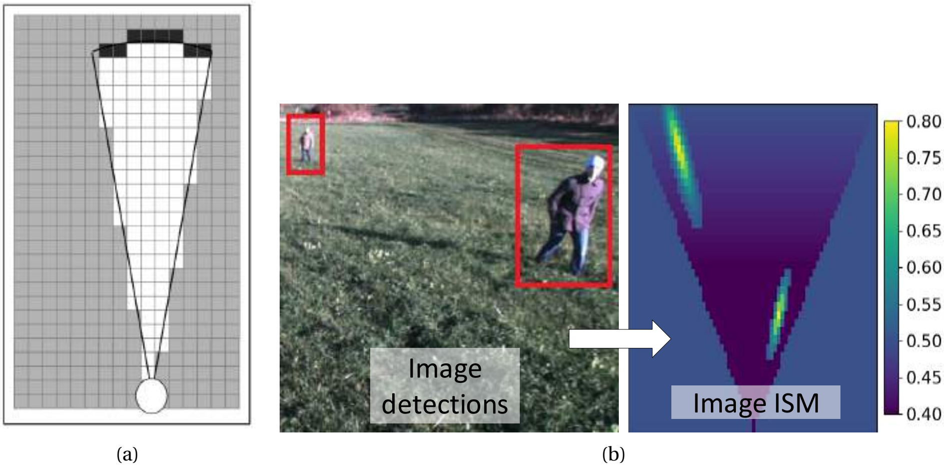


Figure 7.9: (a) Range inverse sensor model, from [134] (b) Image detection are generated into an image inverse sensor model.

to surface transformation and surface to ISM transformation. Figure 7.10, presents rgb image before and after Inverse Perspective mapping.

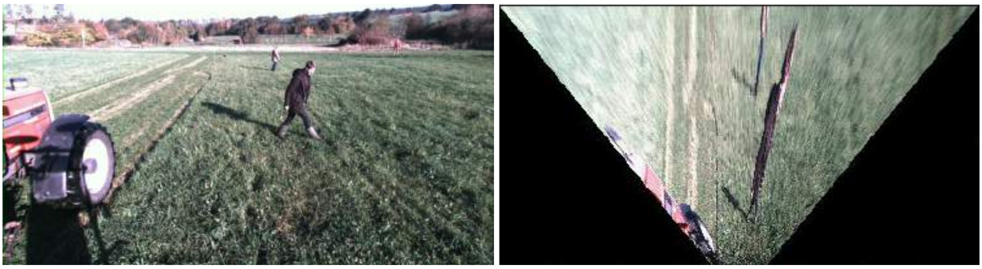


Figure 7.10: Inverse Perspective Mapping of RGB image. From P[9]

Similar to an rgb image, a detection image can be mapped to an ISM using an IPM. Figure 7.11 presents FCN semantic segmentation predictions for human and grass before and after IPM.

Values of ISM are the probability of a grid cell being occupied for a giving obstacle. As presented in Figure 7.11, the area that are not visible by the camera is set to 0.5 - representing that no information is provided for areas that are not visible to the camera. Visible areas with no detections are set below 0.5 to indicate that the area is not expected to be occupied by the given class. Values above 0.5 indicate that the area is expected to be occupied by the given class. For detecting flat elements such as road lane markings or the grass class, the IPM algorithm is able to provide good approximations of the actual inverse perspective mapping. Tall element violate the IPM ground plane assumption and will stretched elements unnaturally/incorrectly across large areas as presented for the human class predictions.

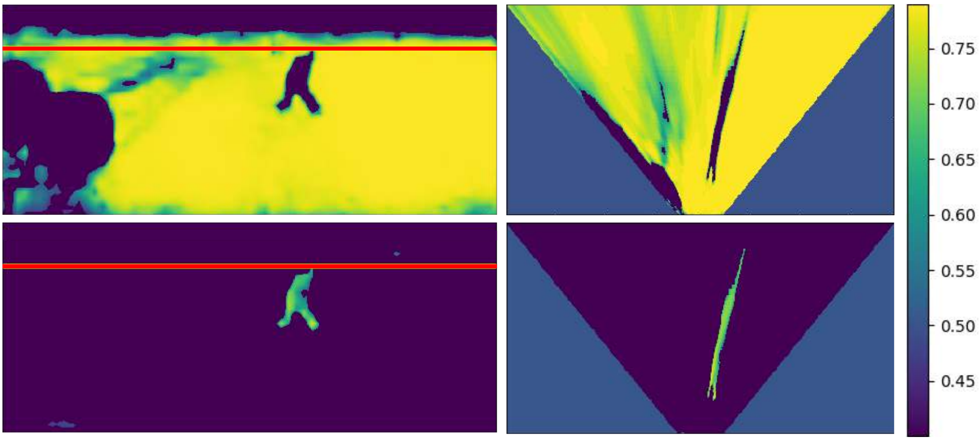


Figure 7.11: (Left) Grass and human predictions by the FCN for semantic segmentation (Right) Using Inverse Perspective Mapping to generate ISMs grass and human class. P[9]

7.2.4 ISM from 3D Bounding Boxes

To avoid “stretching artifacts” of tall objects a second approach is used. An ISM can be generated using 3D bounding boxes as presented in Figure 7.12.

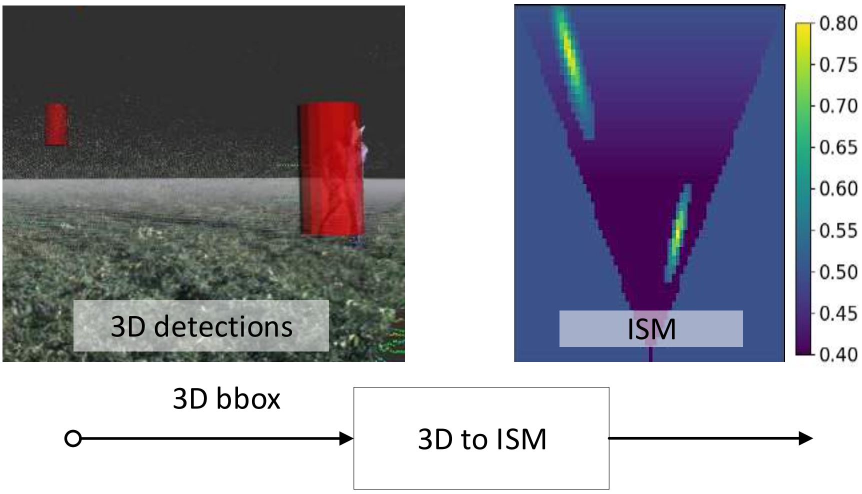


Figure 7.12: 3D Bounding box detection to ISM

Heuristics for mapping 3D detections to an ISM is listed below.

- Areas that are outside the camera FOV is mapped to 0.5. Representing that no information is provided for these areas.
- Areas with no detections inside the FOV are set below 0.5. Values below 0.5, indicate that the algorithm by some certainty is able to reject the existence of an obstacle in a visible areas.

- Most detection algorithms will degrade by the distance. This is incorporated by cropping the ISM beyond a certain distance and by linearly reducing the certainty of not detecting obstacles by the distance. In Figure 7.12, the area inside the FOV increases from 0.4 to 0.5.
- Detections are mapped to values above 0.5 with a Gaussian distribution to indicate that the position of an obstacle is uncertain. The localization uncertainty for a camera is independent for the radial coordinate (distance to the object) and angular coordinate (angle to object). To incorporate this, the polar coordinate (distance and angle) is modelled with two independent uncertainties as in Figure 7.12, where the localization uncertainty of the radial coordinate is larger than the angular coordinate.
- Localization accuracy degrades by the distance. Two uncertainties can be specified for each component of the polar coordinate to allow the uncertainty of each component to increase linearly by the distance.

7.2.5 The mapping of all detection algorithms to an ISM

The pipeline for mapping all detection algorithm into ISMs is presented in Figure 7.13. The overall principle is that objects are be mapped to ISMs using 3D detections and other elements such as grass, water and ground are mapped to ISMs using an IPM. The bounding box based algorithms (YOLOv2 and LDCF) are converted to an ISM using 3D bounding boxes. HeatDetections and five FCN8 classes (Ground, Field, Water, Shelterbelt and Building) are mapped using IPM. Detections from DeepAnomaly and two FCN8 classes (Human and OtherObstacles) are rearranged into connected components and then bounding boxes. These bounding boxes are then mapped first to 3D and then ISMs. Finally, all classes from multiple algorithms have been mapped into ISMs.

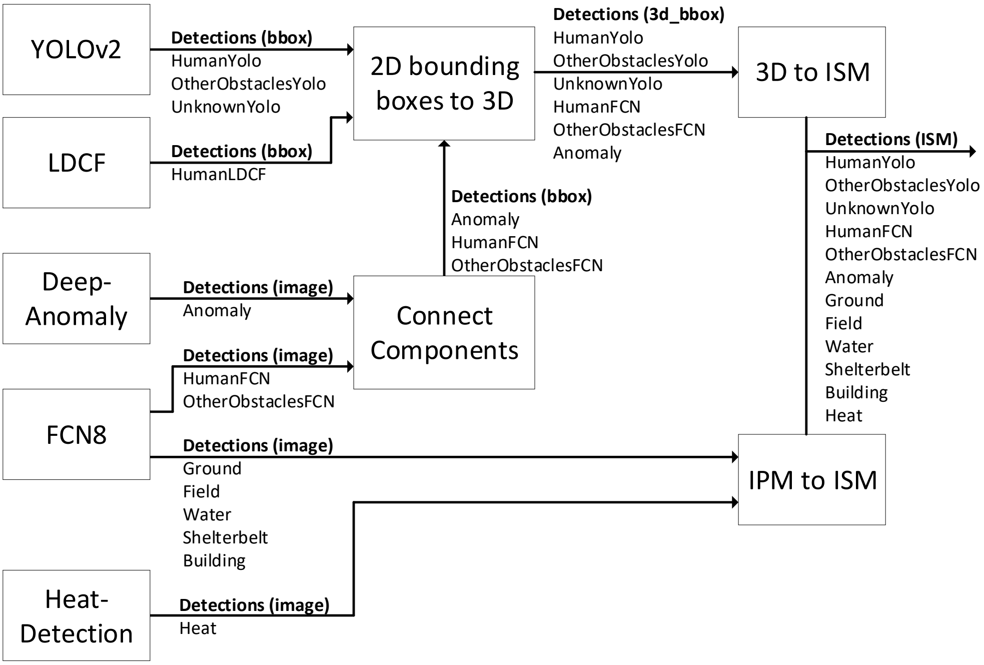


Figure 7.13: The flow from detections to ISM for all camera algorithms

7.3 Concluding Remarks

Multiple sensor technologies and detection algorithms will output information in various formats. To utilize the advantages of a multi-modal system in agriculture, the output format should be mapped to a common representation/format that is interpretable by a robot. The procedure for doing this is twofold. First, Sensor Registration procedures have been presented to calibrate and register thermal camera, stereo camera and lidar. Specifically, a thermal-visual calibration panel have been proposed in P[6] to calibrate and register thermal and rgb cameras. Secondly, procedures have been proposed to map camera detections into a common format as either 3D detections or an ISM representation. Various heuristics have been used to map image detections in 2D to 3D detections using either a static camera assumption or depth from a stereo matching algorithm. Finally, two procedures have been proposed in [9] to generate ISMs using either 3D detections or inverse perspective mapping. The ISM representation is used for fusing multiple sensor technologies and algorithms in P[4]. The 3D detection representation have been used in the SAFE protocol (GH8) and by the BallBot robot presented in Figure 4.11 and a video demo[1]. To be used in an actual autonomous robot application, ROS-package have been implemented (GH6 and GH7) to respectively map detection to 3D and generate ISMs.

7.3.1 Challenges and Future Work

The proposed generation of ISMs from detection algorithms involves many heuristics. Currently, the output of algorithms does not represent actual probabilities. Bayesian deep learning - shortly described in 6.8 Concluding Remarks - would enable this. Furthermore, procedures for better estimating the actual localization uncertainties should be used. Different procedures for generating ISMs from detection have been presented. It would be relevant to compare and evaluate these procedures using quantitative measures. Finally, the ros package (GH7) for generating ISMs can be improved and extended. It is possible to estimate distance to objects using other depth sensors. This would allow the rgb camera to use lidar data to estimate distance of detections and the thermal camera to use stereo or lidar data to estimate the distance to detections.

[1] Video demo is available at YouTube *https://youtu.be/KDa_y-RfkhM?t=109*

8 Localization, Fusion and Mapping

The final processing step is to fuse information from multiple sensors and algorithms into maps. Grid-based maps are an important representation for autonomous systems to represent its surroundings and navigate accordingly. Maps are in this thesis created using occupancy grid maps using information (ISMs) provided by TractorEYE.

Occupancy grid maps [141] are evenly spaced grid maps commonly used in probabilistic robotics [134]. Each grid cell is a stochastic variable representing the probability of an area being occupied. The purpose of an occupancy grid maps is to generate a map by estimating the posterior probability over maps given a sensor measurement - an ISM - and the robot pose. In other words, fusion and mapping will iteratively build up maps using the robot pose and ISMs from multiple sensors and algorithms. Contributions of this chapter is described in P[4, 10] and provide a quantitative evaluation of sensor modalities and algorithms. Occupancy grid maps are a valuable tool to incorporate uncertainty, fuse multiple detections algorithms and sensor models on decision level.

8.1 Localization

To build occupancy grid maps the robot pose is estimated. Localization is performed by the ROS robot_localization package [142] using the extended kalman filter node (ekf_localization_node) and the navsat_transform_node to integrate GPS coordinates. The kalman filter estimates tractor pose based on the VectorNav IMU and heading and position provided by the differential RTK GPS.

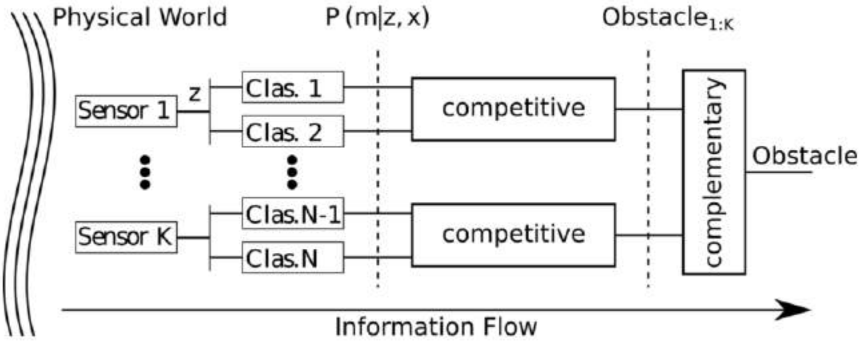


Figure 8.1: Algorithms for one sensor is fused competitively (maximum). Multiple sensors are fused using complementary fusion.

8.2 Fusion and Mapping

Fusion and mapping into occupancy grid maps is achieved using the mapserver ROS package [143] developed by Timo Korthals a collaborator and co-author of P[4, 9, 10]. The mapserver is a occupancy grid server framework with a generic sensor interface. In P[10] sensors use the maximum method to get a competitive fusion across all algorithms from a particular sensor. Each sensor is then fused using a Superbayesian method to get a complementary fusion across all sensors. This is presented in Figure 8.1.

The algorithm is evaluated on dataset D3 using a binary representation (occupied or not occupied). Algorithms and sensors are compared quantitatively. Results are presented in P[10].

The work in P[4] is an elaborated extension to P[10] and summarizes most contributions of this dissertation. Unfortunately, it has not been possible to complete the work of P[4] prior to dissertation submission and only a draft of P[4] has been attached in the dissertation – submission date is 13.10.2017. P[4] will comprises all implemented ROS packages (LDCF, YOLOv2, FCN, DeepAnomaly, dynamic thermal detection) with better and synchronized sensors (SuperSensorKit+MiniSensorKit). A more complicated scheme for better ISM generation is used as presented in Figure 7.13. Finally, the setup is evaluated on the FieldSAFE (D6) dataset which both includes ground truth for static and moving obstacles. An important contribution is the quantitative evaluation and comparison between sensor modalities and algorithms.

9 Contribution Overview

Figure 9.1 presents an overview of all major contributions in the thesis. The Figure is an extended version of Figure 1.1 and 3.1 by representing published papers 📄, engineering products 🔧 and developed ROS packages ⁞⁞⁞ROS with small icons. Though, the details presented in the figure are overwhelming, the Figure is able to compactly capture the covered areas and all major contributions.

Chapter 3 and 4 presents sensor platform and data collections. The sensor platform is developed and described in three publications P[3, 6, 8]. The sensor platforms and collected data is used in multiple papers P[1, 4, 5, 7, 10, 11]. Engineering contributions are SuperSensorKit and MiniSensorKit, field trials (D1-D6) and ground truth of (D5 and D6). Ground truth of D5 is annotations of static obstacles in an orthophoto. Ground truth of D6 (FieldSAFE) is ground truth of both static and moving obstacles using an annotated orthophoto and annotations of footage from a hovering drone.

Chapter 5 presents all detection algorithms. A total of five detection algorithms (LDCF [17], YOLOv2 [99], FCN[19], P[1] and DynamicHeat) have been implemented as ROS packages (GH1, GH2, GH3, GH4 and GH5) and used by TractorEYE. DeepAnomaly and DynamicHeat is proposed in this thesis. Additionally, two detection algorithms – not used in TractorEYE – have been proposed in P[2, 5].

Chapter 6 presents registration of sensors and detections. Stereo camera, thermal camera and lidar are calibrated and registered in P[3, 4, 6, 10]. A thermal calibration board is proposed to perform thermal-visual calibration and registration P[6]. Information from multiple sensors are registered using an ISM representation P[4, 9, 10]. ROS packages for converting 2D bounding boxes to 3D (GH6), converting detection to image ISMs (GH7) and SAFE message types (GH8).

Chapter 7 presents sensor fusion and mapping in P[4, 10]

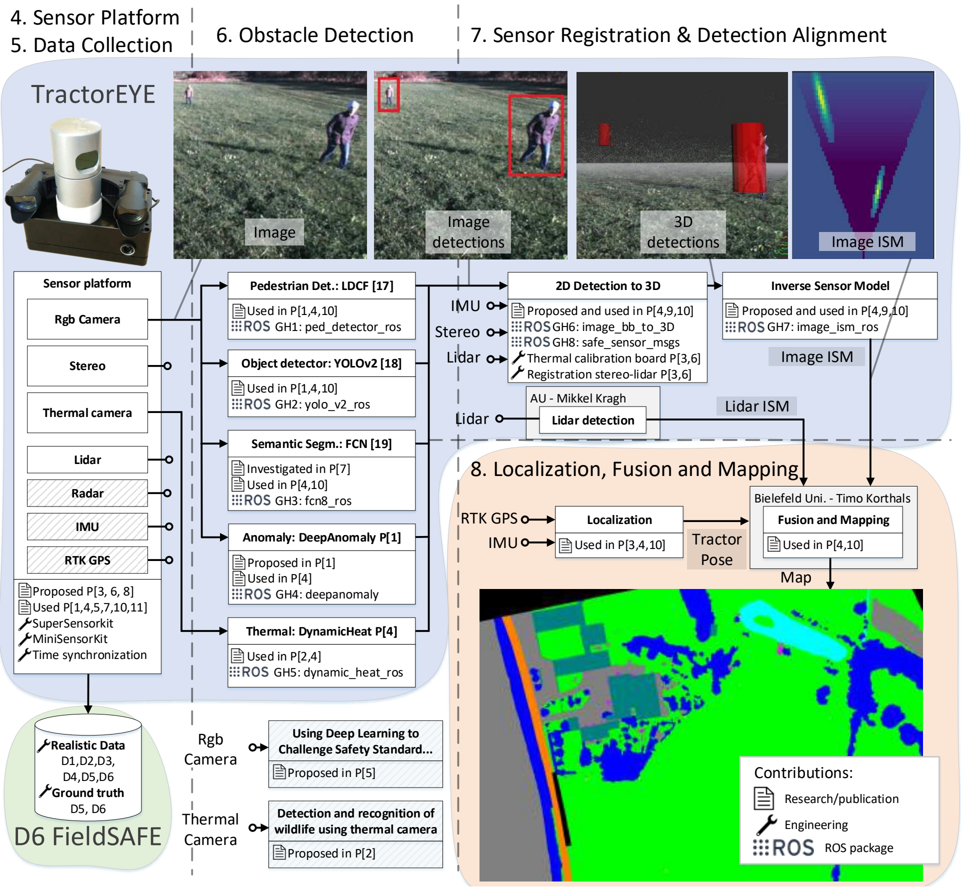


Figure 9.1: Algorithms for one sensor is fused competitively (maximum). Multiple sensors are fused using complementary fusion.

10 Conclusion

Three core contributions of the dissertation is a real-time detection system for autonomous vehicles in agriculture (TractorEYE), a dataset for object detection in agriculture (FieldSAFE) and procedures to fuse information from multiple detection algorithms.

TractorEYE is a complete multi-sensor detection system - comprising thermal camera, stereo camera and multi-beam lidar. Sensors are mounted in a water-resistant and ruggedized chasing suited for agriculture and to ensures stable registration between sensors. A total of six fast detection algorithms have been implemented[1] as ROS packages to run real-time on a GPU platform. Calibration, registration and synchronization procedures have been proposed to ensure that detection information can be registered in a common grid map representation. Two of the six TractorEYE detection algorithms are proposed in this work. A simple dynamic heat detection algorithm for thermal camera and DeepAnomaly. DeepAnomaly is a fast CNN-based anomaly detection algorithm for rgb camera. DeepAnomaly have for an agricultural use case demonstrated high detection accuracy and is able to detect unknown, distant and very occluded objects in an agricultural field. For a specific human detection use case, DeepAnomaly is faster and provides better detection accuracy than the state-of-the-art detection algorithms (Faster R-CNN, YOLO, FCN). Additional two papers have been published on detection. One paper for using a thermal heat signature to automatically detect and recognize wildlife from a drone. The second paper uses a CNN-based object detector to challenging a safety standard.

FieldSAFE is a multi-modal dataset for detection of static and moving obstacles in agriculture. The dataset includes webcam, stereo camera, thermal camera, 360-degree camera, lidar and radar. Precise localization and pose is provided using IMU and GPS. Ground truth of static and moving obstacles (humans, mannequin dolls, rocks, barrels, buildings, vehicles, and vegetation) are available as static annotated orthophoto and GPS coordinates for moving obstacles.

[1] The detection algorithm for Velodyne is developed by Mikkel Fly Kragh

Localization, Fusion and Mapping Finally, the detection algorithms from TractorEYE are fused into a map. Sensor recordings and ground truth data from FieldSAFE allows the individual algorithms and sensors to be evaluated and compared for object detection in agriculture.

This thesis have made many scientific contribution and is state-of-the-art within perception for autonomous tractors. The whole pipeline for a perception system have been developed which includes a dataset *FieldSAFE,* sensor platforms *SuperSensorKit & MiniSensorKit,* detection algorithms such as *DeepAnomaly* and procedures to perform multi-sensor fusion.

A critical deficiency for autonomous farming vehicles and state-of-the-art algorithms is the detection of hardly visible and unknown obstacles that reside inside the crop. DeepAnomaly solves exactly this critical issue by detecting very distance, heavy occluded and unknown obstacles.

Furthermore, important engineering contributions to autonomous farming vehicles have been developed such as easily applicable, open-source software packages and algorithms. These contributions have been demonstrated in an end-to-end real-time detection system *TractorEYE* and used on an actual robot *BallBot.* The contributions of this thesis have demonstrated, addressed and solved critical issues to utilize camera-based perception systems in a multi-sensor setup that are essential to make autonomous vehicles in agriculture a reality.

Bibliography

[1] Peter Christiansen, Lars N Nielsen, Kim A Steen, Rasmus N Jørgensen, and Henrik Karstoft. DeepAnomaly: Combining background subtraction and deep learning for detecting obstacles and anomalies in an agricultural field. *Sensors*, 16(11), 11 November 2016.

[2] Peter Christiansen, Kim Steen, Rasmus Jørgensen, and Henrik Karstoft. Automated detection and recognition of wildlife using thermal cameras. *Sensors*, 14(8):13778–13793, 30 July 2014.

[3] Mikkel Kragh, Peter Christiansen, Morten Stigaard Laursen, Morten Larsen, Kim A. Steen, Ole Green, Henrik Karstoft, Rasmus N. Jørgensen. FieldSAFE: Dataset for obstacle detection in agriculture. *arXiv*.

[4] Timo Korthals, Mikkel Kragh, Peter Christiansen, Henrik Karstoft, Rasmus N. Jørgensen and Ulrich Rückert. (not published) multi-modal detection of static and dynamic obstacles in agriculture for process evaluation. *Multi-modal Sensor Fusion*, 1st(1st), 2017.

[5] Kim Steen, Peter Christiansen, Henrik Karstoft, and Rasmus Jørgensen. Using deep learning to challenge safety standard for highly autonomous machines in agriculture. *Journal of Imaging*, 2(1):6, 15 February 2016.

[6] P Christiansen, M Kragh, K A Steen, H Karstoft, and R N Jørgensen. Platform for evaluating sensors and human detection in autonomous mowing operations. *Precis. Agric.*, 18(3): 350–365, 1 June 2017.

[7] Peter Christiansen, Rene Sørensen, Søren Skovsen, Claes D Jæger, Rasmus Nyholm Jørgensen, Henrik Karstoft, and Kim Arild Steen. Towards autonomous plant production using fully convolutional neural networks. 26 June 2016.

[8] P Christiansen, M K Hansen, K A Steen, H Karstoft, and R N Jørgensen. Advanced sensor platform for human detection and protection in autonomous farming. In *Precision agriculture '15*, chapter 34, pages 291–298.

[9] Timo Korthals, Mikkel Kragh, Peter Christiansen and Ulrich Rückert. Towards inverse sensor mapping in agriculture.

[10] Mikkel Kragh Hansen, Peter Christiansen, Timo Korthals, Thorsten Jungeblut, Henrik Karstoft, and Rasmus Nyholm Jørgensen. Multi-modal obstacle detection and evaluation of evidence grid mapping in agriculture.

[11] Stefan-Daniel Suvei, Leon Bodenhagen, Lilita Kiforenko, Peter Christiansen, Rasmus N Jørgensen, Anders G Buch, and Norbert Krüger. Stereo and Active-Sensor data fusion for improved stereo block matching. In *Image Analysis and Recognition*, pages 451–461. Springer, Cham, 13 July 2016.

[12] Johann Thor Ingibergsson Mogensen, Stefan-Daniel Suvei, Mikkel Kragh Hansen, Peter Christiansen, and Ulrik Pagh Schultz. Towards a DSL for Perception-Based safety systems. *Towards a Dsl for Perception-based Safety Systems*, 2015.

[13] Ole Green, Gareth Thomas Charles Edwards, Claes D Jæger, Kim Arild Steen, Peter Christensen, Mads Dyrmann, and Rasmus Nyholm Jørgensen. Udviklingen inden for præcisionsjordbrug og tilknyttet udstyr. *Plantekongres 2016*, 2016.

[14] Mads Dyrmann and Peter Christiansen. Automated classification of seedlings using computer vision. 2014.

[15] Mads Dyrmann, Peter Christiansen, and Henrik Skov Midtiby. Estimation of plant species by classifying plants and leaves in combination. *J. Field Robotics*.

[16] R N Jørgensen, M B Brandt, T Schmidt, M S Laursen, R Larsen, M Nørremark, H S Midtiby, and P Christiansen. Field trial design using semi-automated conventional machinery and aerial drone imaging for outlier identification. In *Precision agriculture '15*, 18, pages 151–158. Wageningen Academic Publishers, 22 June 2015.

[17] Woonhyun Nam, Piotr Dollár, and Joon Hee Han. Local decorrelation for improved detection. *Adv. Neural Inf. Process. Syst.*, pages 1–9, 2014.

[18] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, Real-Time object detection. 2016.

[19] J Long, E Shelhamer, and T Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, June 2015.

[20] Mary C Potter, Brad Wyble, Carl Erick Hagmann, and Emily S McCourt. Detecting meaning in RSVP at 13 ms per picture. *Atten. Percept. Psychophys.*, 76(2):270–279, February 2014.

[21] Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Lip reading sentences in the wild. 16 November 2016.

[22] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2 February 2017.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[24] D Ciresan, U Meier, and J Schmidhuber. Multi-column deep neural networks for image classification. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, 2012.

[25] Yaniv Taigman, Marc Aurelio Ranzato, Tel Aviv, and Menlo Park. DeepFace : Closing the gap to Human-Level performance in face verification. June 2014.

[26] W Xiong, J Droppo, X Huang, F Seide, M Seltzer, A Stolcke, D Yu, and G Zweig. Achieving human parity in conversational speech recognition. 17 October 2016.

[27] Pranav Rajpurkar, Awni Y Hannun, Masoumeh Haghpanahi, Codie Bourn, and Andrew Y Ng. Cardiologist-Level arrhythmia detection with convolutional neural networks. 6 July 2017.

[28] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 26 February 2015.

[29] ASI. Autonomous Solutions. https://www.asirobots.com/farming/, 2016. Accessed: 2017-8-9.

[30] Kubota. Kubota. http://www.kubota-global.net/news/2017/20170125.html, 2017. Accessed: 2017-8-16.

[31] Case IH. Case IH Autonomous Concept Vehicle. http://www.caseih.com/apac/en-in/news/pages/2016-case-ih-premieres-concept-vehicle-at-farm-progress-show.aspx, 2016. Accessed: 2017-8-9.

[32] A Geiger, P Lenz, and R Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, June 2012.

[33] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

[34] Mikkel Kragh, Rasmus N. Jørgensen, and Henrik Pedersen. *Object Detection and Terrain Classification in Agricultural Fields Using 3D Lidar Data*, pages 188–197. Springer International Publishing, Cham, 2015. ISBN 978-3-319-20904-3. doi: 10.1007/978-3-319-20904-3_18. URL https://doi.org/10.1007/978-3-319-20904-3_18.

[35] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 28 May 2015.

[36] A Krizhevsky, I Sutskever, and Ge Hinton. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.*, pages 1097–1105, 2012.

[37] Ross Girshick, Jeff Donahue, Trevor Darrell, U C Berkeley, and Jitendra Malik. (DeepInsight R-CNN) rich feature hierarchies for accurate object detection and semantic segmentation. *Cvpr'14*, pages 2–9, 2014.

[38] Sachin Sudhakar Farfade, Mohammad Saberian, and Li-Jia Li. Multi-view face detection using deep convolutional neural networks. *Cornell University Library*, 2015.

[39] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y Ng. Deep speech: Scaling up end-to-end speech recognition. 17 December 2014.

[40] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *arXiv preprint arXiv:1311.2901*, 2013.

[41] Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *Int. J. Autom. Comput.*, pages 1–17, 14 March 2017.

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for Large-Scale image recognition. pages 1–13, 4 September 2014.

[43] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. 10 July 2017.

[44] Alex Berg and J Deng. Imagenet large scale visual recognition challenge 2015. *Challenge*, 2015.

[45] Mark Everingham, Sma Eslami, and Luc Van Gool. The pascal visual object classes challenge–a retrospective. *Homepages.Inf.Ed.Ac.Uk*, 2013.

[46] J Stallkamp, M Schlipsing, J Salmen, and C Igel. Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition. *Neural Netw.*, 32:323–332, August 2012.

[47] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proc. CVPR*, 2017.

[48] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, Jie Chen, Jingdong Chen, Zhijie Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Ke Ding, Niandong Du, Erich Elsen, Jesse Engel, Weiwei Fang, Linxi Fan, Christopher Fougner, Liang Gao, Caixia Gong, Awni Hannun, Tony Han, Lappi Johannes, Bing Jiang, Cai Ju, Billy Jun, Patrick LeGresley, Libby Lin, Junjie Liu, Yang Liu, Weigao Li, Xiangang Li, Dongpeng Ma, Sharan Narang, Andrew Ng, Sherjil Ozair, Yiping Peng, Ryan Prenger, Sheng Qian, Zongfeng Quan, Jonathan Raiman, Vinay Rao, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Kavya Srinet, Anuroop Sriram, Haiyuan Tang, Liliang Tang, Chong Wang, Jidong Wang, Kaifu Wang, Yi Wang, Zhijian Wang, Zhiqian Wang, Shuang Wu, Likai Wei, Bo Xiao, Wen Xie, Yan Xie, Dani Yogatama, Bin Yuan, Jun Zhan, and Zhenyao Zhu. Deep

speech 2 : End-to-End speech recognition in english and mandarin. In *International Conference on Machine Learning*, pages 173–182, 11 June 2016.

[49] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. *arXiv Preprint*, 2014.

[50] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks ? *Nips 2014*, 27, 2014.

[51] S Hochreiter and J Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 15 November 1997.

[52] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. 11 December 2014.

[53] K Fukushima. Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.*, 36(4):193–202, 1980.

[54] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, R E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a Back-Propagation network. In D S Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 396–404. Morgan-Kaufmann, 1990.

[55] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[56] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2325–2333, 2016.

[57] Jinwei Gu Xiaodong Yang Shalini De and Mello Jan Kautz. Dynamic facial analysis: From bayesian filtering to recurrent neural network.

[58] Ian Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. 31 December 2016.

[59] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

[60] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. 3 July 2012.

[61] David Parker. Learning logic (report no. 47). cambridge, massachusetts institute of technology. *Center for Computational Research in Economics and Management Science*, 1985.

[62] Yann LeCun. Une procédure d'apprentissage pour réseau a seuil asymmetrique (a learning scheme for asymmetric threshold networks). In *Proceedings of Cognitiva 85, Paris, France*. 1985.

[63] Drghr Williams and Geoffrey Hinton. Learning representations by back-propagating errors. *Nature*, 323(6088):533–538, 1986.

[64] Paul John Werbos. Beyond regression: New tools for prediction and analysis in the behavioral sciences. *Doctoral Dissertation, Applied Mathematics, Harvard University, MA*, 1974.

[65] Y LeCun, L Bottou, Y Bengio, and P Haffner. Gradient based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

[66] Christian Szegedy, Scott Reed, Pierre Sermanet, Vincent Vanhoucke, and Andrew Rabinovich. (GoogLeNet) going deeper with convolutions. pages 1–12, 2014.

[67] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. 16 December 2013.

[68] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift.

[69] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. 2 December 2015.

[70] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, Inception-ResNet and the impact of residual connections on learning. 23 February 2016.

[71] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. 3 May 2015.

[72] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. (3):171–180, 10 December 2015.

[73] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. 19 December 2014.

[74] Dmytro Mishkin and Jiri Matas. All you need is a good init. 19 November 2015.

[75] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, volume 9908 of *Lecture Notes in Computer Science*, pages 630–645. Springer International Publishing, Cham, 2016.

[76] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. 25 August 2016.

[77] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. 16 November 2016.

[78] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. 23 May 2016.

[79] P Viola and M Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 1:I–511–I–518, 2001.

[80] Michael J Jones and Paul Viola. Robust real-time object detection. In *Workshop on statistical and computational theories of vision*, volume 266, page 56, 2001.

[81] Yoav Freund and Robert E Schapire. A Decision-Theoretic generalization of On-Line learning and an application to boosting. *J. Comput. System Sci.*, 55(1):119–139, August 1997.

[82] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, 2005.

[83] Piotr Dollár, Zhuowen Tu, Pietro Perona, and Serge Belongie. Integral channel features. *BMVC 2009 London England*, pages 1–11, 2009.

[84] Piotr Dollár, Serge J Belongie, and Pietro Perona. The fastest pedestrian detector in the west. In *BMVC*, volume 2, page 7, 2010.

[85] P Felzenszwalb, D McAllester, and D Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[86] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, September 2010.

[87] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. *arXiv preprint arXiv . . .*, cs.CV:1–15, 2014.

[88] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. 20 March 2017.

[89] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. OverFeat: Integrated recognition, localization and detection using convolutional networks. 21 December 2013.

[90] J R R Uijlings, K E A van de Sande, T Gevers, and A W M Smeulders. Selective search for object recognition. *Int. J. Comput. Vis.*, 104(2):154–171, 1 September 2013.

[91] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, pages 391–405. Springer, Cham, 6 September 2014.

[92] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *arXiv preprint arXiv . . .*, cs.CV:1–14, 2014.

[93] Ross Girshick. (fast object) fast R-CNN. 2015.

[94] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time object detection with region proposal networks. In C Cortes, N D Lawrence, D D Lee, M Sugiyama, and R Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[95] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. 30 November 2016.

[96] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. DenseBox: Unifying landmark localization with end to end object detection. 16 September 2015.

[97] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. DSSD : Deconvolutional single shot detector. 23 January 2017.

[98] Christian Szegedy, Scott Reed, Dumitru Erhan, Dragomir Anguelov, and Sergey Ioffe. Scalable, High-Quality object detection. 3 December 2014.

[99] J Redmon and A Farhadi. YOLO9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[100] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot MultiBox detector. In *Computer Vision – ECCV 2016*, pages 21–37. Springer, Cham, 8 October 2016.

[101] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: Object detection via region-based fully convolutional networks. In D D Lee, M Sugiyama, U V Luxburg, I Guyon, and R Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 379–387. Curran Associates, Inc., 2016.

[102] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. 9 December 2016.

[103] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898. IEEE, June 2014.

[104] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. *Iclr*, pages 1–12, 2015.

[105] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. 2 June 2016.

[106] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. 17 June 2017.

[107] Philip H S Torr. Conditional random fields as recurrent neural networks. *arXiv preprint*, 2014.

[108] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A deep convolutional Encoder-Decoder architecture for image segmentation. 2 November 2015.

[109] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, pages 234–241. Springer, Cham, 5 October 2015.

[110] R H Hahnloser, R Sarpeshkar, M A Mahowald, R J Douglas, and H S Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789): 947–951, 22 June 2000.

[111] Jonghoon Jin, Aysegul Dundar, and Eugenio Culurciello. Flattened convolutional neural networks for feedforward acceleration. 17 December 2014.

[112] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. 17 April 2017.

[113] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5353–5360, 2015.

[114] Forrest N Iandola, Matthew W Moskewicz, Khalid Ashraf, Song Han, William J Dally, and Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 1MB model size. *arXiv preprint arXiv:1602. 07360*, 2016.

[115] François Chollet. Xception: Deep learning with depthwise separable convolutions. 7 October 2016.

[116] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. 2015.

[117] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In C Cortes, N D Lawrence, D D Lee, M Sugiyama, and R Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1135–1143. Curran Associates, Inc., 2015.

[118] Y Jia. Learning semantic image representations at a large scale. 2014.

[119] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A Horowitz, and William J Dally. EIE: Efficient inference engine on compressed deep neural network. 4 February 2016.

[120] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. XNOR-Net: ImageNet classification using binary convolutional neural networks. In *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 525–542. Springer, Cham, 8 October 2016.

[121] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-Luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz,

Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma, Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon. In-Datacenter performance analysis of a tensor processing unit. 16 April 2017.

[122] Carl Vondrick, Donald Patterson, and Deva Ramanan. Efficiently scaling up crowdsourced video annotation. *Int. J. Comput. Vis.*, 101(1):184–204, 1 January 2013.

[123] Navneet Dalal and Bill Triggs. INRIA person dataset. *Online: http://pascal. inrialpes. fr/data/human*, 2005.

[124] Piotr Dollar, Serge Belongie, and Pietro Perona. The fastest pedestrian detector in the west. *Procedings of the British Machine Vision Conference 2010*, 2010.

[125] P Dollar. Piotr's computer vision matlab toolbox, 2015.

[126] Ron Appel, Serge Belongie, Pietro Perona, and Piotr Doll. Fast feature pyramids for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36:1532–1545, 2014.

[127] Rodrigo Benenson, Markus Mathias, Radu Timofte, and Luc Van Gool. Pedestrian detection at 100 frames per second. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012.

[128] Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. Ten years of pedestrian detection, what have we learned? 16 November 2014.

[129] Joseph Redmon. Darknet: Open source neural networks in c. *h ttp://pjreddie. com/darknet*, 2016, 2013.

[130] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM.

[131] Yarin Gal. *Uncertainty in deep learning.* PhD thesis, PhD thesis, University of Cambridge, 2016.

[132] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? 15 March 2017.

[133] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-Task learning using uncertainty to weigh losses for scene geometry and semantics. 19 May 2017.

[134] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics.* MIT press, 2005.

[135] Lukasz Kaiser, Aidan N Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. One model to learn them all. 16 June 2017.

[136] A Bewley, Z Ge, L Ott, F Ramos, and B Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016.

[137] Wongun Choi, Caroline Pantofaru, and Silvio Savarese. A general framework for tracking multiple people from a moving camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(7): 1577–1591, July 2013.

[138] Paul J Besl, Neil D McKay, and Others. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.

[139] M Bertozzi and A Broggi. GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Trans. Image Process.*, 7(1):62–81, 1998.

[140] M Konrad, D Nuss, and K Dietmayer. Localization in digital maps for road course estimation using grid maps. In *2012 IEEE Intelligent Vehicles Symposium*, pages 87–92, June 2012.

[141] Alberto Elfes. Occupancy grids: A stochastic spatial representation for active robot perception. In *Proceedings of the Sixth Conference on Uncertainty in AI*, volume 2929, 1990.

[142] Thomas Moore and Daniel Stouch. A generalized extended kalman filter implementation for the robot operating system. In *Intelligent Autonomous Systems 13*, Advances in Intelligent Systems and Computing, pages 335–348. Springer, Cham, 2016.

[143] Timo Korthals, Julian Exner, Thomas Schöpping, and Marc Hesse. Semantical occupancy grid mapping framework. 2017.

Publications Part III

Paper 1

DeepAnomaly: Combining Background Subtraction and Deep Learning for Detecting Obstacles and Anomalies in an Agricultural

Peter Christiansen, Lars N. Nielsen, Kim A. Steen, Rasmus N. Jørgensen and Henrik Karstoft

Peer reviewed
Paper published in Sensors, Special Issue Vision-Based Sensors in Field Robotics, November 2016

sensors

Article

DeepAnomaly: Combining Background Subtraction and Deep Learning for Detecting Obstacles and Anomalies in an Agricultural Field

Peter Christiansen [1,*], Lars N. Nielsen [2], Kim A. Steen [3], Rasmus N. Jørgensen [1] and Henrik Karstoft [1]

[1] Department of Engineering, Aarhus University, Aarhus 8200, Denmark; rnj@eng.au.dk (R.N.J.); hka@eng.au.dk (H.K.)

[2] Danske Commodities, Aarhus 8000, Denmark; larsnn@gmail.com

[3] AgroIntelli, Aarhus 8200, Denmark; kas@agrointelli.com

[*] Correspondence: pech@eng.au.dk; Tel.: +45-2759-2953

Academic Editors: Gabriel Oliver-Codina, Nuno Gracias and Antonio M. López

Received: 15 September 2016; Accepted: 7 November 2016; Published: 11 November 2016

Abstract: Convolutional neural network (CNN)-based systems are increasingly used in autonomous vehicles for detecting obstacles. CNN-based object detection and per-pixel classification (semantic segmentation) algorithms are trained for detecting and classifying a predefined set of object types. These algorithms have difficulties in detecting distant and heavily occluded objects and are, by definition, not capable of detecting unknown object types or unusual scenarios. The visual characteristics of an agriculture field is homogeneous, and obstacles, like people, animals and other obstacles, occur rarely and are of distinct appearance compared to the field. This paper introduces DeepAnomaly, an algorithm combining deep learning and anomaly detection to exploit the homogenous characteristics of a field to perform anomaly detection. We demonstrate DeepAnomaly as a fast state-of-the-art detector for obstacles that are distant, heavily occluded and unknown. DeepAnomaly is compared to state-of-the-art obstacle detectors including "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks" (RCNN). In a human detector test case, we demonstrate that DeepAnomaly detects humans at longer ranges (45–90 m) than RCNN. RCNN has a similar performance at a short range (0–30 m). However, DeepAnomaly has much fewer model parameters and (182 ms/25 ms =) a 7.28-times faster processing time per image. Unlike most CNN-based methods, the high accuracy, the low computation time and the low memory footprint make it suitable for a real-time system running on a embedded GPU (Graphics Processing Unit).

Keywords: anomaly detection; obstacle detection; autonomous farming; precision agriculture; camera; background subtraction; change detection; DeepAnomaly

1. Introduction

Anomaly detection refers to the problem of finding patterns in data that do not conform to normal or expected behavior [1]. Using anomaly detection for obstacle detection will, instead of learning/classifying all object types or behavior, model the normal patterns and detects outliers. In an agricultural context, these outliers represent elements that are unnatural to the surrounding environment.

Conventional background subtraction (BS) algorithms are related to anomaly detection, as these methods subtract the background from the image, leaving behind only the foreground, which is an outlier to the background. Traditionally, BS methods model the background using color, intensity or gradients for each pixel using mixture of Gaussians [2], k-nearest neighbor or other classifiers to become invariant to small changes in illumination and moving shadows [3–5]. BS is intended for static cameras and for detecting moving or appearing objects in a video sequence. For a moving camera, the low level features used by conventional BS struggle to model a moving background [5], and many moving camera applications detect obstacles using general object detection algorithms or depth sensors.

Google Car (Google, San Jose, CA, USA) uses a Velodyne LiDAR as a depth sensor to perform convincing obstacle detection, and this is also a valuable sensor for obstacle detection in agriculture [6–8]. The drawbacks of using this sensor is the very high cost and that a depth sensor, especially in the automotive industry, exploits that an obstacle will protrude from the ground surface. In an agricultural context, obstacles may not protrude from the crop surface, introducing the risk of not detecting, e.g., kids, lying humans, hydrants, well covers and animals.

A camera-based system is much less expensive and, in principle, only requires obstacles to be visible and not necessarily protruding. In general, camera-based systems are less applicable for autonomous vehicles in terms of accuracy, range and computation time.

Mobileye is a company developing camera-based real-time systems for the automotive industry that are used in commercially available semi-autonomous vehicles, such as Tesla's Model S. However, solutions by Mobileye are neither accessible to most researchers or trained for agriculture.

In research, deep learning perception algorithms and especially convolutional neural networks (CNN) [9–13] have improved the area of object detection [14–18] and semantic segmentation [19–22]. However, the training is performed on a predefined set of object types, and a large amount of annotated data is required for each object type.

To detect predefined object types, such as humans and some types of animals, data from various benchmarks [23–25] can be used in training. These benchmarks are not intended for agriculture and lack important object classes, such as tractors, fencing, shelter belts, water, etc. Most importantly, if such data were available, the algorithms would, by definition, not be able to detect other unspecified object types or unusual scenarios, e.g., a tent, a large red metal plate or a crashed road vehicle in the field. Secondly, in the context of agriculture, the object detection algorithm and semantic segmentation algorithms struggle to detect objects that are distant and heavily occluded by the crops.

In agriculture, the homogeneous characteristics of an agricultural production field and the fact that obstacles occur rarely and are of distinct appearance compared to the field should be exploited to detect non-predefined obstacles [26–28]. In [29], both distinct appearance (spacial analysis) and motion (temporal analysis) are used for detecting foreground elements.

In this work, the combination of background subtraction algorithms and high level features from a CNN is explored. Low level features used in conventional background subtraction algorithms are replaced with high abstraction features from a CNN. High abstraction features are less invariant to changes in pixel intensities caused by a moving camera and more dependent on actual image content. The intuition is that feature activations are nearly constant for grass, shelter belt or sky for a moving camera until completely new content is introduced in the image. A network trained for image classification on the ImageNet data [25] with 1000 different object types, targets the network features to activate especially on objects. The background model will more easily model the passive features of the background and detect feature activations from foreground objects. Secondly, the method exploits that images taken from a camera in motion (e.g., a tractor) have similar visual characteristics along image rows, as illustrated in Figure 1.

To our knowledge, limited research has combined deep learning with background subtraction or anomaly detection for obstacle detection in agriculture. In [30], a non-convolutional autoencoder has been used to dynamically reconstruct the background and detect foreground elements. In [31], the concept is similar to this work, as high level convolutional features are used in detecting foreground elements. The method is dependent on either human annotations or a simple background subtraction algorithm to initially generate training data. A critical drawback of both [30,31] for a tractor mounted camera is that they are developed for a static camera.

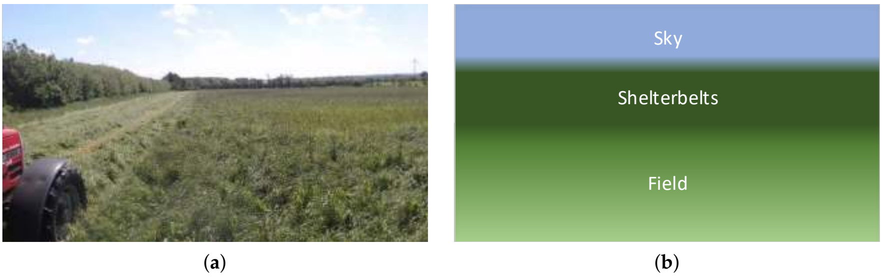


Figure 1. The visually homogenous characteristics of an agricultural field. (**a**) Shows agricultural field from tractor implement. (**b**) Illustration of the few visual components in an agricultural field.

More deep learning research has been dedicated to an area related to anomaly detection called visual saliency [32]. The critical point of the visual-saliency is that it will always find a salient element no matter the image content. In agriculture, obstacles occur rarely, and an image is expected to mostly not contain an anomaly.

An elaborated investigation on combining background subtraction and deep learning is performed. We define a top performing configuration as DeepAnomaly, an anomaly detector that exploits the homogenous characteristics of an agricultural field. As an object detector, it is fast and has high accuracy, compared to the state-of-the-art. DeepAnomaly is intended to assist and not replace CNN-based object detection and semantic segmentation algorithms, when objects are distant, very occluded or unknown. Another property when used in conjunction with other CNN-based methods is that DeepAnomaly will only add little computational cost, as it may use features from another CNN-based obstacle detector.

2. Materials

Images are recorded using a stereo camera composed of two Flea 3 GigE color cameras (Model: FL3-GE-28S4C-C, Point Grey Research Inc, Richmond, Canada) with a global shutter, a resolution of 1920 × 1080, a baseline of 24 cm and a frame rate of 15 Hz. The stereo camera is mounted on a sensor platform [33,34] roughly 2 m above the ground; see Figure 2. The algorithm uses images taken from the left camera, and the data from the right camera are only used for estimating the distance to obstacles when evaluating the proposed algorithm.

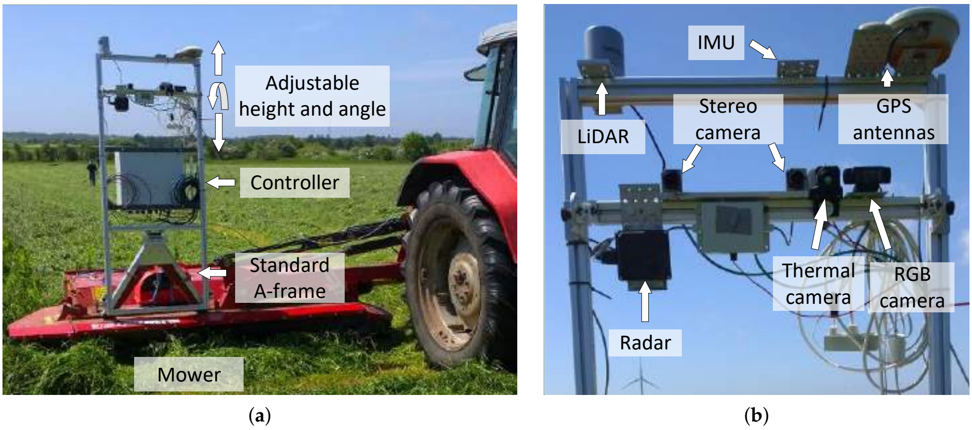


Figure 2. Sensor frame including the controller (**a**). Sensors on the sensor platform (**b**). Figure taken from [34].

Images were recorded during a grass mowing operation in a 7.5-ha grass field near Lem, Denmark, in June 2015 on a sunny and partly cloudy day. To simulate potential obstacles humans, green barrels [35], kid and adult mannequins were placed in the trajectory of the tractor. Image examples are presented in Figure 3. Two datasets are used in this work; background data for generating the background model and test data for evaluating anomaly detection configurations. The background data consist of 56 images recorded in a single crossing of the field from one end to the other. The test data are a selection of 48 images from 12 scenarios with per-pixel annotations of humans, houses, mannequins, barrels and wells. Each scenario contains 3–5 image samples taken in a range of 2–20 m to the obstacle as the tractor approaches. The obstacles and the tractor positions are estimated using GPS measurements and depicted in Figure 4. Other obstacles visible by the camera, such as humans watching the experiment, shelter belt and a house, are not depicted in the figure.

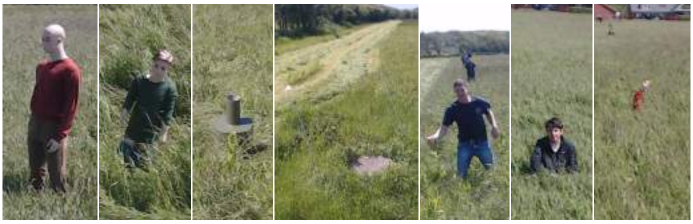


Figure 3. Two mannequins, a barrel, a well and three people, one only showing his arm, in the field.

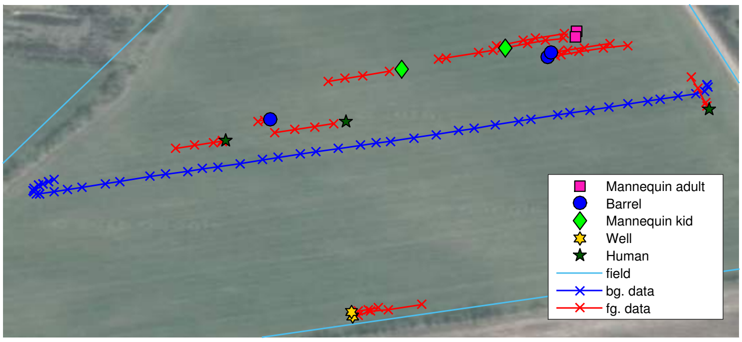


Figure 4. Obstacles and tractor image positions for background data (blue) and test data (red). Orthophoto from Google Maps.

3. Methods

The anomaly detection framework combines BS methods with high-level features, extracted from convolutional layers in a CNN. The high-level features make the BS robust to changes caused by camera motion and sensitive to new content or elements that are unnatural in an agricultural field.

3.1. CNN Features for Anomaly Detection

Figure 5 depicts how the anomaly detection framework uses features from a CNN. The Caffe reference [36] model, a variation of AlexNet [9], forward passes a fixed sized image through the network, generating intermediate features maps, depicted by cubes. The final feature map (6 × 6 × 256) is forwarded through multiple fully-connected neural networks (FC) and a softmax layer generating a prediction vector (1 × 1000). For a CNN trained on ImageNet, the prediction vector contains a value for each of the 1000 object types, which is standard in the ImageNet classification task [25]. A feature map describes the characteristics of the input image, where each channel corresponds to a specific feature. In the first convolutional layer, channels will activate on low level features, such as edges, blobs and colors. In deeper layers, channels will activate on high level features with more abstract characteristics, such as faces, text or vehicles [37].

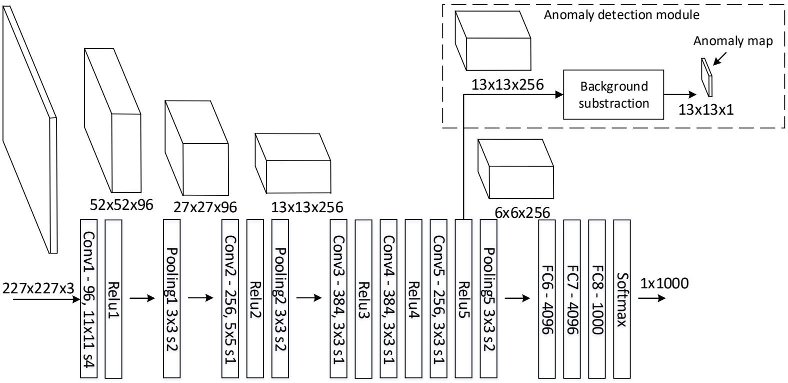


Figure 5. Caffe reference model [36] and the anomaly detection module. The notations of, e.g., "Conv1—96, 11 × 11 s4" mean that convolutional layer Conv1 has 96 kernels with a receptive field of 11 × 11 and a stride of four.

The figure illustrates how feature map dimensions decrease through the network going from an input image of 227 × 227 to a 6 × 6 feature map. Each entry in the 6 × 6 features map contains 256 high level features describing an area (receptive field) of 195 × 195 in the original image for every 32 pixels.

The anomaly detection module uses feature maps from a sequence of images to model the background. With a background model, it is possible to describe the distance from a feature map entry to the background model. As depicted in Figure 5, a feature map of 13 × 13 entries will generate an anomaly map of 13 × 13.

Many state-of-the-art deep learning-based detectors generate feature maps that can be used by an anomaly detection module. Thus, existing CNN based detector are able, to add the anomaly detection module for a small computational cost, to detect anomalies.

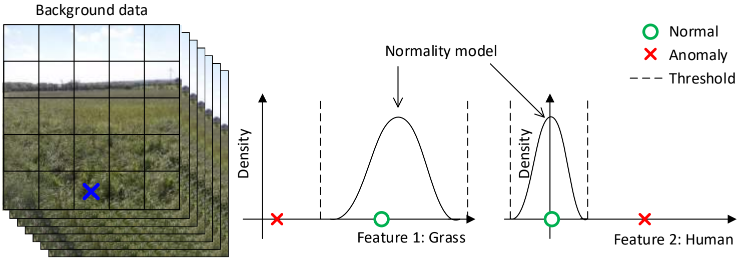


Figure 6. Representation of the background subtraction module for only a single feature map entry and two features.

Figure 6 illustrates the intuition behind the background subtraction module. At the left, feature maps are calculated for a sequence of images. The feature map for an image is represented with a grid, where each feature map entry describes an area in the original image with, e.g., 256 features for the Caffe reference model. For simplicity, only a single feature map entry (marked with a blue cross) and two features are used in this illustration. At the middle and right, the feature map entry is modeled with a Gaussian distribution for respectively a grass-like and a human-like feature and shows that the normality model generally expects high values for grass-like features and low values for human-like features. The normality model is then based on some threshold able to detect an outlier or anomaly.

3.1.1. Mapping of Feature Maps back to the Input Image

To determine what a feature map or an anomaly map entry corresponds to in the input image, the network stride, network receptive field and image boundary must be determined as illustrated in Figure 7. The network stride is the pixel spacing between network predictions. The network receptive field is the area a feature map entry describes in the original image. Valid convolutions will create an undefined area along the image border. In Figure 7, this is defined as the image boundary. The receptive field of a prediction may use the image boundary to provide some implicit description of the undefined area as illustrated in Figure 7. To compare anomaly detection with the ground truth, ground truth images are cropped by the image boundary, and the anomaly detection map is resized by the network stride using nearest neighbor interpolation.

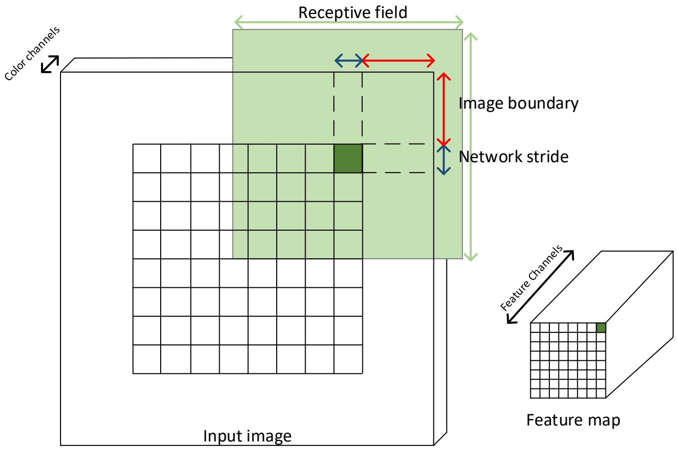


Figure 7. Mapping of a feature map entry (marked with dark green) back to the input image. A feature map entry describes an area of similar size as the network stride in the input image. A feature is determined by the information captured in the receptive field. Image boundary areas are not explicitly described by a feature map entry; only implicitly by the receptive field of nearby feature map entries.

3.1.2. Network Modifications

To target the network for anomaly detection, a few modifications of the Caffe reference CNN architecture is performed. The low 6 × 6 resolution of the final feature map provides poor spatial resolution in the resulting anomaly map. The nature of convolutional and subsampling layers allows larger (in height and width) images to be forwarded through the network and generates higher resolution feature maps. However, unlike convolutional and max-pooling layers, the FC and softmax layers require a fixed sized image. By removing the softmax, the three FC and the final max-pooling layer, the network is able to double the feature map resolution and process larger images. Additional advantages of removing the final layers are a faster forward time and a much lower memory footprint, that is critical for an embedded GPU with limited memory and computation power. For VGG16 on a Titan X GPU, the forward pass drops 39.5% from 20.5 ms to 12.4 ms, and the memory footprint drops 74.6% from 1485 MB to 376 MB. For the Caffe reference model, the forward pass drops 36.8% from 3.75 ms to 2.37 ms, and the memory footprint drops 78.9% from 303 MB to 64 MB.

An unwanted effect of zero-padded convolutions is that feature maps get corrupted or become invalid along the image border. In image classification, this is not critical, as the object of interest is placed in the image center. In anomaly detection, features are required to be valid in all image positions. This is handled by only using valid convolutions or no zero-padding for all layers in a network. Changing between valid and invalid convolutions will, for a network with no FC-layers, not require a network to be retrained.

3.1.3. Network Feature Map Investigation

A range of feature configurations are tested.

1. Use features from both the Caffe reference model (AlexNet) and the VGG architecture.
2. Use features from different layers in a network. Earlier layers are more general [38,39], require less computation and provide higher feature map resolution.
3. Use features before the activation function ReLU (Rectified Linear Unit). A Gaussian distribution will more accurately resemble the output of a convolutional layer before the ReLU, as depicted in Figure 8.

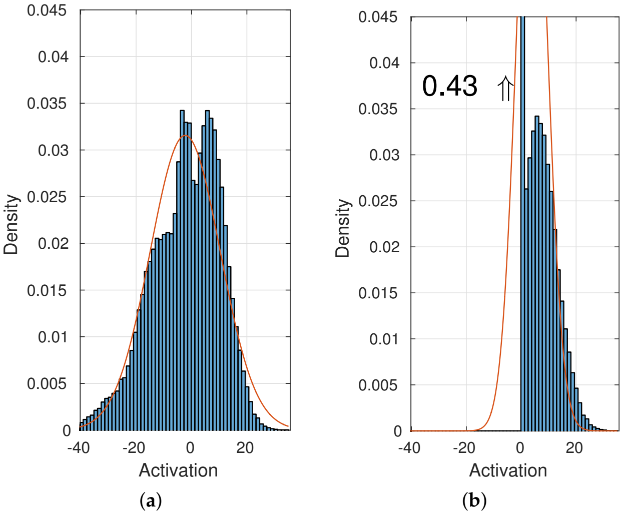


Figure 8. Histogram of a neural network unit before (**a**) and after (**b**) a ReLU (Rectified Linear Unit).

4. Use dilated convolutions as described in [40] to double the feature map resolution for a given input image without doubling the input image size. The feature map is increased by removing max-pooling layers and doubling the dilation factor in subsequent layers.
5. Append an 1 × 1 convolutional layer before the final max-pooling layer to perform feature compression or dimension reduction as in GoogLeNet [11]. The high number of features provided by a CNN, e.g., 256 by AlexNet in the final convolutional layer, makes the computational complexity of the background model high. To avoid retraining a network from scratch with fewer features, a 1 × 1 convolutional layer is appended to an ImageNet pre-trained network. Three network architectures are created with respectively 128, 64 and 32 kernels for the appended 1 × 1 layer and fine-tuned on ImageNet.

3.2. Image Model Geometry

In conventional background subtraction algorithms [41], each pixel is classified as either foreground or background using a model of the background. As illustrated in Figure 6, a normality model is typically generated for each feature map entry over a sequence of images using only features from the same image position. For a front-facing camera in an agricultural field, the image is expected to have a specific geometry as depicted in Figure 1. We investigate various image modeling geometries:

1. Single model: Models the whole image using a single model. The model uses all feature map entries in the image over a sequence of images. As illustrated in Figure 9a, a single model is generated for a 4 × 5 feature map.
2. Row model: Models each row in the image. Each model uses feature map entries from the current row over a sequence of images. As illustrated in Figure 9b, four models are generated for a 4 × 5 feature map.
3. Extended row model: Models each row in the image. Each model uses feature map entries from the current and neighboring rows over a sequence of images. As illustrated in Figure 9c, four models are generated for a 4 × 5 feature map.
4. Traditional BS model: Models each entry in a feature map. Each model uses only the current feature map entry over a sequence of images as the traditional background subtraction algorithm [41]. As illustrated in Figure 9d, 20 models are generated for a 4 × 5 feature map.

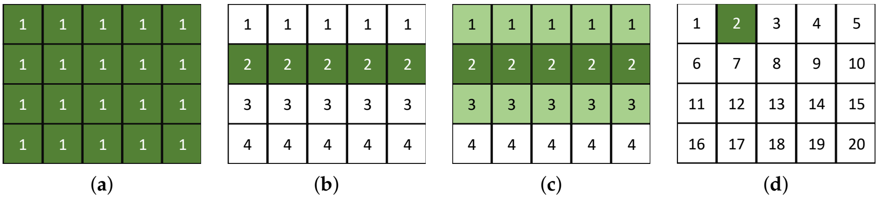


Figure 9. Image model geometries. (**a**) Geometry 1; (**b**) Geometry 2; (**c**) Geometry 3; (**d**) Geometry 4.

3.3. Normality Model Types

An outlier detector uses a background model or a normality model to model the background data. Outliers are defined as a sample outside the normal area. The normal area is defined by the normality model, background data and a threshold; see Figure 10 for two-dimensional examples.

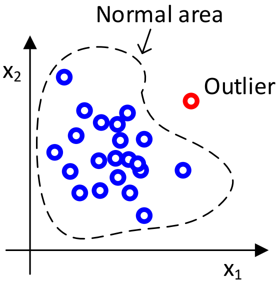


Figure 10. Outlier detection.

A feature sample is defined as an entry from a feature map with D features. A normality model is generated from N feature samples. We denote background feature samples for generating a normality model by $\mathbf{X}$; hence, $\mathbf{X}$ is a $D \times N$ matrix with N samples with D features. The column j of $\mathbf{X}$, defined as $\mathbf{x_j}$, is all of the features for a sample j.

Feature samples are gathered over a sequence of images. However, depending on the model geometry, background samples are gathered from either the whole image, feature map rows or only specific feature map entries. In a conventional background subtraction, a "traditional BS model", a model uses only samples from a specific image position over a sequence of images. In image model Geometry 1, a model uses all samples in all positions in a sequence of images. The origin of feature samples, in terms of the feature map entry and image, are ignored in the following section and simply denoted by $\mathbf{X}$.

The aim of an outlier detector is to determine an anomaly measure, denoted by $\mathcal{M}(\tilde{\mathbf{x}} \mid \boldsymbol{\theta})$. $\mathcal{M}(\tilde{\mathbf{x}} \mid \boldsymbol{\theta})$ measures the distance between a sample, $\tilde{\mathbf{x}}$, with an unknown class and the normality model with parameters $\boldsymbol{\theta}$. The model parameters are dependent on the normality model type and background samples. When $\mathcal{M}(\tilde{\mathbf{x}} \mid \boldsymbol{\theta})$ is larger than a specified threshold, $\tilde{\mathbf{x}}$ is classified as an outlier/abnormality. The threshold value is selected based on the annotated test data. We address this issue in the result sections.

3.3.1. Mean and Median

The mean parameters $\boldsymbol{\theta}_{mean} = \boldsymbol{\mu}$ are the mean value of each feature i over all background samples $\boldsymbol{\mu} = [\mu_1 \dots \mu_i \dots \mu_D]^T$. For a sample, $\tilde{\mathbf{x}}$, the anomaly measure is defined as the Euclidean distance between the mean value of a feature to a sample feature.

$$\mathcal{M}_{\text{mean}}(\tilde{\mathbf{x}} \mid \boldsymbol{\mu}) = \|\boldsymbol{\mu} - \tilde{\mathbf{x}}\| = \sqrt{\sum_{i=1}^{D} (\mu_i - \tilde{x}_i)^2} \tag{1}$$

For a median model, the median value $\mathbf{m}$ is used instead of the mean value.

$$\mathcal{M}_{\text{median}}(\tilde{\mathbf{x}} \mid \mathbf{m}) = \|\mathbf{m} - \tilde{\mathbf{x}}\| \tag{2}$$

3.3.2. k-NN

The kNN model [4] parameters $\boldsymbol{\theta}_{\text{kNN}} = \mathbf{X}$ consist of all background samples, $\mathbf{X}$. The anomaly measure is the Euclidean distance from the k-nearest neighbor sample $\mathbf{x}_{\text{kNN}}$ to a sample $\tilde{\mathbf{x}}$.

$$\mathcal{M}_{\text{kNN}}(\tilde{\mathbf{x}} \mid \mathbf{X}) = \|\mathbf{x}_{\text{kNN}} - \tilde{\mathbf{x}}\| \tag{3}$$

3.3.3. Single Variate Gaussian

The single variate Gaussian (SVG) [41] model parameters $\boldsymbol{\theta}_{\text{SVG}} = \left(\boldsymbol{\mu}, \boldsymbol{\sigma^2}\right)$ comprise the mean value, $\boldsymbol{\mu}$, and the variation for each feature i taken over all samples in the training data $\boldsymbol{\sigma}^2 = [\sigma_1^2 \dots \sigma_i^2 \dots \sigma_D^2]^T$. The anomaly measure is defined as the Mahalanobis distance between a feature sample and the single variate Gaussian distribution along each dimension.

$$\mathcal{M}_{\text{SVG}}\left(\tilde{\mathbf{x}} \mid \boldsymbol{\mu}, \boldsymbol{\sigma}^2\right) = \left\|(\tilde{\mathbf{x}} - \boldsymbol{\mu}) \odot \frac{1}{\boldsymbol{\sigma}}\right\| = \sqrt{\sum_{i=1}^{D} \frac{(\tilde{x}_i - \mu_i)^2}{\sigma_i^2}} \tag{4}$$

Unlike the multivariate Gaussian model described in the next sections, feature dimensions are treated independently (Σ is a diagonal matrix).

3.3.4. Multivariate Gaussian

The multivariate Gaussian (MVG) [41] parameters $\boldsymbol{\theta}_{\text{MVG}} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ comprise the mean value, $\boldsymbol{\mu}$, and the covariance matrix, $\boldsymbol{\Sigma}$.

$$\Sigma = \frac{1}{N-1} \sum_{j=1}^{N} \left(\mathbf{x_j} - \boldsymbol{\mu}\right) \left(\mathbf{x_j} - \boldsymbol{\mu}\right)^T \tag{5}$$

The anomaly measure is defined as the Mahalanobis distance between a sample $\tilde{\mathbf{x}}$ and the Gaussian distribution.

$$\mathcal{M}_{\mathrm{MVG}}\left(\tilde{\mathbf{x}} \mid \boldsymbol{\mu}, \sigma^2\right) = \sqrt{\left(\tilde{\mathbf{x}} - \boldsymbol{\mu}\right)^T \Sigma^{-1} \left(\tilde{\mathbf{x}} - \boldsymbol{\mu}\right)} \tag{6}$$

3.3.5. Gaussian Mixture Model

The Gaussian mixture model (GMM) [3] parameters $\boldsymbol{\theta}_{\mathrm{GMM}} = (\boldsymbol{\mu}_1 \dots \boldsymbol{\mu}_M, \Sigma_1 \dots \Sigma_M)$ comprise **M** Gaussian models with a mean value, $\boldsymbol{\mu}_i$, and the covariance matrix, Σ_i, for each model i. Models are determined using expectation-maximization [42]. The anomaly measure is defined as the Mahalanobis distance between a sample and the nearest neighbor Gaussian model, $\boldsymbol{\theta}_{\mathrm{NN}} = (\boldsymbol{\mu}_{\mathrm{NN}}, \Sigma_{\mathrm{NN}})$.

$$\mathcal{M}_{\mathrm{MVG}}\left(\tilde{\mathbf{x}} \mid \boldsymbol{\mu}_{\mathrm{NN}}, \Sigma_{\mathrm{NN}}\right) = \sqrt{\left(\tilde{\mathbf{x}} - \boldsymbol{\mu}_{\mathrm{NN}}\right)^T \Sigma_{NN}^{-1} \left(\tilde{\mathbf{x}} - \boldsymbol{\mu}_{\mathrm{NN}}\right)} \tag{7}$$

3.4. Implementation Details

CNN models are executed using Caffe, a framework for deep learning [36] and the Caffe-MATLAB interface allowing MATLAB to use CNN features from network layers. The background models and the evaluation of various configurations are implemented in MATLAB. The original images with a resolution of 1080 × 1920 are cropped by 700 pixels to remove the tractor from the left side of the image. Images are resized by a factor of 0.75 and cropped slightly again to form valid dimensions for a CNN network.

4. Results

Results are divided into three subsections. The first subsection shows trends across many network configurations. A specific configuration is not optimal across any image geometric modeling, normality model or output layer, e.g., the optimal normality model type depends on the network layer and the geometric modeling. The first sections are intended to show configuration trends across the vast number of configurations. The second subsection is targeted directly at reaching the most optimal configuration in terms of accuracy and computation time. The third subsection compares a top performing configuration with state-of-the-art object detection algorithms.

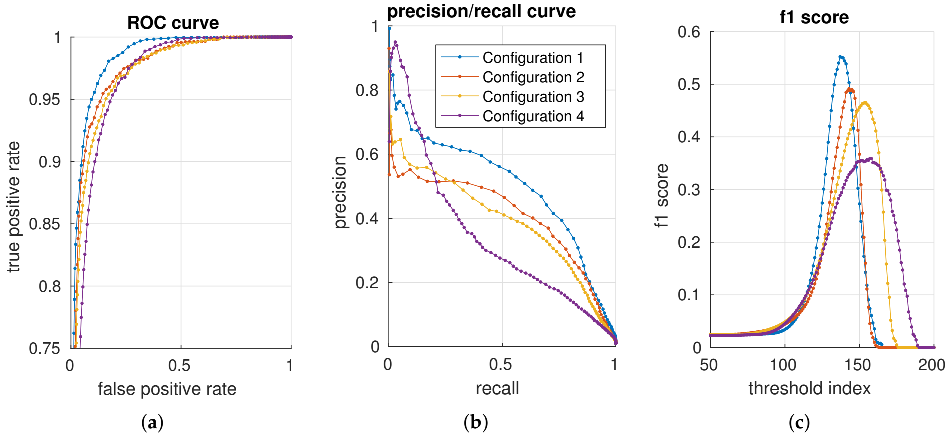


Figure 11. Illustration of an ROC curve (**a**), a precision/recall curve (**b**) and f1 scores (**c**) for four configurations.

We use the background data to create a normality model for all configurations. Each configuration is evaluated against the per-pixel annotated test data. In total, 460 configurations are evaluated. For each configuration, a receiver operating characteristic (ROC) [43] curve, precision/recall (PR) [44] curve and the f1 score [45] are generated with 200 distributed thresholds. Four configurations are presented in Figure 11 using, respectively, an ROC, precision/recall and an f1 score curve. The advantage of precision/recall and the f1 score is that true negatives are not included in the metric. As the data mostly contain negative samples (field samples), the (zoomed) ROC plot shows curves that are squeezed together.

To get a single valued accuracy measure, for each configuration and metric, the maximum f1 score, area under the curve for an ROC (AUC_ROC) curve and the area under the curve for a precision/recall (AUC_PR) curve are used in the following sections.

4.1. Trends Across Configurations

This sections provides an overview of the 460 configurations by using the maximum f1 score and boxplot presentations. In Figure 12, a boxplot presents the accuracy variation (maximum f1 score) for all configurations using a specific output layer. There are, e.g., 56 different configurations for the CaffeRef—Relu5 output layer. Generally, the accuracy degrades for lower feature layers; dilated layers are not preferable, and ReLU layers are mostly preferred over convolutional (Conv) layers. Generally, feature compression using 1 × 1 convolutional layers (Relu6_X and Conv6_X) does not significantly reduce performance. Compression of features will reduce the processing time for an anomaly module and will for some classifiers also improve accuracy.

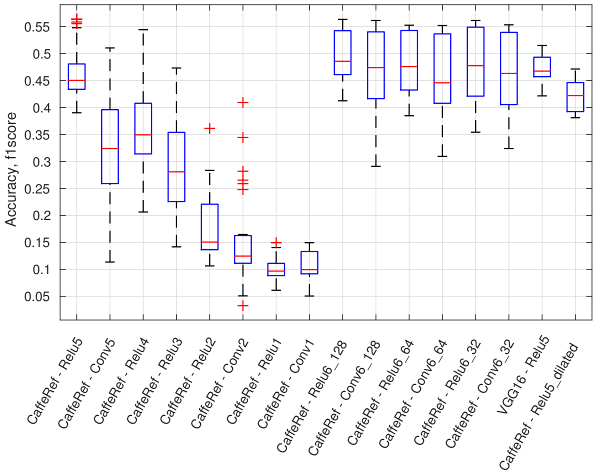
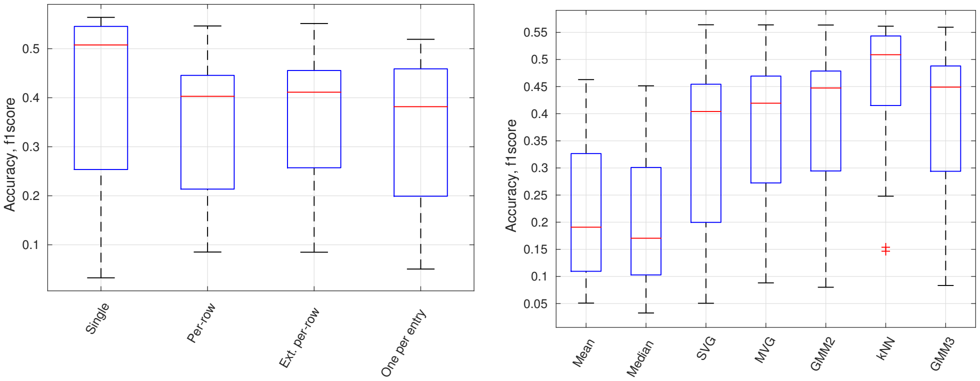


Figure 12. Accuracy (f1 score) for a given output layer across all model normality types and image model geometries.

In Figure 13a, a boxplot presents the accuracy variation for all configurations using a specific image model geometry. It shows that a single model or the model per-row performance is better than the traditional background subtraction geometry with one model per entry. In Figure 13b, a boxplot presents the accuracy variation for all configurations using a specific normality model type. GMM 2 and GMM 3 is a GMM model with respectively 2 (**M** = **2**) and 3 (**M** = **3**) Gaussian models. The kNN-based model is of highest performance followed by the Gaussian-based models. Mean and median models are inferior to other model types.

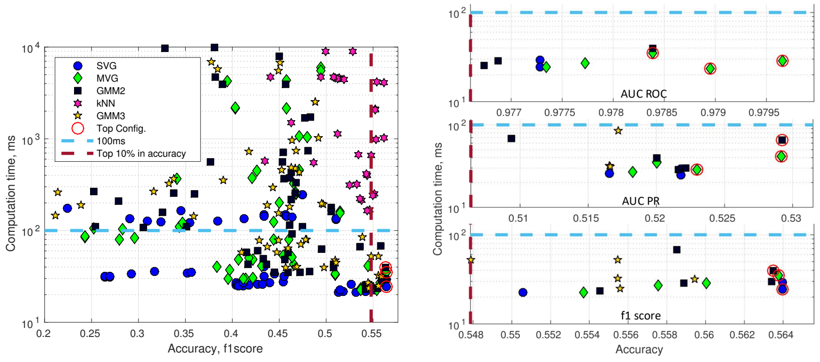


(**a**) (**b**)

Figure 13. Accuracy for model geometries (**a**) and normality models (**b**). (**a**) Accuracy (f1 score) for all image model geometries; (**b**) accuracy (f1 score) for all normality model types.

4.2. Determining the Best Set of Configuration

The optimal configuration is a combination of high accuracy and speed performance, e.g., the kNN classifier generally has a high accuracy performance. However, even the fastest kNN configuration has a computation time of more than 200 ms for a single image. Figure 14a shows the computation time for a single image versus the f1 score accuracy performance. Two lines draw a top configuration rectangle with the fastest (<100 ms) and the highest accuracy (top 10%) anomaly detectors. Figure 14b (a top performing SVG in the bottom plot of Figure 14b is ignored as the SVG just below has identical accuracy) presents the top configuration rectangle for respectively AUC_ROC, AUC_PR and the maximum f1 score. Feature calculations are performed on a GTX Titan X 12GB Maxwell architecture, and the anomaly detection module is executed on a Intel Xeon 2.1 GHz six-core CPU (E5-2620V2). The three configurations with the highest accuracy of each plot are marked with a red circle.



(**a**) (**b**)

Figure 14. Accuracy and computation time. (**a**) Accuracy is measured using the f1 score; (**b**) top configurations rectangle of AUC_ROC, AUC_PR and the max f1 score.

The seven unique top performing configurations have been listed in Table 1; two of the nine configurations have a duplicate. The computation time is listed for a prediction of a single image including feature calculations (Total Pred.), a prediction of a single image without feature calculations (Model Pred.) and for updating the background model (Model Update). The model update is not expected to be performed for every image and is therefore not considered as time critical as the total prediction time for an image. The anomaly detection module can use features from another CNN-based detector and avoid the computational cost of computing its own features. The model prediction time is listed to show the computation cost of adding anomaly detection to an existing CNN-based detector. Apart from Number 2, the top configurations are very similar in accuracy. Generally, the table includes only Gaussian-based normality models and single model-based image model geometry. The slightly faster and simpler Configuration 6 is in this paper defined as DeepAnomaly and used in future experiments. DeepAnomaly has a total prediction time of 25 ms (40 FPS), a model prediction time of only 4 ms and a model update time of 834 ms. The performance of DeepAnomaly is presented in a set of image examples in Figure 15. The pixel accuracy on the annotated test data is used for selecting a threshold.

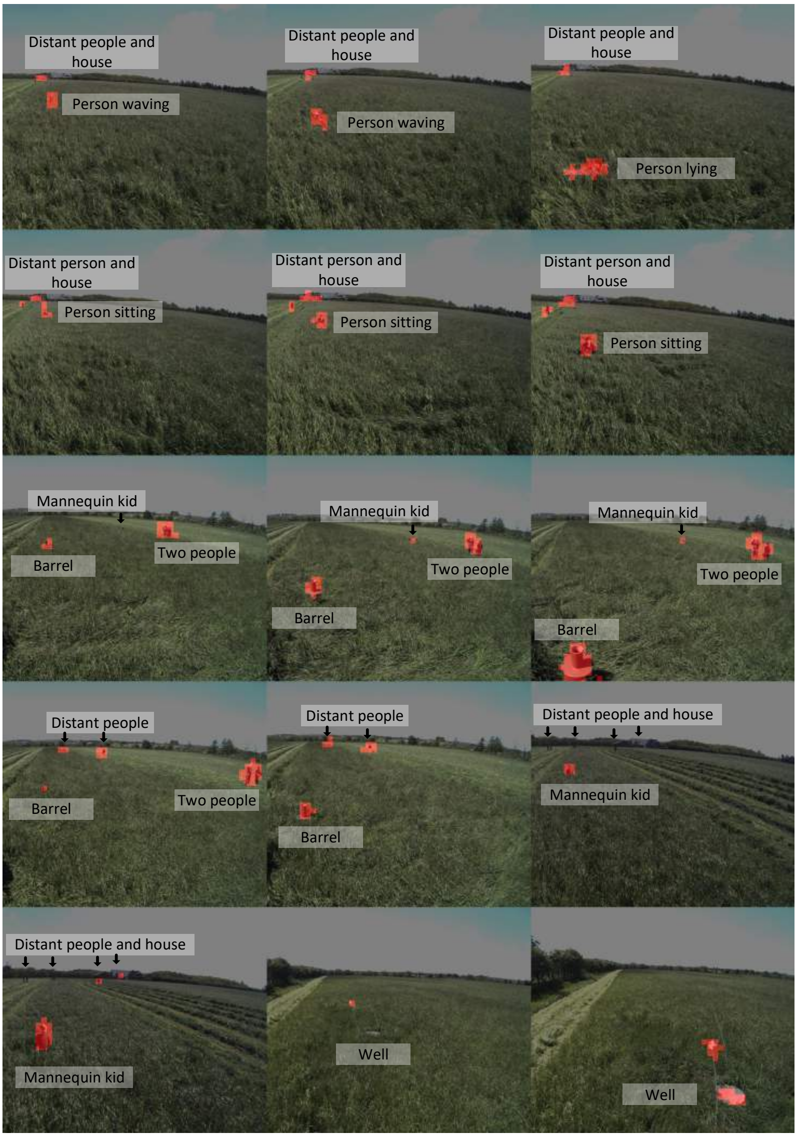


Figure 15. DeepAnomaly detections. No false positives are present in the images.

Table 1. Shows seven top performing configurations in terms of computation time and three accuracy measures.

Nr	Classifier	Metric	Model Area	Layer Output	ROC AUC	PR AUC	F1 Score	Computation Time (ms)		
								Total Pred.	Model Pred.	Model Update
1	MVG	ROC/PR	Single	Conv6_128	0.980	0.523	0.560	29	8	1,219
2	MVG	ROC	Single	Conv6_64	0.979	0.476	0.537	23	4	438
3	MVG	ROC/f1	Single	Relu5	0.978	0.520	0.564	35	14	6,360
4	MVG	PR	Single	Relu4	0.970	0.529	0.544	42	21	12,169
5	GMM2	PR	Single	Relu4	0.969	0.529	0.536	66	42	142,727
6	SVG	f1	Single	Relu5	0.977	0.522	0.564	25	4	834
7	GMM2	f1	Single	Relu5	0.978	0.520	0.564	40	19	5,325

4.3. Object Detection vs. Anomaly Detection

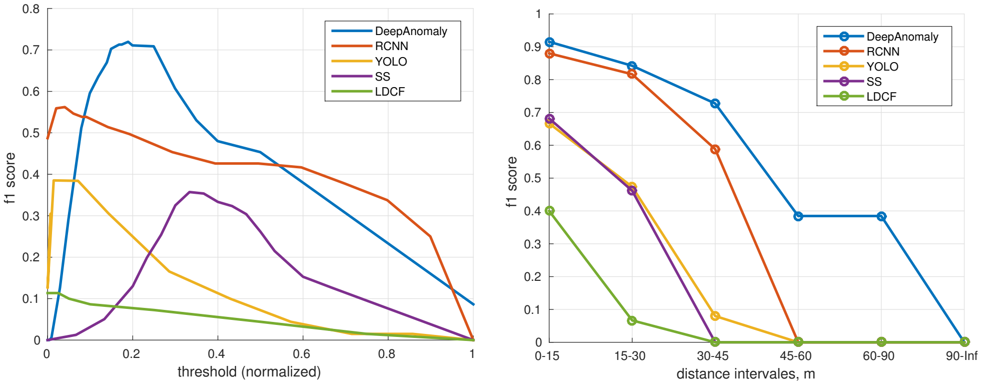
The accuracy of anomaly configurations has been reported using the f1 score and AUC measures, allowing anomaly configurations to be compared mutually. However, such accuracy measures provide only an indication of the performance of DeepAnomaly compared to other state-of-the-art detectors. In this section, a quantitative evaluation metric is defined to evaluate DeepAnomaly with state-of-the-art detection algorithms.

The comparison is challenged by the inconsistent outputs of the selected algorithms. Algorithms may either detect one or multiple object types, and the location of objects are marked with either a bounding box or per-pixel predictions. To solve this inconsistency problem, a detector is only evaluated for its ability to detect them (other annotated obstacles are ignored). Humans are used as test objects, as all algorithms are able to detect humans. Ground truth annotations, which are per-pixel annotations of obstacles, are converted to bounding boxes and extended by 12 pixels on all sides. A detection is true (true positive) when the detection area overlaps an annotated human by more than 50%. An overlap of less than 50% is a false detection (false positive), unless the detection overlaps an annotated non-human obstacle by more than 50%. A false negative is defined as a human annotation that has not been detected.

Four object detection algorithms have been selected: a pedestrian detector “local decorrelated channel features” [46–48] (LDCF) trained on INRIA (Institut national de recherche en informatique et en automatique) Person Dataset ; two deep learning multi object detection algorithms “you only look once” [18] (YOLO) and “faster R-CNN” [16] (RCNN), trained on ImageNet and Pascal VOC (Visual Object Classes)[24]; one semantic segmentation algorithm “fully convolutional neural networks for semantic segmentation” [19,49] (SS) trained on ImageNet and Pascal Context [50]. Figure 16a shows the f1 score for each algorithm sweeping over a set of thresholds. The highest f1 score is achieved by DeepAnomaly (0.720) followed by RCNN (0.562), YOLO (0.385), SS (0.357) and LDCF (0.113).

The normality model of DeepAnomaly is sensitive to the used background samples; meaning that the optimal threshold may change for each model update. However, DeepAnomaly forms a little plateau of top accuracies in the range of 150–250 (0.15–0.25 normalized), showing that the threshold is partly robust to model updates.

Figure 16b shows the f1-score at different distances using the optimal threshold for each algorithm. DeepAnomaly is able to detect humans at longer distances and is either of similar or better performance on short ranges using a smaller CNN model than RCNN, YOLO and SS. Table 2 compares the algorithms’ accuracy, computation time, the number of model parameters (# Model Params) and its ability to classify obstacles (Class.) and detect unknown obstacles (Unk. Types). (The number of model parameters of RCNN, YOLO and LDCF is not directly specified in the respective publications. The number for RCNN only includes parameters of convolutional layers (regression and classification modules parameters are ignored). The number for YOLO is based on the described model. The number for LDCF is a rough estimate.). DeepAnomaly detects humans with better accuracy at longer distances in real time. RCNN shows similar performance on shorter distances. However, the computation time of RCNN is unsuited for real-time applications. A qualitative test is presented in Appendix A, showing detections from all algorithms in 30 images containing humans. A key feature of DeepAnomaly is the ability to detect unknown objects/anomalies and not just a set of predefined objects. Secondly, DeepAnomaly uses only a pre-trained network and does not require the time-consuming task of making algorithm/object-specific training data. The high accuracy, the low number of model parameters and the low compute time make DeepAnomaly suited for a real-time detection system on an embedded GPU. The drawback of DeepAnomaly is that no label or classification is provided for each detection.



(**a**) (**b**)

Figure 16. Accuracy performance (f1 score) of DeepAnomaly and four state-of-the-art object detection algorithms. (**a**) Accuracy relative to thresholds; (**b**) accuracy relative to distance intervals.

Table 2. Comparison of of DeepAnomaly with four state-of-the-art algorithms for obstacle detection. YOLO, you only look once; SS, semantic segmentation; LDCF, local decorrelated channel features.

Name	F1 Score	Compute Time (ms)	Compute Unit	Class.	Range	Unk. Types	Object Specific Training Data	# Model Params.
DeepAnomaly	0.720	25	GPU + CPU	No	Far	Yes	No	3.7 M
RCNN	0.562	182	GPU	Yes	Mid	No	Yes	14.7 M
YOLO	0.385	23	GPU	Yes	Low	No	Yes	262 M
SS	0.357	237	GPU	Yes	Low	Yes	Yes	134 M
LDCF	0.113	348	CPU	Yes	Low	No	Yes	<0.01 M

5. Discussion

Deep learning-based object detection and semantic segmentation have recently showed state-of-the-art results in detecting specific objects. However, in an agricultural context, they have difficulty in detecting heavily occluded and distant objects, and methods are, by definition, trained to recognize a predefined set of object types. DeepAnomaly can exploit the very homogeneous characteristics of an agricultural field to detect distant, heavy occluded and unknown objects. Qualitatively, this is illustrated in Figure 15, where DeepAnomaly detects a distant and occluded mannequin kid, a human showing only his arm, a heavy occluded olive-green barrel (with similar color as the field), a well cover and detections of obstacles with a size of less than 16 × 16 pixels. By using DeepAnomaly in junction with other deep learning algorithms, it can save computations by using convolutional features from other networks. DeepAnomaly also spares the time-consuming task of providing domain- or algorithm-specific annotated data.

A detection metric for detecting humans is defined to compare DeepAnomaly with four state-of-the-art algorithms. The comparison shows that DeepAnomaly is better at detecting humans at longer ranges (45–90 m). RCNN has similar performance at short range (0–30 m). However, with much fewer model parameters and a (182 ms/25 ms=) 7.28-times faster processing time per image, DeepAnomaly is more suitable for real-time applications running on an embedded GPU. The used detection metric copes with dissimilar outputs of the evaluated algorithm and will not favor a precise localization/position of a detection. However, in the context of autonomous vehicles in agriculture, the exact bounding box position or semantic segmentation at pixel-level precision is not of critical importance. Rough localization markings (±12 pixel) are sufficient, and more important is the detector's ability to, in real time, detect obstacles even when they are heavily occluded, distant and potentially unknown. However, it is important to state that DeepAnomaly requires specific conditions in terms of visually-homogenous surrounding and a low incidence of anomalies. This is not a limitation for YOLO, RCNN, SS and LDCF.

Evaluating algorithms that are trained on different data is basically unfair; especially for LDCF, which uses a much smaller dataset. However, YOLO, RCNN, SS and DeepAnomaly use parameters from a network trained on ImageNet, including one additional dataset with algorithm-specific annotations (bounding boxes, per-pixel annotations, background data). One may argue that DeepAnomaly has an (unfair) advantage, as it learns a background model from data recorded in the same field as the test. This is true for a classification problem where test and training must be uncorrelated. However, for background subtraction algorithms, it is a basic concept that an algorithm learns the characteristics of a specific setting to better detect foreground elements. Similar for DeepAnomaly, the algorithm is intended to exploit and learn the characteristics of a particular field to better detect anomalies.

DeepAnomaly does not provide labels as an obstacle detection algorithm. However, by using DeepAnomaly as a region proposal algorithm, labels can be given by forwarding anomalies through a classification network. This is related to RCNN and other deep learning object detection algorithms [14–17] that initially use a region proposal algorithm providing between 300 and 2000 regions per images. Each region is then forwarded through a classification network providing a label for each region. DeepAnomaly can be used as an effective region proposal algorithm providing only a few or no regions per image.

The normality model must be updated regularly without including foreground elements. This difficulty is partly solved in an agricultural context where the incidence of anomalies is very low. Secondly, the experiment shows that the initial model generalizes to many positions in the field, meaning that the model does not require very frequent updates. Furthermore, foreground elements can be filtered out by other obstacle detections algorithms. For, e.g., RCNN, the anomaly algorithm should not include feature map entries in the background model that is inside an RCNN bounding box detection. In future work, we are interested in extending the anomaly framework to other sensor modalities. Depth sensors are able to detect obstacles that protrude from the crop surface, and a thermal sensor can detect outlier heat radiations in the field. The advantage of combining visual, depth and thermal modalities is that anomalies are more independent and described by physically different characteristics, making it unlikely for foreground/non-field obstacles to be included in the background model (unless the foreground element has similar visual appearance, height and temperature as the crop).

The robustness of threshold values and procedures for doing model updates are addressed, but not implemented in actual experiments. This paper is focused on practical considerations for using deep learning features and elaborated investigations. The investigation comprises a total of 460 settings that are evaluated in terms of processing time and accuracy, using three different accuracy metrics. A top performing configuration is then compared to state-of-the-art detection algorithms for their ability to detect humans in general and at different range intervals.

6. Conclusions

This work illustrates that a background subtraction algorithm can be used successfully for a non-static camera in agriculture by using high level features from a deep convolutional neural network.

An elaborated investigation has been conducted on a broad set of configuration to determine a high performing setting for an anomaly detection system. This configuration is named DeepAnomaly. It is a simple algorithm that exploits the homogenous characteristics of an agricultural field, i.e., detect heavily occluded, distant and unknown objects without the time-consuming task of providing algorithm- and object-specific training data. DeepAnomaly is foremost an anomaly detector. However, it has shown comparable or better results for obstacle detection in an agricultural context. It is able to detect humans better and at longer distances than state-of-art networks with 40 FPS using less training data and a smaller network. The low computation time and low memory footprint make it suited as a real-time system and for embedded GPUs. DeepAnomaly is also able to assist an existing deep learning detection system by using the existing feature maps. Thus, for only a small computational cost, 4 ms on a CPU, a CNN-based detector can be extended to also detect distant, heavily occluded and unknown obstacles.

Acknowledgments: This research is sponsored by the Innovation Fund Denmark as part of the project "SAFE—Safer Autonomous Farming Equipment" (Project No. 16-2014-0).

Author Contributions: Rasmus N. Jørgensen, Kim A. Steen and Peter Christiansen jointly conceived of, designed and performed the experiments. Lars N. Nielsen and Peter Christiansen implemented the algorithm and the evaluation of algorithm performance. Henrik Karstoft and Peter Christiansen invented the basic concept of the algorithm and concept extensions.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Figures A1 and A2 show detections by DeepAnomaly, RCNN, SS, YOLO and LDCF for 30 image samples containing humans. Each detector is presented with a specific color. Detections by RCNN, YOLO and LDCF are presented with a bounding box. Detections by SS and DeepAnomaly are presented with respectively green and red coloring. Overlapping detections by SS and DeepAnomaly become yellow.

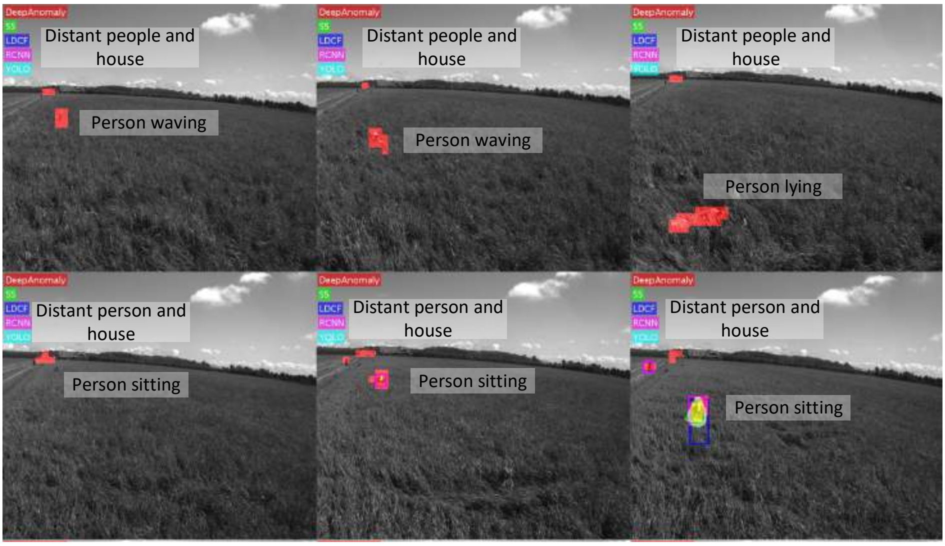


Figure A1. *Cont.*

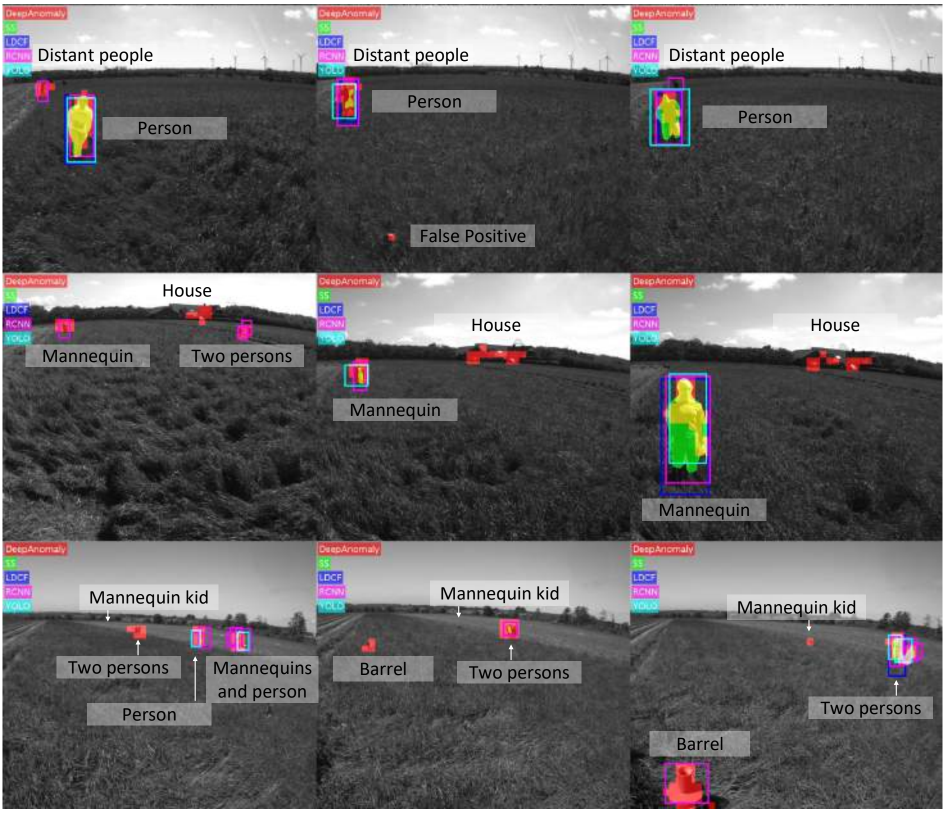


Figure A1. Detection examples of 15 image with humans.

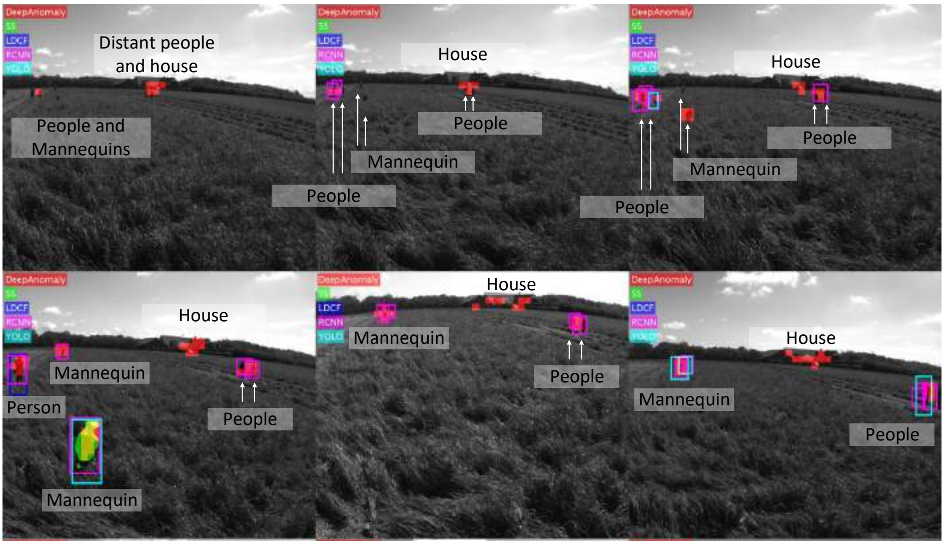


Figure A2. *Cont.*

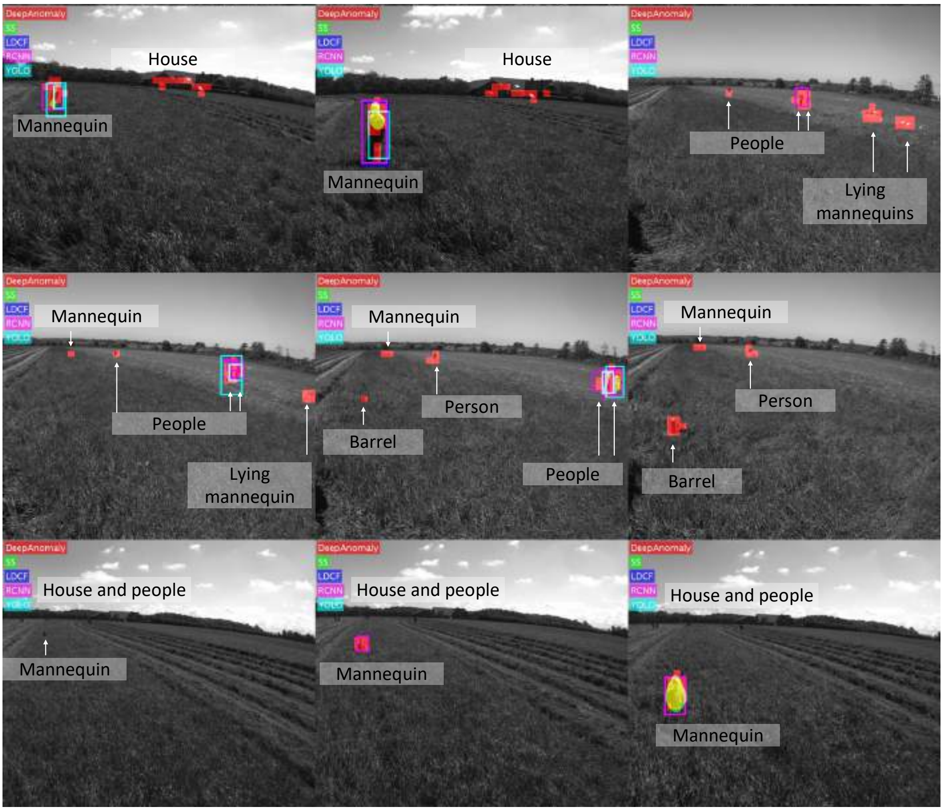


Figure A2. Detection examples of 15 image with humans.

References

1. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv. (CSUR)* **2009**, *41*, 1–58.
2. McLachlan, G.J.; Basford, K.E. Mixture models. Inference and applications to clustering. In *Statistics: Textbooks and Monographs*; Dekker: New York, NY, USA, 1988.
3. Stauffer, C.; Grimson, W.E.L. Adaptive background mixture models for real-time tracking. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Fort Collins, CO, USA, 23–25 June 1999; p. 252.
4. Zivkovic, Z.; van der Heijden, F. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognit. Lett.* **2006**, *27*, 773–780.
5. Bouwmans, T.; Porikli, F.; Höferlin, B.; Vacavant, A. *Background Modeling and Foreground Detection for Video Surveillance*; CRC Press: Boca Raton, FL, USA, 2014.
6. Kragh, M.; Jørgensen, R.N.; Henrik, P. Object Detection and Terrain Classification in Agricultural Fields Using 3D Lidar Data. In Proceedings of the International Conference on Computer Vision Systems, Copenhagen, Denmark, 6–9 July 2015; pp. 188–197.
7. Kragh, M.; Christiansen, P.; Korthals, T.; Jungeblut, T.; Karstoft, H.; Jørgensen, R.N. Multi-Modal Obstacle Detection and Evaluation of Occupancy Grid Mapping in Agriculture. In Proceedings of the International Conference on Agricultural Engineering, Aarhus, Denmark, 26–29 June 2016.
8. Oksanen, T. Laser scanner based collision prevention system for autonomous agricultural tractor. *Agron. Res.* **2015**, *13*, 167–172.

9. Krizhevsky, A.; Sutskever, I.; Hinton, G. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
10. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:cs.CV/1409.1556.
11. Szegedy, C.; Reed, S.; Sermanet, P.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–12.
12. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324.
13. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:cs.CV/1512.03385.
14. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014.
15. Girshick, R.; Donahue, J.; Darrell, T.; Berkeley, U.C.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Beijing, China, 23–28 June 2014; pp. 2–9.
16. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2015**, arXiv:cs.CV/1506.01497.
17. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
18. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2015**, arXiv:1506.02640.
19. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
20. Torr, P.H.S. Conditional Random Fields as Recurrent Neural Networks. In Proceedings of the IEEE International Conference on Computer Vision, Columbus, OH, USA, 24–27 June 2014.
21. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. *arXiv* **2014**, arXiv:1412.7062.
22. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *arXiv* **2016**, arXiv:cs.CV/1606.00915.
23. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 1–15.
24. Everingham, M.; Eslami, S.; Gool, L.V. The Pascal Visual Object Classes Challenge—A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136.
25. Berg, A.; Deng, J. Imagenet large scale visual recognition challenge. *I*nt. J. Comput. Vis. **2015**, *115*, 211–252.
26. Ross, P.; English, A.; Ball, D.; Upcroft, B.; Wyeth, G.; Corke, P. Novelty-based visual obstacle detection in agriculture. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 1699–1705.
27. Ross, P.; English, A.; Ball, D.; Upcroft, B.; Corke, P. Online novelty-based visual obstacle detection for field robotics. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 3935–3940.
28. Steen, K.A.; Therkildsen, O.R.; Green, O.; Karstoft, H. Detection of bird nests during mechanical weeding by incremental background modeling and visual saliency. *Sensors* **2015**, *15*, 5096–5111.
29. Campos, Y.; Sossa, H.; Pajares, G. Spatio-temporal analysis for obstacle detection in agricultural videos. *Appl. Soft Comput.* **2016**, *45*, 86–97.
30. Xu, P.; Ye, M.; Li, X.; Liu, Q.; Yang, Y.; Ding, J. Dynamic Background Learning Through Deep Auto-encoder Networks. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 107–116.

31. Braham, M.; Van Droogenbroeck, M. Deep background subtraction with scene-specific convolutional neural networks. In Proceedings of the 2016 International Conference on Systems, Signals and Image (IWSSIP), Bratislava, Slovakia, 23–25 May 2016; pp. 1–4.
32. Li, G.; Yu, Y. Deep Contrast Learning for Salient Object Detection. *arXiv* **2016**, arXiv:cs.CV/1603.01976.
33. Christiansen, P.; Kragh, M.; Steen, K.; Karstoft, H.; Jørgensen, R.N. Platform for Evaluating Sensors and Human Detection in Autonomous Mowing Operations. *Precis. Agric.* **2015**, submitted.
34. Christiansen, P.; Kragh, M.; Steen, K.A.; Karstoft, H.; Jørgensen, R.N. Advanced sensor platform for human detection and protection in autonomous farming. *Precis. Agric.* **2015**, *15*, 1330–1334.
35. Steen, K.; Christiansen, P.; Karstoft, H.; Jørgensen, R. Using Deep Learning to Challenge Safety Standard for Highly Autonomous Machines in Agriculture. *J. Imaging* **2016**, *2*, 6.
36. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 675–678.
37. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. *arXiv* **2013**, arXiv:1311.2901.
38. Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; Darrell, T. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In Proceedings of the 31st International Conference on Machine Learning (ICML-14), Beijing, China, 21–26 June 2014.
39. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014.
40. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv* **2015**, arXiv:cs.CV/1511.07122.
41. Benezeth, Y.; Jodoin, P.M.; Emile, B.; Laurent, H.; Rosenberger, C. Comparative study of background subtraction algorithms. *J. Electron. Imaging* **2010**, *19*, 033003.
42. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1977**, *39*, 1–38.
43. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874.
44. Davis, J.; Goadrich, M. The Relationship Between Precision-Recall and ROC Curves. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; ACM: New York, NY, USA, 2006; pp. 233–240.
45. Van Rijsbergen, C.J. *Information Retrieval*; Butterworths: London, UK, 1979.
46. Dollar, P.; Belongie, S.; Perona, P. The Fastest Pedestrian Detector in the West. In Procedings of the British Machine Vision Conference, Aberystwyth, UK, 30 August–2 September 2010.
47. Appel, R.; Belongie, S.; Perona, P.; Doll, P. Fast Feature Pyramids for Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1532–1545.
48. Nam, W.; Dollár, P.; Han, J.H. Local Decorrelation For Improved Detection. *arXiv* **2014**, arXiv:1406.1134.
49. Christiansen, P.; Sørensen, R.; Skovsen, S.; Jæger, C.D.; Jørgensen, R.N.; Karstoft, H.; Arild Steen, K. Towards Autonomous Plant Production using Fully Convolutional Neural Networks. In Procedings of the International Conference on Agricultural Engineering, Aarhus, Denmark, 26–29 June 2016.
50. Mottaghi, R.; Chen, X.; Liu, X.; Cho, N.G.; Lee, S.W.; Fidler, S.; Urtasun, R.; Yuille, A. The Role of Context for Object Detection and Semantic Segmentation in the Wild. In Procedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Beijing, China, 23–28 June 2014; pp. 891–898.

Paper 2

Automated Detection and Recognition of Wildlife Using Thermal Cameras

Peter Christiansen, Kim Arild Steen, Rasmus Nyholm Jørgensen and Henrik Karstoft

Peer reviewed
Paper published in Sensors, July 2014

Sensors **2014**, *14*, 13778-13793; doi:10.3390/s140813778

OPEN ACCESS
sensors
ISSN 1424-8220
www.mdpi.com/journal/sensors

Article

Automated Detection and Recognition of Wildlife Using Thermal Cameras

Peter Christiansen *, Kim Arild Steen, Rasmus Nyholm Jørgensen and Henrik Karstoft

Department of Engineering, Aarhus University, Finlandsgade 22, Aarhus, Denmark;
E-Mails: kim.steen@eng.au.dk (K.A.S.); rnj@eng.au.dk (R.N.J.); hka@eng.au.dk (H.K.)

* Author to whom correspondence should be addressed; E-Mail: pech@eng.au.dk;
Tel.: +45-2759-2953.

Received: 15 May 2014; in revised form: 1 July 2014 / Accepted: 17 July 2014 /
Published: 30 July 2014

Abstract: In agricultural mowing operations, thousands of animals are injured or killed each year, due to the increased working widths and speeds of agricultural machinery. Detection and recognition of wildlife within the agricultural fields is important to reduce wildlife mortality and, thereby, promote wildlife-friendly farming. The work presented in this paper contributes to the automated detection and classification of animals in thermal imaging. The methods and results are based on top-view images taken manually from a lift to motivate work towards unmanned aerial vehicle-based detection and recognition. Hot objects are detected based on a threshold dynamically adjusted to each frame. For the classification of animals, we propose a novel thermal feature extraction algorithm. For each detected object, a thermal signature is calculated using morphological operations. The thermal signature describes heat characteristics of objects and is partly invariant to translation, rotation, scale and posture. The discrete cosine transform (DCT) is used to parameterize the thermal signature and, thereby, calculate a feature vector, which is used for subsequent classification. Using a k-nearest-neighbor (kNN) classifier, animals are discriminated from non-animals with a balanced classification accuracy of 84.7% in an altitude range of 3–10 m and an accuracy of 75.2% for an altitude range of 10–20 m. To incorporate temporal information in the classification, a tracking algorithm is proposed. Using temporal information improves the balanced classification accuracy to 93.3% in an altitude range 3–10 of meters and 77.7% in an altitude range of 10–20 m

Keywords: thermal imaging; feature extraction; kNN; DCT; pattern recognition

1. Introduction

In agricultural mowing operations, thousands of animals are injured or killed each year, due to the increased working widths and speeds of agricultural machinery. Several methods and approaches have been used to reduce this wildlife mortality. Delayed mowing date, altered mowing patterns (e.g., mowing from the center outwards [1]) or strategy (e.g., leaving edge strips), longer mowing intervals, the reduction of speed or higher cutting height [1] have been suggested to reduce wildlife mortality rates. Likewise, searches with trained dogs prior to mowing may enable the farmer to remove, e.g., leverets and fawns to safety, whereas areas with bird nests can be marked and avoided. Alternatively, various scaring devices, such as flushing bars [1] or plastic sacks set out on poles before mowing [2], have been reported to reduce wildlife mortality. However, wildlife-friendly farming often results in lower efficiency. Therefore, attempts have been made to develop automatic systems capable of detecting wild animals in the crop without unnecessary cessation of the farming operation. For example, a detection system based on infrared sensors has been reported to reduce wildlife mortality in Germany [3]. The disadvantage of the system proposed in [3] is its low efficiency, as the maximum search power is around 3 ha/h, when the weather conditions are fit.

In the [4], principles from [3] were further developed and tested. They conclude that vision systems are not a viable solution when the cameras are mounted on the agricultural machinery, as image quality is highly affected by the speed and vibrations of the machine. Instead a UAV-based system is utilized [5]. Using this solution, the movement of the tractor does not affect the image quality, and it is possible to manually scan large areas. The authors show that thermal imaging can be used to detect roe deer fawns based on aerial footage. However, the detection is performed manually and should be automated to increase efficiency. They conclude that the thermal imaging strategy is sensitive to the detection of false positives, meaning that objects that are heated by the Sun are falsely labeled (manually) as roe deer fawns.

UAVs are an emerging technology, and in modern agriculture, it can be utilized for many purposes. The UAV technology is capable of performing advanced and high precision tasks, due to the flight capabilities and the possibility to equip the aerial vehicle with computers and sensors, including thermal cameras. During the last two decades, thermal imaging has gained more and more attention in computer vision and digital image processing research and applications. Thermal imaging has become an interesting technology in outdoor surveillance, pedestrian detection and agriculture, due to the invariance to illumination and the lowered price of thermal cameras [6].

In [7,8], thermal imaging is used for person detection. The authors present thermal images of people at different times of the day and during summer and winter. Here, it is clear that the object of interest (people) does not always appear brighter (higher temperature) than the background. They propose background subtraction techniques, followed by a contour-based approach to detect people in the thermal images. Background subtraction is also utilized in [9–11]. However, this approach is not suitable for our UAV-based application with non-stationary cameras, as the background changes rapidly over time, and it is not possible to construct a background image. Another approach is the detection of hot spots based on a fixed temperature threshold [12–15]. In [16], a probabilistic approach for defining the threshold value is presented; however, it is still a fixed value.

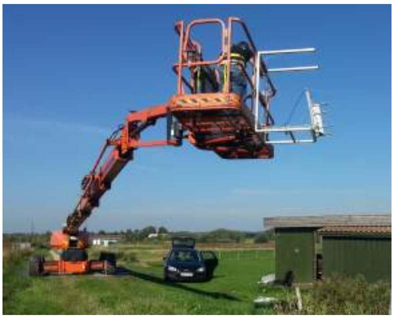
There is little research within the automatic detection and recognition of animals in thermal images. Most research with thermal cameras involve static cameras, where background subtraction has been used for robust people detection in thermal images. In [5], a UAV, equipped with a thermal camera, is used for the detection of roe deer fawns in agricultural fields. Detection is based on manual visual inspection, and the author utilizes automatic gain control to enhance the appearance of living objects. An algorithm for the classification of roe deer fawns in thermal images is presented in [17]. They utilize normalized compression distance as the features followed by a clustering algorithm for classification. The dataset consists of 103 images, with 26 containing fawns hidden in grass. The same dataset is used in [18], where fast compression distance is applied in the feature extraction step and a nearest neighbor classifier is used for classification. In both papers, the features are derived from a dictionary, generated by a compression algorithm. These features are scale invariant; however, they are not rotation invariant, and they rely on absolute temperature measurements, which could be invalidated if animals are heated by the Sun. An algorithm for automatic detection of wildlife in agricultural fields is presented in [19]. However, the distinction between animals and other hot objects is not a part of the results presented. An algorithm for the identification of deer, to avoid deer-vehicle crashes, is presented in [20]. The histogram of oriented gradient (HOG) is used for feature extraction, and support vector machines are utilized in the classification step. Their method relies on occlusion-free side-view images and performs poorly if these criteria are not met.

This paper presents a method for detecting and recognizing animals in thermal images. The method is based on a threshold, dynamically fitted for each frame, and a novel feature extraction algorithm, which is invariant to rotation, scaling and, partly, posture. Detected objects are tracked in subsequent images to include temporal information within the recognition part of the algorithm. The algorithm has been tested in a controlled experiment, using real animals, in the context of wildlife-friendly farming.

2. Materials and Methods

A telescopic boom is used to capture top-view images above a stationary scene, as shown in Figure 1. By using a telescopic boom lift, images can be captured at different altitudes, thus simulating the UAV. Unlike, using a UAV, the captured images are not affected by wind or vibrations within the UAV, which could affect image quality. Furthermore, the setup also avoids the compression of data, which might degrade data quality with respect to classification.

Figure 1. The setup used for capturing visual RGB and thermal images.

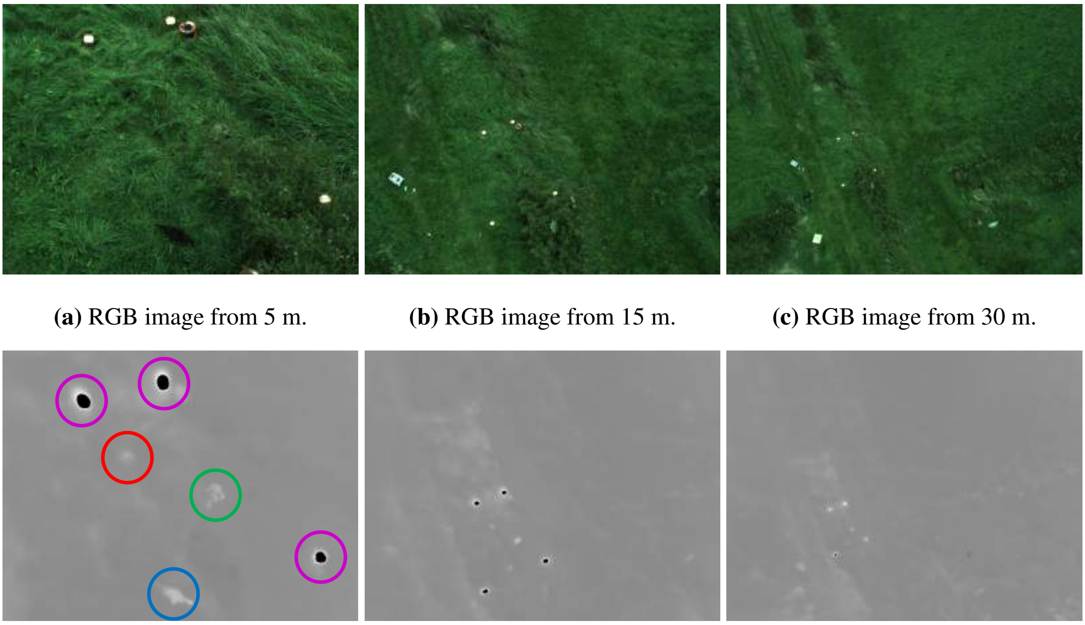


2.1. Data

A rig with a thermal and a regular RGB camera is mounted on the lift, recording 9 frames per second with a resolution of 320 × 240 and 1624 × 1234, respectively. Animals and four halogen spotlights (used as reference points) were manually placed below the lift within the field-of-view of the two imaging sensors. The altitude of the cameras was measured with a GPS. A total of six recordings were made through two days with temperatures of 15–19 °C and 16–23 ° C respectively. The recordings were captured using different areas around the scene shown in Figure 1.

Each recording starts at three meters followed by an increase in height of up to 25–35 m and then back again. The telescopic boom alternates the height position of the camera, while keeping the scene within the image frame. The use of a lift instead of an actual UAV results in less motion blur. The data used in this paper consist of a total of 3987 frames with the presence of animals (rabbit and chicken), together with other hot objects (halogen spotlights, molehills, wooden poles, *etc*.). Animals were able to move within in a certain area due to fixation by a 30-cm leash. In Figure 2 the same scene is captured from 5 m, 15 m and 30 m. All thermal images are rescaled to the same size as the RGB images.

Figure 2. Visual RGB and thermal images capture the same scene from 5 m (**a**), 15 m (**b**) and 30 m (**c**). The scene consists of four halogen spotlights, a molehill, a rabbit and a chicken. The halogen spotlights are easily visible in all images. In (**d**) a molehill, a rabbit, a chicken and three halogen spotlights are marked.



(**a**) RGB image from 5 m. (**b**) RGB image from 15 m. (**c**) RGB image from 30 m.

(**d**) Thermal image from 5 m. (**e**) Thermal image from 15 m. (**f**) Thermal image from 30 m.

2.2. Detection

The measured temperature is not the actual body temperature of the animal, as the measurement is also dependent on heating from the Sun, the insulative properties of the fur, or feather coat, and the distance between the animal and the camera [21]. These factors may vary in outdoor environments; hence, the segmentation and subsequent blob detection needs to adapt to this environment.

We use a threshold dynamically adjusted to each frame by using the median temperature $\tilde{t}$ in the image, to exclude outliers. The threshold value is set by:

$$th = \tilde{t} + c \tag{1}$$

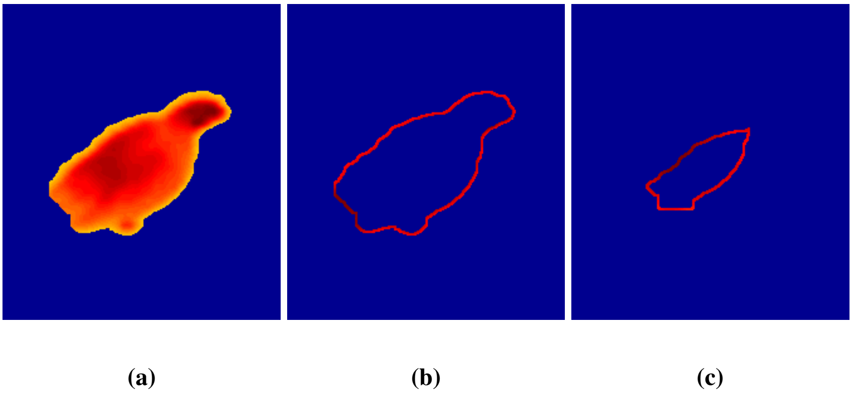
where the constant c ensures that only objects that are significantly warmer than the background are detected.

2.3. Feature Extraction: Thermal Signatures

We propose a novel feature, extracted from the thermal images, that is invariant to translation, rotation, scale and, partly, posture.

Based on the detected object as in Figure 3a, the perimeter contour is extracted using a four-connected neighborhood structuring element. An example of an extracted contour is shown in Figure 3b. For each iteration, the mean value of the contour is determined, and the object is shrinked by the contour. The procedure continues to iterate, until no more contours can be extracted from the object (e.g., in Figure 3b,c, the first and seventh contour are shown).

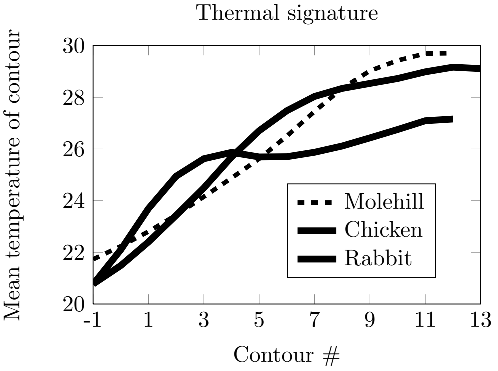
Figure 3. The process of extracting the thermal signature. (**a**) Thermal image of the detected object; (**b**) the first contour of the detected object; (**c**) the seventh contour of the detected object.



(a) **(b)** **(c)**

The thermal signature of an object is defined as the mean thermal value of the contour in each iteration i and denoted as $cm(i)$ for $i = -1, \ldots, M$, where M is the maximum number of iterations possible for the given object. The first iteration is defined as $i = -1$, as the object is initially dilated once to get edge information just outside the object. In Figure 4, $cm(i)$ is shown for different objects. In our dataset, a typical animal signature has a greater temperature increase close to the object boundary than a non-animal object.

Figure 4. Thermal signatures extracted from shrinking thermal contours at a height of 4.9 m. Contour number -1 is not part of the object, but used for edge feature extraction.



2.3.1. Parameterization of Thermal Signatures

The thermal signature describes certain characteristics of the objects. The signature is normalized by subtracting it with the mean temperature of the first contour. To make it invariant to the maximum number of contours, the signature can be approximated by resampling or by matching it to a high order polynomial. However, as the signature has sinusoidal characteristics, a Fourier-related transform is applicable. The discrete cosine transform (DCT) is chosen for is sinusoidal basis functions and its decorrelation properties. A fixed number of DCT coefficients will provide an approximation of the thermal signature and a set of features to be used in the classification. These feature vectors are then classified as either animal or non-animal, based on the k-nearest-neighbor (kNN) algorithm, which is briefly described in the next section.

2.4. Classification

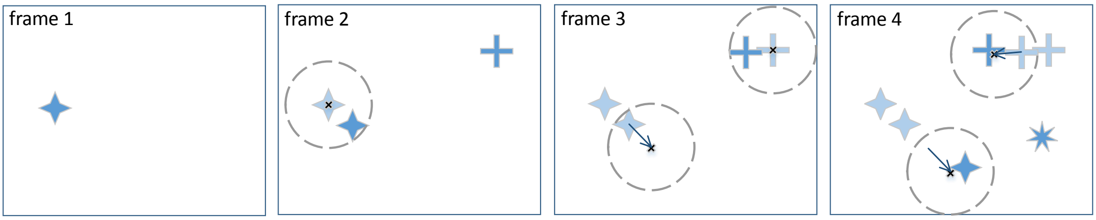
The kNN algorithm is a supervised learning algorithm, which can be used for both classification and clustering [22]. When used for classification, the algorithm is based on labeled training data. We extract 140 animal-feature vectors and 359 non-animal feature vectors as training data for the kNN classifier. More non-animal data are used, as the non-animal class contains more objects with different thermal characteristics. Thus, more training data is required to model this. Based on empirical experiments, the k parameter was set to 11, thereby including the nearest 11 training points during classification, which is based on majority voting.

2.5. Classification Using Temporal Information

A classification based on only a single frame using, e.g., a kNN classifier, discards the important temporal information provided in a recording. A lightweight tracking algorithm is used to link similarly positioned objects through consecutive images in the recordings. As the experiment has been done with a lift in a controlled setting, the tracking algorithm is not designed to compensate for movements of a potential UAV. To end or start new tracks, each track predicts a region defined as a guess region, where a new object needs to be positioned. An object is added to a track if it is within the guess region. A new track is created if a newly detected hot object is outside the guess region of any current tracks. A track is terminated when it fails to include any new objects for a defined number of frames. The guess region is described by a center point and a radius, where the center point is extended by the movement between the two previous objects included in the specific track. An example of the algorithm is provided in Figure 5, where tracks one, two and three are marked with ✦ , ✚ and ✷, respectively.

- In Frame 1, a single object has been detected inside the frame. As no tracks have been registered, the newly detected point creates the first track, ✦.
- In Frame 2 two objects are detected. One point is within the guess region of the first track and is added to the first track. The second point is outside the guess region, and a new track is created, ✚.
- In Frame 3, new points are added to the second track. Notice that a new guess region is predicted by the previous movement, but as no animal has been detected within the guess region, no point is added to the track.
- In Frame 4, three objects are detected. Two points are added to the current two tracks, and the third point creates a new track, ✷.

Figure 5. Tracking procedure.



Every time an object is being assigned to a certain track, the belief is updated to identify the tracked object as either animal or non-animal. The belief of track m is defined as the posterior probability and formulated as the probability of a detected element being an animal A given the newly observed data D_n in frame n.

$$Bel_{A,m}(n) = P\left(A\,|D_n\right) = \frac{P\left(A\right)\cdot P\left(D_n\,|A\right)}{P\left(D_n\right)} \tag{2}$$

The term $P\left(D_n\right)$ describes the evidence of the observed data. The evidence is a scale factor that ensures that the posterior probability sums to one and can be rewritten by using the law of total probability:

$$P\left(D_n\right) = P\left(A\right)\cdot P\left(D_n\,|A\right) + P\left(A^c\right)\cdot P\left(D_n\,|A^c\right) \tag{3}$$

where A^c defines the non-animal objects. The term $P\left(A\right)$ is the prior probability and describes the belief of an object being an animal before the data D_n have been observed, also defined as the belief at $n-1$.

$$P\left(A\right) = Bel_{A,m}(n-1) \tag{4}$$

The probability $P\left(D_n\,|A\right)$ is described as the likelihood and defined as the discriminant function $g_A(D_n)$ of kNN given by the ratio of k_A and k.

$$P\left(D_n\,|A\right) = g_A(D_n) = \frac{k_A}{k} \tag{5}$$

where k_A is the number of kNN samples that are animals, e.g., if $k_A = 6$ (majority vote), the probability is $P(D|A) = \frac{6}{11} \approx 0.55$.

Substituting Equations (3)–(5) into Equation (2) yields an updating scheme for every newly detected object.

$$Bel_{A,m}\left(n\right) = \frac{Bel_{A,m}\left(n-1\right)\cdot g_A\left(D_n\right)}{Bel_{A,m}\left(n-1\right)\cdot g_A\left(D_n\right) + Bel_{A^c,m}\left(n-1\right)\cdot g_{A^c}\left(D_n\right)} \tag{6}$$

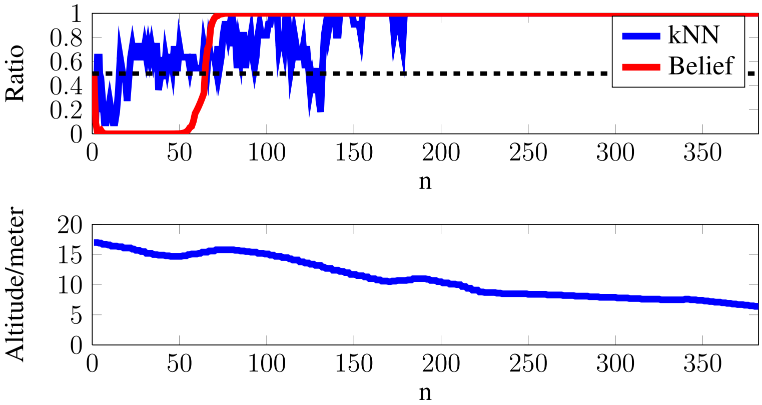
The belief updates every time an object is added to the track, but the track is finally identified as an animal if the belief exceeds 0.5. The prior probability of the first object in a track (n = 1) is set to $Bel_{A,m}(0) = 0.5$. The belief has a high chance of getting stuck in zero or one if the classifier returns, respectively, zero and one. To avoid this, the classifier will, as a minimum, return 0.05 and maximum 0.95.

The algorithm for tracking objects and building belief is fit for detecting animals in large fields using a UAV. The scenario is as follows: The UAV detects hot objects at high altitudes, thus allowing the UAV to cover large areas in a short time. Due to limited resolution, the detected objects are both small and almost uniform in thermal signature at high altitudes. As presented in the results section, this affects detection and recognition performance.

Therefore, the UAV should approach the objects to increase thermal image quality with respect to classification. By using the tracking algorithm, the belief is constantly calculated. Based on this temporal update of the belief, the algorithm can classify a detected object as an animal or a non-animal.

In Figure 6, the uppermost plot presents the kNN ratio from Equation (5) and the belief from Equation (6), which should be read as $1 = \textit{animal}$ and $0 = \textit{non-animal}$. The bottom plot shows the altitude of the recording rig. The example shows how the belief of an object evolves as the altitude decreases. In the example, it is seen that the algorithm believes that the detected object is non-animal. However, as the belief updates, the algorithm discards this when the altitude decreases.

Figure 6. Building a belief for decreasing altitudes.



3. Results

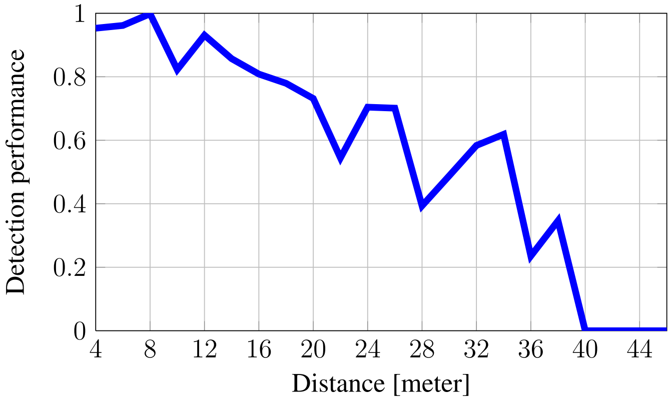
3.1. Detection

Objects are detected using the threshold set by Equation (1). The parameter c is set to $c = 5\,°C$ based on empirical experiments. The detection performance is defined as the ratio between the number of objects detected by the algorithm $l_{detected}$ and the actual number of animals l found by manual labeling.

$$D_{performance} = \frac{l_{detected}}{l}$$

Figure 7 shows how the detection performance rapidly degrades until it reaches zero for increasing altitude.

Figure 7. Detection performance for animals relative to altitude.



3.2. Feature Extraction and Classification

The thermal signature is approximated using seven DCT coefficients, as this describes 95% of the signature information for more than 95% of the provided data. Figure 8 presents an approximation of the thermal signature for two objects using seven DCT coefficients.

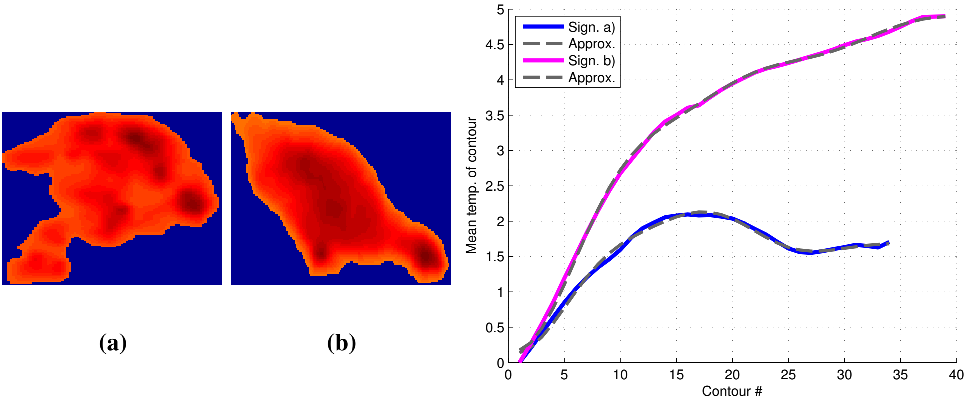
The classification accuracy is a common measure for classifier performance, but as presented in Figure 9a, fewer animals are detected by the segmentation algorithm for increasing altitudes. The loss of detected animals will make the data unbalanced, as it becomes dominated by non-animal samples in high altitudes.

To adjust the unbalanced data, a balanced classification accuracy is used to evaluate the classifier performance:

$$C_{accuracy.balanced} = \frac{sensitivity + specificity}{2} = \frac{TP/\left(TP + FN\right) + TN/\left(FP + TN\right)}{2}$$

where TP, FN, TN and FP are, respectively, true positive, false negative, true negative and false positive. Figure 9b shows the balanced accuracy and how performance degrades for increasing altitudes. The figure also shows that the algorithm is not able to provide satisfactory results for altitudes above 22 m, as the balanced accuracy drops below or around 0.5.

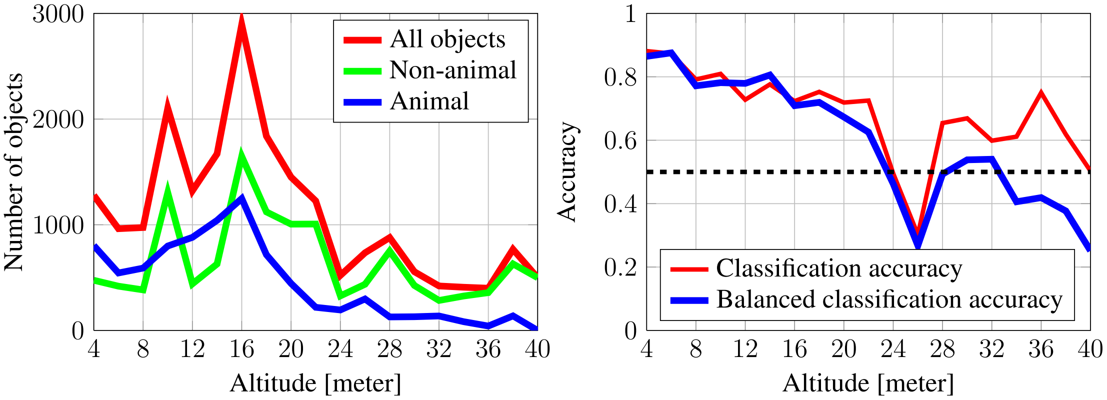
Figure 8. Thermal images and approximations of the thermal signature of a rabbit and chicken. (**a**) Thermal image of a rabbit; (**b**) thermal image of a chicken; (**c**) thermal signature and its seven discrete cosine transform coefficient approximation.



(a) **(b)**

(c)

Figure 9. Evaluation of the classifier relative to altitude. (**a**) The number of detected objects, animals and non-animals relative to altitude; (**b**) the classifier performance using classification accuracy and balanced accuracy relative to altitude.



(a) **(b)**

As the segmentation and classification are highly dependent on altitude, the classification is evaluated in the two different altitude ranges of 3–10 m and 10–20 m, defined as, respectively, the short- and far-range altitudes.

3.3. Tracking

The tracking algorithm has been setup to allow tracking of an object with three missing points and a maximum uncertainty of 190 pixels. The tracks are identified and labeled as animal or non-animal based on the updating scheme from Equation (6). After a track has been identified, all other objects in the track are changed to the similar label.

3.3.1. Short Range Altitudes (3–10 m)

In the altitude range of 3–10 m, the tracker distributes 4173 out of 4381 objects (95.3%) into tracks containing more than five points, where 4104 out of 4173 objects (98.3%) are placed in a track with the majority of the same label, meaning that 1.7% objects are placed in the wrong track. The balanced classification accuracy is 84.8% before the tracks have been identified, e.g., only kNN classification is performed. Combining the classification results from each frame with the temporal information in terms of tracks, the balanced accuracy is improved by 8.7 percentage points to 93.5%. The confusion matrix before and after tracking is provided in Tables 1 and 2. Table 3 shows different performance measures with and without tracking. Sensitivity or the true positive rate (TPR) describes the classifiers ability to identify an animal object correctly. Specificity or the true negative rate (TNR) describes the classifiers ability to identify a non-animal object correctly. After tracking, the TPR and TNR are 90.8% and 96.2%, respectively, indicating that the classifier has an advantage when classifying non-animal objects.

Table 1. Confusion matrix before track identification in the close-range altitudes (3–10 m).

		Observation	
		Animal	Non-animal
Prediction	Animal	2056	332
	Non-animal	330	1663

Table 2. Confusion matrix after track identification in close-range altitudes (3–10 m).

		Observation	
		Animal	Non-animal
Prediction	Animal	2167	76
	Non-animal	219	1919

Table 3. Performance measure in close-range altitudes (3–10 m).

	Performance measure	**No tracking**	**Tracking**
Range 3-10m	Classification accuracy	0.849	0.933
	Balanced classification accuracy	0.848	0.935
	Sensitivity, True positive rate	0.862	0.908
	Specificity, True negative rate	0.834	0.962

3.3.2. Far-Range Altitudes (10–20 m)

At an altitude of 10 to 20 m, the tracker distributes 8024 out of 8456 objects (94.9%) into tracks containing more than five points, where 7673 out of 8024 objects (95.6%) are placed in a track with the majority of the same label, meaning that 4.4% are placed in the wrong track.

The balanced classification accuracy is 75.2% before the tracks have been identified, while the balanced accuracy improves by 2.5 percentage points to 77.7%, when tracks are being identified. The confusion matrix before and after tracking is provided in Tables 4 and 5. Table 6 shows different performance measures with and without tracking. After tracking, the TPR and TNR are 63.4% and 90.2%, respectively, indicating that the classifier especially has difficulties classifying animal objects correctly in far-range altitudes.

Table 4. Confusion matrix before track identification in far-range altitudes 10–20 m.

		Observation	
		Animal	Non-animal
Prediction	Animal	2735	515
	Non-animal	1606	3600

Table 5. Confusion matrix after track identification in far-range altitudes 10–20 m.

		Observation	
		Animal	Non-animal
Prediction	Animal	2753	331
	Non-animal	1588	3784

Table 6. Performance measure in far-range altitudes 10–20 m.

	Performance measure	No tracking	Tracking
Range 10-20m	Classification accuracy	0.749	0.773
	Balanced classification accuracy	0.752	0.777
	Sensitivity, True positive rate	0.630	0.634
	Specificity, True negative rate	0.875	0.920

The results show that information from consecutive frames in terms of determining and identifying tracks will improve performance by 8.7 and 2.5 percentage points for the short- and far-range altitudes, respectively. The system performs best in close-range altitudes with an accuracy of 93.5%, providing a lead of 15.8 percentage points compared to the far altitude range. The system maintains, though, a low number of FP or a high TNR of, respectively, 96.2% and 92.0% for short and far altitudes, meaning that the system preserves the ability to classify non-animals correctly in both ranges. Conversely, the TPR drops from 90.8% to 63.5%, meaning that the classifier especially has difficulties in recognizing animal objects correctly in far-range altitudes.

4. Discussion

The presented feature extraction and classification scheme shows good detection and classification performance for recording heights under 10 m with a balanced classification accuracy of 84.8%. In the altitude range of 10–20 m, the performance drops, having a balanced classification accuracy of 75.2%. The procedure becomes unfit for altitudes above 20–22 m, as detection performance decreases, but the altitude limit is ultimately set by a bad recognition, as the balanced classification accuracy drops below or around 0.5. Multiple arguments demonstrate that the application degrades for increasing altitudes, ultimately making it unfit for detecting and classifying small animals in altitudes above 20 m. The decreased detection relative to altitude is explained by the following reasons:

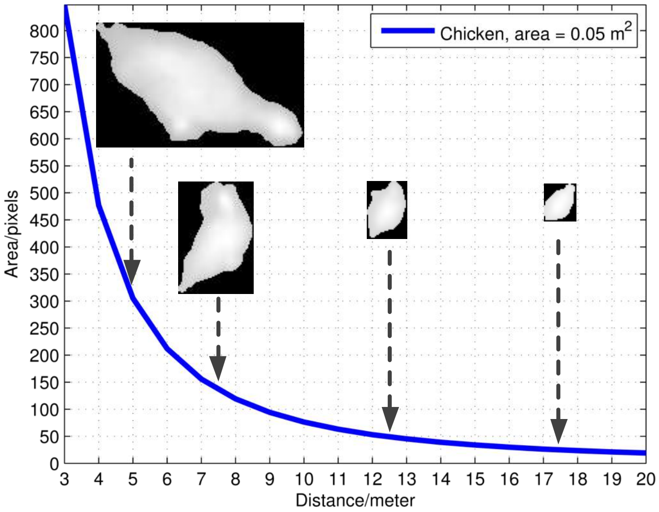
(1) The thermal radiation received by the sensor decreases as the distance to animal increases.
(2) The size of an animal is decreased for increasing altitudes, allowing the animal to be dominated by its colder surroundings.
(3) For a given image resolution and FOV, the spatial resolution or ground sample distance will, above a certain altitude, exceed the size of the animal, making it undetectable for the thermal imaging sensor.

The drop in classifier performance for increasing altitudes is explained by the increasing ground sample distance, causing the object to be presented in lower resolution or by less information, e.g., the area of a chicken (around 0.05 m^2) will, from an altitude of 5 m, theoretically be presented by 305 pixels, while the same chicken is presented by only 19 pixels from an altitude of 20 m. Figure 10 shows how the pixel area of a chicken theoretically decreases relative to altitude and how a chicken, in practice, ends up losing characteristics.

Performance can, though, be improved in high altitudes for both detection and classification by using a higher resolution camera, a more narrow FOV or optical zoom. The decrease in performance for increasing altitudes fits well with observations from [5], where the authors were able to manually detect row deer fawns at 30 m, but had problems at 50 m with a thermal camera with a resolution of 640 × 512 pixels. The animals used in this paper are smaller than roe deer fawns, which results in fewer thermal pixels, compared to the roe deer fawns.

Tracking objects in subsequent images enables us to exploit the temporal information in the recording and improve performance. The proposed tracking algorithm improves the balanced accuracy by 8.7 percentage points to 93.5% in short-range altitudes and by 2.5 percentage points to 77.7% in far-range altitudes. A lightweight tracking algorithm has been applied to simply prove how performance can be improved by exploiting the temporal data. Tracking should, in a real application, handle larger movements in the horizontal plane and could be combined with a gimbal to stabilize the camera, independent of yaw, roll and pitch.

Figure 10. The pixel area relative to the distance for a ground area of 0.05 m^2. Thermal images of a rescaled chicken from different altitudes.



The manually-extracted training data is based on two types of animals: rabbits and chickens. However, other animals are of interest within the scope of wildlife-friendly agriculture. More experiments, including different weather conditions, vegetation, animals and more non-animal candidates to extend the variation of our somewhat limited dataset, should be conducted. These experiments could help improve the existing algorithm or increase our knowledge of using thermal cameras for automatic detection and recognition of wildlife. Furthermore, the applicability of the used methods should be evaluated using footage taken from an actual UAV in motion to include the effects of wind, UAV movements, moving animals and to more easily extend the variety of the dataset.

The set used for the testing and training of the classifier has no overlapping data. However, as the training data have been selected from, e.g., every 50th frame in a recording, the data used for testing and training are correlated to some extent.

This paper focuses on thermal imaging and the proposed feature extraction method. However, sensor fusion, using the RGB camera, could potentially increase classification performance. Therefore, sensor fusion methods should be investigated to accomplish this.

5. Conclusion

We have introduced a method for the automatic detection and recognition of wildlife using thermal cameras for UAV technology. Based on a dynamic threshold, hot objects are detected and subsequent feature extraction is performed. The novel feature extraction method, presented in this paper, consist of an extraction of thermal signatures for each detected object and a parameterization of this based on DCT.

Methods for classification using measurements from both single and multiple frames is presented. Combining measurements from multiple frames achieves the best performance, with a balanced classification accuracy of 93.5% in the altitude range of 3–10 m and 77.7% in the altitude range of

10–20 m, thus demonstrating a clear relationship between the performance of detection and classification relative to altitude. The simulated and limited dataset is favorable in terms of performance for the given algorithms. The actual applicability of the system should therefore be determined using footage from an actual UAV. The proposed detection and classification scheme is based on top-view images of wildlife, as seen by a UAV. The use of UAV-technology for automatic detection and recognition of wildlife is currently part of ongoing research towards wildlife-friendly agriculture.

Author Contributions

Peter Christiansen and Kim Arild Steen have made substantial contributions in the development of the algorithms presented in this paper. Henrik Karstoft has made a substantial contribution in manuscript preparation. Rasmus Nyholm Jørgensen has made a contribution to the definition of the research and data acquisition.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Green, C. *Reducing Mortality of Grassland Wildlife during Haying and Wheat-Harvesting Operations*; Oklahoma State University Forestry Publications: Stillwater, OK, USA, 1998; pp. 1–4.
2. Jarnemo, A. Roe deer Capreolus capreolus fawns and mowing-mortality rates and countermeasures. *Wildl. Biol.* **2002**, *8*, 211–218.
3. Haschberger, P.; Bundschuh, M.; Tank, V. Infrared sensor for the detection and protection of wildlife. *Opt. Eng.* **1996**, *35*, 882–889.
4. Israel, M.; Schlagenhauf, G.; Fackelmeier, A.; Haschberger, P.; Oberpfaffenhofen, D.; GmbH, C.S.; MÃijnchen, T. Study on Wildlife Detection During Pasture Mowing. 2011. Available online: http://elib.dlr.de/65977/1/WildretterVDIv4.pdf (accessed on 26 February 2014); p. 6.
5. Israel, M. A UAV-based roe deer fawn detection system. In Proceedings of the International Conference on Unmanned Aerial Vehicle in Geomatics, Zurich, Switzerland, 14 September 2011; pp. 1–5.
6. Gade, R.; Moeslund, T.B. Thermal cameras and applications: A survey. *Mach. Vision Appl.* **2013**, *25*, 1–18.
7. Davis, J.W.; Sharma, V. Robust background-subtraction for person detection in thermal imagery. In Proceedings of the IEEE International Workshop on Object Tracking and Classification Beyond the Visible Spectrum, New York, NY, USA, 22 June 2004.
8. Davis, J.W.; Keck, M.A. A Two-Stage Template Approach to Person Detection in Thermal Imagery. In Proceedings of the Workshop Applications of Computer Vision, Breckenridge, CO, USA, 5–7 January 2005; pp. 364–369.

9. Goubet, E.; Katz, J.; Porikli, F. Pedestrian tracking using thermal infrared imaging. In Proceedings of the SPIE Conference Infrared Technology and Applications, Cambridge, UK, 15 May 2006; pp. 797–808.
10. Leykin, A.; Ran, Y.; Hammoud, R. Thermal-visible video fusion for moving target tracking and pedestrian classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR'07, Minneapolis, MA, USA, 17–22 June 2007; pp. 1–8.
11. Torresan, H.; Turgeon, B.; Ibarra-Castanedo, C.; Hebert, P.; Maldague, X.P. Advanced surveillance systems: Combining video and thermal imagery for pedestrian detection. Defense and Security. International Society for Optics and Photonics, 2004; pp. 506–515.
12. Cielniak, G.; Duckett, T. People recognition by mobile robots. *J. Intell. Fuzzy Syst.* **2004**, *15*, 21–27.
13. Li, J.; Gong, W.; Li, W.; Liu, X. Robust pedestrian detection in thermal infrared imagery using the wavelet transform. *Infrared Phys. Technol.* **2010**, *53*, 267–273.
14. Fernández-Caballero, A.; López, M.T.; Serrano-Cuerda, J. Thermal-Infrared Pedestrian ROI Extraction through Thermal and Motion Information Fusion. *Sensors* **2014**, *14*, 6666–6676.
15. Rudol, P.; Doherty, P. Human body detection and geolocalization for UAV search and rescue missions using color and thermal imagery. In Proceedings of the 2008 IEEE Aerospace Conference, Big Sky, MT, USA, 1–8 March 2008; pp. 1–8.
16. Nanda, H.; Davis, L. Probabilistic template based pedestrian detection in infrared videos. In Proceedings of the IEEE Intelligent Vehicle Symposium, Dearborn, MI, USA, 17–21 June 2002; Volume 1, pp. 15–20.
17. Cerra, D.; Israel, M.; Datcu, M. Parameter-free clustering: Application to fawns detection. In Proceedings of the 2009 IEEE International, IGARSS, Geoscience and Remote Sensing Symposium, Cape Town, South Africa, 12–17 July 2009; Volume 3, p. III-467.
18. Israel, M.; Evers, S.; für Luft, D.Z.; Raumfahrt, O. Mustererkennung zur Detektion von Rehkitzen in Thermal-Bildern. 2011; pp. 20–28. (In German)
19. Steen, K.A.; Villa-Henriksen, A.; Therkildsen, O.R.; Green, O. Automatic Detection of Animals in Mowing Operations Using Thermal Cameras. *Sensors* **2012**, *12*, 7587–7597.
20. Zhou, D.; Wang, J.; Wang, S. Countour Based HOG Deer Detection in Thermal Images for Traffic Safety. In Proceedings of the International Conference on Image Processing,Computer Vision, and Pattern Recognition, Las Vegas, NV, USA, 16–19 July 2012; pp. 1–6.
21. Systems, F.C.V. *Thermal Imaging: How Far Can You See with It?*; Technical Note; FLIR Systems: Wilsonville, OR, USA; Available online: http://www.flir.com/uploadedfiles/eng_01_howfar.pdf (accessed on 17 January 2014); pp. 1–4.
22. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.

Paper 3

FieldSAFE: Dataset for Obstacle Detection in Agriculture
Mikkel Kragh, Peter Christiansen, Morten S. Laursen, Morten Larsen, Kim A. Steen, Ole Green, Henrik Karstoft and Rasmus N. Jørgensen

Preprint (Submitted to Sensors 28.09.2017)
Paper on arxiv, September 2017

FieldSAFE: Obstacle Detection Dataset in Agriculture

Mikkel Kragh*[1], Peter Christiansen*[1], Morten S. Laursen[1], Morten Larsen[2], Kim A. Steen[3], Ole Green[3] and Rasmus N. Jørgensen[1]

[1]Department of Engineering, Aarhus University, Denmark
[2]Conpleks Innovation ApS, Struer, Denmark
[3]AgroIntelli, Aarhus, Denmark

Abstract

In this paper, we present a novel multi-modal dataset for obstacle detection in agriculture. The dataset comprises approximately 2 hours of raw sensor data from a tractor-mounted recording system in a grass mowing scenario in Denmark, October 2016. Sensing modalities include stereo camera, thermal camera, web camera, 360-degree camera, lidar, and radar, while precise localization is available from fused IMU and GPS. Both static and moving obstacles are present including humans, mannequin dolls, rocks, barrels, buildings, vehicles, and vegetation. All obstacles have ground truth object labels and GPS coordinates.

Keywords — dataset, agriculture, obstacle detection, computer vision, cameras, stereo, thermal, lidar, radar, tracking

I. Introduction

For the past few decades, precision agriculture has revolutionized agricultural production systems. Part of the development has focused on robotic automation, to optimize workflow and minimize manual labor. Today, technology is available to automatically steer farming vehicles such as tractors and harvesters along predefined paths utilizing accurate global navigation systems. However, a human is still needed to monitor the surroundings and act upon potential obstacles in front of the vehicle to ensure safety.

In order to completely eliminate the need for a human operator, autonomous farming vehicles need to operate both efficiently and safely without any human intervention. A safety system must perform robust obstacle detection and avoidance in real-time with high reliability. And multiple sensing modalities must complement each other in order to handle a wide range of changes in illumination and weather conditions.

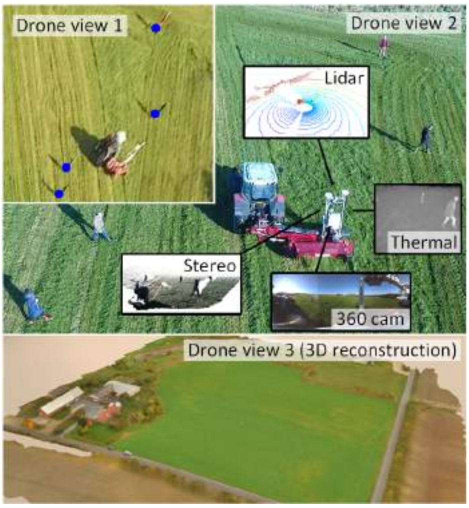


Figure 1: *Recording platform surrounded by static and moving obstacles. Multiple drone views record the exact position of obstacles, while the recording platform records local sensor data.*

A technological advancement like this requires extensive research and experiments to investigate combinations of sensors, detection algorithms, and fusion strategies. Currently, a few R&D projects exist within companies that

*{mkha,pech}@eng.au.dk
Both authors contributed equally.

seek to commercialize the concept [4, 2, 12]. However, no public platforms or datasets are available that address the important issues of obstacle detection in agricultural environments.

Within urban autonomous driving, a number of datasets have recently been made publicly available. Udacity's Self-Driving Car Engineer Nanodegree program has given rise to multiple challenge datasets including stereo camera, lidar, and localization data [1, 19, 18]. A few research institutions such as the University of Surrey [7], Linköping University [11], Oxford [13], and Virginia Tech [10] have published similar datasets. Most of the above cases, however, only address behavioural cloning, such that ground truth data are only available for control actions of the vehicles. No information is thus available for potential obstacles and their location in front of the vehicles.

The KITTI dataset [9], however, addresses these issues with object annotations in both 2D and 3D. Today, it is the de facto standard for benchmarking both single- and multi-modality object detection and recognition systems for autonomous driving. The dataset includes a high-resolution stereo camera, a 360-degree camera, a lidar, and fused GPS/IMU sensor data.

Focusing specifically on image data, an even larger selection of datasets is available with annotations of typical object categories such as cars, pedestrians, and bicycles. Annotations of cars are often represented by bounding boxes [14, 3]. However, pixel-level annotation or semantic segmentation has the advantage of being able to capture all objects, regardless of their shape and orientation. Some of these are synthetically generated images using computer graphic engines that are automatically annotated [17, 8], whereas others are natural images that are manually labeled [6, 15].

In agriculture, currently no similar datasets are publicly available. While some similarities between autonomous urban driving and autonomous farming are present, essential differences exist. An agricultural environment is often unstructured, whereas urban driving involves planar surfaces, often accompanied by lane lines and traffic signs. Further, distinction between traversable, non-traversable and processable terrain is often necessary in an agricultural context such as grass mowing, weed spraying, or harvesting. Here, tall grass or high crops protruding from the ground may actually be traversable and processable, whereas ordinary object categories such as humans, animals, and vehicles are not. In urban driving, however, a simplified traversable/non-traversable representation is common, as all protruding objects are typically regarded as obstacles. Therefore, sensing modalities and detection algorithms that work well in urban driving, do not necessarily work well in an agricultural setting. Ground plane assumptions common for 3D sensors may break down when applied on rough terrain or high grass. And vision-based detection algorithms may fail when faced with visually camouflaged objects such as animals and vegetation typical in a natural environment.

In this paper, we present a flexible, multimodal sensing platform and a dataset for obstacle detection in agriculture. The platform is mounted on a tractor and includes a stereo camera, a thermal camera, a web camera, a 360-degree camera, a lidar, and a radar, while precise localization is available from fused IMU and GPS. The dataset includes approximately 2 hours of recordings from a grass mowing scenario in Denmark, October 2016. Both static and moving obstacles are present including humans, mannequin dolls, rocks, barrels, buildings, vehicles, and vegetation. Ground truth positions of all obstacles were recorded with a drone during operation and have subsequently been manually labeled and synchronized with all sensor data. The dataset can be downloaded from `https://vision.eng.au.dk/fieldsafe/`.

II. Sensor Setup

Figure 2 shows the recording platform mounted on a tractor during grass mowing. The platform consists of the exteroceptive

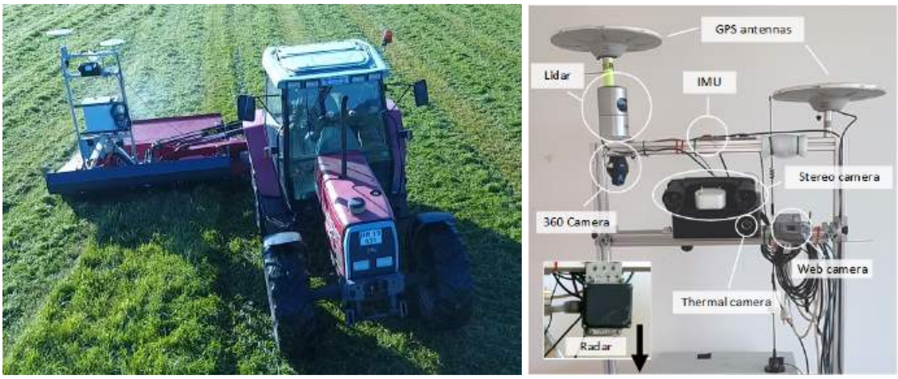


Figure 2: *Recording platform.*

Table 1: *Exteroceptive sensors.*

Sensor	Model	Resolution	FOV	Range	Data rate
Stereo camera	Multisense S21 CMV2000	1024 x 544	85°x 50°	1.5-50m	10 fps
Web camera	Logitech HD Pro C920	1920 x 1080	70°x 43°	-	20 fps
360-degree camera	Giroptic 360cam	2048 x 833	360°x 292°	-	30 fps
Thermal camera	Flir A65, 13 mm lens	640 x 512	45°x 37°	-	30 fps
Lidar	Velodyne HDL-32E	2172 x 32	360°x 40°	1-100 m	10 fps
Radar	Delphi ESR	64 targets/frame	90°x 4.2° 20°x 4.2°	0-60 m 0-174 m	20 fps

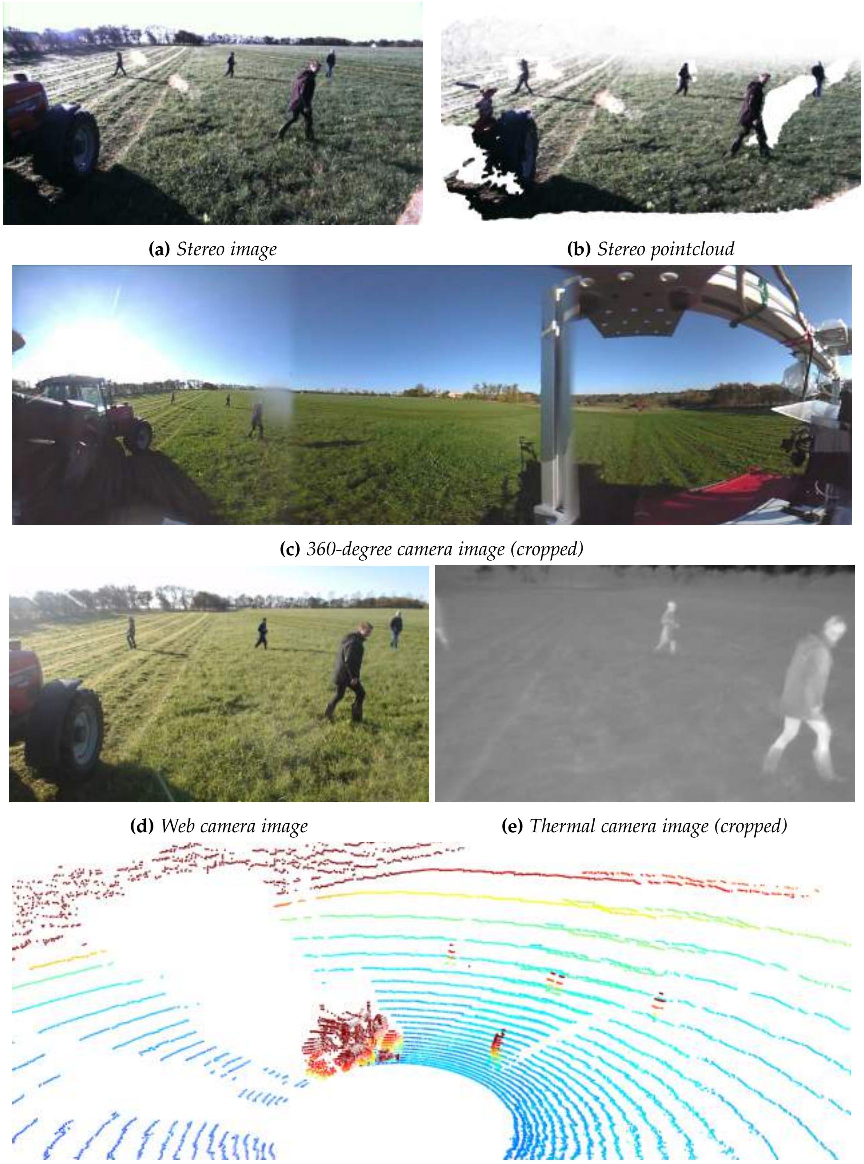
Table 2: *Proprioceptive sensors.*

Sensor	Model	Description
GPS	Trimble BD982 GNSS	Dual antenna RTK GPS system. Measures position and horizontal heading of the platform.
IMU	Vectornav VN-100	Measures acceleration, angular velocity, magnetic field, and barometric pressure.

sensors listed in Table 1, the proprioceptive sensors listed in Table 2, and a controller used for data collection with the Robot Operating System (ROS). Figure 3 illustrates a synchronized pair of frames from the stereo camera, the 360-degree camera, the web camera, the thermal camera, and the lidar.

Synchronization. Trigger signals for the stereo and thermal cameras were synchronized and generated from a PPS signal from the lidar, which allowed exact GPS timestamps for all three sensors. The remaining sensors were synchronized in software using ROS.

Registration. The lidar and the stereo camera were registered with ICP as an average over multiple static scenes. The stereo and thermal cameras were registered using a custom made visual-thermal checker board. The remaining sensors were registered by hand, by estimating extrinsic parameters of their positions. For a more detailed description, we refer the reader to [5].



(a) *Stereo image*

(b) *Stereo pointcloud*

(c) *360-degree camera image (cropped)*

(d) *Web camera image*

(e) *Thermal camera image (cropped)*

(f) *Lidar point cloud (cropped and colored by height)*

Figure 3: *Example frames from the FieldSAFE dataset.*

III. Dataset

The dataset consists of approximately 2 hours of recordings during grass mowing in Denmark, October 25th 2016. Figure 4a shows a map of the field with tractor tracks overlaid. The field is 3.3 ha and surrounded by roads, shelterbelts, and a private property.

A number of static obstacles exemplified in Figure 5 were placed on the field prior to recording. They included mannequin dolls (adults and children), rocks, barrels, buildings, vehicles, and vegetation. Figure 4b shows the placement of static obstacles on the field overlaid on a ground truth map colored by object classes.

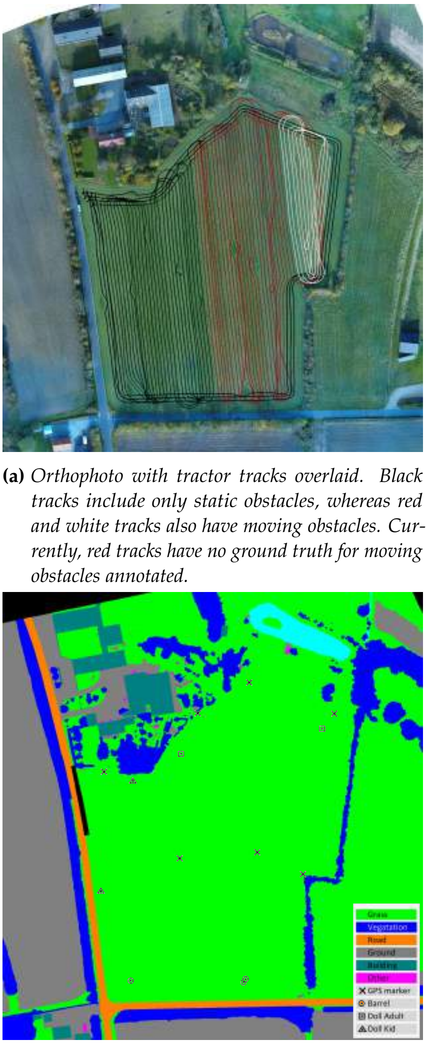
Additionally, a session with moving obstacles was recorded where four humans were told to walk in random patterns. Figure 6 shows the four subjects and their respective paths on a subset of the field. The subset corresponds to the white tractor tracks in Figure 4a. The humans crossed the path of the tractor a number of times, thus emulating dangerous situations that must be detected by a safety system. Along the way, various poses such as standing, sitting, and lying were represented.

During the entire traversal and mowing of the field, data from all sensors were recorded. Along with video from a hovering drone, a static orthophoto from another drone, and corresponding manually annotated class labels, these are all available from the FieldSAFE website.

IV. Ground Truth

Ground truth information on object location and class labels for both static and moving obstacles is available as timestamped GPS coordinates. By transforming local sensor data from the tractor into global GPS coordinates, a simple look-up of class label into the annotated ground truth map is possible.

Prior to traversing and mowing the field, a number of custom-made GPS markers were distributed on the ground and measured with exact GPS coordinates using a handheld RTK



(a) *Orthophoto with tractor tracks overlaid. Black tracks include only static obstacles, whereas red and white tracks also have moving obstacles. Currently, red tracks have no ground truth for moving obstacles annotated.*

(b) *Labeled orthophoto*

Figure 4: *Colored and labeled orthophotos.*

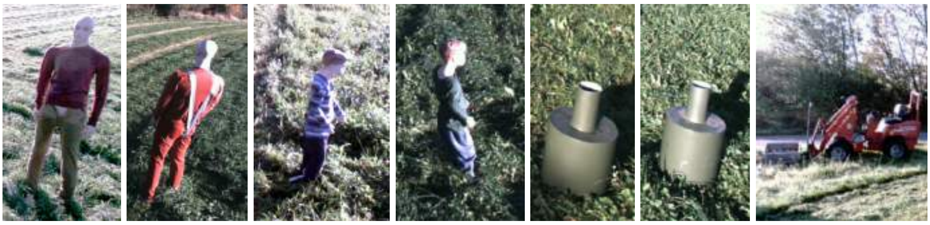
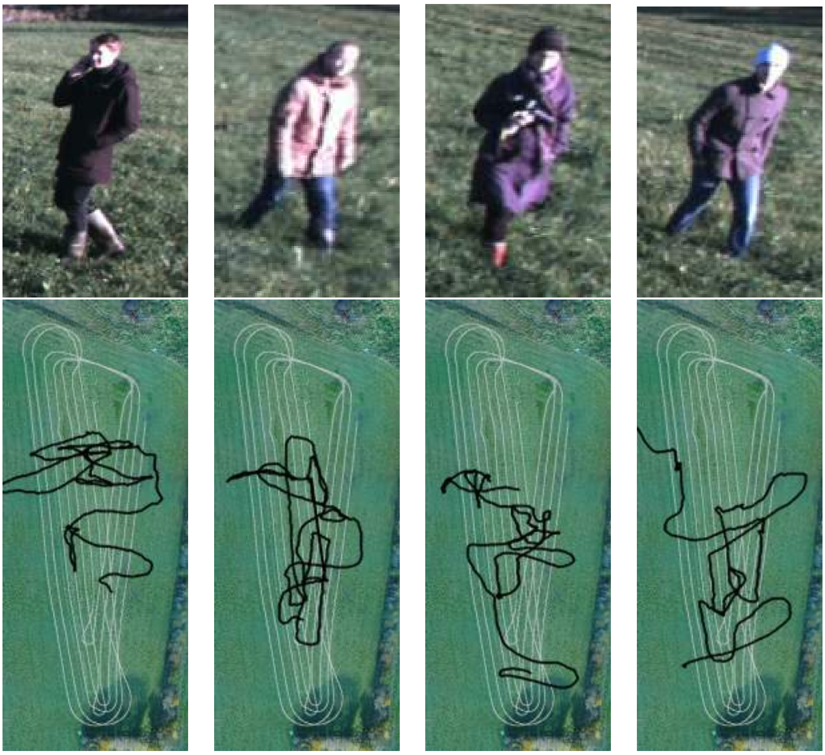


Figure 5: *Examples of static obstacles.*



(a) *Human 1* **(b)** *Human 2* **(c)** *Human 3* **(d)** *Human 4*

Figure 6: *Examples of moving obstacles (from the stereo camera) and their paths (black) overlaid on tractor path (grey).*

GPS. A DJI Phantom 4 drone was used to take overlapping bird's-eye view images of an area covering the field and its surroundings. Pix4D [16] was then used to stitch the images and generate a high-resolution orthophoto (Figure 4a) with a ground sampling distance (GSD) of 2 cm. The orthophoto was manually labeled pixel-wise as either *grass*, *ground*, *road*, *vegetation*, *building*, *GPS marker*, *barrel*, *human*, or *other* (Figure 4b). Using the GPS coordinates of the markers and their corresponding positions in the orthophoto, a mapping between GPS coordinates and pixel coordinates was estimated.

For annotating the location of moving obstacles, a DJI Matrice 100 was used to hover approximately 75 m above the ground while the tractor traversed the field. The drone recorded video at 25 fps with a resolution of 1920x1080. Due to limited battery capacity, the recording was split into two sessions of each 20 minutes. The videos were manually synchronized with sensor data from the tractor by introducing physical synchronization events in front of the tractor in the beginning and end of each session. Using the 7 GPS markers that were visible within field of view of the drone, the videos were stabilized and warped to a bird's-eye view of a subset of the field. As described above for the static orthophoto, GPS coordinates of the markers and their corresponding positions in the videos were then used to generate a mapping between GPS coordinates and pixel coordinates. Finally, the moving obstacles were manually annotated in each frame of one of the videos using the vatic video annotation tool [20]. Figure 6 shows the path of each object overlaid on a subset of the orthophoto. The second video is yet to be annotated.

V. Summary and Future Work

In this paper, we have presented a calibrated and synchronized multi-modal dataset for obstacle detection in agriculture. We envision the dataset to facilitate a wide range of future research within autonomous agriculture and obstacle detection for farming vehicles.

In future work, we plan on annotating the remaining session with moving obstacles. Additionally, we would like to extend the dataset with more scenarios from various agricultural environments while widening the range of encountered illumination and weather conditions.

References

[1] Didi Data Release #2 - Round 1 Test Sequence and Training.

[2] ASI. Autonomous Solutions. `https://www.asirobots.com/farming/`, 2016. Accessed: 2017-08-09.

[3] Claudio Caraffi, Tomas Vojir, Jura Trefny, Jan Sochman, and Jiri Matas. A System for Real-time Detection and Tracking of Vehicles from a Single Car-mounted Camera. In *ITS Conference*, pages 975–982, Sep. 2012.

[4] Case IH. Case IH Autonomous Concept Vehicle. `http://www.caseih.com/apac/en-in/news/pages/2016-case-ih-premieres-concept-vehicle-at-farm-progress-show.aspx`, 2016. Accessed: 2017-08-09.

[5] P. Christiansen, M. Kragh, K. A. Steen, H. Karstoft, and R. N. Jørgensen. Platform for evaluating sensors and human detection in autonomous mowing operations. *Precision Agriculture*, 18(3):350–365, Jun 2017.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] DIPLECS. DIPLECS Autonomous Driving Datasets. `http://ercoftac.mech.surrey.ac.uk/data/diplecs/`, 2015. Accessed: 2017-08-31.

[8] A Gaidon, Q Wang, Y Cabon, and E Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016.

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[10] InSight. InSight SHRP2. `https://insight.shrp2nds.us/`, 2017. Accessed: 2017-08-31.

[11] Philipp Koschorrek, Tommaso Piccini, Per Öberg, Michael Felsberg, Lars Nielsen, and Rudolf Mester. A multi-sensor traffic scene dataset with omnidirectional video. In *Ground Truth - What is a good dataset? CVPR Workshop 2013*, 2013.

[12] Kubota. Kubota. `http://www.kubota-global.net/news/2017/20170125.html`, 2017. Accessed: 2017-08-16.

[13] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.

[14] Kevin Matzen and Noah Snavely. Nyc3dcars: A dataset of 3d vehicles in geographic context. In *Proc. Int. Conf. on Computer Vision*, 2013.

[15] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *International Conference on Computer Vision (ICCV)*, 2017.

[16] Pix4D. Pix4D. `http://pix4d.com/`, 2014. Accessed: 2017-09-05.

[17] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. 2016.

[18] Udacity and Didi. Udacity Didi $100k Challenge Dataset 1.

[19] Inc. Udacity. Udacity Didi Challenge - Round 2 Dataset.

[20] Carl Vondrick, Donald Patterson, and Deva Ramanan. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, 101(1):184–204, 2013.

Paper 4

(DRAFT) Multi-modal Detection of Static and Dynamic Obstacles in Agriculture for Process Evaluation

Timo Korthals, Mikkel Kragh, Peter Christiansen, Henrik Karstoft, Rasmus N. Jørgensen, and Ulrich Rückert

Peer reviewed
In preparation. To be submitted on the 13.10.2017 to frontiers, Multi-modal Sensor Fusion, October 2017

Multi-modal Detection of Static and Dynamic Obstacles in Agriculture for Process Evaluation

Timo Korthals [1], **Mikkel Kragh** [2,*], **Peter Christiansen** [2], **Henrik Karstoft** [2], **Rasmus N. Jørgensen** [2], **and Ulrich Rückert**[1]

[1] *Cognitronics & Sensor Systems, Bielefeld University, Inspiration 1, D-33619 Bielefeld, Germany*
[2] *Department of Engineering, Aarhus University, Finlandsgade 22, DK-8200 Aarhus N, Denmark*

Correspondence*:
Mikkel Kragh
mkha@eng.au.dk

ABSTRACT

In recent years, autonomous robots and systems have been proposed for automating a number of agricultural tasks. Robots can improve workflow, minimize manual labor, and optimize yield. However, for unmanned autonomous vehicles to be certified, not only their specific agricultural tasks must be automated. An accurate and robust perception system automatically detecting and avoiding all obstacles must be in place in order to ensure safety of humans, animals, and other machines. In this paper, we present a multi-modal obstacle detection and recognition approach for process evaluation in agricultural fields. Obstacle detection algorithms are introduced for a variety of sensing modalities including lidar, radar, stereo camera, and thermal camera. Object information is fused across sensors and mapped globally, resulting in accurate traversability assessment and semantical mapping of process-relevant object categories (e.g. *grass*, *ground*, and *object* for mowing operations). Finally, a decoding step extracts relevant process-specific parameters along the trajectory of the vehicle, thus informing a potential control system of unexpected structures in the planned path. The method is evaluated on a public dataset for multi-modal obstacle detection in agricultural fields. Results show that a combination of multiple sensor modalities increases detection performance, and that different fusion strategies must be applied between inter- and intra-class detection algorithms.

Keywords: Occupancy grid maps, Obstacle detection, Precision Agriculture, Sensor Fusion, Multi-Modal Processing, Multi-Modal Perception, Inverse models

1 INTRODUCTION

Autonomous vehicles and robots operating in agricultural fields or orchards are emerging in both research and commercialized projects. The driverless systems must ensure safe operation by perceiving the environment and detecting and avoiding potential obstacles in their way. No sensor can single-handedly guarantee this safety, and thus a heterogeneous and redundant set of perception sensors and algorithms are needed for the purpose.

Contrary to self-driving cars whose primary purpose is to travel from A to B, an autonomous farming vehicle must also process the traversed area along its way. Common agricultural tasks are harvesting, mowing, pruning, seeding, and spraying. For these tasks, a simple representation of the environment into traversable and non-traversable areas is insufficient. Instead, an agricultural vehicle requires distinction between e.g. traversable areas like road and soil, and processable areas like grass, crops, and plants. Therefore, obstacle detection in an agricultural context does not simplify to purely identifying objects that protrude from a common ground plane. High grass or crop may appear non-traversable while actually being processable, whereas flat obstacles such as plant seedlings may appear traversable while being non-traversable. A need therefore exists for a system that can detect and recognize a large variety of object categories, while at the same time combine the extensive and perhaps unmanageable amount of information into process-specific parameters relevant for either the driver or an autonomous controller.

This paper presents a multi-modal obstacle detection and recognition approach for process evaluation in agricultural fields. Object detection algorithms are presented for lidar, radar, stereo camera, and thermal camera, individually. Object information from all sensors is mapped into a global 2D grid-based representation of the environment and fused across object categories, detection algorithms, and sensor modalities. Finally, relevant properties for processing the field such as traversability and yield information along planned trajectories are decoded. The proposed method is evaluated on a public grass mowing dataset recorded in Lem, Denmark, October 2016. The dataset includes both static and dynamic (moving) obstacles such as humans, vehicles, vegetation, barrels, and buildings.

The proposed architecture is depicted in Figure 1. A sensor platform is mounted on a tractor traversing a field along a preplanned trajectory. A number of exteroceptive sensors collect synchronized perception data used for object detection, whereas proprioceptive sensors are used for global localization of the vehicle. For each sensor modality, an inverse sensor model (ISM) includes an algorithm for detecting a number of object categories (e.g. *human*, *vegetation*, and *building*) and a mapping from raw sensor data to a 2D occupancy grid map (OGM) in the local sensor frame. In the fusion and mapping step, OGMs for all sensors and object categories are first localized globally and then updated temporally with the occupancy grid map algorithm. Finally, they are fused spatially to extract a global map of the environment. We present both binary (occupied/unoccupied) and semantical (object category-specific) maps, allowing further processing in subsequent algorithms. A final decoding step operates on the fused semantical maps to extract relevant process-specific (e.g. harvesting, mowing, or weed-spraying) parameters along the predefined trajectory of the vehicle. The final output could be used to alert a driver with human-understandable information, or directly by a control system for completely autonomous operation.

The paper is divided into 6 sections. Section 2 introduces related work on obstacle detection in agricultural applications. Section 3 presents the proposed method consisting of each of the four building blocks from Figure 1. Section 4 presents the experimental dataset and results for static and dynamic obstacle detection as well as decoding of process-relevant parameters. Section 5 provides a discussion of the overall approach, while section 6 concludes the paper and suggests future work.

2 RELATED WORK

Robotic automation is emerging for numerous agricultural tasks. The main objective is to reduce production costs and manual labour, while increasing yield and raising product quality (Luettel et al., 2012; Bechar and

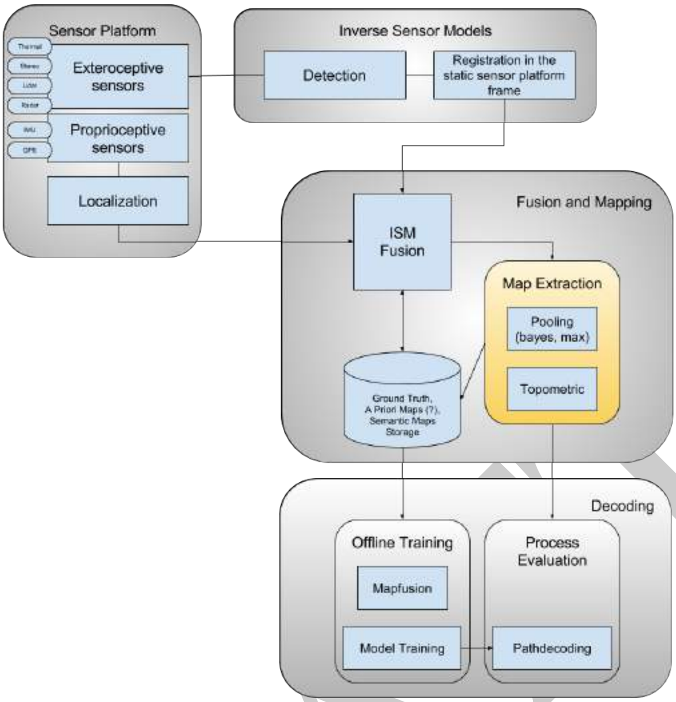


Figure 1. System architecture including information flow.

Vigneault, 2017). A significant milestone is to make robots navigate autonomously in dynamic, rough and unstructured environments, such as agricultural fields or orchards. To some extent, this has been possible for around two decades with automated steering systems utilizing global navigation systems (Abidine et al., 2004). To eliminate the need for a human operator, however, strict safety precautions are required including accurate and robust risk detection and obstacle avoidance.

Today, only small and harmless robots are commercially available that incorporate obstacle avoidance and operate fully autonomously in various agricultural domains (Lely, 2016; Harvest Automation, 2012). Commercialized self-driving tractors or harvesters, however, currently only exist as R&D projects (Case IH, 2016; ASI, 2016; Kubota, 2017).

In the literature, the concept of an autonomous farming vehicle with obstacle avoidance dates back to 1997 where a camera was used as an anomaly detector to identify structures different from crop (Ollis and Stentz, 1997). Since then, several systems have been proposed for detecting and avoiding obstacles (Cho and Lee, 2000; Stentz et al., 2002; Griepentrog et al., 2009; Moorehead et al., 2012; Emmi et al., 2014; Ball et al., 2016).

A simplified representation of the environment into traversable and non-traversable regions is common for autonomous navigation (Papadakis, 2013). A path may be non-traversable if it is blocked by obstacles, or if the terrain is too rough or steep. Similarly, anomaly or novelty detection is used to find anything that does not comply with normal appearance, and thus used to detect obstacles (Sofman et al., 2010; Ross et al., 2015; Christiansen et al., 2016a). However, for many agricultural tasks such as harvesting, mowing and weed spraying, further distinction between obstacles and traversable vegetation is necessary.

In one application, apparent obstacles such as crops or high grass may be traversable, whereas in another, small plants at ground level may represent obstacles and thus be non-traversable. Distinction into object, vegetation, and ground is common (Wellington and Stentz, 2004; Lalonde et al., 2006; Bradley et al., 2007; Kragh et al., 2015), whereas a few approaches explicitly recognize classes such as humans, vehicles and buildings (Yang and Noguchi, 2012; Christiansen et al., 2016b).

In the literature, obstacle detection systems often rely on a single sensor modality (Rovira-Mas et al., 2005; Reina and Milella, 2012; Fleischmann and Berns, 2015). These systems, however, are easily affected by varying weather and lighting conditions and thus present single points of failure. Therefore, a safety system must have a heterogeneous sensor suite with multiple sensing modalities that both overlap and complement each other in terms of detection capabilities and robustness. Sensor fusion is the concept of combining information from multiple sources to reduce uncertainty. Low-level fusion combines raw data from different sensors, whereas high-level fusion integrates information at decision level. In both cases, sensor data need to be compatible.

Lidar, radar, and stereo cameras are all range sensors providing metric 3D coordinates. Lidar and radar have been fused at low level using a joint extrinsic calibration procedure (Underwood et al., 2010) and at high level for augmented traversability assessment (Ahtiainen et al., 2015). Similarly, lidar and stereo camera have been fused at high level for traversability assessment (Reina et al., 2016). Often, a grid-based representation such as occupancy grid maps (Elfes, 1990) is used, allowing simple probabilistic fusion and subsequent path planning.

Monocular cameras operate in a non-metric pixel space and can be fused directly under assumption of negligible parallax errors. Examples are available of color and thermal camera fusion for object detection at both low level (Davis and Sharma, 2007) and high level (Apatean et al., 2010).

Fusion across domains is possible only when a well-defined transformation between them exists. By projecting 3D points onto corresponding 2D images, range sensors can be fused with cameras. With this approach, lidar and color cameras have been combined for semantic segmentation and object recognition both at low level (Dima et al., 2004; Wellington et al., 2005; Häselich et al., 2013) and high level (Laible et al., 2013; Kragh and Underwood, 2017). Similarly, image data in pixel-space have been transformed to metric 3D coordinates with inverse perspective mapping (Bertozzi and Broggi, 1998; Konrad et al., 2012). Here, a ground plane assumption is used to invert the perspective effect applied during image acquisition, such that image data are compatible with e.g. lidar and radar data.

In this paper, sensor data from both lidar, radar, stereo camera, and thermal camera are fused with a probabilistic 2D occupancy grid map. This data representation has been chosen, as it simplifies path planning and is already a standard in the automotive industry.

3 METHOD

In the following, each of the steps from the system architecture in Figure 1 are explained in detail.

3.1 Sensor Platform

The sensor suite presented by Kragh et al. (2017) was used to record multi-modal perception data. The dataset has recently been made publicly available. It includes lidar, radar, stereo camera, thermal camera, IMU, and GNSS[1]. The sensors were fixed to a common platform and interfaced to the Robot Operating

[1] Global Navigation Satellite System

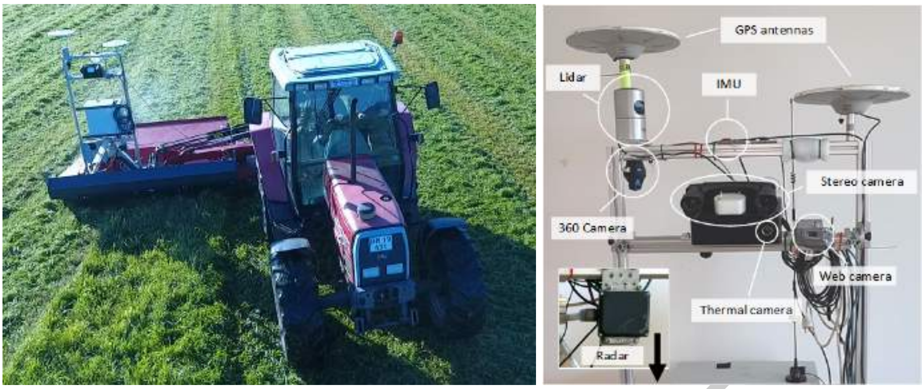


Figure 2. Recording platform. Reprinted from “FieldSAFE: Dataset for Obstacle Detection in Agriculture” by M. Kragh et al., 2017, arXiv preprint arXiv:1709.03526. Reprinted with permission.

System (ROS) (Quigley et al., 2009). A tractor-mounted setup and a close-up of the platform are shown in Figure 2.

The exteroceptive sensors and their properties are listed in Table 1. Proprioceptive sensors used for localization included a Vectornav VN-100 IMU and a Trimble BD982 dual antenna GNSS system.

Table 1. Sensors. Adapted from “FieldSAFE: Dataset for Obstacle Detection in Agriculture” by M. Kragh et al., 2017, arXiv preprint arXiv:1709.03526. Adapted with permission.

Sensor	Model	Resolution	FOV	Range	Data rate
Stereo camera	Multisense S21 CMV2000	1024 x 544	85°x 50°	1.5-50m	10 fps
Web camera	Logitech HD Pro C920	1920 x 1080	70°x 43°	-	20 fps
360-degree camera	Giroptic 360cam	2048 x 833	360°x 292°	-	30 fps
Thermal camera	Flir A65, 13 mm lens	640 x 512	45°x 37°	-	30 fps
Lidar	Velodyne HDL-32E	2172 x 32	360°x 40°	1-100 m	10 fps
Radar	Delphi ESR	32 targets/frame	90°x 4.2° 20°x 4.2°	0-60 m 0-174 m	20 fps

All sensors were synchronized in ROS using a best effort approach. Lidar, stereo camera, and thermal camera were registered before recording in a semi-automatic calibration procedure (Christiansen et al., 2017). All remaining sensors were registered by hand, by estimating extrinsic parameters of their positions.

Global localization from fused IMU and GNSS was obtained with the robot_localization package (Moore and Stouch, 2014) in the Robot Operating System (ROS) (Quigley et al., 2009).

3.2 Fusion and Mapping

Occupancy grid maps are used in static obstacle detection for robotic systems, which is a well-known and a commonly studied scientific field (Hähnel, 2004; Thrun et al., 2005; Stachniss, 2009). They are a component of almost all navigation and collision avoidance systems designed to maneuver through cluttered environments. Another important application is the creation of obstacle maps for traversing unknown areas and the recognizing known obstacles, thereby supporting localization. Recently, occupancy grid maps have been applied to combine LiDAR and RADAR in automotive applications with the goal of creating a harmonious, consistent, and complete representation of the vehicle's environment as a basis for advanced driver assistance systems (Garcia et al., 2008; Bouzouraa and Hofmann, 2010; Winner, 2015).

3.2.1 Occupancy Grid Mapping

Two-dimensional occupancy grid maps (OGM) were originally introduced by Elfes (1990). In this representation, the environment is subdivided into a regular array or a grid of quadratic cells. The resolution of the environment representation directly depends on the size of the cells. In addition to this compartmentalization of space, a probabilistic measure of occupancy is associated with each cell. This measure takes any real number in the interval $[0, 1]$ and describes one of the two possible cell states: unoccupied or occupied. An occupancy probability of 0 represents a space that is definitely unoccupied, and a probability of 1 represents a space that is definitely occupied. A value of 0.5 refers to an unknown state of occupancy.

An occupancy grid is an efficient approach for representing uncertainty, combining multiple sensor measurements at the decision level, and for incorporating different sensor models (Winner, 2015). To learn an occupancy grid M given sensor information z, different update rules exist (Hähnel, 2004). For the authors' approach, a Bayesian update rule is applied to every cell $m \in M$ at position (w, h) as follows: Given the position x_t of a vehicle at time t, let $x_{1:t} = x_1, \ldots, x_t$ be the positions of the vehicle's individual steps until t, and $z_{1:t} = z_1, \ldots, z_t$ the environmental perceptions. For each cell m of the occupancy probability grid represents the probability that this cell is occupied by an obstacle. Thus, occupancy probability grids seek to estimate

$$\mathrm{Odd}\left(P\left(m|z_t, x_t\right)\right) = \frac{P\left(m|z_t, x_t\right)}{1 - P\left(m|z_t, x_t\right)}, \quad P\left(m|z_{1:t}, x_{1:t}\right) = \mathrm{Odd}^{-1}\left(\prod_{t=1}^{T} \mathrm{Odd}\left(P\left(m|z_t, x_t\right)\right)\right) \tag{1}$$

This equation already describes the online capable, recursive update rule that populates the current measurement z_t to the grid, where $P\left(m|z_{1:t}, x_{1:t}\right)$ is the so called inverse sensor model (ISM). The ISM is used to update the OGM in a Bayesian framework, which deduces the occupancy probability of a cell, given the sensor information.

3.2.2 Extension to Agricultural Applications

The adaptation of OGM techniques to agricultural applications appears to be merely a matter of time but is not that obvious and intuitive to apply on the second sight. Robotic and automotive applications have in common that they both want to detect non-traversable areas or objects occupying their paths. Such unambiguous information is used to quantify the whole environment sufficiently for all derivable tasks such as path planning or obstacle avoidance. When assumptions like a flat operational plane or minimum obstacle heights are made, sensor frustums oriented parallel to the ground are sufficient for all tasks.

In agricultural applications, the crucial task is the quantification of the environment as the machines act on and process it. Therefore, quantification of the environment involves features such as processed areas, processability, crop quality, density, and maturity level in addition to traversability. In order to map these features, single occupancy grid maps are no longer sufficient. Instead, semantic occupancy grid maps that allow different classification results to be mapped are used. Furthermore, sensor frustums are no longer oriented parallel to the ground, but rather oriented at an angle to gather necessary crop information (Korthals et al., 2017b).

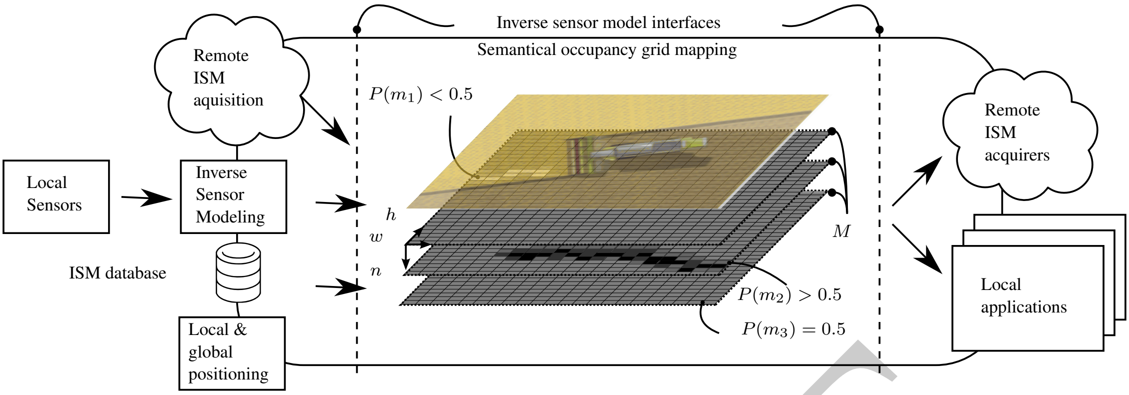


Figure 3. Semantical occupancy grid mapping framework

The extension to semantic occupancy grid maps (SOGM) or inference grids is straightforward and defined by an OGM M with W cells in width, H cells in height, and N semantic layers (see Figure 3):

$$M : \{1, \ldots, W\} \times \{1, \ldots, H\} \to m = (0, \ldots, 1)^N \tag{2}$$

Compared to a single layer OGM which allows the classification into three states $\{$occupied, $\overline{\text{occupied}}$, unknown$\}$, the SOGM supports a maximum of 3^N different states allowing much higher differentiability in environment and object recognition. The corresponding ISMs are fused by means of the occupancy grid algorithm to their nth associated semantical occupancy grid.

The location of information in the maps is required to be completed by *mapping under known poses* approaches (Thrun et al., 2005). As proposed by REP-105[2] and realized by Korthals et al. (2017b), information is mapped locally. The maps themselves are globally referenced which enables consistent storing and loading of information. Further, it allows smooth local mapping in the short term without discrete jumps caused by global positioning systems using a Global Navigation Satellite System (GNSS).

3.2.3 Mapping Capabilities

Requesting preprocessed SOGMs is beneficial for applications only requiring one kind of information which can be derived from a set of SOGMs. Therefore, besides the raw access of SOGMs, the two following fusion techniques among layers are mainly used for evaluation. The first approach is based on a super Bayesian independent opinion pooling P_{B} (Pathak et al., 2007). It is applicable for the case when separate SOGMs with identical feature representations (same object classes) are maintained. Second, a non-Bayesian maximum pooling fusion method P_{M} is applied to heterogeneous feature representations (varying object classes) (Liggins et al., 2001). The fusion techniques are cell-wise and therefore do not introduce any clustering.

$$P_{\text{B}}(m) = \frac{1}{1 + \prod_N \frac{1-P(m_n)}{P(m_n)}}, \; P_{\text{M}}(m) = \max_n P(m_n) \tag{3}$$

[2] http://www.ros.org/reps/rep-0105.html

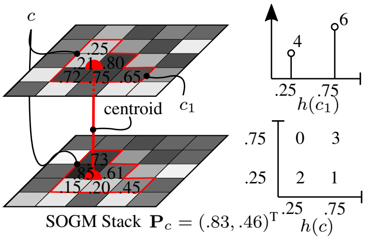


Figure 4. Supercell with $N = 2$ layers and corresponding histrograms with $K = 2$ bins.

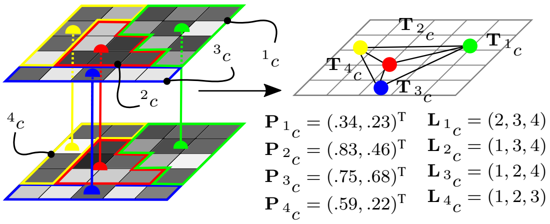

^3c
^1c
^2c
^4c
$\mathbf{T}_{^2c}$
$\mathbf{T}_{^1c}$
$\mathbf{T}_{^4c}$
$\mathbf{T}_{^3c}$
$\mathbf{P}_{^1c} = (.34, .23)^\mathrm{T}$ $\mathbf{L}_{^1c} = (2, 3, 4)$
$\mathbf{P}_{^2c} = (.83, .46)^\mathrm{T}$ $\mathbf{L}_{^2c} = (1, 3, 4)$
$\mathbf{P}_{^3c} = (.75, .68)^\mathrm{T}$ $\mathbf{L}_{^3c} = (1, 2, 4)$
$\mathbf{P}_{^4c} = (.59, .22)^\mathrm{T}$ $\mathbf{L}_{^4c} = (1, 2, 3)$

Figure 5. Conversion of supercells to a graph of centroids labeled with feature vectors.

Clustering on SOGMs was introduced by (Korthals et al., 2017a), using a Supercell Extracted Variance Driven Sampling (SEVDS) algorithm, which tends to find clusters that consist of mainly non-contradicting cells. Unlike single-layer OGM approaches, an SOGM incorporates multiple OGMs with varying classes residing in the map storage. For further applications, respecting every grid cell is not a feasible approach due to noise, sparse data, and potential offsets between layers. Even worse, requesting raw SOGMs would require impractical high bandwidths. To transform the SOGM into a parametrized form by clustering e.g. via Gaussians is unfeasible as well, due to the risk of lacking objects. The authors' approach is therefore a superpixel-like clustering method inspired by computer vision to find homogeneous regions and assign a feature vector for these. This leads to a topometric map, which is derived from the centroids of the superpixels as shown in Figure 5. Utilizing the Superpixels Extracted via Energy-Driven Sampling (SEEDS) algorithm from Stutz et al. (2016), we revise the formulation:

$$H(c) = D(c) + \gamma G(c) \text{ with } D(c) = \sum_{n=1}^{N} e_n(\mathrm{var}(h(c))) \tag{4}$$

to respect the nature, which is the probability and locality of information of SOGMs, more precisely. In Equation 4, c is the supercell of interest and G is the contour function which can be smoothed via the scalar factor γ. The distribution term D of a supercell c is defined as the sum of Eigenvalues e of the covariance matrix C of the probability histogram $h(c)$ (see Figure 4 and Figure 5).

As depicted in Figure 5, for every found supercell a tripel $\mathcal{C} = (\mathbf{T}_c, \mathbf{L}_c, \mathbf{P}_c)$ consisting of its centroid location $\mathbf{T}_c$, a list of adjunct supercell $\mathbf{L}_c$, and a feature vector $\mathbf{P}_c$ is calculated

$$\mathrm{Odd}\left(\mathbf{P}_c\right) = \left(\prod_{m \in c_1} \mathrm{Odd}(P(m)), \ldots, \prod_{m \in c_N} \mathrm{Odd}(P(m))\right)^\mathrm{T} \tag{5}$$

3.3 Inverse Sensor Models

In the following, individual inverse sensor models (ISM) are introduced and explained in detail for each of the sensors. An ISM consists of an algorithm for detecting a number of object categories and a mapping from raw sensor data to a 2D occupancy grid map (OGM) in the local sensor frame.

3.3.1 Camera

A Camera Inverse Sensor model comprises both a detection algorithm and a mapping of detections into a local grid map.

3.3.1.1 Detection algorithms

This section needs image examples of each algorithm A total of four detection algorithms for color camera have been used; Locally Decorrelated Channel Feature (LDCF) for pedestrian detection (Nam et al., 2014), an improved version of You Only Look Once (YOLOv2) (Redmon and Farhadi, 2016; Redmon et al., 2016) for object detection, a Fully Convolutional Neural Network (FCN) for semantic segmentation (Long et al., 2015) and DeepAnomaly (Christiansen et al., 2016a) for anomaly detection. Thermal camera uses a dynamic heat detection algorithm (HeatDetection) to detect hot objects based on a concept from (Christiansen et al., 2014).

LDCF is a pedestrian detection algorithm delimiting instances by a bounding box of a fixed aspect ratio. The detector is public available in a MATLAB-based framework (Dollar, 2015) by Piotr Dollár. The model from (Nam et al., 2014) is trained on the INRIA Person Dataset (**?**). In (Hansen et al., ????) the detector have been converted to C++ and wrapped in a ROS-package[3].

YOLOv2 is a deep learning based object detector delimiting instances by a bounding box of variable aspect ratio. The detector is developed in the deep learning framework Darknet (Redmon, 2013) and trained on ImageNet (Berg and Deng, 2015) and Microsoft COCO (Lin et al., 2014) for detecting 80 object categories. The contribution is a ROS-package[4] to run the algorithm in ROS and to perform a simple remapping of the 80 object classes into three classes (human, other and unknown).

FCN uses the backbone of VGG (Simonyan and Zisserman, 2014) to make a fully convolutional semantic segmentation algorithm that classifies all pixels in an image. The model developed in Caffe (Jia et al., 2014) and is publicly available[5]. One model is trained on the 59 most frequent classes of the Pascal Context dataset (Mottaghi et al., 2014). Unlike the more popular Pascal VOC dataset (Everingham et al., 2013) with only 20 object classes, Pascal Context provides full image annotations of 407 classes. In (Christiansen et al., 2016b) the 59 object classes are remapped to only 11 classes to investigate the semantic segmentation in an agricultural context. In (Hansen et al., ????) the detector have been wrapped in a ROS-package[6] and remapped. In this work, predictions are remapped to eight classes (human, Other, unknown, building, grass, ground, shelterbelt and water).

DeepAnomaly is a deep learning based detection algorithm for detecting anomalies (Christiansen et al., 2016a). The backbone is AlexNet (Krizhevsky et al., 2012) trained on ImageNet and the anomaly detector is modeled using 150 images from another field Christiansen et al. (2017). The output is coarse predictions of the whole image.

This section needs image examples of the heat detection algorithm HeatDetection is a simple heat detection principle from (Christiansen et al., 2014) for detecting hot objects using a thermal camera. The median temperature is determined for all image pixels, and the dynamic threshold is defined by some constant value above the median temperature. For a front facing camera on a tractor, the median temperature is determined for some bottom section of the image roughly corresponding to the ground

[3] ROS package available at https://github.com/PeteHeine/pedestrian_detector_ros.git

[4] ROS package available at https://github.com/PeteHeine/yolo_v2_ros

[5] Model is available at https://github.com/shelhamer/fcn.berkeleyvision.org

[6] ROS package available at https://github.com/PeteHeine/fcn8_ros

surface. Subtracting the image by the dynamic threshold and clipping values below zero, results in a heat map of how much each pixel have exceeded the dynamic threshold. Procedure is presented in Figure XX. A ROS-package is publicly available [7].

3.3.1.2 Camera Grid Mapping

Two methods have been used to transform detection information from camera-based algorithms using Occupancy grid maps also described in reference to 'Towards Inverse Sensor Mapping in Agriculture'. The code for transforming detection into local grid maps has been made publicly available as ROS packages[8,9]. **Inverse Perspective Mapping (IPM)** projects image from camera frame to the ground plane surface using a geometrical transformation (Bertozzi and Broggi, 1998; Konrad et al., 2012). The purpose of IPM is to remove/inverse the perspective effect by changing the viewpoint from camera to bird's eye view. The geometrical transformation for mapping image coordinates to surface is defined by intrinsic camera parameters, camera to surface transformation and surface to OGM transformation. The raw color image is transformed using an IPM in Figure 6. To generate an ISM, the IPM is used for transforming per-pixel predictions into an OGM. Figure 7 presents the predictions from FCN for the human and grass class before and after IPM.

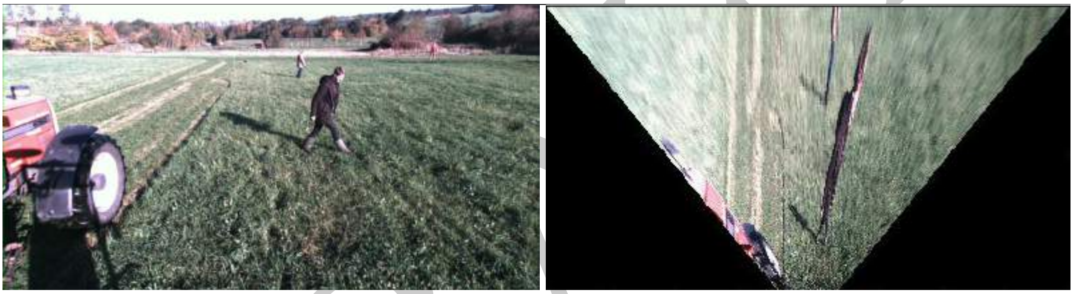


Figure 6. Inverse Perspective mapping of color image.

Areas outside the camera FOV is set to 0.5. Areas inside FOV with no detections are set below 0.5. Detections are given values above 0.5 to indicate that the area is expected to be occupied. As demonstrated for the grass class, the IPM algorithm approximates the actual inverse perspective mapping for flat elements on the surface. However, elements that are protruding or positioned above the ground surface are imprecisely mapped as demonstrated by the human detection in the bottom of Figure 7.

To handle protruding objects and bounding box outputs a second procedure is used, see Figure 8. Image bounding box detections are converted into 3D detections and projected to the ground surface with some localization uncertainty. Image bounding boxes are converted to 3D positions by estimating the distance and convention camera geometry. This work uses depth estimates from stereo matching. However, the distance can also be estimated by other depth sensors or by assuming a statically mounted camera above a flat surface.

Similar to the OGM from Figure 7, areas outside the FOV is set to 0.5 and areas inside the FOV with no detections are set below 0.5. Most detection algorithms degrade by the distance. To model this, the

[7] ROS package available at https://github.com/PeteHeine/dynamic_heat_detection

[8] ROS package available at https://github.com/PeteHeine/image_inverse_sensor_model2

[9] ROS package available at https://github.com/PeteHeine/image_boundingbox_to_3d

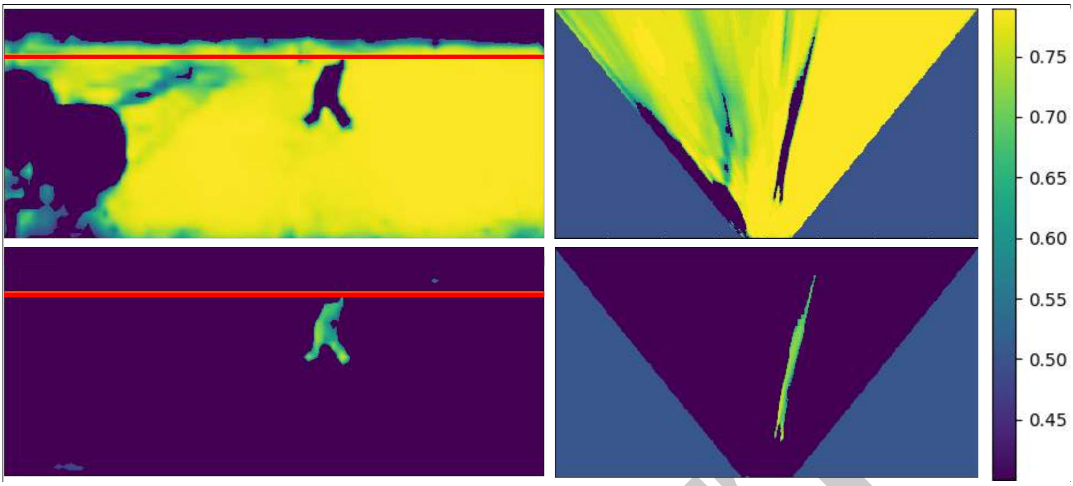


Figure 7. Inverse Perspective mapping of grass (top) and human detections (bottom).

uncertainty of detection is reduced linearly by the distance. In Figure 8 the probability increases from 0.4 to 0.5. Imprecise localization of a detection is modeled by a Gaussian distribution. For a camera the uncertainty of distance (radial coordinate) and angle (angular coordinate) to the object are independent. This is incorporated by modeling each polar coordinate (radial and angular) with independent uncertainties. In Figure 8 the localization uncertainty of the radial coordinate is larger than the angular coordinate.

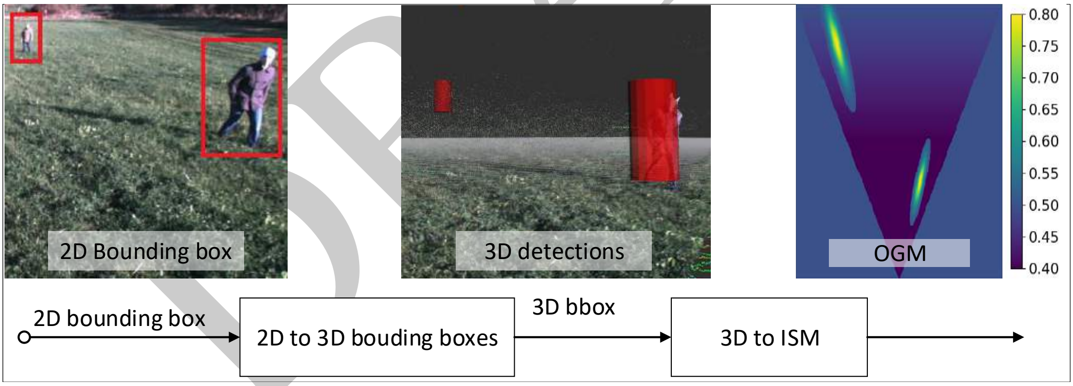


Figure 8. Converting detection to ISM

Human and Other predictions from FCN and the DeepAnomaly output is converted to bounding boxes using a connected components module 9. An IPM

Bounding boxes from YOLOv2 and LDCF is directly passed through the '2D bounding boxes to 3D'-module to estimate 3D positions of obstacles. These 3D positions are then converted to an OGM as illustrated. The localization uncertainty is modeled by an Gaussian distribution. The output of DeepAnomaly and FCN detection is converted to bounding boxes using the 'Connected Component'-module before being passed through the '2D bounding boxes to 3D'-module.

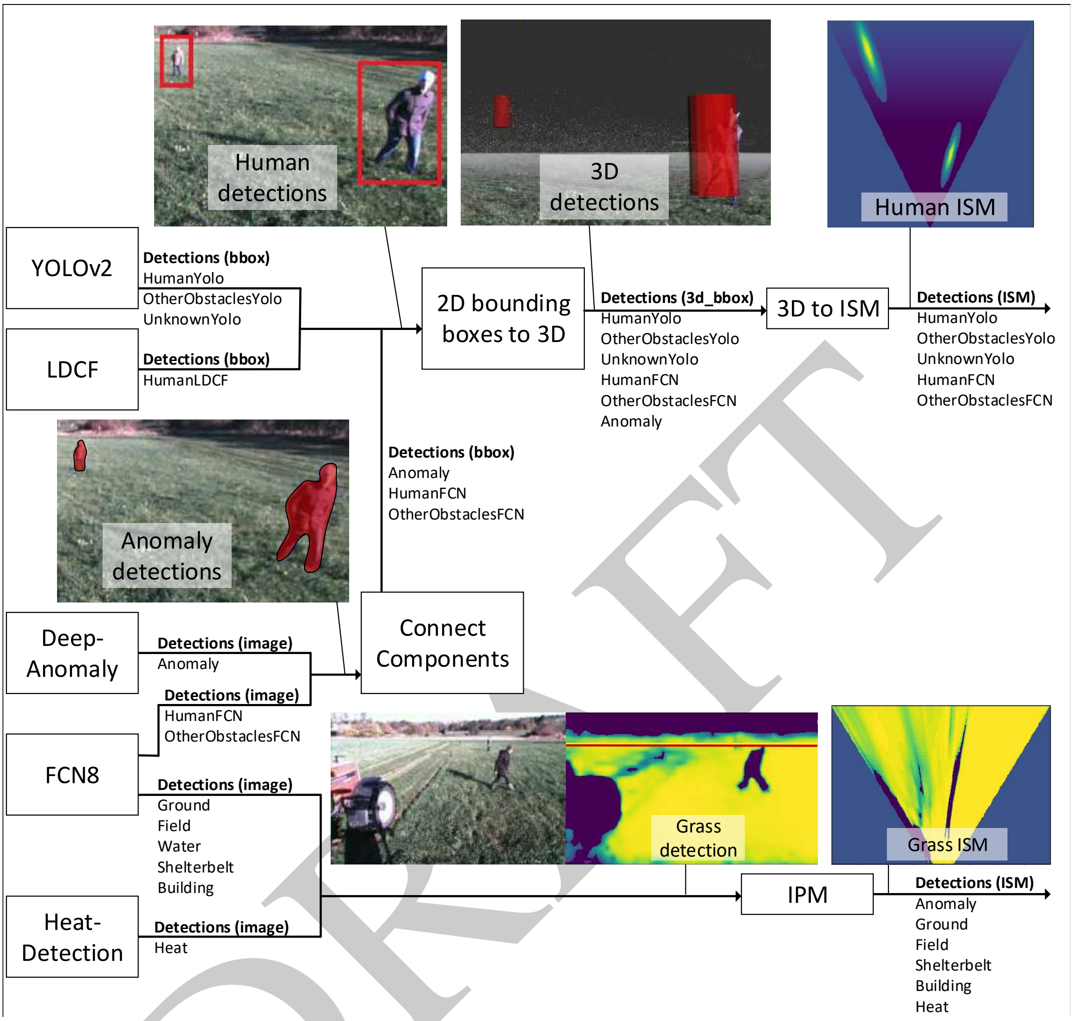


Figure 9. Converting detection to ISM

3.3.2 Lidar

The inverse sensor model for the lidar sensor consists of a detection algorithm and a mapping from raw sensor data to a local 2D grid in the vehicle frame.

The detection algorithm operates directly on 3D point clouds with approximately 70,000 points/frame generated at 10 fps by the Velodyne HDL-32E lidar. First, 13 features are calculated per point using neighborhood statistics that depend on local point densities (Kragh et al., 2015). Second, a Support Vector Machine (SVM) classifies each point as either *ground*, *vegetation*, or *object*. It further assigns probability estimates (Wu et al., 2004) to each class to describe the certainty of each classification. The SVM classifier was trained on the same data used in (Kragh et al., 2015).

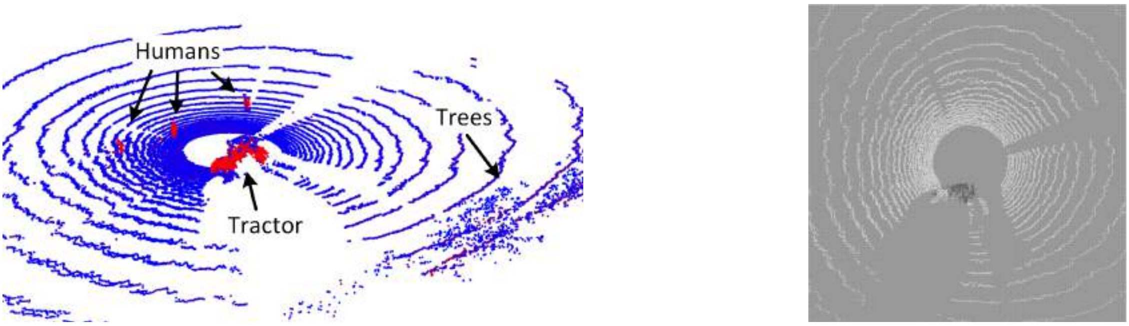


Figure 10. Left: point cloud with pseudo-colored probability estimates of the *object* class. Blue and red denote low and high probabilities, respectively. Right: corresponding OGM for the *object* class illustrating low (bright) and high (dark) probabilities.

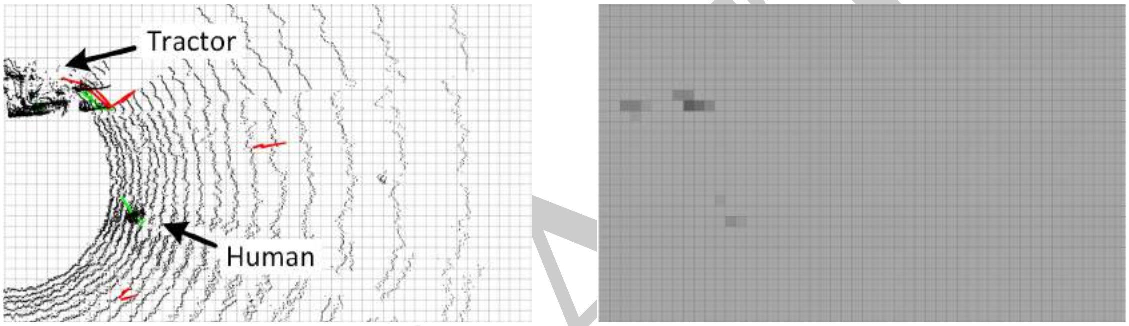


Figure 11. Left: radar detection example with confirmed (green) and unconfirmed (red) radar tracks overlaid on point cloud. Right: resulting radar OGM.

The mapping from detection probabilities to a local 2D grid is handled by projecting and resampling 3D points into 2D grid cells. For each 2D grid cell, class probabilities of all 3D points whose xy-projection lies inside are averaged and normalized such that the three class probabilities sum to 1. This results in three 2D probability grids: P^*_{object}, $P^*_{vegetation}$, and P^*_{ground}.

The three classes are combined into two OGMs by incorporating the *ground* probabilities into the *object* and *vegetation* classes probabilistically. For each grid cell m in an OGM, the log odds ratio of e.g. the *object* class is:

$$\begin{aligned}\text{logOdds}\left(P_{object}\left(m\right)\right) &= \text{logOdds}\left(P^*_{object}\left(m\right)\right) + \text{logOdds}\left(1 - P^*_{ground}\left(m\right)\right) \\ &= log\left(P^*_{object}\left(m\right)\right) - log\left(1 - P^*_{object}\left(m\right)\right) \\ &\quad + log\left(P^*_{ground}\left(m\right)\right) - log\left(1 - P^*_{ground}\left(m\right)\right)\end{aligned} \tag{6}$$

Figure 10 shows an example of a point cloud colored by *object* probabilities from the SVM classifier, and the corresponding *object* OGM.

3.3.3 Radar

The Delphi ESR automotive radar provides a list of up to 32 targets for each frame. Each target is represented by an angle, a range, and an amplitude. Most targets, however, represent internal noise in the radar and have low amplitudes. Simply filtering out these targets with a threshold eliminates radar returns from low-reflective objects such as humans and animals. Therefore, instead we use the approach from our previous paper (Kragh et al., 2016) and apply apply a tracking algorithm between subsequent frames known as the Kuhn-Munkres assignment algorithm (Munkres, 1957). Only radar targets that are less than $2m$ apart between two consecutive frames are associated. A track i is described by its current position and its track length L_i. It is confirmed when $L_i > L_{\text{min}} = 3$ and converted to a detection probability by:

$$P_{radar,i} = \frac{L_i - L_{\text{min}}}{L_i} \tag{7}$$

The mapping from detection probabilities to a local 2D grid is handled by converting from polar (angle and range) to cartesian (x, y) coordinates and resampling into 2D grid cells. For each 2D grid cell, class probabilities of all detections lying inside are averaged. This results in a 2D probability grid P^*_{radar}. Finally, the log odds ratio for each grid cell m in the radar OGM can be expressed as:

$$\text{logOdds}\left(P_{radar}\left(m\right)\right) = log\left(P^*_{radar}\left(m\right)\right) - log\left(1 - P^*_{radar}\left(m\right)\right) \tag{8}$$

Figure 11 shows an example of confirmed (green) and unconfirmed (red) radar tracks overlaid on the corresponding point cloud, as well as the resulting radar OGM.

3.4 Property Decoding along Trajectories

In preparation

4 EVALUATION

4.1 Data Set

The publicly available FieldSAFE dataset (Kragh et al., 2017) for multi-modal obstacle detection in agricultural fields is used for the evaluation.

4.2 Static Scenario

To quantify the detection of static obstacles and to compare it against the ground truth (GT) data from subsection 4.1, the mapserver maps all sensor information as ISMs at their corresponding global position. Afterwards, the resulting maps are stitched together as shown in Figure 12.

Two different evaluations have been performed: evaluation **A** for detecting occupied areas with respect to traverability and evaluation **B** for detecting process-relevant classes exclusively.

For evaluation **A**, GT labels were grouped into three different properties (*occupied*, $\overline{\textit{occupied}}$, and *unknown*) according to their traversability. The labels *Vegetation*, *Mannequin*, *Barrel*, *GPS Marker*, and *Other* were combined to the *occupied* property. The labels *Water* and *Building* were combined to the *unknown* property, as these categories where not visible during the used data sequence. All remaining classes were combined to the $\overline{\textit{occupied}}$ property.

Figure 12. Evaluation pipeline from static recording to evaluation with stitching

For evaluation **B**, GT labels were grouped into four different process-relevant classes (*Vulnerable obstacles*, *Processable*, *Traversable*, and *Non-traversable*). The *Vulnerable obstacles* class included GT label *Mannequin* and covers regions with which a collision needs to be avoided under any circumstance. The *Processable* class included GT label *grass* and represents the goods. The *Traversable* class included GT labels *Grass*, *Road*, and *Ground* and represents areas that can be traversed by the vehicle. Finally, the *Non-traversable* class included GT label *Vegetation* and represents areas that must be avoided to not harm the vehicle. For evaluating the process-relevant detection, each of the four classes was considered in its own property map. Included GT classes were marked as *occupied*, whereas all other classes were treated as *unknown*.

The resulting tri-state maps from GT data and mapping were compared tile-wise against each other, such that for the whole recorded map the true-positives (TP), false-positives (FP), and false-negatives (FN) could be calculated:

- TP $= \sum_{m \in \text{tiles}} m_{\text{GT}} = \textit{occupied} \wedge m_{\text{Mapped}} = \textit{occupied}$
- FP $= \sum_{m \in \text{tiles}} m_{\text{GT}} = \overline{\textit{occupied}} \wedge m_{\text{Mapped}} = \textit{occupied}$
- FN $= \sum_{m \in \text{tiles}} m_{\text{GT}} = \textit{occupied} \wedge m_{\text{Mapped}} = \overline{\textit{occupied}}$

The Precision, Recall, F$_1$ score, and entropy H were calculated as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{FP} + \text{TP}}, \text{Recall} = \frac{\text{TP}}{\text{FN} + \text{TP}}, \text{F}_1 = 2\frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}, \tag{9}$$

$$H(P(M)) = -\sum_{c \in M} P(c) \log P(c) + (1 - P(c) \log(1 - P(c)). \tag{10}$$

Table 2 show the results of evaluation **A**, i.e. detecting occupied areas with respect to traversability. The first column shows individual detection results for each of the algorithms. These are grouped by object categories such that different algorithms from different sensors that detect similar classes are grouped together. In the second column, algorithms from each group of categories are fused with competitive, Bayesian fusion. For classifiers detecting the same object classes, competitive fusion increases the precision while maintaining information gain (entropy). In the third column, detections from all sensors (and algorithms) are fused with complementary, max-pooling fusion. For classifier detecting different object classes, complementary fusion increases recall while maintaining precision. In practice, this results in a more complete detection of the environment.

Table 3 shows the results of evaluation **B**, i.e. detecting process-relevant classes exclusively. Here, both competitive, Bayesian fusion and complementary, max-pooling fusion were applied for all fusion scenarios.

Figure 13. Examples for different mapping results in the different cases

Table 2. Evaluation **A**. Traversability assessment of static obstacles for single classifiers, classifier combinations, and sensor combinations.

	Single evaluation				Bayesion fusion among classifiers				Max-pooling among classifiers			
Classifier	F_1	Prec.	Rec.	Ent.	F_1	Prec.	Rec.	Ent.	F_1	Prec.	Rec.	Ent.
Cam-FCN-human	3.8	25.3	2.1	75.6	13.0	67.38	7.2	89.2	88.8	88.3	89.4	92.5
Cam-ped-human	0.7	3.7	0.4	83.2								
Cam-yolo-human	1.2	6.8	0.7	75.5								
RADAR-tracking	2.6	3.5	2.1	15.9								
thermal-detection	7.3	16.6	4.7	88.6								
LiDAR-SVM-object	7.8	66.8	4.1	89.7								
Cam-FCN-other	4.1	30.8	2.2	76.3	22.3	72.3	13.2	89.5				
Cam-yolo-other	2.0	3.9	1.3	75.6								
Cam-deepanomaly	2.0	3.8	1.4	75.6								
RADAR-tracking		.										
LiDAR-SVM-object		.										
LiDAR-SVM-vegetation	83.5	81.4	85.8	87.9	84.6	88.3	81.6	92.3				
Cam-FCN-shelterbelt	46.7	32.2	84.4	81.2								

4.3 Dynamic Scenario

4.3.1 Evaluation Implementation

To quantify the tracking of dynamic obstacles and to compare it against the GT data from subsection 4.1, the mapserver is commonly applied, but for available time stamp in the ground truth data, the content is extracted. In the ongoing evaluation steps, the content is then processed, such that different fusion and clustering parameters are applied which results are compared to the corresponding GT human positions, as illustrated in Figure 14.

4.3.2 Two-Class Prediction Measure Evaluation

To quantify the detection rate and quality, first the different mapserver layer are fused. The resulting tri-state (s.t. occupied, non-occupied, unknown) likelihood map is first processed, such that the occupied classified cells are clustered via 8-connected clustering. The resulting clusters which are under a specific minimum size are sorted out to prune noisy readings. At last, the GT positions were applied for every time stamp t as follows to calculate the true-positives (TP), false-positives (FP), and false-negatives (FN):

- $TP_t = TP_t + 1$ if a GT position is inside any cluster
- $FP_t = FP_t + 1$ if a cluster does not contain any GT position
- $FN_t = FN_t + 1$ if a GT position is inside the detection range, but not in any cluster

Table 3. Evaluation **B**. Process-relevant object detection for single classifiers, classifier combinations, and sensor combinations.

	Single evaluation			Fusion among classifiers				Fusion among sensors			
Classifier	F_1	Prec.	Rec.	Fusion	F_1	Prec.	Rec.	Fusion	F_1	Prec.	Rec.
Vulnerable Obstacles (Manuequin)											
Cam-ped-human	1.3	0.7	25.9	max.	3.2	1.6	73.4				
Cam-FCN-human	3.4	1.7	73.6	bay.	12.6	7.1	57.4				
Cam-yolo-human	11.7	6.9	36.1								
Processable (Grass)											
Cam-FCN-grass	85.2	94.2	77.8								
Traversable (Grass & Road & Ground)											
Cam-FCN-grass	83.4	96.3	73.6	max.	84.6	96.0	75.6	max.	90.1	89.2	91.0
Cam-FCN-ground	24.0	96.8	13.7	bay.	82.0	97.2	71.0	bay.	87.7	90.8	84.8
LiDAR-SVM-ground	89.7	89.4	90.1								
Non-Traversable (Vegetation)											
LiDAR-SVM-vegetation	83.6	81.4	86.0	max.	51.0	35.2	92.5				
Cam-FCN-shelterbelt	46.6	32.2	84.7	bay.	53.1	37.5	91.0				

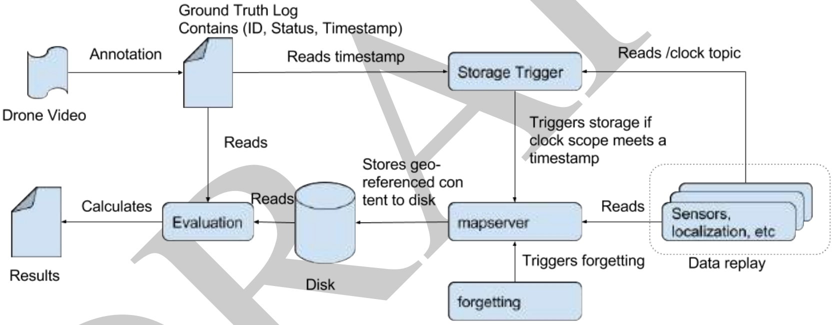


Figure 14. Evaluation pipeline which gets the drone video and recorded data as input to process the results.

True-negatives (TN) cannot be sufficiently measured while these can only alter between 0 and 1 for every step. This comes from the fact that the fused area is consistently detected like shown in Figure 15. Thus, only the two states “no GT position is inside the non-occupied area” or “any GT position is inside the non-occupied area” can be evaluated. Therefore, the F_1 score calculated as a quantitative measure as follows:

$$\text{Recall} = \frac{\sum_t TP_t}{\sum_t FN_t + TP_t}, \text{Precision} = \frac{\sum_t TP_t}{\sum_t FP_t + TP_t}, F_1 = 2\frac{\text{Recall Precision}}{\text{Recall} + \text{Precision}} \tag{11}$$

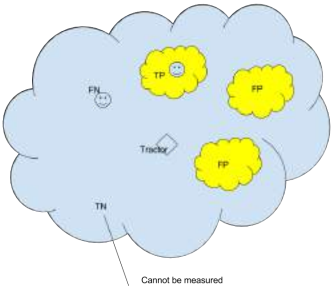


Figure 15. Evaluation example for true-positive (TP), false-positive (FP), and false-negative (FN) acquisition. True-negative (TN) cannot be measured sufficiently.

4.3.3 Results

For comparison results the best sensor/classifier/parameter setup is evaluated in a greedy fashion. Parameters to probe are:

- Sensors/classifiers combinations
- Fusion techniques among classifiers
- Forgetting rate and amplitude
- Minimum cluster size threshold (suppressing noise)
- Dilating factors (to see if at least a person was in the range of a detection)

The different sensor setups are listed in Table 4 from which every unique setup was chosen as initial evaluation. Further, to have a quite independent evaluation of the mapping capabilities, a high forgetting rate (6) and low forgetting value (0.2) was chosen, to evaluate the F-Score over minimal valid cluster sizes (cf. Figure 16). A minimal cluster size needs to be evaluated so that on one hand noisy sensor readings are filtered out but on the other hand, small detection footprints of a person are still valid detections for further evaluation. With robotic navigation and planning algorithms in mind, every valid cluster was dilated by the vehicles radius to ensure safe environmental traversal. Consequently, dilation leads to better scores, since detection and mapping may always be a bit off. Therefore, it can be assumed that a false-negative detection is at least in the range of a false-positive detection. The fact that the F-Score increases for no dilation, and decreases with dilation over the minimal cluster size confirms that statement even more. An already exact sensor like the LiDAR is qualitatively independent of dilation. Setup 5 and 6, which are FCN and PED performing so bad, that they are excluded from further evaluation to not taint the results

Table 5 depicts the result of the fused sensor setup at a certain minimal cluster size. It reveals that for setup 2 (camera-based detection) Bayesian fusion performs best, while on the overall setup, the max pooling surpasses. This behavior becomes obvious when looking at the actual modalities of the sensors. While using just the camera, competitive fusion in terms of the Bayesian formulation leads to a more precise and accurate detection. On the other hand, acquiring information from all different kind of sensors, a complementary fusion in terms of the max pooling results in a more complete detection of the environment. This can be pointed out by investigating the Precision and Recall of Table 5. For setup 1, almost no false-negative (which is a non-detected person in the sensors frustrum) detections were counted, which results in nearly perfect Recall, while on the other hand, the Precision is just as good as in setup 1.

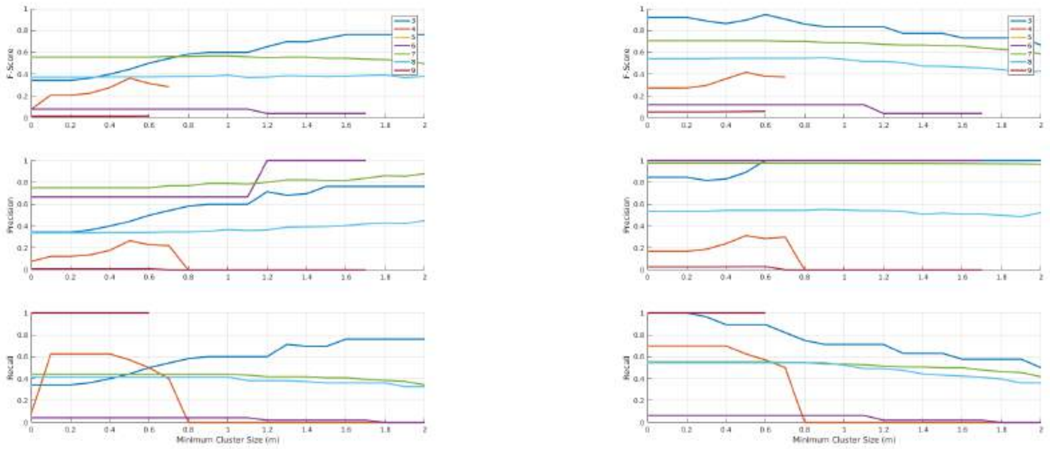


Figure 16. F-Score over increasing minimum cluster size. Left: No dilation. Right: Dilation by vehicle radius of $2.5\,\mathrm{m}$

.

Evaluating dynamic detection problems is actually not satisfied by occupancy grid mapping techniques. This issue is faced with a forgetting technique which is salable in two conditioning ways: First, the *ForgetValue* indicates the steadiness of the fused maps. A value of 1 indicates no forgetting, such that all information remains on the map like in the static case and can only be altered by the detections themselves. On the other hand, 0 indicates total forgetting, so that the maps are cleared immediately. All values in between impact the information accordingly. Second, the *ForgetRate* indicates how often per second the *ForgetValue* is applied to the maps. Figure 17 shows the impact of different forgetting rates (1 – 6) over an increasing forgetting value (0.1 – 0.9). It can be seen that the all scores are drastically influenced by the value and rate. On the first sight, the scores develop counter-intuitive, while everything becomes better with increasing forgetting (low *ForgetValue*). But in fact, the fusion becomes agiler due to this, and thus new temporary information can be taken into account much better. Further, the figure reveals an increase in Precision with increasing *ForgetRate*. This can be explained by the fact, that less false-positive detections occur. The Recall remains almost constant, due to max pooling which results in a very comprehensive detection of the surrounding. The *ForgetValue* limits indicate for all rates a bad progress. For high rates and low values, the score becomes even worse because more and more detections are going to be pruned before they are evaluated. For high values, the overall setup approaches the static mapping case.

Table 4. Listing of setups and the detection algorithms they comprise.

Class	object	heat	object	objects/human	human	human	anomaly
Algorithm	detection	dynamic heat	SVM	fcn8	ped	yolo	deep anomaly
	9 (Radar)	3 (IR)	4 (LiDAR)	5	6	7	8
Setup				2 (Camera)			
	1						

Table 5. Sensor fusion of setup 1 and 2 for minimal cluster size equals $0.5\,\mathrm{m}$

Setup	Fusion	F (%)	Precision (%)	Recall (%)
1	Max	70.81	57.23	92.86
	Bayes	42.58	39.76	45.83
2	Max	57.32	51.14	65.22
	Bayes	61.22	56.96	66.18

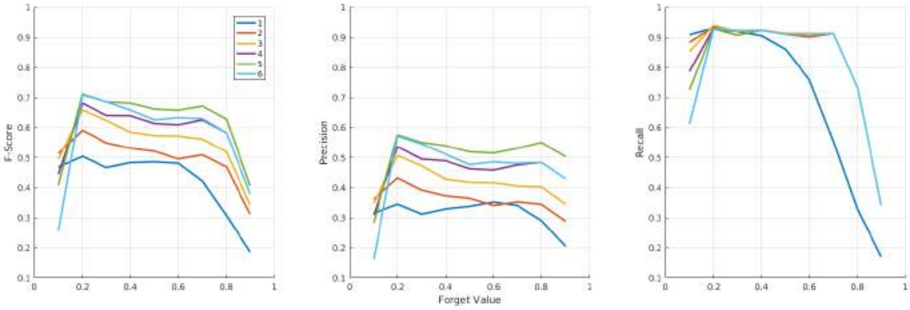


Figure 17. F-Score over increasing *ForgetValue* with different *ForgetRate*.

4.4 Trajectory Decoding

In preparation

5 DISCUSSION

The proposed architecture describes a sensor data processing pipeline from the acquisition to the process relevant detection of properties along the path. Since there exists no compatible baseline architecture known to the authors, this work describes a novelty in the field of research concerning multi-modal acquisition, mapping, and evaluation in agriculture of unstructured environments like grass or corn fields which target high yield throughput. However, approaches in structured environments for sweet pepper or cabbage do exist but are still not applicable to the marked due to impracticability, high setup up costs, or missing infrastructure. Our approach extends agriculture technology without replacing current work habits, but yet allows the incorporation of state-of-the-art algorithms for a very differentiated environment detection via an efficient mapping approach. Furthermore, it allows the easy changeability and extendability which is actually wanted and needed in a daily agriculture scenario.

In comparison to model-based or parametrized approaches, our non-parametric 2-dimensional occupancy grid mapping represents an optimal approach in agriculture scenarios, where mainly the vegetated area is of interest. We have applied analytically solutions as well as several heuristics to build the inverse sensor models (ISM) which incorporate the sensor information as well as its localization. We mostly have respected sensor and localization noise in these models, but taking the whole error chain into account, which reaches from detection and localization noise over sensor registration up until to ground-truth errors, makes the quantification of error propagation practically unfeasible. However, the extraction of human hypotheses based on the fusion of multiple semantical occupancy grid layers reaches an F_1 score of over $70\,\%$ which comprises the detection as well as the localization of humans. The recall of over $92\,\%$, as well as the high F_1 scores for single classifiers in the dilated case, reveals the actual classifiers capabilities, so that either an optimized localization, model-based approach, or tracking could increase our results.

The semantic occupancy mapping technique itself has been widely discussed. However, it still demands the proper combination of information between different layers. It is worth noticing that the hierarchical fusion of first the competitive combination of modalities which are similar to each other, and secondly a complementary combination among these is the only reasonable strategy which leads to an increase of the F_1 score. While with each additive layer this hierarchical combination needs to be redefined and therefore, a trained combination via, for instance boosting, could lead to adaptive and better results. Furthermore, our mapping technique is very prone to miss-classification, which was for example caused by sun blinded cameras, and systematic errors. To address the second case, we have for instance applied blind spots at the location of the tractor so that the mapping of self-classification, heavily caused by the RADAR, could be overcome. However, with our proposed architecture pipeline and information processing we have shown that with each combination of classifiers, an overall increase of the F_1 score can be reached With up to $88.8\,\%$ in a $10\,\mathrm{cm}$ cell-wise, globally mapped evaluation for obstacle scenarios, as well as comparable results for semantical classes which are meaningful in for the grass mowing process, our approach represents a state-of-the-art solution for environment classification in agriculture scenarios.

6 CONCLUSION AND FUTURE WORK

In this work, we have presented an information processing architecture for global mapping and process evaluation in an agricultural grass mowing scenario. The introduced architecture consists of four relevant processing steps: First, the sensor platform which comprises all applied sensors for localization and environment data acquisition, such as a stereo vision camera, a RADAR, a LIDAR, and a thermal camera. Second, the inverse sensor models (ISMs) that describe the sensor' data processing for detecting and localizing of process relevant properties and objects in the environment, like “Grass”, “Vegetation”, and “Humans”. The ISMs are 2D grid-based, parametric free representations of the detections' outputs, which were referenced and fused, based on the occupancy grid mapping algorithm into a semantical occupancy grid map (SOGM) stack, in the third step. In the fourth step, we applied a Hidden Markov Model based approach to first train and then quantify the environment along the vehicle's trajectory to reveal process relevant information out of the SOGMs.

To evaluate the capabilities of the mapping approach, we have compared the mapping and fusion of ISMs in a static and dynamic scenario against the FieldSAFE dataset. For the fusion among SOGMs in the static case, we have achieved improving results in detection and localization of environmental properties through a first stage of competitive fusion among similar modalities, and a second stage of complementary fusion among different ones. For detecting humans in the dynamic evaluation, we only have taken classifiers into account that were able to detect corresponding modalities which were fused accordingly. Further, we have improved the SOGMs with a forgetting capability to adapt the mapping approach to a dynamic environment on which we have applied a grid cell clustering to get consistent human hypotheses. All steps were evaluated and improved to get a sufficient score where again, a combination of multiple sensors leads to an overall improvement in detection.

In our future work, we want to incorporate geodata acquired by satellites, drones, or planes from which we directly derive process relevant information into the detection pipeline. This approach will overcome issues like complex sensor registration, weather conditions, or false detection for all static properties and objects in the environment and will, therefore improve and harden our setup.

CONFLICT OF INTEREST STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

AUTHOR CONTRIBUTIONS

The Author Contributions section is mandatory for all articles, including articles by sole authors. If an appropriate statement is not provided on submission, a standard one will be inserted during the production process. The Author Contributions statement must describe the contributions of individual authors referred to by their initials and, in doing so, all authors agree to be accountable for the content of the work. Please see here for full authorship criteria.

FUNDING

Details of all funding sources should be provided, including grant numbers if applicable. Please ensure to add all necessary funding information, as after publication this is no longer possible.

ACKNOWLEDGMENTS

This research is sponsored by the Innovation Fund Denmark as part of the project “SAFE-Safer Autonomous Farming Equipment” (Project No. 16-2014-0). This research/work was supported by the Cluster of Excellence Cognitive Interaction Technology “CITEC” (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG) and by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster “Intelligent Technical Systems OstWestfalenLippe” (it’s OWL) and managed by the Project Management Agency Karlsruhe (PTKA). We acknowledge support for the Article Processing Charge by the Deutsche Forschungsgemeinschaft and the Open Access Publication Fund of Bielefeld University.

SUPPLEMENTAL DATA

Supplementary Material should be uploaded separately on submission, if there are Supplementary Figures, please include the caption in the same file as the figure. LaTeX Supplementary Material templates can be found in the Frontiers LaTeX folder

REFERENCES

Abidine, A. Z., Heidman, B. C., Upadhyaya, S. K., and Hills, D. J. (2004). Autoguidance system operated at high speed causes almost no tomato damage. *California Agriculture* 58, 44–47. doi:10.3733/ca.v058n01p44

Ahtiainen, J., Peynot, T., Saarinen, J., Scheding, S., and Visala, A. (2015). Learned Ultra-Wideband RADAR Sensor Model for Augmented LIDAR-based Traversability Mapping in Vegetated Environments , 953–960

Apatean, A., Rusu, C., Rogozan, A., and Bensrhair, A. (2010). Visible-infrared fusion in the frame of an obstacle recognition system. In *Automation Quality and Testing Robotics (AQTR), 2010 IEEE International Conference on* (IEEE), vol. 1, 1–6

ASI (2016). Autonomous Solutions. `https://www.asirobots.com/farming/`. Accessed: 2017-08-09

Ball, D., Upcroft, B., Wyeth, G., Corke, P., English, A., Ross, P., et al. (2016). Vision-based Obstacle Detection and Navigation for an Agricultural Robot. *Journal of Field Robotics* 33, 1107–1130. doi:10.1002/rob.21644

Bechar, A. and Vigneault, C. (2017). Agricultural robots for field operations. Part 2: Operations and systems. *Biosystems Engineering* 153, 110–128. doi:10.1016/j.biosystemseng.2016.11.004

Berg, A. and Deng, J. (2015). Imagenet large scale visual recognition challenge 2015. *Challenge*

Bertozzi, M. and Broggi, A. (1998). GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Trans. Image Process.* 7, 62–81

Bouzouraa, M. E. and Hofmann, U. (2010). Fusion of occupancy grid mapping and model based object tracking for driver assistance systems using laser and radar sensors. In *2010 IEEE Intelligent Vehicles Symposium*. 294–300. doi:10.1109/IVS.2010.5548106

Bradley, D. M., Unnikrishnan, R., and Bagnell, J. (2007). Vegetation detection for driving in complex environments. In *Robotics and Automation, 2007 IEEE International Conference on* (IEEE), 503–508

Case IH (2016). Case IH Autonomous Concept Vehicle. `http://www.caseih.com/apac/en-in/news/pages/2016-case-ih-premieres-concept-vehicle-at-farm-progress-show.aspx`. Accessed: 2017-08-09

Cho, S. I. and Lee, J. H. (2000). Autonomous speedsprayer using differential global positioning system, genetic algorithm and fuzzy control. *Journal of Agricultural Engineering Research* 76, 111–119. doi:10.1006/jaer.1999.0503

Christiansen, P., Kragh, M., Steen, K. A., Karstoft, H., and Jørgensen, R. N. (2017). Platform for evaluating sensors and human detection in autonomous mowing operations. *Precision Agriculture* 18, 350–365. doi:10.1007/s11119-017-9497-6

Christiansen, P., Nielsen, L. N., Steen, K. A., Jørgensen, R. N., and Karstoft, H. (2016a). Deepanomaly: Combining background subtraction and deep learning for detecting obstacles and anomalies in an agricultural field. *Sensors* 16, 1904

Christiansen, P., Sørensen, R., Skovsen, S., Jæger, C. D., Jørgensen, R. N., Karstoft, H., et al. (2016b). Towards autonomous plant production using fully convolutional neural networks. In *International Conference on Agricultural Engineering 2016*

Christiansen, P., Steen, K., Jørgensen, R., and Karstoft, H. (2014). Automated detection and recognition of wildlife using thermal cameras. *Sensors* 14, 13778–13793

Davis, J. W. and Sharma, V. (2007). Background-subtraction using contour-based fusion of thermal and visible imagery. *Computer Vision and Image Understanding* 106, 162 – 182. doi:http://dx.doi.org/10.1016/j.cviu.2006.06.010. Special issue on Advances in Vision Algorithms and Systems beyond the Visible Spectrum

Dima, C., Vandapel, N., and Hebert, M. (2004). Classifier fusion for outdoor obstacle detection. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004* (IEEE), vol. 1, 665–671 Vol.1. doi:10.1109/ROBOT.2004.1307225

Dollar, P. (2015). Piotr's computer vision matlab toolbox

Elfes, A. (1990). Occupancy grids: A stochastic spatial representation for active robot perception. In *Proceedings of the Sixth Conference on Uncertainty in AI*. vol. 2929

Emmi, L., Gonzalez-De-Soto, M., Pajares, G., and Gonzalez-De-Santos, P. (2014). New trends in robotics for agriculture: Integration and assessment of a real fleet of robots. *The Scientific World Journal* 2014. doi:10.1155/2014/404059

Everingham, M., Eslami, S., and Gool, L. V. (2013). The pascal visual object classes challenge–a retrospective. *Homepages.Inf.Ed.Ac.Uk*

Fleischmann, P. and Berns, K. (2015). A Stereo Vision Based Obstacle Detection System for Agricultural Applications. *Field and Service Robotics* , 1–14

Garcia, R., Aycard, O., and Vu, T.-d. (2008). High Level Sensor Data Fusion for Automotive Applications using Occupancy Grids , 17–20

Griepentrog, H. W., Andersen, N. A., Andersen, J. C., Blanke, M., andT.E. Madsen, O. H., Nielsen, J., et al. (2009). Safe and reliable: further development of a field robot. *Precision agriculture '09* , 857–866

Hähnel, D. (2004). *Mapping with Mobile Robots*. Ph.D. thesis, University of Freiburg

Hansen, M. K., Christiansen, P., Korthals, T., Jungeblut, T., Karstoft, H., and Jørgensen, R. N. (????). Multi-modal obstacle detection and evaluation of evidence grid mapping in agriculture

Harvest Automation (2012). HV-100. `https://www.public.harvestai.com`. Accessed: 2017-08-09

Häselich, M., Arends, M., Wojke, N., Neuhaus, F., and Paulus, D. (2013). Probabilistic terrain classification in unstructured environments. *Robotics and Autonomous Systems* 61, 1051–1059. doi:10.1016/j.robot.2012.08.002

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., et al. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia* (New York, NY, USA: ACM), MM '14, 675–678

Konrad, M., Nuss, D., and Dietmayer, K. (2012). Localization in digital maps for road course estimation using grid maps. In *2012 IEEE Intelligent Vehicles Symposium*. 87–92

Korthals, T., Exner, J., Schöpping, T., and Hesse, M. (2017a). Semantical Occupancy Grid Mapping Framework. In *European Conference on Mobile Robotics* (IEEE)

Korthals, T., Kragh, M., Christiansen, P., and Rückert, U. (2017b). Towards Inverse Sensor Mapping in Agriculture. In *IROS 2017 Workshop on Agricultural Robotics: learning from Industry 4.0 and moving into the future* (Vancouver)

Kragh, M., Christiansen, P., Korthals, T., Jungeblut, T., Karstoft, H., and Jørgensen, R. N. (2016). Multi-Modal Obstacle Detection and Evaluation of Occupancy Grid Mapping in Agriculture. In *International Conference on Agricultural Engineering* (International Commission of Agricultural and Biosystems Engineering)

Kragh, M., Jørgensen, R. N., and Pedersen, H. (2015). Object Detection and Terrain Classification in Agricultural Fields Using 3D Lidar Data. In *Computer Vision Systems : 10th International Conference, ICVS 2015, Proceedings*, vol. 9163. 188–197. doi:10.1007/978-3-319-20904-3_18

Kragh, M. and Underwood, J. (2017). Multi-modal obstacle detection in unstructured environments with conditional random fields. *CoRR* abs/1706.02908

Kragh, M. F., Christiansen, P., Laursen, M. S., Larsen, M., Steen, K. A., Green, O., et al. (2017). Fieldsafe: Dataset for obstacle detection in agriculture. *arXiv preprint arXiv:1709.03526*

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* , 1097–1105

Kubota (2017). Kubota. `http://www.kubota-global.net/news/2017/20170125.html`. Accessed: 2017-08-16

Laible, S., Khan, Y. N., and Zell, A. (2013). Terrain classification with conditional random fields on fused 3D LIDAR and camera data. In *2013 European Conference on Mobile Robots* (IEEE), 172–177. doi:10.1109/ECMR.2013.6698838

Lalonde, J.-F., Vandapel, N., Huber, D. F., and Hebert, M. (2006). Natural terrain classification using three-dimensional ladar data for ground robot mobility. *Journal of Field Robotics* 23, 839–861. doi:10.1002/rob.20134

Lely (2016). Lely Discovery Collector. `https://www.lely.com/the-barn/housing-and-caring/discovery-collector`. Accessed: 2017-08-09

Liggins, M. E., Hall, D. L., and Llinas, D. (2001). *Handbook of multisensor data fusion*

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). Microsoft COCO: Common objects in context. *arXiv preprint arXiv ...* cs.CV, 1–15

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3431–3440

Luettel, T., Himmelsbach, M., and Wuensche, H.-J. (2012). Autonomous Ground Vehicles—Concepts and a Path to the Future. *Proceedings of the IEEE* 100, 1831–1839. doi:10.1109/JPROC.2012.2189803

Moore, T. and Stouch, D. (2014). A generalized extended kalman filter implementation for the robot operating system. In *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)* (Springer)

Moorehead, S. S. J., Wellington, C. K. C., Gilmore, B. J., and Vallespi, C. (2012). Automating orchards: A system of autonomous tractors for orchard maintenance. *Proc. IEEE Int. Conf. Intelligent Robots and Systems, Workshop on Agricultural Robotics* , 632

Mottaghi, R., Chen, X., Liu, X., Cho, N.-G., Lee, S.-W., Fidler, S., et al. (2014). The role of context for object detection and semantic segmentation in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE), 891–898

Munkres, J. (1957). Algorithms for the Assignment and Transportation Problems. doi:10.1137/0105003

Nam, W., Dollár, P., and Han, J. H. (2014). Local decorrelation for improved detection. *Adv. Neural Inf. Process. Syst.* , 1–9

Ollis, M. and Stentz, A. (1997). Vision-based perception for an automated harvester. *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97* 3, 1838–1844. doi:10.1109/IROS.1997.656612

Papadakis, P. (2013). Terrain traversability analysis methods for unmanned ground vehicles: A survey. *Engineering Applications of Artificial Intelligence* 26, 1373–1385. doi:10.1016/j.engappai.2013.01.006

Pathak, K., Birk, A., Poppinga, J., and Schwertfeger, S. (2007). 3D Forward sensor modeling and application to occupancy grid based sensor fusion. *IEEE International Conference on Intelligent Robots and Systems* 2, 2059–2064. doi:10.1109/IROS.2007.4399406

Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., et al. (2009). Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*

Redmon, J. (2013). Darknet: Open source neural networks in c. *h ttp://pjreddie. com/darknet* 2016

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, Real-Time object detection

Redmon, J. and Farhadi, A. (2016). YOLO9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*

Reina, G. and Milella, A. (2012). Towards Autonomous Agriculture: Automatic Ground Detection Using Trinocular Stereovision. *Sensors* 12, 12405–12423. doi:10.3390/s120912405

Reina, G., Milella, A., Rouveure, R., Nielsen, M., Worst, R., and Blas, M. R. (2016). Ambient awareness for agricultural robotic vehicles. *Biosystems Engineering* 146, 114–132. doi:10.1016/j.biosystemseng.2015.12.010

Ross, P., English, A., Ball, D., Upcroft, B., and Corke, P. (2015). Online novelty-based visual obstacle detection for field robotics. *Proceedings - IEEE International Conference on Robotics and Automation* 2015-June, 3935–3940. doi:10.1109/ICRA.2015.7139748

Rovira-Mas, F., Reid, J., Han, S., et al. (2005). Obstacle detection using stereo vision to enhance safety of autonomous machines. *Transactions of the ASAE* 48, 2389–2397

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for Large-Scale image recognition , 1–13

Sofman, B., Bagnell, J. A., and Stentz, A. (2010). Anytime online novelty detection for vehicle safeguarding. *Proceedings - IEEE International Conference on Robotics and Automation* , 1247–1254doi:10.1109/ROBOT.2010.5509357

Stachniss, C. (2009). *Robotic Mapping and Exploration*. doi:10.1007/978-3-642-01097-2

Stentz, A., Dima, C., Wellington, C., Herman, H., and Stager, D. (2002). A system for semi-autonomous tractor operations. *Autonomous Robots* 13, 87–104. doi:10.1023/A:1015634322857

Stutz, D., Hermans, A., and Leibe, B. (2016). Superpixels: An Evaluation of the State-of-the-Art. *CoRR* abs/1612.0

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics* (Cambridge, Mass.: MIT Press)

Underwood, J. P., Hill, A., Peynot, T., and Scheding, S. J. (2010). Error modeling and calibration of exteroceptive sensors for accurate mapping applications. *Journal of Field Robotics* 27, 2–20

Wellington, C., Courville, A., and Stentz, A. T. (2005). Interacting Markov Random Fields for Simultaneous Terrain Modeling and Obstacle Detection. In *Proceedings of Robotics: Science and Systems*. vol. 17, 251–60. doi:10.1.1.64.1208

Wellington, C. and Stentz, A. (2004). Online Adaptive Rough-Terrain Navigation in Vegetation. *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004* 1, 96–101 Vol.1. doi:10.1109/ROBOT.2004.1307135

Winner, H. (2015). *Handbuch Fahrerassistenzsysteme - Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort* (Wiesbaden: Vieweg+Teubner Verlag)

Wu, T.-F., Lin, C.-J., and Weng, R. C. (2004). Probability Estimates for Multi-class Classication by Pairwise Coupling. *Journal of Machine Learning* 5, 975–1005. doi:10.1016/j.visres.2004.04.006

Yang, L. and Noguchi, N. (2012). Human detection for a robot tractor using omni-directional stereo vision. *Computers and Electronics in Agriculture* 89, 116–125

FIGURE CAPTIONS

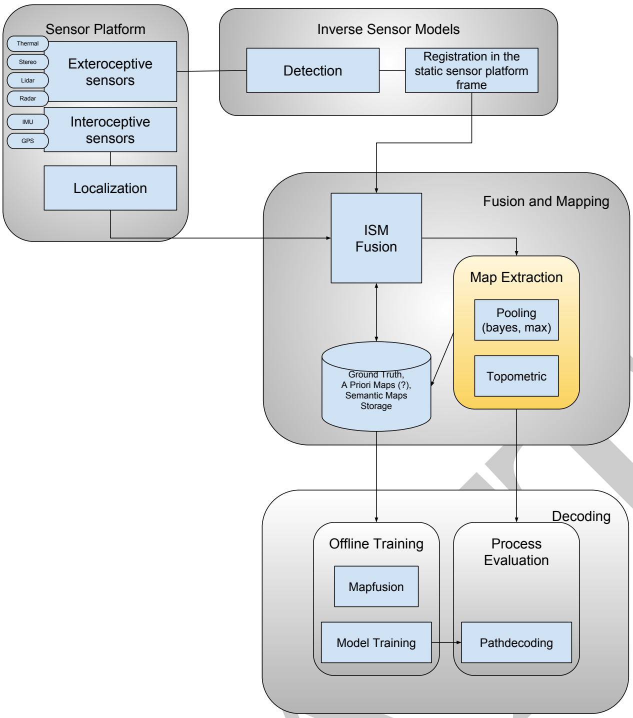


Figure 18. Only jpg and tif files are allowed when submitting, and eps upon paper accept. But for now, let's just use other formats as well. Then we can update figures before submitting.

(19a) A subfigure

(19b) Another subfigure

Figure 19. A figure

Paper 5

Using Deep Learning to Challenge Safety Standard for Highly Autonomous Machines in Agriculture

Kim Arild Steen, Peter Christiansen, Henrik Karstoft and Rasmus Nyholm Jørgensen

Peer reviewed

Paper published in Journal of Imaging, February 2016

Journal of
Imaging

MDPI

Article

Using Deep Learning to Challenge Safety Standard for Highly Autonomous Machines in Agriculture

Kim Arild Steen *,†, Peter Christiansen †, Henrik Karstoft and Rasmus Nyholm Jørgensen

Department of Engineering, Aarhus University, Finlandsgade 22 8200 Aarhus N, Denmark;
pech@eng.au.dk (P.C.); hka@eng.au.dk (H.K.); rnj@eng.au.dk (R.N.J.)
* Correspondence: kim.steen@eng.au.dk; Tel.: +45-3116-8628
† These authors contributed equally to this work.

Academic Editors: Francisco Rovira-Más and Gonzalo Pajares Martinsanz
Received: 18 December 2015; Accepted: 2 February 2016; Published: 15 February 2016

Abstract: In this paper, an algorithm for obstacle detection in agricultural fields is presented. The algorithm is based on an existing deep convolutional neural net, which is fine-tuned for detection of a specific obstacle. In ISO/DIS 18497, which is an emerging standard for safety of highly automated machinery in agriculture, a barrel-shaped obstacle is defined as the obstacle which should be robustly detected to comply with the standard. We show that our fine-tuned deep convolutional net is capable of detecting this obstacle with a precision of 99.9% in row crops and 90.8% in grass mowing, while simultaneously not detecting people and other very distinct obstacles in the image frame. As such, this short note argues that the obstacle defined in the emerging standard is not capable of ensuring safe operations when imaging sensors are part of the safety system.

Keywords: deep learning; obstacle detection; autonomous; ISO

1. Introduction

In order for an autonomous vehicle to operate safely and be accepted for unsupervised operation, it must perform automatic real-time risk detection and avoidance in the field with high reliability [1]. This property is currently being described in an ISO/DIS standard [2], which contains a short description of how to meet requirements for obstacle detection performance. The requirements for tests and the test object are described in Section 5 in the standard. The standard uses a standardized object, shown in Figure 1 which is meant to mimic a human seated (torso and head). This standardized object makes sense for non-imaging sensors such as ultrasonic sensors, LiDARs or Time-of-Flight cameras, which measures the geometrical properties of the objects or distance to the objects. However, for an imaging sensor such as an RGB camera, the definition of this standardized object does not guarantee safety. In this short note, we will present how deep learning methods can be used to design an algorithm that robustly detects the standardized object in various situations, including high levels of occlusion. Based on this, the algorithm is able to comply with the standard, but at the same time, it is not detecting people and animals, as they are not part of the trained model.

Deep convolutional neural networks have demonstrated outstanding performance in various vision tasks such as image classification [3], object classification [4,5], and object detection [4–6]. LeCun et al. formalized the idea of the deep convolutional architecture [7], and Krizhevsky *et al.* introduced a paradigm shift in image classification with the AlexNet [3]. In recent years the AlexNet has been used in various deep learning related publications.

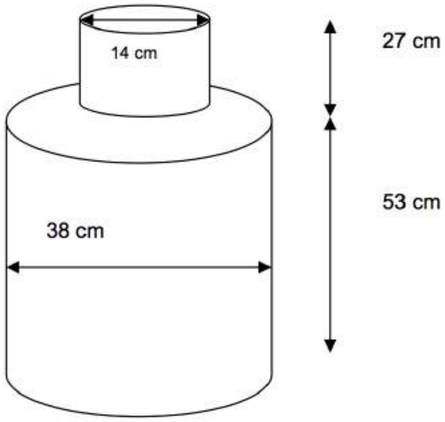


Figure 1. Standardized obstacle.

2. Materials and Methods

In this section, we present the data used for this paper, together with a short description of how the deep learning algorithm was trained and implemented.

2.1. Data Collection

The results in this paper are based on recordings performed in both row-crop fields and grass fields. The recordings were part of a research project aimed at improving safety for semi-autonomous and autonomous agricultural machines [1]. The recordings contain various kinds of obstacles, such as people, animals, well covers and the ISO standardized obstacle. All obstacles are stationary and the data is recorded while driving towards and past them. The sensor kit (seen in Figure 2a) used for the experiments includes a number of imaging devices: a thermal camera, a 3D LiDAR, an HD-webcam and a stereo camera. In this paper, we only focus on the RGB-images. Images from the experiments and recordings are seen in Figure 2.

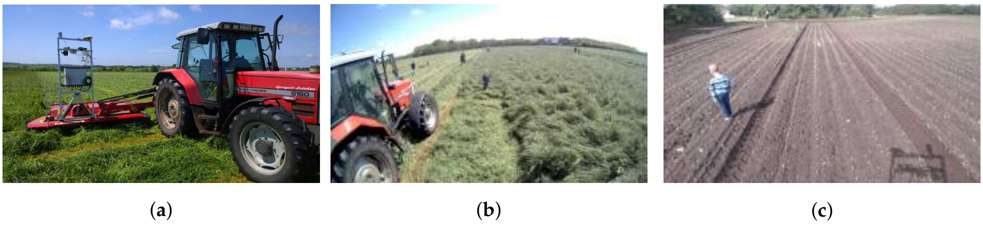

(**a**) (**b**) (**c**)

Figure 2. Images from experiments and recordings. (**a**) The sensor kit mounted on a mower; (**b**) An example image from the recordings in grass; (**c**) An example image from the recordings in row crops.

2.1.1. Training Data

An iPhone 6 was used to record five short videos of the ISO standardized obstacle. The recordings include various rotations, scales, and intensity of the object. A total of 437 frames from the videos were extracted and bounding boxes of the object were created. Example frames from the videos can be seen in Figure 3.

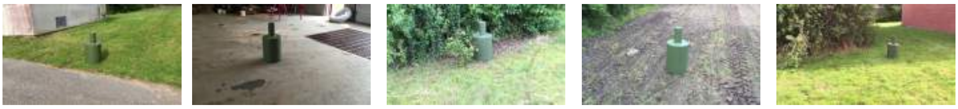


Figure 3. Examples of training data.

2.2. Training of Deep Convolutional Network

Deep learning is utilized for barrel detection using a sliding window approach similar to [6]. We start by fine-tuning AlexNet (Available at the Caffe model zoo [8]), which is pre-trained on data from the image-net competition [9] for ISO standardized obstacle detection. By fine-tuning the neural network to images of the ISO obstacle, which has a specific shape, texture and color, the algorithm will be very good at detecting the occurance of this specific object in the image. At the same time, the algorithm will also be very good at rejecting other objects in the image (animals, people, etc.), thereby meeting the standard with respect to performance, but not with respect to safety.

To increase the number of training examples (both positive and negative), we randomly sampled sub-windows of the extracted training images. A sub-window was labeled as positive (containing an object) if it had over 50% intersection over union with the labeled bounding boxes. To include additional negatives, non-face sub-windows are collected from *Annotated Facial Landmarks in the Wild* database [10] in a similar approach. A total of 1925 positive and 11, 550 negative samples have been used in this paper. These examples were then resized to 114 × 114 pixels and used to fine-tune a pre-trained AlexNet model. The original AlexNet model outputs a vector of 1000 units, each representing the probability of the 1000 different classes. In our case, we only want to detect if an image patch contains an ISO object or not. Hence, the last layer is changed to output a two-dimensional vector. For fine-tuning, we used 14K iterations and batch size of 100 images, where each batch contained 67 positive and 33 negative examples. During fine-tuning, the learning rate for the convolutional layers was 10 times smaller than the learning rate of the fully connected layers.

After fine-tuning, the fully-connected layers of the AlexNet model can be converted into convolutional layers by reshaping layer parameters [11]. This makes it possible to efficiently run the network on images of any size and obtain a heatmap of the ISO obstacle classifier. Each pixel in a heatmap shows the network response, which is the likelihood of having an ISO obstacle, corresponding to the 114 pixel ×114 pixel region in the original image. In order to detect ISO obstacles of different sizes, the input images can be scaled up or down. The chosen training image resolution is half the resolution used in the original AlexNet. Reducing the resolution of the training images allows us to reduce the input image by half, thus reducing processing time, while maintaining the resolution of the resulting heatmaps. An example of a resulting heatmap is illustrated in Figure 4.

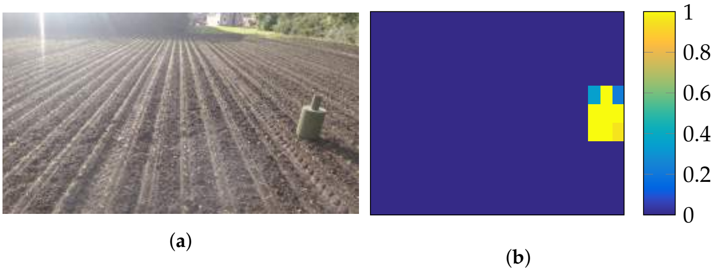


Figure 4. Illustration of ISO obstacle and resulting heatmap. (**a**) RGB image from the row crop field; (**b**) Resulting heatmap.

The model was trained using the Caffe framework [12] using a single GPU (4 GB Quadro K2100M). The training time was approximately 1–2 h.

2.3. Detection of ISO Obstacle using Deep Convolutional Network

When the deep convolutional network has been trained, it can be used to detect the ISO obstacle in color images. In order to detect the obstacle at multiple distances, the input image needs to be scaled accordingly. In this paper, we use 13 scales, which are all processed by the same network structure. We use 13 scales to be able to detect the barrel when it is far away (57 pixel × 57 pixel in the original image) and up close (908 pixel × 908 pixel). As described in the previous section, the output is a heatmap, where the intensity reflects the likelihood of an ISO obstacle.

Based on the heatmap, one can detect the obstacle and draw a bounding box. Each pixel in the heatmap corresponds to a 114 pixel × 114 pixel sub-window in the input image. To remove redundant overlapping detection boxes, we use non-maximum suppression [6] with 50% overlap threshold.

3. Results

The results are based on data collected in two different cases, at three different dates. The data has been collected at different times during the days to ensure different lighting conditions. In both cases, we use the model trained on the data presented in Section 2.1.1. Despite this, the results in this paper are of a preliminary nature, as various weather conditions and scenarios are not included in the data.

3.1. Row Crops

Based on the presented algorithm, a total of 7 recordings have been evaluated with respect to detection of the ISO obstacle in row crops. The recordings also contain other kinds of obstacles such as people and animals. The recordings contain a total of 14, 153 frames with 20, 414 annotated obstacles (8126 of those are the ISO obstacle).

In the ISO standard, the obstacles needs to be detected within a defined safety distance. The safety distance is a product of the expected working speed and machine type. Hence, there is no fixed value for this. In Figure 5, a histogram of the achieved detection distances is shown. It is seen that the algorithm is able to detect the obstacle both at close range (3–6 m) and also at far range (over 15 m). The ISO obstacle is present in front of the machine a total of 14 times and the algorithm is able to detect the obstacle everytime. The detection distance, which is the distance of the first positive detection, for these 14 times, ranges from 10 m to 20 m, with an average of 14.56 m.

Evaluating all frames, at frame level with all annotated objects, we achieve TP (true positive) = 2851, FP (false positive) = 1, TN (true negative) = 7105 and FN (false negative) = 4919. The high number of false negatives is a result of annotations, where the ISO obstacle is located more than 20 m away, which is more than the achieved detection range. Based on this, we achieve a hit rate of 36.7% Equation (1), which is the ratio between positive detections and all annotations of the ISO obstacle, and a precision of 99.9% Equation (2) (As there are less than 10, 000 datapoints, we are not able to present the last decimal as a result). In the standard, it is stated that the system must achieve a success rate of 99.99%, however, it is unclear how this should be tested. As stated above, the algorithm is able to detect the ISO obstacle when the ISO obstacle is within 10 m of the machine at a precision of almost 100%.

$$hit\ rate = \frac{TP}{TP + FN} = \frac{2851}{2851 + 4919} = 0.3669 \tag{1}$$

$$precision = \frac{TP}{TP + FP} = \frac{2851}{2851 + 1} = 0.9996 \tag{2}$$

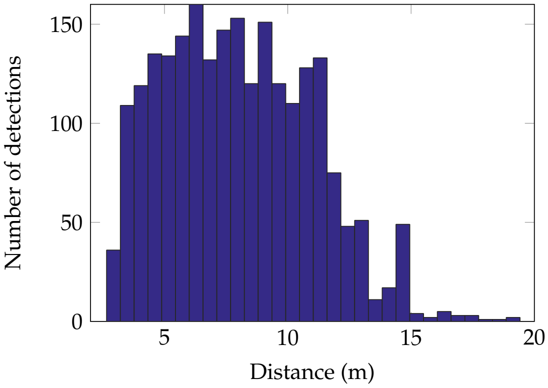


Figure 5. Histogram of detection distances.

In Figure 6, the achieved hit rate for different distance ranges is shown. It is seen that the algorithm is not able to detect the obstacle in all frames (the hit rate is below 1). However, with a precision close to 100%, a single detection is reliable enough to be considered a detection of the ISO obstacle present in the recordings. Hence, the success rate is 99.9%, estimated at frame level, within an average safety distance of 14.56 m.

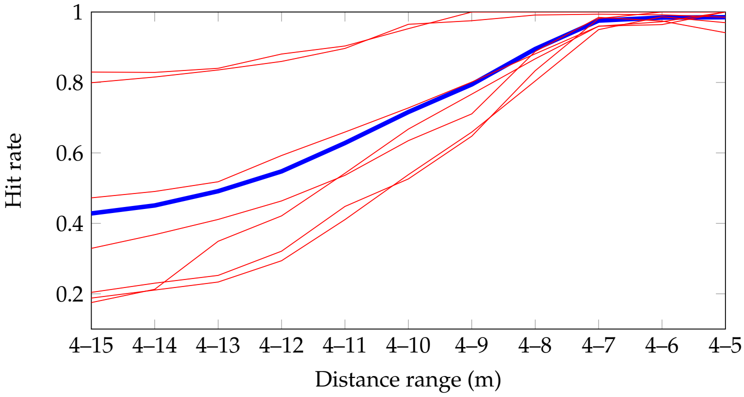


Figure 6. Hit rate, evaluated at frame level, for different distance ranges (e.g. 0.97 is the mean hit rate for all frames in range 4–7 m). Red: hit rate for the 7 recordings, and blue: average hit rate.

3.2. Grass Mowing

We also evaluated the algorithm in a much more difficult case; grass mowing. In grass mowing, the obstacles are often highly occluded (a scenario that is not described in detail in the standard). The recording contains the ISO obstacle and people. The recordings contain a total of 19, 390 frames with 936 annotated ISO obstacles.

As with the row crops case, we evaluate the achieved detection distance. In the recording, the ISO obstacle is detected when the ISO obstacle is within approximately 6 m of the machine. The ISO obstacle is present in front of the machine a total of 8 times and the algorithm is able to detect the obstacle everytime.

Evaluating all frames at frame level, we achieve TP = 307, FP = 31, TN = 18,337 and FN = 787. The high number of false negatives is a result of annotations, where the ISO obstacle is located more than 6 m away, which is more than the achieved detection range. Based on this, we achieve a hit rate of 28.1% Equation (3)

$$hit\ rate = \frac{TP}{TP + FN} = \frac{307}{307 + 787} = 0.281 \tag{3}$$

$$precision = \frac{TP}{TP + FP} = \frac{307}{307 + 31} = 0.908 \tag{4}$$

and a precision of 90.8% Equation (4). Again, the ISO obstacle is successfully detected everytime we are driving towards it, however, the achieved precision is lower than 99.99% which is stated in the standard. As seen in Figure 7, the ISO obstacle is highly occluded. In the standard, it is noted that the obstacle must appear unobscured to ensure high levels of system confidence and if the obstacle is obscured, the manufacturer must understand how this affects performance. In the grass case, we show how the system is affected by this. The precision drops from 99.9% to 90.8% and the achieved detection distance drops from an average of 14.56 m to 6 m. The lower precision is due to a higher number of false positives. Most of these false positives are the tractor in the image. The tractor will always be present in the same position in the image and further developments could remove this in post-processing or by including tractor images as negatives in the training data.

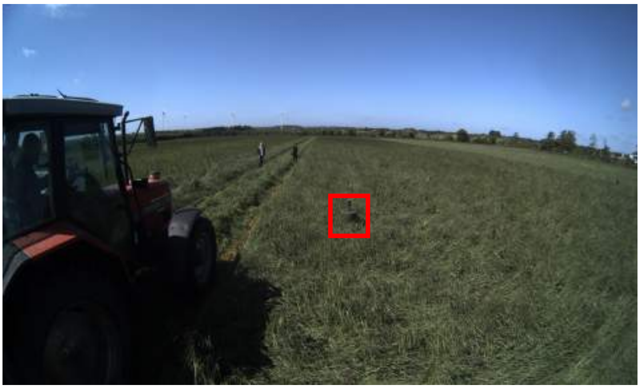


Figure 7. Detections in grass mowing case. Notice that people are not detected.

4. Discussion

The results show that we are able to robustly detect the presence of the ISO obstacle in various conditions including different lighting and heavy occlusion. The results also show that the algorithm is not able to detect the presence of the other obstacles within the image frame—even people. This is not a surprise, as the model has been trained on the ISO obstacle and not on other types of obstacles.

We are using a large deep learning model to detect a simple object and it might seem like we are overdoing it. However, by using deep learning and pre-trained networks, we have been able to design a robust classifier for detecting the ISO obstacle in various scenarios, using only a few minutes of training video. This shows the power of these models and how they can be exploited.

In this paper, we have implemented the algorithm using Caffe and the corresponding MATLAB interface, which means that it does not run in real-time. However, CNN implementations ready for real-time applications exist in literature [13]. Furthermore, deep learning algorithms are also being utilized in detection systems for the car industry, where deep learning models are able to classify between a number of different obstacles. This is powered by high-end embedded boards from NVIDIA, which has recently launched a 300 USD board [14], enabling real-time deep neural nets on UAVs and UGVs. These observations make it fair to assume that agricultural machines could also benefit from that computing power.

The ISO standard states that the system must have a success rate of 99.99% in various weather conditions, however, it is unclear how this success rate should be measured. We are not able to detect the ISO obstacle in all frames, however, we achieve a precision of 99.9%, and are able to detect the ISO obstacle at an average distance of 14.56 m in row crops. In the grass mowing case, the obstacle was highly occluded which affected the achieved detection distance. Furthermore, the presence of the tractor within the image frame resulted in more false positives. These should be removed to ensure higher precision. We have not been able to test the performance in various weather conditions, hence, the results obtained in this paper is of a preliminary nature.

The most important result in this paper is that we are able to show that an imaging system can be designed to comply with the ISO standard and completely miss important obstacles such as people—both kids and adults. We argue that the standardized obstacle presented in the standard is not fully adequate to ensure safe operations with highly autonomous machines in agriculture. The design of the obstacle is based on a human head and torso (human sitting down), but we show that we can detect this type of obstacle and at the same time completely miss the people in front of the machines. As the ISO standard aims to represent a human sitting down, we suggest that the standardized obstacle should resemble a more life-like person, if a standardized object is required. In our experiments, we have used mannequins for this task. It is important that the life-like obstacles have the same properties as the current ISO obstacles with respect to color, material, and temperature (hot water). The life-like obstacle could be used to test different possible postures, such as standing, lying and sitting. However, other kinds of obstacles could also be present in the fields. Including other types of obstacles, such as animals, in the requirements, could potentially increase safety overall. However, even doing this, there is no guarantee that the methods will be able to detect real obstacles in the fields, as they might not be perfectly described by the trained algorithm. Deep learning methods have achieved very good performance in very difficult image and object recognition tasks. This is accomplished through access to a vast amount of image training data, where objects like people, animals and cars are depicted in thousands of different postures, sizes, colors and situations. The deep learning framework is able to encapsulate this great amount of variability and thereby produce beyond-human performance in object recognition tasks. This is being exploited in the work towards autonomous cars and could also be done in agriculture.

5. Conclusions

In an emerging standard for safety of highly automated machinery in agriculture, a barrel-shaped obstacle is defined as the obstacle which should be robustly detected to comply with the standard. In this paper, we show that by using deep learning techniques, we are able to achieve a high level of precision in two different cases in agricultural operations, with one of the cases concerning highly occluded obstacles. The algorithm detects the ISO specified obstacle in every test run, but it completely misses important obstacles such as people.

Therefore, we argue that the standardized obstacle presented in the standard is not fully adequate to ensure safe operations with highly autonomous machines in agriculture and further work should be conducted to describe an adequate procedure for testing the obstacle detection performance of highly autonomous machines in agriculture.

Acknowledgments: This research is sponsored by the Innovation Fund Denmark as part of the project "SAFE - Safer Autonomous Farming Equipment" (Project No. 16-2014-0) and "Multi-sensor system for ensuring ethical and efficient crop production" (Project No. 155-2013-6). We would also like to thank Ole Green from Agro Intelligence ApS for his contributions to the projects.

Author Contributions: Kim Arild Steen and Peter Christiansen contributed to the data aquisition, algorithm development, algorithm testing and writing of the manuscript. Henrik Karstoft has contributed with internal review of the manuscript, and Rasmus Nyholm Jørgensen has contributed with formalizing the hypothesis of this contribution, field experiment planning, algorithm testing and internal review of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Christiansen, P.; Hansen, M.; Steen, K.; Karstoft, H.; Jørgensen, R. Advanced sensor platform for human detection and protection in autonomous farming. In *Precision Agriculture'15*; Wageningen Academic Publishers: Wageningen, The Netherlands, 2015; pp. 1330–1334.
2. ISO/DIS 18497. Available online: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=62659 (accessed on 17 December 2015).
3. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2012; pp. 1097–1105.
4. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. Available online: http://arxiv.org/pdf/1409.4842v1.pdf (accessed on 18 December 2015).
5. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. Available online: http://arxiv.org/pdf/1409.1556v6.pdf (accessed on 18 December 2015).
6. Farfade, S.S.; Saberian, M.; Li, L.J. Multi-view Face Detection Using Deep Convolutional Neural Networks. Available online: http://arxiv.org/pdf/1502.02766v3.pdf (accessed on 18 December 2015).
7. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324.
8. AlexNet. Available online: http://caffe.berkeleyvision. org/model_zoo.html (accessed on 10 October 2015).
9. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; *et al*. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2014**, *115*, 1–42.
10. Koestinger, M.; Wohlhart, P.; Roth, P.M.; Bischof, H. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In Proceedings of the First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies, Barcelona, Spain, 6–13 Novenber 2011.
11. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. Available online: http://arxiv.org/pdf/1411.4038v2.pdf (accessed on 18 December 2015).
12. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 675–678.
13. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. Available online: http://arxiv.org/pdf/1506.02640v4.pdf (accessed on 18 December 2015).
14. Nvidia Jetson TX1. Available online: http://www.nvidia.com/object/jetson-tx1-dev-kit.html (accessed on 1 December 2015).

Paper 6

(Not open-access) Platform for evaluating sensors and human detection in autonomous mowing operations

Peter Christiansen, Mikkel Kragh, Kim A. Steen, Henrik Karstoft, Rasmus N. Jørgensen

Peer reviewed

Paper published in Precision Agriculture, June 2017

Precision Agric (2017) 18:350–365
DOI 10.1007/s11119-017-9497-6

Platform for evaluating sensors and human detection in autonomous mowing operations

P. Christiansen[1] · **M. Kragh**[1] · **K. A. Steen**[1] · **H. Karstoft**[1] · **R. N. Jørgensen**[1]

Published online: 13 January 2017

Abstract The concept of autonomous farming concerns automatic agricultural machines operating safely and efficiently without human intervention. In order to ensure safe autonomous operation, real-time risk detection and avoidance must be undertaken. This paper presents a flexible vehicle-mounted sensor system for recording positional and imaging data with a total of six sensors, and a full procedure for calibrating and registering all sensors. Authentic data were recorded for a case study on grass-harvesting and human safety. The paper incorporates parts of ISO 18497 (an emerging standard for safety of highly automated machinery in agriculture) related to human detection and safety. The case study investigates four different sensing technologies and is intended as a dataset to validate human safety or a human detection system in grass-harvesting. The study presents common algorithms that are able to detect humans, but struggle to handle lying or occluded humans in high grass.

Keywords Safe farming · Sensor platform · Object detection · Computer vision · ISO 18497 · Autonomous farming

Introduction

Current technology is capable of automatically navigating and operating agricultural machinery, such as tractors and harvesters, efficiently and more precisely compared to manual operation. However, a crucial deficiency in this technology concerns the safety aspects. In order for an autonomous vehicle to operate safely and be certified for

✉ P. Christiansen
pech@eng.au.dk

[1] Department of Engineering-Signal Processing, Faculty of Science and Technology, Aarhus University, Finlandsgade 22, 8200 Aarhus N, Denmark

unsupervised operation, it must perform automatic real-time risk detection and avoidance of humans in the field with high reliability (ISO 18497 2015).

Robust risk detection imposes a number of challenges for the sensor system. Varying weather and lighting conditions influence the image quality of sensing technologies in different ways, and thus no sensor is single-handedly capable of detecting objects reliably under all conditions. Active sensors such as LiDAR, and passive sensors such as RGB camera, stereo camera and thermal camera have different strengths and weaknesses concerning weather, lighting, range and resolution, and therefore a variety of these sensors are needed to cover all scenarios (Rasshofer and Gresser 2005). In addition, attitude estimation sensors such as accelerometers, gyroscopes and GPS are needed for estimating the vehicle position, velocity and orientation and for synchronizing and registering subsequent frames acquired from the imaging sensors.

Today, driver assistance systems are available for a large number of modern passenger cars, and completely autonomous vehicles operating in urban and sub-urban environments are emerging for experimental usage (Paden et al. 2016).

In the agricultural sector, a variety of machines have been operating autonomously for a decade using either precise GPS co-ordinates and/or cameras detecting structures in the field (CLAAS Steering Systems 2011; Pilarski et al. 2002). Efforts have been made to fully automate the process in a driverless solution, but safety aspects currently prevent authorization for this. In Freitas et al. (2012), Yang and Noguchi (2012) and Wei et al. (2005), human detection was performed using only a single sensor (laser scanner or stereo camera). However, multiple sensor modalities should be investigated to evaluate their ability to detect humans. For instance, the QUAD-AV project has investigated microwave radar, stereo vision, LiDAR and thermography for detecting obstacles in an agricultural context (Rouveure et al. 2012). Within the project, a detailed study of stereo vision has shown promising results on ground/non-ground classification (Reina and Milella 2012).

In urban environments, autonomous vehicles can exploit obstacles protruding from the surface. In farming operations, obstacles are commonly placed below or just above an uneven surface of crops introducing specific challenges for autonomous vehicles in agriculture. The likelihood of a human being one of these obstacles is small. However, a child or a fallen, injured or unconscious human provides a risk as these non-protruding objects have reduced mobility. To investigate these challenges, data from agricultural fields and algorithms are needed.

Human safety is addressed in ISO 18497 (an emerging standard for safety of highly automated machinery in agriculture) by defining a minimum obstacle that must be detected with an accuracy of 99.99% (ISO 18497 2015). The minimum obstacle is specified as an olive green barrel shaped object that resembles a small or seated human in green clothing (in this paper defined as an ISO-barrel).

This paper describes a flexible vehicle-mounted sensor platform targeting agricultural fields. The sensor platform records imaging data and vehicle position for a moving vehicle using three passive imaging sensors, one active sensor and two attitude/position estimation sensors. The sensor platform is designed to record simultaneous data from all sensors, thus preparing for subsequent offline processing. Offline processing and visualization of sensor data is presented to investigate the object detection potential for the different sensors. The current paper is an extended version of Christiansen et al. (2015) providing more authentic data in grass-harvesting operations and addressing human safety in more detail. An ISO-barrel was produced under the specification defined in ISO 18497. The ISO-barrel as well as humans and mannequins were placed in standing and lying positions in front of the setup to create recordings that could be used in an actual validation of a human detection system during grass-harvesting. The extended edition also presents a full procedure for calibrating and registering all sensors using a single calibration thermal checkerboard.

Materials and methods

Sensors

An overview of the strengths and weaknesses of the selected imaging and active sensors are presented in Table 1. The qualities are evaluated individually and under various conditions. A weakness is marked with '−' and a strength is marked with '+'.

Sensor modalities refer to the information a sensor measures. In this paper, a sensor modality is either visual light, depth or heat radiation.

An *RGB camera* captures the modality of visible light. The sensor is useful for identifying the perceived objects as it provides visual characteristics such as texture, color and shape in high resolution at low cost. It is invariant to protrusion, meaning that non-protruding objects such as small animals, a fallen human or humans/animals in high crops are still visible. However, visual characteristics are affected by occlusion from crops, weather conditions (rain, fog and snow) and illumination such as dim light (night) or direct light (causing shadows). An RGB camera is not able to exploit depth information to emphasize protruded objects and the lack of depth makes the positioning of objects in 3D space difficult.

A stereo camera enables 3D imaging data (depth and color information). Depth and color information are registered and the sensor is thus able to exploit the advantages of both modalities. Depth information can be used to see protruded objects and visually camouflaged animals easily while determining the position of an object relative to the vehicle. In this way, depth-aware algorithms can abstract from the very different visual characteristics of objects (shape, color and texture) creating simple detection algorithms. Like the RGB camera, the stereo camera is sensitive to illumination and weather conditions, although the depth information is in some cases still retrievable.

Table 1 Strengths and weaknesses of sensors (Christiansen et al. 2015)

Names	RGB	RGB stereo	Thermal	LiDAR
Specification				
Range	Medium	Medium	Medium	Long
Resolution	+	+	−	−
Depth information	−	+	−	+
Heat information	−	−	+	−
Color information	+	+	−	−
Cost	Low	Medium	Medium	High
Robustness				
Light changes	−	−	+	+
Weather changes	−	−	−	+
Camouflaged objects	−	+	+	+
Protruding objects	−	+	−	+
Non-protruding objects	+	−	+	−

A *thermal camera* is an imaging sensor that captures heat radiation represented by intensities (temperatures) to form a monochromatic image. A thermal camera perceives objects of distinct temperatures, making it ideal for detecting living objects in temperate and colder climates, and even in foggy weather (Serrano-Cuerda et al. 2014). A key ability is that the sensed data are unaffected by non-protruded or visually camouflaged animals and that the distinctness of living objects becomes more apparent at night. However, these capabilities are much affected by the ambient temperature as living objects become indistinct when the temperature difference between the objects and background becomes small (Serrano-Cuerda et al. 2014). The cost of a well-performing and high resolution thermal camera is very high, but low cost cameras are emerging. Object recognition capabilities are low due to a limited resolution and limited visual characteristics.

A *LiDAR* measures range data to a set of surrounding points and generates a point cloud where each point is represented by a 3D position and reflection intensity. The LiDAR is a high cost sensor, but has dropped significantly in price in recent years. Compared to a stereo camera, the LiDAR provides very exact depth information at greater range and some models can capture in 360° horizontally. It is invariant to illumination, temperature and camouflage. The lack of visual and thermal information makes recognition of objects difficult and non-protruding objects are almost or fully undetectable.

Physical design

The sensor platform consisted of seven sensors and a controller mounted on a common rack of 2 m by 0.8 m in size. The left side of Fig. 1 shows the rack mounted on a tractor and the right side shows the physical placement of sensors. A standard A-frame was mounted at the bottom of the rack to enable easy mounting on tractors. The category 1 A-frame was mounted with dampers for absorbing internal engine vibrations from the vehicle to reduce the amount of mechanical noise acting on the sensors. The horizontal profile in the middle was adjustable in height and angle such that the imaging sensors could be oriented in a downward angle depending on the vehicle height. The LiDAR was placed above the sensor frame to minimize view obstructions for the sensor. The rack allowed sensors to be placed roughly 2 m above ground to provide a more downward view into the crop to better detect hidden obstacles. Placing sensors on top of the tractor would provide a similar downward view. However, the tall rack and the A-frame allowed the sensors to be

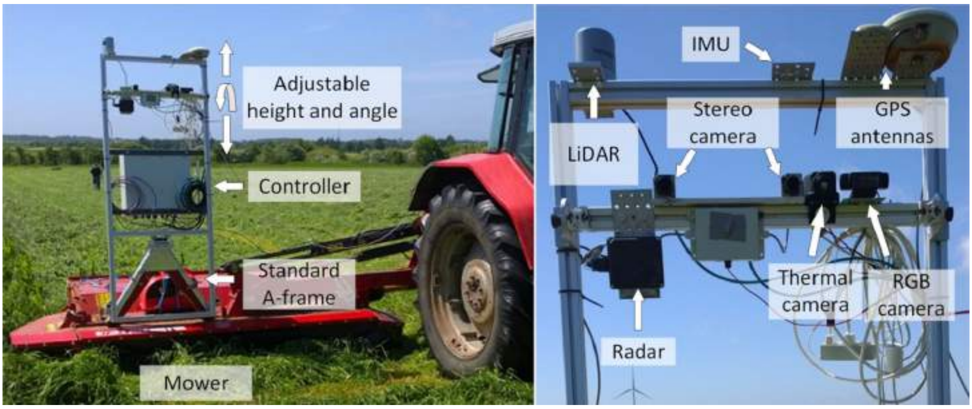


Fig. 1 *Left* sensor frame including controller. *Right* sensors on the sensor platform

easily swapped to another tractor, an all-terrain vehicle or directly on a ground socket while keeping the downward view under data acquisition.

A Logitech HD Pro C920 from Logitech (Silicon Valley, USA) webcam providing 1 920 × 1 080 pixels at 30 fps was used as the RGB camera. The stereo camera was composed of two hardware synchronized Flea3/FL3-GE-28S4C-C cameras from Point Grey (Richmond, Canada) with global shutter and 1 928 × 1 448 pixels at 15 fps. The thermal camera was a shutterless HawkVision analog thermal camera from Tonbo Imaging (Bangalore, India) providing 640 × 480 pixels at 25 fps (interlaced). The HDL-32E LiDAR from Velodyne (Morgan Hill, USA) was a 32-beam laser scanner providing 70 000 points at 10 Hz with 1–100 m range. Figure 1 shows an automotive Delphi ESR 64-target radar from Delphi (Washington, DC, USA) not addressed in the current paper as it was intended for detecting pieces of metal and not humans. The GPS was an AG GPS361 real time kinematic (RTK) GPS from Trimble (Sunnyvale, USA) enhancing the precision of GPS up to centimeter-level accuracy. The IMU was a VN-100 from Vectornav (Dallas, USA) providing synchronized three-axis accelerometers, gyros, magnetometers and a barometric pressure sensor. The data-collecting controller was an Innovation Robotech Controller 701 from Conpleks (Struer, Denmark). It is an embedded computer with external interfaces for all sensors that using ROS-middleware (robot operating system) to easily integrate them into a common framework.

System architecture

Figure 2 further illustrates the connections between the sensors and the controller. In ROS, each sensor was given its own node (an executable file) that was responsible for publishing one or more topics. For instance, the IMU had its own node including hardware-specific drivers, and it published different topics related to the readings of the accelerometer, the gyroscopes and the magnetometers. For each topic, the node could send messages containing sensor data whenever a new sensor-reading was available. Each node was connected to the ROS Master which handled interactions between nodes and supplied all messages with exact timestamps. Using the rosbag package, a recording of all desired topics (and all associated messages) to a single rosbag data-file could be obtained. A JavaScript browser interface was developed to easily monitor and record specific sensors, and enabled the platform to be controlled through Wi-Fi using a mobile phone, tablet or computer.

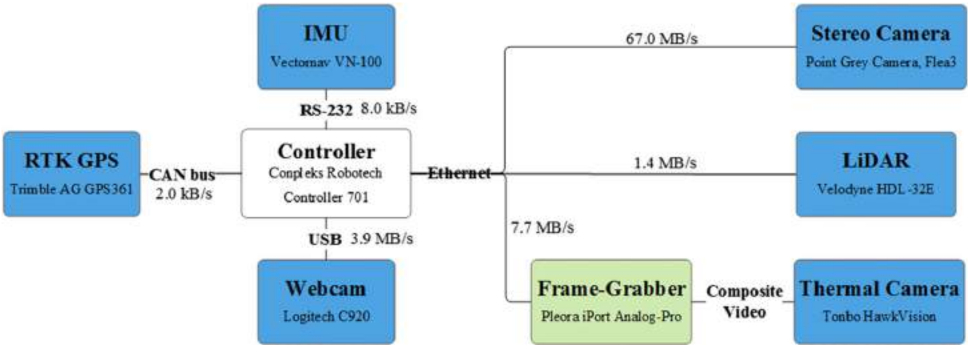


Fig. 2 System overview illustrating bandwidths and interfaces for sensors

Data

Data were collected on a grass field of roughly 7.5 ha near Lem in Denmark (latitude 56.059679° N, longitude 8.368701° E) in the beginning of June 2015. To get authentic data, sensors were mounted to a tractor working in a normal grass-harvesting operation. In operation, obstacles were placed in the trajectory of the tractor to simulate collision hazards. For each obstacle, the tractor approached the object and stopped just before collision. To enable some form of reproducibility and to ensure safety, standing/lying adult and child mannequins were used instead of real humans in the field. To incorporate safety standards, the ISO-barrel was also used. Finally, the mower was turned off and two recordings with real humans were captured. Obstacles from the data are presented in Fig. 3.

In Fig. 4, obstacle positions and the tractor route (divided into laps) are presented, where lap 17 and 18 contained real human obstacles.

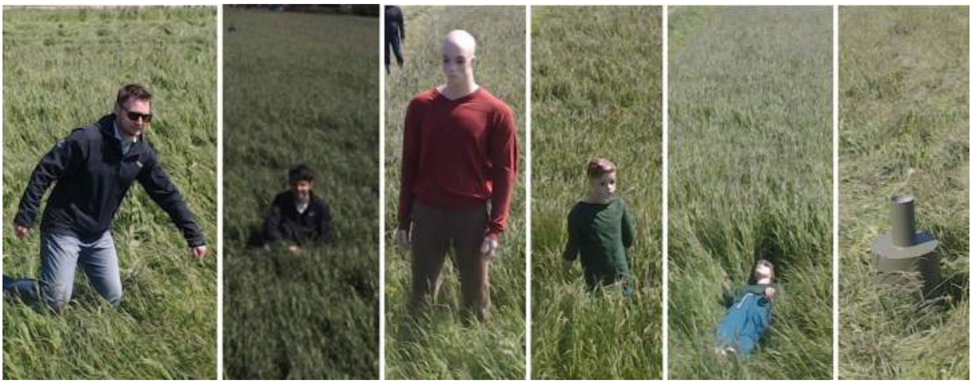


Fig. 3 Two real humans, three mannequins and the ISO-barrel

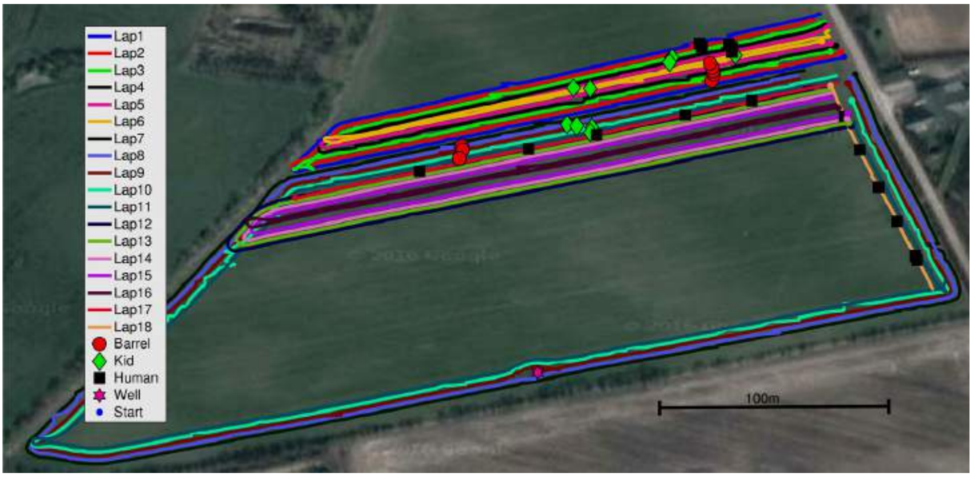


Fig. 4 Tractor route (*lines*), barrel (*circles*), kid mannequin (*diamonds*), adult mannequin (*squares*), well (*stars*) and lap starting point (*small dots*)

Registration of sensors

Registration or sensor fusion is essential for a multi-sensor system to merge and exploit information from all sensors. Registration in multiple modalities is non-trivial and can be handled in different ways (Bahnsen 2013; Zhao and Cheung 2014; Krotosky and Trivedi 2007). In particular, Bahnsen (2013) provided a coherent description of registration methods and the complications for registering different modalities, when objects are not positioned at the same distance.

In this work, common camera and sensor view geometry combined with depth information from the stereo camera were used to project points between sensor frames (Johnson and Bajcsy 2008). Such projections require the *intrinsic* parameters to calibrate cameras individually and *extrinsic* parameters—describing the inter-displacement of sensors—to finalize registration. The inter-displacement between LiDAR and stereo camera was found by matching the two point clouds using the iterative closest point algorithm (Zhang 1994).

The stereo camera and the webcam was calibrated individually using a normal checkerboard and MATLABs computer vision: calibration tool (2015). The calibration tool was able to detect checkerboards, calibrate cameras, map checkerboard to 3D position automatically and, for the stereo camera, find the inter-displacement between the left and right camera. For the webcam, the extrinsic parameters was determined by finding the transformation that matched corresponding 3D checkerboards to, in this setup, the left stereo camera. However, to calibrate and find inter-displacement between thermal and RGB cameras using a traditional and automated calibration tool, the checkerboard must be visible in both modalities. Therefore, a custom-made visual–thermal checkerboard is proposed.

Visual–thermal registration

A normal checkerboard exposed to sunlight can be used to perform thermal–visual registration as black absorbs more energy than white areas. However, the quality of the thermal calibration is dependent on weather conditions, and heat/energy is transferred in the material between black and white areas making square transitions indistinct.

A registration/calibration board was therefore developed using a circuit board with copper squares as shown in Fig. 5 (left).

The circuit board was heated by attaching an aluminum plate mounted with impact resistors on the backside of the board as in Fig. 5 (right). The 60 resistors delivered 216 W of heat using a 12 V car battery. Copper has a low emissivity coefficient, which effectively made the material work as a reflector. Thus, the non-copper squares emitted heat radiation

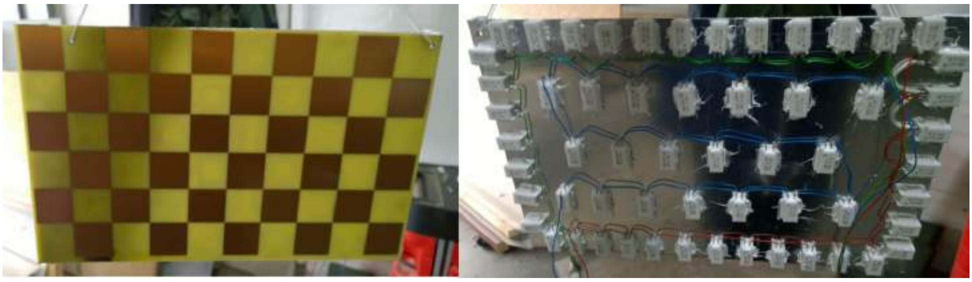


Fig. 5 Front and back side of registration board

from the heated circuit board, and the copper areas reflected heat of the surroundings, giving a distinct transition between copper and non-copper squares.

The thermal checkerboard would, in a normalized thermal image, resemble a traditional black and white checkerboard as presented in Fig. 6 (right). The thermal camera was then calibrated using traditional and automated calibration tools.

The thermal checkerboard did not, for RGB images, resemble a traditional black and white checkerboard as depicted in Fig. 6 (left). Thus, calibrations tools could not be applied directly. To use RGB images, a MATLAB script was developed to enable a user to mark an area inside the checkerboard. This area was then cropped and converted to the LAB color space. Automatically, the A and B channels were modeled into two clusters using a Gaussian mixture model (McLachlan and Basford 1988). Copper and non-copper areas were separated into two individual clusters. The posterior probability of each pixel belonging to a specific cluster generated a gray-scaled image that made the registration board resemble a traditional black and white checkerboard, see Fig. 6 (mid).

Converting RGB images, enabled all camera sensors to be calibrated and registered using only the proposed registration board. However, the procedure required the user to place a rectangular area inside the checkerboard for each image. In Fig. 7, the detected boards and the inter-displacement of sensors are visualized.

In Fig. 8 (middle left), two humans are annotated in the left stereo camera and projected to the stereo point cloud in Fig. 8 (top). The distance to objects inside the annotation was determined using the median distance of pixels inside the bounding box. The bounding box was then defined as four points in the stereo point cloud that could be projected to other

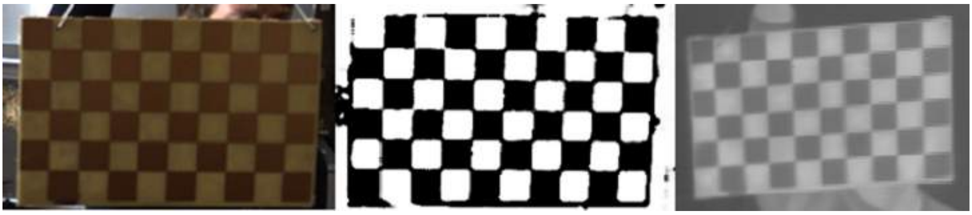


Fig. 6 The registration board (*left*) is transformed into a "classic" checkerboard (*mid*) using a Gaussian mixture model. Thermal image of the registration board (*right*)

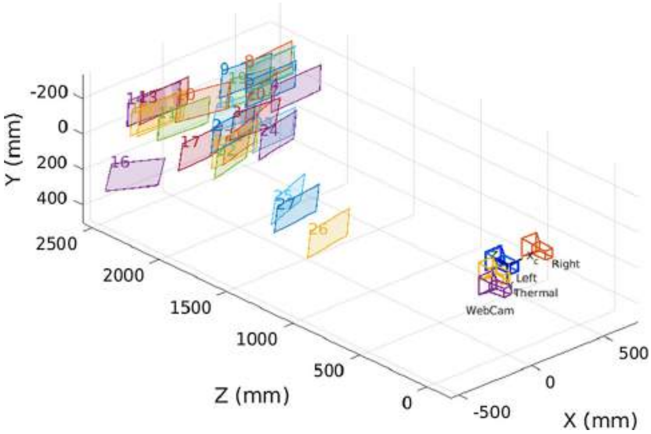


Fig. 7 Registration board placements (numbered *1*–*25*) and inter-displacement of sensors

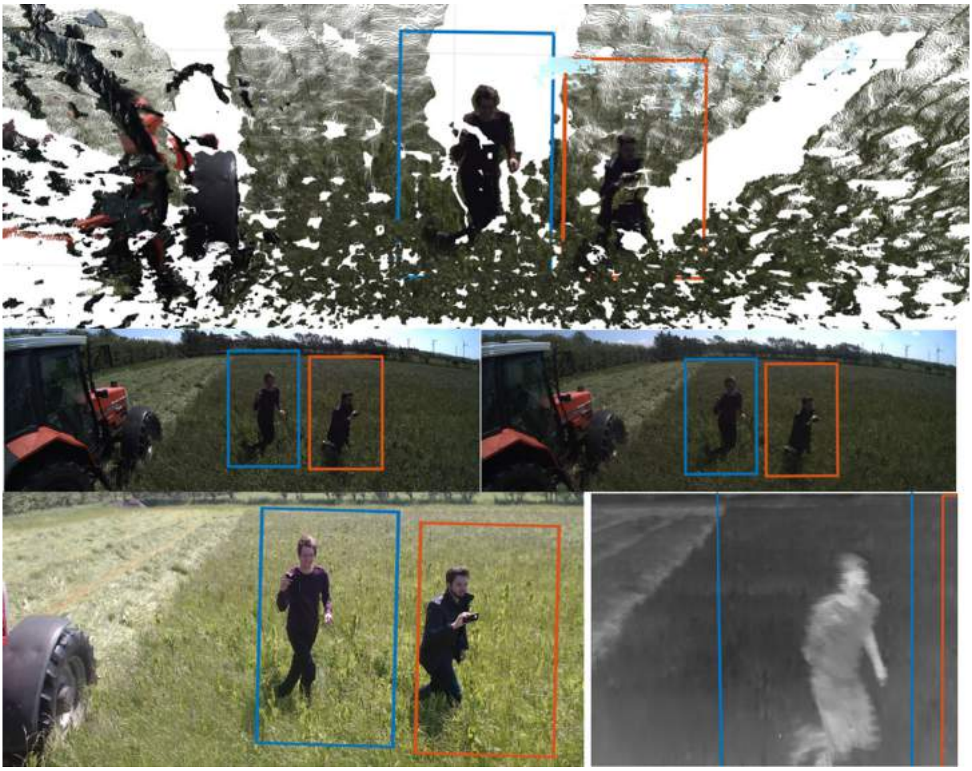


Fig. 8 Annotations in the *left image* are projected onto the stereo point cloud (*top*). These annotations are then projected to the right and left stereo camera (*middle left* and *right*), the webcam (*bottom left*) and the thermal camera (*bottom right*)

sensor frames as in Fig. 8. To make a more exact registration of sensors, the registration board should be placed at a broader range of distances from the cameras.

A more quantitative evaluation of the visual–thermal registration is presented in "Appendix: Thermal–visual registration and evaluation" section.

Signal processing

To provide an initial qualitative validation of detection performance of the different sensors in an agricultural environment, preliminary tests using different object detection algorithms have been carried out on the sensors.

Using only an RGB camera for detecting all possible obstacles in the field is complex and difficult and requires a very large dataset with many representations of each object. Constraining detection to only humans provided a more realistic case in this preliminary study. The RGB camera images were therefore processed using a state-of-the-art pedestrian detection algorithm (Dollar et al. 2010).

After stereo camera calibration (Zhang 2000), a point cloud could be generated for each stereo image pair. For both stereo and LiDAR, the same algorithm was used to better compare sensors. A ground plane was estimated on the acquired point cloud using the RANSAC algorithm (Fischler and Bolles 1981). Protruding objects were visualized by determining the height of points relative to the estimated ground plane.

The thermal camera images were processed by thresholding the (temperature-related) intensities by a constant value above the median intensity of the image (Christiansen et al. 2014). Subsequent connected components analysis was used for extracting only components that exceeded a certain area.

Results and discussion

An initial validation of detection algorithms is presented in four scenarios. The first scenario is humans of different sizes, appearances and postures similar to Christiansen et al. (2015) in low grass. Scenarios 2–4 are, respectively, a barrel, a lying child mannequin and a sitting human in high-grass taken from the above described data.

Figure 9 depicts the human detection performance evaluated at single, synchronized frames for the RGB camera, the stereo camera and the LiDAR. At the top left, the RGB camera is shown with bounding boxes indicating results of the pedestrian detection algorithm. In the top middle, the disparity map of the stereo camera is shown and, at the top right, a protrusion map indicating objects that protrude from the ground plane is visualized. At the bottom left, the thermal camera is shown with overlaid thresholded components and, at the bottom right, the LiDAR data are visualized with a ground plane and protruding points.

Using only single frames, pedestrian detection applied to the RGB camera failed to detect all humans in the image. Problems concerning occlusion and humans seen from the side or from behind have been observed. However, utilizing a sequence of frames would greatly improve detection performance, as the algorithm most often failed for just a single frame and not for an entire sequence of frames. The stereo camera performed well for detecting humans that protruded from the ground plane. However, the algorithm assumed a certain level of protrusion in order to detect an object. The thermal camera detected all humans when their faces were visible. However, potential problems concern well-insulated clothes that cover an entire body and warm weather where temperature differences are much smaller than in the present recording. The LiDAR detected most humans robustly when they protruded significantly from the ground.

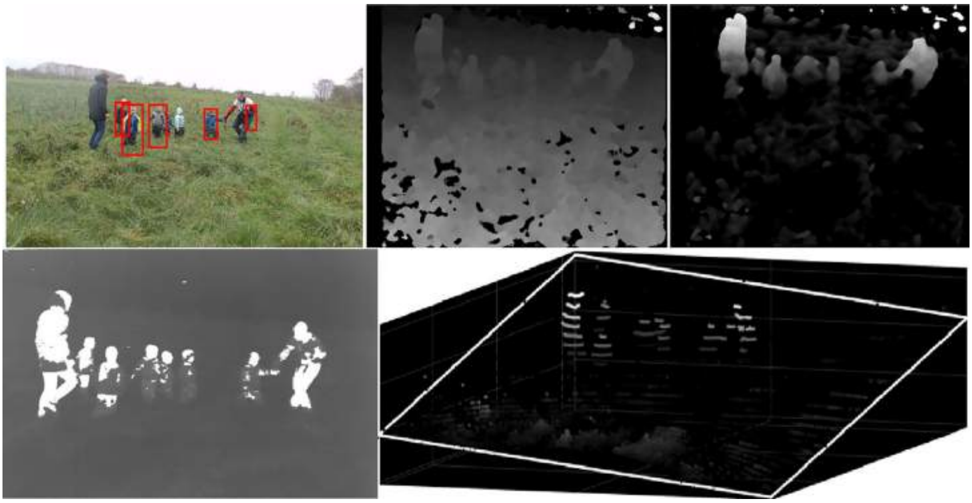


Fig. 9 Human detection. RGB (*top left*), stereo camera disparity map (*top middle*), stereo camera protrusion map (*top right*), thermal camera (*bottom left*), LiDAR (*bottom right*; Christiansen et al. 2015)

Figure 10 depicts three cropped scenarios in high grass with respectively a barrel, a lying child mannequin and a sitting human. The pedestrian detector was able to detect the sitting human as the face and torso were upright and visible. To detect the lying

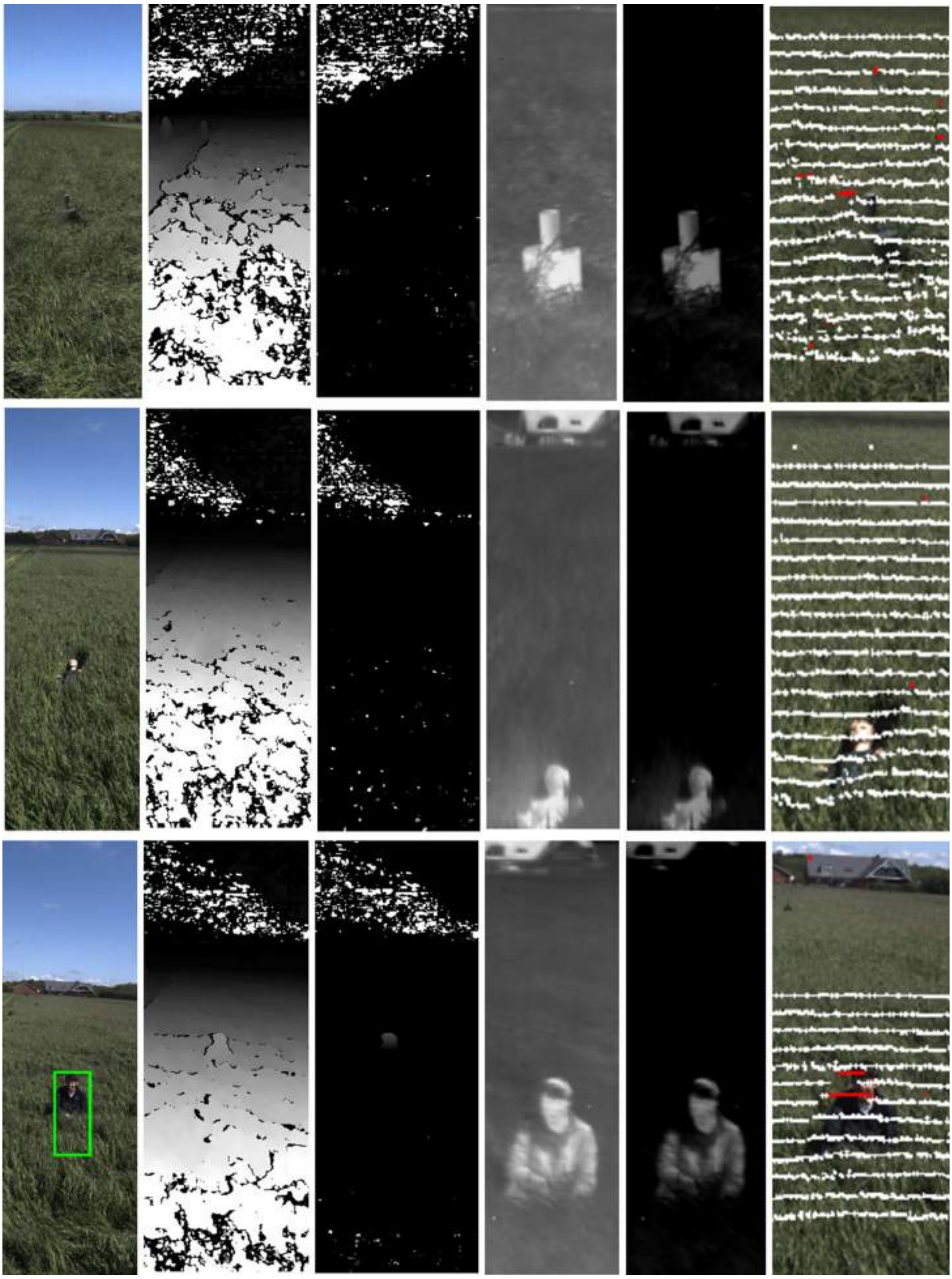


Fig. 10 The three *rows* show respectively a barrel, a lying child mannequin and a sitting human. The *columns* show respectively pedestrian detections, a disparity map from stereo imaging, an object height map based on this, the thermal signature, thermal signature after subtracting the median temperature of the *bottom half* of the image, and the LiDAR projected onto the left stereo camera, where points protruding from the surface by more than 0.25 m are visualized

mannequin, the detector needs to be trained on new data showing humans in similar scenarios. However, the given detector had limited capacity in terms of detecting objects with huge inter-class variation. In the high-grass case, there was a limited reliability of the stereo point cloud which impacted detection performance such that only the sitting human and not the barrel were visible. Exploiting also visual information from the stereo camera should be utilized to improve performance. The LiDAR was more reliable and was able to visualize that both the sitting human and the barrel protruded. The thermal camera achieved robust and reliable detection performance. In scenarios 2–4, all sensors apart from the thermal camera had problems with high grass/crop, presenting a specific challenge that should be addressed in agriculture. The thermal camera will undoubtedly be significantly worse on a warm and sunny day as experienced by Steen et al. (2012) and Serrano-Cuerda et al. (2014). A single sensor is therefore insufficient for detecting all objects reliably, invariant of temperature and lighting changes.

Conclusions

A flexible vehicle-mounted sensor platform was developed for capturing time-stamped data in the agricultural domain using imaging sensors (RGB, thermal and stereo camera), an active sensor (LiDAR) and attitude estimation sensors (RTK GPS and IMU). A registration board was proposed to provide a simple tool for calibrating and registering all sensors in the setup using a single registration board. Authentic data in an actual high grass harvesting operation with a specific focus on human detection were recorded, and an initial evaluation of the potential of different sensor modalities for detecting standing and lying humans including an ISO-barrel was given. Using a common pedestrian detection algorithm, an RGB camera was able to detect upright humans, but degraded rapidly in performance for more complex scenarios. The depth aware sensors (LiDAR and stereo camera) were efficient for detecting objects that protruded significantly above the ground. The LiDAR was invariant towards changing weather and lighting conditions, whereas the stereo camera had the highest resolution making it useful for classifying objects. The thermal camera showed great capabilities in the captured dataset as it was able to detect objects of distinct temperature using a simple procedure that worked well for humans regardless of posture. However, the detection would be much more complicated in environments of higher temperature, where the heat signatures of living objects become indistinct.

The authenticity of the data enabled an initial validation of a human detection system using multiple sensors in a high grass harvesting operation. However, the above arguments and the case study concludes that the use of multiple modalities, more complicated procedures and a fusion of the different modalities is required to achieve robust human detection in high grass harvesting.

Acknowledgements This research is sponsored by the Innovation Fund Denmark as part of the Project “SAFE - Safer Autonomous Farming Equipment” (Project No. 16-2014-0) and “Multi-sensor system for ensuring ethical and efficient crop production” (Project No. 155-2013-6).

Appendix: Thermal–visual registration and evaluation

First a total of 47 thermal and stereo synchronized images were selected from a single calibration recording. For each image, a rectangle area inside the checkerboard was marked manually to specify an image cropping, see Fig. 11. For RGB images, the cropped image was converted to the LAB color space and a Gaussian mixture model separated the pixels into two clusters (copper and non-copper areas). The posterior probability of belonging to one of the Gaussian clusters was determined for all pixels in the original image, see Fig. 12. For thermal images, the cropped image was normalized—transforming pixel

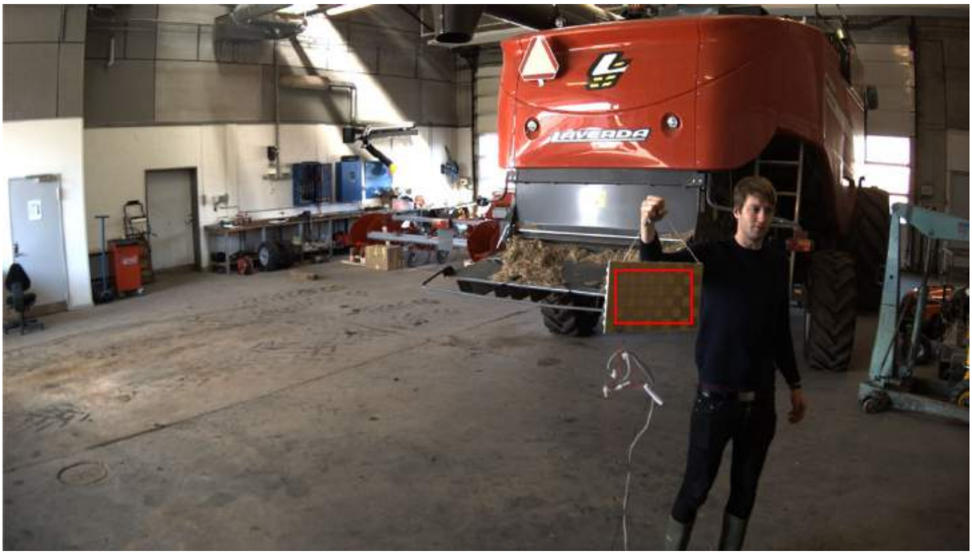


Fig. 11 Image example and a manually marked *rectangle*

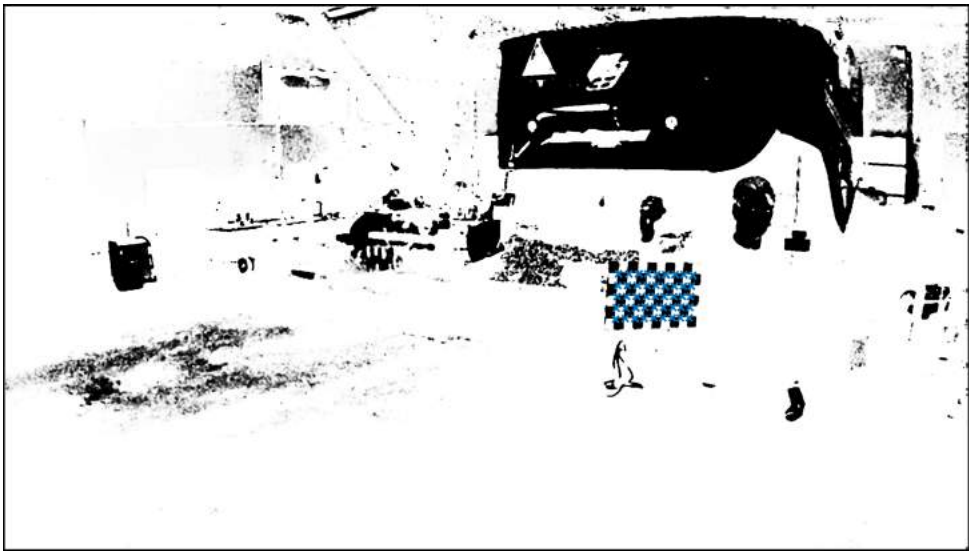


Fig. 12 Posterior probability of belonging to one of the Gaussian clusters for all pixels in the image example. Checkerboard detection is marked with *blue crosses* (Color figure online)

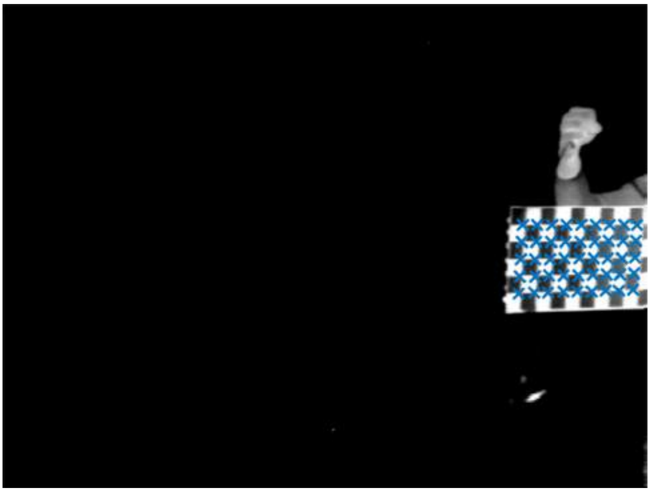


Fig. 13 Thermal image is normalized relative to the checkerboard. Checkerboard detection is marked with *blue crosses* (Color figure online)

values in the range [0 1] by shifting and scaling. The same normalization was applied to the whole thermal image, see Fig. 13. The MATLAB calibration toolbox was able to automatically detect checkerboards of the transformed RGB and thermal images. The calibration toolbox was able to detect the checkerboard in 27 and 43 out of the 45 images for respectively stereo and thermal images. The 27 stereo images were used for calibrating the intrinsic and extrinsic parameters of the stereo camera. The 43 thermal images were used for determining the intrinsic parameters of the thermal camera.

In 25 out of 47 synchronized images, the checkerboard was successfully detected by the MATLAB calibration toolbox for both RGB and thermal images. The toolbox estimated the 3D position of the checkerboard in all 25 images for each camera. The extrinsic parameters of the thermal camera were determined as the least square rigid transformation that mapped the estimated checkerboards from the left RGB camera to the thermal camera (in 3D).

The registration was evaluated on the 25 images to provide a quantitative evaluation of the thermal–visual registration. The camera calibration for the left stereo camera

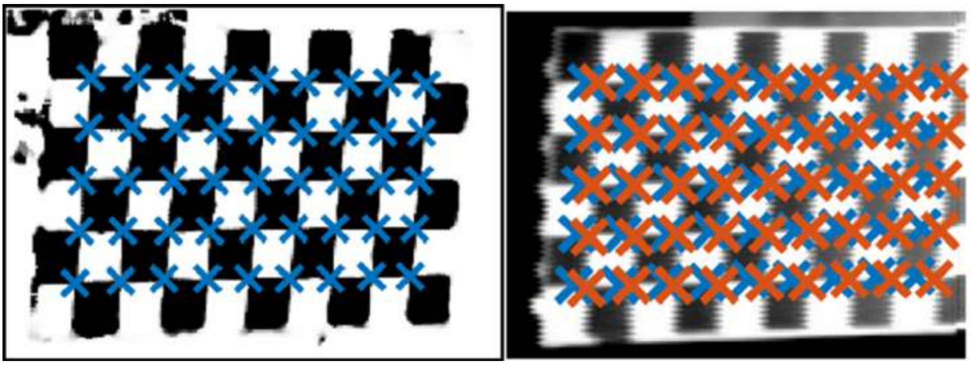


Fig. 14 Zoomed images. *Blue crosses* mark *corners* detected by the MATLAB calibration toolbox for both an RGB image (*left*) and a thermal image (*right*). The *red crosses* (*left*) show how 3D points are projected to the thermal camera (Color figure online)

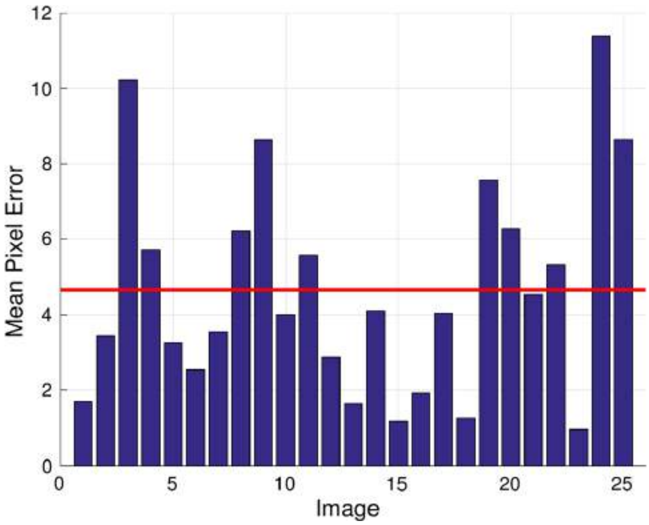


Fig. 15 The mean pixel error for 25 images (*blue bars*) and the mean pixel error across all images (*red line*) (Color figure online)

estimated—as already described—the checkerboard positions in 3D. These positions were then projected to the thermal image using the estimated extrinsic and intrinsic parameters of the thermal camera, see Fig. 4 (right).

The error was determined as the distance between the detected checkerboard and the projected 3D positions. Figure 15 shows the mean pixel error for each of the 25 images and the mean pixel error across all images on 4.66 pixels. The image example used in Figs. 11, 12, 13, and 14 is image 21 with a mean pixel error close to the mean pixel error across all images.

References

Bahnsen, C. (2013). Thermal-visible-depth image registration. Unpublished Master Thesis, Aalborg University, Aalborg, Denmark.

Christiansen, P., Kragh, M., Steen, K. A., Karstoft, H., & Jørgensen, R. N. (2015). Advanced sensor platform for human detection and protection in autonomous farming. *Precision Agriculture, 15,* 291–298.

Christiansen, P., Steen, K. A., Jørgensen, R. N., & Karstoft, H. (2014). Automated detection and recognition of wildlife using thermal cameras. *Sensors, 14*(8), 13778–13793.

CLAAS Steering Systems. (2011). *Tracking control optimisation*. Retrieved 2016, 26 September from http://claas.via-us.co.uk/booklets/gps-steering-systems/download.

Dollar, P., Belongie, S., & Perona, P. (2010). The fastest pedestrian detector in the west. In F. Labrosse, R. Zwiggelaar, Y. Liu & B. Tiddeman (Eds.), *Proceedings of the British machine vision conference* 2010 (pp 68.1–68.11). BMVA Press, Durham University, UK.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM, 24*(6), 381–395.

Freitas, G., Hamner, B., Bergerman, M., & Singh, S. (2012). A practical obstacle detection system for autonomous orchard vehicles. In *2012 IEEE/RSJ international conference on intelligent robots and systems* (pp 3391–3398).

ISO/DIS 18497:2015: *Agricultural and forestry tractors and self-propelled machinery—Safety of highly automated machinery*. Retrieved 2016, 26 September from https://drive.google.com/file/d/0B1ilODNTH9nzRUV2N0JzbklubFU/view.

Johnson, M. J., & Bajcsy, P. (2008). Integration of thermal and visible imagery for robust foreground detection in tele-immersive spaces. In P. Solbrig (Ed.), *Proceedings of the 11th international conference on information fusion* (pp. 1265–1272). Piscataway, USA: IEEE.

Krotosky, S. J., & Trivedi, M. M. (2007). Mutual information based registration of multimodal stereo videos for person tracking. *Computer Vision and Image Understanding, 106*(2–3), 270–287.

McLachlan, G. J., & Basford, K. E. (1988). Mixture models: Inference and applications to clustering. In *Statistics*: *textbooks and monographs*. New York, USA: Dekker.

Paden, B., Cáp, M., Yong, Z. S., Yershov, D., & Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles, 1*(1), 33–55. arXiv:cs.CV/1604.07446v1.

Pilarski, T., Happold, M., Pangels, H., Ollis, M., Fitzpatrick, K., & Stentz, A. (2002). The Demeter System for automated harvesting. *Autonomous Robots, 13,* 9–20.

Rasshofer, R. H., & Gresser, K. (2005). Automotive radar and lidar systems for next generation driver assistance functions. *Advances in Radio Science, 3,* 205–209.

Reina, G., & Milella, A. (2012). Towards autonomous agriculture: Automatic ground detection using trinocular stereovision. *Sensors, 12*(12), 12405–12423.

Rouveure, R., Nielsen, M., & Petersen, A. (2012). The QUAD-AV Project: Multi-sensory approach for obstacle detection in agricultural autonomous robotics. In *International conference of agricultural engineering*. Valencia, Spain: EurAgEng.

Serrano-Cuerda, J., Fernández-Caballero, A., & López, M. (2014). Selection of a visible-light vs. thermal infrared sensor in dynamic environments based on confidence measures. *Applied Sciences, 4*(3), 331–350.

Steen, K. A., Villa-Henriksen, A., Therkildsen, O. R., & Green, O. (2012). Automatic detection of animals in mowing operations using thermal cameras. *Sensors, 12*(6), 7587–7597.

The MathWorks, Inc. (2015). *MATLAB and computer vision system toolbox*. Natick, MA, USA: The MathWorks, Inc.

Wei, J., Rovira-Mas, F., Reid, J. F., & Han, S. (2005). Obstacle detection using stereo vision to enhance safety of autonomous machines. *Transactions of the ASAE, 48*(6), 2389–2397. doi:10.13031/2013.20078.

Yang, L., & Noguchi, N. (2012). Human detection for a robot tractor using omni-directional stereo vision. *Computers and Electronics in Agriculture, 89*, 116–125.

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision, 13*(2), 119–152.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(11), 1330–1334.

Zhao, J., & Cheung, S. S. (2014). Human segmentation by geometrically fusing visible-light and thermal imageries. *Multimedia Tools and Applications, 76*(1), 7361–7389.

Paper 7

Towards Autonomous Plant Production using Fully Convolutional Neural Networks

Peer reviewed
Paper presented at International Conference on Agricultural Engineering (ICPA), June 2016,
Aarhus, Danmark

Towards Autonomous Plant Production using Fully Convolutional Neural Networks

Peter Christiansen[a,*], René Sørensen[a], Søren Skovsen[a], Claes D. Jæger[b], Rasmus Nyholm Jørgensen[a], Henrik Karstoft[a] and Kim Arild Steen[b]

[a] Department of Engineering, Aarhus University, Aarhus, Denmark

[b] AgroIntelli, Aarhus, Denmark

* Corresponding author. Email: pech@eng.au.dk

Abstract

In order for autonomous agricultural vehicles to operate cost efficiently and to be able to operate unsupervised, they must adhere to current EU legislation, and be able to perform automatic real-time path-planning, risk detection and obstacle avoidance. In this context, sensor technologies must be utilized to perceive surroundings and allow autonomous machines to act accordingly.

This paper investigates the perception capabilities of a deep learning approach called "fully convolutional neural network for semantic segmentation" (pixel level classification) in agriculture using an rgb camera sensor. Training a network for semantic segmentation requires the comprehensive task of providing whole scene per-pixel labelling on a large data set. To avoid the task of creating per-pixel labelled data we investigate using a network trained on two already existing databases (ImageNet and Pascal-context) with mostly non-agricultural specific images and classes. A pre-trained network performs pixel wise classification on 59 classes and by remapping the 59 classes to agricultural specific classes (e.g. sky, field, shelterbelts, animal, human and obstacles), we are able to test the network's ability to generalize to agriculture in two case studies: grass mowing and row crop operations.

Based on a small set of 10 per-pixel labelled test images, we show that the network is able to generalize to a grass mowing use cases with an pixel classification accuracy of 95.25%. In the row crop case, the network is less reliable with a classification accuracy of 70.54%. By showing detections of state-of-the-art object detection algorithms (a pedestrian detector and a deep learning object detector), a small qualitative comparison between object detection methods and the semantic segmentation algorithm is made.

Keywords: Deep learning, Autonomous Tractors, Semantic Segmentation, Per-Pixel labelling, Agriculture

1. Introduction

In order for autonomous agricultural vehicles to operate cost efficiently and to be able to operate unsupervised, they must adhere to current EU legislation, and be able to perform automatic real-time path-planning, risk detection and obstacle avoidance. In this context, sensor technologies must be utilized to perceive surroundings and allow autonomous machines to act accordingly.

In the automotive industry, a range of companies (Google, Ford, Uber, Tesla etc.) have demonstrated autonomous vehicles in both prototype and commercial products. Autonomous vehicles in agriculture uses sensor technologies and algorithms found in the automotive industry. However, certain challenges in perception (and path-planning) is specific to agriculture.

For perception, Google Car is highly dependent on both a very detailed static 3D map and under operation expensive laser scanners (Velodyne LiDAR) to get an immediate measurement of its surroundings. A static 3D map is not as feasible in agriculture as the crop is under constant transformation and laser scanners are currently too expensive for farmers. Secondly, depth sensor will hardly detect obstacles that are not protruding the crop surface such as kids, lying humans, hydrants, wells and animals.

Conventional automotive companies evolve from semi- to fully-autonomous by adding affordable features and sensor. Relying on detailed 3D maps is problematic as the car must operate on roads without maps and rely on sensors affordable to consumers and car manufacturers. Mobileye is a leading company in delivering real-time camera-based solution for automotive companies including Audi, Ford and Tesla. However, solutions by Mobileye are not all suited, accessible or trained for agriculture, and algorithms are mostly unpublished. The unpublished results by Mobileye and the limited access for non-automotive industries, makes the actual accuracy performance of Mobileye solutions unclear to researchers, and also if the perception capabilities are comparable to recent deep learning perception algorithms.

Deep learning is an emerging field in Artificial Intelligence/Machine Learning that recently moved the boundary of a computer intelligence and perception. In 2015 deep learning was able to reach human performance in image classification (He, Zhang, and Sun 2015) and speech recognition (Amodei et al. 2015). Especially Convolutional Neural Networks

(CNN) (LeCun et al. 1998) have recently outperformed traditional image recognition methods used for traffic sign recognition(Ciresan, Meier, and Schmidhuber 2012), face detection (Farfade, Saberian, and Li 2015), face recognition (Taigman et al. 2014), image classification (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2014; He, Zhang, and Sun 2015), general object detection (Ren et al. 2015; Redmon et al. 2016; Girshick et al. 2014; He et al. 2014) and semantic segmentation (Long, Shelhamer, and Darrell 8 Mar, 2015; Chen et al. 2015; Torr 2014). Fast and high accuracy CNN-based object detection algorithms have recently been published and open-sourced (Ren et al. 2015; Redmon et al. 2016). Especially YOLO (Redmon et al. 2016) is able to process images at real-time speeds using a high-end GPU on the 20 object types from Pascal VOC (Everingham, Eslami, and Gool 2013). A drawback of object detection algorithms in the context of agriculture, is that not all elements or "stuff" are precisely delimited with a bounding box such as shelterbelt, ground, crop and water. Secondly, object detection algorithms are challenged in agriculture, where obstacles are likely to be heavily or partly occluded by the crop.

Semantic segmentation is an image recognition method that classifies each pixel in the image. In (Long, Shelhamer, and Darrell 8 Mar, 2015) a CNN is converted and modified to a Fully Convolutional Neural Network for Semantic Segmentation (FCS). In (Chen et al. 2015; Torr 2014) Conditional Random Fields are appended to a modified CNN to improve accuracy, though with additional computational cost.

In relation to object detection algorithms, semantic segmentation performs pixel classification and is therefore more suited to detect both obstacles and "stuff" and is presumably less challenged by occlusion.

A drawback of training a semantic segmentation model sufficiently is the requirement for a lot of data with per-pixel level annotations. Transferring the algorithm to agriculture is therefore obstructed by the comprehensive task of making per-pixel level annotations on new data.

This work is a preliminary study demonstrating the perception capabilities of semantic segmentation in agricultural by remapping the 59 predictions from the model in (Long, Shelhamer, and Darrell 8 Mar, 2015) to 11 agriculture specific classes (animal, building, field, ground, obstacle, person, shelterbelt, sky, vehicle, water and unknown). The perception capability is presented by two measures: the classification per-pixel accuracy on 10 test images and a qualitative comparison (by examples) of semantic segmentation and object detection.

The purpose of the publication is to emphasize the powerful perception capability of deep learning semantic segmentation. Traditionally, image recognition algorithms fail to generalize on data not similar to the training data. We show that the comprehensive task of creating training data can in some cases be avoided and that a network trained on general images (PASCAL-Context) is able to generalize to a different context.

2. Materials and Methods

In this section the data used for training and testing FCNN is first presented followed by a description of FCN and the simple remapping approach.

2.1. Data - For training

ImageNet (Berg and Deng 2015) is a image recognition benchmark for image classification, object localization and object detection. The ImageNet benchmark have since 2012 pushed the performance of CNNs and provided a common ground for leading research teams to compete (Google, Microsoft, Baidu). Especially the image classification challenge with an incredible amount of 1.400.000 images with image notation of 1000 different object types.

PASCAL Visual Object Classes (VOC) (Everingham, Eslami, and Gool 2013) is another image recognition benchmark. The benchmark includes a semantic segmentation competition on 20 object classes. In PASCAL-Context (Mottaghi et al. 2014) six in-house annotators have used three months to extend PASCAL VOC with whole scene annotation on 10,103 images, extending the number of object classes from 20 to 407. Figure 1 shows the difference between PASCAL VOC and PASCAL-Context. Note especially annotations of "stuff" such as road, grass and trees is also provided.

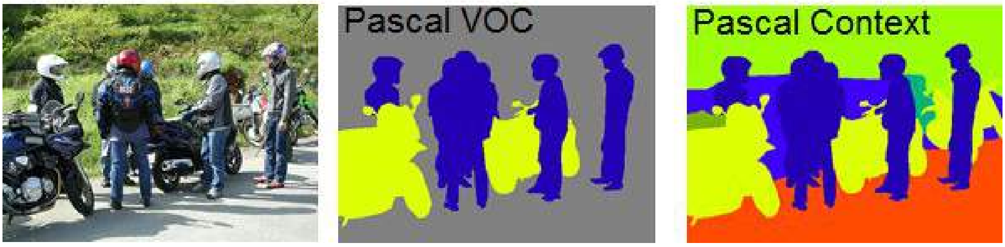


Figure 1. Shows the much denser pixel-annotations provided in PASCAL-Context.

2.2. Data - For testing

A rolling shutter Logitech C920 webcam with a resolution of 1920x1080 and a framerate of 30Hz have been used to record image data. The webcam is placed on a sensor platform including other sensors as described in (Christiansen et al. 2015a; Christiansen et al. 2015b). The sensor platform includes a metal frame with a standard A-frame in the bottom. The A-frame is easily mountable to a tractor and places the camera roughly 2.0m from the ground.

The grass mowing use case is recorded in a 7.5ha grass field near Lem, Denmark in June 2015 under an actual mowing operation. Obstacles are placed in the trajectory of the tractor to simulate collision situations (Christiansen et al. 2015a). The row crop operation use case is recorded in mid-September 2015 in a low row crop maize field in Foulum, Denmark. Unlike the moving use case, objects are placed just outside the tractor trajectory allowing them to remain static. A total of 10 images - 5 from each use case - have been selected and roughly annotated.

2.3. Methods

Semantic segmentation is described in the first section followed by a section describing remapping of model predictions.

2.3.1. Fully Convolutional Neural Network for Semantic Segmentation

A traditional CNN performs image classification, thus it can only take a fixed sized input image and output a single prediction/label. A CNN trained on e.g. faces can only tell if there is a face or not in the image as shown in Figure 2. A CNN can be transformed into a fully convolutional neural network (FCNN) by converting the fully connected layers into convolutions. A FCNN is able to forward larger images through the network and output a grid of prediction, thus providing information on both the object type and object location in the image. Converting a CNN to a FCNN for e.g. face image classification, will provide a coarse heat map as presented in Figure 3, showing the position of a specific sized face in the image. Training the CNN to recognize other objects will provide multiple heat maps one for each object type.

Figure 2. CNN training examples for face image classification. Respectively a negative and positive sample.

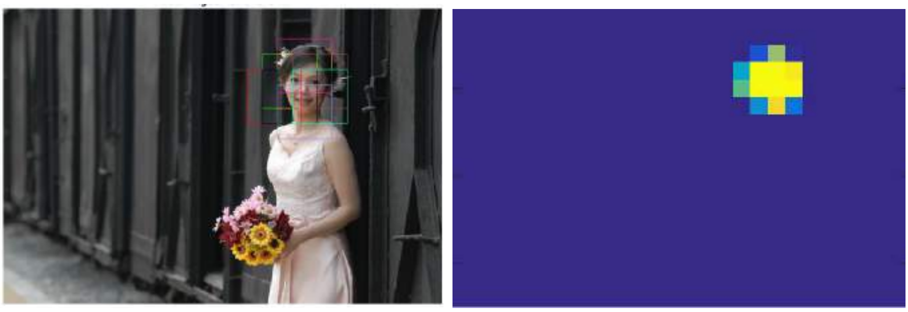


Figure 3. A fully convolutional neural network provides a grid of predictions or a heat map as output.

The above described principle is used in (Long, Shelhamer, and Darrell 8 Mar, 2015) to perform semantic segmentation with FCNN. The network is based on a very deep CNN (Simonyan and Zisserman 2014) (VGG) with 16-layers trained for image classification on ImageNet (Berg and Deng 2015). The CNN network is transformed into a FCNN by discarding the classification layer of VGG and converting the fully connected layers to convolutional layers. A 1x1 convolution is appended with a channel for each object type. The specific structure of VGG allows it to provide classifications for every 32 pixels. By adding a deconvolutional layer, the network will upsample the heatmap to the size of the original image. The network is now able to perform end-to-end training on semantic segmented images. This is defined as a FCN-32 network. To get denser spatial predictions, a 1x1 convolution with a channel for each object is appended to the output of two previous pooling layers with respectively a stride of 16 and 8. The output of earlier pooling layers will provide weaker predictions, but better spatial precision. By fusing the output layers with respectively a stride of 32, 16 and 8, the combination of strong predictions and the refined spatial precision, is found to improve the overall network accuracy. This network is defined as a FCN-8 network. Three FCN-8 models are provided by (Long, Shelhamer, and Darrell 8 Mar, 2015) one trained on 21 classes (PASCAL Voc object including a background class) and two models on PASCAL-Context for both the 33 and 59 most frequent classes. As described in (Long, Shelhamer, and Darrell 8 Mar, 2015), the algorithm is able to roughly process images with 4Hz using a high-end GPU. Using the principles from (Han, Mao, and Dally 2015), the memory footprint and the processing requirements can be reduced for a CNN without damaging accuracy, thus making it suitable for real-time applications.

2.3.2. Prediction mappings

Preferably a network is retrained only on PASCAL-Context classes relevant to agriculture. Alternatively, all object classes from PASCAL-Context are mapped to a few agriculture super-categories and retrained. E.g. dog, cat, cow and horse are all mapped to animal, or road, ground, sand, floor is all mapped to ground. However, as a preliminary study we perform simple mapping of predictions provided by a FCN-8 network to the following 11 agricultural super-categories; *animal, building, field, ground, obstacle, person, shelterbelt, sky, vehicle, water, and unknown*. The result of a prediction and remapping is presented in the Figure 4.

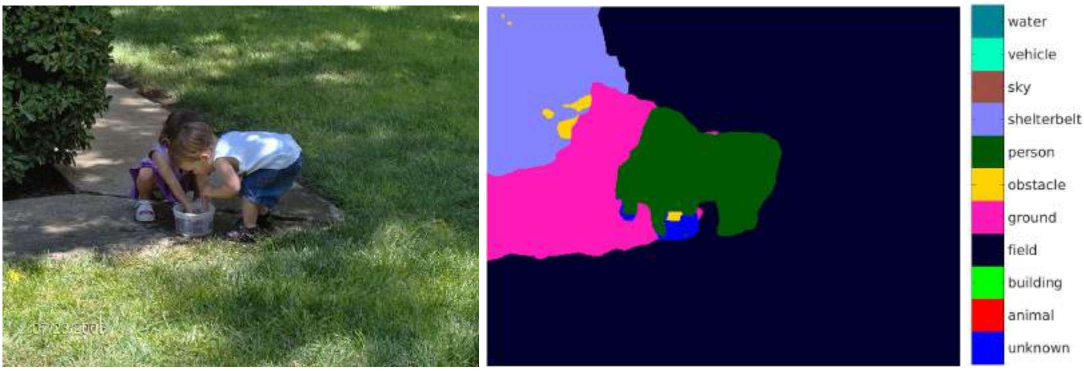


Figure 4. Left: Input image. Right: Result after remapping FCN-8 predictions to agriculture specific classes.

3. Results and Discussion

3.1. Results

Using the ground truth annotations for the 10 test images, the overall classification accuracy is 82.81%. Evaluating the the grass and row crop individually shows - with a classification accuracy of respectively 95.25% and 70.54% - a significant spread between the two use cases. The spread is also clearly demonstrated in Figure 5-9, where the grass and the row crop cases are presented in respectively Figure 5-6 and Figure 7-9. To make a qualitative comparison between semantic segmentation and object detection algorithms the left images in each figure show the input image to the FCN-8 algorithm including detections performed by YOLO (Redmon et al. 2016) and a pedestrian detector (Nam, Dollár, and Han 2014). Both object detection algorithms are close to real-time performance.

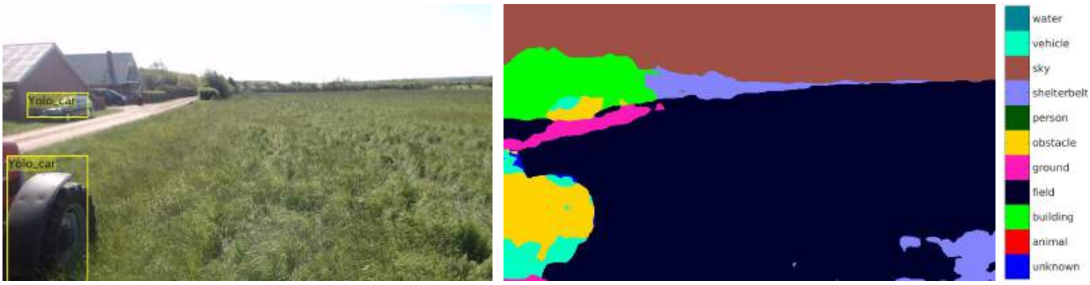


Figure 5. Test image in grass. Semantic segmentation detects field, road, building, shelterbelt, sky, bits of vehicle. However, the tractor is classified as both vehicle, obstacle and bits of unknown. A bit of high grass to the right is classified as shelterbelt. The vehicle right next to house is mostly detected as an obstacle. YOLO is able to detect both tractor and vehicle.

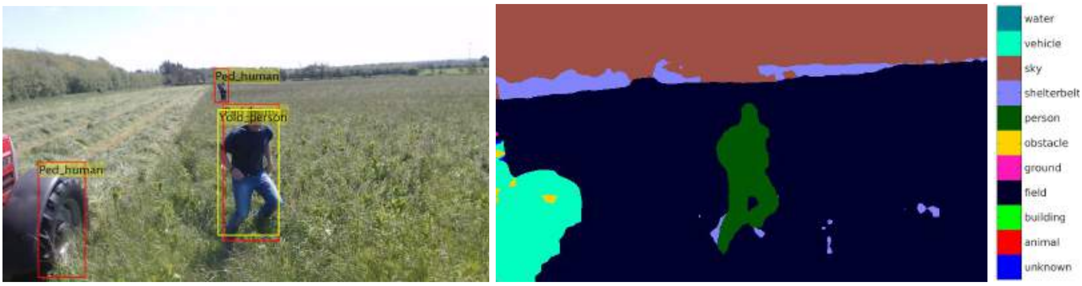


Figure 6. Test image in grass. Semantic segmentation detects field, tractor, sky, shelterbelt and person. However, bits of the field are classified as shelterbelt, a section of the distant shelterbelt is classified as sky and the distant human is classified as field. YOLO detects only the first person. The pedestrian detector detects the close and distant human, but provides also a false positive on the wheel.

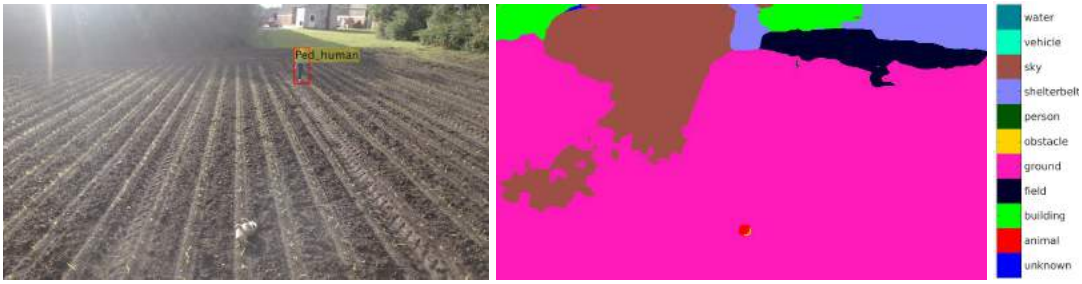


Figure 7. Test image in row crop. Semantic segmentation detects ground, shelterbelt, animal, field, building and just a bit of person. However, a large section of the field and shelterbelt is classified as sky and the dark area in the top left corner is classified as building. The top left corner - with very low contrast - is presumably classified as building as many

images in PASCAL-Context are taken inside houses with low contrast walls. Finally, the pedestrian detector and not YOLO detects a human.

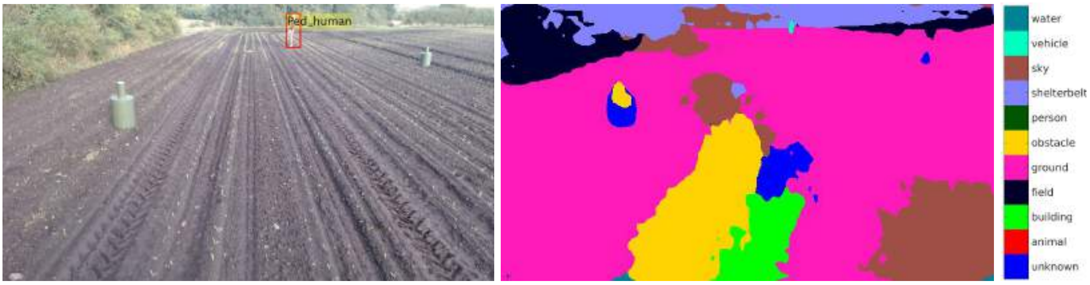


Figure 8. Test image in row crop. Large areas of shelterbelt and ground is classified correctly and the green barrels are classified as unknown or obstacles. The shelterbelt to the left is - as a reasonable guess - classified as field. However, large areas of ground are classified as obstacle, building and sky. The distant person is only detected by the pedestrian detector.

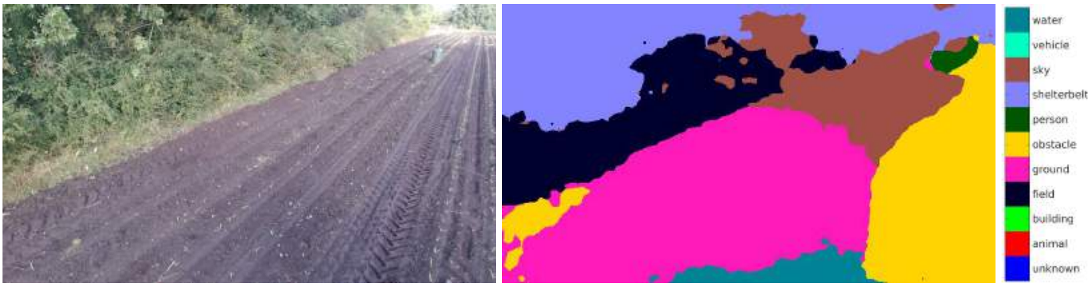


Figure 9. Test image in row crop. Large areas of shelterbelt and ground are classified correctly. The shelterbelt to the left is - as reasonable guess - again classified as field. However, large areas of ground are classified as obstacle, water, sky and person.

3.1.1. Discussion

This preliminary study uses a simple remapping to show the application of deep learning semantic segmentation for autonomous vehicles in agriculture. For an algorithm trained on a completely different data set, a classification accuracy of 95.25% and the presented image examples, show very convincing perceptive capabilities for a grass mowing use case. The row crop use case is less reliable with a classification accuracy of 70.54%. However, the inferior performance in row crops can be explained by the data from PASCAL-Context that do not contain a row crops class. However, we have showed that deep learning semantic segmentation trained on PASCAL-Context is able to generalize to a grass mowing use case, thus allowing us to avoid the comprehensive task of making per-pixel labelling. The preliminary study encourages us to train a new network only on agriculture specific classes from the PASCAL-Context data or alternatively remap all 407 classes to a few agricultural specific classes prior to training. Finally, whole scene annotations of agricultural images would provide even better results.

The image examples show that the object detection algorithms provide fewer misclassifications compared to semantic segmentation. The pedestrian detector is better at detecting people at further distances. However, the YOLO detector is able to detect multiple object types.

Semantic segmentation is able to detect animal, human and vehicle obstacles - as an object detector. However, the benefit of semantic segmentation is both its ability to classify elements that are not precisely delimited with a bounding box and that it provides much denser information of the environment. This information can be used to detect non-traversable areas such as shelterbelts, water, buildings and even unknown obstacles as the barrel. However, a classification of traversable areas such as road, ground or field is also favorable to autonomous farming vehicles when performing navigation and path-planning. To deploy semantic segmentation in a real application, the processing requirements must be evaluated. The current algorithm runs with 4 Hz using a high-end GPU. A smaller CNN network with lower resolution input images and the principles described in (Han, Mao, and Dally 2015), is able to improve its real-time performance.

Semantic segmentation is as well as most visual camera based solutions not fully reliable and struggles to detect far away elements. A visual camera is also sensitive to weather conditions (rain, fog and snow) and illumination such as direct or dim light (night) (Christiansen et al. 2015b). In the context of agriculture with obstacles below the crop surface, an advantage for a monocular camera is that objects only needs to be visible and not necessarily protruding.

An autonomous vehicle in agriculture should - as in the automotive industry - rely on multiple algorithms and sensor technologies to get more reliable perception of especially visually hidden or camouflaged obstacles and obstacles at far ranges (Christiansen et al. 2015b). However, the low cost of a camera and the power of deep learning perceptive algorithms makes it consumer affordable for unsupervised autonomous vehicles in agriculture. In (Hansen et al. 2016) - also presented at the CIGR conference - the outcome of this work is fused with other sensor technologies and algorithms using occupancy grid maps to detect static obstacles in an agricultural grass field.

4. Conclusions

This preliminary study uses a simple remapping to show the application of deep learning semantic segmentation for autonomous vehicles in agriculture. For an algorithm trained on a completely different data set, a classification accuracy of 95.25% and the presented image examples, show very convincing perceptive capabilities for a grass mowing use case. The row crop use case is less reliable with a classification accuracy of 70.54%. The perception benefits of semantic segmentation compared to object detection has been described and demonstrated using image examples.

Acknowledgements

To Timo Korthals from Bielefeld University for providing a segmentation labelling tool. This research is sponsored by the Innovation Fund Denmark as part of the project "SAFE - Safer Autonomous Farming Equipment" (project no. 16-2014-0).

References

Amodei, Dario, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, et al. 2015. "Deep Speech 2: End-to-End Speech Recognition in English and Mandarin." *arXiv [cs.CL]*. arXiv. http://arxiv.org/abs/1512.02595.

Berg, Alex, and J. Deng. 2015. "Imagenet Large Scale Visual Recognition Challenge 2015." *Challenge* . http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Large+Scale+Visual+Recognition+Challenge+2010#2.

Chen, Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 2015. "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs." *Iclr*, 1–12. http://arxiv.org/abs/1412.7062.

Christiansen, P., Mikkel Kragh Hansen, Kim Steen, Henrik Karstoft, and Rasmus Nyholm Jørgensen. 2015a. "Platform for Evaluating Sensors and Human Detection in Autonomous Mowing Operations." *(Only Submitted) Precision Agriculture, Special Issue ECPA*.

Christiansen, P., M. K. Hansen, K. A. Steen, H. Karstoft, and R. N. Jørgensen. 2015b. "Advanced Sensor Platform for Human Detection and Protection in Autonomous Farming." In *Precision Agriculture '15*, 291–98. doi:10.3920/978-90-8686-814-8_35.

Ciresan, D., U. Meier, and J. Schmidhuber. 2012. "Multi-Column Deep Neural Networks for Image Classification." *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3642–49. doi:10.1109/CVPR.2012.6248110.

Everingham, Mark, Sma Eslami, and Luc Van Gool. 2013. "The Pascal Visual Object Classes Challenge–a Retrospective." *Homepages.Inf.Ed.Ac.Uk*. doi:10.1007/s11263-014-0733-5.

Farfade, Sachin Sudhakar, Mohammad Saberian, and Li-Jia Li. 2015. "Multi-View Face Detection Using Deep Convolutional Neural Networks." *Cornell University Library*. http://arxiv.org/abs/1502.02766v2.

Girshick, Ross, Jeff Donahue, Trevor Darrell, U. C. Berkeley, and Jitendra Malik. 2014. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." *Cvpr'14*, 2–9.

doi:10.1109/CVPR.2014.81.

Hansen, Mikkel Kragh, Peter Christiansen, Timo Korthals, Thorsten Jungeblut, Henrik Karstoft, and Rasmus Nyholm Jørgensen. 2016. "Multi-Modal Obstacle Detection and Evaluation of Evidence Grid Mapping in Agriculture." In . Aarhus University.

Han, Song, Huizi Mao, and William J. Dally. 2015. "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding." http://adsabs.harvard.edu/abs/2015arXiv151000149H.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2014. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition." *arXiv Preprint arXiv …* cs.CV: 1–14. doi:10.1109/TPAMI.2015.2389824.

He, Kaiming, Xiangyu Zhang, and Jian Sun. 2015. "Deep Residual Learning for Image Recognition." *arXiv [cs.CV]*. arXiv. http://arxiv.org/abs/1512.03385.

Krizhevsky, A., I. Sutskever, and Ge Hinton. 2012. "Imagenet Classification with Deep Convolutional Neural Networks." *Advances in Neural Information Processing Systems*, 1097–1105.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. "Gradient Based Learning Applied to Document Recognition." *Proceedings of the IEEE* 86 (11): 2278–2324.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell. 8 Mar, 2015. "Fully Convolutional Networks for Semantic Segmentation"

Mottaghi, Roozbeh, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. 2014. "The Role of Context for Object Detection and Semantic Segmentation in the Wild." In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 891–98. IEEE. doi:10.1109/CVPR.2014.119.

Nam, Woonhyun, Piotr Dollár, and Joon Hee Han. 2014. "Local Decorrelation For Improved Detection." *Advances in Neural Information Processing Systems*, 1–9. http://papers.nips.cc/paper/5419-local-decorrelation-for-improved-pedestrian-detection.

Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. "You Only Look Once: Unified, Real-Time Object Detection." http://arxiv.org/abs/1506.02640v3.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. 2015. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." *arXiv [cs.CV]*. arXiv. http://arxiv.org/abs/1506.01497.

Simonyan, Karen, and Andrew Zisserman. 2014. "Very Deep Convolutional Networks for Large-Scale Image Recognition." *arXiv [cs.CV]*. arXiv. http://arxiv.org/abs/1409.1556.

Taigman, Yaniv, Marc Aurelio Ranzato, Tel Aviv, and Menlo Park. 2014. "DeepFace : Closing the Gap to Human-Level Performance in Face Verification," June.

Torr, Philip H. S. 2014. "Conditional Random Fields as Recurrent Neural Networks." *arXiv Preprint*. http://arxiv.org/abs/1502.03240v1.

Paper 8

(Not open-access) Advanced Sensor Platform for Human Detection and Protection in Autonomous Farming

Peer reviewed
Paper presented at the 10th European Conference on Precision Agriculture (ECPA), 2015 July, Tel Aviv, Israel

Advanced sensor platform for human detection and protection in autonomous farming

P. Christiansen[1], M. Kragh[1,†], K. A. Steen[1], H. Karstoft[1], R. N. Jørgensen[1]
[1]*Department of Engineering – Signal Processing, Faculty of Science and Technology, Aarhus University, Finlandsgade 22, 8200 Aarhus N, Denmark*
[†]Corresponding author: mkha@eng.au.dk

Abstract

The concept of autonomous farming concerns automatic agricultural machines operating safely and efficiently without human intervention. In order to ensure safe autonomous operation, real-time risk detection and avoidance must be performed. This paper presents a flexible vehicle-mounted sensor platform for recording positional and imaging data with a total of seven sensors. Different imaging modalities are chosen for robust detection performances in a variety of weather and lighting conditions. Different algorithms applied to recordings from a grass-harvesting case study show that it is possible to detect humans, whereas small animals located in front of the vehicle represent a much greater challenge.

Keywords: safe farming, sensor platform, object detection, computer vision

Introduction

Current technology is capable of automatically navigating and operating agricultural machinery, such as tractors and harvesters, efficiently and more precisely compared to manual operation. However, a crucial deficiency in this technology concerns the safety aspects. In order for an autonomous vehicle to operate safely and be certified for unsupervised operation, it must perform automatic real-time risk detection and avoidance in the field with high reliability.

Robust risk detection imposes a number of challenges for the sensor platform. Varying weather and lighting conditions influence the image quality of sensor modalities in different ways, and thus no sensor is single-handedly capable of detecting objects reliably under all conditions. Active sensors such as radar and LiDAR, and passive sensors such as RGB camera, stereo camera and thermal camera have different strengths and weaknesses concerning weather, lighting, range and resolution, and therefore a variety of these sensors are needed to cover all scenarios (Rasshofer & Gresser 2005). In addition, pose estimation sensors such as accelerometers, gyroscopes and GPS are needed for estimating the vehicle position, velocity and orientation and for synchronizing and registering subsequent frames acquired from the imaging sensors.

Today, driver assistance systems are available for a large number of modern passenger cars, and completely autonomous vehicles operating in urban and sub-urban environments are emerging for experimental usage (Luettel et al. 2012). In the agricultural sector, a variety of machines have been operating autonomously for a decade using either precise GPS coordinates and/or cameras detecting structures in the field (CLAAS Steering Systems 2011). Efforts are made to fully automate the process in a driverless solution, but safety aspects currently prevent authorization for this. For instance the QUAD-AV project has investigated microwave radar, stereo vision, LiDAR and thermography for detecting obstacles in an agricultural context (Rouveure et al. 2012). Within the project, a detailed study of stereo vision has shown promising results on ground/non-ground classification (Reina & Milella 2012).

The paper describes a flexible vehicle-mounted sensor platform. The sensor platform records imaging data and vehicle position for a moving vehicle using three passive imaging sensors, two active sensors and two pose/position estimation sensors. The sensor platform is designed to record simultaneous data from all sensors, thus preparing for subsequent offline processing. Recordings from a grass-harvesting case study are documented. In the study, different objects including humans of different sizes, appearances and postures, as well as different animals are placed in front of the setup and detected automatically. Based on different object detection algorithms carried out on the imaging sensors, an initial evaluation of the different sensors is given.

Sensors

An overview of the strengths and weaknesses of the selected imaging and active sensors are presented in Table 1. The qualities are evaluated individually and under various conditions.

Table 1. Strengths and weaknesses of sensors.

	Name	RGB	Stereo	Thermal	LiDAR	UWB Radar
Specification	Range	medium	medium	medium	long	medium
	Resolution	+	+	-	-	-
	Depth information	-	+	-	+	+
	Heat information	-	-	+	-	-
	Color information	+	+	-	-	-
	Cost	low	medium	medium	high	medium
Robustness	Light changes	-	-	+	+	+
	Weather changes	-	-	-	+	+
	Camouflaged objects	-	+	+	+	+
	Protruding objects	-	+	-	+	+
	Non-protruding objects	+	-	+	-	-

An *RGB camera* captures the modality of visible light. The sensor is useful for identifying the perceived objects as it provides visual characteristics such as texture, color and shape in high resolution at low cost. It is invariant to protrusion, meaning that non-protruding objects such as small animals, a fallen human or humans/animals in high crops are still visible. However, visual characteristics are affected by weather conditions (rain, fog and snow) and illumination such as dim light (night) or direct light (causing shadows). An RGB camera is not able to exploit depth information to emphasize protruded objects and the lack of depth makes the positioning of objects in 3D space difficult.

A *stereo* camera enables 3D imaging data (depth and color information). Depth and color information are registered and the sensor is thus able to exploit the advantages of both modalities. Depth information can be used to see protruding objects and visually camouflaged animals easily while determining the position of an object relative to the vehicle. In this way, depth-aware algorithms can abstract from the very different visual characteristics of objects (shape, color and texture) creating simple detection algorithms. Like the RGB camera, the stereo camera is sensitive to illumination and weather conditions, although the depth information is in some cases still retrievable.

A *thermal camera* is an imaging sensor that captures heat radiation represented by intensities (temperatures) to form a monochromatic image. A thermal camera perceives objects of distinct temperatures, making it ideal for detecting living objects in temperate and colder climates, and even in foggy weather (Serrano-Cuerda et al. 2014). A key ability is that the sensed data are unaffected by non-protruding or visually camouflaged animals and that the distinctness of living objects becomes more apparent at night. However, these capabilities are much affected by the ambient temperature as living objects become indistinct when the temperature difference between the objects and the background becomes small (Serrano-Cuerda et al. 2014). The cost of a well-performing and high resolution thermal camera is very high, but low cost cameras are emerging. Object recognition capabilities are low due to a limited resolution and limited visual characteristics.

A *LiDAR* measures range data to a set of surrounding points and generates a point cloud where each point is represented by a 3D position and a reflection intensity. The LiDAR is a high cost sensor, but has dropped significantly in price in recent years. Compared to a stereo camera the LiDAR provides very exact depth information at further range and captures up to 360° horizontally. It is invariant to illumination, temperature and camouflage. The lack of visual and thermal information makes recognition of objects difficult and non-protruding objects are almost or fully undetectable.

A *radar* measures range and/or velocity information of objects by transmitting radio waves and measuring object reflections. A variety of radar technologies exist with both low and high costs. Depending on object materials and sizes, different radar frequencies are optimized for different applications. For human detection applications, ultra-wideband (UWB) short range radar operating at a few GHz is common. Radar is invariant towards changing temperature and light conditions.

Physical design

The sensor platform consists of seven sensors and a controller mounted on a common rack. The left side of Figure 1 shows the rack mounted on a tractor and the right side shows the physical placement with antennas and inertial measurement unit (IMU) at the top, sensors in the middle and the controller at the bottom. The horizontal profile in the middle is adjustable in height and angle such that the imaging and active sensors can be oriented at a downward angle depending on the vehicle height. A standard A-frame is mounted at the bottom of the rack to enable easy mounting on tractors. The A-frame is mounted with dampers for absorbing internal engine vibrations from the vehicle to reduce the amount of mechanical noise acting on the sensors. The LiDAR protrudes from the other sensors such that it has an unobstructed 180° forward field of view.

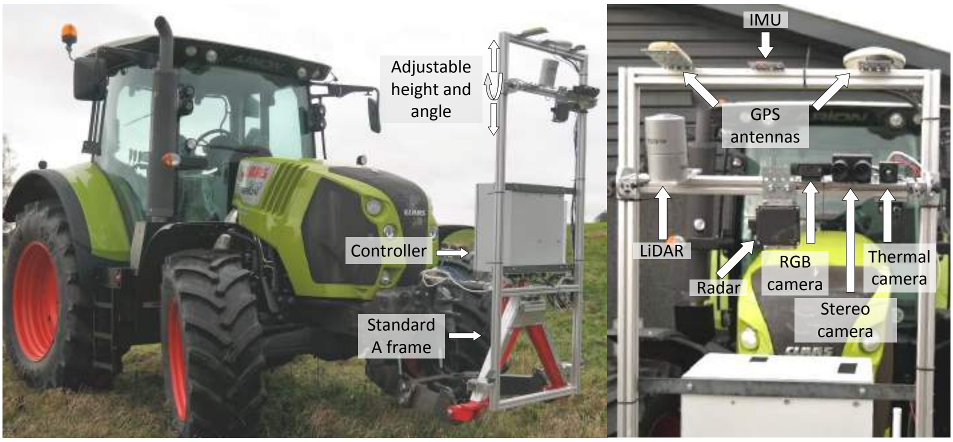


Figure 1. Sensor frame including controller.

Figure 2 presents the specific sensors and the controller used in the setup. A Logitech (Newark, California, USA) C920 webcam providing 1920×1080 pixels at 30 fps is used as the RGB camera. The stereo camera is a high dynamic range camera with logarithmic, global shutter New Imaging Technology (Paris, France) NSC1003 CMOS sensors providing 1280×1024 pixels at 25 fps. The camera uses 12-bit GRBG Bayer pixel format. The thermal camera is a shutterless Tonbo Imaging Inc (East Palo Alto, California, USA) HawkVision analog IR camera providing 640×480 pixels at 25 fps. The LiDAR is a 32-beam Velodyne (Morgan Hill, California, USA) HDL-32E laser scanner providing 70,000 points at 10 Hz with 1-100 m range. The radar is a 76 GHz Delphi ESR radar with 0.5-80 m range. The GPS is a Trimble (Sunnyvale, California, USA) AG GPS361 Real Time Kinematic (RTK) GPS enhancing the precision of GPS up to centrimetre-level accuracy. The IMU is a Vectornav (Dalla, Texas, USA) VN-100 providing synchronized 3-axis accelerometers, gyros, magnetometers and a barometric pressure sensor. The data-collecting controller is a Conpleks Robotech Controller 701. It is an embedded computer with external interfaces for all sensors that uses ROS-middleware (Robot Operating System) to easily integrate all sensors in a common framework.

System architecture

Figure 2 further illustrates the connections and bandwidths between the sensors and the controller. In ROS, each sensor is given its own node (an executable file) that is responsible for publishing one or more topics. For instance, the IMU has its own node including hardware specific drivers, and it publishes different topics related to the readings of the accelerometer, the gyroscopes and the magnetometers. For each topic, the node can send messages containing sensor data whenever a new sensor-reading is available. Each node is connected to the ROS Master which handles interactions between nodes and supplies all messages with exact timestamps. Using the rosbag package (Dirk n.d.), a recording of all desired topics (and all associated messages) to a single rosbag data-file can be obtained.

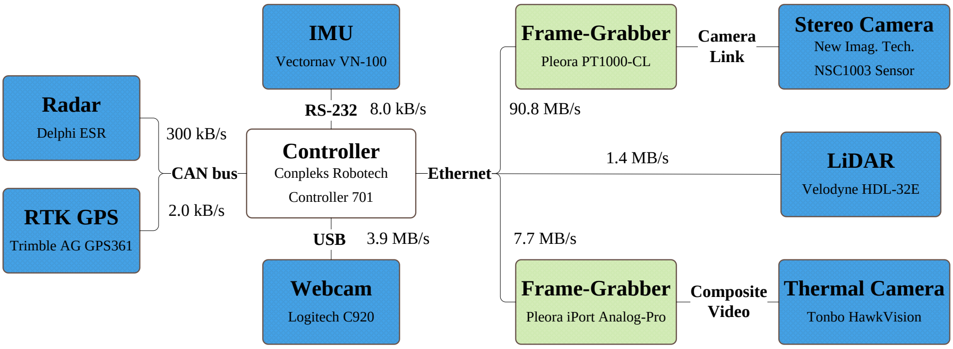


Figure 2. System overview illustrating bandwidths and interfaces between sensors, converters and the controller.

Signal processing

In order to experimentally evaluate detection performances of the different sensors in an agricultural environment, preliminary tests using different object detection algorithms have been carried out on the different imaging and active sensors.

Using only an RGB camera for detecting all possible obstacles in the field is complex and difficult and requires a very large dataset with many representations of each object. Constraining detection to only humans provides a more realistic case in this preliminary study. The RGB camera is therefore processed using a state-of-the-art pedestrian detection algorithm (Dollar et al. 2010). The stereo camera has been calibrated with a stereo calibration algorithm using a checkerboard pattern (Zhang 2000). Subsequently, a ground plane is estimated on the acquired point cloud using the RANSAC algorithm (Fischler & Bolles 1981), and points that lie above this ground plane with a certain threshold are clustered. The LiDAR data is processed using ground plane estimation and clustering of points not belonging to the ground (Moosmann et al. 2009). Clusters with more than 30 points are detected as objects. The thermal camera is processed by thresholding the (temperature-related) intensities by a constant value above the median intensity of the image (Christiansen et al. 2014). Subsequent connected components analysis is used for extracting only components that exceed a certain area. The radar was unfortunately malfunctioning during the data acquisition. Therefore no radar data is available for processing and evaluation.

Results and discussion

Data from six sensors have been recorded in a grass-harvesting case study performed in Denmark in early November. These comprise an RGB camera, a stereo camera, a thermal camera, a LiDAR, a GPS and an IMU. The radar sensor described above unfortunately malfunctioned during the recordings and is therefore omitted in the experimental evaluation.

In the following, two recordings are evaluated including 1) humans of different sizes, appearances and postures and 2) small animals placed in front of the setup.

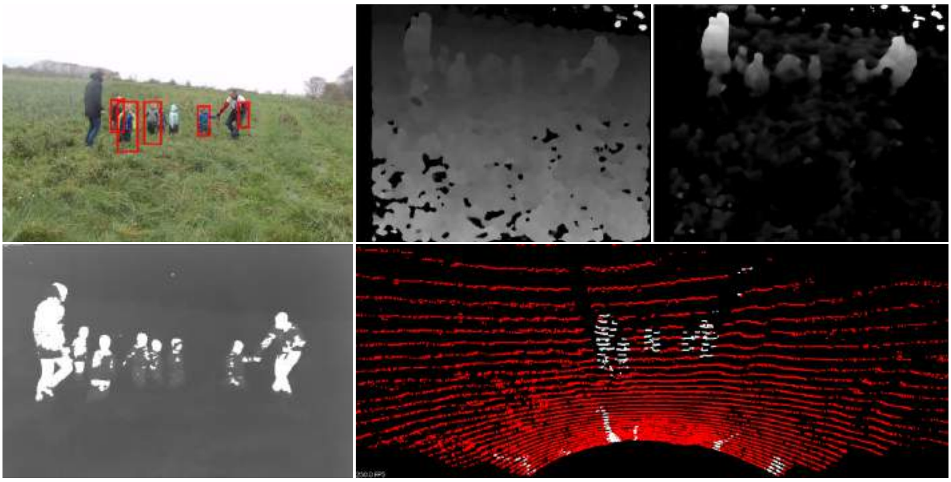


Figure 3. Detection of humans. RGB camera (top left), stereo camera disparity map (top middle), stereo camera protrusion map (top right), thermal camera (bottom left), LiDAR (bottom right).

Figure 3 depicts the human detection performances evaluated at single, synchronized frames for the RGB camera, the stereo camera, the thermal camera and the LiDAR. At the top left, the RGB camera is shown with bounding boxes indicating results of the pedestrian detection algorithm. In the top middle, the disparity map of the stereo camera is shown and, at the top right, a protrusion map indicating objects that protrude from the ground plane is visualized. At the bottom left, the thermal camera is shown with overlaid thresholded components and, at the bottom right, the LiDAR data is visualized with a ground plane and clustered objects (white).

Using only single frames, pedestrian detection applied on the RGB camera fails to detect all humans in the image. Problems concerning occlusion and humans seen from the side or from behind have been observed. However, utilizing a sequence of frames would greatly improve detection performance, as the algorithm most often fails for just a single frame and not for an entire sequence of frames. The stereo camera performs well for detecting humans that protrude from the ground plane. However, the algorithm assumes a certain level of protrusion and a flat surface in order to detect an object. The thermal camera detects all humans when their faces are visible. However, potential problems concern well insulated clothes that cover an entire body and warm weather where temperature differences are much smaller than in the present recording. Using the LiDAR clustering algorithm, most humans are detected robustly when they protrude significantly from the ground. However, problems concerning noise near the sensor due to a higher point density must be solved to avoid false alarms.

Figure 4 depicts animal detection capabilities of a rabbit and a hen. In this scenario, only the thermal camera was capable of detecting the animals. Obviously pedestrian detection applied to the RGB camera is incapable of detecting animals, and since both the algorithms of the stereo camera and LiDAR rely on significantly protruded objects, these modalities both fail to detect small animals. It is therefore clear that more advanced and task-specific algorithms must be investigated for the RGB

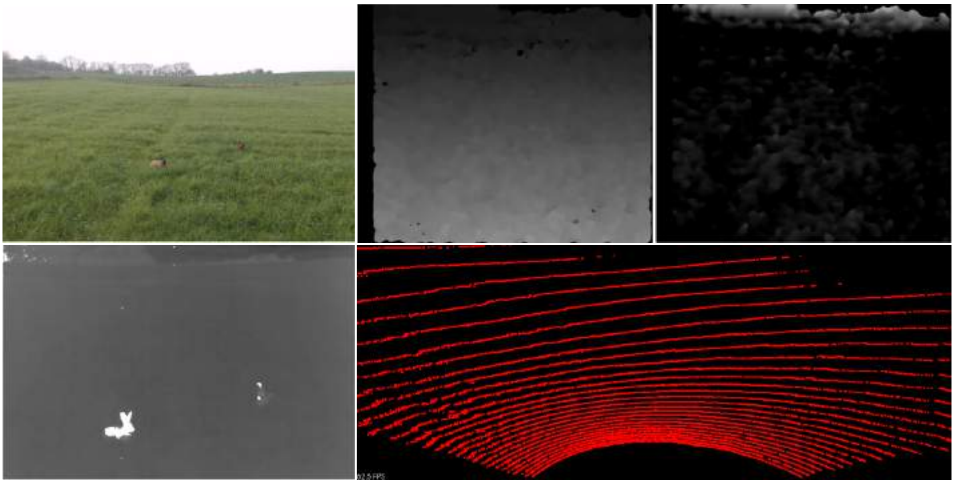


Figure 4. Detection of animals (rabbit and hen). RGB camera (top left), stereo camera disparity map (top middle), stereo camera protrusion map (top right), thermal camera (bottom left), LiDAR (bottom right).

camera, the stereo camera and the LiDAR. Although the thermal camera achieves robust and reliable detection performance for both humans and animals in this study, the results would undoubtedly be significantly worse on a warm and sunny day as reported by (Steen et al. 2012) and (Serrano-Cuerda et al. 2014). A single sensor is therefore insufficient for detecting all objects reliably invariant of temperature and lighting changes.

Conclusion

A flexible vehicle-mounted sensor platform has been developed for capturing time stamped data in the agricultural domain using imaging sensors (RGB, thermal and stereo camera), active sensors (LiDAR and radar) and pose estimations sensors (RTK GPS and IMU). Based on a case study in grass fields, an initial evaluation of the potential of different sensor modalities for detecting humans and animals is given. Using a common pedestrian detection algorithm, an RGB camera is able to detect upright pedestrians, but degrades in performance for more complex poses. The depth-aware sensors (LiDAR and stereo camera) are efficient for detecting objects that protrude significantly above the ground. The LiDAR is invariant towards changing weather and lighting conditions, whereas the stereo camera has the highest resolution making it useful for classifying objects. The thermal camera shows great capabilities in the captured dataset as it is able to detect objects of distinct temperature using a simple procedure that works both for humans and living obstacles. However, the detection would be remarkably more complicated in higher temperature environments, where living objects become indistinct in their heat signatures.

The above arguments and the case study concludes that the use of multiple modalities, more complicated procedures and a fusion of the different modalities is required to achieve a robust detection of obstacles under variable conditions. To provide a thorough evaluation of the algorithms and procedures, the dataset must be expanded to represent more scenarios including more variable lighting and weather conditions and more representations of more objects.

Acknowledgements

This research is sponsored by the Innovation Fund Denmark as part of the project “SAFE - Safer Autonomous Farming Equipment” (project no. 16-2014-0) and “Multi-sensor system for ensuring ethical and efficient crop production” (project no. 155-2013-6). Additionally, the authors would like to thank “Traktor & Høstspecialisten A/S” for providing a CLAAS tractor for testing.

References

Christiansen, P., Steen, K., Jørgensen, R., and Karstoft, H., 2014. Automated Detection and Recognition of Wildlife Using Thermal Cameras. *Sensors*, 14(8), 13778–13793.

CLAAS Steering Systems, 2011. Tracking control optimisation. Available at: http://claas.via-us.co.uk/booklets/gps-steering-systems/download. (last accessed 12/12/14).

Dirk, T., ROS Wiki: rosbag package. Available at: http://wiki.ros.org/rosbag [Accessed March 20, 2015].

Dollar, P., Belongie, S., and Perona, P., 2010. The Fastest Pedestrian Detector in the West. In *Procedings of the British Machine Vision Conference 2010*. British Machine Vision Association, pp. 68.1–68.11.

Fischler, M.A. and Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.

Luettel, T., Himmelsbach, M., and Wuensche, H.-J., 2012. Autonomous Ground Vehicles—Concepts and a Path to the Future. *Proceedings of the IEEE,* 100(Special Centennial Issue), 1831–1839.

Moosmann, F., Pink, O., and Stiller, C., 2009. Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion. *2009 IEEE Intelligent Vehicles Symposium*, 215–220.

Rasshofer, R.H. and Gresser, K., 2005. Automotive Radar and Lidar Systems for Next Generation Driver Assistance Functions. *Advances in Radio Science*, 3, 205–209.

Reina, G. and Milella, A., 2012. Towards Autonomous Agriculture: Automatic Ground Detection Using Trinocular Stereovision. *Sensors*, 12(12), 12405–12423.

Rouveure, R., Nielsen, M., and Petersen, A., 2012. The QUAD-AV Project: multi-sensory approach for obstacle detection in agricultural autonomous robotics. *Proceedings of 2012 International Agricultural Engineering CIGR-AgEng, Valencia, Spain*, 8–12.

Serrano-Cuerda, J., Fernández-Caballero, A., and López, M., 2014. Selection of a Visible-Light vs. Thermal Infrared Sensor in Dynamic Environments Based on Confidence Measures. *Applied Sciences*, 4(3), 331–350.

Steen, K.A., Villa-Henriksen, A., Therkildsen, O.R., and Green, O., 2012. Automatic detection of animals in mowing operations using thermal cameras. *Sensors (Basel, Switzerland)*, 12(6), 7587–97.

Zhang, Z., 2000. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), 1330–1334.

Paper 9

Towards Inverse Sensor Mapping in Agriculture
Timo Korthals, Mikkel Kragh, Peter Christiansen and Ulrich Rückert

Peer reviewed
Paper accepted to conference, Agricultural Robotics: learning from Industry 4.0 and moving into the future, September 2017, Vancouver, Canada

Towards Inverse Sensor Mapping in Agriculture

Timo Korthals[1], Mikkel Kragh[2], Peter Christiansen[2], and Ulrich Rückert[1]

Abstract— **In recent years, the drive of the Industry 4.0 initiative has enriched industrial and scientific approaches to build self-driving cars or smart factories. Agricultural applications benefit from both advances, as they are in reality mobile driving factories which process the environment. Therefore, acurate perception of the surrounding is a crucial task as it involves the goods to be processed, in contrast to standard indoor production lines. Environmental processing requires accurate and robust quantification in order to correctly adjust processing parameters and detect hazardous risks during the processing. While today approaches still implement functional elements based on a single particular set of sensors, it may become apparent that a unified representation of the environment compiled from all available information sources would be more versatile, sufficient, and cost effective. The key to this approach is the means of developing a common information language from the data provided. In this paper, we introduce and discuss techniques to build so called inverse sensor models that create a common information language among different, but typically agricultural, information providers. These can be current live sensor data, farm management systems, or long term information generated from previous processing, static drone images, or satellites. In the context of Industry 4.0, this enables the interoperability of different agricultural systems and allows information transparency.**

I. Introduction

Agricultural vehicles are complex, mobile processors of biological products that operate in unstructured and constantly changing environment. While the operation of these vehicles was initially relatively simple, today their setup and use requires trained specialists due to the requirement of increasing efficiency and lowering overall costs. However, without automation and the augmenting of parameter optimization in the process chain, throughputs, and farming yields would be much smaller than usual. For instance, automated steering systems employed in harvesting use LiDAR systems to scan the area between the crop and stubble in order to automatically guide the harvester along the edge; and seed drills save GPS data and the machine parameters of sowing which are used later to minimize the utilization of fertilizer spreaders.

Focusing the automation and in particular its implementation, all applications follow the same paradigm of having a distinctive set of sensors, a processing unit, and an actuator interface to steer the vehicle or manipulate process parameters. While this approach allows simple, distributed and modular modification, with increases in automated functionality its installation and maintenance becomes unfeasible due to the sheer number of sensors and processing units required. Furthermore, the potential for sensor fusion is completely squandered. An alternative approach is pursued by the authors, that of building a common inner semantical representation of the environment based on occupancy grid maps, from which all further automation is derived [1], [2]. These grid maps are arranged in multiple overlapping layers, where each one is occupied by localized classifications.

While the authors have already provided a proof-of-concept of semantical grid mapping approaches in agriculture [3], requisite information and instructions for building sensor models based on sensors and other data sources is still lacking. In contrast to robotic and automotive approaches, where grid mapping based applications are well known, agricultural environments and applications especially vary greatly and therefore have to be treated accordingly. With respect to Fig. 1 and [4], this contribution focuses on the *Inverse Sensor Modeling* component.

The paper is organized as follows: Section II presents a brief introduction to occupancy grid maps, their extension to the semantical representation. Section III presents the gathered experience and approaches to building sensor models derived from previous agricultural research projects. Finally, Section IV presents further ideas and points to next steps in agricultural applications in Industry 4.0.

II. Related Work

Occupancy grid maps are used in static obstacle detection for robotic systems, which are a well-known and a commonly studied scientific field [5], [6], [7]. They are a component of almost all navigation and collision avoidance systems designed to maneuver through cluttered environments. Another important application is the creation of obstacle maps for traversing an unknown area and the recognition of known obstacles, so supporting the localization. Recently, occupancy grid maps have been applied to combine LiDAR and RADAR in automotive applications, with the goal of creating a harmonious, consistent and complete representation of the vehicle's environment as a basis for advanced driver assistance systems [8], [9], [10].

A. Occupancy Grid Mapping

Two-dimensional occupancy grid maps (OGM) were originally introduced by Elfes [11]. In this representation, the environment is subdivided into a regular array or a grid of quadratic cells. The resolution of the environment representation directly depends on the size of the cells. In addition to

[1] Bielefeld University, Cluster of Excellence Cognitive Interaction Technologies, Cognitronics & Sensor Systems, Inspiration 1, 33619 Bielefeld, Germany, `http://www.ks.cit-ec.uni-bielefeld.de/`, `{tkorthals, rueckert}@cit-ec.uni-bielefeld.de`

[2] Aarhus University, Department of Engineering, Finlandsgade 22, DK-8200 Aarhus N, Denmark `http://eng.au.dk/`, `{mkha, pech}@eng.au.dk`

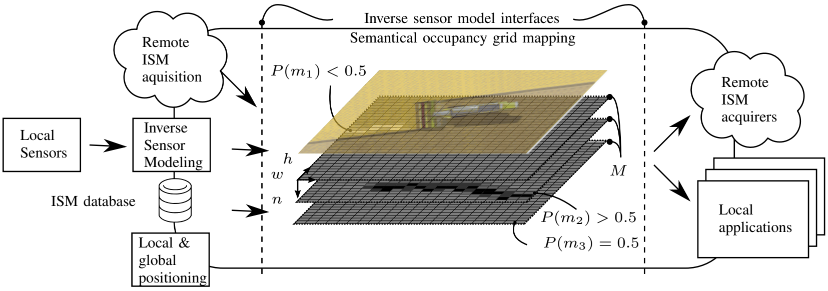


Fig. 1: Semantic occupancy grid mapping framework

this compartmentalization of space, a probabilistic measure of occupancy is associated with each cell. This measure takes any real number in the interval $[0, 1]$ and describes one of the two possible cell states: unoccupied or occupied. An occupancy probability of 0 represents a space that is definitely unoccupied, and a probability of 1 represents a space that is definitely occupied. A value of 0.5 refers to an unknown state of occupancy.

An occupancy grid is an efficient approach to representing uncertainty, combining multiple sensor measurements at the decision level, and to incorporating different sensor models [10]. To learn an occupancy grid M given sensor information z, different update rules exist [5]. For the authors' approach, a Bayesian update rule is applied to every cell $m \in M$ at position (w, h) as follows: Given the position x_t of a vehicle at time t, let $x_{1:t} = x_1, \ldots, x_t$ be the positions of the vehicle's individual steps until t, and $z_{1:t} = z_1, \ldots, z_t$ the environmental perceptions. For each cell m of the occupancy probability grid the probability that this cell is occupied by an obstacle. Thus, occupancy probability grids seek to estimate

$$P\left(m|z_{1:t}, x_{1:t}\right) = \mathrm{Odds}^{-1}\left(\prod_{t=1}^{T} \underbrace{\frac{P\left(m|z_t, x_t\right)}{1 - P\left(m|z_t, x_t\right)}}_{\mathrm{Odds}(P(m|z_t,x_t))}\right) \quad (1)$$

This equation already describes the online capable, recursive update rule that populates the current measurement z_t to the grid, where $P\left(m|z_{1:t}, x_{1:t}\right)$ is the so called inverse sensor model (ISM). The ISM is used to update the OGM in a Bayesian framework, which deduces the occupancy probability of a cell, given the sensor information.

B. Extension to Agriculture Applications

The adaptation of OGM techniques to agricultural applications appears to be merely a matter of time but is not that obvious and intuitive to apply on the second sight. Robotic and automotive applications have in common that they both want to detect non-traversable areas or objects occupying their path. Such unambiguous information is used to quantify the whole environment sufficiently for all derivable tasks, such as path planning or obstacle avoidance, to be completed. When assumptions like a flat operational plane or minimum obstacle heights are made, sensors frustums oriented parallel to the ground are sufficient for all tasks

In agricultural applications, obstacle recognition is not essential as they act on and process their environment. Therefore, quantification of the environment involves features such as processed areas, processability, crop quality, density, and maturity level in addition to traversability. In order to map these features, single occupancy grid maps are no longer sufficient and therefore, semantic occupancy grid maps that allow different classification results to be mapped are used. Furthermore, sensor frustums are no longer oriented parallel to the ground, but rather oriented at an angle to gather necessary crop information (cf. Fig. 2).

The extension to semantic occupancy grid maps (SOGM) or inference grids is straightforward and is defined by an OGM M with W cells in width, H cells in height, and N semantic layers (c.f. Fig. 1):

$$M : \{1, \ldots, W\} \times \{1, \ldots, H\} \rightarrow m = \{0, \ldots, 1\}^N \quad (2)$$

Compared to a single layer OGM which allows the classification into three classes $\left\{\text{occupied}, \overline{\text{occupied}}, \text{unknown}\right\}$, the SOGM supports a maximum of $\left|\left\{\text{occupied}, \overline{\text{occupied}}, \text{unknown}\right\}\right|^N = 3^N$ different classes allowing much higher differentiability in environment and object recognition. The corresponding ISMs are fused by means of the occupancy grid algorithm to their nth associated semantical occupancy grid.

The location of information in the maps is required to be completed by *mapping under known poses* approaches [6]. As proposed by REP-105[1] and realized by the authors in [4], information is mapped locally via Kalman filtered odometry and inertial navigation measurement. The maps themselves are globally referenced which on the one side allows smooth local mapping in the short term without the discrete jumps caused by global positioning systems using a Global Navigation Satellite System (GNSS), but also allows global consistent storing and loading of information.

While the actual features are very diverse of agriculture applications, this publication does not primarily focus on classification, but rather on geographical interpretation and sensor building.

[1] http://www.ros.org/reps/rep-0105.html

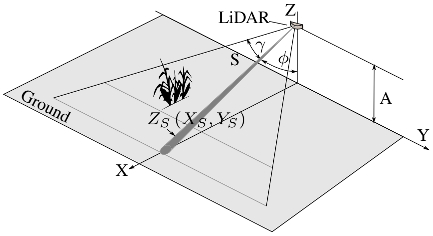


Fig. 2: Ground oriented LiDAR for crop rectification

III. Explicit ISM Generation for Specific Sensors

A. Local Sensor Based ISM

1) LiDAR based Mapping: LiDAR sensors measure the distance to an object and depending on their capabilities, also the reflectance. The distance can directly be used to deduce free (s.t. the area between the measured distance and the sensor) and occupied space (s.t. the location of measured distance) in a planar environment. This is commonly utilized for robotic and automotive tasks, where a well-known inverse sensor modelling technique directly derives the corresponding ISM. In agriculture, however, it is common for LiDAR sensors to face downwards as shown in Fig. 2, in order to detect the soil or crop that needs to be processed. This results in the circumstance that the measurement can only be taken at the corresponding target point, and no implications can be done along the measurement.

Naively mapping the related classification in the point of measurement in the vehicles coordinate frame would result in scattered maps from which further applications are hardly derivable (c.f. Fig. 3). Therefore, the actual Gaussian measurement uncertainty σ_S needs to be introduced as in the common planar model, but with its appropriate error propagation. Assuming σ_ϕ, σ_ξ, σ_γ beeing gaussian noise in the angular positioning caused by vehicle's steering, and σ_x, σ_y, σ_z to be the positioning caused by vibrations of the vehicle it is possible to calculate the resulting full covariance matrix $\sum_{X_S}$ at the point of interest as follows: First, the transformation of the scalar distance measurement S in the LiDAR frame to the euclidean point X_S in the vehicle frame is

$$X_S = \begin{pmatrix} c_\phi c_\gamma \\ c_\xi s_\gamma + c_\gamma s_\phi s_\xi \\ s_\xi s_\gamma - c_\gamma s_\phi c_\xi \end{pmatrix} S + \mathrm{T}(x, y, z) \tag{3}$$

where T is the translation between the sensor and the vehicle frame. For error propagation, the functions need to be linearized by calculating the Jacobian:

$$J^{\mathrm{T}} = \begin{pmatrix} c_\phi c_\gamma & c_\xi s_\gamma + c_\gamma s_\phi s_\xi & s_\xi s_\gamma - c_\gamma s_\phi c_\xi \\ -S s_\phi c_\gamma & S c_\gamma c_\phi s_\xi & -S c_\gamma c_\phi c_\xi \\ -S c_\phi s_\gamma & S c_\xi c_\gamma - S s_\gamma s_\phi s_\xi & S s_\xi c_\gamma + S s_\gamma s_\phi c_\xi \\ 0 & -S s_\xi s_\gamma + S c_\gamma s_\phi c_\xi & S c_\xi s_\gamma + S c_\gamma s_\phi s_\xi \end{pmatrix}^{\mathrm{T}} \tag{4}$$

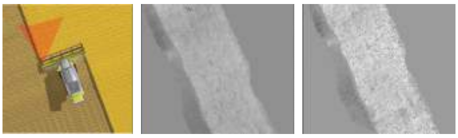


Fig. 3: Harvesting scenario (left), resulting SOGM from crop classification ISM with (middle) and without (right) error propagation

$$\sum_{X_S} = J\,\mathrm{diag}(\sigma_s^2, \sigma_\phi^2, \sigma_\gamma^2, \sigma_\xi^2) J^{\mathrm{T}} + \mathrm{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2) \tag{5}$$

The Jacobian is a function of its arguments $J(S, \phi, \gamma, \xi)$, which means that it is required to be evaluated for every new sensor measurement. Equation 5 describes the full covariance matrix which can be applied to calculate the uncertainty distribution for every measurement.

Two assumptions have been made in this model to make the error model tractable: first, that the uncertainty in angular movements resides in the coordinate frame of the laser scanner and second, that the uncertainty in translation is uncorrelated from the angular ones. The assumptions do not fully hold, due to the fact that rolling, pitching and yawing do not occure in the laser scanner frame, but in some other arbitrary frame, depending on the current ground conditions and vehicle's steering. To simplify the model even more, the uncertainty in z can be omitted, because in the later sensor modeling component, only the projection into the xy-plane is important. Further, rolling is omitted as it is negligible in comparison to the other influences [12]:

$$\begin{aligned} X'_S &= \begin{pmatrix} c_\phi c_\gamma \\ s_\gamma \\ -c_\gamma s_\phi \end{pmatrix} S + \mathrm{T}(x, y, z) \\ J' &= \begin{pmatrix} c_\phi c_\gamma & -S s_\phi c_\gamma & -S c_\phi s_\gamma \\ s_\gamma & 0 & S c_\gamma \end{pmatrix} \\ \sum_{X'_S} &= J'\,\mathrm{diag}(\sigma_s^2, \sigma_\phi^2, \sigma_\gamma^2) J'^{\mathrm{T}} + \mathrm{diag}(\sigma_x^2, \sigma_y^2) \end{aligned} \tag{6}$$

The influences of error propagation are depicted in Fig. 3 where a two class classifier for crop derives the ISMs which are mapped to the global coordinate system. The resulting map without error propagation is very sparse which makes further functionality derivation without heuristical post processing unfeasible. Introducing error propagation and respecting the model uncertainties, on the other hand, results in a much more sufficient and consistent map where further classification can easily be applied.

Further improvements in classification can be achieved by first mapping the raw LiDAR data to a globally referenced representation from which further ISMs with much higher quality can be derived. More advanced LiDAR systems scanning in multiple planes bypass the raw mapping and directly enable rich classifiers like Support Vector Machines to process the data as proposed by [3].

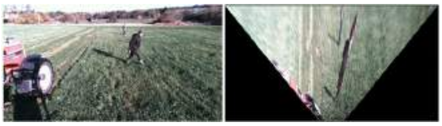


Fig. 4: Inverse Perspective Mapping of RGB image

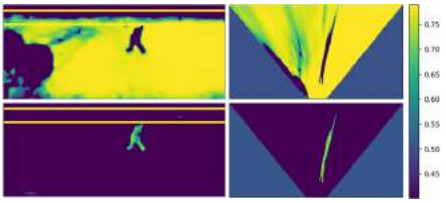


Fig. 5: (Left) Grass and human predictions in a mowing application classified by a fully convolutional network for semantic segmentation [15] and the corresponding ISMs generated by IPM (right)

2) Inverse Perspective Mapping: Inverse Perspective Mapping (IPM) is a geometrical transformation that projects an image to a ground plane surface as shown in Fig. 4. For a flat surface, the perspective effect is removed by transforming the viewpoint from a camera view to a birds eye view. This technique has been used in automotive applications where assumptions about camera pose and a flat world with respect to the street are sufficient [13], [12]. However, even slight deviations in camera inclination and height result in large errors, more advanced, adaptive techniques have been developed which calculate the camera pose online by using the borders of the road or lane markers [14].

However, an unstructured agricultural environments does permit such dynamic techniques and thus, they are either treated as a static scenario, where the camera pose relative to ground surface does not change, or the transformation between the extrinsic and flat plane is calculated dynamically with support of an inertial measurement unit (IMU). The whole IPM for mapping image coordinates $\mathbf{x}_\mathrm{P}|_\mathrm{px} = (u, v, 1)^\mathrm{T}$ to surface $\mathbf{x}_\mathrm{FP}|_\mathrm{m} = (x, y, z \equiv 0, 1)^\mathrm{T}$ is defined by three parameter transformations: the intrinsic ${}^\mathrm{P}\mathbf{T}_\mathrm{C}$ from the camera perspective to the camera frame, the extrinsic ${}^\mathrm{C}\mathbf{T}_\mathrm{V}$ from the camera frame to the vehicle frame, and ${}^\mathrm{V}\mathbf{T}_\mathrm{FP}$ which transforms from the vehicle frame to the flat plane (FP) frame. This leads to

$$\mathbf{x}_\mathrm{P}|_\mathrm{px} = {}^\mathrm{P}\mathbf{T}_\mathrm{C} \cdot {}^\mathrm{C}\mathbf{T}_\mathrm{V} \cdot {}^\mathrm{V}\mathbf{T}_\mathrm{FP} \cdot \mathbf{x}_\mathrm{FP}|_\mathrm{m} \tag{7}$$

To build the actual ISM, the image first needs to be classified and then transformed to the flat plane by means of Equation 7 (c.f. Fig. 5).

Values of an ISM are the probability of a grid cell being occupied by a giving classification. As indicated in Fig. 5, the area that is not visible by the camera is set to 0.5 to represent the fact no information is provided for areas that are not visible to the camera. Visible areas with no detections are set below 0.5 to indicate that the area is not expected to be occupied by the given class. Values above 0.5 indicate that the area is expected to be occupied by the given class.

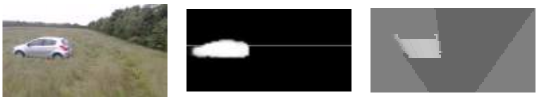


Fig. 6: Input image (left), classification based on semantic segmentation (middle) and corresponding ISM with detection cut-off after class occurrence along the focal axis

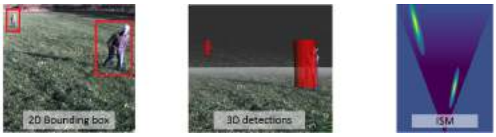


Fig. 7: Bounding box detection to ISM

For detecting flat class elements such as road-lane markings or grass, the IPM algorithm is able to provide good approximations of the actual inverse perspective mapping. Elevated elements violate the IPM ground plane assumption and will stretch elements unnaturally and incorrectly across large areas as indicated in Fig. 4.

To avoid the stretching artifacts of tall objects, different approaches are proposed. A naive approach for pixel based classifiers states that all objects classified as being other than ground are standing perpendicular on the ground. Therefore, one can perform a ray trace along the focal axis and mark all cells behind a detected object as unknown (c.f. Fig. 6) [16], [3].

Another approach generates three dimensional object location hypotheses by first estimating the distance to the corresponding detection. This can be achieved by either using the abovementioned naive approach or using a depth sensor like a stereo camera or LiDAR which is registered to the camera.

Second, when using classifiers like YOLO [17] which offers classified bounding boxes, the four bounding box corners are mapped to real world coordinates using the estimated distance to a detection and the intrinsic camera parameters. The bounding box position and extent are derived in 3D and is represented as depicted in Fig. 7 by cylinder specified by a center, height, and width.

Detections are mapped to values above 0.5 with a Gaussian distribution to indicate the existence of an obstacle with corresponding localization uncertainties. The localization uncertainty for the camera depends on the radial coordinate (distance to the object) and angular coordinate (angle to object), where accuracy degrades with increasing distance and angle. The procedure for converting a 2D bounding box to an ISM using distance estimates is presented in Fig. 7. Using the estimated distance of a detected object and the intrinsic camera parameters, the four bounding box corners are mapped to world coordinates.

Lastly, the concept of contradicting IPM is introduced for crop processing in harvesting scenarios. In comparison

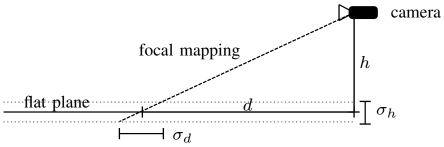


Fig. 8: Simplified error assumption in flat plane assumption according to height

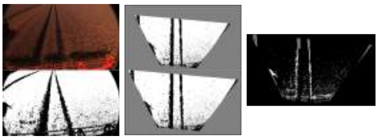


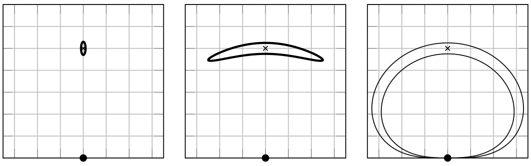
Fig. 9: RGB input image and scanline based classification for crop plane (left), inverse perspective mapping of classification for crop and ground plane (middle) and corresponding fused contradicting ISM (right)

with the abovementioned IPM scenarios, this discrimination is necessary as the camera rectifies no common ground in the lower areas of the image as depicted in Fig. 9 which refutes former assumptions. Neglecting this fact would result in drastically wrong localization of detections, as visualized in Fig. 8, which indicates that the localization error σ_d in depth d depends on the error σ_h of height h as follows:

$$\sigma_d = \frac{d}{h}\sigma_h. \tag{8}$$

If this simple error propagation is applied to a hypothetical example of small crop with for example a height of 0.5 meters and a camera installation height of 1.5 meters where a feature 10 meters away should be mapped, the resultant error is one of 3 meters. Therefore, two flat plane assumptions are calculated, one for the ground and one for the crop height resulting in two different ISMs. These can then be combined by Dempsters rule of combination leading to contradictions [18], which is visualized in Fig. 9. From the emerging contradictions in Fig. 9 (right), it can be seen that vehicle traces appear which are actually the contradicting occlusion in both IPMs.

3) Ambiguous Sensor Mapping: Ambiguous sensor readings originate from sensors with very bad angular or distance resolution by definition of the authors. As depicted in Fig. 10 LiDAR systems can achieve very accurate positioning and are therefore the preferred sensors for mapping. However, they are by far the most cost- and power intensive systems. Other sensing techniques are more cost and power efficient but are commonly neglected due to their high noise or inaccuracy. Nevertheless, the authors have demonstrated that even with poorly embedded sensors, sufficient environment detection can be achieved [19] by designing an inverse particle filter which samples from the sensors uncertainty distribution. At present, this technique has only been applied in laboratory conditions and therefore, real agricultural applications remain pending.



(a) LiDAR (b) SONAR (c) Proximity

Fig. 10: Standard error contour of qualitative sensor cones (•: Sensor position, x: Obstacle, -)

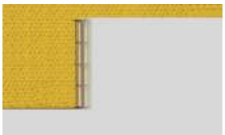


Fig. 11: Top view of crop field with an applied inverse sensor model for the cutter bar: gray shaded area being of high probability that the cutter bar has been applied on that region

B. Application Models

Application models are straight forward to implement and only depends on the localizing accuracy. Building such a model is only dependent on the geometrical shape of the agricultural implement. That means on the other hand, that ISM is a static and primitive shape in the local frame of the vehicle which leaves a probabilistic footprint where the implement has been applied to the crop as depicted in Fig. 11. When incorporating inaccurate localization, the shape needs to be transformed accordingly.

C. Map Services

Geodata acquired by satellites, drones, or planes with high recording frequencies as well as its partially free availability, make this information increasingly attractive for agriculture. In this context worth mentioning are the Sentinel program[2], the hyperspectral system EnMap[3], the RapidEye constellation[4] as well as the start-up companies Skybox Imaging[5] and Planet Labs[6]. In addition, the release of the long-standing Landsat archive now offers many opportunities for agricultural applications, such as the generation of profit potential maps. There is a trend towards direct access to such data and towards appropriate image excerpts using web servers or APIs. As part of spatial data infrastructures, data (e.g. land and terrain data) are published interoperably and often free of charge via web services. In particular, Annex III of the INSPIRE Directive[7] requires EU member states to provide data. However, for a precision farming service or a precision farming application further different data sources have to be

[2] http://www.esa.int/Our_Activities/Observing_the_Earth/Copernicus/Overview4/

[3] http://www.enmap.org/

[4] http://blackbridge.com/rapideye/

[5] http://www.skyboximaging.com/

[6] https://www.planet.com/

[7] http://inspire.ec.europa.eu/

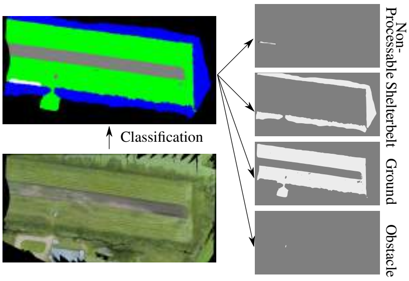


Fig. 12: Classification decomposition of hand labeled orthographic photograph [3]

linked (for example, weather data play a crucial role in most agricultural processes), or complex procedures and algorithms are required to derive the desired information from the data. Subsequent downstream services will continue to play an increasingly important role in agriculture. The European Union, for example, specifically supports the development of such services based on Copernicus data by SMEs. At the endpoint of the downstream services, information products (such as humidity maps, biomass maps and yield forecast maps) are often available, which can be integrated into other applications or devices. The combination and the inclusion of all the information sources and their derivation for the identification of machine parameters is one essential part which can be handled by ISMs. As an example, a static and classified drone image can be easily transferred to a semantic ISM by decomposing all classes and loading the appropriate area during operation (c.f. Fig. 12).

IV. Conclusion and Outlook

The authors have presented an information representation as semantic grids which can be maintained among different modalities and sources. It utilizes the idea of the ISOBUS standard, which was designed with machinery interoperability in mind, and allows every sensory source to publish or access its information in a general grid format. The main aspect of this contribution focused on different techniques, originating from literature, practical experiments, and experience, of actually building these representations.

As the acquisition and localization of data are sufficiently solved, further research will concentrate on planning and control of such diverse data. Furthermore, learning approaches have not been confronted in this application which directly maps a sensor reading to the appropriate locality and probability. These techniques were introduced by Thrun [6] and have been applied by the authors. However, following the engineering path of building inverse sensor models is far more robust and intuitive. At present, only a few approaches are known to the authors and therefore, more applications extending from direct control architectures up to holistic farm management systems are of great interest. Approaching rich control architectures in agricultural environments allows an interesting area of overlap between robotics and Industry 4.0 to emerge, s.t. simultaneously planning and processing. Mathematical frameworks exist, where in agriculture the particular issue will driven by the information representation and how it is incorporated into environmental processing.

Acknowledgment

This research was supported by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG) and by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster "Intelligent Technical Systems OstWestfalenLippe" (it's OWL) and managed by the Project Management Agency Karlsruhe (PTKA).

References

[1] T. Korthals, A. Skiba, and T. Krause, "Evidenzkarten-basierte Sensorfusion zur Umfelderkennung und Interpretation in der Ernte," in *Informatik in der Land-, Forst und Ernährungswirtschaft*, 2016, pp. 15–18.

[2] ——, "Einsatz Event-Basierter Systemarchitektur für Erntemaschinen zur Elektronischen Umfelderkennung," in *74. Tagung LAND.TECHNIK*. VDI e.V., 2016.

[3] M. Kragh, P. Christiansen, T. Korthals, T. Jungeblut, H. Karstoft, and R. N. Jørgensen, "Multi-Modal Obstacle Detection and Evaluation of Occupancy Grid Mapping in Agriculture," in *International Conference on Agricultural Engineering*, Aarhus, 2016.

[4] T. Korthals, J. Exner, T. Schöpping, and M. Hesse, "Semantical Occupancy Grid Mapping Framework," in *European Conference on Mobile Robotics*. IEEE, 2017.

[5] D. Hähnel, "Mapping with Mobile Robots," Ph.D. dissertation, University of Freiburg, 2004.

[6] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, Mass.: MIT Press, 2005.

[7] C. Stachniss, *Robotic Mapping and Exploration*, 2009.

[8] R. Garcia, O. Aycard, and T.-d. Vu, "High Level Sensor Data Fusion for Automotive Applications using Occupancy Grids," no. December, pp. 17–20, 2008.

[9] M. E. Bouzouraa and U. Hofmann, "Fusion of occupancy grid mapping and model based object tracking for driver assistance systems using laser and radar sensors," in *2010 IEEE Intelligent Vehicles Symposium*, 2010, pp. 294–300.

[10] H. Winner, *Handbuch Fahrerassistenzsysteme - Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*. Wiesbaden: Vieweg+Teubner Verlag, 2015.

[11] A. Elfes, "Occupancy Grids: A Stochastical Spatial Representation for Active Robot Perception," in *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, 1990.

[12] M. Konrad, D. Nuss, and K. Dietmayer, "Localization in digital maps for road course estimation using grid maps," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 87–92, 2012.

[13] M. Bertozzi and a. Broggi, "Real-time lane and obstacle detection on the GOLD system," *Proceedings of Conference on Intelligent Vehicles*, 1996.

[14] N. Simond and P. Rives, "Homography from a vanishing point in urban scenes," *International Conference on Intelligent Robots and Systems*, pp. 1005–1010, 2003.

[15] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[16] S. Kohlbrecher, "Grid-based occupancy mapping and automatic gaze control for soccer playing humanoid robots," *... Humanoid Soccer Robots ...*, no. October, 2011.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Cvpr 2016*, pp. 779–788, 2016.

[18] G. Shafer, *A Mathematical Theory of Evidence*. Princeton: Princeton University Press, 1976.

[19] T. Korthals, M. Barther, and S. Herbrechtsmeier, "Occupancy Grid Mapping with Highly Uncertain Range Sensors based on Inverse Particle Filters," 2016.

Paper 10

Multi-Modal Obstacle Detection and Evaluation of Occupancy Grid Mapping in Agriculture
Mikkel Kragh, Peter Christiansen, Timo Korthals, Thorsten Jungeblut, Henrik Karstoft, Rasmus Nyholm Jørgensen

Peer reviewed
Presented at International Conference on Agricultural Engineering, June 2016, Aarhus, Denmark

Multi-modal Obstacle Detection and Evaluation of Occupancy Grid Mapping in Agriculture

Mikkel Kragh[a,*], Peter Christiansen[a,*], Timo Korthals[b,*], Thorsten Jungeblut[b], Henrik Karstoft[a], Rasmus N. Jørgensen[a]

[a] Department of Engineering, Aarhus University, Finlandsgade 22, DK-8200 Aarhus N, Denmark

[b] Cognitronics & Sensor Systems, Bielefeld University, Inspiration 1, D-33619 Bielefeld, Germany

* Corresponding author. Email: {mkha, pech}@eng.au.dk, tkorthals@cit-ec.uni-bielefeld.de

Abstract

In recent years, mapping and automation has been increasingly investigated and applied in precision agriculture. The ultimate goal of this development is to apply autonomous vehicles operating efficiently without any human intervention. Such autonomous operation imposes severe safety hazards, demanding accurate and robust risk detection, and avoidance systems. It is unlikely that one sensor can single-handedly guarantee this, and therefore multiple sensing modalities are often combined in order to increase detection performance and introduce redundancy. In this paper, we present a global mapping approach utilizing diverse sensor technologies to achieve a uniform obstacle interpretation of the environment. Using occupancy grid maps, we fuse information from a monocular color camera, a RADAR, and a LIDAR in combination with IMU-assisted GPS-positioning. For each sensor, we present detection algorithms, mapping from raw sensor data to a 2D grid-based obstacle interpretation of the environment. These are then fused temporally with the occupancy grid algorithm, and afterwards spatially in a competitive and complementary way to produce a combined global obstacle map. The method is evaluated on an extensive dataset recorded at Research Centre Foulum, Denmark, in June 2015. The dataset comprises sensor data from a tractor-mounted recording system in a grass mowing scenario with various obstacles. A ground truth map has been obtained with a mapping drone. Results show promising obstacle detection capabilities and an increase in performance when fusing information across sensor modalities and layers. The proposed mapping framework is able to fuse a vast amount of information across a diverse sensor set, using an efficient and novel approach for obstacle detection in agriculture.

Keywords: Multi-modal Sensor Fusion, Obstacle Detection, Occupancy Grid Mapping, Precision Farming, Agriculture

1. Introduction

The application of robots or vehicles operating autonomously in agricultural fields demands extreme perception capabilities of the safety system. It is unlikely that a single perception sensor is capable of ensuring this safety alone, and thus multiple sensor technologies must be combined to provide accurate and robust risk detection and avoidance. These sensors might operate in different coordinate systems with different representations. For instance, a LIDAR operates in 3D cartesian coordinates, an automotive RADAR operates in 2D polar coordinates, and cameras operate in projective spaces of 2D pixel coordinates. Sensor fusion can be handled on various abstraction levels such as data-, feature- or decision-level, but all methods require a mapping to a common representation. One such fusion algorithm on feature-level is occupancy grid maps (Elfes 1990). In 2D, they represent a global map of the environment and are generated from inverse sensor models (ISMs). An ISM is associated with a specific sensor and includes a detection algorithm of a certain feature (e.g. "vehicle", "human", "field", "ground") and a mapping from sensor data to a local 2D grid in the vehicle frame.

In research on automotive vehicles, 2D grid mapping is widely applied for fusing information across sensing modalities, providing a simple yet efficient framework (Winner 2015). In agricultural environments, a few applications with grid mapping have been proposed as well (Reina and Milella 2012; Ahtiainen et al. 2015). However, these only use a single or two sensing modalities, and thus do not provide a full evaluation of the potential of occupancy grid mapping.

In this paper, we present a global mapping approach utilizing simultaneous information from a monocular color camera, a thermal camera, a RADAR, and a LIDAR in combination with IMU-assisted GPS-positioning. For each of the sensors, we present detection algorithms, mapping from raw sensor data to a 2D grid-based obstacle interpretation of the environment. These grids represent multiple obstacle layers ("human", "object", "vegetation", etc.) and are updated temporally using the occupancy grid algorithm. Finally, they are fused spatially across layers and sensor modalities using competitive and complementary fusion.

2. Materials and Methods

2.1. Setup

A variety of sensor modalities and corresponding detection algorithms are used to ensure detection and provide redundancy for all relevant obstacle types. A Velodyne HDL-32E LIDAR (laser range scanner) is used for long range depth estimation and is robust towards changes in illumination and weather. A Delphi ESR automotive RADAR is used for mid and long range depth and velocity estimation, and is even more robust towards changes in illumination and weather than the LIDAR. A Logitech C920 color camera is used to detect and distinguish between different obstacle types, but is significantly more sensitive towards changes in illumination. Finally, a thermal camera is useful for capturing heat radiation from humans and animals. However, since only static, non-living obstacles are present in the dataset, this sensor is excluded from the paper. Together, the sensors both complement and overlap each other in terms of detection capabilities and robustness. A Vectornav VN-100 Inertial Measurement Unit (IMU) and a Trimble AG GPS361 Real Time Kinematic (RTK) GPS unit are used for pose estimation. Offline calibration is performed by hand by estimating extrinsic parameters of sensor positions. The specific sensor platform used for the experiments is presented and explained in detail in a previous paper (Christiansen et al. 2015).

2.2. Detection Algorithms

In the following sections, the algorithms used to produce classifications and their conversions to ISMs are described.

2.2.1. LIDAR

A single LIDAR scan provides a 3D point cloud consisting of depth measurements distributed 360° horizontally around the vehicle. For each point, we calculate 13 features using statistics from a local neighborhood (Kragh, Jørgensen, and Pedersen 2015). These features describe the height, shape, orientation and reflectance of the structure and help distinguish between points representing three classes: “ground”, “vegetation”, and “object”. A Support Vector Machine (SVM) classifier with probability estimates (Wu, Lin, and Weng 2004) is then trained to classify individual points into these classes. Figure 1 (left) shows an example of pseudo-colored probability estimates of the “object” class.

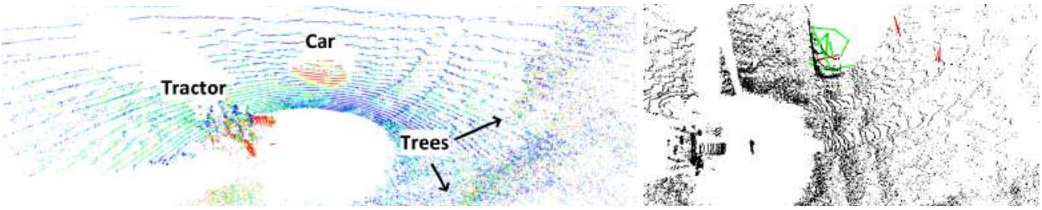


Figure 1. Left: Point cloud with pseudo-colored probability estimates of “object” class illustrating low (blue) and high (red) probabilities. Right: RADAR tracks overlaid on point cloud. Green are confirmed tracks and red are unconfirmed.

2.2.2 RADAR

The automotive RADAR combines mid- and long-range functionality simultaneously, so that it can detect close-distance objects with a horizontal field of view (FOV) of ±45° and far-distance objects with a narrow FOV of ±10°. The RADAR itself provides a processed list of up to 32 tracked objects, each with an angle and a range. However, most of these represent internal noise in the RADAR and therefore need to be processed further. For that, we apply the Kuhn-Munkres assignment algorithm (KMA), tracking detections from subsequent frames (Munkres 1957). Only detections that are less than 2 m apart from one frame to the next are associated. A track i is described by its current position and its track length L_i and is confirmed when $L_i \geq L_{min} = 3$. All confirmed tracks are then converted to detection probabilities:

$$P_{radar,i} = \frac{L_i - L_{min}}{L_i}$$

2.2.3 Color Camera

For the color camera, we apply three detection algorithms; Locally Decorrelated Channel Features for Pedestrian detection (PED) (Nam, Dollár, and Han 2014), You Only Look Once (YOLO) (Redmon et al. 2016), and Fully Convolutional Network for Semantic Segmentation (SS) (Long et al. 2015).

PED is a state-of-the-art pedestrian detector trained on the INRIA dataset (Dalal and Triggs 2005). PED uses three color and seven edge feature channels followed by a local decorrelation step creating 40 decorrelated feature channels. The algorithm uses an AdaBoost (Freund and Schapire 1996) based classifier and detects humans at multiple locations and scales using a speed efficient multiscale sliding window approach.

YOLO is a deep convolutional neural network (CNN) for object detection trained on 20 object classes on the Pascal Visual Object Classes (VOC) dataset (Everingham, Eslami, and Gool 2013). In this work, the 20 objects are mapped to three object classes: “human”, “vehicle”, and “unknown”.

In agriculture, elements such as the field and shelterbelts cannot naturally be delimited by a bounding box as normally provided by object detection algorithms. SS is a semantic segmentation method, meaning that each pixel in the image is classified as an object class. The algorithm is trained to recognize 60 object classes in the PASCAL-Context dataset (Mottaghi et al. 2014). As described in (Christiansen et al. 2016), these element classes can be remapped to a few agricultural classes. In this work, the classes are remapped to “unknown”, “grass”, “ground”, “human”, “shelterbelt”, “vehicle”, and “water”. An example of the outputs from the algorithms described above is presented in two cropped images in Figure 2.

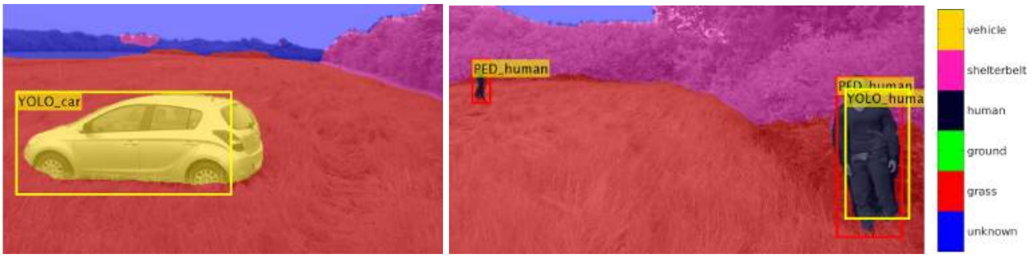


Figure 2. Example output of camera algorithms. PED detects both humans. YOLO is able to detect the vehicle and a human, but fails to detect the more distant human. SS detects both humans, the car, sky, ground and most of the shelterbelt. However, SS fails to detect the shelterbelt at far distance and around the human.

PED and YOLO algorithms output bounding box coordinates that are converted to a new image for each object class with a rectangle filled with a confidence measure of a detection. SS outputs an image for each object class, where each pixel contains a confidence measure of classification.

2.3. Mapping

Within this publication, two challenges are faced by mapping the algorithms' detections into a map representation of the vehicle's environment: First, by locating and mapping the detections into a map, evaluation against a ground truth map is easily applicable. Second, the map representation serves as the common way of fusing detections of a single algorithm temporally, and spatially across different modalities. A technique which suits these requirements is the Occupancy Grid Mapping (OGM).

2.3.1. Occupancy Grid Mapping

Two-dimensional occupancy grids were originally introduced by Elfes (Elfes 1990). In this representation, the environment is subdivided into a regular array or a grid of rectangular cells. The resolution of the environment representation directly depends on the size of the cells. In addition to this discretization of space, a probabilistic measure of occupancy is associated with each cell. This measure takes on any real number in the interval [0, 1] and describes one of the two possible cell states: occupied or unoccupied. An occupancy probability of 0 means definitely unoccupied space, and a probability of 1 means definitely occupied space. A value of 0.5 refers to an unknown state of occupancy.

The occupancy grid is an efficient approach for representing uncertainty, fusing multiple sensor measurements, and to incorporate different sensor models (Winner 2015). To learn an occupancy grid M given sensor information z, different update rules exist (Hähnel 2004). For our approach, we use the Bayesian update rule which is applied to every cell $m \in M$ as follows: Given the positions x_t of the vehicle at each point in time t, suppose $x_{1:t} = x_1, ..., x_t$ are the positions of the vehicle at the individual steps in time, and $z_{1:t} = z_1, ..., z_t$ are the perceptions of the environment. Occupancy probability grids determine for each cell c of the grid the probability that this cell is occupied by an obstacle. Thus, occupancy probability grids seek to estimate

$$P\,(m|z_{1:T}, x_{1:T}) = \prod_{t=1}^{T} \frac{P\,(m|z_t,x_t)}{1-P\,(m|z_t,x_t)} = \prod_{t=1}^{T} Odds(m|z_t, x_t)\ .$$

This equation already describes the online capable, recursive update rule that populates the current measurement z_t to the grid, where $P\,(m|z_t, x_t)$ is the so called inverse sensor model (ISM). The ISM is used to update the OGM in a Bayesian framework, which deduces the occupancy probability of a cell, given the sensor information.

2.3.2. Inverse Sensor Modelling

The ISM implements the inverse measurement model, which deduces from the sensor measurement to the occupancy probability at the particular cell. It is commonly used for sensors with a planar sensor lobe oriented parallel to the ground. In that case, a quite simplistic model can be applied, e.g. for a laser range finder. Each cell m that is covered by the beam of the observation z and whose distance to the sensor is shorter than the measured one, is supposed to be unoccupied. The cell in which the beam ends (the measurement point) is supposed to be occupied, and everything behind is unknown (Stachniss 2009). For our implementation, however, the cameras, LIDAR, and RADAR are non-planar, as their sensor lobes are tilted. Every non-planar sensor, compared to planar operating sensors, can only be evaluated at the measurement point, and thus do not provide any information in front of the measurement. Each sensor-algorithm combination requires its own ISM, converting from the algorithm's output to a 2D measurement grid representation. For this, a geometric interpretation is needed in order to transform features from the sensor frame to the vehicle frame.

2.3.2.1 ISM for LIDAR

From the SVM classifier, a 3D point cloud with class probabilities is provided for each class: "ground", "vegetation", and “object”. A 2D class probability grid is created for each class by projecting all points onto a locally estimated plane and averaging over class probabilities of points lying within a grid cell. From these class probability grids P^*_{class}, two ISM obstacle layers are produced: “object” and “vegetation”. Figure 4 (left) illustrates an example of the “object” layer. The calculation of the log odds ratio of “object” combines the probability of the cell m being an object and the cell not being ground:

$$\begin{aligned} logOdds(P_{object}(m)) &= logOdds(P^*_{object}(m)) + logOdds(1 - P^*_{ground}(m)) \\ &= log(P^*_{object}(m)) - log(1 - P^*_{object}(m)) + log(1 - P^*_{ground}(m)) - log(P^*_{ground}(m)) \end{aligned}$$

2.3.2.2 ISM for RADAR

An ISM obstacle layer “radar” is produced by converting all confirmed detections from polar to cartesian coordinates and averaging over detection probabilities of tracks lying within a grid cell. This provides a probability grid $P^*_{ground}(m)$. The calculation of the log odds ratio of “radar” for cell m is then given by:

$$logOdds\,(P_{radar}(m)) = logOdds\,(P^*_{radar}(m)) = log\,(P^*_{radar}(m)) - log\,(1 - P^*_{radar}(m))$$

2.3.2.3 ISM for Camera - Inverse Perspective Mapping

Within this chapter, the projection of a camera image onto a planar ground map is described. We assume a pinhole model for the camera, a constant transformation between the camera frame and the vehicle’s footprint, and a flat world. To calculate the pixel-wise transformation from the camera frame into the vehicle frame, the inverse perspective mapping introduced by (Bertozzi and Broggi 1996) is applied.

Because of the flat world assumption, the projection is ill-defined for any detection that does not reside on the ground level. Kohlbrecher bypasses this problem by assuming every detected object to be grounded (Kohlbrecher 2011). In this way, an occupancy grid is generated by traversing through every column of a detection image starting from the bottom. This creates a ray in the occupancy grid, starting at the sensor position towards the horizon. When a detection ($P > 0.5$) occurs along this ray, the given cell is mapped accordingly and all subsequent cells are mapped as unknown ($P = 0.5$).

In this work, a positive detection pixel is extended by the estimated depth of a given obstacle before mapping unknown pixels. Figure 3 illustrates an example of this procedure. In the center image, a positive detection (white blob) of a vehicle seen by the SS algorithm is shown along with the estimated horizon. At the right, the same image converted through inverse perspective mapping to an occupancy grid is visualized, showing how the vehicle is assumed to have a depth of 2 meters.

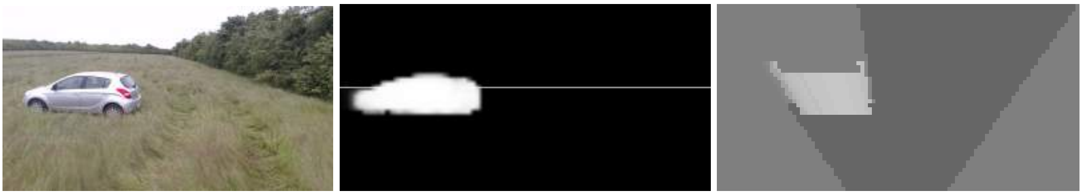


Figure 3. Left: Input image. Center: Horizon and detection of vehicle with semantic segmentation. Right: Inverse perspective mapping showing vehicle, FOV and unknown areas both behind the vehicle and outside the FOV.

2.3.3 Grid Map Representation

Different approaches exist for handling the residency of a map. For spatially limited applications, commonly one global map is used. To reduce the memory consumption, so called topo-metric maps are used as well, where the map size is reduced to e.g. rooms which are interconnected by a graph (Hähnel 2004). For automotive applications, temporary maps have proven their worth. They are build up by different sensors for a short time scenery of the environment (Winner 2015). This paper formulates an independent and global coordinate system which holds multiple two-dimensional grid maps for small areas. The whole area is divided into patches, and for each timestep only one patch, namely the Region-Of-Interest (ROI) is loaded. As depicted in Figure 4 (center and right), the patches overlap at the point where the vehicle crosses the border from the inner to outer ROI to the outer margin. If the vehicle passes this border, a new patch map is loaded. This provides two advantages: First, the memory consumption is reduced to a minimum and second, drift over multiple maps can be reduced by realigning all maps subsequently. Our solution can be compared to the patch map approach by (Konrad et al. 2011). Konrad aligns all maps vertically and horizontally with an overlap at their margins. Compared to this, our approach is able to respect former recorded data by transforming it into the upcoming ROI.

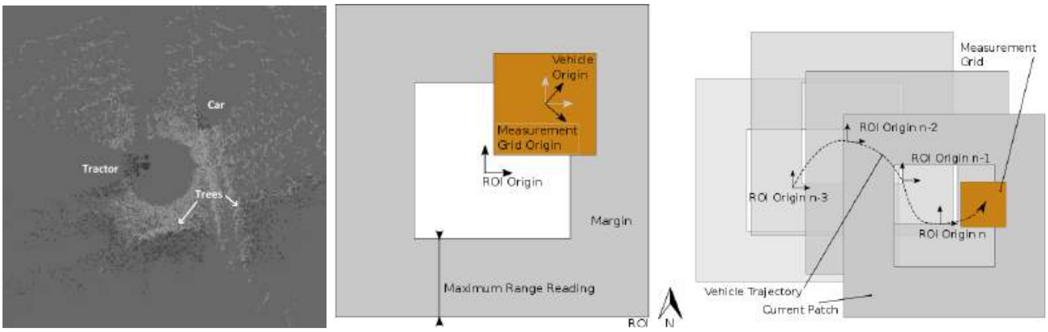


Figure 4. Left: Inverse sensor model as measurement grid of LIDAR for class "object". Center: Current patch as Region-Of-Interest. Right: Overlaid patches along a vehicle's trajectory

2.3.4 Mapping Uncertainty

Every ISM is influenced by the vehicle's pose uncertainty. This includes the latitude and longitude and the roll/pitch/yaw angles. Furthermore, because of the flat-plane assumption, the error caused by the assumed sensor height above the ground is respected as well. All uncertainties of every grid cell are modeled by a two-dimensional Gaussian function. To respect all Gaussian uncertainties in the ISM, all cell neighbours have to be taken into account. Thus, first an ISM without position uncertainties is created and then convolved by a Gaussian kernel $F \in \Re^{IxJ}$. To respect the fact that we deal with probabilities inside the ISM, we define the convolution function P^* for a single probability P of a cell $m_{x,y}$ at point (x, y) in the grid M as follows:

$$P^*(m_{x,y}) = logOdds^{-1} \sum_{i=x-I/2}^{x+I/2} \sum_{j=y-J/2}^{y+J/2} logOdds(F(i,j)(P(m_{i,j}) - 0.5) + 0.5)$$

3. **Results and Discussion**

3.1. Dataset

The evaluation of the grid mapping is performed on a dataset recorded at Research Centre Foulum, Denmark, in June 2015. The sensor platform described in section 2.1 is mounted in front of a tractor in a grass mowing scenario, recording over a 15 minute traversal in the field. Apart from naturally occurring elements in the field (shelterbelts, grass, ground, and water flooding), static obstacles (wells, a car, barrels, and adult and kid mannequin dolls) are placed and measured with precise GPS positions. The dataset also includes a single moving object (walking pedestrian). A ground truth map is generated by recording the field and obstacles with a Phantom 2 drone and manually annotating with per-pixel labeling. Figure 5 shows the orthophoto of the field with overlaid ground truth annotations.

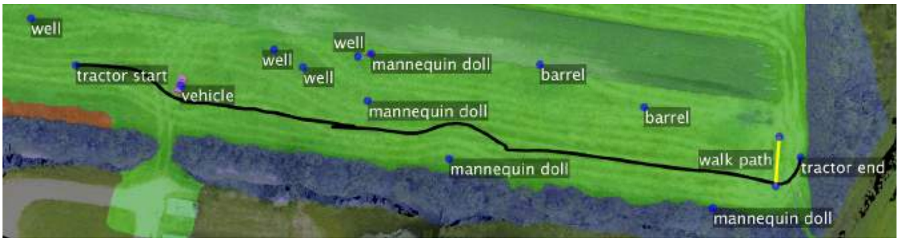


Figure 5. Orthophoto with static objects, tractor trajectory (black line) and human walk path (yellow line). An overlay shows the ground truth of vegetation (blue), ground (green) and non-traversable ground (red).

3.2. Evaluation and Results

To obtain the mapping results, the ISM methods are applied to their specific sensors to extract the measurement grids. To locate the measurement grid inside the current patch and globally, the extended Kalman filter by (Moore and Stouch 2016) is used, taking GPS, IMU, and GPS carrier measurements (Bevly and Cobb 2010) into account. As proposed in (Korthals, Skiba, and Krause 2016), multiple layers N of maps are needed to respect a diverse and heterogeneous sensor setup. This is used to overcome the drawback of the Bayesian update equation, which does not respect different sensor impacts or update rates. Thus, across each of the $N = 15$ sensor-algorithm-class sets, fusion is performed at a later stage by composing cell probabilities. In our implementation, two different fusion techniques are applied: First, the fusion based on a Superbayesian Independent Opinion Pool formula P_B (Pathak et al. 2007). It is applicable for the case when separate occupancy grids with identical feature representations (e.g. set of maps for class “obstacle”) are maintained. Second, a non-Bayesian fusion methods by taking the maximum P_M is applied to heterogeneous feature representations (e.g. set of maps for “vehicle” and “human”). It is worth mentioning that these fusion techniques are again cell-wise and therefore online applicable.

$$P_B(m) = \frac{\prod_N P_n(m)}{\prod_N P_n(m) + \prod_N (1 - P_n(m))}, \qquad P_M(m) = max_n P_n(m)$$

As evaluation metrics, precision, recall, F1 score, accuracy, True-Positive-Rate (TPR) and False-Positive-Rate (FPR) of the Receiver-Operator-Characteristic (ROC), and normalized entropy are calculated for all detected cells. For the given algorithms and sensors, the fusion and evaluation scores are not directly applicable. Even if the Bayesian framework allows the representation of the presence and absence of a feature, some algorithms do not make use of it. To name two examples, the LIDAR allows the deduction of free or occupied space based on its physical measurement principle. On the other hand, a camera based algorithm is fairly good for detecting the presence of a class, but easily fails in detecting the absence, due to e.g. a possible lack in the training set. Thus, the metrics recall, F1 score, TPR, and FPR can be calculated for LIDAR based detections, but not for camera and RADAR based detections. To give a better interpretation, the normalized entropy H_N of all true negative and true positive classified cells is used to calculate the remaining uncertainty normalized by a completely unknown map:

$$H(P(M)) = -\sum_{c \in M} P(c) \log(P(c)) + (1 - P(c)) \log(1 - P(c)), \quad H_N(P(M)) = H(P(M)) / H(P(M) \equiv 0.5)$$

This gives a quantitative value of the information gain among different setups where the range of the normalized entropy reaches from 0, meaning that there is no unknown space left, to 1, meaning the map is completely unknown.

Layers produced by the same sensor are fused by the maximum method to get a competitive fusion across algorithms, and the outcome of these layers is fused by the Superbayesian method to get a complementary fusion across different sensors as shown in Figure 6.

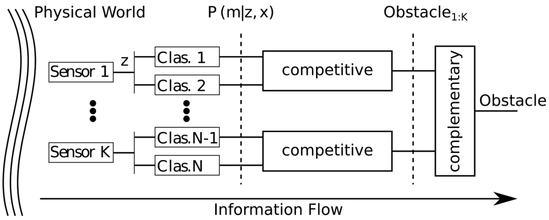


Figure 6. Fusion framework

Table 1. List of sensor setups. 1-3 use competitive fusion across classes, whereas 4-7 use complementary fusion.

Setup	Fusion	Sensors	Detection Algorithm	Input Classes	Output Classes
		Camera	SS	shelterbelt, human, vehicle	
1	Competitive	Camera	YOLO	human, vehicle	obstacle_C
		Camera	PED	human	
2	Competitive	LIDAR	SVM	object, vegetation	obstacle_L
3	Competitive	RADAR	KMA	radar	obstacle_R
4	Complementary	Camera, LIDAR	-	obstacle_C, obstacle_L	obstacle
5	Complementary	LIDAR, RADAR	-	obstacle_L, obstacle_R	obstacle
6	Complementary	Camera, RADAR	-	obstacle_C, obstacle_R	obstacle
7	Complementary	Camera, LIDAR, RADAR	-	obstacle_C, obstacle_L, obstacle_R	obstacle

Table 2. Evaluation scores for the different sensor setups (ill-defined scores omitted by “-”)

Setup	Fusion	Precision	Recall	F1 score	Accuracy	TPR	FPR	Entropy
1	Maximum	0.889	-	-	0.889	-	-	0.984
2	Maximum	**0.897**	0.922	0.910	0.957	0.922	**0.0320**	0.821
3	Maximum	0.789	-	-	0.789	-	-	0.991
4	Superbayes	0.896	0.941	0.918	0.960	0.941	0.0342	0.819
5	Superbayes	0.889	0.944	0.916	0.960	0.944	0.0357	0.820
6	Superbayes	0.827	-	-	0.827	-	-	0.979
7	Superbayes	0.889	**0.958**	**0.922**	**0.961**	**0.958**	0.0376	**0.818**

For the evaluation, a constant map resolution of 10 cm per cell is used. To measure the impact of each sensor, all permutations of the sensors (camera, LIDAR, and RADAR) are performed as shown in Table 1. Particularly for the camera based detection, only classes representing objects are taken into account. For setup 1, 2, and 3, the fusion P_M is applied competitively, outputting "obstacle_C", "obstacle_L" and "obstacle_R" for camera, LIDAR, and RADAR respectively. These outputs are then fed into the complementary fusion P_B , outputting "obstacle". The results for all different setups are shown in Table 2. The first noticeable fact is the decrease of entropy for every complementary fusion. This shows, that with the introduction of new sources of information, the unknown area is reduced. Thus, the lowest entropy is evaluated for setup 7. The same is the case for the other scores, where setup 7 performs the best. The only exceptions arise for precision and FPR. For precision, the LIDAR performs better, but also has a bad recall resulting in the worst F1 score. This coincides with the FPR, as the number of misclassifications may rise with more sensors coming into play due to the fact, that in the evaluation scenario the sensor lobes do not fully overlap at all positions. Therefore, wrong classifications can not be corrected by sensor fusion.

As can be seen in Figure 7, misclassifications occur mainly at object borders. Due to the fact that the errors are evenly distributed around them, it can be assumed that they are caused by statistical errors from the sensors, the detection algorithm, or the vehicle's position uncertainty. To quantify this error, the standard deviations of all distinctive misclassified regions across obstacle borders are averaged with the result of $\sigma = 0.332\,\text{m}$. In Figure 8, the final fused detection of all obstacle layers can be seen. To highlight one example, the car is almost perfectly detected with the only exception of the tail. Having in mind that the upper right edge of the car has not been seen by any sensor, the result of the fusion concept is even more convincing.

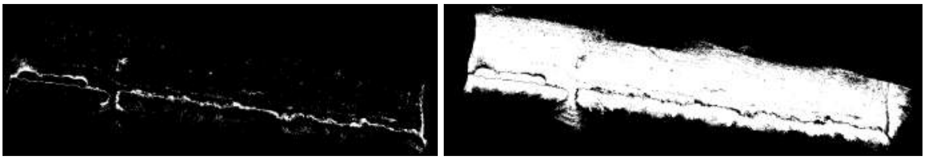


Figure 7. Binary mask created by setup 7 of false (left) and correct (right) classifications

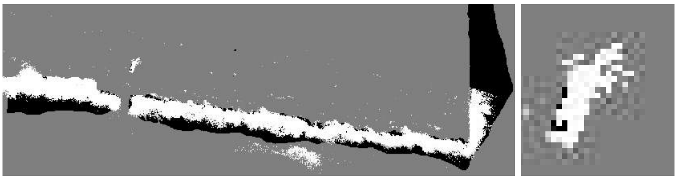


Figure 8. Left: Ground truth (black) with overlaid obstacle detection (white) by setup 7. Right: Magnified area of the car

4. **Conclusions**

In this work, we have presented a global mapping approach fusing information from a monocular color camera, a RADAR, and a LIDAR. For each sensor, we have introduced detection algorithms, mapping from raw sensor data to a number of 2D grid-based obstacle interpretations of the environment, such as "human", "vehicle", and "vegetation". These representations are first fused competitively for each sensor to provide a sensor-specific obstacle representation. Then, complementary fusion is used to fuse across sensor modalities, providing a final combined obstacle interpretation.

Based on data from a grass mowing scenario with various static obstacles, we have evaluated the proposed mapping approach for all combinations of sensors. We have shown that any combination of sensors performs better than the same sensors individually, and that we achieve a mapping accuracy for detected cells of 96% and an F1 score of 92%, when combining information across all three sensors. Future work will focus on introducing dynamic obstacles and training the fusion algorithm to weigh information from sensors and algorithms individually. Also, a more comprehensive evaluation from different fields and sensor setups is planned, investigating generalization performance of the proposed method.

Acknowledgements

This research is sponsored by the Innovation Fund Denmark as part of the project "SAFE - Safer Autonomous Farming Equipment" (project no. 16-2014-0).

This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster "Intelligent Technical Systems OstWestfalenLippe" (it's OWL) and managed by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the contents of this publication.

References

Ahtiainen, J., T. Peynot, J. Saarinen, S. Scheding, and A. Visala. 2015. “Learned Ultra-Wideband RADAR Sensor Model for Augmented LIDAR-Based Traversability Mapping in Vegetated Environments.” In *Information Fusion (Fusion), 2015 18th International Conference on*, 953–60.

Bertozzi, M., and A. Broggi. 1996. “Real-Time Lane and Obstacle Detection on the GOLD System.” *Proceedings of Conference on Intelligent Vehicles*. doi:10.1109/IVS.1996.566380.

Bevly, D. M., and S. Cobb. 2010. *GNSS for Vehicle Control*. GNSS Technology and Applications Series. Artech House.

Christiansen, P., M. K. Hansen, K. A. Steen, H. Karstoft, and R. N. Jørgensen. 2015. “Advanced Sensor Platform for Human Detection and Protection in Autonomous Farming.” In *Precision Agriculture ’15*, 291–98.

Christiansen, P., R. Sørensen, S. Skovsen, C. D. Jæger, R. N. Jørgensen, H. Karstoft, and K. A. Steen. 2016. “Towards Autonomous Plant Production Using Fully Convolutional Neural Networks.” In . Aarhus University.

Dalal, Navneet, and Bill Triggs. 2005. “Histograms of Oriented Gradients for Human Detection.” In *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*. doi:10.1109/CVPR.2005.177.

Elfes, Alberto. 1990. “Occupancy Grids: A Stochastical Spatial Representation for Active Robot Perception.” In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*.

Everingham, Mark, Sma Eslami, and Luc Van Gool. 2013. “The Pascal Visual Object Classes Challenge–a Retrospective.” *Homepages.Inf.Ed.Ac.Uk*. doi:10.1007/s11263-014-0733-5.

Freund, Yoav, and Robert E. Schapire. 1996. “Experiments with a New Boosting Algorithm.” In *ICML*, 96:148–56.

Hähnel, Dirk. 2004. “Mapping with Mobile Robots.”

Kohlbrecher, Stefan. 2011. “Grid-Based Occupancy Mapping and Automatic Gaze Control for Soccer Playing Humanoid Robots.” *... Humanoid Soccer Robots ...*, no. October.

Konrad, Marcus, Magdalena Szczot, Florian Schüle, and Klaus Dietmayer. 2011. “Generic Grid Mapping for Road Course Estimation.” *IEEE Intelligent Vehicles Symposium, Proceedings*, no. Iv: 851–56.

Korthals, Timo, Andreas Skiba, and Thilo Krause. 2016. “Evidenzkarten-Basierte Sensorfusion Zur Umfelderkennung Und Interpretation in Der Ernte.” In *Informatik in Der Land-, Forst Und Ernährungswirtschaft*, 15–18.

Kragh, Mikkel, Rasmus N. Jørgensen, and Henrik Pedersen. 2015. “Object Detection and Terrain Classification in Agricultural Fields Using 3D Lidar Data.” In *Computer Vision Systems*, 188–97. Lecture Notes in Computer Science. Springer International Publishing.

Long, Jonathan, Long Jonathan, Shelhamer Evan, and Darrell Trevor. 2015. “Fully Convolutional Networks for Semantic Segmentation.” In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2015.7298965.

Moore, Thomas, and Daniel Stouch. 2016. “A Generalized Extended Kalman Filter Implementation for the Robot Operating System.” In *Intelligent Autonomous Systems 13*, edited by E. Menegatti, N. Michael, K. Berns, and H. Yamaguchi, 335–48. Advances in Intelligent Systems and Computing 302. Springer International Publishing.

Mottaghi, Roozbeh, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. 2014. “The Role of Context for Object Detection and Semantic Segmentation in the Wild.” In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 891–98. IEEE.

Munkres, James. 1957. “Algorithms for the Assignment and Transportation Problems.” *Journal of the Society for Industrial and Applied Mathematics* 5 (1): 32–38.

Nam, Woonhyun, Piotr Dollár, and Joon Hee Han. 2014. “Local Decorrelation For Improved Detection.” *Advances in Neural Information Processing Systems*, 1–9.

Pathak, Kaustubh, Andreas Birk, Jann Poppinga, and Sören Schwertfeger. 2007. “3D Forward Sensor Modeling and Application to Occupancy Grid Based Sensor Fusion.” *Proceedings of the ... IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE/RSJ International Conference on Intelligent Robots and Systems* 2: 2059–64.

Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. “You Only Look Once: Unified, Real-Time Object Detection.” http://arxiv.org/abs/1506.02640v3.

Reina, Giulio, and Annalisa Milella. 2012. “Towards Autonomous Agriculture: Automatic Ground Detection Using Trinocular Stereovision.” *Sensors* 12 (9). Molecular Diversity Preservation International: 12405–23.

Stachniss, Cyrill. 2009. *Robotic Mapping and Exploration*.

Winner, Hermann. 2015. *Handbuch Fahrerassistenzsysteme - Grundlagen, Komponenten Und Systeme Für Aktive Sicherheit Und Komfort*.

Wu, Ting-Fan, Chih-Jen Lin, and Ruby C. Weng. 2004. “Probability Estimates for Multi-Class Classification by Pairwise Coupling.” *Journal of Machine Learning Research: JMLR* 5 (December). JMLR.org: 975–1005.