International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

Cooperative Robot-Drone Agents for Obstacle Avoidance using Smart Vision

Taylor Ripke, Kellen Reason, and Tony Morelli
Department of Computer Science
Central Michigan University
Mt Pleasant, MI 48858

Abstract **- Recent advancements in robotics and computer science are pushing the boundaries of exploration and our understanding of intelligent systems. These systems have the potential to interact with one another to solve difficult tasks, such as exploring a new planet or providing emergency assistance. Our research focuses on studying the interaction between an autonomous robot and drone and how they perceive and understand the environment to perform a given task. Utilizing advanced path-planning algorithms, image processing techniques, and various sensors for localization, the robot and drone can interact with the environment to solve generic tasks, such as finding an optimal path around a series of obstacles if the path is blocked utilizing a Greedy approach. Furthermore, the system is an adaptive learning model. The drone autonomously takes off and lands following the directions from the robot, increasing battery duration while providing the robot with additional sensory information. The robot and drone are two separate intelligent systems that work together solving different tasks utilizing the same dataset for learning and interacting.**

Keywords: Localization, Path-Planning, Cooperative-Agents, Navigation

INTRODUCTION

Autonomous systems are becoming more prevalent as technology continues to exponentially grow and deep networks are utilized to train machines to certain situations. In machine learning, researchers train the robot to generate statistical probabilities of an event given some input. Artificial intelligence takes it one step further using the results generated to make a decision to further advance the system. Therefore, an autonomous system needs to be able to reason about its environment using the data it has previously gathered to complete its goal. However, in the process, the autonomous system's framework should allow for unsupervised learning for a system is to be considered truly intelligent.

Figure 1: Titan: Cooperative Robot-Drone Agents

The following paper introduces Project Titan consisting of Atlas (All-Terrain Localizing Autonomous System) (robot) and Calypso (drone) that work together to solve a specific task as shown in Fig 1. Our work was greatly inspired by the revolutionary work done by [10,17]. [17] demonstrates a fully autonomous system capable of delivering medical supplies to a person in a mock-up disaster scenario [17]. [10] presents an approach for object avoidance utilizing the Wavefront technique mapping algorithm for path planning. Titan is a cooperative system that employs a robot as the central hub and a drone providing assistance. The drone is responsible for serving as a second pair of eyes for the robot. [17] utilizes an on-board computer on the drone to find obstacles and relay the information to the robot. [10] utilizes an overhead camera for path planning assistance.

Dissimilar to [17], our system does not implement a feature to remove obstacles from the generated path utilizing a robotic arm demonstrated in [17]. Therefore, our system requires the robot to find a path around the obstacles if possible. Path planning is the challenging task of finding the shortest path through a series of obstacles once detected. The objects do not need to be recognized, but their position, orientation, and boundaries must be determined. There are multiple ways to achieve this task, including Canny edge detection and color thresholding. Once the objects are detected, their position relative to the robot and end goal must be determined to generate a sequential movement plan for the robot while checking for new environment variables. Many autonomous robots utilize stereo or monocular vision

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

on the robot to plan their movements, giving the robot a forward sense of direction. Object occlusion presents a problem for these algorithms as some objects may be blocking others that would interfere with the local path generated by the robot. In general, there are four steps in autonomous navigation summarized by [14]:

1. Perceiving and modeling the environment.
2. Localizing the vehicle within the environment.
3. Planning and deciding the vehicle's desired motion.
4. Executing the vehicle's desired motion.

To remedy the problem of object occlusion and low confidence, the robot is accompanied by an autonomous drone that is responsible for mapping out terrain ahead of the robot. In addition to the forward-facing camera on the robot, the drone allows for a top-down aerial view of the terrain in front of the robot. The aerial perspective is preferred because the drone is capable of seeing all objects within its range of view. The robot's forward-facing camera only provides a limited range of viewing at an angle and essential depth information is lost.

Our work specifically focuses on the cooperative interaction between two intelligent systems tasked with finding the optimal shortest path utilizing the data gathered by both. Image processing is done initially on the robot's forward facing camera and supplemented by the drone's ground camera if the robot is not able to generate a path to the end goal or a path with a low confidence is generated. If either of these conditions are present, the robot launches the drone to perform terrain localization and mapping, which is processed locally on the robot and fed to the central hub that makes the final decision. The drone is equipped with autonomous landing utilizing similar vision processing techniques.

The following paper is organized as follows. Section II describes related work in path-planning, localization, and cooperative-agents. Section III provides details on the implementation of the robot and drone image processing and localization techniques. Section IV depicts the current system in practice followed by discussion and future research.

LITERATURE REVIEW

Intelligent systems encompass the challenges current and historic mathematicians, biologists, psychologists, and neuroscientists strive to understand and reverse engineer. The roots of artificial intelligence some contend spurred from the 1956 Dartmouth Summer Research Project on Artificial Intelligence (AI) where pioneers of the field met to discuss current and future research in the accelerating field [1]. The workshop was based on a proposal in 1955 that outlined research plans to study automatic computers, neuron nets, abstractions, and self-improvement [2]. The field of computer science and artificial intelligence has grown exponentially since its creation; however, the current challenges researchers face echo its origin. The specific work focused on in this paper encompasses these fundamental ideas in relation to an intelligent system.

An *intelligent system*, according to our working definition, is any entity capable of reasoning about its environment abstractly, envisioning and creating solutions to problems, acting independently, but respective to other objects and entities in the environment. It is synonymous with the core concepts of an *autonomous vehicle* or *system*, which can be summarized as a ``self-acting and self-regulating" and ``able to operate in and react to its environment without outside control" [14]. Irrespective to the complexity of a given task, the autonomous robot should provide a generalized solution, whether it is completing the task or determining that the goal is unobtainable.

The following literature focuses on relevant path planning algorithms. It is important to note the this review encompasses the most prevalent, although there are many more solutions. The most notable algorithms can be divided into two categories: deterministic/heuristic-based algorithms and probabilistic/sampling-based algorithms [13]. These algorithms are *anytime* algorithms which are ``able to trade off running time and solution quality in domains where quick reactions are required" [13]. However, utilizing the solutions to sub-problems, better results are directly proportional to the amount of time given. For example, [14] introduces the ARA* algorithm that handles the drawbacks of traditional A* and [15] utilizes Rapidly-exploring Random Trees (RRTs) to quickly generate a solution and continue to improve given enough time. In this paper our implementation assumes a static environment but has features to compensate for dynamic events in the environment, such as a person suddenly walking in front of the robot.

A and D* Variants*

The A* algorithm is a popular path finding algorithm designed to move from one position to another without colliding with an obstacle [18]. It ``uses heuristic knowledge in form of approximations of the goal distances to focus the search and solve search problems much faster than uninformed search methods" [20]. It has been used in many autonomous projects, for example in [17]. Any change in the environment that would affect the graph forces the algorithm to recompute. However, Trovato notes that typically only a few nodes or arrows change. Exploiting this fact, the Differential A* algorithm was developed that was designed ``to adapt the pre-exisiting stable solution-graph to the new changes and produce a new solution-graph to one that would have been generated from complete initialization and A*" [18]. The original A* algorithm has many modifications, including LPA* (Lifelong Planning A*) [19], which lead to the creation of D* Lite around 2002 [20]. D* Lite is similar to its predecessor D*, but different algorithmically as it ``uses only one tie-breaking criterion when comparing priorities" [20].

Another branch from the original A* algorithm was the Focused D* Algorithm which is a ``full generalization of A* for dynamic environments" [21]. In an environment where variables may be changing, the A* algorithm needs to

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

recompute a path which can be computationally expensive. The autonomous system must either pause or risk running into an object on the wrong path. Stentz stresses the importance of rapid re-planning and discusses how ``The D* algorithm (Dynamic A*) plans optimal traverses in real-time by incrementally repairing paths to the robot's state as new information is discovered" [21]. It is important to note that the Focus Dynamic A* (D*) can ``achieve a speedup of one to two magnitudes(1) over repeated A*" [23].

Furthermore, following the developments of D* and D* Lite, the Delayed D* algorithm was created to solve the same problems, but be significantly more efficient [22]. It is important to note that these algorithms can be application specific and many options should be considered before concrete implementation. For example, Koenig and Likhachev describe how LPA*, D*, and D* Lite have several disadvantages, such as it being ``difficult to prove them correct and that there are situations where they are slower than standard A*"[24]. Therefore, they proposed Adaptive A* around 2006 that ``expands no more states than standard A* and thus cannot be slower than standard A* (except for a small number of bookkeeping actions)" [24]. Further work has been done, such as 3D Field D* that expends from D* and D* Lite algorithms ``that uses interpolation to produce less costly paths through 2D grids" [25]. Anavatti et al. provide a more detailed description of these dynamic re-planning algorithms [12].

Bug Algorithm Family
The Bug algorithm family is a solution to navigating unknown environments without having to generate a map of the environment or store any information. An introduction to the family of Bug algorithms is described in [13]. A generic summary of a Bug algorithm is to move towards the end goal until an object is encountered. The robot follows the perimeter of the object (either left or right) keeping track of the optimal distance to the robot. Once a direct path towards the object is present, the robot will break from the object and head towards the end goal. The Bug algorithm family makes three assumptions about the robot [13]:

1. The robot is a point object.
2. The robot has perfect localization ability.
3. The robot has perfect sensors.

This approach eliminates the need to generate a map of the environment and instead focus on the most immediate points relative to the goal. The algorithm does have the ability to recognize if the goal is not reachable and will terminate [13].

Road Maps: Rapidly exploring random trees (RRT)
Anavatti et al. introduce multiple road map methodologies and some of its models [12]. They defined a road map as ``a graph that finds connections between a robot's free spaces as a set of one-dimensional (1D) curves" [12]. Our research was interested in a specific type of road map known as a *Rapidly exploring random trees (RRT)*. In the RRT model, ``a planner beings at the start location and randomly expands a path, or tree, to cover the configuration space away from previously constructed vertices', thereby allowing the planner to rapidly search large and high-dimensional spaces" [12].

A RRT contains similar proprieties to a probabilistic road map [26]. For example, ``both are designed with as few heuristics and arbitrary parameters as possible" which ``tends to lead to better performance analysis and consistency of behavior" [26]. LaValle also describes the unique advantage of RRTs being that ``they can be directly applied to nonholonomic and kinodynamic planning" [26]. Further proprieties of RRTs can be read in [26]. This method can be further expanded by growing trees from the goal and start until common ground is discovered to create a link [27].

Potential Fields
In potential fields, the robot can be viewed as a magnet being pulled to its counterpart (goal) while avoiding forces that push it away. The forces of the applied to the robot determine the respective direction of motion [12]. The robot is ultimately attracted to the end goal; however, it is influenced more by the local forces than the global forces [12]. A strong repulsive force, such as an object directly in front of the robot, will cause the robot to react drastically altering its course. [12] describes how local minima can prevent the robot from reaching its goal. Intricit details of these potential fields can be explored in [28,29].

Brief Localization Overview
Localization and mapping of an unstructured, noisy environment has seen numerous advances; however, it remains a prominent challenge for current autonomous mobile robots. Research from Smith and Cheeseman in the late 1980s [4,5] created the foundation for SLAM or simultaneous localization and mapping that was further developed by Whyte et al. in the early 1990s [6-8]. Subsequent work has been done utilizing CML or concurrent mapping and localization [9].

In general, path planning algorithms strive to answer three questions [12]:

1. Completeness - Does a solution exist?
2. Optimality - Is finding the lowest cost path guaranteed?
3. Time Complexity - How long does it take to find a solution?

Following processing of the initial images taken by the robot, is an apparent solution exist, whether it is a global or local optimum? Depending on the situation, steps can be taken to achieve local end goals that will eventually converge to the global goal. If the global goal is not present within the image, the robot should try to determine a local solution and proceed in any movement necessary to reduce the distance to the goal. However, in some situations, a local solution may not be initially present. For example, consider a situation where a small wall or series of obstacles completely blocks the path of the robot to its end goal. There are multiple solutions to this problem. If the robot had access

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

to a vertical ground-facing camera, it could evaluate the environment from above. However, this is not applicable in practice in an unknown environment, such as planet exploration.

Cooperative-Agents

The benefits of exploiting the advantages of unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) are gaining traction in the field of autonomous systems. Saska et al. outline the general consensus of UAVs and UGVs. UAVs are characterized by short-flight durations (approximately 10 minutes) and limited payloads [32]. Furthermore, UGVs are capable of running for extended periods of time and carrying heavy payloads. However, given the size of the UGV, ``their movement is limited by obstacles and terrain traversability'' [32]. These cooperative systems are capable of exploration, providing search and rescue assistance, and much more.

Generalized Purpose

Saska et al. introduced a cooperative system capable of visiting places of interest [32]. The system consists of a primary UGV whose navigation is initially trained by a security guard who guides the robot that generates a map along the way. Once trained, the robot is capable of traversing the learn path autonomously [32]. To get the most out of their UAV, the UAV only performs its tasks of scanning the area of interest and then returning to land on the helipad on the UGV. The UAV spends limited time in the air to assure that the system can support many more flights than continuous flying. It only becomes active to assess areas inaccessible by the UGV [32]. Our implementation features a similar setup; however, our focus is primarily on object detection and avoidance utilizing efficient path planning algorithms.

Cantelli et al. also present a system that utilizes a UAV that autonomously follows a ground robot [33]. The motivation for their system was supported by the need for an increased field of view to best decide the subsequent navigation strategy [33]. The operator is only responsible for controlling the UGV while the UAV collects images to build traversability maps [33]. Titan employed similar image processing techniques to detect objects in images using thresholding, white/back pixel extraction, and noise removal [33]. Further information about similar systems is extensively discussed in their paper [33].

Cooperative agents are becoming more prevalent as shown in current literature, although there is much to do. Cooperative systems entail two or more entities working together to solve a task. Our robot-drone system is one example that is currently being researched. Hausman et al. describe a system of multiple robots that ``estimate the position of a moving target using on board sensing'' [34]. For example, multiple drones could be responsible for tracking two ground robots. They briefly outline the benefits being a ``reduction in tracking uncertainty, increased coverage, and robustness to failure'' [34]. Hsieh et al. introduce a framework where a human operator can ``deploy a heterogeneous team of autonomous air and ground robots to cooperatively execute tasks'', such as target search and surveillance ``within an urban environment while providing high-level situational awareness for a remote human operator'' [35]. Garzon et al. present a similar system to our own that maps a large, unknown outdoor environment [36]. A drone flies above the robot to providing additional information to the robot for localization and mapping [36]. Our research is similar, however, it is primarily more focused on the path-planning and did not incorporate GPS information into our model.

Search and Rescue Applications

Among recent advances in cooperative autonomous systems, the range of applications is still being realized. They have been used to find simulated mines [37] and have been considered for planetary exploration [38]. Furthermore, they have been used in various search and rescue scenarios. As discussed previously, part of our work was motivated by research conducted by [17], however, there are many more examples of similar situations. For example, rescuers utilized UGVs and UAVs to access damage to buildings and other areas following two major earthquakes in Mirandola, Italy [39]. A few more examples of relevant literature concerning autonomous robots in disaster and search/rescue situations can be read in [40,41].

IMPLEMENTATION

Titan was originally conceived as an exploration robot stemming from its capabilities to navigate rough terrain and avoid dynamic obstacles utilizing a drone and advanced computer vision mapping and object detection algorithms. The past decade has brought about a new era of exploration as NASA deployed and landed Curiosity, a manual and autonomous rover, on Mars to explore the origins of life. The rover has the capability to receive location commands, but it is ultimately up to the robot to make the appropriate movements to reach the goal, subsequently avoiding obstacles if present. Similarly, our mobile robot has the capability to navigate an unknown environment utilizing onboard sensors and vision.

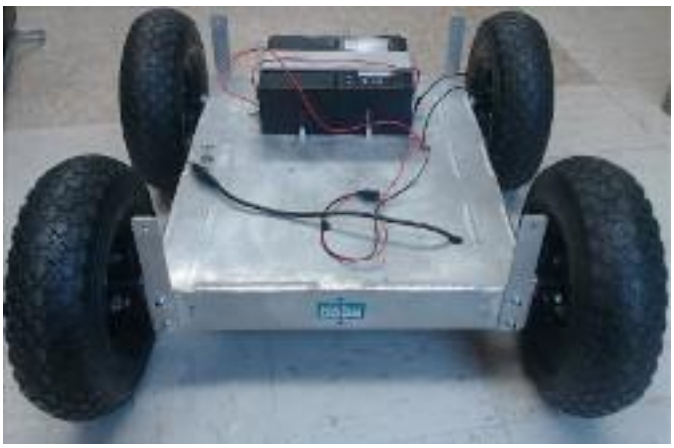


Figure 2: IG42-SB4, 4WD All-Terrain Robot

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

Figure 3: (Left) Sabertooth 2x25 Motor Controller, (Right) Arduino Mega 2560

```
checksum = 127 & (130 + 0 + motor1Speed)
packet = struct.pack('BBBB',130,0,motor1Speed,checksum)
ser.write(packet)
checksum = 127 & (130 + 4 + motor2Speed)
packet = struct.pack('BBBB',130,4,motor2Speed,checksum)
ser.write(packet)
```

Figure 4: Example of Packetizied-Serial commands

The robot is accompanied by a drone companion that serves as another pair of 'eyes' for the robot. The robot and drone are separate entities that work together to solve a given task. For example, if there were a series of obstacles ahead of the robot, the drone would be launched to map out the terrain ahead and plan a path through the obstacles utilizing a combination of localization and grid mapping using a Greedy Best-First Search approach. Information from the drone is relayed to the on-board computer on the robot that processes the images which pipes the information to the central hub. Both the robot and drone entities are connected through the central hub which makes the final decision of movement based on the information received from the robot and drone.

Robot

Titan is a custom built exploration robot that navigates autonomously utilizing a drone and on-board sensors. The robot is an IG42-SB4, 4WD All Terrain Robot Platform purchased from SuperDroid Robots as shown in Fig 2. The robot is powered by two 12V batteries in series and has the capability of moving approximately 25 lbs [30]. The robot has four IG42 24VDC 078 RPM Gear Motors that move each of the 10 inch wheels.

The motors are controlled via the Sabertooth Dual 25A motor driver utilizing Packetized serial communication as shown in (Left) Fig 3. the robot can be controlled using only one line of communication in S1 on the motor controller. Simplified Serial requires approximately two seconds between commands, which was not feasible for our project. However, a drawback to using Packetized Serial Model is that is a one-direction interface. No information can be recieved from the motor controller. For the purposes of our implementation, the robot's behavior was acceptable and there was a need to correct utilizing information from the attachable motor encoders. Furthermore, Packetized Serial allows for multiple motors controllers to be connected as the commands are directed to each respective device using the address byte. An example of how to send a movement command to the robot can be viewed in Fig 4.

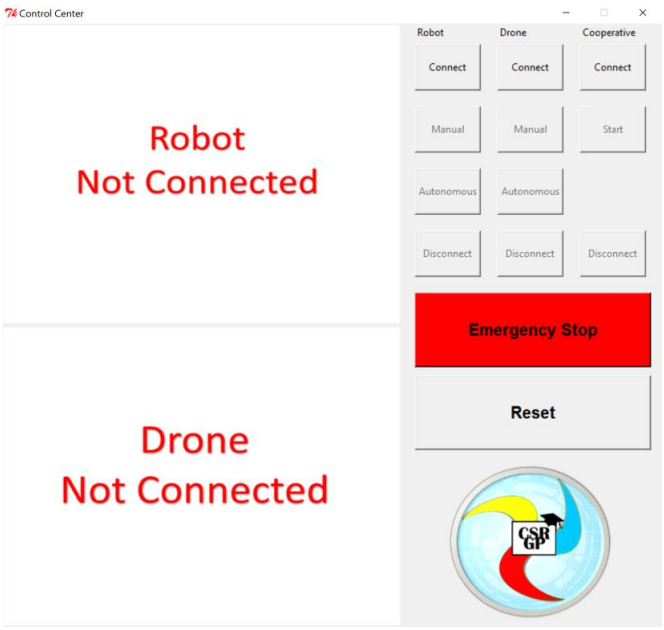


Figure 5: Client GUI for interacting with the system

Figure 6: AR.Drone 2.0

The central hub on the robot is a Raspberry Pi 3. It is responsible for handling requests from the user and making the final decision regarding autonomous actions. The robot can be controlled in two ways. It can be controlled manually from a computer on the local network or can operate autonomously, moving forward and avoiding obstacles until it is stopped. For manual control, the client launches a GUI programmed using the Tkinter module that is used to connect to the robot as shown in Fig 5. First, a TCP connection is established between the robot and client. Next, a JpegStream is initialized on the Pi using SimpleCV. The feed from the Logtech C920e camera attached to the robot is displayed in the GUI and can be accessed by anyone on the local network that has the IP address of the Pi.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

Once the client has successfully connected and the stream is visible, the user can manually drive the robot. In instances where the network connection is suddenly dropped, the robot is programmed to turn around and head back in the direction of its previous connection and reestablish. The GUI also gives the client the ability to enter the system into autonomous mode, where the robot will begin moving towards a goal, for example a red circle. The details of the implementation are discussed in an upcoming section. The Raspberry Pi 3 is connected via USB to an Arduino Mega 2560, which serves as a direct communication to the motor controller as shown in (Right) Fig 3.

AR Drone 2.0

The drone used to accompany Atlas is the AR Drone 2.0 by Parrot Inc. The platform is very lightweight at 390 grams with the internal frame and 420 grams with the external frame [31]. The drone has a 720p 30fps front facing camera. It also has a ground-facing camera that utilized for positioning and object detection. The drone has an ARM Cortex A8 1 Ghz 32-bit processor and it communicates with via a Linux machine [31]. The drone is a flying hot-spot and is easily connected with the Linux machine (Raspberry Pi 3) for control. It includes a variety of different sensors including an Accelerometer, Magnetometer, Pressure sensor, Altitude ultrasound sensor, and vertical camera [31].

The AR drone 2.0 is a relatively stable platform that was used to capture images from above for the robot. As shown in Fig 6, the drone resides on a wooden platform on top of the robot. On the wooden platform is a red circle, which is what the robot utilizes to autonomously land. The autonomous landing is performed by thresholding the image to detect the red circle, finding the respective contours in the image and their center, and finally minimize the distance between the center coordinates of the image and the position of the red circle.

Robot Path Planning

The robot path planning algorithm employs two generic strategies for solving the problem of finding the shortest path. As discussed in our literature review, there are many ways to solving the path finding problem and each of them has their advantages and disadvantages. The motivation for our approach comes from [10,17]. The first part of the algorithms stems from our own creation of a three parameter check. First the robot checks if something is in front of it. If there is, it checks the left and right at a respective 45 degree angle. If not object is immediately present within one of the other two vectors, the robot turns ands heads in that direction as a novel form of obstacle avoidance. As shown in the second and third images in Fig 7, the detection algorithm is capable of detecting single and multiple objects in the scene and their position relative to the robot. The green/red box is the respective width of the robot. As long as no objects are within the bounds of that box, it is assumed that the robot can pass safely through. However, as stated previously, if an object is present, more measures need to be taken.

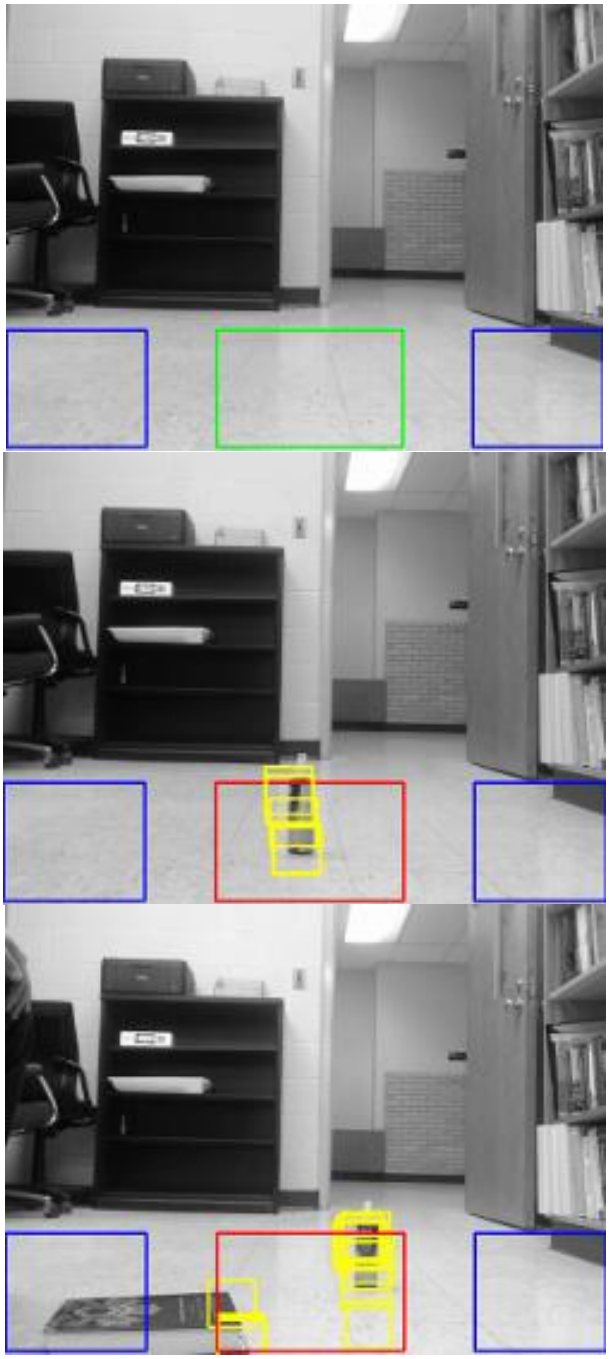
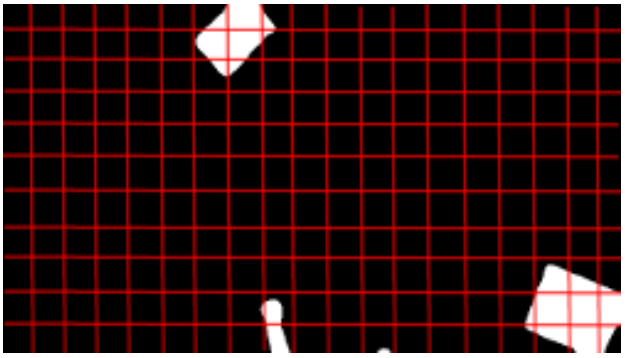


Figure 7: Top: No object detected, Middle: Single object detected, Bottom: Multiple objects detected



International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

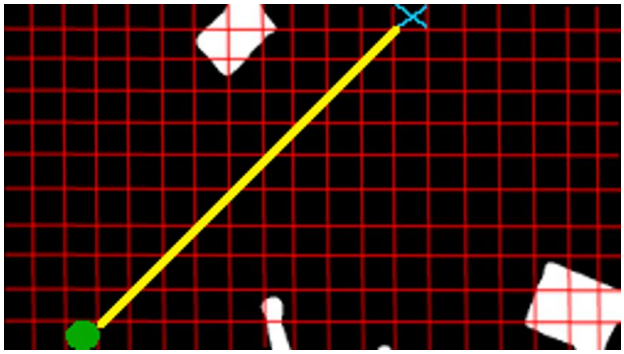


Figure 8: Grid approach inspired by [10] utilizing Best-First Search

DRONE PATH PLANNING

The motivation for Titan's object detection system was partially motivated by the research conducted by Chandak et al. utilizing the Wavefront technique mapping algorithm for efficient object detection and path planning [10]. Our approach uses a drone with a ground-facing camera to take pictures of the terrain in front of the robot. The image received from the drone undergoes a series of steps depicted in Fig 9. First, a Gaussian blur is applied to the image to reduce noise and provide more accurate contour detection. Second, the image undergoes thresholding utilizing OpenCV's inRange function. The color of the floor is initially known and the respective HSV values were computed. The noise variation in the floor was eliminated by the Gaussian blur. The filter is applied to the image and any object that is not of that color will be white in the output image. Following Chandak et al. approach, a grid is imposed abstractly onto the image. Any sector containing white is deemed to have an object. The grid is not present in the actual computation. It was added for visual representation.

Our approach currently implements a Greedy Best-First Search algorithm to find the quickest and shortest path to the local end. As depicted in Fig 8, the green circle represents the center of the robot and the blue 'x' represents the ending position. This is the local end goal. The drone was flying at approximately 160cm. The global goal is to reach the red circle which the drone can see with its forward facing camera. The drone flies ahead of the robot and takes a picture of the terrain for the robot to analyze and compute a path to the other side of the image without hitting any obstacles. If such a path is not present, given the number of obstacles detected in the image utilizing OpenCV's contour functions, the drone will fly higher to capture more of the scene, or an A* algorithm utilized by [17] in their mock-disaster scenario will be employed.

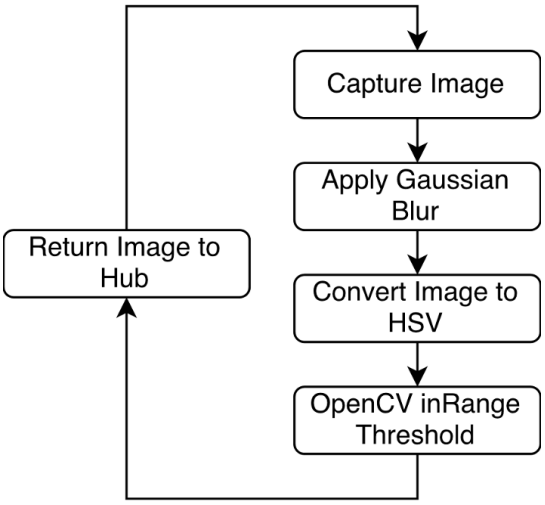


Figure 9: Generic operation to detect objects and landing platform

The end goal of the robot is local, as it is a further means of reaching the final goal. The local goal of the robot is to traverse the empty grid while avoiding objects. The program use an A* algorithm to find the shortest path to the goal, motivated by similar work conducted by Geus et al [11]. The AR parrot drone flies at approximately 73cm from the ground, which allows us to calculate the approximate size of each element of the matrix in inches. Therefore, given the current orientation of the robot, a series of movement commands are generated sequentially and fed to the motor controller.

CONCLUSION

The future of Project Titan is bright as many projects are currently under development. A future paper will describe the autonomous landing portion of the drone on the moving platform including a detailed analysis of current techniques. In addition, although the primary purpose is exploration, it is also designed to be social, learning robot. In future iterations, the robot will be able to recognize and not just detect objects utilizing Convolutional Neural Networks built using Keras and a Tensorflow backend. Similar to research currently being conducted in the autonomous car industry, the CNN should be able to recognize objects that can and cannot be run over. The robot currently stops at all objects, even if it is a piece of paper lying on the ground. To OpenCV and our image processing techniques, it is equivalent to a large rock. The ability to 'think' abstractly about an object is crucial to the success of future object detection/recognition tasks.

Furthermore, future iterations of the system will have multiple, smaller drones rather than one larger one. Significantly more data can be gathered using three vertical cameras than a single one. If the robot needed to be employed in a search and rescue operation, three independent drones have a much better chance at finding a person or object quickly than a single drone. Three independent drones also widens the connectivity of the cluster.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

REFERENCES

[1] R. J. Solomonoff, "The time scale of artificial intelligence: Reflections onsocial effects,"Human Systems Management, vol. 5, no. 2, pp. 194-153,1985.
[2] J. McCarthy, M. L. Minsky, N. Rochester, C. E. Shannon, "A Proposalfor the Dartmouth Summer Research Project on Artificial Intelligence,"AI Magazine, vol. 27, no. 4, 2006.
[3] N. J. Nilsson, "Shakey the Robot," SRI INTERNATIONAL, Tech. Rep.323, 1984.
[4] R. C. Smith, P. Cheeseman, "On the Representation and Estimation ofSpatial Uncertainty,"The International Journal of Robotics Research,vol. 5, no. 4, 1986
[5] R. C. Smith, M. Self, P. Cheeseman, "Estimating Uncertain SpatialRelationships in Robotics,"Autonomous robot vehicles, pp. 167-193,1990.
[6] H. Durrant-Whyte, S. Majumder, S. Thrun, M. Battista, S. Scheding, "ABayesian Algorithm for Simultaneous Localisation and Map Building,"Robotics Research, STAR 6, pp. 49-60, 2003.
[7] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, M.Csorba, "A Solution to the Simultaneous Localisation and Map Building(SLAM) Problem,"IEEE Transactions on Robotics and Automation, col.17, iss. 3, 2001.
[8] J. J. Leonard, H.F. Durrant-Whyte, "Simultaneous Map Building andLocalization for an Autonomous Mobile Robot,"IEEE/RSJ InternationalWorkshop on Intelligent Robots and Systems IROS '91, 1991.
[9] J. J. Leonard, H. J. S. Feder, "A Computationally Efficient Method forLarge-Scale Concurrent Mapping and Localization,"Hollerbach J.M.,Koditschek D.E. (eds) Robotics Research, Springer, London, 2000.
[10] A. Chandak, K. Gosavi, S. Giri, S. Agrawal, P. Kulkarni, "Path Planningfor Mobile Robot Navigation using Image Processing,"InternationalJournal of Scientific Engineering Research, vol. 4, iss. 6, 2013.
[11] A. R. Geus, M. H. Stoppa, S. F. Silva, "PathFinder: An autonomous mo-bile robot guided by Computer Vision,"Int'l Conf. Artificial IntelligenceICAI'15, 2015.
[12] S. G. Anavatti, S. LX. Francis, M. Garratt, "Path-Planning Modules forAutonomous Vehicles: Current Status and Challenges,"Int'l Conf. onAdvanced Mechantronics, Intelligent Manufactur, and Industrial Appli-cation, 2015.
[13] J. Ng, T. Brunl, "Performance Comparison of Bug Navigation Algo-rithms,"Journal of Intelligent and Robotic Systems, vol. 50, iss. 1, pp.73-84, 2007.
[14] J. Wit, C. D. Crane III, D. Armstrong, "Autonomous Ground VehiclePath tracking,"Journal of Robotic Systems, vol. 21, no. 8, pp. 439-449,2004
[15] M. Likhachev, G. Gordon, S. Thrun, "ARA*: Anytime A* with ProvableBounds on Sub-Optimality,"Advances in Neural Information ProcessingSystems 16: Proceedings of the 2003 Conference (NIPS-03, 2003.
[16] D. Ferguson, A. Stentz, "Anytime RRTs,"Proceedings of the 2006IEEE/RSJ International Conference on Intelligent Robots and Systems,pp. 5369-5375, 2006.
[17] E. Mueggler, M. Faessler, F. Fontana, D. Scaramuzza, "Aerial-guidedNavigation of a Ground Robot among Movable Obstacles,"IEEE Inter-national Symposium on Safety, Security, and Rescue Robotics, 2014.
[18] K. Trovato, "Differential A*: An Adaptive Search Method Illustratedwith Robot Path Planning for Moving Obstacles Goals, and an UncertainEnvironment,"IEEE International Workshop on Tools for ArtificialIntelligence, 1989.
[19] S. Koenig, M. Likhachev, "Incremental A*,"Proceedings of the NeuralIfnromation Processing Systems, 2002.
[20] S. Koenig, M. Likhachev, "D* Lite,"The AAAI Conference on ArtificialIntelligence, 2002.
[21] A. Stentz, "The Focussed D* Algorithm for Real-Time Replanning,"InProceedings of the International Joint Conference on Artificial Intelli-gence, 1995.
[22] D. Ferguson, A. Stentz, "The Delayed D* Algorithm for Efficient PathReplanning,"In Proceedings of the 2005 IEEE International Conferenceon Robotics and Automation, 2005.
[23] S. Koenig, M. Likhachev, "Improved Fast Replanning for Robot Naviga-tion in Unknown Terrain,"In Proceedings of the 2002 IEEE InternationalConference on Robotics and Automation, 2002.
[24] S. Koenig, M. Likhachev, "A New Principle for Incremental HeuristicSearch: Theoretical Results,"AAAI American Association for ArtificialIntelligence, 2006.
[25] J. Carsten, D. Ferguson, A. Stentz, "3D Field D*: Improved PathPlanning and Replanning in Three Dimensions,"In Proceedings ofthe 2006 IEEE/RSJ International Conference on Intelligent Robots andSystems, 2006.
[26] S. M. LaValle, "Rapidly-Exploring Random Trees: A New Tool for PathPlanning," , 1998.
[27] S. M. LaValle, J. J. Kuffner, Jr., "Randomized Kinodynamic Planning,"In Proceedings 1999 IEEE International Conference on Robotics andAutomation, 1999.
[28] J. Barraquand J. Latombe, "Robot Motion Planning: A Distributed Rep-resentation Approach,"The International Journal of Robtoics Research,vol. 10, no. 6, 1991.
[29] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mo-bile Robots,"The International Journal of Robotics Research, vol. 5, no.1, 1986.
[30] SuperDroid Robots, "IG42-SB4, 4WD All Terrain Robot Plat-form," , [Online] http://www.superdroidrobots.com/shop/item.aspx/ig42-sb4-4wd-all-terrain-robot-platform/1474/ Accessed 4 April 2017.
[31] ParrotInc,"ParrotAR.Drone2.0,",[Online]https://www.parrot.com/us/drones/parrot-ardrone-20-elite-editionparrot-ardrone-20-elite-edition-details Accessed 4 April 2017.
[32] M. Saska, T. Krajnik, L. Preucil, "Cooperative UAV-UGV autonomousindoor surveillance,"International Multi-Conference on Systems, Signals,and Devices, 2012.
[33] L. Cantelli, M. Lo Presti, M. Mangiameli, C. D. Melita, G. Muscato,"Autonomous Cooperation Between UAV and UGV to Improve Nav-igation and Environmental Monitoring in Rough Environments,"10thInternational symposium Humanitarian Demining, 2013.
[34] K. Hausman, J. Mller, A. Hariharan, N. Ayanian, G. Sukhatme, "Coop-erative Multi-Robot Control for Target Tracking with Onboard Sensing*,"The International Journal of Robotics Research (00):1-13, 2010.
[35] M. Hsieh, A. Cowley, J. Keller, L. Chaimowicz, B. Grocholsky, V.Kumar, C. Taylor, "Adaptive Teams of Autonomous Aerial and GroundRobots for Situational Awareness,"Journal of Field Robotics, 2007.
[36] M. Garzon, J. Valente, D. Zapata, An. Barrientos, "An Aerial-GroundRobotic System for Navigation and Obstacle Mapping Large OutdoorAreas,"Sensors, 13, 1247-1267, 2013.
[37] E. Z. MacArthur, D. MacArthur, C. Crane, "Use of CooperativeUnmanned Air and Ground Vehicles for Detection and Disposal ofSimulated Mines,"Proc. SPIE5999, Intelligent Systems in Design andManufacturing VI, 599909, 2005.
[38] P. Isarabhakdee, Y. Gao, "Cooperative Control of a Multi-Tier Multi-Agent Robotic System for Planetary Exploration,"Proc. 'IJCAI-09Workshop on Artifical Intelligence in Space, 2009.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

[39] G. M. Kruijff, V. Tretyakov, T. Linder, F. Pirri, M. Gianni, P. Papadakis,M. Pizzoli, A. Sinha, E. Pianese, S. Corrao, F. Priori, S. Febrini, S.Angeletti, ”Rescue Robots at Earthquake-Hit Mirandola, Italy: A Field Report,”In Proceedings of the 10th International Symposium on Safety Security and Rescue Robotics, 2012.

[40] D. P. Stormont, V. H. Allan, ”Managing Risk in Disaster Scenarios with Autonomous Robots,” [Online]http://digital.cs.usu.edu/allanv/Pubs/Stormont-Allan-RMCI2008.pdfAccessed April 4 2017.

[41] J. Q. Cui, S. K. Phang, K. Z. Y. Ang, F. Wang, X. Dong, Y. Ke, S.Lai, K. Li, X. Li, F. Lin, J. Lin, P. Liu, T. Pang, B. Wang, K. Wang,Z. Yang, B. Chen, ”Drones for Cooperative Search and Rescue in Post-Disaster Situation,”Cybernetics and Intelligent Systems (CIS) and IEEEConference on Robotics, Automation and Mechatronics (RAM), 2015.