

Regular Article

Using perception cues for context-aware navigation in dynamic outdoor environments

Maggie Wigness¹, John G. Rogers III¹, Chieh-En Tsai², Christoph Mertz²,
Luis Navarro-Serment² and Jean Oh²

¹DEVCOM Army Research Laboratory, Adelphi, Maryland, USA

²The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA

Abstract: Continued advancements in robot autonomy have allowed the research community to shift from using robots as tools in the field to deploying robot teammates capable of learning, reasoning, and executing tasks. Autonomous navigation is one necessary capability of a robot teammate that must operate in large field environments. In relatively static environments a simple navigation solution such as obstacle avoidance along the shortest path may suffice; however, as robot teammates are deployed to highly dynamic environments with changing mission requirements, additional environment context may be necessary to ensure safe and reliable navigation. Although recent works in urban autonomous driving have advanced the state-of-the-art in context-aware decision making, the spectrum of behaviors deployed for context-switching is more narrowly focused (by defining constraints specific to operation in structured environments) than what might be required for human-agent teaming field missions. As such, establishing a context-aware intelligent system for dynamic, unstructured environments is still an open problem. We discuss our approach to the integration of several context-aware navigation behaviors on a small unmanned ground vehicle (UGV) and a perception stack that provides cues used to transition between these different learned behaviors. Specifically, we integrate socially compliant, terrain-aware, and covert behaviors in an outdoor navigation scenario where the UGV encounters moving pedestrians, different terrains, and weapon threats. We provide a detailed account of the overall system integration, experiment design, component- and system-level analysis, and lessons learned.

Keywords: context-aware navigation, behavior learning, tactical object detection

1. Introduction

Robotic platforms are being fielded to support a number of real-world applications, ranging from agriculture (Bakken et al., 2019; Kushwaha et al., 2016; Shamshiri et al., 2018) to humanitarian assistance and disaster response (Delmerico et al., 2019; Murphy, 2014; Nagatani et al., 2013). In many

This work was performed while Chieh-En Tsai was a graduate student at CMU; he is currently affiliated with Waymo.

Received: 14 September 2020; revised: 22 January 2021; accepted: 12 March 2021; published: 19 October 2021.

Correspondence: Maggie Wigness, DEVCOM Army Research Laboratory, Adelphi, MD 20783, USA,
Email: maggie.b.wigness.civ@army.mil

This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © 2021 Wigness, Rogers, Tsai, Mertz, Navarro-Serment and Oh

of these cases, the robotic asset can be leveraged as a tool to complete a tedious or dangerous task that would otherwise require a human. However, the continued advancements in robotic autonomy and intelligence provide the necessary components to begin fielding robots in more complex, dynamic, and unstructured scenarios, where the robot can be treated as less of a tool, and more as a teammate capable of executing high-level tasks and missions.

Building high-level, context-aware intelligence for enabling human-robot teams to perform complex tasks is one of the remaining challenges in AI (Schaefer et al., 2019). Consider a scenario where a robot is performing a screening or surveillance task in a crowded place such as a street market. The robot navigates naturally among pedestrians on its route, but monitors the surrounding environment to be ready to change to a covert posture when necessary. Performing such a task is nontrivial and requires the robot to exhibit high-level intelligence. Specifically, the robot teammate needs to understand the operating context by recognizing objects and scenes of its environment as well as understanding various spatial and social relationships between objects and pedestrians in the environment. Such semantic understanding and reasoning capabilities enable the robot’s situational awareness and its grounding of a task in the context of its current environment state.

As seen in the previous example, deliberate maneuver behavior is essential for a robot teammate to successfully execute tasks in a diverse and dynamic operating environment. In relatively simple scenarios, the ability to identify obstacles and navigate around them is sufficient for the robot to maintain its safety, but additional scene complexity would make this strategy appear insufficient or uninformed. This additional complexity could be observed at the environment level, e.g., dynamic objects, or at the mission level, e.g., dynamic threats. Significant research advancements have been made for context-aware autonomous navigation in the presence of dynamic objects. This body of work has been exemplified in social navigation (Fridovich-Keil et al., 2020; Trautman et al., 2015; Tsai & Oh, 2020; Turnwald & Wollherr, 2018) and urban autonomous driving (Codevilla et al., 2018; Sadat et al., 2020; Taş et al., 2017) applications, where dynamic obstacles are primarily pedestrians and vehicles. In these works, navigation behavior can be constrained given the structure of the environment and task at hand. That is, the robot’s behavior is designed to follow social norms or rules of the road.

Our motivating scenario is unique in that the autonomous agent is deployed in an environment as a teammate with the expectation that its maneuver behavior should be representative of what one of its human teammates might exhibit. Each dimension of complexity, i.e., environment or mission, introduces unique rules and actions, requiring specific navigation behaviors that an intelligent system must address to ensure safe, efficient and effective navigation. In that sense, the range of behaviors we would like our agent to exhibit is more diverse than current individual examples in social and urban navigation. We describe our approach to leverage multiple traversal behaviors to address the need for operation in complex and dynamic environments. In this system, three separate learned traversal behaviors are deployed to assist in scenarios that require 1) terrain-aware navigation, 2) socially compliant navigation, and 3) navigation in the presence of threats. We use vision and LiDAR-based perception cues to identify the operational context, and this information allows the intelligent agent to make decisions about when to transition between the different navigation behaviors. Figure 1 provides an illustrative example of how a robot would transition between these behaviors during operation as the scene changes throughout the mission.

This paper focuses on the system integration of several navigation and perception components needed to field a mobile robot capable of intelligent mobility in dynamic environments. This system integration was performed as part of the Robotics Collaborative Technology Alliance (RCTA), a U.S. Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory (ARL) funded program. We discuss how this system integration is different from other systems previously seen in the literature. We also describe the field experimentation scenario that was designed to test this capability for the RCTA Capstone event, and provide an analysis of our described system in this operational context. Finally, developing a system with a diverse set of behavior requirements presents many challenges, and one major contribution of this work is an extensive discussion on



Figure 1. Illustrative example of a robot transitioning between traversal behaviors based on cues received from the perception stack. The robot proceeds from its start location in the upper left to the goal location denoted by a star in the road intersection on the right. Along its path the robot will encounter several perception cues that affect its navigation behavior. The robot starts with its default terrain-aware behavior, shown in green. When the robot encounters a group of pedestrians, a socially compliant navigation behavior shown in blue is used. Finally, when the robot detects people armed with rifles, a covert navigation behavior shown in red is used to avoid the threat.

lessons learned when integrating a system of this nature. This discussion identifies open areas of research we see as essential to continue to advance intelligence and autonomy for mobile robots.

2. Related systems

Robotic navigation systems have an extensive history, and current advancements are being made at a rapid pace. In the rest of this section we discuss related navigation systems and context-aware intelligence architectures. Specifically, we focus on approaches that make use of visual perception, learning and that are specifically designed for navigation in dynamic environments. We also provide a brief history of programs that resulted in significant system integration for challenge problems, many of which focused on off-road and unstructured environments.

Other related research in perception and learning is described in detail throughout Section 3 during our description of the individual components that make up our integrated context-aware navigation system.

2.1. Navigation

Significant research efforts have been made to advance autonomous navigation. A full discussion and history of these advancements is outside the scope of this paper, but we refer the reader to several survey papers (Bonin-Font et al., 2008; Desouza & Kak, 2002; Gul et al., 2019; Pandey et al., 2017) that outline a number of different approaches that have been developed throughout the years. Instead, we provide a more thorough discussion on particular approaches used for dynamic obstacle avoidance and current learning-based trends for autonomous navigation.

Navigation in the presence of dynamic obstacles has been addressed with a number of approaches. The use of neural networks and particle swarm optimization (Pandey et al., 2020) was used to demonstrate navigation amongst a mixture of static and dynamic obstacles. Other approaches have used reinforcement learning (L. Huang et al., 2018) or fuzzy logic (Nasrinahar & Chuah, 2018; Pandey et al., 2019) to avoid dynamic obstacles. A self-awareness model, based on the interdependence of subcomponents in a navigation system, has also been deployed to adapt to current driving conditions including dynamic obstacles (Taş et al., 2017).

Certain dynamic obstacles move in deliberate ways. People are one such example and have been well-studied in social navigation for robots. Early works treated humans as dynamic obstacles that are not responsive nor interactive, which resulted in several issues including improper behavior, e.g., blocking the paths of pedestrians (Ziebart et al., 2009), unanticipated motions known as “reciprocal dance” where a robot and a human alternate in taking unpredictable actions (Feurtey, 2000), or the “freezing robot” problem where a robot gives up too quickly due to overestimated uncertainty in crowd actions (Trautman et al., 2015). To address these issues, recent works model the interaction between agents either explicitly (Fridovich-Keil et al., 2020; C. Mavrogiannis et al., 2019; C. I. Mavrogiannis & Knepper, 2018; Turnwald & Wollherr, 2018) or implicitly (C. Chen et al., 2019; Y. F. Chen, Everett, et al., 2017; Y. F. Chen, Liu, et al., 2017; Everett et al., 2018, 2021; Tsai & Oh, 2020).

End-to-end learning for navigation has become a popular approach to avoid the design of sub-components (e.g., dynamic obstacle modeling) and the relationships between these components in a navigation system. Approaches using only geometric data have been shown to exhibit goal-seeking (Pfeiffer et al., 2017; Tai et al., 2017; Zhang et al., 2017) and exploration-based (Zhang et al., 2020; Zhelo et al., 2018) navigation behaviors. With the addition of visual sensor information, e.g., RGB imagery, systems have been generated that allow a robot to accept high-level semantic goal specifications (Zhu et al., 2017), and that improve upon general obstacle avoidance, goal-seeking navigation (Tai, Li, et al., 2016; Zhou et al., 2019). Visual information has also helped advance learning of higher-level navigation behaviors such as urban on-road driving (Bojarski et al., 2016; Z. Chen & Huang, 2017; Codevilla et al., 2018; Sadat et al., 2020) and terrain adaptation (Siva et al., 2019). We refer the reader to surveys on learning-based motion control and navigation (Tai, Zhang, et al., 2016; Xiao et al., 2020) for additional works and discussion in this area.

The existing work we have outlined can in most cases be categorized as producing robot behavior that is obstacle-avoiding and goal-seeking. Exceptions such as terrain adaptation result in low-level behavior changes only. This leaves the body of research on urban and social navigation as some of the systems that most resemble ours. However, as outlined in our motivation, the high-level behaviors deployed by these systems are much more constrained than the system we integrate and evaluate in this paper.

2.2. Navigation challenge problems and legacy programs

Several large-scale, multi-year programs, such as the Defense Advanced Research Projects Agency (DARPA) Unmanned Ground Combat Vehicle (UGCV), PerceptOR, UGCV-Perception for Off-Road Robots Integration (UPI), and the DARPA Grand Challenge and Urban Challenge have focused on research and development of autonomous robotic systems. Most closely aligned to the work we present in this paper is the DARPA Learning Applied to Ground Robots (LAGR) program (Jackel et al., 2006), which focused on development of perception-based and learning-capable systems for off-road navigation. That program existed during a time when image processing was beginning to become faster and more sophisticated, making visual perception a novel tool for autonomous navigation. This resulted in systems that looked at stereo vision (Bajracharya et al., 2009; Konolige et al., 2008) and reverse optical flow (Lookingbill et al., 2007) to gather terrain context at longer distances and at different scales. Another approach included learning controls from demonstration or self-supervision (Albus et al., 2006) that allowed the robotic system to adapt to changing terrain. At the time of the LAGR program these systems were state of the art in identifying traversable terrain to autonomously navigate to defined goal locations.

In an earlier phase of the RCTA program, costmaps were used to couple perception and planning in a complex unstructured environment, such that perceived terrain features could be translated in terms of numeric costs (Ratliff et al., 2006; Ratliff et al., 2009; Silver et al., 2010). Learning approaches were used to train a model using a set of human expert demonstrations in order to generate a traversability costmap given an image of the environment. They demonstrated that a robotic vehicle could autonomously navigate for several kilometers in unknown, unstructured outdoor environments. While significant, semantic understanding in this work did not expand much beyond differentiating traversable and non-traversable terrain, similar to what was seen in the LAGR program.

Since the previously described works, onboard computation, visual perception and behavior learning have demonstrated great advancements. Although previous systems share commonalities with our presented system, e.g., behavior learning, perception advances allow our system to make use of finer-grained terrain categorization to support richer understanding of semantic meanings of both environmental and social contexts. Further, we are able to introduce operating scenarios that go beyond navigation in static scenes. Namely, we are able to produce a navigation stack that makes use of perception, that safely and autonomously navigates among dynamic obstacles, and that can begin to reason about higher level behaviors based on contextual cues.

2.3. Context-aware intelligence architectures

In previous RCTA work, a robot intelligence architecture was comprised of the mission, behavior, and interaction levels (Oh et al., 2015). First, the mission-level intelligence analyzes a context and makes cognitive-level decisions based on abstract information. Next, the behavior-level intelligence governs decision making for a specific task, such as navigation or monitoring, that requires a tight integration between perception and actions. Finally, the interaction level is directly connected to the actual control of a robotic platform and presents the least amount of uncertainty within the model. In this architecture, information flows both top-down and bottom-up, supporting not only the traditional model of choosing an action based on perceived information, but also using knowledge available at the mission level to guide lower-level perception.

Using the previously described intelligence architecture, behaviors are specified by a human commander through a natural language command, e.g., “Navigate covertly to the back of the building,” rather than based on the robot’s own situational awareness as in the work presented in this article. To ground a given command to the current environmental context, semantic perception is used (Boularias et al., 2015). Prior knowledge from the cognitive level is also used for a robot to hypothesize about the unseen part of an environment. This architecture was fully integrated on a Clearpath Husky robot platform equipped with a camera and LiDAR sensors, and evaluated in multiple outdoor, semi-structured environments for its semantic navigation and screening mission performances (Oh et al., 2017).

Additional human supervision has been demonstrated as an alternative over high-level intelligence architectures in Gregory et al. (2016). In this work, the goal of the multi-robot and human commander team was to find a hidden radio transmitter in a military training facility. The human commander oversaw the entire mission by defining waypoints for navigation instead of allowing the robot to determine where to traverse next given its understanding of the current environment and mission state. Further, the commander would instruct the robots to capture images based on an accumulated radio signal strength model. In this scenario, all decision making was supervised by the human commander.

In this paper, we demonstrate a bottom-up approach where semantic perception serves as a cue for a robot to determine the type of behavior in addition to grounding commands. In the market surveillance example, semantic perception is used to keep the robot vigilant to the dynamic changes happening in an environment so that a visual cue can trigger a change at the behavior or mission level. This bottom-up approach provides flexibility in the robot’s action choices and allows the robot to react more robustly against dynamic changes and complex environments when compared to strictly

following directions. At the same time, increased autonomy can alleviate a human commander from the need for constantly watching and directing the robot.

3. System integration for context-aware navigation

The supporting software components of our robotic system constitute the autonomy architecture. Briefly, these components consist of a mapping and localization system, a perception system, and a navigation system. Figure 2 provides a high-level overview of this architecture, where raw sensor data is processed by the mapping and perception stacks to produce representations of the current environment context. This context is used as input to the navigation behaviors and also to the behavior executor to determine when to transition between these navigation modules. Additional details of specific components are provided in the remainder of this section.

3.1. Mapping

The mapping system of our autonomy architecture is based on the *OmniMapper* described in Trevor et al. (2014). It builds a factor graph of relative pose measurements based upon point cloud alignments determined via generalized iterative closest point (G-ICP) (Segal et al., 2009) of subsequent locations along the robot’s trajectory and loop-closure constraints when locations are revisited. LiDAR measurements along the robot’s trajectory are rendered into an occupancy grid in a global frame of reference. LiDAR scan points are classified as ground strikes if they are within a small threshold of the lowest z height in a vertical column of space and are classified as obstacle hits otherwise. These measurements are used as evidence to update a log-odds occupancy grid.

The output of this process is a map that indicates the location of obstacles or other navigational hazards. On its own, this map produces information for obstacle-avoidance navigation. In our system, this obstacle map is used in conjunction with semantic information as input to our inverse optimal control (IOC) behavior learning technique. The two learned behavior models that use this input are discussed in Sections 3.3.2 and 3.3.3.

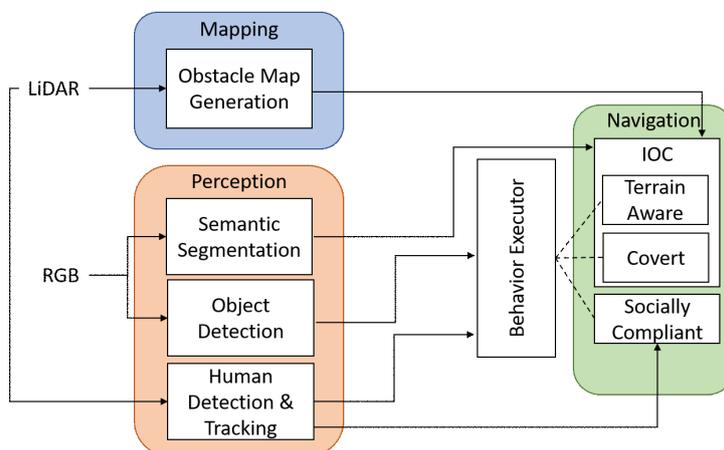


Figure 2. High-level overview of the autonomy architecture used in this paper. Raw sensor input is processed by the mapping and perception stacks to provide semantic context to the individual navigation behaviors. The perception stack also provides detection input to the behavior executor, which is used to determine when to trigger navigation behavior transitions based on the current environment context.

3.2. Perception

The perception stack in the autonomy architecture is designed to provide information about the environment from LiDAR and RGB camera data. We describe the perception modules used in our system to detect and localize, identify, and track (when applicable) objects of interest, terrain, and people. The perception stack provides both the static environment context and the current state of the dynamic scene, which serves as 1) input to the system’s behavior executor to determine navigation behavior transitions and 2) features for the IOC navigation behaviors. Both of these components are discussed in further detail in Section 3.3.

The accuracy of the perception stack plays a significant role in the robot’s decision making. If information about the environment is misclassified, the robot will fail to select the most appropriate traversal behavior, which can ultimately yield sub-optimal operation with respect to safety and mission requirements. In addition to providing a general overview of the perception modules used in this system, we also discuss the robustness of these modules, with respect to new environments, and techniques used to provide the most reliable models in anticipation of domain shift.

3.2.1. Human detection and tracking

The inclusion of people¹ in a scene makes an environment highly dynamic. Detecting and tracking these humans is necessary for a robot to navigate in a socially compliant manner (discussed in Section 3.3.1), as it provides location and trajectory information about people along the robot’s path. We leverage work from previous years of the RCTA to implement a LiDAR-based pedestrian detection and tracking system (Navarro-Serment et al., 2010) that is suited for operation within the computing and sensing budgets of our experimental robot (Mertz et al., 2013). The system is built around a Velodyne VLP-16 LiDAR with a 360° horizontal field of view (FOV) and 30° vertical FOV that provides full coverage of the scene surrounding the robot. This sensor generates distance measurements at 5 Hz to 20 Hz, which is suitable for accurate estimation of location and velocity of pedestrians on foot. The detection and tracking system operates in real time on an Intel Core i7 CPU, producing tracking information at 25 Hz. A sample image illustrating the tracker in action is shown in Figure 3.

We now present a general description of the algorithm we use for pedestrian detection and tracking. Specifically, we discuss how LiDAR data is filtered, segmented, tracked, and classified to provide the system with pedestrian motion information. Additional details can be found in (Navarro-Serment et al., 2010).

First, LiDAR data from a single scan cycle is filtered based on distance to the sensor (we only process measurements close enough to have an impact on social navigation), and it is then voxelized to further reduce the number of points to process. Points located at a certain height above ground are isolated to prevent processing ground returns that may be interpreted as false apparent motion. We also estimate the ground elevation of each cell to allow the system to adapt to different environments. The ground elevation is stored in a scrolling grid that contains accumulated LiDAR points and is centered at the robot’s current position. The mean and standard deviation of the heights of all points inside each cell are computed recursively, and the cell elevation is calculated by subtracting one standard deviation from the mean. This results in elevation that is never below the lowest point while still having about 80% of points above ground. Finally, to increase robustness against sensor misalignment, a decay factor is applied to the height measurements, giving more recent points greater weight on these statistics.

Next, the filtered set of non-ground points goes through the processes of object segmentation. The segmentation step groups together points believed to belong to the same object. We use a distance threshold segmentation process, which clusters points that fall within a fixed distance from each other. Each segmented object is represented by the center of the bounding voxel enclosing its respective points. A segmented object’s size is represented by the length of the diagonal of its bounding voxel.

¹ We use the terms people, humans, and pedestrians interchangeably throughout this paper.

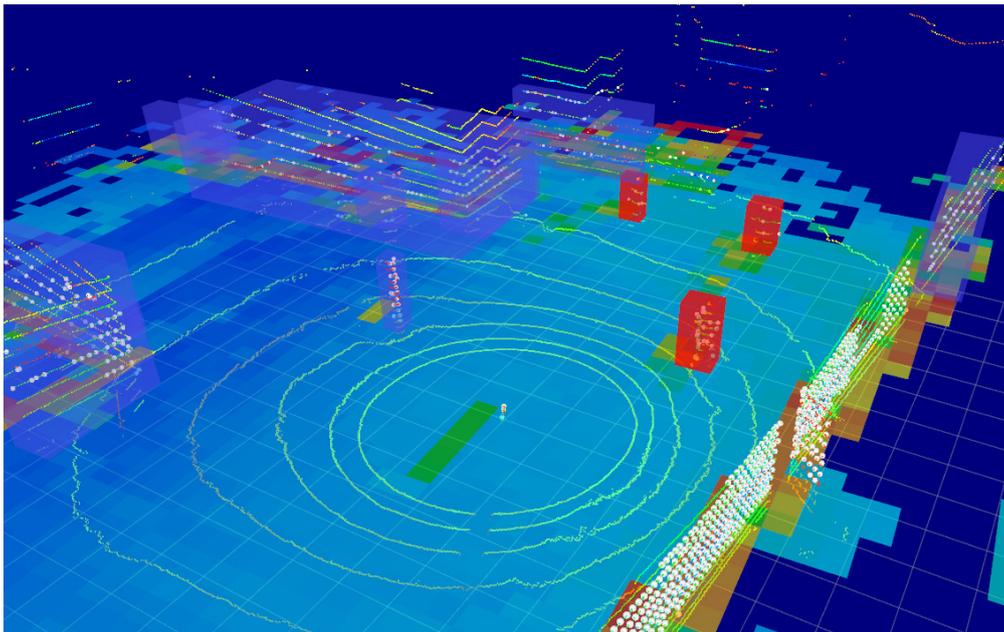


Figure 3. Visualization of human detection and tracking component. This example shows the raw VLP-16 LiDAR point cloud, three pedestrians indicated as red cuboids, several static objects indicated as blue cuboids, the ground elevation map indicated as a colored costmap (blue is low and red is high), and the filtered point cloud indicated by the white markers, showing the measurements from the ground plane that were removed.

In the object tracking step, objects detected in the current scan are tested for association with objects from previous scans. If the bounding voxels of two objects (i.e., one from the current and the other from the previous scan) overlap, they are identified as the same object. Voxels are considered to be overlapping if they have at least one measurement point in common, taking into account predicted motion path. Lines and corners are fit to the points, effectively becoming the feature points that are fed into a Kalman filter, which estimates the position and velocity of the object. These estimates are stored and made available during the classification stage to determine if an object is a pedestrian. The tracking algorithm is designed to track objects through temporary loss. If an object disappears, e.g., because it becomes occluded or leaves the field of view, its track is maintained for several cycles and its position is updated according to its last motion estimate. If the object re-emerges and its current position is close to the estimated position, the object is associated with the old track.

In addition to estimates of position and velocity, each LiDAR segmented track contains a predicted object class and associated *strength of detection* (SOD) measure, which represents how strongly the tracker believes the prediction is correct. Object predictions come from one of four classes: *unclassified*, *unknown-small*, *unknown-big*, and *pedestrian*. Three factors are used to calculate an object’s SOD: its size, the distance it has traveled, and the time during which the object has been moving. These factors are particularly useful to help distinguish moving people in the scene. Size distinguishes people from large objects, such as cars and walls. Distance traveled discriminates against stationary objects, such as barrels and posts. Finally, objects that have been perceived as moving for long durations are very likely people. This evidence is used to increase the object’s SOD, where scores are on the range of $[0, 1]$.

3.2.2. Object detection

Objects detected in an operating environment can provide context that allows a robot to reason about its current location, and where and how it may want to maneuver in the future. For our system,

we use object detection to identify potential threats, e.g., weapons, in the scene. Threat detection is used as a perception cue to transition into a covert navigation mode (discussed in Section 3.3.3).

Research in object detection has made great advances, and many popular neural network approaches exist for this task including SSD (Liu et al., 2016), YOLO (Redmon & Farhadi, 2017), R-FCN (Dai et al., 2016), and Faster-RCNN (Ren et al., 2017). These can be run with different backbones, and their trade-offs in speed, accuracy, memory footprint, and other qualities have been analyzed in previous work (J. Huang et al., 2017; Soria et al., 2020). A more specific analysis was performed, focusing on object detection for autonomous vehicles in urban environments (Carranza-García et al., 2020). Although a number of these architectures could be leveraged for our threat detection problem, the detector we trained for our system uses a standard Faster-RCNN neural net implementation (Massa & Girshick, 2018) with the provided pre-trained ResNet-50 as its backbone. This architecture and backbone gave us good performance with per-frame detections occurring at about 5 Hz.

As with most deep learning approaches, to properly train these detectors, one needs to have on the order of thousands or hundreds of thousands of labeled images. This annotation process is tedious and labor intensive. To collect a comprehensive dataset for our rifle model, the Husky robot (discussed in Section 4) was teleoperated around the testing environment to collect images from its onboard camera. The data collection was performed across several days under different weather conditions and at different times of the day. People with and without mock rifles² (seen in Figure 4) were part of the scene.

The bounding box annotation of objects in these images was outsourced to Appen,³ and internal verification was performed on each labeled image to ensure high-quality training data. Overall the dataset was made up of about 150,000 images with bounding box annotations for many classes used for a general perception pipeline (Narayanan et al., 2020). We only use the $\sim 15,000$ rifle instances from this data. This labeling approach took several months and worked well to establish a large dataset for research, but it has severe practical limitations. Namely, it requires significant time, human resources, and full access to the environment to capture objects in context. This process could be infeasible for real-world applications where robots may need to be deployed with little notice.

For situations when large sets of training data are not readily available, approaches such as few-shot learning (Y. Wang et al., 2020) or meta-learning (Y.-X. Wang et al., 2019) have emerged to reduce the number of labeled images required at training time. A secondary approach is to create images synthetically either by cutting objects out of one image and pasting them on a different background (Dwibedi et al., 2017) or by learning to generate via compositing (Tripathi et al., 2019). In this work, we assume that we have access to either the physical object or a CAD model of the object that needs to be detected. However, we address the challenge of not having adequate time to collect large sets of real image data with the object, or access to the relevant environment in which the object may be scene during deployment. Specifically, we discuss the creation of synthetic rifle data to train a detection system for applications with limited real world data resources.



Figure 4. Example images of mock rifles used during training data collection and field experimentation.

² All rifles used for data collection and experimentation, seen in the images throughout this paper, are fake.

³ <https://appen.com/>

We generate the synthetic imagery by pasting cutouts of objects onto various backgrounds. We used three different methods to produce the cutouts. The first is described in Dwibedi et al. (2017), and begins by taking pictures of the rifles from different viewpoints. Two examples are shown in Figure 4. The rifles in these images are cut out with OpenCV GrabCut. The second method creates a 3D mesh model. To produce this 3D model we used Colmap (Schönberger & Frahm, 2016) for the dense point cloud and Meshlab to clean the point cloud and construct the final mesh. The third method is to directly use a 3D CAD model of the object. Once the 3D mesh or CAD models are generated, virtual snapshots are captured from all viewpoints. Figure 5 shows an example snapshot from the CAD model (left), and the final 3D mesh model of the rifle (right). In addition to adding the rifles to the synthetic imagery, we also added distractor objects as seen in Figure 6. Overall we created about 100,000 synthetic training images with these methods.

During our navigation field experiments (discussed in Section 4.3), we deploy an object detection model trained with both the hand-labeled annotations and synthetic training data. We also provide a quantitative analysis of models trained separately with the two types of data. We provide this analysis to demonstrate the potential for learning models with only synthetic imagery. It is important to note that very often rifles are occluded by body parts, e.g., by the hands that carry it. We therefore provide an evaluation for normal and difficult (i.e., partially occluded) situations separately using a



Figure 5. CAD model (left) and 3D mesh model (right) of a rifle.



Figure 6. Example synthetic training image using an operationally relevant background and pasting a rifle and distractor object on top.

Table 1. AP for normal and difficult rifle scenarios from models trained with hand-labeled images and synthetic images.

Model training data	Normal scenarios	Difficult scenarios
Hand labeled	78.7%	33.2%
Synthesized/augmented	53.5%	16.8%

disjoint test set. Table 1 compares the average precision (AP) for the two models under the normal and difficult (i.e., with significant occlusions) rifle scenarios. The hand-labeled model performs quite well with an AP = 78.7%. Although the model trained with synthetic data yields lower performance, AP = 53.5%, this is still a reasonable result given it was achieved without any human annotation effort on operational data. It is not surprising that the models perform worse in difficult scenarios. The low performance of the model trained with synthetic data, AP = 16.8%, can be attributed to the fact that our augmentation model does not produce realistic occlusions. Specifically, it does not place rifles in the same context (i.e., held by a person), which results in significant occlusion by parts of the body. In future research we want to simulate moving people carrying rifles in realistic postures.

3.2.3. Semantic segmentation

The traversal behaviors discussed in Sections 3.3.2 and 3.3.3 are learned with respect to terrain and object features that are identified in the environment. To generate these feature maps, we run a neural network that performs semantic segmentation on images collected from the robot’s onboard camera sensor. The trained network is a fully convolutional network (FCN) with a skip architecture and VGG16 (Simonyan & Zisserman, 2015) network structure for the initial input layers. Convolutional layers 6 and 7 in VGG16 are replaced with layers that up-sample and combine data from three different layers from FCN8 (J. Long et al., 2015).

The FCN network weights were initialized using a pre-trained model for Pascal VOC (Everingham et al., 2010). Fine-tuning was performed using outdoor data collected from a MOUT site (Lennon et al., 2013) and was supplemented with some additional outdoor data. The ontology for this learned model comprises twelve classes: *sky*, *tree*, *asphalt*, *grass*, *building*, *object*, *concrete*, *gravel*, *fire hydrant*, *traffic barrel*, *vehicle*, and *pole*.

At inference time, the FCN is run onboard the robot using a Nvidia GTX 1060 GPU. Images coming from the onboard camera sensor are downsampled to a resolution of 344×275 and run through the network. The GPU rate of inference for these images is ~ 7 Hz.

The output of the semantic segmentation is used to generate features for both learning and inference of certain traversal behaviors, and the described FCN works reliably in both the environment where training data was collected and in novel environments. Figure 7 shows example segmentation results from the network in the training environment (top row) and the experimental testing environment (bottom row). Notice that the boundaries between various terrain types and objects are extremely clean within the training environment. Qualitatively, limited performance degradation is seen in the operating environment, which provides segmentation results that are of sufficient quality to successfully use the output as semantic feature maps to learn traversal behaviors.

This claim is reinforced by the quantitative evaluation we performed. Figure 8 shows the pixel-wise accuracy of classes when using the trained network in the testing environment. A single video sequence was captured and 137 frames were annotated with ground truth for this evaluation. The results show close to 90% accuracy across all pixels and particularly good results across the terrain and major objects in the scene. The average (i.e., mean per-class accuracy) is pulled down by the *pole* and *object* class results. The lower performance across these classes can be attributed largely to the scarcity (i.e., few instances) and size (i.e., minimal pixel coverage) in the training data. Further, the general notion of “object” actually encompasses multiple types of objects, each of which may have different feature characteristics, making it a more difficult concept to learn. However, for the presented system,



Figure 7. Example semantic segmentation results on images captured in the environment used to train the FCN (top row) and in the experimental operating environment (bottom row).

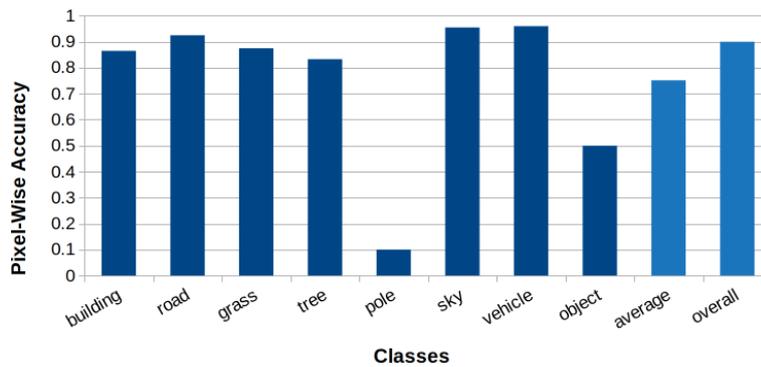


Figure 8. Quantitative evaluation of the FCN semantic segmentation model on a video stream from the field-test environment.

semantic segmentation is mainly used to identify terrain in the environment, and LiDAR information can be used to enhance obstacle detection.

3.3. Navigation

The navigation behaviors exhibited by our system are controlled by a behavior executor, seen on the right hand side of the autonomy architecture overview in Figure 2. The behavior executor implements the state machine shown in Figure 9. First, a high-level navigation goal (i.e., move to the location $(x, y, \theta)_{UTM}$) is introduced in the filled-in circle on the left of the figure. The state machine then selects from a set of learned behaviors based upon detected conditions present in the vicinity of the robot. These transitions are made to respect a reasonable precedence order consistent with the underlying urgency of the detected conditions. The THREAT condition has the highest priority – whenever a weapon is detected (described in Section 3.2.2) in the visual field of the robot it will immediately transition to the threat mode controller. Likewise, in the absence of the threat mode, the socially compliant navigation controller will be used if pedestrians are detected (described in

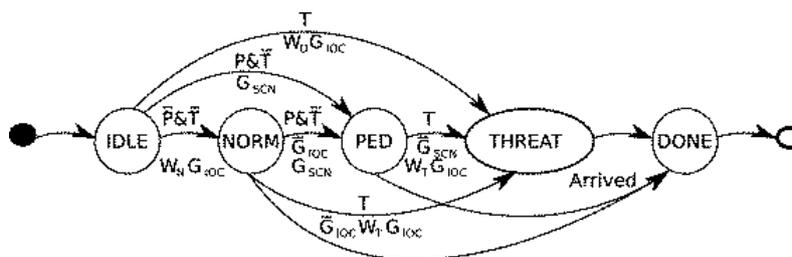


Figure 9. The state machine that controls the activation of control modules as a response to input state. The state machine proceeds through five states, IDLE, NORMAL, PEDESTRIAN, THREAT, and DONE, as shown abbreviated in the figure. Input state signals are shown above each transition arc; detected pedestrians and threats are shown by P and T , respectively, and the absence of these signals is indicated with an overbar (i.e., \bar{P} and \bar{T}). Module activations and configuration are shown below each transition arc. Symbols G_{SCN} and G_{IOC} indicate the goal command was sent to the socially compliant NavigAN and the learned inverse-optimal controller, respectively. An overbar on a goal command indicates a cancellation sent to terminate module activity prior to a preempt. Configuration changes to the IOC controller are indicated by W_N and W_T for setting normal mode weights and threat mode weights, respectively.

Section 3.2.1) in the area, and the normal-mode controller will be used in the absence of either of these conditions. Specifics of these three navigation controllers are provided later in this section.

Behavior transition stimulus conditions are monitored along with destination arrival while the state machine remains in one of the NORMAL, PEDESTRIAN, or THREAT states. In a mission where the robot must first move through an empty section of road and then encounters a group of pedestrians, it would transition from the NORMAL state to the PEDESTRIAN state. The state machine receives a pedestrian detection signal, represented in the figure as a P , and it then cancels the current navigation controller, which was the terrain-aware IOC behavior defined as the normal mode. It then sends a new goal to the socially compliant navigation controller to continue the operation. When the robot encounters a threat, designated as the symbol T , it cancels either the terrain-aware or socially compliant navigation controller depending on which is running, reconfigures the IOC controller for covert operation, and sends a new goal to the IOC controller. In the current system, transitions back to normal navigation from pedestrian mode are implemented but not shown in Figure 9 to reduce clutter. There are no transitions back from threat mode; the robot will continue the use of covert navigation until the goal is reached. These conditions are reset on subsequent navigation goals.

Each of the traversal behaviors integrated in our system are learned from data. Establishing a learning pipeline allows us the flexibility to learn additional behaviors when presented with novel mission requirements and adapt behaviors that may degrade over time due to environment or social changes. Although in some cases it may be feasible to hand code a behavior given expert domain knowledge, this becomes more challenging for complex behaviors. Further, as our motivating scenario specifies, the human teammates of this system may not be learning experts capable of such encoding. When behavior modification is needed quickly in the field, having a learning pipeline in place allows non-experts to simply execute the learning command with the new data. In the remainder of this section we present a description of the three behaviors and the learning techniques used to arrive at the behavior models.

3.3.1. Socially compliant navigation

The implementation and model discussed in this section represents our socially compliant navigation (SCN) behavior that is triggered when the robot is in the presence of pedestrians. This mode corresponds to the PEDESTRIAN state in the behavior executor.

When a robot navigates in human-populated environments, the robot needs to not only avoid collision with moving pedestrians but also comply with social norms such that pedestrians are comfortable with sharing the environment with the robot. Generally, the objectives of social navigation are composed of comfort, naturalness, and high-level societal rules (Kruse et al., 2013). Existing works in the social navigation domain mainly use reinforcement learning (RL) or inverse reinforcement learning (IRL). RL-based methods (Y. F. Chen, Everett, et al., 2017; Y. F. Chen, Liu, et al., 2017; Everett et al., 2021; P. Long et al., 2018) focus on avoiding collision to ensure comfort but can result in policies that are not natural to humans. Conversely, IRL-based approaches (Kitani et al., 2012; Kretschmar et al., 2016; Okal & Arras, 2016; Pfeiffer et al., 2016; Vasquez et al., 2014) are trained to mimic natural crowd navigation, but can generate unsafe trajectories because collision avoidance is not explicitly penalized during training. Our approach aims to achieve both comfort and naturalness.

Following the theory of social force (Helbing & Molnár, 1995), we develop NaviGAN, a generative adversarial network architecture for social navigation that represents an agent’s mixed intention to accomplish its self-interested goal while trying to comply with social norms of crowd navigation behavior (Tsai & Oh, 2020). An overview of NaviGAN is replicated in Figure 10. NaviGAN consists of three building blocks: (i) Intention-force \vec{F}_{intent} generator that models an agent’s intention for reaching a destination, (ii) Social-force generator that models the social force \vec{F}_{social} and fluctuation force \vec{F}_{fluct} , and (iii) Social-aware discriminator that discovers the latent social influences from discriminating those navigation paths generated by the generator against expert demonstrations.

The NaviGAN approach (Tsai & Oh, 2020) was evaluated in terms of discounted accumulated reward, known as *social score* using the reward function proposed in C. Chen et al. (2019) that combines the task success rate and the level of comfort based on distance. As an upper-bound reference, humans are able to reach their goals while maintaining comfortable distance from others 96% of the time. The lower-bound reference used in our experiment was a navigation algorithm optimized for reaching a goal without social awareness, which allows a robot to reach its goal 91% of the time but in a reckless manner with its comfort rate of 81%. In a set of experiments, our approach reaches the goal at a lower rate of 85% than the baseline but achieves substantially higher social awareness score of 97% that is comparable to human performance. NaviGAN is a real-time algorithm that runs at 10 Hz when using an onboard Jetson computer during our experiments.

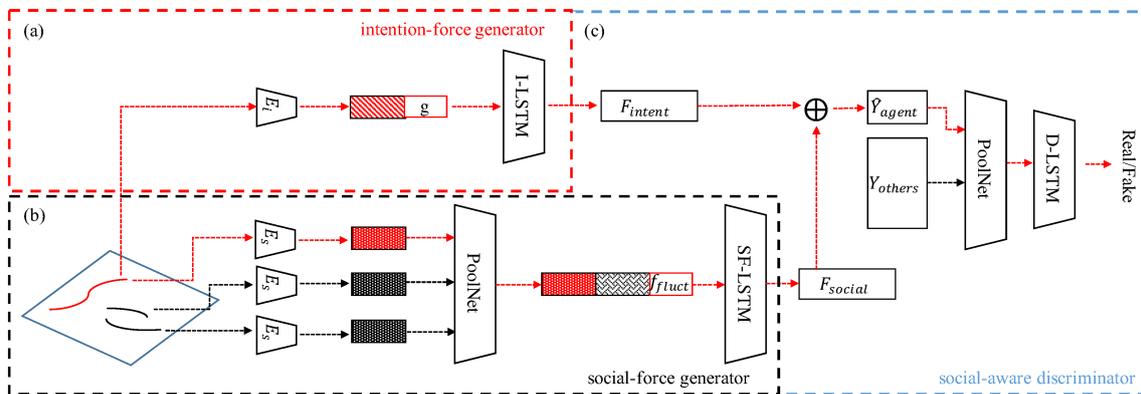


Figure 10. Overview of NaviGAN (Tsai & Oh, 2020). NaviGAN is composed of three major building blocks: 1. Intention-force generator (block (a)) that models the intention force \vec{F}_{intent} toward goal state; 2. Social-force generator (block (b)) that models the social force \vec{F}_{social} and fluctuation force \vec{F}_{fluct} ; and 3. Social-aware discriminator (block (c)) that discovers the latent social aspects from discriminating between expert demonstrations and navigation behavior generated by the generator. The red color indicates that the vector holds data relevant to the target agent, and the black color is used for other agents.

3.3.2. Terrain-aware traversal

In some autonomous navigation scenarios, planning trajectories that only consider obstacle avoidance is sufficient. However, additional environment context such as the specific terrain in the scene can be used to enhance mobility, e.g., path planning only on traversable terrain, adherence to “rules of the road,” or mission formation. These types of behavior models can be encoded as a reward function, where each terrain feature is represented by a reward/cost value that describes how desirable that terrain is to drive across.

To learn this reward function, we employ IOC, also commonly referred to as IRL. IOC learning makes use of demonstrations, which encode an optimal policy, to learn a reward function. Our previously proposed IOC traversal behavior learning approach (Wigness et al., 2018) is replicated in Figure 11. In this specific formulation, a demonstration is a trajectory driven by a human teleoperating a robot, and the trajectory encodes optimal traversal with respect to a set of environment terrains and obstacles identified by semantic segmentation (discussed in Section 3.2.3) and LiDAR (discussed in Section 3.1).

Both linear (Abbeel & Ng, 2004; Kitani et al., 2012; Ratliff et al., 2006; Ziebart et al., 2008), and nonlinear reward function learning (Choi & Kim, 2013; Levine et al., 2011; Wulfmeier et al., 2016) have been explored in the literature. The requirement of a large number of demonstrations to learn reliable nonlinear reward functions makes these approaches less ideal for applications where a robot may need to adapt quickly in the field. Thus, we leverage the Maximum Entropy IRL algorithm (Ziebart et al., 2008) as it resolves ambiguity in distribution selection by focusing on matching feature expectations, and as shown in previously results, can learn from a relatively small number of demonstrations (Wigness et al., 2018).

The first IOC traversal behavior is designed to represent the robot’s normal operating mode, denoted as the NORMAL state in the behavior executor, which considers the type and relative proximity of terrains when planning paths. More specifically, this model should encode the behavior of driving on the edge of the road. Unlike traversability analysis literature (Papadakis, 2013) that focuses on assigning or learning fixed costs for terrain classes (Milella et al., 2015; Roncancio et al., 2014; Shneier et al., 2008; Silver et al., 2008; Suger et al., 2015; Talukder et al., 2002), this behavior

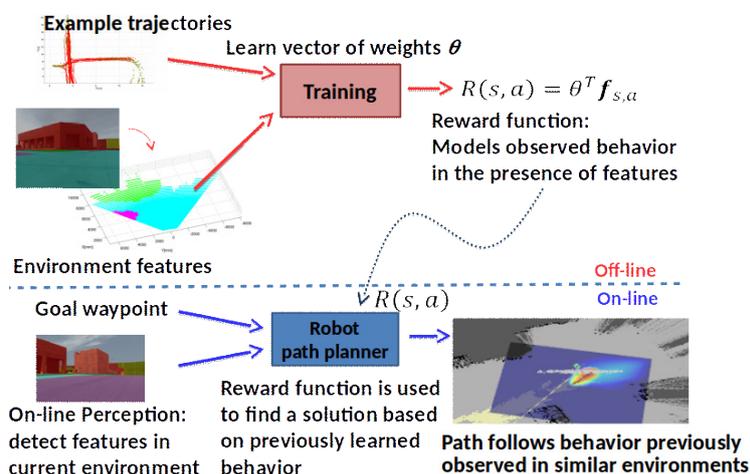


Figure 11. Overview of the IOC behavior learning system (Wigness et al., 2018). This system learns a reward function off-line given a set of human-demonstrated trajectories and environment feature maps. The demonstrations represent the optimal behavior according to the human demonstrator with respect to the features observed. The learned reward function is used during on-line operation to assign costs to the environment and plan paths to the goal.

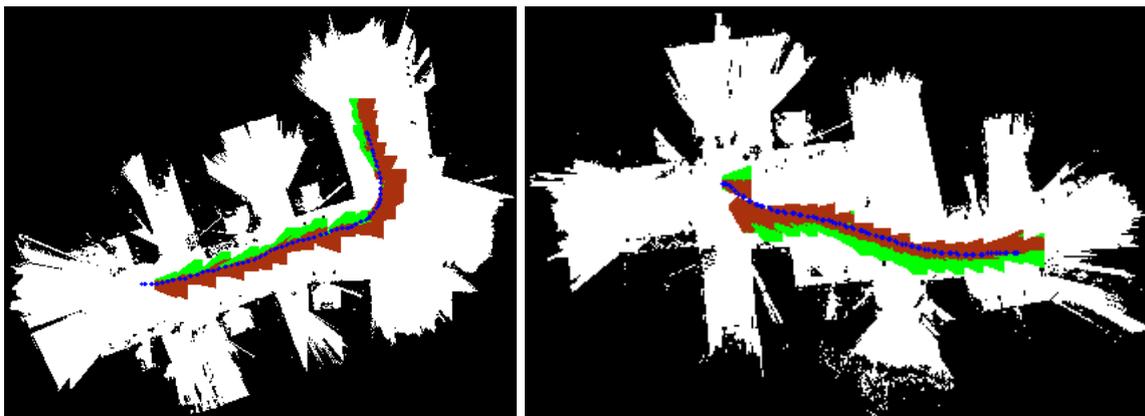


Figure 12. Example training demonstrations for terrain-aware traversal that focuses on driving near the edge of the road. The demonstrated trajectory is illustrated in blue. The terrain features grass, road, and obstacles are colored green, red, and black, respectively.

model must learn weights for a feature representation that encodes the relative proximity of terrains and objects in the environment.

Figure 12 shows some example trajectory demonstrations that represent the optimal behavior of driving near the edge of the road. The demonstrated trajectory is seen in blue and overlays the environment feature maps – grass (green), road (red), and obstacles (black) – used during the learning process. The reward function for this behavior was learned with six demonstrations, where each demonstration covered about 5 m to 10 m of distance from start to end. This behavior has been demonstrated thoroughly in previous field experiments (Wigness et al., 2018).

3.3.3. Covert traversal

The second IOC learned behavior is meant to resemble covert traversal when the robot perceives a threat in the environment. This behavior is represented as the THREAT state in the behavior executor, and is triggered by the rifle detector discussed in Section 3.2.2. Covert in these experiments is defined as a trajectory that avoids large open areas, e.g., the middle of the road, that are highly visible from a vantage point. Instead, this learned behavior should generate trajectories along walls and buildings, which provide some cover to the moving robot.

Covertiness is often associated with tactical operation amidst the presence of possible danger and adversaries. We clarify here that the learned behavior we present is certainly not an accurate representation of covert tactics used in real-world operation. Instead, this covert representation is intended to be a stand-in to simply expand the diverse set of traversal behaviors that may be triggered in an environment based on different operational context.

Our previous field demonstration of this behavior showed that we could learn a reward function from two human demonstrations (Wigness et al., 2018). To quantitatively show the behavior was learned, we compared the IOC-generated trajectory between two waypoints to that of a human-demonstrated trajectory between the same two waypoints. Using the Modified Hausdorff Distance (MHD) (Shao et al., 2010), we showed that on average across all trials performed, IOC has an MHD of 1.415, where a lower MHD indicates higher similarity. The trajectories collected by running a baseline planner that does simple obstacle avoidance achieved a mean MHD of 5.201.

Figure 13 illustrates the difference in training demonstrations that could be provided for the covert and terrain-aware behaviors when the initial (green circle) and goal locations (yellow circle) are the same in each case. The red trajectory is an illustration of a covert demonstration, where the robot would deviate toward the building edge for as long as possible until finishing at the goal destination.

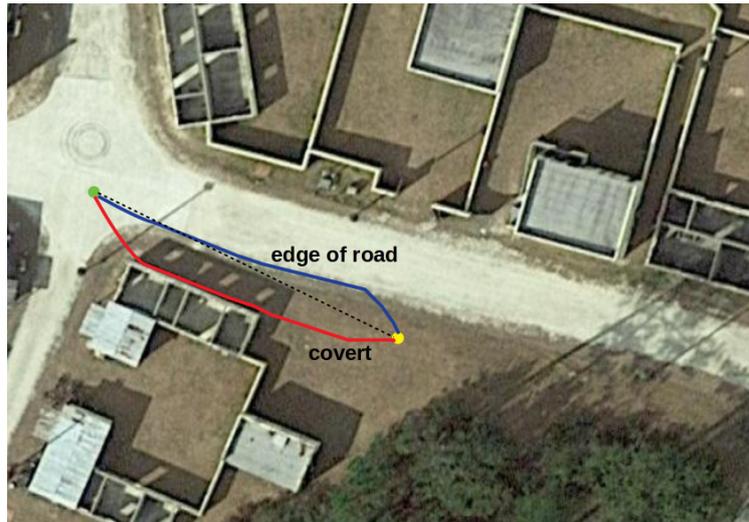


Figure 13. Illustration of demonstration trajectories that could be provided for two different IOC learned behaviors that begin and end at the same locations. Covert traversal near a building edge is shown as the red trajectory, edge of road traversal is shown as the blue trajectory, and the black dotted line represents the shortest path between the starting and ending waypoints.

The blue trajectory, on the other hand, illustrates a demonstration that traverses near the edge of the road. The black dotted line represents the shortest path between the two waypoints.

4. Field testing

The integration of these perception and navigation modules, along with the rest of the autonomy stack, allows us to deploy an intelligent autonomous system to the field for evaluation. Specifically, we define a scenario in which the robot must maneuver in the environment given dynamic conditions in which certain traversal behaviors are more desirable than others.

Field experimentation of the integrated system was performed in an outdoor environment located at a military training facility. The environment is meant to resemble a small village containing various buildings and gravel roads in addition to other terrain and object classes. The basic experiment setting involves the robot autonomously navigating from its current location to a specified goal waypoint in the environment.

We run several trials of the experiment in two different locations of the environment. Figure 14 is an overhead view of the field environment and roughly depicts the two locations in which evaluation is performed. The robot must traverse about 50 m to 100 m to reach its goal location in each trial. During these trials we record the perception package output, behavior transitions, and trajectories of the learned behaviors. We use this data to evaluate the individual components and provide an overall analysis of the integrated system. In the remainder of this section we describe the scenario defined for testing and provide an analysis of the trials performed.

4.1. Scenario design

We define a particular scenario for the experimental trials that is intended to mimic a dynamic environment with changing mission context. In this scenario, as the robot navigates to its goal waypoint it will need to traverse through an area that is densely populated by pedestrians. Most of these pedestrians are non-hostile, but the detection of weapons in the environment should indicate a threat, and the robot will need to act accordingly by assuming the environment has become hostile.



Figure 14. Overhead view of the field experiment environment with the two testing locations outlined in blue. Possible start and goal locations for the robot in each trial are indicated by green circles.

Specifically, while moving toward the goal location the robot may encounter the following environment conditions:

1. There are no moving pedestrians or weapons present
2. Moving pedestrians are within 10 m of the robot
3. A weapon is within 10 m of the robot
4. Both moving pedestrians and weapons are near the robot

These conditions will be identified by the robot using the onboard perception stack (i.e., pedestrian tracking and weapon detection). These perception cues are used by the robot to transition between traversal behaviors most appropriate given the current mission context, as shown in Figure 9. The conditions of the scenario are presented in roughly the same order for each of the experimental trials. First, the robot begins operation without the presence of humans or weapons. Next, the robot encounters pedestrians. Finally, a weapon is introduced into the scenario.

Although the sequence of events remains constant, there are a number of varying conditions between the trials. These include the number of pedestrians, emergence of pedestrians, weapon occlusion, and duration of conditions. Specifically, an experimental trial can include one to six pedestrians. Pedestrians can emerge in the environment by walking head-on toward the robot, crossing the street in front of the robot, moving toward the robot from behind, or some combination of these. Weapons in the environment enter the scene being held by a pedestrian, and various grasps introduce different degrees of rifle occlusion. Finally, the duration of each condition is different in each trial.

4.2. Robot description

The robot platform used for experimentation is a Clearpath Husky, which can be seen in Figure 15. The sensor payload is mounted on a optical plate that is secured in an elevated position from the robot's hull. A Velodyne VLP-16 is mounted on the upper plate and is used for mapping as described in Section 3.1, as well as person tracking as described in Section 3.2.1. An Intel RealSense d435i is also attached to the upper plate and is used for semantic segmentation and object recognition



Figure 15. Clearpath Husky robot used for system integration and evaluation.

as described in Section 3.2. The robot’s orientation is determined by a Microstrain GX5-25 IMU mounted on vibration isolators on the upper plate. This orientation is combined with platform wheel odometry to provide an estimate of platform motion for use in the mapping process described in Section 3.1. Several additional sensors, also visible in Figure 15, are used in other experiments not described in this paper.

Two computers are located onboard the Husky. A Brix with an Intel Core i7 is used for navigation, mapping, and high-level control. A Zotac EN1060K with a 6GB Nvidia GTX 1060 is used for visual processing with neural networks running on the GPU, including both semantic segmentation and object detection. Socially compliant navigation is currently implemented on an offboard NVidia Jetson computer; however, this could be easily integrated on the platform. A second Husky lead-acid battery is integrated into the robot hull to power these sensors and computers via DC/DC converters.

4.3. Results

We conducted a total of 12 experimental trials for the scenario specified in Section 4.1. Two occurred at Location A, as seen in Figure 14, while Location B was used for the remaining 10 trials. We discuss the general performance results seen across all 12 trials without differentiating location. Qualitative evaluation is discussed largely by referencing visualizations of the perception components and navigation paths displayed in Rviz during operation. We briefly outline the visualizations that will be seen throughout the remainder of this section. The basis of many visualizations is the mapping produced from our SLAM system. Overlain this map are:

1. Terrain information from semantic segmentation, grass and road are shown as green and purple regions, respectively
2. LiDAR object and pedestrian tracks, shown as blue and red cuboids, respectively
3. Planned paths for navigation behaviors shown as green trajectories

Object detection bounding boxes are overlaid camera images. Note that class “1” represents entrance detection, which is not specifically used in this scenario design, and class “2” represents rifle detections. Detection confidence scores can be seen next to the class label for each detection.

Overall, our system successfully completed 11 of the 12 navigation trials. Success is defined by the robot’s arrival at the goal waypoint while transitioning between the navigation behaviors when environment conditions change. In these initial experiments we do not provide a quantitative evaluation of “time to transition,” (i.e., we do not record the delay incurred from the moment the environment changes to the moment of behavior transition). However, we note that qualitatively the perception cues were issued in a timely manner and correctly triggered the behavior designed for each condition. Figure 16 shows the sequence of traversal behavior transitions during one experimental trial. During

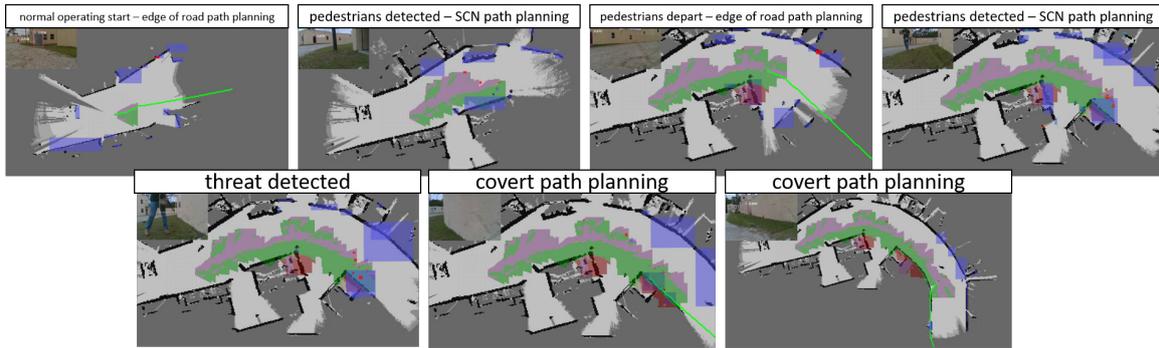


Figure 16. Example sequence of behavior changes for one of the experimental trials. Top row: Transitions between normal operation and socially compliant navigation when pedestrians are detected. Bottom row: Detection of a threat and covert paths planned.

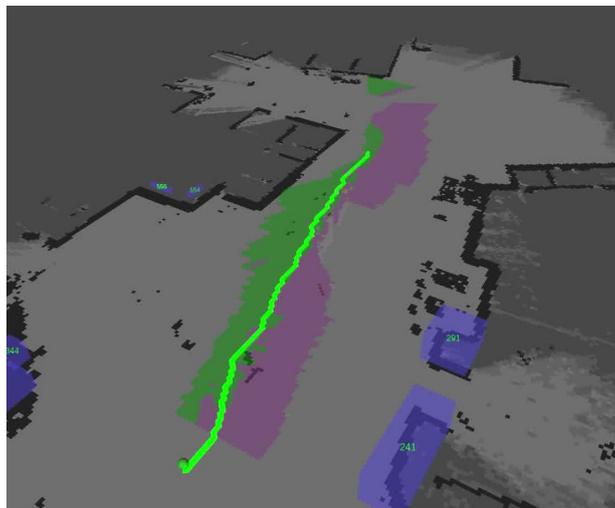


Figure 17. Paths generated in an experimental trial using terrain-aware navigation to follow the edge of road, which is the initial behavior of the scenario when no pedestrians or threats are detected in the scene. The green line represents the planned path. The green and purple colors on the map represent perceived grass and road, respectively. Building edges and other obstacles are seen in the map as black regions captured from LiDAR.

this trial the robot transitions between normal and socially-compliant navigation multiple times, as pedestrians enter and leave the scene, until finally a threat is detected and the robot transitions into covert traversal. The other experimental trials represent a similar flow in transition between navigation behaviors. Additional detailed analysis and discussion of these results follow.

Navigation Startup: Each experimental trial started with a static scene (i.e., no moving pedestrians or threats perceived). We used the terrain-aware navigation behavior, training via IOC, as the initial navigation mode. Recall that the terrain-aware behavior was trained to drive near the edge of the road. Figure 17 shows an example planned path for this initial behavior. The trajectory shown as the bright green line indicates the behavior was learned well, as it aligns with the grass and road terrain, shown as green and purple ground cover in the map.

Pedestrian Detection and Socially-Compliant Navigation: Pedestrian location and movement was varied throughout the experimental trials to show that the socially compliant navigation behavior

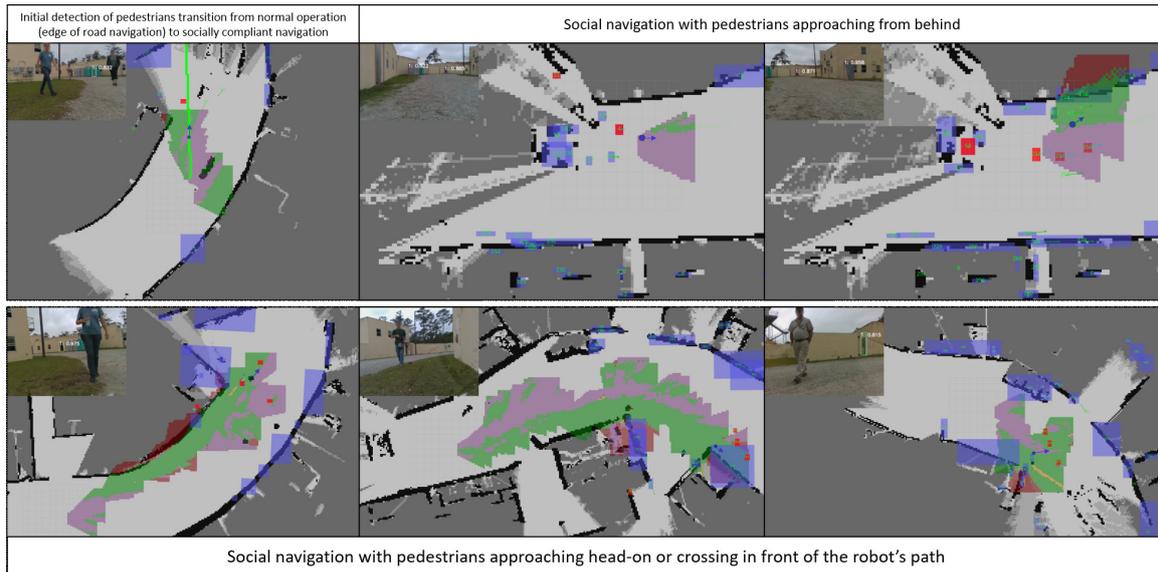


Figure 18. Example visualizations of the environment when pedestrians entered the scene to trigger socially compliant navigation during the experimental trials.

could reason about pedestrians in a number of different scenarios. Figure 18 shows some example visualizations of the operating environment when pedestrians entered experimental trials to trigger socially compliant navigation. In these visualizations the robot is marked by a blue sphere with an arrow indicating its forward moving direction. The red cuboids represent pedestrians being tracked by the system, and green arrows on the pedestrians indicate the direction of their movement. Notice that pedestrians move toward the human head on, approach from behind, and cross directly in front of the robot.

As pedestrians are tracked, socially-compliant navigation generates a planned path that not only avoids collision with the pedestrians, but also maintains a comfortable distance from the pedestrian, as learned by the model. Throughout the experimental trials, the robot commonly chose to veer to its right to establish distance between itself and the pedestrians approaching head on. Two example sequences of this can be seen in Figure 19, where each row represents a different experimental trial with an ordered sequence of frames showing the pedestrians in the scene. These frames are captured on a GoPro mounted to the robot and were not used as sensor input for perception. The robot’s mounting plate can be seen in the bottom of the image and can be used as a reference to see how the robot’s path migrates to the right over time. In the top trial the robot plans a path into the grass to avoid an approaching pedestrian, and in the bottom trial the robot moves to the right to pass between two approaching pedestrians.

Another common behavior when in socially compliant navigation mode was for the robot to simply stop when a pedestrian approached quickly. This stopping decision ensured that the robot did not plan a path into the pedestrian and allowed the pedestrian to make a decision about where to move to avoid collision. Although this behavior can be seen as a safety feature, our current work aims to relax this behavior to retain a robot’s pace to reach its destination more efficiently.

Threat Detection and Covert Traversal: Mixed within the group of pedestrians that the robot encountered was a pedestrian meant to resemble a threat in the environment. This threat was represented by a physical weapon, which the onboard object detection model was trained to detect. Recall that the training of this model was done by leveraging large numbers of synthetic training



Figure 19. Two sequences of frames captured from a GoPro mounted on the robot during experimental trials. These frames capture the scenario when the robot was in socially compliant navigation mode. Pedestrians are uniquely labeled with integer IDs for visualization. Top: Robot deviates to the right into the grass. Bottom: Robot deviates to the right to pass between two approaching pedestrians.



Figure 20. Example images showing the initial threat detection from the experimental trials (rifle = object 2). Also shown are some entrance detections (object 1).

images (discussed in Section 3.2.2). We provide a thorough analysis of how this trained model performs in the field to cue covert behavior.

Figure 20 shows example images of the initial threat detection that transitioned the robot into covert behavior. Across all experimental trials these initial detections were made when the rifle was between 0.5 m and 9.9 m away from the robot. In most cases when the initial detection is relatively close, the pedestrian carrying the threat approached the robot from behind or the side, keeping the rifle out of the camera field of view for quite some time. In addition to being able to detect the rifle from various distances, the detections are made under different degrees of occlusion.

The maximum range for threat detection during experimentation was 11.5 m. Figure 21 provides a more detailed view of the threat detection ranges for each experimental trial (left), and the average

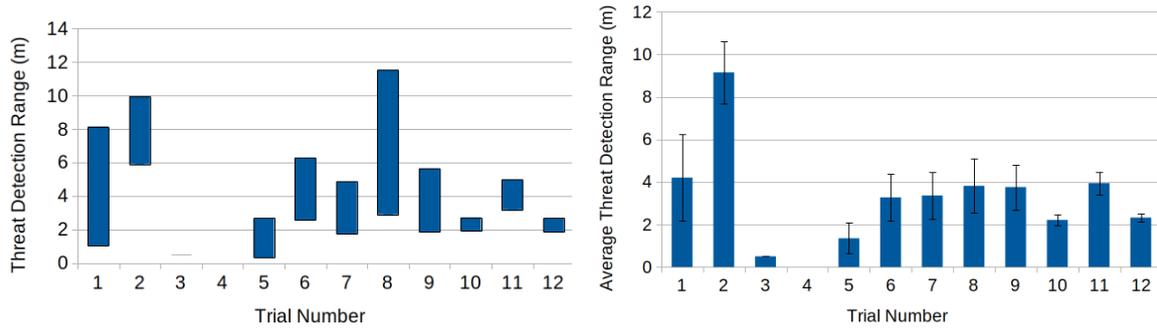


Figure 21. Images that outline the threat detection range during each experimental trial.

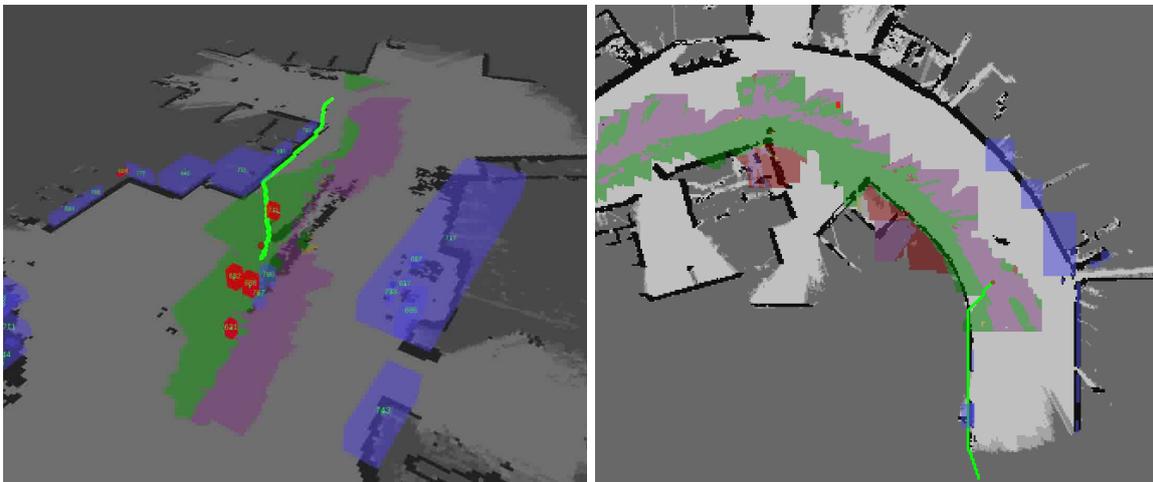


Figure 22. Paths generated using covert behavior in two experimental trials. The green line represents the planned path. The green and purple colors on the map represent perceived grass and road, respectively. Building edges and other obstacles are seen in the map as black regions captured from LiDAR.

range at which rifle detections were made for each trial (right). In trial 3 the detection distance is quite close because the pedestrian approached the robot from behind, appeared in front of the robot long enough to gather a detection, and then quickly left the field of view. Trial 4 represents the only experimental trial where a false rifle detection was made that caused the robot to transition into covert mode prematurely. Thus, no true positive detection distance data is recorded. We delay the discussion of false detections and lessons learned to Section 5. Overall, these plots highlight the range diversity of the object detection model during experimentation.

The transition to covert traversal throughout the experimental trials was quite distinct because most operation prior to identifying a threat was done at a fair distance from buildings. Recall from Figure 14 that the robot's initial and destination locations are roughly along a road network, and pedestrians enter the scene as the robot is navigating along this road. Thus, when a threat is perceived the robot must deviate from the road significantly to find a building edge to achieve covert behavior. Figure 22 shows two example covert trajectories produced from the learned model after a threat was perceived.

5. Lessons learned, discussion and future work

Intelligent systems are far from flawless. During our system integration and experimentation we identified a number of outstanding challenges that still need to be addressed in our proposed system and within the robotics research community. In this section, we discuss some of our failures, system configurations used to mitigate these failures, and provide a general discussion on lessons learned and future work related to these challenges.

Mitigating false detections and addressing occlusion challenges

Many characteristics make the detection of rifles difficult. They have little texture, are long and thin and thereby in many cases occupy only a small portion of training bounding boxes, and when carried by people are often occluded by portions of the human body. Figure 23 shows some examples of false rifle detections seen during experimentation. During testing, to mitigate false behavior transitions, we filtered out low confidence detections and required detections in three sequential frames before triggering a transition to covert behavior. With these configurations, false detections leading to incorrect navigation transitions only happened once during the field test.

From the quantitative evaluation in Section 3.2.2 we saw that using only the synthetically generated data to train a rifle detector was outperformed by the model that used the real-world data with hand annotations. Yet, the results were promising given that the approach required far less human effort, and it allows for the training of a detector in cases where it is not possible to collect images of the object in relevant environments. We discussed how part of the poorer performance can be attributed to the fact that the synthetic data did not include examples of people carrying rifles, which causes significant object occlusion.

However, we also hypothesize that using the synthetic data helped generalize rifle detections beyond the case when a human was holding the object. Although in our experiments the rifle was always initially introduced into the scene when carried by a pedestrian, the detector was able to correctly detect the rifle when it was not being carried as well. Figure 24 provides some examples of the weapon being detected in an upright position. Notice in the right image the color of the weapon and the pole it is leaning against are very similar, making this a more challenging condition to detect. So although the synthetic data generated was limited in that it did not represent scenarios of weapons being held by a human, it certainly enhanced the training set for scenarios where the rifle was found in the environment in isolation.

Learning from trajectories only may be too weak of a signal for social navigation

In the majority of existing work in social navigation, the problem formulation only considers trajectories without including environmental constraints, such as terrain types, static obstacles, or structural features (e.g., sidewalks or doors). Furthermore, because pedestrians are represented as their trajectories, other significant features of their states—such as full body pose, eye gaze, facial expressions, heights, and other physical features—are stripped from training data. In order to model



Figure 23. Example images showing false-positive threat (rifle) detections from the experimental trials.



Figure 24. Example images show threat (rifle = object 2) detections when the weapon is not being carried by a person. Also shown is one entrance detection (object 1).

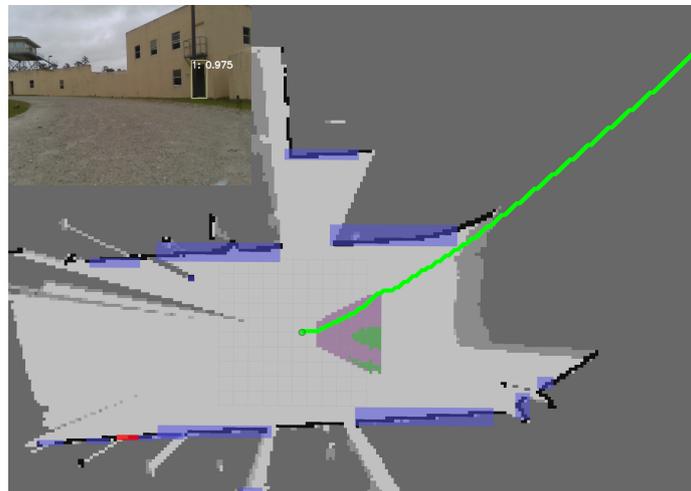


Figure 25. Visualization of a challenging scenario for the IOC behavior navigation where terrain features are provided only at a short range for path planning. The lack of terrain information results in a long-range trajectory to the goal that is mostly evaluating unknown space.

pedestrian behaviors and learn to navigate among them, future directions should expand the problem formulation beyond trajectories.

Limited terrain projection results in poor long distance planning

There are still some limitations to the learning from demonstration approach used to generate behaviors. These behaviors rely on the terrain and obstacle feature maps that are produced from semantic segmentation and LiDAR strikes. Currently when operating on-line, the generation of these terrain feature maps is projected only 5 m in front of the robot, making it difficult to reason about the best trajectory through unexplored regions of the environment. Figure 25 shows an example of this during a field test trial. The experimental scenario has just begun and there is limited terrain information available. The edge of the road is not clearly defined from this information as only a portion of it has been classified. In these cases, the limited information results in planned trajectories to the goal that are not as clearly representative of the trained behavior as they could be with a longer range view of the terrain.

The IOC behavior path planner does re-plan as additional terrain features are discovered, which allows new information to modify the planned trajectory. However, the additional terrain data continually only provides short-range information, meaning the planner always lacks long-range information needed to fully mimic the demonstrated behavior. We are investigating two approaches for future

work to improve the current implementation. First, existing terrain maps could be loaded as prior information and modified on-line to reflect any environment changes. These maps could be coarse terrain identifications from aerial vehicle teammates or from previous operation of a ground vehicle. This provides a more informative global view of the environment at the start of operation. A second approach could be to simply increase the range of terrain projection used to generate the feature maps. This is ideal for operation in novel environments where no a priori information is available. This is feasible in the sense that the semantically classified images encode terrain information at a much greater distance than we are currently projecting, but increasing the distance for feature maps requires more onboard computation. Improving the speed of this computation will provide longer range information that can better direct the planning system for these behaviors.

Related to this shortcoming, the robot can also deviate from the planned trajectory due to unmodeled dynamics or unanticipated terrain interactions. In our current system, the local planner attempts to return to the planned trajectory; however, in doing this the robot may not exhibit the desired behavior. Furthermore, due to periodic global trajectory replanning, oscillatory behavior can be observed when the trajectory deviation is large. One improvement over the planning system described in this paper would be to incorporate platform dynamics and generate a policy in the neighborhood of the ideal trajectory that would both tend to better favor the desired terrain as well as recover quickly from deviations.

Integrated intelligence takes more than system integration

In this article, we showcase an example of high-level robot intelligence where a robot visually analyzes the situational context of its environment to seamlessly interleave multiple behaviors according to the context. As illustrated in this example, many of the decision-making problems in robotics applications require intellectual capabilities, including semantic perception, understanding of environmental or social contexts, and reasoning about complex constraints. Despite the hype over the successes of end-to-end deep learning, building this type of cognitive-level intelligence in robotics is one of the remaining challenges in intelligent robotics technologies. As a community, we need to define the science of integrated intelligence in the context of robotics.

There are a number of trade-offs between end-to-end learning and a modularized system for the scenario we have defined. End-to-end learning provides a holistic representation of the environment, which arguably provides a more cohesive context during operation over individual perception cues. However, the question of mission complexity plays a critical role in determining the effectiveness of this holistic approach. The emerging research results for end-to-end learning techniques have focused on highly structured tasks thus far, which allows the behavior of the autonomous system to be constrained. Examples include placing books on shelves (Singh et al., 2019), consumer electronics (Kim et al., 2018), and navigation for agriculture crop row following (Bakken et al., 2019) or urban on road navigation (Bojarski et al., 2016; Z. Chen & Huang, 2017; Codevilla et al., 2018; Sadat et al., 2020). While these results are not to be diminished, significant progress is still required to determine how such an approach might be useful in our defined scenario, which displays much less structure and requires a larger and more diverse set of behaviors to negotiate the current environment.

In many cases, decision making depends on a larger context than what is directly perceived. For instance, observing a knife in a kitchen is normal, whereas a knife in a nursery room raises alarm. In more complex examples, present perception is simply not enough for deciding critical actions. Whereas an end-to-end design may be able to capture the complexity, e.g., using a very large network and a massive amount of training data, abstraction in the form of hierarchical or structured architecture can support a compact representation of complex decision making under various contexts.

Several important factors need to be considered. First, as the complexity of the problem increases, the need for larger sets of training data will also likely increase. As previously mentioned, for some real-world applications, the ability to collect domain-relevant training data prior to deployment can be challenging. Further, collecting enough training data for reliable operation may simply be infeasible. Second, as robots are deployed with human teammates, an aspect of explainability and transparency will likely need to exist to ensure that the human fully trusts its robotic teammate. The

lack of transparency that currently exists in end-to-end systems is a major shortcoming compared to the use of perception cues, which clearly define why an action, e.g., navigation behavior transition, was taken.

Future research directions include communication protocol for subcomponents that can be easily customized to support cognitive abilities. For instance, Robot Operating Systems (ROS) has been widely used for this purpose, despite its limited expressibility. Interdisciplinary research between cognitive science and robotics fields should also be further encouraged.

Testbeds and evaluation metrics are needed for human-robot teaming

Our social navigation approach is based on a model trained using human pedestrian datasets. During the experiments, we observed that humans adapt to robot behaviors as quickly as the robot also tries to achieve naturalness. In addition to the trajectories, which are the only input to the majority of existing algorithms including ours, human reactions appear sensitive to many other factors. Some of these factors include those pertaining to the physical appearance of a robot, e.g., shape, height, rigidity, and sound, e.g., motor sound, that comes from the robot.

Currently, these additional factors are not encoded in datasets that can be used for robot learning, causing learned models to rely on only some of the data a pedestrian uses when interacting with these autonomous agents. As robotics research continues to emphasize applications that maintain humans and robots in close proximity, and directly interact/react with one another, e.g., hospital aids, household appliances, street couriers, the need for richer datasets, testbeds, and metrics that comprise all the previously mentioned factors will become crucial.

6. Conclusion

The continued development of intelligent robotic systems has helped push the state of the art in autonomy and has led to unique human-robot teaming capabilities. This is particularly relevant for fielded applications where human teammates need to maintain situational awareness to ensure their safety. In these scenarios, a fully autonomous robot that can make navigation decisions without requiring human oversight or teleoperation will be invaluable.

As part of the research in the RCTA that aimed to develop these capabilities, we have presented a fully integrated intelligent system that makes use of mapping, perception, and navigation stacks to autonomously navigate in complex, unstructured, and dynamic environments. In this system, the robot leverages both vision and LiDAR-based perception cues to constantly update its current understanding of the environment. This simple, yet effective, approach allows the robot to transition between various learned navigation behaviors that provide safe and reliable traversal given the current scenario context.

Our research in this area will continue to evolve as we look at ways to further enhance the intelligence architecture used onboard the robot, and make use of additional perception information during the learning process for navigation behaviors. Further analysis related to the use of synthetically generated data for learning in domains with limited training data and novel benchmarking metrics for human-robot teaming applications are other areas that we expect to investigate in future work to continue to provide meaningful impact to the robotics community at large.

Acknowledgments

We would like to thank Julia Donlon, Matthew Young, and Merle DeLancey for their help during experimentation. A special thanks to Jason Gregory, Daniel Sahu, and John Kozar for their hardware technician support. Finally, thank you also to the RCTA leadership team, including but not limited to, Stuart Young, Dave Baran, Dilip Patel, and Matthew Weiker, for their support throughout the program.

ORCID

Maggie Wigness  <https://orcid.org/0000-0003-1707-8106>
 John G. Rogers III  <https://orcid.org/0000-0002-6074-0823>
 Christoph Mertz  <https://orcid.org/0000-0001-7540-5211>
 Luis Navarro-Serment  <https://orcid.org/0000-0002-2803-3977>
 Jean Oh  <https://orcid.org/0000-0001-9709-2658>

References

- Abbeel, P., & Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. *Twenty-first international conference on Machine learning - ICML '04*, 1–8. <https://doi.org/10.1145/1015330.1015430>
- Albus, J., Bostelman, R., Chang, T., Hong, T., Shackleford, W., & Shneier, M. (2006). Learning in a hierarchical control system: 4D/RCS in the DARPA LAGR program. *Journal of Field Robotics*, 23(11-12), 975–1003. <https://doi.org/10.1002/rob.20162>
- Bajracharya, M., Howard, A., Matthies, L. H., Tang, B., & Turmon, M. (2009). Autonomous off-road navigation with end-to-end learning for the LAGR program. *Journal of Field Robotics*, 26(1), 3–25. <https://doi.org/10.1002/rob.20269>
- Bakken, M., Moore, R. J. D., & From, P. (2019). End-to-end learning for autonomous crop row-following. *IFAC-PapersOnLine*, 52(30), 102–107. <https://doi.org/10.1016/j.ifacol.2019.12.505>
- Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., & Zieba, K. (2016, April). *End to end learning for self-driving cars*. arXiv: 1604.07316 [cs.CV].
- Bonin-Font, F., Ortiz, A., & Oliver, G. (2008). Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems*, 53(3), 263. <https://doi.org/10.1007/s10846-008-9235-4>
- Boularias, A., Duvall, F., Oh, J., & Stentz, A. (2015). Grounding spatial relations for outdoor robot navigation. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 1976–1982. <https://doi.org/10.1109/ICRA.2015.7139457>
- Carranza-García, M., Torres-Mateo, J., Lara-Benítez, P., & García-Gutiérrez, J. (2020). On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data. *Remote Sensing*, 13(1), 89. <https://doi.org/10.3390/rs13010089>
- Chen, C., Liu, Y., Kreiss, S., & Alahi, A. (2019). Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 6015–6022. <https://doi.org/10.1109/icra.2019.8794134>
- Chen, Y. F., Everett, M., Liu, M., & How, J. P. (2017). Socially aware motion planning with deep reinforcement learning. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1343–1350. <https://doi.org/10.1109/iros.2017.8202312>
- Chen, Y. F., Liu, M., Everett, M., & How, J. P. (2017). Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 285–292. <https://doi.org/10.1109/icra.2017.7989037>
- Chen, Z., & Huang, X. (2017). End-to-end learning for lane keeping of self-driving cars. *2017 IEEE Intelligent Vehicles Symposium (IV)*, 1856–1860. <https://doi.org/10.1109/ivs.2017.7995975>
- Choi, J., & Kim, K.-E. (2013). Bayesian nonparametric feature construction for inverse reinforcement learning. *IJCAI International Joint Conference on Artificial Intelligence*, 1287–1293.
- Codevilla, F., Muller, M., Lopez, A., Koltun, V., & Dosovitskiy, A. (2018). End-to-end driving via conditional imitation learning. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1–9. <https://doi.org/10.1109/icra.2018.8460487>
- Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 379–387.
- Delmerico, J., Mintchev, S., Giusti, A., Gromov, B., Melo, K., Horvat, T., Cadena, C., Hutter, M., Ijspeert, A., Floreano, D., Gambardella, L. M., Siegwart, R., & Scaramuzza, D. (2019). The current state and future outlook of rescue robotics. *Journal of Field Robotics*, 36(7), 1171–1191. <https://doi.org/10.1002/rob.21887>
- Desouza, G. N., & Kak, A. C. (2002). Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2), 237–267. <https://doi.org/10.1109/34.982903>

- Dwibedi, D., Misra, I., & Hebert, M. (2017). Cut, paste and learn: Surprisingly easy synthesis for instance detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, 1301–1310. <https://doi.org/10.1109/iccv.2017.146>
- Everett, M., Chen, Y. F., & How, J. P. (2018). Motion planning among dynamic, decision-making agents with deep reinforcement learning. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3052–3059. <https://doi.org/10.1109/iros.2018.8593871>
- Everett, M., Chen, Y. F., & How, J. P. (2021). Collision avoidance in pedestrian-rich environments with deep reinforcement learning. *IEEE Access*, 9, 10357–10377. <https://doi.org/10.1109/access.2021.3050338>
- Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2), 303–338. <https://doi.org/10.1007/s11263-009-0275-4>
- Feurtey, F. (2000). *Simulating the collision avoidance behavior of pedestrians* (Master’s thesis). University of Tokyo. Tokyo, Japan.
- Fridovich-Keil, D., Ratner, E., Peters, L., Dragan, A. D., & Tomlin, C. J. (2020). Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 1475–1481. <https://doi.org/10.1109/ICRA40945.2020.9197129>
- Gregory, J., Fink, J., Stump, E., Twigg, J., Rogers, J. G., III, Baran, D., Fung, N., & Young, S. (2016). Application of multi-robot systems to disaster-relief scenarios with limited communication. *Field and Service Robotics: Results of the 10th International Conference*, 639–653. https://doi.org/10.1007/978-3-319-27702-8_42
- Gul, F., Rahiman, W., & Alhady, S. S. N. (2019). A comprehensive study for robot navigation techniques (K. Chen, Ed.). *Cogent Engineering*, 6(1), 1632046. <https://doi.org/10.1080/23311916.2019.1632046>
- Helbing, D., & Molnár, P. (1995). Social force model for pedestrian dynamics. *Physical Review E*, 51(5), 4282–4286. <https://doi.org/10.1103/physreve.51.4282>
- Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., & Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7310–7311. <https://doi.org/10.1109/cvpr.2017.351>
- Huang, L., Qu, H., Fu, M., & Deng, W. (2018). Reinforcement learning for mobile robot obstacle avoidance under dynamic environments. In X. Geng & B.-H. Kang (Eds.), *PRICAI 2018: Trends in Artificial Intelligence* (pp. 441–453). Springer International Publishing. https://doi.org/10.1007/978-3-319-97304-3_34
- Jackel, L. D., Krotkov, E., Perschbacher, M., Pippine, J., & Sullivan, C. (2006). The DARPA LAGR program: Goals, challenges, methodology, and phase I results. *Journal of Field Robotics*, 23(11-12), 945–973. <https://doi.org/10.1002/rob.20161>
- Kim, Y.-H., Jang, J.-I., & Yun, S. (2018). End-to-end deep learning for autonomous navigation of mobile robot. *2018 IEEE International Conference on Consumer Electronics (ICCE)*, 1–6. <https://doi.org/10.1109/icce.2018.8326229>
- Kitani, K. M., Ziebart, B. D., Bagnell, J. A., & Hebert, M. (2012). Activity forecasting. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, & C. Schmid (Eds.), *Computer vision – ECCV 2012* (pp. 201–214). Springer. https://doi.org/10.1007/978-3-642-33765-9_15
- Konolige, K., Agrawal, M., Bolles, R., Cowan, C., Fischler, M., & Gerkey, B. (2008). Outdoor mapping and navigation using stereo vision. In O. Khatib, V. Kumar, & D. Rus (Eds.), *Experimental Robotics: The 10th International Symposium on Experimental Robotics* (pp. 179–190). Springer. https://doi.org/10.1007/978-3-540-77457-0_17
- Kretzschmar, H., Spies, M., Sprunk, C., & Burgard, W. (2016). Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research*, 35(11), 1289–1307. <https://doi.org/10.1177/0278364915619772>
- Kruse, T., Pandey, A. K., Alami, R., & Kirsch, A. (2013). Human-aware robot navigation: A survey. *Robotics and Autonomous Systems*, 61(12), 1726–1743. <https://doi.org/10.1016/j.robot.2013.05.007>
- Kushwaha, H. L., Sinha, J. P., Khura, T. K., Kushwaha, D. K., Ekka, U., Purushottam, M., & Singh, N. (2016). Status and scope of robotics in agriculture. *International Conference on Emerging Technologies in Agricultural and Food Engineering*, 12, 163.
- Lennon, C., Bodt, B., Childers, M., Camden, R., Suppé, A., Navarro-Serment, L., & Florea, N. (2013). *Performance evaluation of a semantic perception classifier* (tech. rep. ARL-TR-6653). Army Research Labs.

- Levine, S., Popovic, Z., & Koltun, V. (2011). Nonlinear inverse reinforcement learning with Gaussian processes. *Proceedings of Advances in Neural Information Processing Systems*, 19–27.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single shot MultiBox detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016* (pp. 21–37). Springer International Publishing. https://doi.org/10.1007/978-3-319-46448-0_2
- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431–3440. <https://doi.org/10.1109/cvpr.2015.7298965>
- Long, P., Fanl, T., Liao, X., Liu, W., Zhang, H., & Pan, J. (2018). Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 6252–6259. <https://doi.org/10.1109/icra.2018.8461113>
- Lookingbill, A., Rogers, J. G., III, Lieb, D., Curry, J., & Thrun, S. (2007). Reverse optical flow for self-supervised adaptive autonomous robot navigation. *International Journal of Computer Vision*, 74(3), 287–302. <https://doi.org/10.1007/s11263-006-0024-x>
- Massa, F., & Girshick, R. (2018). *maskrcnn-benchmark: Fast, modular reference implementation of instance segmentation and object detection algorithms in PyTorch* [Computer software]. GitHub. <https://github.com/facebookresearch/maskrcnn-benchmark>
- Mavrogiannis, C., Hutchinson, A. M., Macdonald, J., Alves-Oliveira, P., & Knepper, R. A. (2019). Effects of distinct robot navigation strategies on human behavior in a crowded environment. *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 421–430. <https://doi.org/10.1109/hri.2019.8673115>
- Mavrogiannis, C. I., & Knepper, R. A. (2018). Multi-agent path topology in support of socially competent navigation planning. *The International Journal of Robotics Research*, 38(2-3), 338–356. <https://doi.org/10.1177/0278364918781016>
- Mertz, C., Navarro-Serment, L. E., MacLachlan, R., Rybski, P., Steinfeld, A., Suppé, A., Urmson, C., Vandapel, N., Hebert, M., Thorpe, C., Duggins, D., & Gowdy, J. (2013). Moving object detection with laser scanners. *Journal of Field Robotics*, 30(1), 17–43. <https://doi.org/10.1002/rob.21430>
- Milella, A., Reina, G., & Underwood, J. (2015). A self-learning framework for statistical ground classification using radar and monocular vision. *Journal of Field Robotics*, 32(1), 20–41. <https://doi.org/10.1002/rob.21512>
- Murphy, R. R. (2014). *Disaster robotics*. The MIT Press. <https://doi.org/10.7551/mitpress/9407.001.0001>
- Nagatani, K., Kiribayashi, S., Okada, Y., Otake, K., Yoshida, K., Tadokoro, S., Nishimura, T., Yoshida, T., Koyanagi, E., Fukushima, M., et al. (2013). Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots. *Journal of Field Robotics*, 30(1), 44–63. <https://doi.org/10.1002/rob.21439>
- Narayanan, P., Yeh, B., Holmes, E., Martucci, S., Schmeckpeper, K., Mertz, C., Osteen, P., & Wigness, M. (2020, April). An integrated perception pipeline for robot mission execution in unstructured environments. In T. Pham, L. Solomon, & K. Rainey (Eds.), *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II* (pp. 447–465). SPIE. <https://doi.org/10.1117/12.2560699>
- Nasrinahar, A., & Chuah, J. H. (2018). Intelligent motion planning of a mobile robot with dynamic obstacle avoidance. *Journal on Vehicle Routing Algorithms*, 1(2-4), 89–104. <https://doi.org/10.1007/s41604-018-0007-4>
- Navarro-Serment, L. E., Mertz, C., & Hebert, M. (2010). Pedestrian detection and tracking using three-dimensional LADAR data. *The International Journal of Robotics Research*, 29(12), 1516–1528. https://doi.org/10.1007/978-3-642-13408-1_10
- Oh, J., Howard, T. M., Walter, M. R., Barber, D., Zhu, M., Park, S., Suppe, A., Navarro-Serment, L., Duvallat, F., Boularias, A., Romero, O., Vinokurov, J., Keegan, T., Dean, R., Lennon, C., Bodt, B., Childers, M., Shi, J., Daniilidis, K., . . . Stentz, A. (2017). Integrated intelligence for human-robot teams. *2016 International Symposium on Experimental Robotics*, 309–322. https://doi.org/10.1007/978-3-319-50115-4_28
- Oh, J., Suppé, A., Duvallat, F., Boularias, A., Navarro-Serment, L., Hebert, M., Stentz, A., Vinokurov, J., Romero, O., Lebiere, C., & Dean, R. (2015). Toward mobile robots reasoning like humans. *Proceedings of 29th AAAI Conference on Artificial Intelligence (AAAI '15)*, 1371–1379.
- Okal, B., & Arras, K. O. (2016). Learning socially normative robot navigation behaviors with bayesian inverse reinforcement learning. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2889–2895. <https://doi.org/10.1109/ICRA.2016.7487452>

- Pandey, A., Kashyap, A. K., Parhi, D. R., & Patle, B. K. (2019). Autonomous mobile robot navigation between static and dynamic obstacles using multiple ANFIS architecture. *World Journal of Engineering*, 16(2), 275–286. <https://doi.org/10.1108/wje-03-2018-0092>
- Pandey, A., Pandey, S., & Parhi, D. R. (2017). Mobile robot navigation and obstacle avoidance techniques: A review. *International Robotics & Automation Journal*, 2(3), 00022. <https://doi.org/10.15406/iratj.2017.02.00023>
- Pandey, A., Panwar, V. S., Hasan, M. E., & Parhi, D. R. (2020). V-REP-based navigation of automated wheeled robot between obstacles using PSO-tuned feedforward neural network. *Journal of Computational Design and Engineering*, 7(4), 427–434. <https://doi.org/10.1093/jcde/qwaa035>
- Papadakis, P. (2013). Terrain traversability analysis methods for unmanned ground vehicles: A survey. *Engineering Applications of Artificial Intelligence*, 26(4), 1373–1385. <https://doi.org/10.1016/j.engappai.2013.01.006>
- Pfeiffer, M., Schaeuble, M., Nieto, J., Siegwart, R., & Cadena, C. (2017). From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 1527–1533. <https://doi.org/10.1109/icra.2017.7989182>
- Pfeiffer, M., Schwesinger, U., Sommer, H., Galceran, E., & Siegwart, R. (2016). Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. <https://doi.org/10.1109/iros.2016.7759329>
- Ratliff, N. D., Bagnell, J. A., & Zinkevich, M. A. (2006). Maximum margin planning. *Proceedings of the 23rd International Conference on Machine Learning - ICML '06*, 729–736. <https://doi.org/10.1145/1143844.1143936>
- Ratliff, N. D., Silver, D., & Bagnell, J. A. (2009). Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots*, 27(1), 25–53. <https://doi.org/10.1007/s10514-009-9121-3>
- Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7263–7271. <https://doi.org/10.1109/cvpr.2017.690>
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/tpami.2016.2577031>
- Roncancio, H., Becker, M., Broggi, A., & Cattani, S. (2014). Traversability analysis using terrain mapping and online-trained terrain type classifier. *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 1239–1244. <https://doi.org/10.1109/ivs.2014.6856427>
- Sadat, A., Casas, S., Ren, M., Wu, X., Dhawan, P., & Urtasun, R. (2020). Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. *Computer Vision – ECCV 2020*, 414–430. https://doi.org/10.1007/978-3-030-58592-1_25
- Schaefer, K. E., Oh, J., Aksaray, D., & Barber, D. (2019). Integrating context into artificial intelligence: Research from the Robotics Collaborative Technology Alliance. *AI Magazine*, 40(3), 28–40. <https://doi.org/10.1609/aimag.v40i3.2865>
- Schönberger, J. L., & Frahm, J.-M. (2016). Structure-from-motion revisited. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4104–4113. <https://doi.org/10.1109/cvpr.2016.445>
- Segal, A., Haehnel, D., & Thrun, S. (2009). Generalized-ICP. *Robotics: Science and Systems*, 4. <https://doi.org/10.15607/rss.2009.v.021>
- Shamshiri, R. R., Weltzien, C., Hameed, I. A., Yule, I. J., Grift, T. E., Balasundram, S. K., Pitonakova, L., Ahmad, D., & Chowdhary, G. (2018). Research and development in agricultural robotics: A perspective of digital farming. *International Journal of Agricultural and Biological Engineering*, 11(4), 1–14. <https://doi.org/10.25165/j.ijabe.20181104.4278>
- Shao, F., Cai, S., & Gu, J. (2010). A modified Hausdorff distance based algorithm for 2-dimensional spatial trajectory matching. *Proceedings of the International Conference on Computer Science and Education*, 166–172. <https://doi.org/10.1109/iccse.2010.5593666>
- Shneier, M., Chang, T., Hong, T., Shackelford, W., Bostelman, R., & Albus, J. S. (2008). Learning traversability models for autonomous mobile vehicles. *Autonomous Robots*, 24(1), 69–86. <https://doi.org/10.1007/s10514-007-9063-6>
- Silver, D., Bagnell, J. A., & Stentz, A. (2008). High performance outdoor navigation from overhead data using imitation learning. *Robotics: Science and Systems*. <https://doi.org/10.15607/rss.2008.iv.034>

- Silver, D., Bagnell, J. A., & Stentz, A. (2010). Learning from demonstration for autonomous navigation in complex unstructured terrain. *International Journal of Robotics Research*, 29(12), 1565–1592. <https://doi.org/10.21236/ada525288>
- Simonyan, K., & Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition*. arXiv: 1409.1556 [cs.CV].
- Singh, A., Yang, L., Hartikainen, K., Finn, C., & Levine, S. (2019). *End-to-end robotic reinforcement learning without reward engineering*. arXiv: 1904.07854 [cs.LG].
- Siva, S., Wigness, M., Rogers, J. G., III, & Zhang, H. (2019). Robot adaptation to unstructured terrains by joint representation and apprenticeship learning. *Robotics: Science and Systems XV*. <https://doi.org/10.15607/rss.2019.xv.030>
- Soria, L. M., Ortega, F. J., Álvarez-García, J. A., Velasco, F., & Fernández-Cerero, D. (2020). How efficient deep-learning object detectors are? *Neurocomputing*, 385, 231–257. <https://doi.org/10.1016/j.neucom.2019.10.094>
- Suger, B., Steder, B., & Burgard, W. (2015). Traversability analysis for mobile robots in outdoor environments: A semi-supervised learning approach based on 3D-LiDAR data. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 3941–3946. <https://doi.org/10.1109/icra.2015.7139749>
- Tai, L., Li, S., & Liu, M. (2016). A deep-network solution towards model-less obstacle avoidance. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2759–2764. <https://doi.org/10.1109/iros.2016.7759428>
- Tai, L., Paolo, G., & Liu, M. (2017). Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 31–36. <https://doi.org/10.1109/iros.2017.8202134>
- Tai, L., Zhang, J., Liu, M., Boedecker, J., & Burgard, W. (2016, December). *A survey of deep network solutions for learning control in robotics: From reinforcement to imitation*. arXiv: 1612.07139 [cs.R0].
- Talukder, A., Manduchi, R., Castaño, R., Owens, K., Matthies, L., Castano, A., & Hogg, R. (2002). Autonomous terrain characterisation and modelling for dynamic control of unmanned vehicles. *2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1, 708–713. <https://doi.org/10.1109/IRDS.2002.1041474>
- Taş, O., Hörmann, S., Schäufele, B., & Kuhnt, F. (2017). Automated vehicle system architecture with performance assessment. *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 1–8. <https://doi.org/10.1109/ITSC.2017.8317862>
- Trautman, P., Ma, J., Murray, R. M., & Krause, A. (2015). Robot navigation in dense human crowds: Statistical models and experimental studies of human-robot cooperation. *The International Journal of Robotics Research*, 34(3), 335–356. <https://doi.org/10.1177/0278364914557874>
- Trevor, A. J. B., Rogers, J. G., III, & Christensen, H. I. (2014). OmniMapper: A modular multimodal mapping framework. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 1983–1990. <https://doi.org/10.1109/icra.2014.6907122>
- Tripathi, S., Chandra, S., Agrawal, A., Tyagi, A., Rehg, J. M., & Chari, V. (2019). Learning to generate synthetic data via compositing. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 461–470. <https://doi.org/10.1109/cvpr.2019.00055>
- Tsai, C.-E., & Oh, J. (2020). A generative approach for socially compliant navigation. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2160–2166. <https://doi.org/10.1109/icra40945.2020.9197497>
- Turnwald, A., & Wollherr, D. (2018). Human-like motion planning based on game theoretic decision making. *International Journal of Social Robotics*, 11(1), 151–170. <https://doi.org/10.1007/s12369-018-0487-2>
- Vasquez, D., Okal, B., & Arras, K. O. (2014). Inverse reinforcement learning algorithms and features for robot navigation in crowds: An experimental comparison. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1341–1346. <https://doi.org/10.1109/iros.2014.6942731>
- Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples. *ACM Computing Surveys*, 53(3), 1–34. <https://doi.org/10.1145/3386252>
- Wang, Y.-X., Ramanan, D., & Hebert, M. (2019). Meta-learning to detect rare objects. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9924–9933. <https://doi.org/10.1109/iccv.2019.01002>
- Wigness, M., Rogers, J. G., III, & Navarro-Serment, L. E. (2018). Robot navigation from human demonstration: Learning control behaviors. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1150–1157. <https://doi.org/10.1109/icra.2018.8462900>

- Wulfmeier, M., Wang, D. Z., & Posner, I. (2016). Watch this: Scalable cost-function learning for path planning in urban environments. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2089–2095. <https://doi.org/10.1109/iros.2016.7759328>
- Xiao, X., Liu, B., Warnell, G., & Stone, P. (2020, November). *Motion control for mobile robot navigation using machine learning: A survey*. arXiv: 2011.13112 [cs.R0].
- Zhang, J., Springenberg, J. T., Boedecker, J., & Burgard, W. (2017). Deep reinforcement learning with successor features for navigation across similar environments. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2371–2378. <https://doi.org/10.1109/iros.2017.8206049>
- Zhang, J., Tai, L., Liu, M., Boedecker, J., & Burgard, W. (2020). *Neural SLAM: Learning to explore with external memory*. arXiv: 1706.09520 [cs.LG].
- Zhelo, O., Zhang, J., Tai, L., Liu, M., & Burgard, W. (2018, April). *Curiosity-driven exploration for mapless navigation with deep reinforcement learning*. arXiv: 1804.00456 [cs.R0].
- Zhou, X., Gao, Y., & Guan, L. (2019). Towards goal-directed navigation through combining learning based global and local planners. *Sensors*, 19(1), 176. <https://doi.org/10.3390/s19010176>
- Zhu, Y., Mottaghi, R., Kolve, E., Lim, J. J., Gupta, A., Fei-Fei, L., & Farhadi, A. (2017). Target-driven visual navigation in indoor scenes using deep reinforcement learning. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 3357–3364. <https://doi.org/10.1109/icra.2017.7989381>
- Ziebart, B. D., Maas, A. L., Bagnell, J. A., & Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, 1433–1438.
- Ziebart, B. D., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J. A., Hebert, M., Dey, A. K., & Srinivasa, S. S. (2009). Planning-based prediction for pedestrians. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3931–3936. <https://doi.org/10.1109/iros.2009.5354147>

How to cite this article: Wigness, M., Rogers, J. G., III, Tsai, C.-E., Mertz, C., Navarro-Serment, L., & Oh, J. (2021). Using perception cues for context-aware navigation in dynamic outdoor environments. *Field Robotics*, 1, 1–33.

Publisher's Note: Field Robotics does not accept any legal responsibility for errors, omissions or claims and does not provide any warranty, express or implied, with respect to information published in this article.