Collaborative Unmanned Air and Ground Vehicle Perception for Scene Understanding, Planning and GPS-denied Localization

Gordon A. Christie

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Engineering

Kevin Kochersberger, Co-Chair
Dhruv Batra, Co-Chair
Devi Parikh
Pratap Tokekar
Pinhas Ben-Tzvi

November 7, 2016
Blacksburg, Virginia

Keywords: Scene Understanding, Semantic Segmentation, Unmanned Systems, UAV, UGV, Path Planning

Collaborative Unmanned Air and Ground Vehicle Perception for Scene Understanding, Planning and GPS-denied Localization

Gordon A. Christie

ABSTRACT

Autonomous robot missions in unknown environments are challenging. In many cases, the systems involved are unable to use *a priori* information about the scene (*e.g*. road maps). This is especially true in disaster response scenarios, where existing maps are now out of date. GPS-denied areas are another concern, especially when the involved systems are tasked with navigating a global path planned by a base station. Scene understanding via robots' perception data can greatly assist in overcoming these challenges. This dissertation makes three contributions that help overcome these challenges, where there is a focus on the application of autonomously searching for radiation sources with unmanned aerial vehicles (UAV) and unmanned ground vehicles (UGV) in unknown and unstructured environments. While LiDAR is used in the experiments presented, there is generally a push toward vision-based solutions. The three main contributions of this dissertation are: (1) An approach to overcome the challenges associated with constructing a joint model for simultaneous reasoning about different modules of perception (*e.g*. segmenting image data and point clouds). (2) Algorithms and experiments involving scene understanding for real-world autonomous search tasks. The experiments involve a UAV and a UGV searching for potentially hazardous sources of radiation is an unknown environment. (3) An approach to the registration of a UGV in GPS-denied environments using image and LiDAR data, where registration is performed in a 2.5D overhead map generated from imagery captured by a low-flying UAV.

Collaborative Unmanned Air and Ground Vehicle Perception for Scene Understanding, Planning and GPS-denied Localization

Gordon A. Christie

GENERAL AUDIENCE ABSTRACT

Autonomous robot missions in unknown environments are challenging. In many cases, the systems involved are unable to use *a priori* information about the scene (*e.g*. road maps). This is especially true in disaster response scenarios, where existing maps are now out of date. Areas without GPS are another concern, especially when the involved systems are tasked with navigating a path planned by a remote base station. Scene understanding via robots' perception data (*e.g*. images) can greatly assist in overcoming these challenges. This dissertation makes three contributions that help overcome these challenges, where there is a focus on the application of autonomously searching for radiation sources with unmanned aerial vehicles (UAV) and unmanned ground vehicles (UGV) in unknown and unstructured environments. The three main contributions of this dissertation are: (1) An approach to overcome the challenges associated with simultaneously trying to understand 2D and 3D information about the environment. (2) Algorithms and experiments involving scene understanding for real-world autonomous search tasks. The experiments involve a UAV and a UGV searching for potentially hazardous sources of radiation is an unknown environment. (3) An approach to the registration of a UGV in areas without GPS using 2D image data and 3D data, where localization is performed in an overhead map generated from imagery captured in the air.

Dedication

To my parents, brother, and wife.

Acknowledgments

There are many individuals to thank for the successful completion of this degree and my well-being during the process. While I will not list them all, I will offer a general thank you to all of those that helped me. Here are some honorable mentions for those I feel directly helped me finish this degree:

- Kevin Kochersberger (my advisor). Thank you for supporting me throughout the majority of my graduate career and believing in me. I greatly appreciate you providing me with the flexibility to explore many research ideas, providing guidance, and generally being a decent and understanding person.

- Dhruv Batra (co-advisor) and Devi Parikh (committee member). Thank you for your guidance, invaluable understanding of computer vision and machine learning, and providing me with opportunities to publish at top conferences. I can feel your influence every time I approach a new problem or work on a paper. I can finish in peace knowing that the holistic paper was accepted somewhere.

- Pratap Tokekar and Pinhas Ben-Tzvi (committee members). Thank you for being on my committee and for all of the feedback on my projects and dissertation.

- Sponsors. I would like to thank the Defense Threat Reduction Agency, the U.S. Army Research Laboratory (ARL), and the Center for Unmanned Aircraft Systems for the funding for the funding I received. Also, thank you Stuart Young for inviting me to spend the summer at ARL.

- Friends and Co-workers. There are too many graduate students to list in this section, but I am forever grateful to so many of them. The Unmanned Systems Lab provided both talented co-workers and friendships. Special thanks to my lab/"the closet"-mates Alfred (stepping stone), Brian and Haseeb for the distractions and hilarious conversations over the last couple years. Haseeb, don't be 40 years late for teaching me how to fly a quad. Thanks to all of the current and former CVMLP lab members for their help. Stan, thank you for being a good friend and putting up with me in the CEL and inside and out of the computer vision world. Thank you Aishwarya and Ankit for all of your help on the holistic project. The paper would not have been accepted without you guys. I am also grateful to all of the talented people at the U.S. Army Research Laboratory that I had the opportunity to work with during my internship. I'd like to especially thank Garrett for all of his helpful comments and suggestions that were incorporated into part of the work presented in this dissertation.

- My family. Words cannot express my appreciation for all you have done for me. To my parents and brother, thank you for all of your care throughout my life. To my loving wife, you are the most important person in my time at Virginia Tech. Your love and support are the reasons I have managed to make it through my time here.

Contents

List of Figures

List of Tables

Nomenclature

ALE	Automatic Labeling Environment [6]
COTS	Commercial off-the-shelf
DEM	Digital elevation map
DySMAC	The Center for Dynamic Systems Modeling and Control
GPS	Global positioning system
IMU	Inertial measurement unit
LiDAR	Light Detection and Ranging
NLP	Natural Language Processing
PCL	Point Cloud Library [7]
PMVS	Path-based Multi-view Stereo [3]
RANSAC	Ransom sample consensus
RBF	Radial basis function
RGB	Red Green Blue (color model)
SfM	Structure from motion
SIFT	Scale invariant feature transform [8]

SLAM	Simultaneous localization and mapping
SVM	Support vector machine
TSP	Traveling salesman problem
TURTLE	Terrestrial Unmanned Robots for Teamed Learning and Exploration
UAV	Unmanned aerial vehicle
UGV	Unmanned ground vehicle

Chapter 1

Introduction

Autonomous unmanned systems have the potential to provide safer and more efficient solutions to problems that currently rely on manned missions. Relevant applications include disaster response, search and rescue, public transportation, infrastructure health monitoring and precision agriculture. The application focused on for this dissertation is the autonomous search for hazardous sources of radiation. In order to autonomously search for such sources, the unmanned systems involved in the missions must be able to perceive the scene they are scouting. This dissertation provides methods to improve perception systems used for autonomous unmanned air and ground systems collaborating to perform this task. The specific focus is on improving the ability to perform simultaneous reasoning about 2D (*e.g*. RGB image) and 3D (*e.g*. LiDAR, image-derived point clouds) data. The world around us is incredibly complex, and therefore building accurate perception systems for most applications is very challenging, especially when the systems operate in unstrucured environments. In addition, some areas that these unmanned systems are operating in are GPS-denied, and therefore alternative methods of localization are required. This dissertation addresses three different problems that arise when autonomous systems are collaborating to search for radiation sources in a GPS-denied environment with no *a*

priori information about the scene. The problems addressed are:

1. **Scene Understanding**. This part of the dissertation focuses on joint reasoning for multiple perception modules. For perception tasks, autonomous vehicles typically must reason about multiple modules of perception. As an example, say we have one module (module A) performing semantic segmentation for 2D RGB images, and another module (module B) performing semantic segmentation for 3D point clouds, where some or all of the 3D points are visible in the 2D image. Semantic segmentation here means classifying the 2D pixels of the images, and the 3D points of the point clouds with semantic labels (*e.g.* road). Constructing a joint model that simultaneously performs reasoning for these tasks can be difficult. A joint model must reason about all possible solutions in module A (2D semantic segmentations) and all possible solutions in module B (3D semantic segmentations) simultaneously, which leads to a search space explosion. However, there is still a need to exchange information between the modules, as these modules have complementary strengths. Even the state of the art semantic segmentation models for these tasks make mistakes, and having each module output their most likely belief in isolation will likely mean less accurate solutions.
2. **Planning**. This part focuses on using a 3D semantic understanding of a scene developed by a UAV to instruct a UGV to visit points of interest on the ground using path planning. The specific application of interest for this part is the autonomous search for potentially hazardous radiation sources, and how we can use UAV and UGV to find them.
3. **GPS-denied Localization**. When a robot is executing a planned mission, it is critical for it to be able to access its position and heading information for the scene it is operating within. Typically unmanned systems rely on GPS to perform local-

ization. However, GPS is not always available, and therefore alternate methods of localization must be considered. Using perception data to perform registration in an overhead map is possible, but existing maps may be out of date.

1.1 Background

There is a growing interest in advancing perception capabilities for autonomous systems. From self-driving passenger vehicles to UAV inspection of infrastructure, there is a long list of applications for autonomous robots that continues to grow. We can divide the environments for these applications into two categories: structured and unstructured. An example of an application in a structured environment would be self-driving passenger vehicles. The vehicles for this task typically have *a priori* information about the scene (*e.g*. road maps) and a built in understanding of the traffic laws for the area it is operating within. An example of an application that involves an unstructured environment would be post-disaster surveying in a GPS-denied environment where existing maps are now meaningless. This dissertation focuses more on advancing perception capabilities for systems in unstructured environments.

Semantic segmentation is used in all chapters of this dissertation. This is the task of classifying each point in the data (*e.g*. pixel in an image, 3D point in a point cloud) with a semantic category (*e.g*. building). Many existing works have focused on this problem, from building better segmentation models to better perform on datasets with a general set of categories (*e.g*. PASCAL [9]) [10] to building better models for segmenting specific categories for specific tasks, such as autonomous driving [11, 12]. In this dissertation, the Automatic Labeling Environment (ALE) [6] is used to perform semantic segmentation. This code does not run in real-time. However, while custom segmentation approaches

were developed and tested, they did not out perform ALE, and were therefore not used. In the future, it is likely that much larger datasets will be publicly released for the tasks presented in the following chapters that will allow for real-time deep learning approaches to be used for semantic segmentation. These approaches will be able to replace ALE in the work presented in this dissertation for better performance of each system.

Existing methods for image-based 3D reconstructions are also used in this dissertation. The two main methods of reconstructing a scene from imagery are stereo vision and structure from motion (SfM). Stereo vision estimates the depth from images captured by two or more calibrated cameras at the same time. In SfM, points in the images that represent the same point in the scene are matched in two or more images in a sequence of images taken at different times. Advantages and disadvantages of each method will be discussed, as well as the specific algorithms used. Image-based 3D reconstruction is still an active area of study in the computer vision community, and therefore we can expect these methods to become faster and more accurate. Similar to the discussion of semantic segmentation above, these new 3D reconstruction methods will be able to replace the ones used in this dissertation to improve the performance of each system.

1.2 Contributions

This dissertation makes three main contributions by addressing each of the three problems listed above. These contributions are:

1. An approach to joint reasoning for multiple perception modules. This approach to holistic scene understanding overcomes the challenges associated with constructing a joint model that simultaneously reasons about two or more separate perception tasks. Our approach has been shown to work for language and vision exper-

iments [13], and simultaneous RGBD semantic segmentation and object support estimation [19]. An urban scene understanding experiment is presented, where the model simultaneously reasons about 2D RGB images in a 2D semantic segmentation module, and 3D point clouds (derived from stereo vision) in a 3D semantic segmentation module.

2. A method to autonomously estimate and confirm the locations of radiation sources with UAV and UGV using 3D scene understanding. Scene understanding is used to understand an unknown outdoor environment using aerial imagery with a supervised machine learning approach. We incorporate aerial semantic segmentation results into the A* path planning algorithm so that a UGV will prefer to follow roads over grass and stay clear of obstacles. We also demonstrate the ability to detect obstacles locally on the ground with LiDAR and then find a path around the obstacle using both local and global information.
3. An approach to localize a UGV in a GPS-denied environments. A multi-robot system capable of autonomously understanding a scene in GPS-denied environments via joint semantic reasoning about the scene from appearance and depth data is presented. A UGV localization algorithm is also presented, which is shown to localize a UGV in an urban environment with an average difference to GPS under 5m, where the algorithm is robust to appearance-based scene changes, small structural scene changes, and occasional ambiguous regions.

1.3 Organization of the Dissertation

The rest of this dissertation is organized as follows:

- Chapter 2. This provides an overview of the related work for each of the three

contributions presented.

- Chapter 3 (first contribution). The approach to joint reasoning for multiple perception modules is presented, along with an urban scene understanding experiment.

- Chapter 4 (second contribution). Here the approach and experiments to radiation search operations using scene understanding in an unknown environment using a UAV and UGV are presented.

- Chapter 5 (third contribution). In this chapter, the approach to localizing a UGV in GPS-denied environments is presented.

- Chapter 6. This chapter discusses the major conclusions of the dissertation and opportunities for future work.

Chapter 2

Literature Review

This chapter provides an overview of the related work for each of the three main contributions of this dissertation.

2.1 Joint Reasoning for Multiple Perception Modules

Works related to the high-level theme of joint reasoning across multiple modules are described first. After, specific papers relevant to urban scene understanding experiment, presented in the following chapter, are discussed.

Holistic Perception and Joint Reasoning. Recent works have looked at both joint [20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30] and sequential/cascaded [31, 32, 33, 34, 35, 36] reasoning over different perception modules (*e.g.*, segmentation, recognition, depth estimation, and scene classification). Both directions have their own shortcomings. Joint modeling involves constructing a single (usually restrictive) probabilistic model that reasons about all variables in all modules at the expense of performance-limiting independence assumptions and limited interactions to ensure tractable inference. Sequential/cascaded models abandon the probabilistic joint-prediction framework altogether and simply feed

the output of one module into another, but this results in propagation of errors and general mismanagement of uncertainty. Our proposed approach leverages the best of both worlds and has the potential to bring about a paradigm shift. Each perception module may be treated as a sophisticated black box, as long as it allows computation of a set of plausible hypotheses. A second layer on top is then free to perform joint reasoning between these modules since the search space is limited to the Cartesian product of these hypothesis sets. Perhaps the closest to our goal is [31], where the single "most probable" outputs of different modules are repeatedly fed as features to other modules in a cascaded manner. Our approach is orthogonal – we have a shallow 2-layer cascade, but each module produces a set of plausible solutions, not just a single one. Combining the two ideas to produce a deep cascade where each step produces multiple hypotheses is an interesting direction for future work.

Comparison to Boosting [37] and Mixture of Experts (MoE) [38]. AdaBoost involves sequentially training weak learners that are all solving the same problem, and finally providing the weighted average of their predictions. Mixture of Experts involves independently training multiple experts/models that all solve the same problem, with a final prediction picked via a 'gating function'. In our approach, a large problem is broken into smaller modules, which work independently to produce plausible solutions and then communicate to pick the best tuple of solutions. Neither of these ideas (decomposition, consistency-check) are present in AdaBoost. At a high-level our approach may be thought of as a gating function, but the idea of decomposition into modules is novel and crucial as we move towards more sophisticated problems. We do not have weak learners, but sophisticated systems that are difficult to integrate except via diverse solutions.

Urban Scene Understanding. In recent years, several street-view datasets have become available [39, 11, 40, 41, 42] and significant progress has been made on the semantic

segmentation of street scenes [43, 44, 45, 46, 11, 42, 47]. One similar work to ours for this task is by Munoz *et al*. [42], where a co-inference approach was used to jointly perform semantic segmentation of 2D RGB images and 3D point clouds from a laser scanner. They exchange information between these two modalities during inference, whereas we exchange this information after generating a diverse set of hypotheses within each module and selecting a consistent tuple.

2.2 Scene Understanding for Radiation Search Operations

Unmanned Systems Collaboration. The collaboration between autonomous unmanned systems has been studied for a large number of applications. These unmanned systems include autonomous underwater vehicles (AUV), unmanned surface vehicles (USV), unmanned aerial vehicles (UAV), and unmanned ground vehicles (UGV). Some examples of the applications of these unmanned systems are search and rescue operations, post-disaster surveying, target localization and tracking, and precision agriculture monitoring. Previous works have focused on the collaboration between multiple UAV [63, 64, 65, 66], multiple UGV [67, 68, 69, 70, 71], the collaboration between UAV and UGV [72, 73, 74, 75, 76], and much more. Garzon *et al*. present a solution for multiple UGV to perform signal searching tasks in large outdoor scenarios [77]. They propose different path planning strategies for coverage, which depend on the size and shape of the field. For the topic of UAV-UGV collaboration, Tokekar *et al*. studied the problem of coordinating UAV and UGV for precision agriculture [74], where they found energy efficient ways to visit areas with misclassified nitrogen levels. UAV and UGV have also been used in a collaborative manner to perform target localization [78, 79]. In [80] a mock-up disaster scenario was setup, where a UAV maps the area and then computes the fastest mission for a UGV to

reach the destination and deliver a first-aid kit. Cooperative environment mapping [81] and surveillance [82] have also been studied. In our experiments we are interested in simply reaching a destination and returning to the start position, but note that our semantic segmentation results could be used to find better paths to take for such tasks in outdoor environments. While our experiments are farily specific, and therefore difficult to compare to existing approaches, Schneider *et al*. discuss how EURATHLON and ELROB have provided a way of standardizing and benchmarking the evaluation of methods in outdoor robotics through competition [83]. Teams at these competitions build impressive systems that are capable of executing missions in real-time for important tasks such as search and rescue. Others have used overhead imagery to improve UGV path planning capabilities. In [84], a self-supervised online learning algorithm is used on a UGV to learn a model that integrates information about the current terrain and overhead imagery that is then used to predict traversal costs at other regions in the overhead map. These predicted traversal costs were then used to perform path planning. While many of these works demonstrate successful collaboration between UAV and UGV, we try to focus more on using semantic segmentation for scene understanding in a real-world search task by training on a dataset of imagery that is annotated with semantic categories. As more images are captured an annotated by low-flying aircraft, we believe it will be important to integrate existing models with online learning algorithms, such as the one presented in [84]. These models will be able to provide valuable context to a UGV during tasks such as radiation search, as existing maps (*e.g*. satellite) may be too old to capture important information about the scene.

Scene Understanding. Perception for autonomous robotic systems has seen tremendous progress in many applications. A variety of possible sensing methods (RGB-depth sensors, visual cameras, acoustic sensors, LiDARs, *etc.*) have allowed these systems to

perceive the world and make intelligent decisions. Semantic segmentation has been the focus of many works, with state-of-the-art models [10, 85] capable of achieving high accuracy for many different tasks, and large datasets with semantic annotations available for training and evaluation [86, 9]. However, to the best of our knowledge, no publicly available dataset of semantically annotated images from low-flying UAV currently exists. In this work we create our own dataset to train a model to perform semantic segmentation with 2D color images and ground truth annotations, then evaluate on unseen image tiles of the orthophoto for the area in which we are searching for hazardous radiation sources.

Similar to this work, Montoya-Zegarra *et al*. explored road mapping [87] and the semantic segmentation of aerial images with higher-order cliques [88]. Radford studied the problem of real-time roadway classification from aerial imagery for UGV path planning [89], where k-means clustering and image mosaicking were used. This approach, however, relies on an initialization step where the algorithm is first shown which cluster is a road. Our approach uses supervised learning to perform semantic segmentation of aerial imagery for several categories, which tends to scale well and requires no human supervision at test time. Supervised classification of LiDAR point clouds has also been studied [90, 91]. Joint semantic segmentation of 2D and 3D data simultaneously has been the focus of several other works. Floros *et al*. presented an approach to perform semantic segmentation of 2D images and 3D point clouds generated from stereo pairs with a joint model that incorporated temporal consistency between subsequent frames [11]. Munoz *et al*. developed an approach to jointly perform semantic segmentation of 2D images and 3D LiDAR point clouds by integrating information between overlapping parts of the scene [42]. In a work by Sturgess *et al*., structure from motion features were incorporated into the semantic segmentation of road scenes [92]. In this work, however, we do not perform additional semantic segmentation from the ground. The LiDAR on the

TURTLE is used to detect obstacles on its current path by analyzing elevation gradients, which we found sufficient for our task. For the semantic segmentation of the orthophoto generated from the imagery captured by the RMAX, we have a two-stage approach where we analyze the DEM separately to make better category predictions at each pixel. Ideally we would implement a joint framework, such as the ones presented in [11, 42], but since this requires much more data than we have available we use do not do this.

Yingze *et al*. presented an approach to generate image-based 3D reconstructions while recovering the locations, poses, and categories of objects in a scene [93]. In a work by Kundu *et al*., an approach was presented for joint inference of 3D scene structure and semantic segmentation of urban street scene imagery [43]. Incorporating semantic maps into path planning for mobile robots has also been studied. Hatao *et al*. proposed a semantic map making system based on road structures, where trajectories of moving objects, landmarks, building entry points, and traffic signs are added to the map [94]. They combine laser range finders with an omnidirectional camera for perception on the robot.

Others have also studied optimal camera positions for UAV collecting imagery to be used for image-based 3D reconstructions [95]. While this is ideal for generating a better orthophoto and DEM, navigating to 3D positions not on scan lines with the same altitude increases the amount of time to complete the mission, and makes analyzing the radiation spectral data more difficult. We therefore use scan lines when planning the missions for the RMAX.

Radiation Sensing. There has also been research on using UAV and UGV for radiation mapping missions. Kochersberger *et al*. studied mapping radiation levels in an unknown environment using a UAV to collect radiation data from the air and deploy a tethered UGV to collect samples from the ground [56]. While similar to this work, their work focused on active radiation search strategies with no focus on the planning for the tethered UGV

to reach the destination. In this chapter, we present a full system that performs an analysis of the radiation data after the UAV lands, plans a path for a UGV to visit points of interest collecting additional radiation data while avoiding obstacles on the way to the destination. We also use vision-based scene understanding to complete the missions, which allows for low-cost cameras to be used. Vetter *et al*. use an RMAX to map radiation and propose a "Nuclear Street View" [96]. Schneider *et al*. discuss possible scenarios for collecting radiation measurements with unmanned systems [97], where one type of scenario is the prevention of incidents involving radiation and the other post-incident analysis. Our experiments focus on the prevention scenario. Benedetto *et al*. developed an approach to identifying regions of interest in radiation data by means of clustering that is driven by diffusion operators as applied to a data graph representation of the collection of radiation spectra [98]. While more advanced reasoning could easily be incorporated into our search, such methods are not necessary to demonstrate the successful automation of the process of finding and localizing radiation anomalies. We find that the use of a simple approach based on the local maxima in the overall intensity (calculated as the sum of of the counts in all spectral channels for each measurement) to indicate potential source locations works well in our experiments. We instead focus more on augmenting semantic information into the search process. In another work by Schneider *et al*. [99], a prototype of an unmanned multi-robot reconnaissance system to detect chemical, biological, radiological, nuclear, and explosive (CBRNE) threats was presented, where the environment is not known a priori. Chemical and biological samples are obtained from the environment. Path planning is also performed so that trajectories can be generated to avoid obstacles.

Strategies for radiation search have also been explored. Cortez *et al*. propose two different motion planning strategies for building a radiation map [100]. One involves searching areas with higher uncertainty levels, and another involves visiting all cells in a grid

where the amount of time spent at each cell depends on the uncertainty. Minamoto *et al*. estimate the intensities of radiation sources on the ground surface in 3D using a dosimeter [101]. By moving the dosimeter around in 3D, they perform a MAP estimation of the source intensities by using characteristics of attentuation. In the work by Towler *et al*., present a grid-based robust Bayesian estimator to localize a single radiation source, and a contour analysis technique to localize an arbitrary number of radioactive sources [102]. All of these experiments were completed using simulated data. Brewer proposed a control strategy for a Yamaha RMAX unmanned helicopter to search for radiation sources using particle swarm particle filtering [103].

2.3 GPS-denied Localization of a UGV

The problem of localizing image and LiDAR data in overhead maps has been the focus of several previous works, including those that consider (1) global location estimation of images, (2) localization of image data in an overhead map of a local area, and (3) our problem; localizing UGV data in a local overhead map with high-level scene representations.

Global Localization of Images. The problem of directly estimating geo-location from images has been studied in several works [128, 129, 130]. In [131], deep convolutional neural networks (CNNs) are used to perform geolocalization of ground-level query images by matching to georeferenced aerial images. [132] use CNNs to recognize geo-informative attributes (*e.g*. population density). More recently, [133] used CNNs to perform global localization of an image, where they extend their model to incorporate an LSTM that reasons about temporal coherence to localize an entire photo album. For our task, we are focused on a small area of interest represented by a 2.5D orthophoto gen-

erated by low-flying UAV imagery. We estimate a precise location of the UGV, where we leverage 2D, 3D, and semantic information about the scene. With much more data, we believe these other approaches that recognize general areas could be integrated with our approach. A two-stage approach to localize a UGV precisely anywhere on the globe would then be possible.

Local Localization of Images. An approach to register video with structure from motion point clouds with temporal contstraints was developed by [134]. Contrary to our work, they use low-level representations of scene appearance, which we argue will fail in many scenarios. In [135], a vision-only approach was used to localize a UGV in a satellite image with manually-defined edges of buildings. Descriptor matching was performed, where descriptors describing a 360° view were calculated for pixels in the satellite image. Similar descriptors were calculated from the ground by identifying building edges in omnidirectional images taken from the on-board camera. Their work inspired a similar descriptor-based approach used in our work. However, our approach includes depth, semantic, and temporal information to perform localization. We also localize the UGV in an aerial map generated by UAV imagery with imperfect depth data, where we automatically label obstacles and semantic categories without human supervision.

Our problem. In [136, 137], road networks and visual odometry are used to perform localization with distributed computation for real-time performance. Our technique does not rely on *a priori* road network information. We believe the most similar work to ours is [138]. With a similar philosophy, they perform vision-based robot localization in a satellite image across seasons with segmentation outputs. However, they do not perform localization in a complicated urban environment and do not exploit elevation data available from the satellite view to asssist in localization.

Chapter 3

Joint Reasoning for Multiple Perception Modules

3.1 Introduction

In this chapter, we present an approach to holistic scene understanding that integrates information from multiple sub-components of perception or "modules" by reasoning about a diverse set of plausible hypotheses from each module. We present an urban scene understanding experiment, involving semantic segmentation of 2D and 3D data. We show that these modules have complementary strengths, and that joint reasoning produces more accurate results than any module operating in isolation. We also show that multiple hypotheses are crucial to multiple-module reasoning.

Perception and intelligence problems are hard. Whether we are interested in understanding an image or a point cloud, our algorithms must operate under tremendous levels of ambiguity. For instance, out of context, a patch from an image may seem like a face, but may simply be an incidental arrangement of tree branches and shadows, causing a

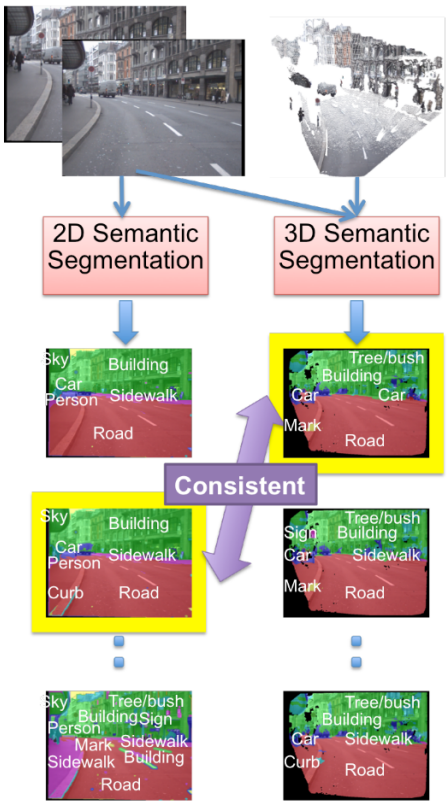


Figure 3.1: Overview of our proposed approach which was tested in an urban scene understanding experiment. We jointly reason about semantic segmentation for both 2D images and 3D point clouds generated using stereo vision. Each module produces M plausible hypotheses that we use for tractable joint reasoning. We develop a model, MEDIATOR, that scores all possible tuples for *consistency* and then picks the highest scoring one as our final output.

face detection system to produce nonsensical results, such as hallucinating faces floating on tree branches and building walls. Similarly, a cluster of 3D points in a laser scan captured by an unmanned ground vehicle may seem like a building, but actually be a truck waiting to drive into an intersection.

There are two main roadblocks that keep us from writing a single unified model (say a graphical model) to perform both tasks: (1) Inaccurate Models – empirical studies [14, 15, 16] have repeatedly found that models are often inaccurate and miscalibrated – their "most-likely" beliefs are placed on solutions far from the ground-truth.

(2) Search Space Explosion – jointly reasoning about multiple modalities is difficult due to the combinatorial explosion of search space ($\{$exponentially-many 2D segmentations$\}$ $\times$ $\{$exponentially-many 3D segmentations$\}$).

Proposed Solution and Contributions. In order to overcome these challenges, we propose to extract and leverage a small set of *diverse plausible hypotheses* or guesses from perception modules. Our main thesis is that such a set of plausible hypotheses can serve as a concise interpretable summary of uncertainty in perception modules (What does this module believe about the world?) and form the basis for tractable joint reasoning (How do we reconcile what module A believes about the world with what module B believes?). An illustration is shown in Figure 3.1.

Such a hypothesis set has the potential to overcome both barriers previously described (Inaccurate Models and Search Space Explosion). Producing multiple diverse plausible hypotheses increases the chances of extracting at least one accurate solution from the module. Moreover, joint reasoning over all perception modules may simply be restricted to the Cartesian product of the hypothesis sets – $\{$M-2D-segmentations$\}\times\{$M-3D-segmentations$\}$ – keeping the search space tractable.

Given n modules with M hypotheses each, how can we integrate beliefs across the modules and pick the best tuple (*e.g.*, 2D segmentation, 3D segmentation) of hypotheses? Our approach has been shown to work for simultaneous reasoning about language and vision [13], where inspiration for our approach came from psycholinguistic evidence suggesting that when people hear ambiguous words they momentarily assess and then rule out their irrelevant meanings [17, 18]. However, this approach is not limited to joint reasoning about language and vision. Our approach is general and extends to other applications that require joint reasoning about multiple perception modules where constructing a joint model is difficult or intractable. Our key focus is *consistency* – correct hypotheses from

different modules will be correct in a consistent way, but incorrect hypotheses will be incorrect in incompatible ways. Specifically, we develop a MEDIATOR model that scores tuples for consistency and searches over all M^n tuples to pick the highest scoring one.

We demonstrate our proposed approach on the following experiment:

- **Urban Scene Understanding** on CITY dataset [11]
  - Module 1: 2D Semantic Segmentation
  - Module 2: 3D Semantic Segmentation

However, the approach has been shown to work for other applications, as well [13, 19]. Our experimental setups span different modalities (2D images *vs*. 3D point-clouds) with the same output space (semantic segmentation labels).

We show that the different perception modules have complementary strengths and confusions. In all cases, our holistic reasoning approach produces more accurate results than any module operating in isolation. Our approach is fairly general and has the potential to impact a large array of applications.

3.2 Approach

In order to emphasize the generality of our approach, and to show that our approach is compatible with a wide class of implementations of 2D and 3D semantic segmentation, we present our approach with the modules abstracted as "black boxes" that satisfy a few general requirements and minimal assumptions. In Section 3.3, we describe each of the modules in detail, making concrete their respective features, and other details.

3.2.1 What is a Module?

The goal of a module is to take input variables $\mathbf{x} \in \mathcal{X}$ (images or point clouds), and predict output variables $\mathbf{y} \in \mathcal{Y}$ (2D semantic segmentation) and $\mathbf{z} \in \mathcal{Z}$ (3D semantic segmentation). The two requirements on a module are that it needs to be able to produce *scores* $S(\mathbf{y}|\mathbf{x})$ for potential solutions and a list of *plausible hypotheses* $\mathbf{Y} = \{\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^M\}$.

Multiple Hypotheses. In order to be useful, the set $\mathbf{Y}$ of hypotheses must provide an accurate summary of the score landscape. Thus, the hypotheses should be plausible (*i.e.*, high-scoring) and mutually non-redundant (*i.e.*, diverse). Our approach (described next) is applicable to any choice of diverse hypothesis generators. In our experiments, we use the DivMBest algorithm [49] for both semantic segmentation modules. Once we instantiate the modules in Section 3.3, we describe the diverse solution generation in more detail.

3.2.2 Joint Reasoning Across Multiple Modules

We now show how to intergrate information from both 2D and 3D segmentation modules. Recall that our key focus is *consistency* – correct hypotheses from different modules will be correct in a consistent way, but incorrect hypotheses will be incorrect in incompatible ways. Thus, our goal is to search for a pair (2D and 3D semantic segmentation) that is mutually consistent.

Let $\mathbf{Y} = \{\mathbf{y}^1, \ldots, \mathbf{y}^M\}$ denote the M 2D semantic segmentation hypotheses and $\mathbf{Z} = \{\mathbf{z}^1, \ldots, \mathbf{z}^M\}$ denote the M 3D semantic segmentation hypotheses.

MEDIATOR Model. We develop a "mediator" model that identifies high-scoring hypotheses across modules in agreement with each other. Concretely, we can express the MEDIATOR model as a factor graph where each node corresponds to a module (2D and 3D semantic segmentation). Working with such a factor graph is typically completely intractable because each node $\mathbf{y}, \mathbf{z}$ has exponentially-many states (image segmentations,

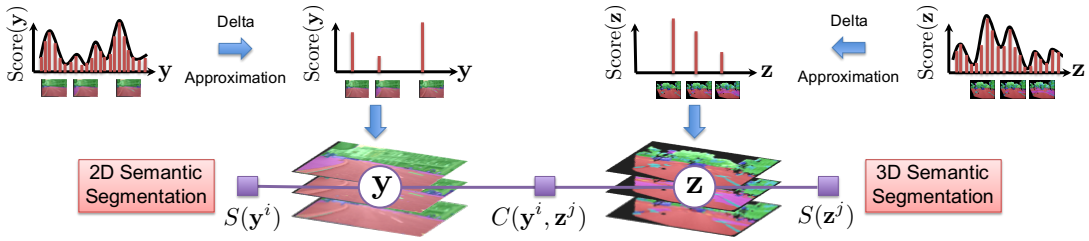


Figure 3.2: Illustrative inter-module factor graph. Each node takes exponentially-many or infinitely-many states and we use a 'delta approximation' to limit support.

point cloud segmentations). As illustrated in Figure 3.2, in this factor-graph view, the hypothesis sets $\mathbf{Y}, \mathbf{Z}$ can be considered 'delta-approximations' for reducing the size of the output spaces.

Unary factors $S(\cdot)$ capture the score/likelihood of each hypothesis provided by the corresponding module for the image/point cloud at hand. Pairwise factors $C(\cdot, \cdot)$ represent consistency factors. Importantly, since we have restricted each module variables to just M states, we are free to capture *arbitrary domain-specific high-order relationships* for consistency, without any optimization concerns. In fact, as we describe in our experiments, these consistency factors may be designed to exploit domain knowledge in fairly sophisticated ways.

Consistency Inference. We perform exhaustive inference over all possible tuples.

$$\underset{i,j \in \{1,\ldots,M\}}{\operatorname{argmax}} \Big\{ \mathcal{M}(\mathbf{y}^i, \mathbf{z}^j) = S(\mathbf{y}^i) + S(\mathbf{z}^j) + C(\mathbf{y}^i, \mathbf{z}^j) \Big\}. \tag{3.1}$$

Notice that the search space with M hypotheses each is M^2. In our experiments, we allow each module to take a different value for M, and typically use around 10 solutions for each module, leading to a mere 100 pairs, which is easily enumerable. We found that even with such a small set, at least one of the solutions in the set tends to be *highly accurate*, meaning that the hypothesis sets have relatively high recall. This shows the

power of using a small set of diverse hypotheses. For a large M, we can exploit a number of standard ideas from the graphical models literature (*e.g.* dual decomposition or belief propagation). In fact, this is one reason we show the factor in Figure 3.2; there is a natural decomposition of the problem into modules.

Training MEDIATOR. We can express the MEDIATOR score as $\mathcal{M}(\mathbf{y}^i, \mathbf{z}^j) = \mathbf{w}^\intercal \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}^i, \mathbf{z}^j)$, as a linear function of *score and consistency features* $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y}^i, \mathbf{z}^j) = [\phi_S(\mathbf{y}^i); \phi_S(\mathbf{z}^j); \phi_C(\mathbf{y}^i, \mathbf{z}^j)]$, where $\phi_S(\cdot)$ are the single-module (2D and 3D semantic segmentation) score features, and $\phi_C(\cdot, \cdot)$ are the inter-module consistency features. We describe these features in detail in the experiments. We learn these consistency weights $\mathbf{w}$ from a dataset annotated with ground-truth for the two modules $\mathbf{y}, \mathbf{z}$. Let $\{\mathbf{y}^*, \mathbf{z}^*\}$ denote the `oracle` pair, composed of the most accurate solutions in the hypothesis sets. We learn the MEDIATOR parameters in a discriminative learning fashion by solving the following Structured SVM problem:

$$\min_{\mathbf{w}, \xi_{ij}} \quad \frac{1}{2}\mathbf{w}^\intercal \mathbf{w} + C \sum_{ij} \xi_{ij} \tag{3.2a}$$

$$\begin{aligned} s.t. \quad & \underbrace{\mathbf{w}^\intercal \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}^*, \mathbf{z}^*)}_{\text{Score of } \texttt{oracle} \text{ tuple}} - \underbrace{\mathbf{w}^\intercal \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}^i, \mathbf{z}^j)}_{\text{Score of any other tuple}} \\ & \geq \underbrace{1}_{\text{Margin}} - \underbrace{\frac{\xi_{ij}}{\mathcal{L}(\mathbf{y}^i, \mathbf{z}^j)}}_{\text{Slack scaled by loss}} \qquad \forall i, j \in \{1, \ldots, M\}. \end{aligned} \tag{3.2b}$$

Intuitively, we can see that the constraint (3.2b) tries to maximize the (soft) margin between the score of the `oracle` pair and all other pairs in the hypothesis sets. Importantly, the slack (or violation in the margin) is scaled by the loss of the tuple. Thus, if there are other good pairs not too much worse than the `oracle`, the margin for such tuples will not be tightly enforced. On the other hand, the margin between the `oracle` and bad tuples will be very strictly enforced.

This learning procedure requires us to define the loss function $\mathcal{L}(\mathbf{y}^i, \mathbf{z}^j)$, *i.e.*, the cost

of predicting a tuple (2D semantic segmentation, 3D semantic segmentation). We use a weighted average of individual losses:

$$\mathcal{L}(\mathbf{y}^i, \mathbf{z}^j) = \alpha\ell(\mathbf{y}^{gt}, \mathbf{y}^i) + (1-\alpha)\ell(\mathbf{z}^{gt}, \mathbf{z}^j) \tag{3.3}$$

The measure of accuracy used for both 2D and 3D semantic segmentation is mean per-class recall. In our experiments, we report results with such a convex combination of module loss functions (for different values of α).

3.3 Experiments

We now describe the setup of our experiment, provide implementation details of the 2 modules, and describe the consistency features.

Baseline: INDEP. We compare our proposed approach (MEDIATOR) to independently predicted highest-scoring solution from each module. Since our hypothesis lists are generated by greedy M-Best algorithms, this corresponds to predicting the $(\mathbf{y}^1, \mathbf{z}^1)$ tuple. This comparison establishes the importance of joint reasoning.

Ablative Study: Ours-CASCADE: We study the importance of multiple hypotheses. For each module (say $\mathbf{y}$), we feed the single-best output of the other module $\mathbf{z}^1$ as input. Each module learns its own weight $\mathbf{w}$ using *exactly the same* consistency features and learning algorithm as MEDIATOR and predicts one of the plausible hypotheses $\hat{\mathbf{y}}^{\text{CASCADE}} = \text{argmax}_{\mathbf{y}\in\mathbf{Y}}\, \mathbf{w}^\intercal \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}, \mathbf{z}^1)$. This ablation of our system is similar to [31] and helps us in disentangling the benefits of multiple hypothesis and joint reasoning.

Finally, we note that both INDEP and Ours-CASCADE can be viewed as special cases of MEDIATOR. Let MEDIATOR-$(M_\mathbf{y}, M_\mathbf{z})$ denote our approach run with $M_\mathbf{y}$ hypotheses for the first module and $M_\mathbf{z}$ for the second. Then INDEP corresponds to MEDIATOR-

$(1, 1)$ and CASCADE corresponds to MEDIATOR-$(M_{\mathbf{y}}, 1)$ + MEDIATOR-$(1, M_{\mathbf{z}})$. To get an upper-bound on our approach, we report `oracle`, the accuracy of the most accurate tuple in $M_{\mathbf{y}} \times M_{\mathbf{z}}$ tuples. We report unweighted mean accuracies of the two modules.

Urban Scene Understanding

Here we provide details and results of our urban scene understanding experiment.

Problem. The goal of this experiment is semantic segmentation of urban street-view scenes.

Dataset. We used the CITY dataset [11], which contains 63 `train` images and 30 `test` images. Nearly half (33/63) of `train` images are monocular, while the rest of `train` and `test` images are from a stereo setup. All images are annotated with 14 semantic categories (road, building, *etc*.), including a void class.

Module 1: 2D semantic segmentation (2D SS) $\mathbf{y}$. We use the Automatic Labeling Environment (ALE) [50, 6].

Module 2: 3D semantic segmentation (3D SS) $\mathbf{z}$. We generated 3D point clouds from disparity maps using the 'ELAS' algorithm [51]. A CRF on supervoxels [7] was constructed on these point clouds. The node potential of the CRF are derived from a logistic regression using spectral and directional features as in [11], RGB values and relative location of points, as well as 'height', which we calculate as a distance to a RANSAC plane fit [52] to the points as the ground plane.

Edge potentials are Potts model with another logistic regression to find the cost of label-disagreement between adjacent supervoxels. Inference is performed using [53].

Solution refinement. Since the output space of the two modules happens to be the same, this experiment presents an opportunity to refine the solutions in a pre-processing step. Every 3D point is paired with its projection onto a 2D point. We train a linear SVM

to choose between the disagreeing 2D and 3D label assignments in the solutions evaluated by the MEDIATOR model. We use the unaries and labels from the `oracle` tuples of the training set to train the SVM, where the `oracle` tuples here are defined as those with the highest accuracy after being refined with a perfect model (*i.e.*, if one module is correct for a particular set of points, and the other is not, the incorrect points are corrected to ground truth). In a leave-one-out process, the solutions of the training data are refined so that the MEDIATOR model can be trained on the refined solutions. This refinement step allows the model to learn when to trust the 2D module over the 3D module and *vice versa*.

MEDIATOR and consistency features. We train the MEDIATOR using both module features and combination features.

- **Module features (80-dim)**: We use histograms for the labels of the solutions within each module, absolute differences in histograms between the 2D and 3D modules, and the energies output from each CRF.
- **Combination features (364-dim)**: We find the confusion matrix between the labels of each diverse solution and the solution prior to being combined with the other module during the solution refinement step.

Results. To evaluate both modules, we use the mean per-class recall. Table 3.1 shows our results. Using cross-val, we chose $(M_{\mathbf{y}} = 15, M_{\mathbf{z}} = 7)$. Our cross-val matrices to choose $M_{\mathbf{y}}$ and $M_{\mathbf{z}}$ are shown in Figure 3.3.

Figure 3.4 shows our results as we vary α, which controls the weighting on our loss function. $\alpha = 0$ is the case where only 3D semantic segmentation accuracy is used to train the MEDIATOR model and $\alpha = 1$ is the case where only the 2D semantic segmentation accuracy is used. The trends of these plots are more difficult to analyze than other experiments that use this approach [13, 19], because of the solution refinement step.

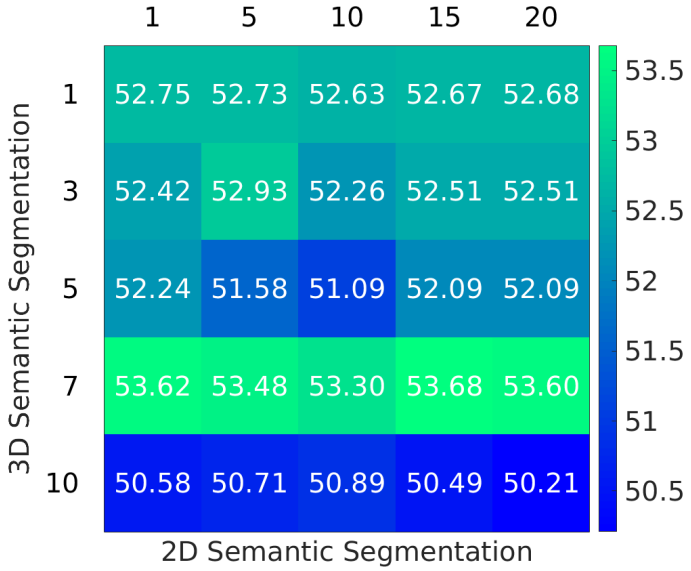


Figure 3.3: Cross validation matrix for different values of $M_{\mathbf{y}}$ and $M_{\mathbf{z}}$. Using this matrix, we choose $M_{\mathbf{y}} = 15, M_{\mathbf{z}} = 7$ as the number of solutions for each module (2D and 3D semantic segmentation).

Table 3.1: Results on the CITY dataset.

	2D SS Acc.	3D SS Acc.	Average
Floros *et al*. [11]	54.80	-	-
3D SS	-	32.07	-
Average	-	-	43.435
Ours CASCADE	**55.65**	57.16	56.405
Ours MEDIATOR	**55.65**	**57.98**	**56.815**
`oracle`	57.82	61.15	59.485

Figure 3.5 shows some qualitative examples for both the 2D and 3D modules. The top row for each diverse set of hypotheses show the 2D module's hypotheses, and the bottom row shows the 3D module's hypotheses.

Discussion. Our proposed approach outperforms the INDEP baseline in both modules demonstrating efficacy of joint reasoning. We are unable to compare to the full approach of [11], as they use temporal information in their model and do not report results for a comparable setting to our approach.

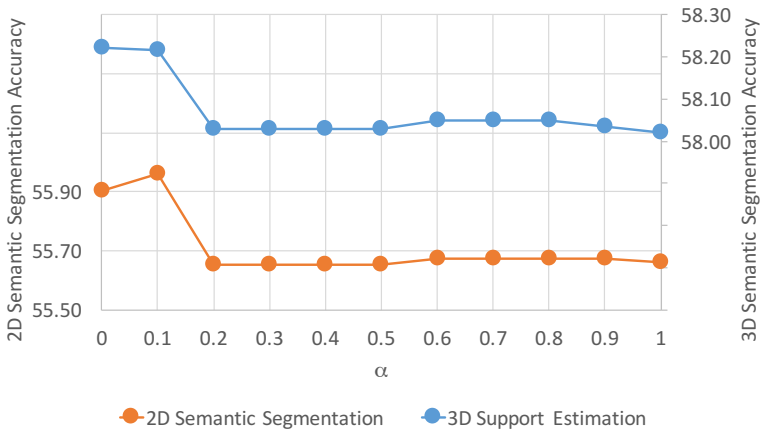


Figure 3.4: Mean per-class recall scores for different values of α in our urban scene understanding experiment, where α is used to weight the different modules. When α is 0, our approach only considers the accuracy of the 3D module, and when α is 1 it only considers the accuracy of the 2D module.

3.4 Discussions and Conclusion

We presented a holistic approach to scene understanding that integrates beliefs across multiple modules to pick the best tuple of a diverse set of hypotheses. In an urban scene understanding experiment, we demonstrate that our holistic approach significantly outperforms independent prediction on each module, which demonstrates a need for information exchange between the modules, as well as a need for a diverse set of hypotheses to reason about. Furthermore, we see from the oracle accuracies that even larger gains are possible. This approach has also been shown to work for language and vision experiments [13], and simultaneous RGBD semantic segmentation and 3D object support estimation [19].

What is ambiguity? Ambiguities exist in different modules for different reasons. In vision, ambiguities arise due to limitations in our models (or *Approximation*) – we do not have models accuracte enough to predict correctly. In other applications, such as sentence parsing, ambiguities arise due to *Bayes error* – there is not enough information available in the input sentence $\mathbf{x}$ to know which interpretation $\mathbf{y}$ is correct [13]. Our approach

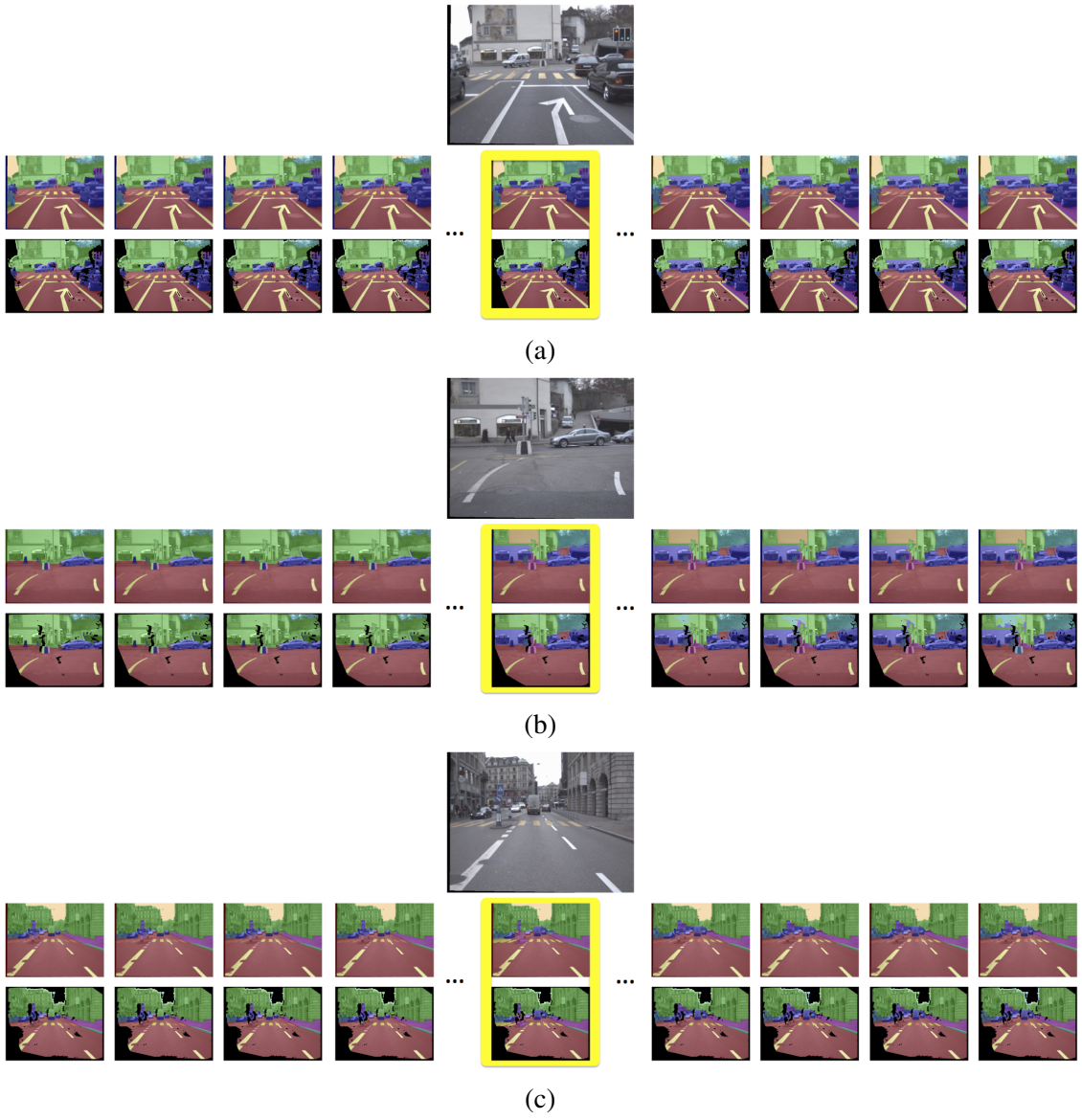


Figure 3.5: This figure shows qualitative examples of both the 2D and 3D semantic segmentation modules. In each sub-figure, the top row shows diverse segmentations for the 2D module, and the bottom row shows diverse segmentations for the 3D module. Our approach uses 15 diverse solutions for the 2D module and 7 diverse solutions for the 3D module, which was chosen by a cross-validation procedure. The highlighted pairs of solutions show the solutions picked by the MEDIATOR model.

naturally handles both kinds, but our results suggest that resolving the latter kind is easier.

Future Work: Our approach needs access to a single dataset with multiple modules annotated. This can be fixed with latent-variable MEDIATOR models, where each dataset has only one module as observed variable and other modules are treated as latent variables. Such a formulation will allow us to leverage existing disparate datasets with different kinds of annotations in each. We are working on this generalization.

Chapter 4

Scene Understanding for Radiation Search Operations

Introduction

Autonomously searching for hazardous radiation sources requires the ability of the aerial and ground systems to understand the scene they are scouting. In this chapter, we present systems, algorithms, and experiments to perform radiation search using unmanned aerial vehicles (UAV) and unmanned ground vehicles (UGV) by employing semantic scene segmentation. The aerial data is used to identify radiological points of interest, generate an orthophoto along with a digital elevation model (DEM) of the scene, and perform semantic segmentation to assign a category (*e.g.* road, grass) to each pixel in the orthophoto. We perform semantic segmentation by training a model on a dataset of images we collected and annotated, using the model to perform inference on images of the test area unseen to the model, and then refining the results with the DEM to better reason about category predictions at each pixel. We then use all of these outputs to plan a path for a UGV car-

rying a LiDAR to map the environment and avoid obstacles not present during the flight, and a radiation detector to collect more precise radiation measurements from the ground. Results of the analysis for each scenario tested favorably. We also note that our approach is general and has the potential to work for a variety of different sensing tasks.

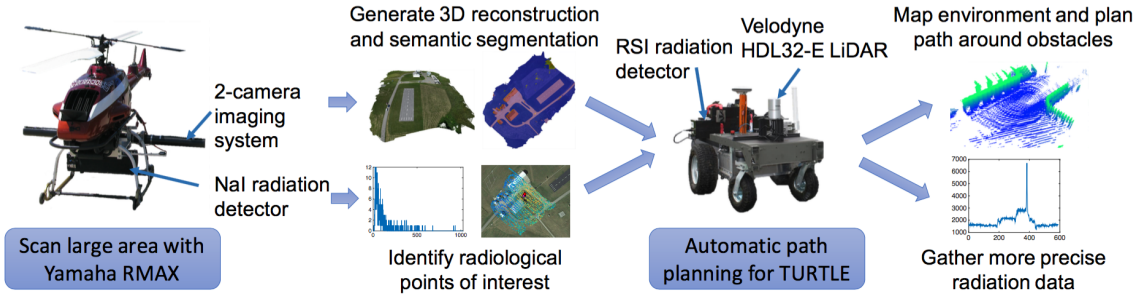


Figure 4.1: Overview of our approach to the autonomous search for radiation sources in an unknown environment. The Yamaha RMAX is used to autonomously search a large area for radiation activity by collecting gamma radiation data. By simultaneously collecting 2D color imagery, a 2D orthophoto and DEM can be generated for the area, which are then used to perform semantic segmentation. Using all image data outputs, a path can then be planned for a UGV, named the TURTLE, to collect more precise measurements around the point of interest. Since objects that were not present during the time of the flight may appear, LiDAR is used on-board the TURTLE to detect obstacles, which are then used to update a global map and find an alternate route.

Searches for illicit radiological or nuclear material, that would comprise a radiological dispersal device (RDD) or improvised nuclear device (IND), are becoming increasingly routine as commercial-off-the-shelf (COTS) radiation detection equipment has become more widely distributed among local, state and federal law enforcement and emergency response agencies. This increased capability comes at the cost of the time and personnel that must be allocated to the radiation/nuke search mission. Therefore, expediting the search process becomes paramount. The search process in general consists of the detection of anomalies, the localization of these anomalies and the identification of the sources of the anomalies (in this case radionuclides). Although the radiation data collected in this work can readily provide an unambiguous radionuclide identification, automated spec-

troscopic identification is not the subject of our research and has been well-studied elsewhere [54, 55]. Instead, we focus on detecting the anamolies using autonomous ground and aerial robots.

Autonomously searching for hazardous radiation sources provides a safer approach to what is possible via manned surveys. It can also be more efficient since a UAV is capable of autonomously scanning large areas to collect radiation data. Furthermore, existing maps for the area of interest may not be available or out-of-date. By taking images from a UAV it is possible to generate an updated 3D map of the area. Machine learning methods can be used to provide a semantic understanding of the scene that can be used to plan a path for a UGV to reach radiological points of interest. Once at the destination the UGV can then collect additional radiation data, transmit video to operators at a remote base station, and update the understanding of other unmanned systems simultaneously searching the area.

Performing this autonomous search in unknown environments is a challenging task. In our approach to the problem we use a UAV and UGV to carry out the search missions. We use a Yamaha RMAX unmanned helicopter with an imaging system that takes 2D color images synchronized with GPS, and a Sodium Iodide (NaI) radiation detector, designed and built by Sandia National Laboratories, to collect gamma radiation spectral data. The imagery collected from the RMAX is used to generate a 3D point cloud that can be processed into an orthophoto and DEM. By performing semantic segmentation on this data to assign each pixel in the orthophoto with a semantic category, more intelligent reasoning can be used to plan a path for a UGV to visit the points of interest. The spectral data is analyzed to output these points of interest where sources are possibly located.

While aerial scans are often capable of providing precise locations of radiation sources with high confidence of a source being present, this is not always the case. The scan

lines in flight paths may not be dense enough for precise location estimates. Also, these location estimates may be at positions where no significant source of radiation exists. We therefore use a UGV (the TURTLE), designed and built by The Center for Dynamic Systems Modeling and Control (DySMAC) at Virginia Tech, to visit the points of interest on the ground. The TURTLE is equipped with a LiDAR, which generates a 3D point cloud to map the environment and detects and avoids obstacles while en route to the estimated source location to collect additional measurements from a radiation detector mounted on-board. We do not perform an active search of the area from the RMAX, which was done in [56], since we consider the scenario where the UAV may be scanning a much larger area, and relies on one or more UGV to more closely inspect the scene.

A lot of work has focused on the task of generating maps of an area from aerial imagery. There are several ways to to accomplish this task. One approach is to perform image stitching, where images are mosaicked together using feature matches to create a 2D image of the scene. Each pixel in the resulting map can then be georegistered if needed. While image stitching is fast and has many implementations available [57, 58, 59], this does not provide 3D information, which is important to perform more accurate semantic segmentation and to plan better paths for UGV. Stereo vision provides another solution to this problem, where a calibrated two-camera imaging rig can be used to generate fast local 3D reconstructions from pairs of images. By reasoning about matching feature points in subsequent pairs of images, these local 3D reconstructions can be transformed into a global coordinate frame to create a full map of the area. There are several publicly available implementations for simultaneous localization and mapping (SLAM) from stereo vision [60, 61, 62], but these implementations typically require high frame rates to work well. The calibrated two-camera imaging rig that we use in our experiments has two low-cost point-and-shoot cameras that are not capable of high frame rates. While we

are actively exploring ways of generating high-quality 3D reconstructions from the stereo pairs collected from our imaging rig, we use single-camera 3D reconstructions that are georegistered to obtain the aerial maps used in our experiments.

Although localizing radiation sources in an unknown environment is the primary motivation behind our work, the focus of this chapter is providing autonomy to the aerial and ground systems collaborating to find them. Figure 4.1 shows an overview of our approach to the autonomous search for sources of radiation. While this work is considered basic research, the end goal that we have in mind is for our system to be able to autonomously identify the locations of potentially hazardous radiation sources with UAV and UGV. The UAV should be able to scan a large area to provide valuable context to a UGV that can efficiently search the area and confirm the presence of sources at the estimated locations. Although not presented in this work, future goals of this project include the ability of one or more UGV searching the area from the ground to scan the other areas in the scene to identify sources located at locations not identified by the UAV. References to work focused on finding sources from aerial data are provided in Chapter 2. The main contribution of this work is a method to autonomously estimate and confirm the locations of radiation sources with UAV and UGV applying scene understanding in an unknown outdoor environment using aerial imagery with a supervised machine learning approach. We also incorporate aerial semantic segmentation results into the A* path planning algorithm so that a UGV will prefer to follow roads over grass and stay clear of obstacles. We also demonstrate the ability to detect obstacles locally on the ground with LiDAR and then find a path around the obstacle using both local and global information.

4.1 Overview of the Method

Our method to autonomously search for hazardous radiation sources in an unknown outdoor environment uses a UAV (Yamaha RMAX) and UGV (TURTLE) to collaboratively understand the scene. We perform two separate missions in two separate adjacent areas of Kentland Farm, Blacksburg, VA, where in the first mission we set up a single radiation source location, and in the second mission we set up two source locations. The RMAX missions are planned by using sets of scan lines. The goal of each mission is to find and confirm the existence and locations of anomalous radiation sources. The TURTLE missions are planned automatically using outputs from the RMAX missions, where the start positions were arbitrarily set to the edge of the map on one of the roads entering the scene.

The RMAX carries an imaging system to take 2D color images and a radiation detector to collect gamma radiation data. We use the images from both missions to create an orthophoto and DEM[1] for the combined flight areas to plan paths for the TURTLE, but treat the radiation data separately for each mission. The orthophoto and DEM are used to perform semantic segmentation. We train a segmentation model on a dataset of images that we annotated with different categories (road, grass, building, vehicle, vegetation, and shadow) at each pixel, where the images in this dataset were taken from low-flying UAV in a variety of environments. This semantic segmentation is used for planning paths for the TURTLE. The NaI radiation detector on-board the RMAX is used to estimate locations of potential sources. The radiation spectral data is output in the form of 1024-d vectors, where the sum of these vectors is called the counts. Stronger sources can typically be found by looking only at the counts, but for weak sources located near stronger sources, more advanced reasoning is typically required. In our experiments, we use the simpler

[1]The orthophoto and DEM are the same size, and when overlaid on one another represent the same part of the scene at each pixel.

approach of using counts. We also note that the max counts value that is found is a global maximum, meaning that only one source per scan be found with this approach.

The spectral data that is output as 1024-dimensional vectors are synchronized with GPS to provide geospatial information about each detector reading. The source locations in each mission are estimated from the aerial data by the GPS position associated with the maximum counts (sum of the 1024-d vectors). To confirm that the radiological points of interest from the aerial data actually contain a potentially hazardous source of radiation, we use the estimated source locations in the discrete set of aerial measurements as destinations for the TURTLE to visit in each mission. For the TURTLE to visit these points, we use the orthophoto, DEM, and segmentation to intelligently plan a path that prefers roads and keeps a safe distance from obstacles. An RSI 701 radiation detector[2] is mounted to the TURTLE to collect additional measurements around the estimated location. Since the scene may change between the end of the flight and the beginning of the ground operation, the TURTLE is equipped with LiDAR to identify obstacles and send coordinates bounding the obstacle to the global path planner to find an alternate route to the destination. LiDAR scans are also used to build a global map of the scene. Figure 4.1 provides an overview of the aerial and ground operations to perform the search.

We provide details for each step of the method in the following sections, which are organized as follows: Section 4.2 provides details of the Yamaha RMAX, the TURTLE, and their hardware. Details of the image-based scene understanding, including the 3D reconstruction and semantic segmentation of the aerial imagery, are presented in Section 4.3. Section 4.4 discusses the path planning for the TURTLE to visit points of interest and how the semantic segmentation is incorporated. In Section 5.3 we present our experiments for both the RMAX and TURTLE missions. Finally, our thoughts on the experiments and

[2]The RSI 701 is a different radiation detector to the NaI radiation detector mounted on the RMAX.

potential future work are presented in Section 4.6.

4.2 Unmanned Systems

In this section we detail the unmanned systems used to complete all of the experiments presented in this chapter.

4.2.1 Unmanned Aerial Vehicle – Yamaha RMAX

The UAV used is a 2005 Yamaha RMAX (model: L17-2), an aircraft originally developed for crop dusting in Japan. The wePilot autopilot system is used to interface with the flight control system and ground control allowing for autonomous operation. The RMAX has a 94kg gross weight, a max payload capacity of 28kg, and flight endurance time of approximately 45 minutes. The RMAX is shown in Figure 4.2 during one of the missions carrying the radiation detector and imaging system.

Radiation Detector and Imaging Hardware. The radiation detector used to collect radiation spectral data is an NaI scintillation-type detector with a 9in length and 3in diameter. In order to understand the measurements of the detector during the missions, we first take background measurements and measurements with a ^{137}Cs radiation source next to the detector for 10 minutes each. Histograms for each case are shown in Figure 4.3.

The imaging system mounted on the RMAX is a two-camera stereo boom designed and built by the Unmanned Systems Lab at Virginia Tech[3]. Two off-the-shelf Canon A-810 cameras were placed inside a carbon fiber tube resulting in a 1.38m baseline. External power is provided to each camera, eliminating the need to remove the cameras for battery

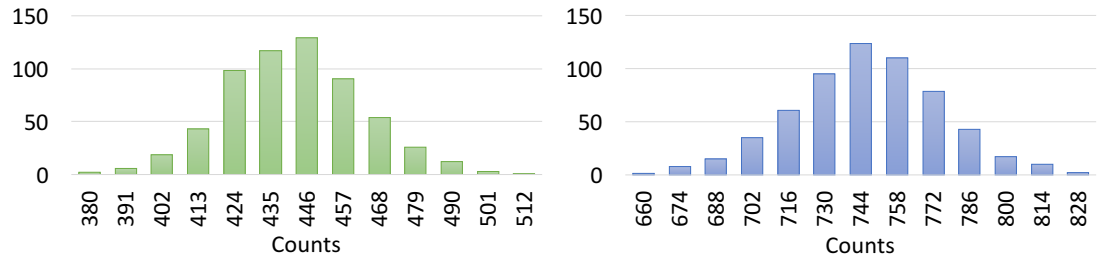
[3]Although we use a two-camera system, the orthophoto and DEM used in our experiments were generated from images from only one of the cameras.

Figure 4.2: The NaI radiation detector and imaging system mounted to the RMAX during one of the missions.

replacement, which would require aligning the cameras and performing a stereo calibration after each replacement. In addition, SD cards are attached with extension cables that allow for quick mounting and dismounting. In order to synchronize the triggering of the cameras, we use a microcontroller that sends pulses over the USB power line and the Stereo Data Maker firmware [104].

4.2.2 Unmanned Ground Vehicle – TURTLE

DySMAC designed and built four identical UGV referred to as the Terrestrial Unmanned Robots for Teamed Learning and Exploration (TURTLEs), one of which is used in our experiments. The control strategy to navigate the waypoints from the global planner was developed in [105]. The base design of the TURTLE includes a differential drive system,



(a) Counts histogram for background with no radiation sources near detector ($\mu = 436.1$, $\sigma = 20.8$, $N = 600$).

(b) Counts histogram with radiation source next to detector ($\mu = 739.7$, $\sigma = 28.5$, $N = 600$).

Figure 4.3: Histograms (not normalized) of the counts from the NaI radiation detector mounted to the RMAX over a period of 10 minutes for (a) background measurements and (b) with a radiation source (^{137}Cs) present for calibration.

powered by two brushless motors located in the rear. Each motor can run continually at speeds up to 10 mph (4.5 m/s) and produce torque up to 322 in-lbs. These specifications, along with four wheel independent suspension, allow for traversal over a wide variety of terrains in both urban and rural environments. Moreover, the vehicle has been tested with payloads up to 100 lbs, a feature which allows for the deployment of the radiation detector and Velodyne HDL32-E LiDAR mounted on-board.

The TURTLE contains an on-board computer, 5 GHz radio, and GPS/INS system. The TURTLE's computer has an i-7 Intel processor, 80 GB SSD, and 8 GB of RAM. This allows for full vehicle control and sensor collection along with building a global DEM in real-time. The radio establishes high bandwidth full inter-vehicle communication, which can broadcast over several miles, facilitating wide scale implementation. This network can easily be augmented to include personnel communication as well. Using this network, processing can easily be distributed on a need-be basis. Moreover, the network allows for clear position knowledge from every other unit, strengthening the estimate. The built in NovAtel SPAN-CPT GPS/INS system is rated up to a position accuracy of 1m. This, without SLAM, was enough to acheive acceptable global LiDAR maps. The TURTLE

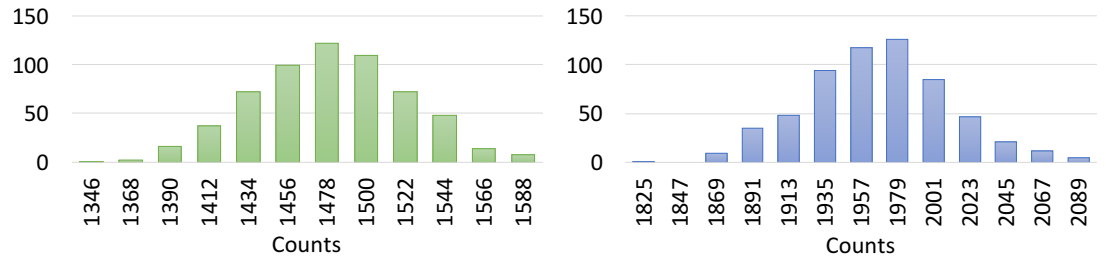
used is shown in Figure 4.4 at the Kentland Farm test area. The computer runs Windows 7, where LabVIEW is used for all control of the robot and processing of the LiDAR data.

There are several reasons to send in a UGV for further inspection. One reason is that the UAV may be performing a scan of a larger area, where scan lines are not that dense. Having a UGV inspect the scene can provide a more precise estimate of the location, as it can get closer to the source. The UGV may also be able to visit areas that are difficult for a UAV to reach. It is also possible to perform long-dwell measurements and to carry larger detectors with higher sensitivity on the TURTLE than the RMAX, which allows for the collection of statistically better data and better localization.

Radiation Detector The radiation detector is a 2x4x16 inch NaI(Tl) system, manufactured by Radiation Solutions, Inc (RSI) that is shown mounted to the back of the TURTLE in Figure 4.4b. Similar to the Sandia system the RSI detector records second-by-second gamma-ray spectra into GPS-tagged, 1024-channel histograms that span an energy range of 0-3000 keV. We performed a calibration of the detector by taking background measurements and measurements with a ^{137}Cs radiation source placed next to the detector for 10 minutes each. A histogram of the counts for each case is shown in Figure 4.5.

(a) (b)

Figure 4.4: (a) The TURTLE at Kentland Farm, Blacksburg, VA where all experiments took place. (b) RSI 701 radiation detector mounted on the back of the TURTLE.



(a) Counts histogram for background with no radiation sources near detector ($\mu = 1469.4$, $\sigma = 42.7$, $N = 600$).

(b) Counts histogram with radiation source next to detector ($\mu = 1956.5$, $\sigma = 43.4$, $N = 600$).

Figure 4.5: Histograms (not normalized) of the counts from the RSI 701 radiation detector mounted to the TURTLE over a period of 10 minutes for (a) background measurements and (b) with a ^{137}Cs radiation source present for calibration.

4.3 Image-based Scene Understanding

Having a 3D reconstruction of the scene is necessary to be able to reliably perform segmentation and plan a path for a UGV. Using only 2D information from the images to plan a path can fail when the segmentation algorithm used confuses the traversable and non-traversable categories it is segmenting. For our aerial operations we chose to use 2D color cameras for perception, as they provide a reliable and low-cost solution compared to LiDAR for generating 3D reconstructions. We were also able to complete all experiments using off-the shelf Canon PowerShot cameras, for which there is no noticeable loss of accuracy in the 3D reconstructions when compared to expensive machine vision cameras previously tested over Kentland Farms in the same flight area. The 2D color images were also proven to be useful for performing semantic segmentation, especially when distinguishing between categories with similar elevation patterns, such as grass and roads.

4.3.1 3D Reconstruction

We considered two different methods of image-based 3D reconstructions in this work. The first method is stereo vision, where 3D positions of pixels matched between the left and right images are calculated using a calibration of the imaging system. Advantages of this approach include fast computation and the ability to track dynamic parts of the scene in 3D. The second approach is using structure from motion, where a 3D point cloud is generated by reasoning about pixels matched between two or more images. This results in a more accurate 3D reconstruction than with stereo vision because the depth resolution is increased by viewing most of the points from more than two camera positions. However, structure from motion is usually much slower than stereo vision, as this now involves optimizing for the 3D position of each point using the pixel positions from all images it is visible within, and also optimizing for the camera positions. Dynamic parts of the scene are also difficult to model with this approach, which typically results in their absence from the final 3D reconstruction. An advantage of structure from motion over stereo vision is that it tends to create an more accurate orthophotos and DEM, which is useful for applications such as path planning, which we explore in this chapter. When attempting to stitch local stereo reconstructions together, we found the results much more noisy than what Agisoft output, which were significantly cleaner and more accurate. This allows for better obstacle detection, which is used to segment the scene. A natural drawback of structure from motion, however, is the inherent scale ambiguity associated with a monocular setup in the absence of GPS. In GPS-denied areas it is better to use stereo vision, as it is capable of providing 3D reconstructions with known scale.

For structure from motion we tested two different implementations. The first implementation tested was VisualSFM [1, 2], which we combined with a multi-view stereo implementation, PMVS [3] to generate a dense 3D reconstruction after initializing itself

with the sparse reconstruction output by VisualSFM. We also tested the professional edition of Agisoft [106]. Of the two, Agisoft provided superior results out-of-the-box, and has the additional capability of generating orthophotos and DEMs, which are more convenient inputs to path planning algorithms.

In this chapter, we collected stereo images, but images from only one camera were used to generate 3D reconstructions from SfM. In future work, stereo vision could potentially be used for real-time reasoning about possible radiation source locations. To generate a stereo reconstruction, we first undistort and rectify the images of the left and right cameras using a calibration file generated using the MATLAB Calibration Toolbox [107]. To calculate the disparity map for the image pair, we then use the semi-global block matching algorithm provided by OpenCV [108]. Figure 4.6 shows the process of generating a 3D point cloud from a pair of stereo images taken from the Canon A-810 cameras during one of the RMAX missions. The disparity map is first calculated, after which the calibration data is used to generate 3D points for each pixel with a valid disparity value as

$$w \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \mathbf{Q} \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -C_x \\ 0 & 1 & 0 & -C_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -\frac{1}{B} & \frac{(C_x - C'_x)}{B} \end{bmatrix} \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} \tag{4.1}$$

where (X_C, Y_C, Z_C) is the 3D point with units in meters, (x,y) is the pixel position, and d is the disparity value. The back projection ($\mathbf{Q}$) matrix is generated using the calibration and is used to transform the local pixel coordinates and disparity value to a 3D point. In this matrix, (C_x, C_y) is the camera center, f is the focal length, and B is the baseline distance between the cameras, where all parameters are from the left camera except for

C'_x [108]. It was observed that cleaner reconstructions were possible by convolving the depth image with a median filter. We found the results from local stereo reconstructions to be impressive given that these are off-the-shelf cameras with retractable lenses taking images while mounted to the RMAX, which causes a lot of vibration.

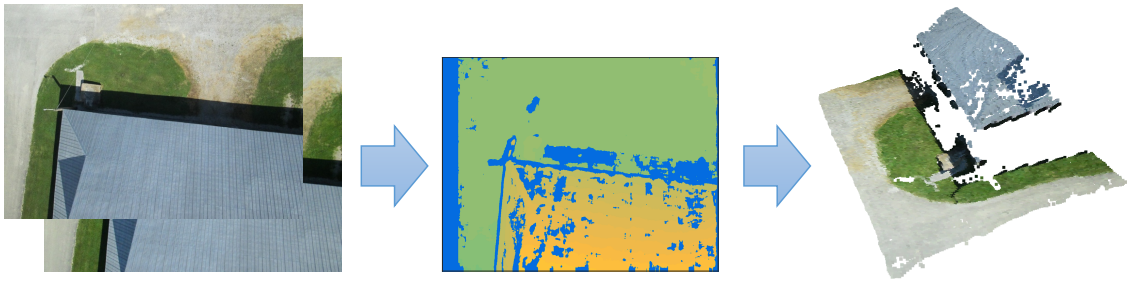


Figure 4.6: Example stereo vision 3D reconstruction, with example image pair (left), disparity map calculated using example images (middle), and the 3D reconstruction generated from the disparity map and calibration file (right).

We chose to use Agisoft in this work, which typically takes multiple days of computation to reconstruct the an area similar in size of Kentland Farm. The orthophoto and DEM used in our work help to demonstrate the capabilities of our system. However, our approach does not rely on Agisoft, and our work does not intend to improve upon existing reconstruction methods. If more expensive machine vision cameras are used than the point-and-shoot cameras used in our work, then a real-time 3D reconstruction of the area can be generated using existing code [60, 109, 110]. If real-time 3D reconstructions are provided, then a real-time response by our system would be possible.

We also started developing another method of image-based 3D reconstruction. With our current imaging rig, containing Canon A-810 cameras, the fastest frame rate (approximately 1 Hz) was found in tests to not work well with existing methods of stereo vision and visual odometry. However, we wish to use the calibration of the imaging rig to develop faster and more accurate 3D reconstructions than what is possible with a monocular setup. To accomplish this, we first generate SIFT [8] feature points in all of the left and

right image pairs. These features are matched between all left and right image pairs, where good matches are kept and backprojected into 3D using the stereo calibration. Outliers are then removed in each local 3D reconstruction using our own implementation for outlier removal similar to the statistical outlier removal package from the PCL [7]. After, SIFT features are matched in image pairs taken nearby, which are found using GPS. Clustering other high-level image descriptors could possibly be used instead of GPS, depending on how much the scene's appearance varies.

Instead of processing the images in order of when they were taken, we follow a similar approach to Bundler [111]. We batch process all pairs of images (left, right) incrementally and build the reconstruction in an order we decide. We initialize the reconstruction with the pair of images containing the most matches (after the outlier rejection step). From there we incrementally build the reconstruction by adding points generated by neighboring image pairs with the most matches. To add these new 3D points and view positions to the global coordinate frame containing the current 3D reconstruction of the scene, we find transformations between the matching points of the local 3D points being added to the global reconstruction, and the points already registered in the reconstruction. To find this transformation, we find rotations and translations using singular value decomposition inside of a RANSAC loop, where we choose the rotation and translation corresponding to the ones that yielded the highest number of inliers. We implement a priority queue to process pairs of different view points, where the priority is based on the number of SIFT matches between the pairs. By indexing unique points, we can perform Bundle adjustments, using SSBA [112], to minimize the reprojection error of the 3D points in the reconstruction into their corresponding 2D points in the images in which they are visible. A preliminary result is shown in Figure 4.7.

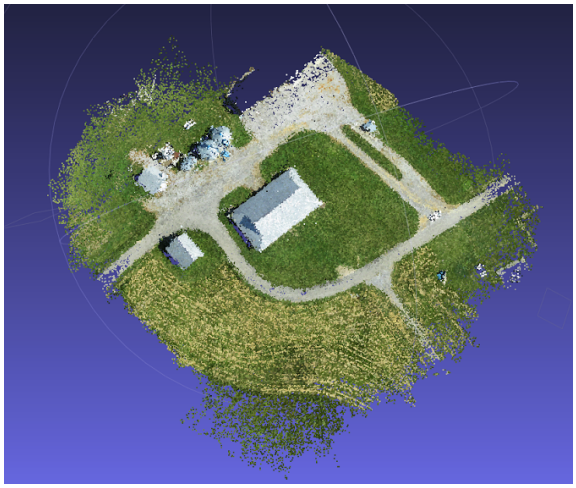


Figure 4.7: Result (preliminary) of reconstructing a scene using stereo vision. Left and right image pairs are incrementally added to a global reconstruction, where we initialize the reconstruction with the left/right image pair containing the most SIFT matches. In an incremental process, points from other image pairs are taken from a priority queue and added to the reconstruction. Priorities are are based on the highest number of matches the image pair has with other points already registered in the global reconstruction.

4.3.2 Aerial Dataset and Semantic Segmentation

Some researchers performing segmentation of aerial imagery use unsupervised approaches, where no training data is used to help make predictions [89, 113]. These often fail due to assumptions of good initializations that subsequent segmentations rely on (*e.g.* operator labeling the road in the first image of a road-tracking UAV) and arbitrary hand-crafted parameters (*e.g.* RGB thresholds for classification) that do not extend to other scenes. For this reason, we decided to take a supervised approach to the problem, where we train a segmentation model to predict one of several semantic categories for each pixel in an image. This approach requires no initializations of any kind, and is also scalable, since no algorithm changes are required when testing on a different type of scene. In the case the approach does fail, then it is likely that there is a simple need for more training data.

For the work presented in this chapter, we annotated a collection of images taken from low-flying UAV in a variety of environments with several semantic categories to

be able to train the segmentation model that predicts these categories on the unseen test images of Kentland Farms. The images were annotated using LabelMe [4]. Figure 4.8 shows an example annotation from our dataset and the legend for the colors for each category. The full dataset consists of 230 annotated images, where 54 come from tiles of the orthophoto for the Kentland Farms flight, 119 come from an RMAX flight conducted by the Unmanned Systems Lab at Virginia Tech in Fort Indiantown Gap, PA, and 57 come from a variety of flights taken from low-flying UAV. However, for training, we only use a subset of the 119 Fort Indiantown Gap images to prevent the model from overfitting to this scene. We use 15 images from this part of the dataset, resulting in a total of 72 training images when testing on the orthophoto of the Kentland Farms imagery. Ideally we would collect a very large dataset with more semantic categories so that a deep semantic segmentation model, such as DeepLab-CRF [10], could be used. However, collecting such a dataset is difficult, since a diverse set of images from low-flying UAV are not easy to find, and the annotation procedure is exensive.

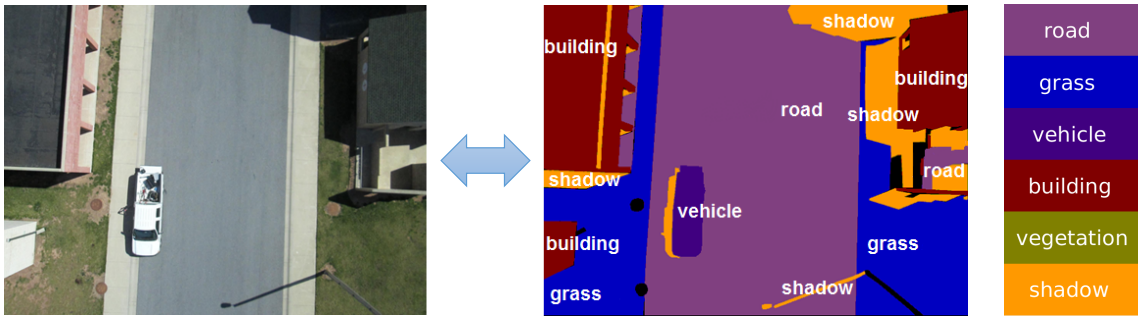


Figure 4.8: We annotate 2D RGB images taken from low-flying UAV with 6 different semantic categories that we use to train a model to predict the categories for pixels of unseen test images.

To perform semantic segmentation we use the Automatic Labeling Environment (ALE) [5], which trains a model using 2D images and annotations and then uses that model to perform inference at the pixel-level on images in a test set unseen to the model. The traversable

categories are road and grass, and the rest are the non-traversable categories. While shadows often contain traversable regions, we treat them as obstacles. It is possible to postpone analysis of those areas until a UGV enters the scene with LiDAR to analyze whether or not they are traversable, but we do not do this. We identify obstacles in the DEM by calculating the gradient magnitude and filling regions surrounded by larger gradients. Pixels within these regions that contain traversable category labels are assigned the mode of the most likely non-traversable categories within the region using the unaries computed by TextonBoost [114] in the ALE framework. An overview of our approach to performing semantic segmentation is shown in Figure 4.9.

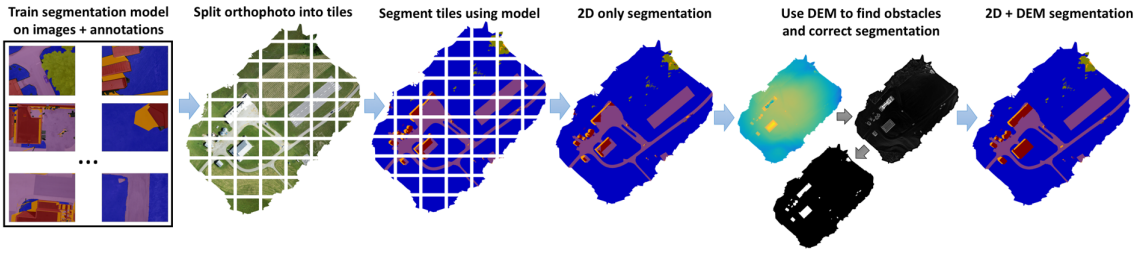


Figure 4.9: Overview of our approach to performing semantic segmentation of the aerial imagery. We first train a segmentation model on a dataset of images taken from low-flying UAV and their annotations. We then take the orthophoto and divide it into tiles so that each can be segmented individually. These segmentations are then combined to make the 2D only segmentation result. To improve the segmentation results the DEM is then used to make updates. Regions surrounded by larger gradients are identified after which any pixels within those regions classified as traversable categories are assigned the mode of the most likely non-traversable categories within those regions.

We note that we make corrections to regions of the segmentation where an obstacle has been detected and a traversable category has been classified, but do not make corrections to regions where no obstacle has been detected. We do this for two reasons: 1) grass and road are segmented with high precision, as evidenced by our results, and 2) there may be some obstacles that are not detected with the DEM.

4.3.3 Other Approach

We choose to use Agisoft and the Automatic Labeling Environment based on observations made in previous experiments [115, 116]. We have experimented with open source code such as Bundler [111], VisualSFM [117, 118], and PMVS [3] (used in combination with Bundler and VisualSFM) for image-based 3D reconstructions. Prior to using our current stereo imaging rig, containing the Canon A-810 cameras, we used more expensive machine vision cameras (Kappa Zelos-655) mounted inside a carbon fiber tube with a 1.53m baseline. The rig, shown in Figure 4.10, also contained an IMU/GPS system, and a FitPC2i for data collection. In one experiment, we used the images captured from this imaging rig to generate a 3D reconstruction of the scene using Bundler and PMVS. The result is shown in Figure 4.12a.

All images, from both the left and right cameras, were used to reconstruct the scene. The output of Bundler and PMVS is a point cloud with ambiguous scale. To scale the point cloud to an interpretable size (meters), we identify each pair of camera positions (left, right) and compute the relative baseline by taking the median value of all of the relative baselines (distances between the left and right camera positions in the 3D reconstruction) between the left and right camera positions. We then compute the scale by dividing the known baseline of the imaging rig (1.53m) by the relative baseline. In addition to scaling the point cloud, we rotate it so that the ground plane roughly aligns with the xy plane so that we can measure the 3D points. We find a plane that fits the points well with RANSAC using the code of [52]. Using the normal vector of the plane (v_x) and two orthogonal vectors (v_y and v_z), we construct a 3×3 rotation matrix ($R = \begin{bmatrix} v_{x,1} & v_{x,2} & v_{x,3} \\ v_{y,1} & v_{y,2} & v_{y,3} \\ v_{z,1} & v_{z,2} & v_{z,3} \end{bmatrix}^{-1}$) that is used to rotate the points to the xy plane.

In addition to using Bundler+PMVS, we have tested VisualSFM+PMVS. While the results remain roughly the same, we have noticed that the VisualSFM completes the re-

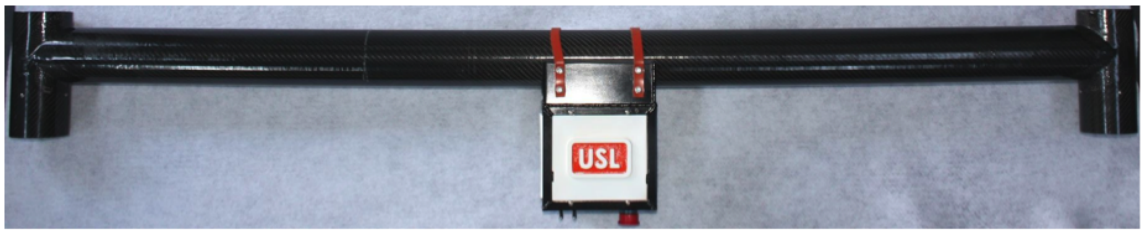


Figure 4.10: Stereo imaging rig made of carbon fiber, which contains 2 Kappa Zelos-655 cameras, and IMU/GPS system, and a FitPC2i for data collection. The baseline of the imaging rig is 1.53m.

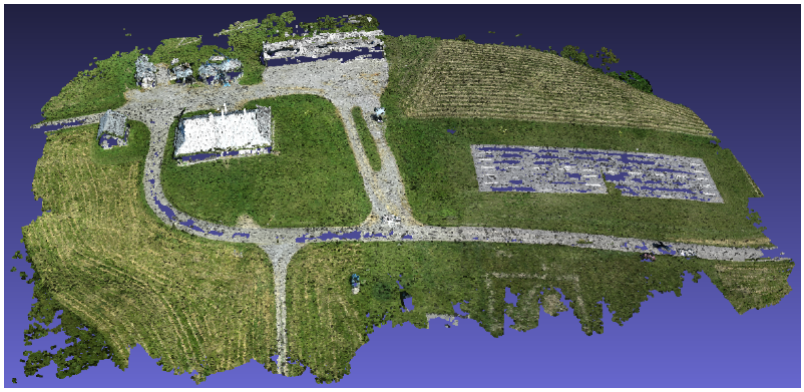


Figure 4.11: Image-based 3D reconstruction of Kentland Farm, VA with VisualSFM [1, 2] and PMVS [3].

construction process faster. The reconstruction of Kentland Farm, VA is shown in Figure 4.11.

We then performed semantic segmentation of the point cloud data. The ground truth labels for the point cloud we are trying to segment are shown in Figure 4.12b. Prior to performing segmentation, the statistical outlier package of the Point Cloud Library (PCL) [7] is used to remove the noisy points output by Bundler [111].

To train a semantic segmentation model, we use an annotated LiDAR dataset collected by [119], which contains 1.6 million 3D points with 44 labels. We group and eliminate categories of this dataset so that the categories match the ground truth categories of our test set (Kentland Farm point cloud). The labels for our dataset are pole, ground, building, and vehicle.

For efficiency, semantic segmentation is performed on supervoxels instead of individual points. These supervoxels are generated using k-means clustering on the 3D positions of the points. An illustration of this is shown in Figure 4.12c, where each cluster is assigned a random color. These supervoxels are calculated for both the train set and the test set. Features are calculated for all of the supervoxels in the train and test set. These features, also used in other works [11], include point-ness (λ_0), linear-ness ($\lambda_0 - \lambda_1$), and surface-ness ($\lambda_1 - \lambda_2$), where λ_i is an eigenvalue calculated by performing eigenvalue decomposition on the covariance matrix of the 3D points. These features help to describe the shapes of the local cluster of 3D points. For example, a cluster of points from a pole will have a large λ_0 and small λ_1 and λ_2. We also use the mean height of the points in each supervoxel, the bouding box dimensions, as well as the mean and variance of the distances between each point and its nearest neighbor. A graphical model is contstructed over the supervoxels, where we reason about the likelihoods of each category for each supervoxel (unary terms), and enforce smoothness between neighboring supervoxels (pairwise terms). We train an SVM with a radial-basis function (RBF) kernel on features for each supervoxel in the training set. We then calculate the unary terms as the negative log likelihood values of the probability estimates output by LIBSVM [120]. For the pairwise terms, we use the differences between the 3D features of the supervoxels, as well as the mean RGB values of the points in each supervoxel. To perform inference, we use GCMex [121].

We compare three difference approaches:

- SVM. In this approach, we simply classify each supervoxel with the label output by LIBSVM [120].
- 3D-CRF. Here we perform inference using the graphical model, but with only 3D feature differences in the pairwise term.

(a) (b) (c)

Figure 4.12: (a) Image-based 3D reconstruction of Kentland Farm, VA using Bundler and PMVS. (b) Ground truth annotations of Kentland Farm point cloud. (c) Supervoxels used to perform semantic segmentation on the point cloud.

- OURS-CRF. This is our full model, which uses color differences in addition to 3D feature differences.

The average precision and recall values of the approaches are shown in Table 4.1.

Table 4.1: Results of testing on the Kentland Farm point cloud generated by Bundler and PMVS. Three different approaches to semantic segmentation were tested.

	Average Precision	Average Recall
SVM	78.53	66.14
3D-CRF	78.76	88.50
OURS-CRF	**79.60**	**89.68**

4.4 Path Planning

4.4.1 UAV Path Planning

For the experiments presented in this chapter, we use equally spaced scan lines for taking images that are used to generate a 3D reconstructions. The process of capturing images and generating 3D reconstructions results in a trade off between the accuracy of the reconstruction, the flight time, and the time required to build the reconstruction with the software. More images results in more accurate point clouds, but longer processing times

for the reconstruction software. Flying closer to the objects in the scene will also result in more accurate point clouds, but also require more images (longer processing times), and longer flight times to cover the entire area. We chose the altitude and scan line width based on the desired flight time and observations made in previous tests.

For radiation surveys, the established preferred method is to perform the data collection with uniform line spacing at constant altitude [122]. The acceptance of the radiation detector, upon which parameters such as the exposure rate coefficient depend, varies with altitude. Therefore the accuracy of any quantities derived from the radiation data benefit from the maintenance of a consistent altitude. Furthermore, the line spacing should be at least close enough so that the acceptance areas under the aircraft overlap slightly lest any resulting interpolation be performed across regions that have not been sampled. Consistent line spacing also greatly simplifies the interpretation of the interpolations by maintaining a uniform contribution to the uncertainty by the altitude and lateral distances in any derived quantities. We also note that variations in detector angles can introduce undesired variations in the spectral data, which is the reason we do not use other flight patterns aiming to optimize time *vs.* coverage, such as the Archimedean spiral.

For applications outside of radiation sensing, there are ways of choosing camera positions that are best for capturing images of the scene with limited time. If we have a 3D LiDAR scan of the scene, for example, then model-based view planning can be performed. By considering sets of scan lines at different altitudes which can be scored based on how well they view the scene, we can use the cost function presented in [122] to pick the best set of scan lines and simultaneously generate a trajectory for the UAV.

A preliminary experiment and method are presented in [122], where we specifically focus on fixed-wing aircraft, since they are more challenging to plan trajectories for. Each view point is scored based on three conditions:

1. Depth resolution. First we check what points are within a depth threshold. The value chosen was based on the depth resolution formula for stereo vision, $\Delta Z = \frac{Z^2}{Bf}\Delta d$, where ΔZ is the depth resolution, Z is the distance from the camera to the point of interest, B is the baseline of the imaging rig, f is the focal length of the camera, and Δd is the smallest possible disparity, which we set to 1 to obtain the max depth resolution given the other parameters. This condition helps remove points that are unlikely to be reconstructed well with stereo vision.
2. Visibility. Points within the depth resolution threshold are tested for visibility (*i.e.* are within the field of view of the camera). To do this we simply calculate 5 planes that define a pyramid and find points that lie on the correct side of all planes (*i.e.* are inside the pyramid). This method is fast and extends to other polyhedrons.
3. Occlusions. Of the points that are within the depth resolution threshold and visible, we then find the points that are not occluded by other points. We use an efficient approach to test for occlusions. We iterate through each point, and define a cuboid that extends from the view point to the point of the current iteration in the visible set. If any points lie within this cuboid, then we say that the current point of interest is not visible.

The score for each view point is the number of visible points as defined by the three conditions above. The score of a particular scan line is the sum of the visible points at each view point on the scan line.

Once we have the set of scan lines we want to visit, we can then try to minimize the flight time by formulating the problem of how to visit each scan line as a traveling salesman problem (TSP). With n scan lines, there are $n!$ possible paths. We use the aircraft cost functions of [122] to find the cost of transitioning between the end of one scan line, and the start of the other. The problem is formulated as a mixed integer linear program to

optimize the following:

$$\min_{x} \; c^{\mathrm{T}} x \tag{4.2}$$

$$\textit{s.t.} \quad Ax = 2 \tag{4.3}$$

$$x_i = 1 \quad i \in E \tag{4.4}$$

where x is an indicator vector to represent whether or not there is an edge between two nodes, where the nodes are the endpoints of the scan lines ad an auxiliary node to define start and end positions of the path. c represents the cost of adding an edge between two nodes. There is no cost for transitioning between any node and the auxiliary node. If a pair of nodes lie on the same row of endpoints for the scan lines, including endpoints of different altitudes, then adding an edge between these nodes is assigned a cost based on the aircraft dynamics. Since the transitions between scan lines are symmetric, a lookup table is formed for possible transition costs to eliminate redundant calculations during the optimization. For pairs of nodes that are not endpoints of the same scan line and do not lie on the same row of endpoints, there is a very high cost. The first constraint ensures that every node is connected to exactly two others. The second constraint ensures that nodes that are endpoints of the same scan line have an edge between them, forcing the path to travel along all scan lines.

We solve the above TSP with the Gurobi solver [123]. As noted in previous work, this can result in sub-tours, where there is no continuous path between all of the nodes, or in this case no continuous path from the start node to the end node. We use the approach detailed in the MathWorks documentation for the TSP to iteratively remove sub-tours. We note that this approach does not take into account the redundancy in the imagery

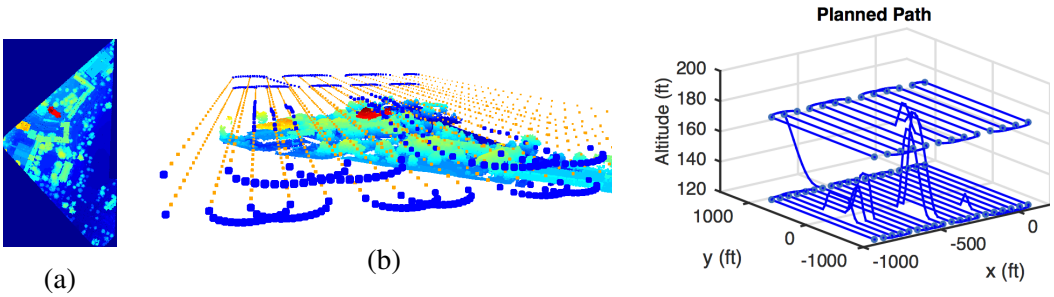
introduced by overlap in the turns between scan lines, as the turn data is typically not used for radiation surveys. However, this would be an interesting study for future work.

Obstacles in the scene also pose a problem. The scan line proposals that are generated include some which pass directly through objects in the scene. During the scoring process, points within objects are relocated above the object using a minimum clearance parameter. The points are identified using a digital elevation model (DEM) generated using the point cloud. However, once we have the path, we now wish to generate a trajectory over the obstacle. We do this by iterating through a set of discrete points leading up to the obstacle, calculating the resulting trajectories and associated costs, and then selecting the trajectory with the lowest cost. Similarly for returning to the scan line from the top of the obstacle, we calculate trajectories by iterating through points on the scan line beyond the obstacle. The trajectory generated to fly over the obstacle is a clamped cubic spline, which first leads to a point above the obstacle, followed by a spline back down to the scan line. By introducing this intermediary point above an obstacle for the aircraft to travel to, the obstacle can be avoided by ensuring the necessary clearance while maintaining a minimal dynamic cost. Some preliminary results of our approach applied to a LiDAR scan[4] of an area in Blacksburg, VA are shown in Figure 4.13.

4.4.2 UGV Path Planning

To plan a path for the UGV to visit points of interest on the ground we consider first how to plan a path between two points given an orthophoto, DEM, and segmentation. Our method of choice was A* [124], which extends Dijkstra's algorithm [125] via a heuristic to assist in finding the best path between the two points. The orthophoto, DEM, and segmentation have the same image dimensions, and we therefore define nodes of our

[4]This scan was provided by the United States Geological Survey (USGS).



(a) (b) (c)

Figure 4.13: (a) A digital elevation model generated from the LiDAR point cloud, which is used to identify obstacles for the scan lines. (b) The output of our path in 3D space, including points of the LiDAR point cloud. The blue spheres on the path show the trajectory calculated using our method between scan lines and over obstacles. The orange spheres show waypoints of the scan lines where images should be taken. (c) A plot of the same path is shown in (b) to help visualize specific transitions and obstacle avoidance taking place. Note axes are not to scale, making some trajectories for fixed wing aircraft appear infeasible.

graph to be the pixel positions with 8-pixel connectivity. The size of the grid paths are calculated on is 458x440 (201,520 total nodes), which translates to each pixel representing an area of approximately 0.6m x 0.6m. Downsampled versions of the orthophoto, DEM, and segmentation[5] were used to make calculating the paths efficient but still accurate. The cost function for A* search is defined as

$$f(n) = g(n) + h(n) \tag{4.5}$$

where $g(n)$ is the cost of making a move between the current node (x_c) and a neighboring node (x_n), and $h(n)$ is the heuristic that estimates the cost of moving from x_c to the goal node (x_g).

Our implementation will find a path between two points in the orthophoto using the semantic segmentation results, where there is a preference that the path chosen follows the roads. We experimented with using the DEM in the cost function, but found it made little

[5]The original dimensions for each of these outputs was 18137x17454.

difference, possibly because obstacles are not added as traversable nodes in the graph. However, for other scenes this cost can easily be included if necessary. The motivation for following roads over grass is that grass tends to be more difficult to traverse for UGV, as well as obstacles and ditches being less visible. To provide further motivation for this design choice, we show the power consumption of the motors when traversing pavement and grass as a function of the percent speed set in Figure 4.14. The power consumption values were calculated by taking the median value over all peaks in the plot over time as the UGV traversed both grass and pavement surfaces. As seen, traversing grass always results in a higher power consumption than when traversing pavement.

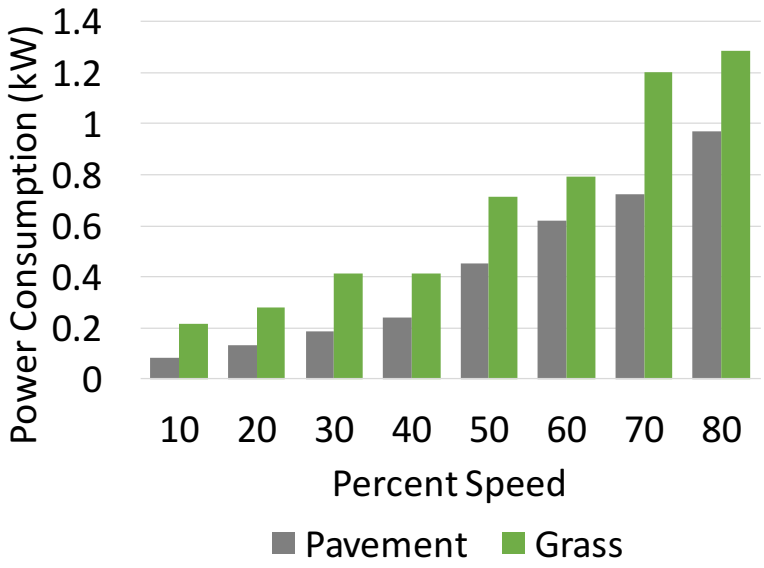


Figure 4.14: This shows the power consumption for the motors of the TURTLE when operating on pavement vs grass for different speed settings. The power consumption measurements were calculated by taking the median value of all the peaks in the plot over time. Significantly less power is consumed on pavement compared to grass.

We calculate the heuristic function $h(x_c)$ as the euclidean distance in pixels to the goal position, and the cost of moving between x_c and x_n as

$$g(n) = \mathbf{w}^{\intercal}[\phi_1(x_c); \phi_2(x_n); \phi_3(x_c)]. \tag{4.6}$$

The specific weights we use are $\mathbf{w} = [5; 2; 5]$. Here we note that prior to the experiments we used different weights, with regular (not inverse) distances for $\phi_1(x_c)$ and $\phi_3(x_c)$, and a value of 1 in $\phi_2(x_c)$ when x_n is classified as road and 0 otherwise. We inverted the formulation so that negative weights were not used and observed *very* similar planned paths to the ones presented in the chapter. However, we have presented the weights and features with the correct formulation. The reason the planned paths may be similar are that they do not navigate around too many obstacles to reach the destination, and are still being rewarded for actions in a very similar way. The following describes each feature ϕ_i:

- $\phi_1(x_c)$ is the inverse of the distance to the nearest x_i not classified as road, which is 0 when the x_c is not classified as road. This rewards the algorithm for staying near the center of the road.
- $\phi_2(x_n)$ is an indicator variable that is 1 when x_n is not classified as road and 0 otherwise. This encourages the algorithm to pick roads over nodes classified as other categories.
- $\phi_3(x_c)$ is the inverse distance to the nearest x_i not classified as road or grass, which is 0 when x_c is not classified as grass. This helps the path stay clear of obstacles in the scene.

The weights (w) used above are up to the system designer to pick for their application. With our current weights, there is a preference to follow roads over grass, but the algorithm

does not minimize the amount of time spent on grass. While we do not use weights based on the values of Figure 4.14, this could be done with more data. This also extends to an arbitrary category size if traversability data can be gathered for each category. We did not use these costs in our experiments, because the choice between grass and roads was based on common sense reasoning that roads are better than grass. However, with other terrain types this might not be so clear, and so these types of plots may be very useful. Power consumption or other characteristics related to different terrain types being classified could be directly incorporated into these weights. These weights could also be actively modified as a UGV navigates a particular scene learning more about the environment.

4.5 Experiments

4.5.1 Experimental Setup

Radiation Sources. The sources used in the experiments are listed in Table 4.2. Sources in these activity ranges are typically used for system checks of laboratory equipment. As such they are relatively weak and sit roughly at the threshold of detection for the detector systems used. The sources were placed in Nalgene bottles and positioned on top of thin steel stands 1m off the ground for the aerial data collection and taped to the bottom of the stands for the ground collection to ensure that there was no attenuation from the stand during the ground-based measurements.

RMAX Missions. Two flights, using the RMAX, were conducted at Kentland Farms, Blacksburg, VA, where different configurations of radiation sources were placed in the scene for each flight. In Mission 1, all of the sources listed in Table 4.2 were placed at a single location. In Mission 2, both Ho sources were placed at one location, and the

Table 4.2: Information for each of the 4 radiation sources used in the experiments. Different combinations of these sources are used when creating each source location.

Nuclide	Half-life (yr)	Activity (μCi)
^{137}Cs	30.2	10.0
^{133}Ba	10.7	16.1
^{166m}Ho	1200.0	138.7
^{166m}Ho	1200.0	147.1

Ba and Cs sources were placed at another location. Source locations in each mission are estimated by the GPS locations associated with the maximum counts (sum of the 1024-d radiation signal from the detector).

For each flight, paths were generated in Mission Planner [126], with an altitude of 30m, and a distance between scan lines of 4m, which was chosen to ensure sufficient overlap in the imagery for generating a high quality 3D reconstruction and also to obtain more dense measurements for the radiation data. The height of 30m allowed testing of the detection system's capabilities given the low activity level of the sources with a significant background signature present. The velocity of the RMAX was set to 3m/s, and images were captured once a second, resulting in around a 90% overlap for subsequent images. A total of 1644 images (874 for Mission 1 and 770 for Mission 2) were collected for both missions, with 3288 images if considering stereo pairs. Image taking is also synchronized to simultaneously log GPS and radiation data. The RMAX is seen flying one of the missions in Figure 4.15. While the altitude was chosen based on observations of the scene to ensure there would be no hazard for the RMAX during the flight, this value could be chosen using a 3D point cloud of the scene. If one is not available, then real-time sense-and-avoid methods could be applied.

Mission 1 lasted approximately 23 minutes, and Mission 2 lasted approximately 26

Figure 4.15: The Yamaha RMAX mid-flight during the first search mission.

minutes. This includes take off, landing, the navigation to and from the start/end waypoints. Agisoft required several days of computation, while segmentations were finished in a matter of seconds per image. We performed the UGV missions at a later date than the flights due to logistics. Moving forward we are exploring methods to obtain 3D reconstructions much faster by using stereo vision so that this is not necessary.

The reason that we did not survey the entire area at once is because, 1) we wanted to conduct multiple missions, and 2) the endurance of the RMAX may not have been sufficient. The RMAX can fly for approximately 45 minutes before needing to refuel. If endurance is needed, then a fixed wing aircraft is better, but may move too quickly to collect statistically meaningful data. There are higher endurance helicopters, but they are much more expensive than the RMAX, which was within our budget. Our requirement to be able to carry over 30 lbs of payload also narrowed the helicopter selection.

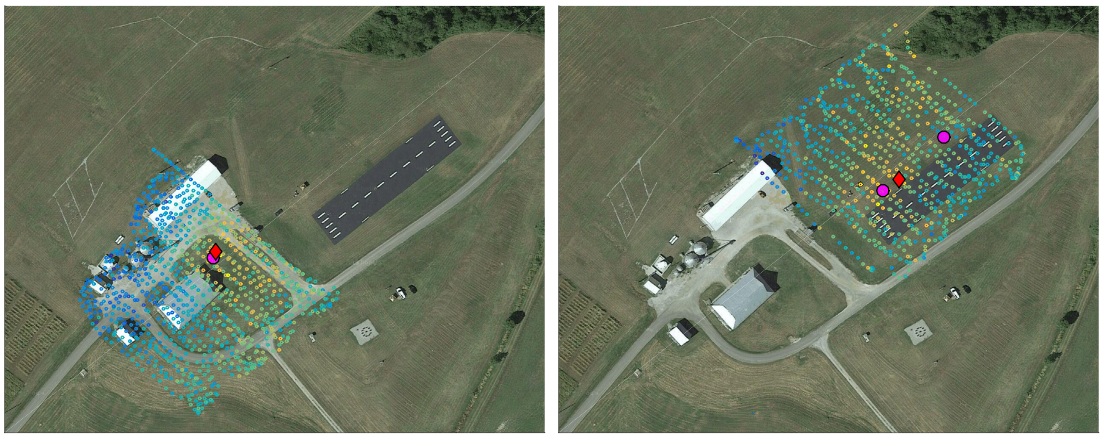
TURTLE Missions. We ran two separate missions for each configuration of radiation sources. The destinations are defined by the position of max counts from the aerial radiation data. The same start position is set for both missions, which is located on one of the roads at the exterior of the scene. Since the scene may change between when the flights take place and when the TURTLE is deployed, we placed an obstacle on the planned path for the TURTLE in both missions so that it was forced to detect the obstacle and then find a path around it by updating the global map and planning an alternative route. The global map is updated by removing nodes in the 2D grid containing the obstacle from the set of traversable nodes.

4.5.2 RMAX Results.

Radiation Results. The position histories for each flight are shown in Figure 4.16, where the color of each point represents the counts value (from blue/low to red/high). The ground truth locations of the sources are shown with magenta circles, and the red diamonds show the positions of max counts set as destinations for the TURTLE. In our results we are successfully able to identify 2 of the 3 source locations of the combined missions. The failure case is the Ba and Cs source combination, as seen in Figure 4.16b as the ground truth position farthest from the max counts estimate. It was found to be too weak to be seen by different nuclear anomaly detection algorithms at the altitude flown by the RMAX (30m). The counts value at the closest reading was 612, and the median of the 10 closest measurements was 617.5, both of which were below the average counts for all of the aerial readings taken during that flight. For reference, the counts for the position closest to the other source location (2 Ho sources) is 654, with a median for the closest 10 points of 658. Therefore the TURTLE is never instructed to visit anywhere near this position unless the starting point is set in such a way that it passes right by it on the way

to the location of the 2 Ho sources, which is much stronger.

While our experiments proved what we set out to prove, the failure case does provide motivation for UGV-based search methods to be applied. The particle filter method presented for aerial search in [56] is one example of an approach that could be applied for ground search operations. Other approaches, such as maximum-likelihood estimation (MLE) and contour following [102], also have potential. This work is part of a fundamental research project, so there are currently no specific end-user requirements for the UAV-UGV teaming side of our work. Future work may include coming up with performance metrics to evaluate the performance of more advanced radiation search tasks.

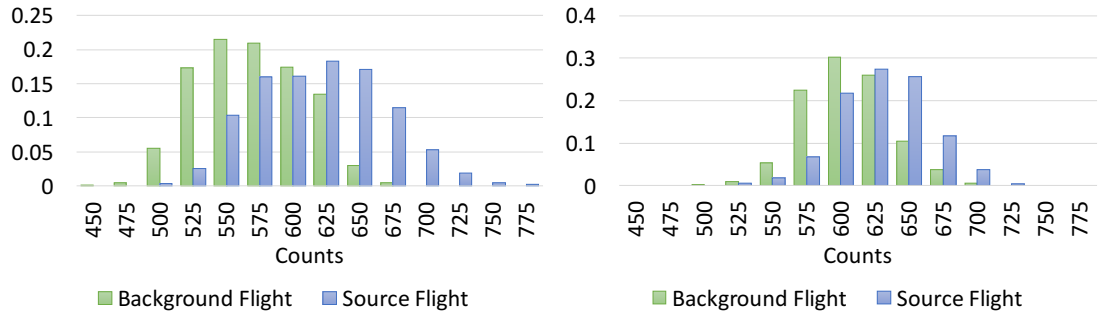


(a) Source configuration 1. (b) Source configuration 2.

Figure 4.16: The first (a) and second (b) flight paths at Kentland Farms, Blacksburg, VA, shown in Google Maps, where the color of each point represents the counts, calculated by summing the 1024-d spectral vector at each position. The magenta circles show the ground truth locations of the sources, and the red diamonds show the positions of max counts, which are set as destinations for the UGV to visit and take additional measurements. (a) Aerial search path for the first configuration, where 4 radiation sources (2 Ho, 1 Ba, 1 Cs) are placed at a single location. (b)Aerial search path for the second configuration, where 2 Ho sources are placed at one location (position closest to the location of max counts), and 1 Ba and 1 Cs sources are placed at a second location.

For each mission, we performed a background scan of the mission area and a flight for the main mission with radiation sources present. These background scans are never used

to assist in finding the sources, but help provide context for the data observed during the source flights. For each mission we ran paired t-tests between the background and source flights for each mission to test the null hypothesis that the counts (not normalized) have identical means. In both cases we were able to reject this null hypothesis with a p-value of 0.05, and conclude that statistically significant observations were made during the source flights. Histograms of the background and source flights for each mission are shown in Figure 4.17.



(a) Mission 1. Background flight: $\mu = 558$, $\sigma = 38.9$. Source flight: $\mu = 606.7$, $\sigma = 48.1$.

(b) Mission 2. Background flight: $\mu = 593.9$, $\sigma = 30.9$. Source flight: $\mu = 617.6$, $\sigma = 33.4$.

Figure 4.17: Histograms of the counts for each mission (normalized), which includes the main mission with radiation sources and the background scans. For each mission we ran t-tests between the counts for the background and source flights to verify that statistically significant differences were observed. In both cases reject the null hypothesis, that their means are identical, with a p-value of 0.05.

Orthophoto and DEM. The orthophoto and DEM output by Agisoft are shown in Figure 4.18. Note that the DEM values are incorrect for the building with the white roof. This does not affect path planning, however, as this area can still be identified as non-traversable because of the discontinuity with surrounding regions. Also, this provides further motivation for the 2D semantic segmentation of the aerial images. A close up of vehicles is shown in Figure 4.19 to illustrate the level of detail in the final output by Agisoft.

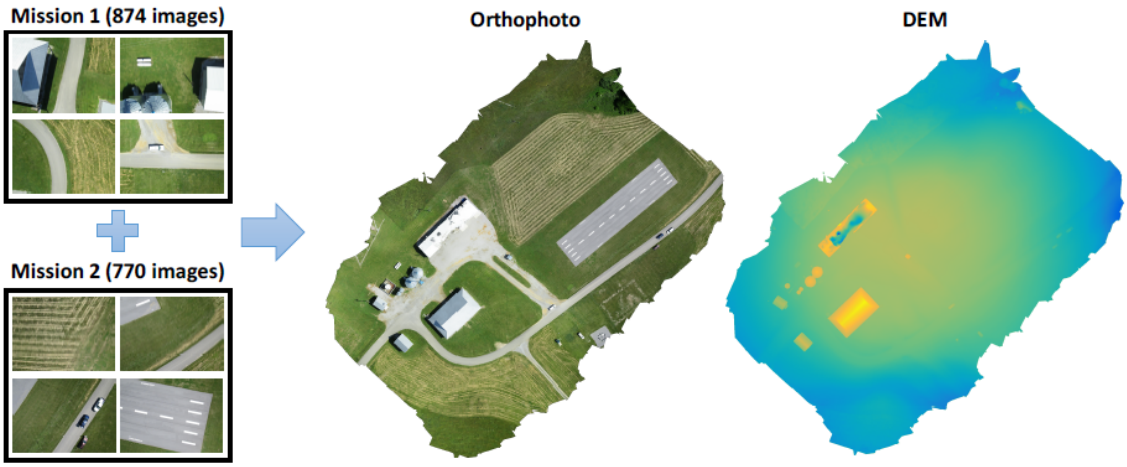


Figure 4.18: The orthophoto and DEM generated by Agisoft.

Figure 4.19: A view of the point cloud of Kentland Farm generated using Agisoft that illustrates the level of detail possible using off-the-shelf cameras.

While GPS was used for all experiments presented in this chapter, the approach has the potential to work in GPS-denied areas, where the mission is still feasible, but certainly more difficult. The approach presented in Chapter 5 could be used. SfM and stereo vision can still be used to generate a global map, but the individual systems must now operate by transforming their local coordinates to shared global coordinates. By registering the UGV's position and heading in the aerial map, the maps generated by the UGV on the

ground could then be aligned with the aerial map to make updates to the global map with observations made from the ground. Generating aerial maps (SfM, stereo, *etc.*) without GPS can mean longer run times and less accurate reconstructions. For example, part of the SfM pipeline includes features being matched between pairs of images when generating the 3D reconstruction. A naive approach would search for matches in $\binom{\#images}{2}$ pairs of images. GPS can be used to only search for matches in pairs of images located near one another. Without GPS, we can still limit the number pairs to be searched by image clustering (*e.g.* clustering using GIST image descriptors [127]). While not as robust as using GPS, and still resulting in pairs of images with no matches, this is still much faster than the naive approach.

When dealing with uncertainty in the GPS measurements, algorithms such as Kalman filters can be used with visual SLAM to update position beliefs. One concern for our application is having an inaccurate path be sent to the UGV. While this is not ideal, the UGV would still be able to scan the terrain around it to determine what is traversable and what is not. To correct the path, identifying landmarks that are matched to the aerial map to make the correction would be one possibility. Another possibility would be to have the UGV perform semantic segmentation to make the correction. For example, the GPS coordinates output for a path along a road may be located on a neighboring grass region. Semantic segmentation would allow the UGV to identify the location of the road nearby and make the correction.

Another problem with using a single-camera system in GPS-denied environments is the inherent scale ambiguity associated with structure from motion. Our 2-camera imaging system can be used to resolve this [116], by scaling the 3D reconstruction to an interpretable size using the known baseline between the cameras. A UGV could then localize itself in a local coordinate frame defined by the 3D reconstruction.

Semantic Segmentation and Path Planning. We perform semantic segmentation on tiles of the orthophoto using ALE [5] and then compare our results to ground truth annotations. Per-category results are shown in Table 4.3. We measure results in terms of precision and recall, where for each category c precision calculates how many of the instances classified as c are correct, while recall calculates how many of the ground truth instances labeled c have been correctly classified. True positives (TP), false positives (FP), and false negatives (FN) are used to calculate precision and recall as

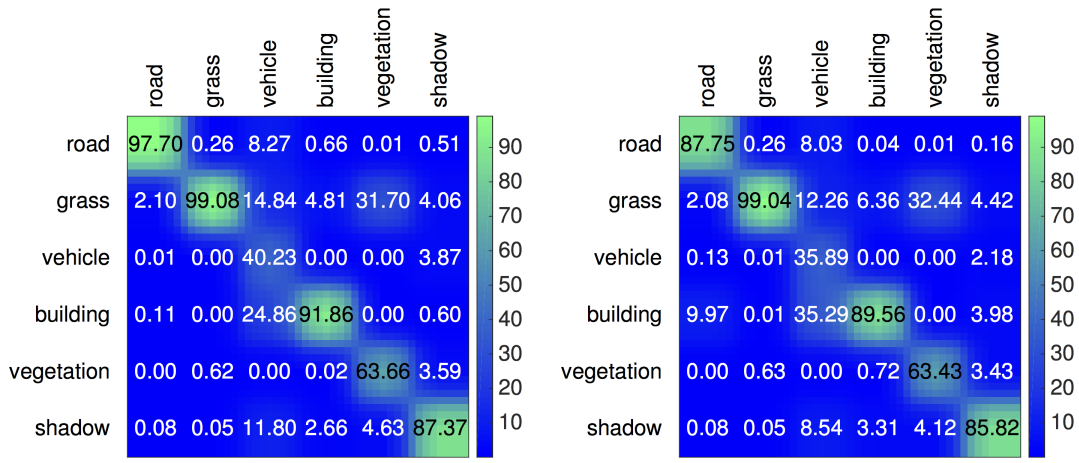
$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \ \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{4.7}$$

We show that our approach of reasoning about the 2D orthophoto and DEM to output final predictions performs better than a baseline that only reasons about the 2D orthophoto. The road and grass categories have very high accuracy, which is expected given that they are usually visually distinguishable from the other categories and each other. When confused with non-traversable categories, the DEM can be used to make corrections. The reliability of the model to segment these categories is also important for path planning, as these are the traversable categories for the UGV. The confusion matrices for the results of our approach and the baseline are shown in Figure 4.20. Note that the non-traversable categories are typically confused with one another. This makes no difference for the path planner, but there is still motivation to improve performance on these categories as this is useful for high-level reasoning, such as understanding that a radiation source is more or less likely to be present at certain coordinates (*e.g.* inside a vehicle). The ground truth annotation and semantic segmentation result are shown in Figure 4.21.

The planned missions are shown in Figure 4.22, where the red pixels display the path, the blue squares shows the locations of the obstacles, and the yellow triangles show the start/end positions. As seen, the planned path plans around the vehicles on the road that

Table 4.3: Quantitative results for the semantic segmentation of the Kentland Farms imagery, showing per-category, average, and global accuracies for our approach (2D + DEM) that uses the orthophoto and DEM to reason about category prediction, and a 2D only baseline.

method / metric	road	grass	vehicle	building	vegetation	shadow	Global	Average
2D precision	87.75	99.04	35.89	89.56	63.43	85.82	-	76.92
2D + DEM precision	**97.70**	**99.08**	**40.23**	**91.86**	**63.66**	**87.37**	-	**79.98**
2D recall	**98.57**	**98.78**	55.22	42.89	60.68	**85.42**	96.20	73.59
2D + DEM recall	98.41	98.74	**61.96**	**97.85**	**62.29**	81.06	**97.89**	**83.39**

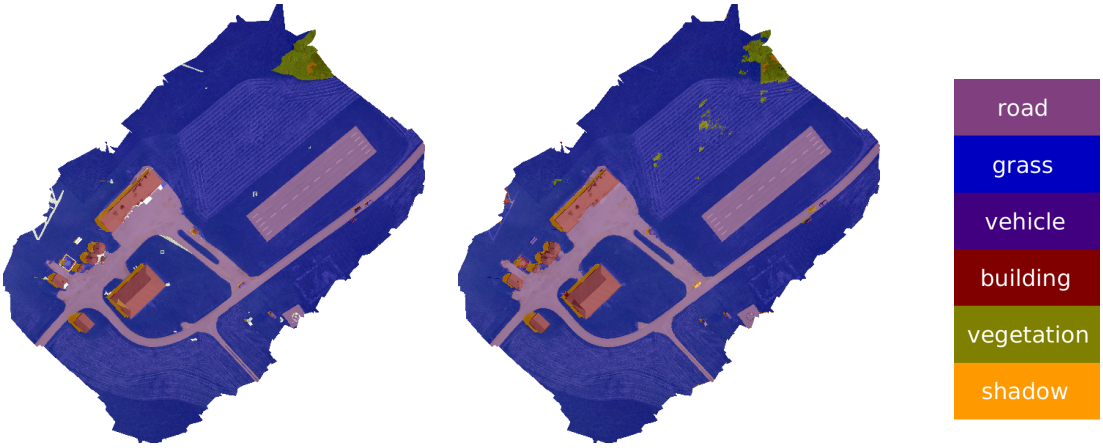


(a) 2D + DEM confusion matrix.

(b) 2D only confusion matrix.

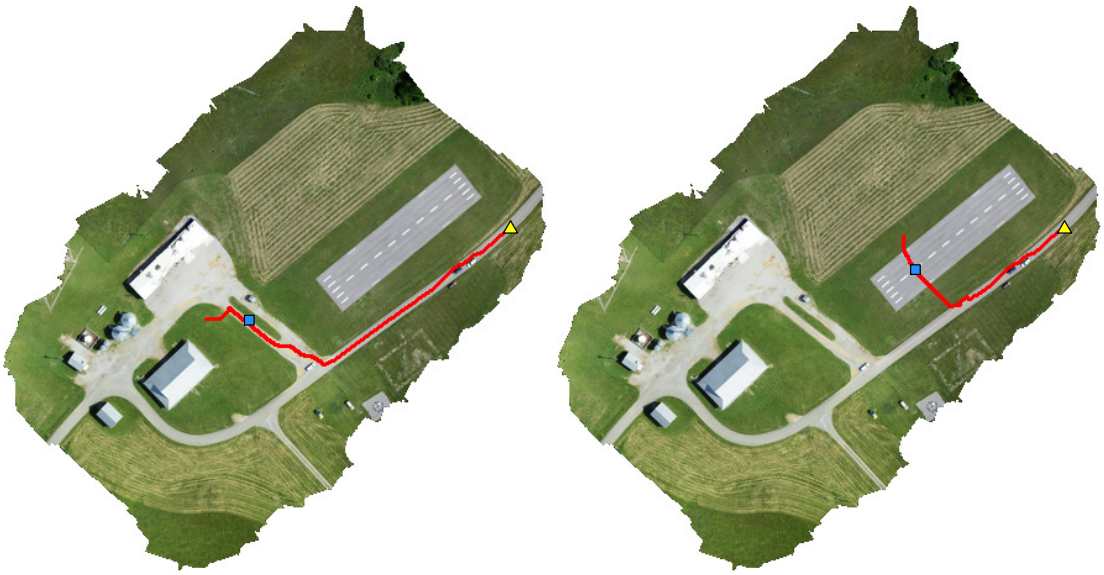
Figure 4.20: The confusion matrices for both our approach of using the orthophoto and DEM to perform semantic segmentation, and a 2D only baseline that only uses the orthophoto. The diagonal elements of the confusion matrices show the precision values from Table 4.3. The different colors in confusion matrices represent values between 0 and 100.

were present during the flight, but not during the ground experiments. In our experiments, we do not update the global map to remove obstacles that were present in the aerial map, but are no longer in the scene. This was not necessary for our experiments, but note that this could easily be incorporated by adding an additional process to analyze the LiDAR data.



(a) Ground truth image. (b) Segmentation result.

Figure 4.21: (a) Ground truth image of the orthophoto of Kentland Farms done with LabelMe [4]. (b) Result of segmenting the orthophoto by training the ALE [5] on our dataset and then refining the results using the DEM. The legend on the right can be used to map colors to categories.



(a) Planned path for Mission 1. (b) Planned path for Mission 2.

Figure 4.22: The planned paths for each of the two radiation source configurations. The start position (yellow triangles) was set on the exterior points on the orthophoto containing a road. The blue square shows the position where an obstacle was placed so the TURTLE was forced to find an alternative path when encountered. These paths were each generated in a matter of seconds.

4.5.3 TURTLE Results

The global LiDAR maps (DEMs) generated for each mission[6] by the TURTLE are shown in Figure 4.23. When multiple height values are observed at the same (x,y) the values are averaged. This was done for efficiency reasons, as storing all previous values so that the nth percentile can be calculated requires a significant amount more storage. We also experimented with taking the max, but observed that this was susceptible to noise.

Obstacle Detection and Avoidance. Local LiDAR scans were analyzed to find obstacles on or near the current path, and were used to update the global DEM and segmentation. Specific pixels associated with the obstacle are set in the segmentation, the region of which is dilated as a cautionary measure to make sure the full size of the obstacle is contained within the region that defines it in the segmentation. An updated path is then generated using the same path planning algorithm by taking the current position of the TURTLE as the start position, using the same goal position, and using the updated segmentation. The final paths taken by the TURTLE, with obstacles avoided, are shown in Figure 4.24. As seen, at each position in the path history the counts were mapped to a color value to represent intensity. The difference from Figure 4.22 can be seen where it has identified the obstacle and navigated around it. To return to the start position, we simply keep track of the waypoints visited on the way to the destination and then follow them back.

As we approach the source in each mission, we observe a significant increase in the counts, thereby confirming that a source is present. A plot of the counts over time for each mission can be seen in Figure 4.25. The distance to the goal for each mission is also shown to help understand the trends of the counts, see when the TURTLE is stationary, *etc*. For the counts of Mission 1, shown in Figure 4.25a, we see a gradual increase as the

[6]These can be used in post processing to help understand the scene around the area of radiation activity.

(a) LiDAR point cloud for source configuration 1.

(b) LiDAR point cloud for source configuration 2.

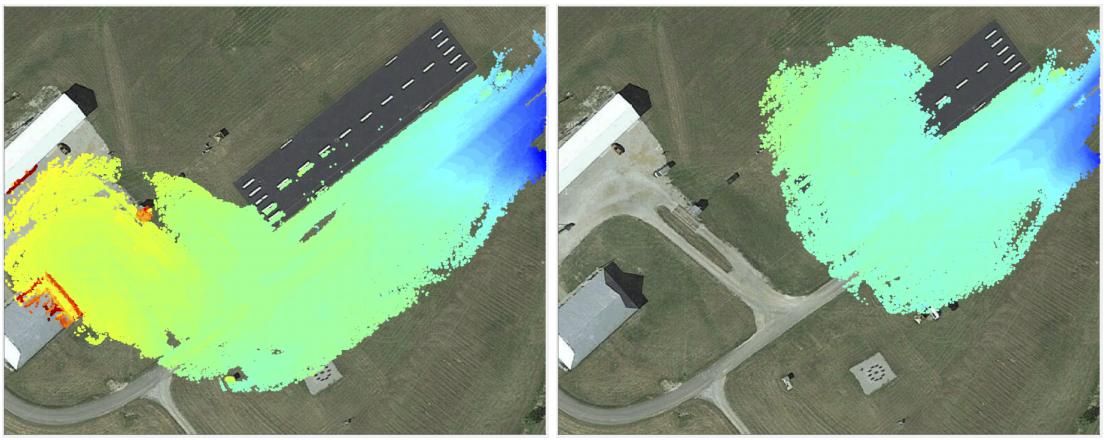
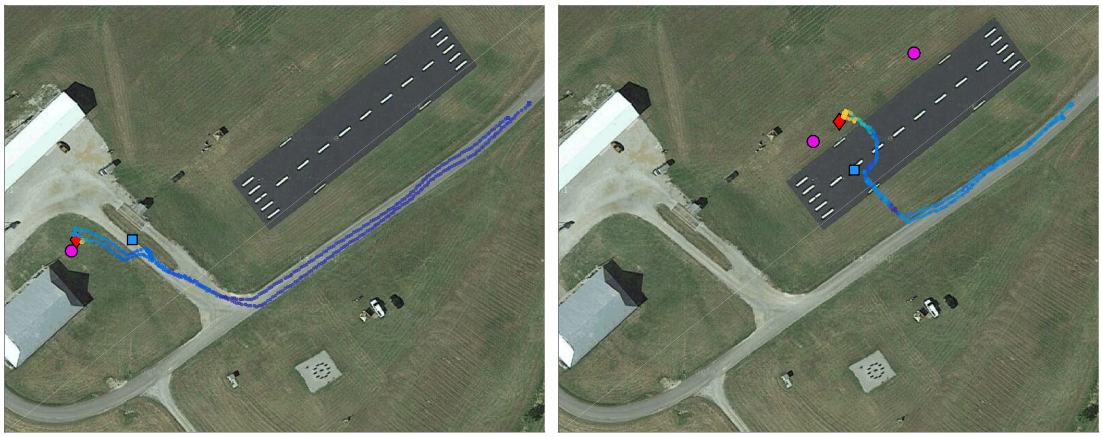


Figure 4.23: Global DEMs generated by the TURTLE's LiDAR for each search mission. During the construction of the DEM, height values were averaged for points with the same (x,y).

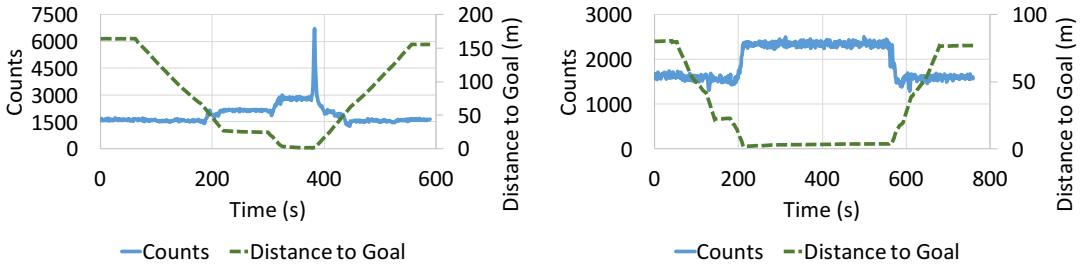


(a) Path taken for Mission 1.

(b) Path taken for Mission 2.

Figure 4.24: Paths taken for the missions of both source configurations where the counts were used to map to the colors seen at each waypoint. The magenta circles show the ground truth locations of the two source positions, the red diamond shows the position of max counts from the aerial data, and the blue square shows the position of where the obstacle was placed. As seen in both missions, the TURTLE avoids the obstacles, which was done by reasoning with both local and global information.

TURTLE moves closer to the source before remaining in place for several minutes. The spike in the counts, observed by a single data point, was attributed to the change in the



(a) Source configuration 1.

(b) Source configuration 2.

Figure 4.25: Plots of the counts over time for both radiation source configurations. The distance to the goal position is also plotted to help understand the trends in the counts. Upon arriving at the destination, the TURTLE performed a long-dwell measurement by remaining in place for a few minutes before returning to the start position, which explains the longer period with increased counts. The spike in (a) is believed to be a result of the TURTLE turning around to return home, during which the direction of the detector changed causing it to pick up a much stronger signal.

direction when the TURTLE turned around to return to the start position, resulting in a stronger signal to be seen by the radiation detector. By looking at Figure 4.25a, the spike occurs right at the time the TURTLE ends the dwell period and begins to return to the start position. For Mission 2, shown in Figure 4.25b, we see a sudden increase in the counts as it reaches the destination before performing the long-dwell measurement. The source location visited in Mission 2 is not as strong as the location of Mission 1, and therefore we do not see the gradual increase seen in Mission 1. In both cases, however, the presence of a radiation source near both locations of max counts from the aerial measurements was clearly confirmed by the TURTLE. In practical applications, images and video could be transmitted back to a remote base station where operators could take control of the TURTLE to perform additional tasks.

4.6 Conclusions

We presented an approach to the autonomous search for hazardous radiation sources in an unknown environment. We tested our approach in a 7 acre area containing buildings, roads, grass, vegetation, *etc*. To collect radiation data, elevation information, and a semantic understanding of the entire area, we used a UAV (the Yamaha RMAX) to fly over the area and collect gamma radiation data and 2D color images from off-the-shelf cameras. The radiation data was used to output positions of the strongest reading from the detector as a destination for a UGV (the TURTLE) to visit and collect more data. The imagery was used to create a georeferenced orthophoto and DEM of the scene, which were then used to perform semantic segmentation (*i.e*. assign a category label to each pixel in the orthophoto/DEM) with high accuracy. By using the DEM to reason about category predictions we were able to achieve significant improvements over the 2D only baseline (orthophoto only). These image-based outputs were then used to plan a path for the TURTLE to visit the points of interest from the radiation data, where costs of the path planning algorithm were dependent on the semantic segmentation. This resulted in a preference for the TURTLE to follow roads over grass.

After planning the paths, we deployed the TURTLE to run the two missions, where we place obstacles on each path so that it was forced to identify the obstacle and find an alternative route. The algorithms were successfully able to identify the obstacles, update the global map, and plan a new path around the obstacles to each destination. We also observed significant increases in the counts (sum of the 1024-d vector from the radiation detector) as the TURTLE approached the destination in each mission, confirming that sources were present in both cases. We demonstrated success for both the aerial and ground operations in our experiments to estimate and validate radiation source locations in an unknown environment. In future work, we plan to test our approach of using image-

based reasoning to perform more complicated search tasks in more challenging scenes. Also, although our experiments focus on the task of autonomously searching for radiation sources, we note that this approach can be applied to many sensing tasks with the possibility of multiple aerial and ground vehicles driving the search effort.

We believe that with real-time 3D reconstructions from imagery, a real-time response with our system is possible. With more expensive machine vision cameras we believe we could have used existing reconstruction software to accomplish this. However, we note that we drastically reduce the price of the system with our 2 off-the-shelf Canon A-810 cameras, which were triggered by an Arduino microcontroller. For future work, we are currently developing our own code to perform faster 3D reconstructions from images taken from our stereo setup with the Canon A-810 cameras by taking advantage of the known extrinsics of the imaging rig. We believe that this will help close the gap between cost and efficiency. We also believe that the annotated dataset we used to train the semantic segmentation model does generalize to many similar types of scenes, and have observed this by performing a qualitative evaluation on other test areas that we have not yet annotated to measure full performance. As more data is annotated with additional categories, and as models start to make better predictions, we believe that a system similar to the one presented in this work will become very useful for many types of disaster response scenarios.

Overall the experimental results that we obtained were favorable. We did learn the importance of active search from the ground. In one of our experiments we unexpectedly failed to identify the location of the second radiation source from an altitude of 30m both by manual inspection of the counts, and with highly capable radiation detection algorithms that analyze all dimensions of the radiation data coming from the detector [98]. We therefore used the max counts as the estimated position of the source in each experiment.

In future work we plan to expand the search from the ground to better detect these weaker sources. We still believe that our semantic maps of the area can assist in this process. For example, if we know the locations of buildings and vehicles, then the UGV can be tasked to visit these places and attempt to enter them to collect data not observed by the UAV. We may also be able to learn radiation background signatures for different semantic categories and use that to make more informed decisions about the presence or absence of a radiation source at a particular location.

Chapter 5

GPS-denied Localization of a UGV

5.1 Introduction

Localization in a global map is critical to success in many autonomous robot missions. This is particularly challenging for multi-robot operations in unknown and adverse environments. Here, we are concerned with providing a small unmanned ground vehicle (UGV) the ability to localize itself within a 2.5D aerial map generated from imagery captured by a low-flying unmanned aerial vehicle (UAV). We consider the scenario where GPS is unavailable and appearance-based scene changes may have occurred between the UAV's flight and the start of the UGV's mission. We present a GPS-free solution to this localization problem that is robust to appearance shifts by exploiting high-level, semantic representations of image and depth data. Using data gathered at an urban test site, we empirically demonstrate that our technique yields results within five meters of a GPS-based approach.

UAV-UGV collaboration in GPS-denied environments is hard. Shared instructions often require the coordinate systems of the UAV and UGV to be registered with one an-

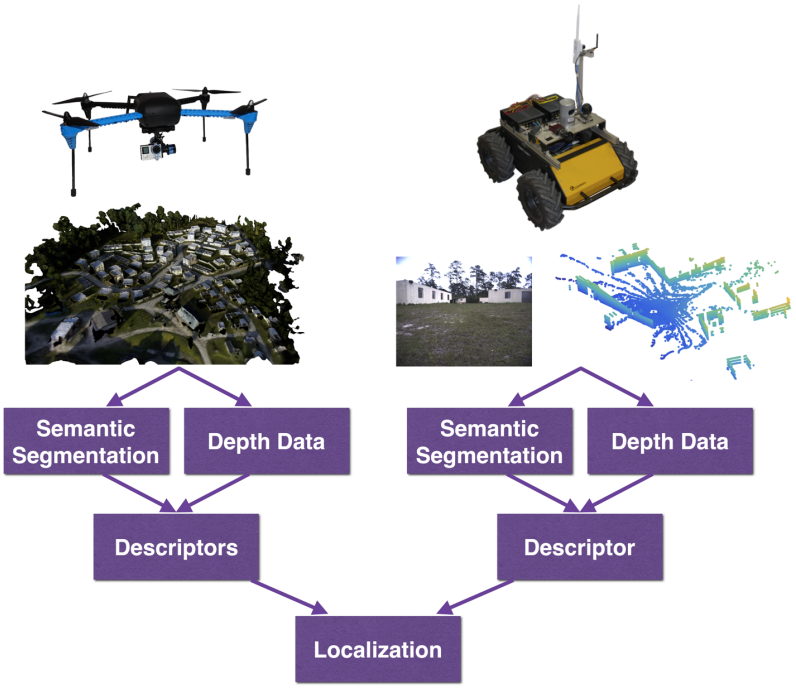


Figure 5.1: An overview of our approach. A UAV captures overhead imagery of a scene to generate a 2.5D orthophoto. Using semantics and depth information, descriptors are created for every traversable pixel in the aerial map. The UGV captures imagery and laser scans. Semantic segmentations and range data are then used to create a descriptor for the UGV data. Descriptor similarities are used to score each traversable pixel in the aerial map, after which a particle filter is used to reason about the location of the UGV.

other. For example, when a mission control system uses an overhead map to plan paths for a UGV, it becomes critical for the UGV to register its position in this map. For time-sensitive missions, this must be done quickly, where visiting several areas before estimating the location may not be possible. GPS can be used to help perform registration, but it is not always available.

Our scenario of interest is one in which multi-robot teams must operate in unknown and adverse environments, and so we consider the problem of localizing a UGV in a map generated by a UAV when GPS information is unavailable. Accurate registration is much more difficult without GPS. The UGV must use data collected from a different perspective than that of the UAV, and the scene itself may have changed since the overhead

map was generated. In addition, registration techniques often rely on data with noisy measurements, imperfect machine learning models, and small errors in sensor calibration.

We perform tests in a challenging urban environment with no *a priori* information (*e.g.* road networks) about the scene. We focus on the case where a UAV flight takes place before the UGV mission. Clearly, the scene may change between the end of the flight and the start of the UGV mission. Structural scene changes (*e.g.* object moves to another location) are one type of change that may occur. The approach we present is robust to these types of small scene changes. This is helpful not only for the small changes that may occur, but also for the inherent perspective problem of the task where structures simply look different from the air than the ground. For example, a laser scan from the UGV may not see high enough to observe any structure above a large opening in a building. Therefore, points returned within the range of angles that capture this opening will not match well to the corresponding parts of the aerial map that observe a roof.

Appearance-based scene changes (*e.g.* trees losing leaves) are another concern. Matching color information directly has the potential to help immensely, but will likely fail in the presence of these appearance-based changes. We therefore do not use such low-level representations. We use semantic segmentations of the aerial and ground data to classify points with category labels (*e.g.* grass). This creates a high-level representation of the scene's appearance, where pixels and 3D points are now represented by semantic categories instead of raw color values. This makes our approach robust to appearance-based scene changes.

We propose a GPS-free solution that requires only image, LiDAR, and vehicle odometry data. An overview of our approach can be seen in Figure 5.1. The contributions are:

1. A multi-robot system capable of autonomously understanding a scene in GPS-

denied environments via joint semantic reasoning about the scene from appearance and depth data.

2. A UGV localization algorithm shown to localize a UGV in an urban environment with an average difference to GPS under 5m, where the algorithm is robust to appearance-based scene changes, small structural scene changes, and occasional ambiguous regions.

5.2 Approach

We propose an approach that integrates range, semantic, and trajectory information to localize a UGV in an aerial map. We pose the problem as one of finding a mapping between the UGV's trajectory, generated without GPS, to coordinates in the 2.5D orthophoto. In Section 5.3, we describe how we generate the semantic segmentations used.

5.2.1 Descriptors and Scoring

To localize the UGV in the aerial map, we score the similarity between descriptors generated from UGV data and similar descriptors generated with the UAV data. Our proposed descriptors include range and semantic information to describe each pixel in the 2.5D orthophoto, and each local image and laser scan from the UGV. We use the same process to generate both the aerial and ground descriptors. We define N scan lines in a 360° view, where the angle between subsequent scan lines is $\alpha = \frac{360°}{N}$. In our experiments, we use $N = 60$, $\alpha = 6°$. We search along each scan line until an obstacle is detected or the max distance (40m) is reached. If an obstacle is detected, then the appropriate element of the descriptor is set to the distance of the obstacle from the current pixel position in the 2.5D orthophoto or the origin of the laser scan from the UGV. If no obstacle is detected, then

an ‘invalid’ label is assigned to the descriptor element.

To incorporate the semantic information into the descriptor, we assign the appropriate element of the descriptor to the semantic label of the segmentation at the location of where the obstacle was detected. If no obstacle was detected, then this element is also set to an ‘invalid’ label. An illustration of these descriptors is shown in Figure 5.2.

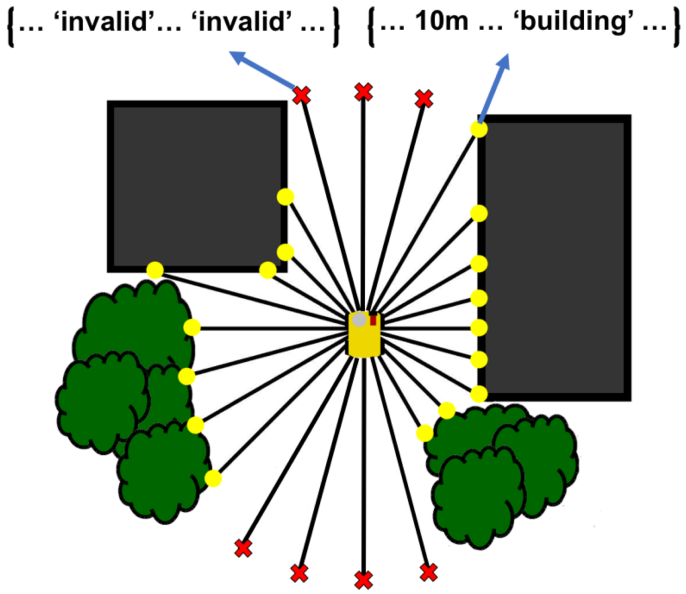


Figure 5.2: Illustration of the range and semantic descriptors. These descriptors are generated for both the 2.5D orthophoto and the UGV data. Scan lines are generated at equally spaced angles (α) up until a max distance. If an obstacle is detected on a scan line, the distance to the obstacle and its semantic label are recorded at the appropriate elements. If no obstacle is detected, then invalid labels are recorded for the appropriate range and semantic elements.

Aerial Descriptors. Before the UGV mission, descriptors are calculated for all M traversable pixels in the 2.5D orthophoto. We define the matrix of aerial descriptors for all traversable points as $\mathbf{D}_{uav} = \{\mathbf{F}_{depth}, \mathbf{F}_{semantic}\}$, where $\mathbf{F}_{depth}$ and $\mathbf{F}_{semantic}$ are matrices of size $N \times M$ that hold the range and semantic portion of the descriptors, respectively.

Ground Descriptors. We define descriptors of the UGV data for a particular time (t) and viewing angle in the aerial map (ω) as $\mathbf{d}_{ugv}(t, \omega) = \{\mathbf{f}_{lidar}(t, \omega), \mathbf{f}_{semantic}(t, \omega)\}$, where

$\mathbf{f}_{lidar}$ and $\mathbf{f}_{semantic}$ are N-vectors that hold the range and semantic portion of the descriptors. Because of the UGV camera's limited field of view, we include the term ω to define the view angle of the UGV. Changes to ω are represented by circular shifts of $\mathbf{D}_{uav}$.

In Section 5.3, we show how we identify obstacles. By projecting the 3D points classified as obstacles into the xy plane, we generate a 2D obstacle map similar to the one generated from the aerial data. We then feed this obstacle map for the laser data into the same function that generates the aerial descriptors. However, since we do not use an omnidirectional camera, it is only possible to obtain valid semantic labels for a subset of the scan lines. During the descriptor similarity scoring process, we search over all values of ω to score each possible position p.

Descriptor Similarity Scoring. Given a descriptor for the UGV data $\mathbf{d}_{ugv}(t, \omega)$ and the set of aerial descriptors $\mathbf{D}_{uav}$, we search for the closest $\mathbf{d}_{uav}(p) \in \mathbf{D}_{uav}$ with a custom similarity measure. Binary vectors represent the element-wise similarity between $\mathbf{d}_{uav}(p)$ and $\mathbf{d}_{ugv}(t, \omega)$, where we define elements of the range and semantic portions as

$$\delta_{range,i} = \begin{cases} 1, & \text{if } |\mathbf{f}_{lidar,i}(t, \omega) - \mathbf{F}_{depth,ji}| < d_{th} \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

$$\delta_{semantic,i} = \begin{cases} 1, & \text{if } \mathbf{f}_{semantic,i}(t, \omega) = \mathbf{F}_{semantic,ji} \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

where d_{th} is the max difference in distance allowed between the two descriptors being scored. We set d_{th} to 10% of the max LiDAR distance (40m) on each scan line (4m) to accommodate differences in scale of the depth data and the different perspectives. We do not set $\delta_{semantic,i}$ to 1 if one or both of the descriptor values of the segmentation are invalid

(*i.e.* no obstacle was found on the corresponding scan line of that element).

We calculate the similarity between the descriptors as

$$s(\mathbf{d}_{ugv}(t,\omega),\mathbf{d}_{uav}(p)) = \sum_{i}^{n} \delta_{range,i} + \gamma\delta_{semantic,i} \tag{5.3}$$

where γ is used to scale the segmentation score. There are a fixed number of scan lines that *can* have valid semantic labels. However, not all scan lines in this subset will find obstacles, and therefore there are a variable number of valid semantic labels. This is the reason we scale the semantic score with γ. Given N, the length of $\mathbf{f}_{semantic}$, and the number of valid labels for the segmentation portion of the current UGV descriptor (v), we set $\gamma = \frac{N}{v}$.

We chose this similarity measure over alternatives (*e.g.* Euclidean distance), because it is robust to small structural changes in the scene. For example, if a new obstacle appears in the scene after the UAV flight, then a subset of each UGV descriptor that observes the new obstacle will be affected. The aerial descriptors for the corresponding points in the 2.5D orthophoto will not observe this obstacle, and will therefore potentially observe other obstacles much farther away. If Euclidean distance is used to measure similarity between the aerial and ground descriptors, then it is not likely these descriptors will match well. With our similarity measure, we can still score these descriptors as being similar, as long as the rest of the scan lines, that do not observe the new obstacle, match well.

We score each position of the UGV independent of previous predictions and odometry as

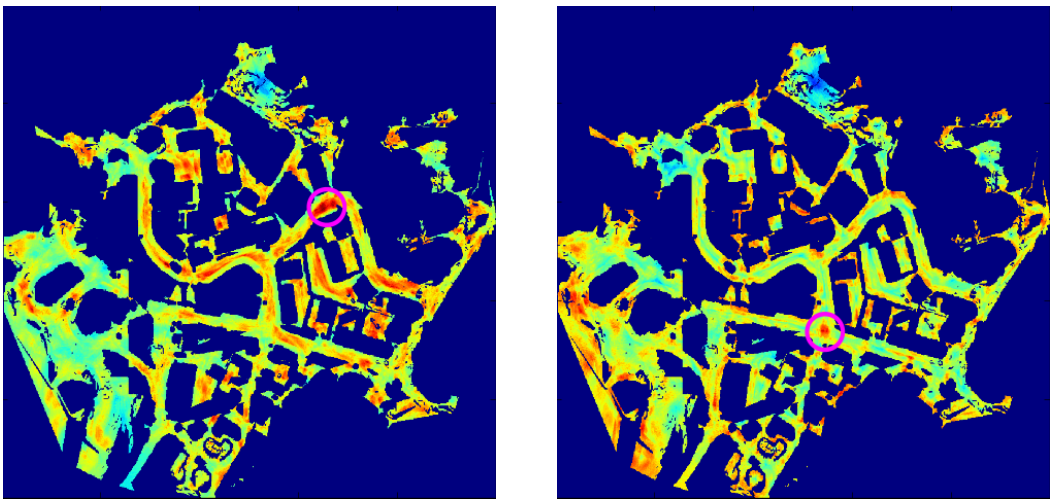
$$\mathbf{S}(p) = \max_{\omega}\ s(\mathbf{d}_{ugv}(t,\omega),\mathbf{d}_{uav}(p)), \tag{5.4}$$

where $\mathbf{d}_{uav}(p)$ is the descriptor in $\mathbf{D}_{uav}$ that corresponds to position p. Independent pre-

dictions for the UGV's location are made by finding

$$\hat{x}_t = \max_p \mathbf{S}(p). \tag{5.5}$$

By mapping these descriptor similarities to colors, we can display heat maps for each descriptor generated with the UGV's data. Two examples of this are shown in Figure 5.3. In Figure 5.3a, we show how positions at street centers with buildings on both sides can cause ambiguity. In Figure 5.3b, we show how more unique locations are easier to identify.



(a) Heat map in an ambiguous region. (b) Heat map at an intersection.

Figure 5.3: Descriptor similarity heat maps, where the pink circles in each figure contain the ground truth location of the UGV. (a) Ambiguous region in between two buildings, where the UGV is at the street center. (b) UGV at an intersection (less ambiguous).

5.2.2 Incorporating Odometry

We obtain trajectories in GPS-denied environments for the UGV using a GPS-free version of the SLAM approach described in [139], where trajectories are generated using images, LiDAR, and the UGV's odometry. Our problem can be viewed as mapping these trajectories to positions in the 2.5D orthophoto. Our approach involves reasoning about local

likelihoods from descriptor similarities and a prior generated using the UGV's estimated current position from the trajectory data. To calculate this prior, we find a transformation using the position estimates ($\hat{x}_0 \dots \hat{x}_{t-1}$) and the global trajectory ($x_0 \dots x_{t-1}$), which has not been georegistered. We find transformations between this trajectory and our position estimates using RANSAC. At each iteration, we sample points and find Procrustes transformations. We define the predicted position from the current transformation as $\tilde{x}_t$, around which we center the prior.

Note that scale ambiguity is not a concern here, since we are mapping the trajectories from a local coordinate frame to the orthophoto. Therefore, even monocular SLAM approaches, such as [110], can be used with our approach.

The reason we use previous independent predictions of position ($\hat{x}_0 \dots \hat{x}_t$) at each iteration with RANSAC is that we do not want the next position estimate ($\tilde{x}_t$) to be too heavily influenced by the last estimate ($\tilde{x}_{t-1}$). For example, if the UGV is driving through an ambiguous part of the map, then it may not localize well until it reaches a more distinctive part of the scene. Incorrect predictions in ambiguous parts of the map tend to be scattered, and therefore removed as outliers when finding transformations with RANSAC.

5.2.3 Particle Filter

We use a particle filter to exploit temporal information and predict the location of the robot at each iteration. Each particle calculates its weight using the likelihoods output by our descriptor similarities and the prior distribution of the UGV's next position. We define the position of the ith particle at time t as y_t^i. Our likelihoods from our descriptor similarities are defined as $\frac{\mathbf{S}(y_t^i)}{\sum_p \mathbf{S}(p)}$, where $\mathbf{S}(y_t^i)$ is the score in $\mathbf{S}$ for y_t^i. We use a prior distribution of the UGV's position for each particle by sampling from $\mathcal{N}(\tilde{x}_t, \sigma^2\Sigma)$. At each time step we update the particles as

$$y_t^i = y_{t-1}^i + \tilde{x}_t - \tilde{x}_{t-1} + u_i, \quad u_i \sim \mathcal{U}[0, \lambda] \tag{5.6}$$

where $\tilde{x}_t - \tilde{x}_{t-1}$ shifts the particles in the direction of the next estimated position $\tilde{x}_t$, and u_i is a sample from a uniform distribution, where we set $\lambda = 15$ pixels (5.1m). The importance weight of each particle (w_t^i) is then calculated as

$$w_t^i \propto \frac{\mathbf{S}(y_t^i)}{\sum_p \mathbf{S}(p)} n_i, \quad n_i \sim \mathcal{N}(\tilde{x}_t, \sigma^2 \Sigma) \tag{5.7}$$

where we normalize these importance weights and resample the particles with them. We estimate the position of the UGV at each time step as

$$\bar{x}_t = \sum_i w_t^i y_t^i \tag{5.8}$$

where $\bar{x}_t$ is the weighted average of the particles' positions.

5.3 Experiments

5.3.1 Systems and Setup

We perform experiments at an urban test environment. The scene contains buildings, vegetation, grass, and roads. To capture the aerial imagery used to generate our 2.5D orthophoto, we mount a GoPro camera to a 3DR IRIS. The 2.5D orthophoto is generated with Pix4D[1], but other 3D reconstruction techniques [110] may be used.

On the ground, color images and laser scans are captured by a Prosilica GT2750C camera and Velodyne HDL-32E LiDAR device, respectively. These sensors are calibrated and mounted on-board a Husky robot manufactured by Clearpath [140]. We use the calibra-

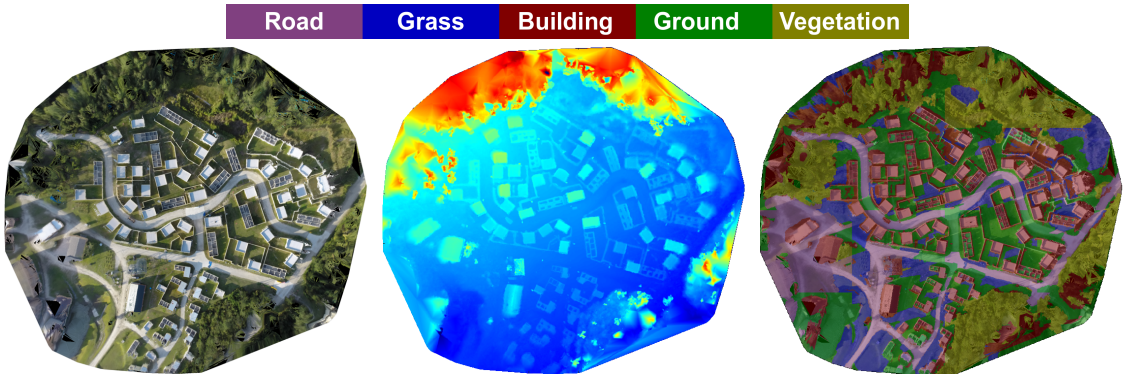
[1] https://pix4d.com/

tion of the two sensors to project 3D points into the 2D images. By performing semantic segmentation on the 2D images, we can obtain semantic labels for a subset of the points in each laser scan. We perform semantic segmentation on the 2.5D orthophoto and the UGV's imagery by training on annotated aerial and ground image datasets, respectively, which contain no images at or near the test site. Range-based descriptors are computed using the digital elevation map (DEM) from the aerial data and the laser scans from the ground. We log GPS from the UGV and georegister the orthophoto only to obtain an approximate measurement of accuracy for our estimated position outputs. No geospatial information was used to estimate the location of the UGV in the 2.5D orthophoto.

5.3.2 Segmentation

To perform localization without relying on low-level appearance-based features (*e.g.* color, texture), we use segmentations of the 2.5D orthophoto generated from the UAV imagery, and segmentations of the ground imagery and LiDAR. Below we describe each segmentation process.

Aerial-view Segmentation. We use a two-stage approach to segmenting the 2.5D orthophoto. First, we use the Automatic Labeling Environment (ALE) [5] to train a segmentation model using a dataset of 78 images, annotated with ground truth categories, captured from low-flying UAV in different environments. The orthophoto is then segmented using the trained model. The second stage of the process is to use the DEM to refine the segmentation. Using obstacles identified with the DEM, we correct the predictions of pixels inside of obstacle regions classified as traversable categories (*e.g.* road) by reclassifying them as non-traversable categories (*e.g.* building). We follow a similar procedure for pixels classified as non-traversable categories inside of non-obstacle regions. We assume that roads will be confused with buildings, and that grass will be confused



(a) Orthophoto (b) DEM (c) Segmentation

Figure 5.4: The aerial data used in our experiments, which was generated using color images taken at an urban test site by a small, low-flying UAV. (a) is the orthophoto of the test site, (b) is the DEM, and (c) is the semantic segmentation generated using the orthophoto and DEM. The legend at the top shows colors of the semantic categories.

with vegetation. We use this rule-based approach to make the corrections.

To identify obstacles in the DEM, we first generate an edge map. We initialize the set of ground pixels at the position farthest from an edge. The set of ground pixels is then iteratively expanded, where neighboring pixels with an elevation difference below a threshold are added to the set. Pixels that do not belong to the ground region are considered obstacles. The orthophoto, DEM and segmentation are shown in Figure 5.4. The categories shown at the top of Figure 5.4c are the final set of categories for the segmentation. However, the segmentation model is trained with the category 'shadow'. Traversable regions in the output classified as 'shadow' are assigned the label 'ground', and obstacle regions classified 'shadow' are assigned the label 'building'. The obstacle map is shown in Figure 5.5.

Ground-view Segmentation. We segment images from the UGV's camera using the ALE [5] to train a model on a set of 100 annotated images taken from scenes outside of the test site. In a similar procedure to the segmentation of the aerial data, we identify obstacles in the laser scans to refine the semantic predictions of the 3D points. We represent the 3D

Figure 5.5: Obstacles identified in the scene using the DEM generated by the aerial imagery. By seeding the ground region and iteratively expanding the region, the points in the overhead map not classified as ground are classified as obstacles.

point clouds as $\{x, y, h\}$, where h is the height of the points. We downsample the points by rounding the x, y values to the nearest tenth and then keep the unique points. Delaunay triangulation is run on the unique points $\{x, y\}$ to create an adjacency matrix for the points after projection to the xy plane. For a pair of neighboring points, say $\{(x_i, y_i), (x_j, y_j)\}$, point (x_i, y_i) is said to be an obstacle if $h_i - h_j$ is greater than 0.2m. We use the same rule-based approach as the aerial segmentation to correct the confusion between roads vs. buildings, and grass vs. vegetation.

We make an assumption that we can be located on either road or grass. By predicting which surface type we are currently on, we can use that information to better localize the UGV by possibly reducing ambiguity. To predict the surface type, we simply use the image segmentation output by the ALE, and take the mode of the segmentation outputs for the bottom portion of the image. If we predict that we are on a road, then we set the scores for all grass regions to 0. Similarly, we set scores for road regions to 0 when we

predict that the UGV is on grass.

5.3.3 Results

When comparing to GPS, we consider two approaches: (1) We output our estimate of the position using only the information we have up until that point. This measures the accuracy of how well the UGV was able to localize itself in real-time. This is important for the UGV to make decisions immediately. (2) Given the entire trajectory at the end of the mission, we map it to the orthophoto. This measures how accurately the UGV was able to localize its position history, which may be important for subsequent missions.

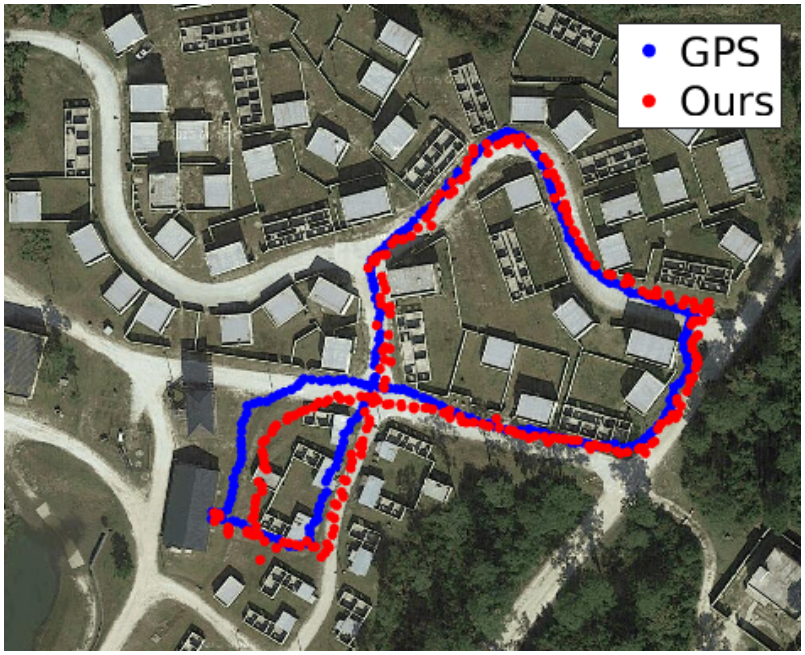


Figure 5.6: Google Maps overlay of path from GPS (blue) and our predictions (no GPS used).

We present results in Table 5.1. RANGE and RANGE-FULL only use the range portion of descriptors. The RANGE and RANGE-SEMANTIC approaches predict predict the position estimates with the data available up until the time of the current prediction. This is

more challenging and tests the ability of the approaches to localize the UGV quickly so that real-time decisions can be made. The RANGE-FULL and RANGE-SEMANTIC-FULL approaches predict the position estimates once the entire trajectory has been generated. This shows that improved localization can be performed once the whole trajectory becomes available. To obtain these results we calculated the average differences to GPS of the full pipeline run for 20 iterations, where at each iteration we calculated the average difference to GPS over all positions. This was done because of the random element of our approach from RANSAC and the particle filter.

Table 5.1: Average difference to GPS (meters) with standard errors for each approach (lower is better).

RANGE	RANGE-FULL	RANGE-SEMANTIC	RANGE-SEMANTIC-FULL
5.392 ± 0.125	5.185 ± 0.285	4.676 ± 0.014	$\mathbf{4.61} \pm 0.088$

We show an overlay of our outputs for one of the runs with RANGE-SEMANTIC and the GPS measurements on a satellite image of the test site in Figure 5.6. This helps illustrate that GPS is not ground truth, since we know the UGV was located on the road, not entering and exiting the buildings. We also observe that the largest difference with GPS seems to occur when the UGV is in the grass near the left-most points of the path, near the end of the mission. We believe this difference is caused by a mixture of ambiguous descriptors and possible drift associated with the SLAM algorithm. We also observe that RANGE-SEMANTIC generates less ambiguous descriptors in some regions than the range-only approaches. One example is near the right-most points of the path where the UGV is in between the trees and the buildings. Our approach correctly segments vegetation on one side of the UGV and buildings on the other side, which are incorporated into the descriptors in that area.

5.4 Conclusions and Future Work

We have demonstrated a successful approach for localizing a UGV in GPS-denied environments. The difference of our approach and GPS is under 5m for the complicated urban environment that we test at. The UGV uses images and laser scans to localize itself in a 2.5D orthophoto generated from aerial imagery captured by a low-flying UAV *and* we show semantics help. We represent the appearance portion of the aerial and ground data with semantic segmentations so descriptor similarity scores are robust to appearance-based scene changes. Our similarity measure to score pairs of descriptors also allows for small structural scene changes.

One potential direction for future work would be to develop an active search strategy to assist in localization. The UGV could navigate to areas it believes will be less ambiguous to the localization algorithm. Another possibility would be to register UAV and UGV satellite imagery, where the systems could collaborate to perform localization. For example, the UAV could be used to quickly gather higher resolution imagery and depth data in areas that will potentially assist the UGV's ability to localize itself.

Chapter 6

Conclusions and Future Work

In this chapter, the contributions of the dissertation are summarized, directions for future work are discussed, and final conclusions are made.

6.1 Summary of Contributions

The objective of this dissertation is to help advance the perception capabilities of autonomous unmanned systems operating in unstructured environments.

The first contribution presented was the approach to joint reasoning for multiple perception modules. There were two main challenges of constructing a joint model that the approach is able to overcome: (1) a joint model must reason about all possible combinations of solutions across all of the perception modules, which leads to a search space explosion; (2) models are inaccurate, so if each module outputs its most likely belief in isolation then performance will likely be worse than a joint model (difficult to construct) that integrates information between the modules. In the approach presented, each module independently proposes a diverse set of plausible hypotheses that are then jointly reasoned about with a "mediator" model. This overcomes challenge (1), because if both modules

are proposing only 10 hypotheses, meaning the model has to choose a pair of hypotheses from only 100 pairs, which is very small compared to a joint model that tries to reason about all possible combinations of solutions across the modules. The approach overcomes challenge (2), because although each module operate independently, the hypothesis set is not limited to the first-best output. An urban scene understanding experiment, which performs simultaneous 2D semantic segmentation of image data and 3D semantic segmentation of image-derived point cloud data, was presented that demonstrates the success of this approach.

For the second contribution, an approach to autonomous radiation search in unknown environments was presented. By developing a semantic understanding of the area of interest, path planning was performed for a UGV to visit radiological points of interest with semantics incorporated into the A* algorithm. We also demonstrated the ability to detect obstacles locally on the ground with LiDAR and then find a path around the obstacle using both local and global information.

In the third and final contribution, an approach to localize a UGV in a GPS-denied environment was presented. The UGV was able to successfully localize itself in a 2.5D orthophoto generated from aerial imagery with an average difference from GPS under 5m. By using range data from the depth data of the aerial map and the LiDAR scans from the ground, as well as the semantic segmentations of the aerial and ground imagery, descriptor similarities were used along with SLAM trajectories to accurately estimate the position of the UGV in the aerial map. This approach is also robust to appearance-based scene changes, small structural scene changes, and occasional ambiguous regions.

6.2 Future Work

For each contribution presented there are different possible future directions. There is also the opportunity to try and put all of these contributions together in a single system. For example, in the task of radiation search in an unknown environment, a UAV could gather image and radiation data that is used to plan paths for a UGV to visit points of interest. By simulating a GPS-denied environment (disallow the UGV to use GPS during the mission), the UGV could use the registration approach presented. The approach presented in Chapter 3 could be used for generating a semantic understanding of the scene. The system could also be extended to multiple UAV and UGV.

One direction for future work in joint reasoning about multiple perception modules would be to extend the approach to three or more modules. For two module experiments, success has already been demonstrated for experiments in language and vision [13], indoor scene understanding (RGBD segmentation and object support estimation) [19], and the urban scene understanding experiment presented. When extending to three or more modules, other modules of perception could be considered, such as object detection, monocular depth estimation, and surface normal estimation.

As part of future experiments extending the second contribution, where scene understanding was applied to radiation search operations, faster reconstruction methods should be used to develop 3D reconstructions of the scene. While software such as Agisoft provides detailed reconstructions of the scene, it is too slow for real-time response in emergency situations. For future experiments, there are many possibilities. One idea is to improve search strategies from the ground. If there are several clusters of interest identified by the aerial data, and there is a time budget for the UGV to inspect the scene, then path planning strategies could be developed and tested so that the optimal amount of time is spent within each cluster. Another direction would be to incorporate background radi-

ation models. One possibility would be to use imagery to predict background radiation levels and help better model the likelihood of a radiation source at a particular location. For example, if we can detect a certain type of building, or whether or not it has rained in the scene recently, then this knowledge could potentially be used to build better models.

For localization of a UGV in an unknown environment, there are opportunities to improve each of the components used in the approach, such as using better 3D reconstructions, better semantic segmentation models, and better descriptors and similarity scoring techniques. As for new directions of research, one idea would be to develop an active strategy to localization, where the UGV navigates to locations it believes will be better for performing localization.

There are also many interesting directions for incorporating natural language processing (NLP) for search tasks involving UAVs and UGVs. In [13], ambiguities in semantic segmentation and NLP were resolved simultaneously. This could be exteneded to robot instructions for real-world search missions. For example, an operator may provide a UGV the instruction "find the radiation source inside the building to the right of the store with the blue roof". Here there is an ambiguity as to whether or not it is the building with the source or the store that has the blue roof. The UGV must be able to understand that an ambiguity exists in case the most likely sentence parse of the instruction does not have prepositional attachments that represent reality. Another possibility involving NLP would be to generate paths for a UGV using operator instructions. If the operator provides the instruction "make a left around the corner and then wait", then a path has to be planned for the UGV to execute this instruction. This involves understanding what "corner" means in the given context of the scene. A dataset of planned paths and instructions could be collected so that a model can be trained to output paths given an instruction. When ambiguities arise, the UGV could ask questions to help clarify intentions. Having such a

system would allow for a more natural interaction between the human and the robot.

6.3 Closing Remarks

This dissertation has focused on a few different problems related to perception for autonomous unmanned vehicles. While the experiments presented use existing code for semantic segmentation and image-based 3D reconstructions that do not allow for real-time performance, the algorithms presented for the problems of interest do not rely on this code. As faster, more accurate, methods for semantic segmentation and 3D reconstruction become available, the algorithms presented within this dissertation can hopefully be extended or modified to better solve future perception tasks.

References

[1] Changchang Wu. Towards linear-time incremental structure from motion. In *3DV*, 2011. xii, 42, 50

[2] Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M Seitz. Multicore bundle adjustment. In *CVPR*, 2011. xii, 42, 50

[3] Yasutaka Furukawa and Jean Ponce. Accurate, Dense, and Robust Multiview Stereopsis. *PAMI*, 32(8), 2010. xii, xviii, 42, 49, 50

[4] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. LabelMe: A Database and Web-Based Tool for Image Annotation. *IJCV*, 77(1-3), 2008. xiv, 47, 70

[5] Lubor Ladicky. *Global Structured Models towards Scene Understanding*. PhD thesis, Oxford Brookes University, 2011. xiv, 47, 68, 70, 87, 88

[6] Lubor Ladicky and Philip H.S. Torr. The Automatic Labelling Environment. http://cms.brookes.ac.uk/staff/PhilipTorr/ale.htm. xviii, 3, 24

[7] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *ICRA*, 2011. xviii, 24, 45, 50

[8] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2), 2004. xviii, 44

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 88(2), 2010. 3, 11

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In *ICLR*, 2015. 3, 11, 47

[11] Georgios Floros and Bastian Leibe. Joint 2d-3d temporally consistent semantic segmentation of street scenes. In *CVPR*, 2012. 3, 8, 9, 11, 12, 19, 24, 26, 51

[12] Richard Zhang, Stefan A Candra, Kai Vetter, and Avideh Zakhor. Sensor Fusion for Semantic Segmentation of Urban Scenes. In *ICRA*, 2015. 3

[13] Gordon Christie, Ankit Laddha, Aishwarya Agrawal, Stanislaw Antol, Yash Goyal, Kevin Kochersberger, and Dhruv Batra. Resolving Language and Vision Ambiguities Together: Joint Segmentation & Prepositional Attachment Resolution in Captioned Scenes. In *EMNLP*, 2016. 5, 18, 19, 25, 27, 95, 96

[14] Talya Meltzer, Chen Yanover, and Yair Weiss. Globally Optimal Solutions for Energy Minimization in Stereo Vision Using Reweighted Belief Propagation. In *ICCV*, 2005. 17

[15] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors. *PAMI*, 30(6), 2008. 17

[16] Jörg H. Kappes, Bjoern Andres, Fred A. Hamprecht, Christoph Schnörr, Sebastian Nowozin, Dhruv Batra, Sungwoong Kim, Bernhard X. Kausler, Jan Lellmann, Nikos Komodakis, and Carsten Rother. A Comparative Study of Modern Inference Techniques for Discrete Energy Minimization Problems. In *CVPR*, 2013. 17

[17] Cyma Van Petten. Words and sentences: Event-related Brain Potential Measures. *Psychophysiology*, 32, 1994. 18

[18] Mohammad I. Khawalda and Emad M. Al-Saidat. Structural Ambiguity Interpretation: A Case Study of Arab Learners of English. *Global Journal of Human Social Science*, 2012. 18

[19] Gordon Christie, Ankit Laddha, Aishwarya Agrawal, Stanislaw Antol, Yash Goyal, Kevin Kochersberger, and Dhruv Batra. Holistic Scene Understanding via Multiple Structured Hypotheses from Perception Modules . *Submitted to Transactions on Image Processing*, Submitted 2016. 5, 19, 25, 27, 95

[20] Erik B. Sudderth, Antonio Torralba, William T. Freeman, and Alan S. Willsky. Learning Hierarchical Models of Scenes, Objects, and Parts. In *ICCV*, 2005. 7

[21] Jian Yao, Sanja Fidler, and Raquel Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012. 7

[22] Christian Wojek and Bernt Schiele. A dynamic conditional random field model for joint labeling of object and scene classes. In *ECCV*, 2008. 7

[23] Xavier Boix, Josep M. Gonfaus, Joost van de Weijer, Andrew D. Bagdanov, Joan Serrat Gual, and Jordi Gonzàlez. Harmony potentials - fusing global and local scale for semantic image segmentation. *IJCV*, 96(1):83–102, 2012. 7

[24] Congcong Li, Adarsh Kowdle, Ashutosh Saxena, and Tsuhan Chen. Towards holistic scene understanding: Feedback enabled cascaded classification models. In *NIPS*, 2010. 7

[25] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Closing the loop in scene interpretation. In *CVPR*, 2008. 7

[26] Wongun Choi, Yu-Wei Chao, Caroline Pantofaru, and Silvio Savarese. Understanding indoor scenes using 3d geometric phrases. In *CVPR*, 2013. 7

[27] Sanja Fidler, Abhishek Sharma, and Raquel Urtasun. A Sentence is Worth a Thousand Pixels. In *CVPR*, 2013. 7

[28] Chen Kong, Dahua Lin, Mohit Bansal, Raquel Urtasun, and Sanja Fidler. What are you talking about? Text-to-Image Coreference. In *CVPR*, 2014. 7

[29] Dahua Lin, Sanja Fidler, and Raquel Urtasun. Holistic Scene Understanding for 3D Object Detection with RGBD Cameras. In *ICCV*, 2013. 7

[30] David Bradley. *Learning In Modular Systems*. PhD thesis, Carnegie Mellon University, 2009. 7

[31] Geremy Heitz, Stephen Gould, Ashutosh Saxena, and Daphne Koller. Cascaded Classification Models: Combining Models for Holistic Scene Understanding. In *NIPS*, 2008. 7, 8, 23

[32] Chunhui Gu, Joseph J. Lim, Pablo Arbelaez, and Jitendra Malik. Recognition using regions. In *CVPR*, pages 1030–1037, 2009. 7

[33] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2009. 7

[34] Stephen Gould, Tianshi Gao, and Daphne Koller. Region-based segmentation and object detection. In *NIPS*, 2009. 7

[35] L'ubor Ladický, Paul Sturgess, Karteek Alahari, Chris Russell, and Philip H. S. Torr. What, where and how many? combining object detectors and crfs. In *ECCV*, 2010. 7

[36] T. Brox, L. Bourdev, S. Maji, and J. Malik. Object segmentation by alignment of poselet activations to image contours. In *CVPR*, 2011. 7

[37] Yoav Freund and Robert E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 1999. 8

[38] Michael I Jordan and Robert A Jacobs. Hierarchical Mixtures of Experts and the EM Algorithm. *Neural computation*, 6(2), 1994. 8

[39] L'ubor Ladicky, Paul Sturgess, Chris Russell, Sunando Sengupta, Yalin Bastanlar, William Clocksin, and Philip H.S. Torr. Joint optimisation for object class segmentation and dense stereo reconstruction. In *BMVC*, 2010. 8

[40] Michael Warren, D. McKinnon, H. He, and Ben Upcroft. Unaided stereo vision based pose estimation. In Gordon Wyeth and Ben Upcroft, editors, *Australasian Conference on Robotics and Automation*, Brisbane, 2010. Australian Robotics and Automation Association. 8

[41] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 8

[42] Daniel Munoz, J. Andrew Bagnell, and Martial Hebert. Co-inference machines for multi-modal scene analysis. In *ECCV*, 2012. 8, 9, 11, 12

[43] Abhijit Kundu, Yin Li, Frank Daellert, Fuxin Li, and James M. Rehg. Joint Semantic Segmentation and 3D Reconstruction from Monocular Video. In *ECCV*, 2014. 9, 12

[44] Timo Scharwächter, Markus Enzweiler, Uwe Franke, and Stefan Roth. Stixmantics: A medium-level model for real-time semantic scene understanding. In *ECCV*, 2014. 9

[45] Hayko Riemenschneider, András Bódis-Szomorú, Julien Weissenberg, and Luc Van Gool. Learning where to classify in multi-view semantic segmentation. In *ECCV*, 2014. 9

[46] Sunando Sengupta, Eric Greveson, Ali Shahrokni, and Philip H. S. Torr. Urban 3d semantic modelling using stereo vision. In *ICRA*, 2013. 9

[47] Wenqi Huang, Xiaojin Gong, and Zhiyu Xiang. Road scene segmentation via fusing camera and lidar data. In *ICRA*, 2014. 9

[48] Liang Huang and David Chiang. Better k-best Parsing. In *IWPT*, pages 53–64, 2005.

[49] Dhruv Batra, Payman Yadollahpour, Abner Guzman-Rivera, and Gregory Shakhnarovich. Diverse M-Best Solutions in Markov Random Fields. In *ECCV*, 2012. 20

[50] L. Ladickỳ, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical CRFs for object class image segmentation. *ICCV*, 2009. 24

[51] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient Large-Scale Stereo Matching. In *ACCV*, 2010. 24

[52] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>. 24, 49

[53] Vladimir Kolmogorov. Convergent Tree-Reweighted Message Passing for Energy Minimization. *PAMI*, 28(10), 2006. 24

[54] Kenneth D Jarman, Robert C Runkle, Kevin K Anderson, and David M Pfund. A Comparison of Simple Algorithms for Gamma-Ray Spectrometers in Radioactive Source Search Applications. *Applied Radiation and Isotopes*, 66(3), 2008. 32

[55] KK Anderson, KD Jarman, ML Mann, DM Pfund, and RC Runkle. Discriminating Nuclear Threats from Benign Sources in Gamma-Ray Spectra Using a Spectral Comparison Ratio Method. *Journal of Radioanalytical and Nuclear Chemistry*, 276(3), 2008. 32

[56] Kevin Kochersberger, Kenneth Kroeger, Bryan Krawiec, Eric Brewer, and Thomas Weber. Post-Disaster Remote Sensing and Sampling via an Autonomous Helicopter. *JFR*, 31(4), 2014. 12, 33, 64

[57] M. Brown and D. G. Lowe. Automatic Panoramic Image Stitching using Invariant Features. *IJCV*, 74(1), 2007. 33

[58] Harsh Agrawal, Clint Solomon Mathialagan, Yash Goyal, Neelima Chavali, Prakriti Banik, Akrit Mohapatra, Ahmed Osman, and Dhruv Batra. CloudCV: Large-Scale Distributed Computer Vision as a Cloud Service. In *Mobile Cloud Visual Media Computing*. Springer, 2015. 33

[59] Microsoft. Microsoft Image Composite Editor, 2015. 33

[60] Frank Dellaert. Factor graphs and GTSAM: A hands-on introduction, 2012. 33, 44

[61] Andreas Geiger, Julius Ziegler, and Christoph Stiller. StereoScan: Dense 3D Reconstruction in Real-time. In *IV*, 2011. 33

[62] Hernán Badino and Takeo Kanade. A Head-Wearable Short-Baseline Stereo System for the Simultaneous Estimation of Structure and Motion. In *IAPR MVA*, 2011. 33

[63] Bocheng Yu, Xiwang Dong, Zongying Shi, and Yisheng Zhong. Formation Control for Quadrotor Swarm Systems: Algorithms and Experiments. In *CCC*, 2013. 9

[64] Jesus Pestana, Jose Luis Sanchez-Lopez, Paloma de la Puente, Adrian Carrio, and Pascual Campoy. A Vision-based Quadrotor Swarm for the Participation in the 2013 International Micro Air Vehicle Competition. In *ICUAS*, 2014. 9

[65] Alex Kushleyev, Daniel Mellinger, Caitlin Powers, and Vijay Kumar. Towards a Swarm of Agile Micro Quadrotors. *Autonomous Robots*, 35(4), 2013. 9

[66] Xiwang Dong, Bocheng Yu, Zongying Shi, and Yisheng Zhong. Time-Varying Formation Control for Unmanned Aerial Vehicles: Theories and Applications. *Control Systems Technology*, 23(1), 2015. 9

[67] Bernd Brüggemann, Michael Brunner, Dirk Schulz, Maria Teresa Lazaro, Pablo Urcola, Luis Montano, Jose A Castellanos, Marco Cognetti, Paolo Stegagno, Antonio Franchi, et al. Outdoor Navigation with a Coordinated Multi-Robot System That Maintains Spatial Constraints. In *Multivehicle Systems*, volume 2, 2012. 9

[68] Heonyoung Lim, Yeonsik Kang, Jongwon Kim, and Changwhan Kim. Formation Control of Leader Following Unmanned Ground Vehicles Using Nonlinear Model Predictive Control. In *AIM*, 2009. 9

[69] David A Anisi, Petter Ögren, Xiaoming Hu, and Therese Lindskog. Cooperative Surveillance Missions with Multiple Unmanned Ground Vehicles (UGVs). In *CDC*. IEEE, 2008. 9

[70] Talib Hussain, David Montana, and Gordon Vidaver. Evolution-based Deliberative Planning for Cooperating Unmanned Ground Vehicles in a Dynamic Environment. In *GECCO*, 2004. 9

[71] Pedro Deusdado, Eduardo Pinto, Magno Guedes, Francisco Marques, Paulo Rodrigues, André Lourenço, Ricardo Mendonça, André Silva, Pedro Santana, José Corisco, et al. An Aerial-Ground Robotic Team for Systematic Soil and Biota Sampling in Estuarine Mudflats. In *Robot 2015: Second Iberian Robotics Conference*, 2016. 9

[72] Haibin Duan. Multiple UAV/UGV Heterogeneous Control. In *Bio-inspired Computation in Unmanned Aerial Vehicles*. Springer, 2014. 9

[73] Jin Hyo Kim, Ji-Wook Kwon, and Jiwon Seo. Multi-UAV-based Stereo Vision System Without GPS for Ground Obstacle Mapping to Assist Path Planning of UGV. *Electronics Letters*, 50(20), 2014. 9

[74] Pratap Tokekar, Joshua Vander Hook, David Mulla, and Volkan Isler. Sensor Planning for a Symbiotic UAV and UGV System for Precision Agriculture. In *IROS*, 2013. 9

[75] Carol Cheung and Benjamin Grocholsky. UAV-UGV Collaboration with a PackBot UGV and Raven SUAV for Pursuit and Tracking of a Dynamic Target. In *SPIE*, 2008. 9

[76] Connie Phan and Hugh HT Liu. A Cooperative UAV/UGV Platform for Wildfire Detection and Fighting. In *System Simulation and Scientific Computing*, pages 494–498. IEEE, 2008. 9

[77] Mario Garzón, João Valente, Juan Jesús Roldán, Leandro Cancar, Antonio Barrientos, and Jaime Del Cerro. A Multirobot System for Distributed Area Coverage and Signal Searching in Large Outdoor Scenarios. *JFR*, 2015. 9

[78] Herbert G Tanner. Switched UAV-UGV Cooperation Scheme for Target Detection. In *ICRA*, 2007. 9

[79] Ben Grocholsky, James Keller, Vijay Kumar, and George Pappas. Cooperative Air and Ground Surveillance. *Robotics & Automation Magazine*, 13(3), 2006. 9

[80] Elias Mueggler, Matthias Faessler, Flavio Fontana, and Davide Scaramuzza. Aerial-Guided Navigation of a Ground Robot Among Movable Obstacles. In *SSRR*, 2014. 9

[81] Nathan Michael, Shaojie Shen, Kartik Mohta, Yash Mulgaonkar, Vijay Kumar, Keiji Nagatani, Yoshito Okada, Seiga Kiribayashi, Kazuki Otake, Kazuya Yoshida, et al. Collaborative Mapping of an Earthquake-Damaged Building via Ground and Aerial Robots. *JFR*, 29(5), 2012. 10

[82] Martin Saska, Tomáš Krajník, and Libor Pfeucil. Cooperative μUAV-UGV Autonomous Indoor Surveillance. In *SSD*, pages 1–6. IEEE, 2012. 10

[83] Frank E Schneider, Dennis Wildermuth, and Hans-Ludwig Wolf. ELROB and EURATHLON: Improving search & Rescue Robotics Through Real-World Robot Competitions. In *RoMoCo*, 2015. 10

[84] Boris Sofman, Ellie Lin, J Andrew Bagnell, John Cole, Nicolas Vandapel, and Anthony Stentz. Improving Robot Navigation Through Self-Supervised Online Learning. *JFR*, 23(11-12), 2006. 10

[85] Guosheng Lin, Chunhua Shen, Ian D. Reid, and Anton van den Hengel. Efficient Piecewise Training of Deep Structured Models for Semantic Segmentation. *CoRR*, abs/1504.01013, 2015. 11

[86] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *ECCV*, 2012. 11

[87] Javier A Montoya-Zegarra, Jan D Wegner, L'ubor Ladickỳ, and Konrad Schindler. Mind the gap: Modeling local and global context in (road) networks. In *Pattern Recognition*. Springer, 2014. 11

[88] JA Montoya-Zegarra, JD Wegner, L Ladickỳ, and K Schindler. Semantic Segmentation of Aerial Images in Urban Areas with Class-Specific Higher-Order Cliques. *ISPRS*, 2(3), 2015. 11

[89] Scott Radford. Real-time Roadway Mapping and Ground Robotic Path Planning via Unmanned Aircraft. Master's thesis, Virginia Tech, 2014. 11, 46

[90] Xuehan Xiong, Daniel Munoz, J. Andrew (Drew) Bagnell, and Martial Hebert. 3-D Scene Analysis via Sequenced Predictions over Points and Regions. In *ICRA*, 2011. 11

[91] Joachim Niemeyer, Franz Rottensteiner, and Uwe Soergel. Contextual Classification of LiDAR Data and Building Object Detection in Urban Areas. *ISPRS*, 87, 2014. 11

[92] Paul Sturgess, Karteek Alahari, Lubor Ladicky, and Philip HS Torr. Combining Appearance and Structure from Motion Features for Road Scene Understanding. In *BMVC*, 2009. 11

[93] Sid Yingze Bao and Silvio Savarese. Semantic Structure from Motion. In *CVPR*, 2011. 12

[94] Naotaka Hatao, Satoshi Kagami, Ryo Hanai, Kimitoshi Yamazaki, and Masayuki Inaba. Construction of semantic maps for personal mobility robots in dynamic outdoor environments. In *FSR*, 2014. 12

[95] Ronald A. Martin, Ivan Rojas, Kevin Franke, and John D. Hedengren. Evolutionary View Planning for Optimized UAV Terrain Modeling in a Simulated Environment. *Remote Sensing*, 8(1), 2015. 12

[96] Kai Vetter, Dan Chivers, and Brian Quiter. Advanced Concepts in Multi-dimensional Radiation Detection and Imaging. In *Nuclear Threats and Security Challenges*. Springer, 2015. 13

[97] Frank E Schneider, Bastian Gaspers, Kari Peräjärvi, and Magnus Gårdestig. Possible Scenarios for Radiation Measurementsand Sampling Using Unmanned Systems: ERNCIP Thematic Group Radiological and Nuclear Threats to Critical Infrastructure Task 3 Deliverable 2, 2015. 13

[98] J Benedetto, A Cloninger, W Czaja, T Doster, K. Kochersberger, B. Manning, T. McCullough, and M. McLane. Operator Based Integration of Information in Multimodal Radiological Search Mission with Applications to Anomaly Detection. In *SPIE*, 2014. 13, 75

[99] Frank E Schneider, Jochen Welle, Dennis Wildermuth, and Markus Ducke. Unmanned Multi-Robot CBRNE Reconnaissance with Mobile Manipulation System Description and Technical Validation. In *ICCC*, 2012. 13

[100] R Andres Cortez, Xanthi Papageorgiou, Herbert G Tanner, Alexei V Klimenko, Konstantin N Borozdin, Ron Lumia, and William C Priedhorsky. Smart Radiation Sensor Management. *Robotics & Automation Magazine*, 15(3), 2008. 13

[101] Gaku Minamoto, Eijiro Takeuchi, and Satoshi Tadokoro. Estimation of Ground Surface Radiation Sources from Dose Map Measured by Moving Dosimeter and 3D Map. In *IROS*, 2014. 14

[102] Jerry Towler, Bryan Krawiec, and Kevin Kochersberger. Radiation Mapping in Post-Disaster Environments Using an Autonomous Helicopter. *Remote Sensing*, 4(7), 2012. 14, 64

[103] Eric Thomas Brewer. Autonomous Localization of 1/Rˆ 2 Sources Using an Aerial Platform. Master's thesis, Virginia Tech, 2009. 14

[104] Stereo Data Maker. SDM - for Creative Photography with Canon compact cameras, 2011. 38

[105] Adam Shoemaker and Alexander Leonessa. Bioinspired Tracking Control of High Speed Nonholonomic Ground Vehicles. *Journal of Robotics*, 2015. 38

[106] Agisoft LLC. Agisoft PhotoScan User Manual: Professional Edition, Version 1.0.0, 2013. 43

[107] Jean-Yves Bouguet. Camera Calibration Toolbox for Matlab, 2004. 43

[108] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc., 2008. 43, 44

[109] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *ECCV*, 2014. 44

[110] Raul Mur-Artal, JMM Montiel, and Juan D Tardós. ORB-SLAM: a Versatile and Accurate Monocular SLAM System. *Transactions on Robotics*, 31(5), 2015. 44, 85, 86

[111] Noah Snavely. Bundler: Structure from Motion (SfM) for Unordered Image Collections. Website - http://phototour.cs.washington.edu/bundler/. 45, 49, 50

[112] Christopher Zach. Robust Bundle Adjustment Revisited. In *ECCV*. 2014. 45

[113] Yucong Lin and Srikanth Saripalli. Road Detection from Aerial Imagery. In *ICRA*, 2012. 46

[114] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation. In *ECCV*, 2006. 48

[115] Gordon A Christie, L Justin Stiltner, Kenneth Kroeger, and Kevin Kochersberger. 3-D Scene Understanding from Image-based Reconstructions Using a Small Unmanned Aircraft. In *AIAA*, 2014. 49

[116] Gordon Christie, L. Justin Stiltner, Kevin Kochersberger, Morgan Mclean, and Wojtek Czaja. Synchronous Radiation Sensing and 3D Urban Mapping for Improved Source Identification. In *SPIE*, 2014. 49, 67

[117] Changchang Wu. VisualSFM : A Visual Structure from Motion System. Website - http://www.cs.washington.edu/homes/ccwu/vsfm/. 49

[118] Changchang Wu, Sameer Agarwal, Brian Curless, and Steven Seitz. Multicore Bundle Adjustment. In *CVPR*, 2011. 49

[119] Daniel Munoz, J Andrew Bagnell, Nicolas Vandapel, and Martial Hebert. Contextual Classification with Functional Max-Margin Markov Networks. In *CVPR*, 2009. 50

[120] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A Library for Support Vector Machines. *TIST*, 2(3), 2011. 51

[121] Yuri Boykov and Vladimir Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *PAMI*, 26(9), 2004. 51

[122] Gordon Christie, Haseeb Chaudhry, and Kevin Kochersberger. Aircraft Path Planning for Optimal Imaging Using Dynamic Cost Functions. In *SPIE*, 2015. 53, 54

[123] Gurobi Optimization, Inc. Gurobi Optimizer Reference Manual, 2015. 55

[124] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *Transactions on Systems Science and Cybernetics*, 4(2), 1968. 56

[125] Edsger W Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische mathematik*, 1(1), 1959. 56

[126] Michael Osborne. Mission Planner, 2014. 61

[127] Aude Oliva and Antonio Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *IJCV*, 42(3), 2001. 67

[128] Claudia Hauff and Geert-Jan Houben. Geo-Location Estimation of Flickr Images: Social Web Based Enrichment. In *ECIR*, 2012. 14

[129] Pavel Serdyukov, Vanessa Murdock, and Roelof Van Zwol. Placing Flickr Photos on a Map. In *SIGIR*, 2009. 14

[130] Olivier Van Laere, Steven Schockaert, and Bart Dhoedt. Combining Multi-Resolution Evidence for Georeferencing Flickr Images. In *SUM*, 2010. 14

[131] Scott Workman, Richard Souvenir, and Nathan Jacobs. Wide-Area Image Geolocalization With Aerial Reference Imagery. In *ICCV*, 2015. 14

[132] Stefan Lee, Haipeng Zhang, and David J Crandall. Predicting geo-informative attributes in large-scale image collections using convolutional neural networks. In *WACV*, 2015. 14

[133] Tobias Weyand, Ilya Kostrikov, and James Philbin. PlaNet - Photo Geolocation with Convolutional Neural Networks. In *ECCV*, 2016. 14

[134] Till Kroeger and Luc Van Gool. Video Registration to SfM Models. In *ECCV*, 2014. 15

[135] Philip David and Sean Ho. Orientation Descriptors for Localization in Urban Environments. In *IROS*, 2011. 15

[136] Marcus Brubaker, Andreas Geiger, and Raquel Urtasun. Lost! Leveraging the Crowd for Probabilistic Visual Self-Localization. In *CVPR*, 2013. 15

[137] Marcus A Brubaker, Andreas Geiger, and Raquel Urtasun. Map-based Probabilistic Visual Self-Localization. *PAMI*, 38(4), 2016. 15

[138] Anirudh Viswanathan, Bernardo R Pires, and Daniel Huber. Vision Based Robot Localization by Ground to Satellite Matching in GPS-denied Situations. In *IROS*, 2014. 15

[139] Jason Gregory, Jonathan Fink, Ethan Stump, Jeffrey Twigg, John Rogers, David Baran, Nicholas Fung, and Stuart Young. Application of Multi-Robot Systems to Disaster-Relief Scenarios with Limited Communication. *FSR*, 2016. 84

[140] Clearpath Robotics Husky. http://www.clearpathrobotics.com/husky/. Accessed: 2016-07-18. 86