Obstacle Avoidance for Autonomous UAVs in Dynamic Indoor Environments

by

Jazib Ahmad

University of Toronto & SOTI

Supervised by

Igor Gilitschenski, PhD

University of Toronto

AND

Sunil Jacob, PhD

SOTI

A report submitted in conformity with the requirements
for the degree of Master's of Science in Applied Computing
Graduate Department of Computer Science
University of Toronto

Abstract

Obstacle Avoidance for Autonomous UAVs in Dynamic Indoor Environments

Jazib Ahmad
Master's of Science in Applied Computing
Graduate Department of Computer Science
University of Toronto
2023

Structured industries have been revolutionized by robotics, but now the focus is on the more complex, human-populated unstructured environments. Safety, once a byproduct of isolation, now hinges on advanced autonomy in robots. SOTI's development of an autonomous indoor drone is an example of such a system that is expected to operate safely alongside human agents in cluttered indoor environments. However, the existing obstacle avoidance systems have not kept up with the increasing safety expectations of an indoor, cluttered, and dynamic environment with humans. Therefore, this project researched obstacle avoidance methods for autonomous UAVs in dynamic indoor environments, with a focus on unmapped, cluttered indoor environments and dynamic obstacles. Our work included an adaptation of 3DVFH* to indoor environments, improvements to an event camera based method for detecting dynamic objects in 3D, and utilizing these 3D object detections in an Artificial Potential Field specialized for dynamic objects. We tested our methods in Gazebo simulation environments as well as our real drone. Our experiments show that our methods are successfully able to navigate unmapped, cluttered indoor environments and avoid dynamic objects in real-time.

Acknowledgements

I would like to start by thanking all of my colleagues at SOTI, especially everyone at SOTI Aerospace. They not only gave me this phenomenal applied research opportunity, but have also supported me tremendously in my research. This work would not have been possible without them. Most notably, Danial Khan, Dr. Ahmed Elamin, and Soroush Farghadani spent many hours helping me execute on my research. Second, I would like to thank my industry supervisor Dr. Sunil Jacob and my academic supervisor Dr. Igor Gilitschenski for their mentorship, unwavering support, and ideas that made this research possible. Third, I would like to thank my family, friends, and professors who generously supported me throughout my academic journey. A good friend of mine, Xuduo Gu, also deserves my gratitude for personal support and friendship through this project. Finally, I would like to thank the Government of Canada for the financial support through the Mitacs Accelerate Fellowship.

Contents

Chapter 1

Introduction

SOTI Inc. is a Canadian company that specializes in mobile and IoT device management software. The SOTI ONE Platform, the company's core product, is an integrated suite of solutions designed to enhance business mobility. This platform has expanded its capabilities to manage a variety of devices, including Android, iOS, Windows, Linux, and IoT devices. The company's recent ventures have extended into areas like mobile-first help desk services and rapid app development.

SOTI serves a diverse range of enterprise customers, including many in the logistics industry. Many of these customers have warehouses with manual inventory management solutions. Therefore, in 2020, SOTI expanded its scope of operations by establishing SOTI Aerospace. This division focuses on research and development in aerial drone technology and robotics, marking a new direction for the company. The goal of this research is to take enterprise mobility to the next level by bringing autonomy and the SOTI ONE platform to indoor drones.

Indoor environments present unique challenges for Unmanned Aerial Vehicles (UAVs), especially in applications like industrial inspections and inventory management. Because these environments are cluttered, dynamic, and GPS-denied, obstacle avoidance is one such unique challenge. Therefore, the aim of this project was to research and implement obstacle avoidance methods to allow autonomous UAVs to operate safely in dynamic indoor environments.

Chapter 2

Research Goals and Outcome

2.1 Research Goals

The goal of this project was to research and implement obstacle avoidance methods to allow autonomous Unmanned Aerial Vehicles (UAVs) to operate safely in dynamic indoor environments. The objective required that the methods be compatible with the autonomy stack at SOTI Aerospace, including the following sensors:

1. A stereo camera - used to perform Visual Inertial Odometry (VIO), which helps estimate the local position of the UAV together with an Inertial Measurement Unit (IMU).
2. A forward facing Time of Flight (TOF) camera - uses light to measure distances and generate a point cloud similar to LiDAR.
3. An event camera - a novel type of sensor that can measure changes in pixel intensities (events) at a very high frame rate.

In addition to these sensors, the onboard compute hardware included a Pixhawk flight controller, an NVIDIA Jetson Xavier NX computer with 8GB of memory, and PX4 to provide 3D setpoints to the drone. Finally, obstacle avoidance methods were needed for two kinds of scenarios:

1. Exploration - In this scenario, a pre-existing map is not available. Exploration algorithms aimed at building a map provide sparse goals rather than a dense path. The obstacle avoidance method is tasked with creating and following a locally optimal plan towards these goals without collisions.
2. Path Following - In this scenario, a pre-existing map is available. A path planning algorithm provides dense coordinates from an optimized path. The obstacle avoidance method is tasked with avoiding dynamic objects while following this path as well as avoiding static objects that are not in the map.

2.2 Related Work

The existing literature in Indoor UAV Obstacle Avoidance is summarized in [1]. This survey paper categorizes the existing work into the sensor modalities for obstacle avoidance, the perception algorithms, the path planning methods, and the control algorithms. Our work differed in several ways. First, we used different sensor modalities. The presented literature does not include a TOF camera or an event camera. One of requirements of our obstacle avoidance system was that it must be robust to different lighting conditions. Therefore, a stereo camera alone was insufficient. Further, the context of many of the existing obstacle avoidance methods is different. For example, many methods do not provide a locally optimal path based on sparse navigation goals and multiple objects in the scene [2], which was a requirement for our Exploration scenario. Many other methods such as the ones based on Reinforcement Learning have thus far only been successful in simulation [3] [4]. While these are notable achievements for the academic context, they are not able to deal with real-world uncertainty. This includes uncertainty in obstacle positions, sensor measurements, UAV position estimation, and more. Some other existing literature currently only addresses obstacle detection but not the full avoidance pipeline [5]. Finally, most methods do not address fast-moving dynamic objects [6].

2.3 Outcomes

Our work took advantage of some of the existing obstacle avoidance literature, with various improvements and adaptations for dynamic indoor environments. This included the work of [7], which proposed the 3D Vector Field Histogram* (3DVFH*) algorithm. The 3DVFH* algorithm has the advantage of being able to navigate in unmapped areas with sparse navigation goals points and multiple obstacles in a locally optimal way, which satisfied our exploration requirement. However, we made some modifications to the algorithm to adapt it to indoor, cluttered environments. The algorithm and our modifications are described in Chapter 3. We also took advantage of the Artificial Potential Field (APF) method described in [8] . This APF is specialized for better performance with fast moving dynamic objects by, for example, including a decay for dynamic occluded objects and including the velocity in the force direction calculations. It is also described in greater detail in Chapter 3. To detect dynamic objects in 3D, we utilized the algorithm proposed in [9], FAST Dynamic Vision, which is based on an event camera and a depth camera. However, we made enhancements to this algorithm to improve the results. This method and our enhancements are also described in Chapter 3.

Chapter 3

Methods

Our research included an obstacle avoidance method for the Exploration scenario (Section 3.1), and an obstacle avoidance & dodging method for fast moving dynamic objects using an event and a depth camera (Section 3.2). This chapter describes our methodologies, the choices we made, and the reasoning for those choices.

3.1 Obstacle Avoidance for Exploration

One of the required obstacle avoidance scenarios was the Exploration scenario described in Chapter 2. For this scenario, we chose to adopt the 3DVFH* algorithm proposed in [7] to indoor environments.

3.1.1 3DVFH*

The 3DVFH* algorithm initiates by processing the raw point cloud data acquired from our Time of Flight (TOF) camera to remove noise and build a 2D histogram. The vertical dimension of the histogram represents elevation angle and horizontal dimension represents the azimuth angle. The value in each bin in the histogram represents the average distance of the points in that direction from the current position of the UAV. In this first step, the algorithm also uses memory to store an old point cloud, which retains the recently seen points that are no longer in the Field of View (FOV). This gives it short-term memory and the ability to create locally optimal plans. The final histogram includes distance data from both the current point cloud and the one in memory.

In the next step, the average distance values are used to create a cost matrix or image with the same dimensions as the histogram. Each cell represents the cost to go in that specific direction. The cost matrix does not only use the average distance values from the histogram, but also other heuristics such as the distance from the goal, the change in the orientation required to go in that direction, the difference from the current path, etc. These help guide the UAV to its goal while navigating more smoothly in its FOV.

The last step is to utilize this cost matrix to build a local plan and select the next 3D waypoint from the current plan. This is done by selecting the best few cells in the cost matrix, setting them to the new origin, reevaluating the cost matrix, and repeating this process for some number of timesteps into the future. From the resulting tree, the path with the lowest cost is selected and the next waypoint is calculated based on path smoothness and speed requirements. Further details of the algorithm are available in [7].

3DVFH* was selected primarily due to its ability to meet the requirements of our Exploration scenario. It does not require a map and can still create locally optimal plans from sparse goals to explore unknown environments. Its use of short-term memory enables this while still operating in real-time and leaving system resources for other tasks. We utilized the implementation of 3DVFH* provided by [10] to reduce the development time. However, our experiments showed that the original implementation was not suitable for indoor navigation, especially in cluttered spaces. Section 3.1.2 describes our adaptation to make it suitable for indoor environments.

3.1.2 3DVFH* for Indoor Environments

With the original implementation of 3DVFH*, we noticed that the point cloud from the floor and ceiling were closer than the obstacle avoidance distance in indoor navigation spaces. This rendered the UAV unable to navigate. Therefore, we introduced a cutoff for the point cloud to artificially remove points above and below the UAV by a certain margin. The resulting implementation also required modifying how the point cloud is stored in memory, since the artificial FOV is different from the real FOV. In addition, it required increasing the cost of changing elevation, since those areas are no longer visible.

Similarly, the density of the 3D space used in [10] was insufficient for cluttered indoor environments. Therefore, we modified it to accommodate denser point clouds. This implementation also provided other minor features that made the algorithm unsafe for indoor navigation. For example, it discarded tree nodes if they were closer than 0.2 meters, which is often the case in indoor environments. In a similar way, its speed calculations and other minor parameters were also adapted to be suitable for cluttered indoor environments.

3.2 Dynamic Obstacle Avoidance

In alignment with our research goals described in Chapter 2, our research also included the avoidance of fast moving dynamic objects. For this purpose, we improved and adapted the approach described in [9], FAST Dynamic Vision, to detect dynamic objects, and combined it with the approach described in [8] to avoid them using an Artificial Potential Field (APF).

3.2.1 FAST Dynamic Vision

FAST Dynamic Vision [9] is an approach to detect dynamic objects in 3D by utilizing an event camera and a camera that can provide depth images, for which we used our TOF camera. The event camera's time image is first filtered to remove events generated by ego-motion (motion of the UAV). This is done with the help of the IMU and depth images. Next, the dynamic object is segmented in the time image by subtracting the mean timestamp of the events and some configurable threshold. This is based on the principle that events from the static objects are generated uniformly in time. Iterative Gaussian Fitting is then used to create a bounding box around the object. The bounding box is then projected onto the depth image. From here, the nearest significant depth is used to create a mask and locate the object in 3D. We refer readers to [9] for the detailed discussion on this interesting method.

We chose this method because it is not only the current state-of-the-art in detecting dynamic objects using event cameras, but also presents a practical method for using depth images to accurately estimate the 3D position of those dynamic objects. Furthermore, it has real-time performance and low-latency, making it possible to react to fast moving objects in time to avoid a collision. However, our experiments showed several shortcomings that resulted in missed detections as well as many false positives. These shortcomings and our solution to address them are as follows.

3.2.2 Better FAST Dynamic Vision

First, we observed that using Iterative Gaussian Fitting to detect objects was resulting in bounding boxes that were too small. As a result, the dynamic objects were often being filtered out as noise, or missed in the depth image. Therefore, we used a single iteration, or simply Gaussian Fitting which performed significantly better. This also allowed us to filter out small detections as noise not only based on the bounding box area, but also based on the bounding box width and height. Larger bounding boxes were also more likely to include the dynamic object when projected onto the depth image. The result was fewer missed detections, fewer false positives, and more accurate estimation of the 3D position of the dynamic object.

Another challenge was that event and depth images were not aligned in time. Since the depth image is not directly sensed, it incurs a small delay in generation. This delay meant that the bounding box from the event image often did not correctly project onto the object in the depth image, especially for fast moving dynamic objects. This can result in missed detections or incorrectly estimated positions. To solve this problem, we experimentally measured the number of depth image frames that the depth image was behind the event image, and added this delay in the bounding box projection.

Furthermore, the method did not account for regions of the depth image being empty. This occurred often, because of the difference in the event and depth sensors including image sizes as well as the limited range of the TOF camera for estimating depth. Therefore, we included a check that involved filtering out detections for which the bounding box contained primarily empty depth values. Finally, we added significant optimizations to the method such as removing trajectory estimation, which was based on optimistic assumptions, that resulted in a reduction in latency by 70% to 90%.

3.2.3 APF for Dynamic Obstacles

Artificial Potential Field (APF) is an obstacle avoidance approach that uses artificial repulsive forces from obstacles and artificial attractive forces towards goals to guide the motion of the UAV. The APF proposed in [8] is specialized for avoiding fast approaching obstacles. Its key features include a repulsive direction that is based on both the obstacle position and velocity using a cross product of the obstacle to UAV position vector and the obstacle velocity vector. This ensures that the avoidance maneuver does not intersect with the obstacle path. Furthermore, this APF uses a force formulation that results in stronger repulsion at a greater obstacle distance without a large (destabilizing) repulsive constant. The magnitude of the obstacle velocity is also a factor in the magnitude of the repulsive force. Finally, the repulsion decays over time, which allows missed observations and occluded obstacles to be accommodated. We chose this APF for these desirable features, as well as experimental results which proved its effectiveness. We refer the reader to [8] for a more detailed explanation of the approach.

Chapter 4

Results and Discussions

Our experiments included testing 3DVFH* for Indoor Environments in simulation and real-world for the Exploration scenario described in 2.1, testing Better FAST Dynamic Vision in simulation and real-world, testing the APF for Dynamic Obstacles in simulation and real-world, and testing the full pipeline of Dynamic Obstacle Avoidance in simulation and real-world including the vision and APF. Simulation experiments were conducted using the Gazebo simulator. Real-world experiments were conducted using the X500 drone kit equipped with an Intel Realsense T265 for VIO, an Orbbec Femto TOF camera, and an Inivation DAVIS 346 Event Camera. Our results are reported in the following sections.

4.1 3DVFH* for Indoor Environments

We conducted experiments to ensure that the 3DVFH* for Indoor Environments algorithm is able to meet the requirements of our Exploration scenario, described in Chapter 2. The algorithm was provided sparse goals in an unmapped environment containing multiple static obstacles, such as walls, trees, mannequins, and other objects. In our experiments, it was able to autonomously navigate to the goals while avoiding obstacles. See Figure 4.1 which shows a visualization of the result for a real-world indoor environment with a mannequin and walls. The processed point cloud, 2D histogram, cost image, current goal (yellow sphere), and planned waypoints are shown. This result shows that the obstacles were correctly observed, processed in the 2D histogram and cost image, and that a good planned path was chosen. The resulting planned path was also around the left side of the mannequin, which was a good choice because the right side contained the observed wall. The plan was followed and ultimately the UAV reached the goal without collisions.

4.2 Better FAST Dynamic Vision

We conducted experiments to test the performance of our 3D detection of dynamic objects. Our experimental approach involved testing the algorithm in environments with dynamic objects including basketballs, drones, mops, and humans. We tested the algorithm with and without ego-motion. Since there was no dataset with ground truth detections, a quantitative analysis of the performance was not possible. However, visual inspections of the results showed that the approach is successfully able to detect dynamic objects and estimate their 3D position. We observed that false positives were rare without

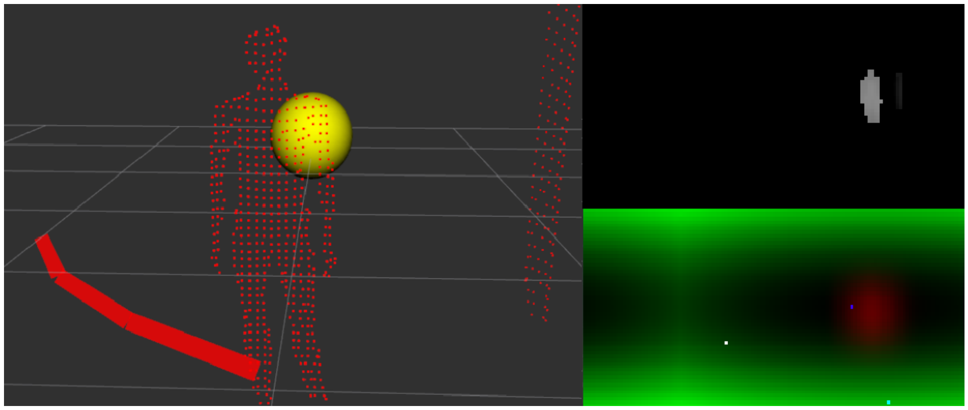


Figure 4.1: 3DVFH* Result. The goal is the yellow sphere. The processed point cloud is show in red. The right shows the 2D histogram and cost image. The selected path is displayed with the point cloud on the left.

ego-motion. However, with ego-motion, false positives were frequent and we believe this is because the ego-motion compensation is not able to completely remove the events caused by ego-motion. We discuss this challenge as an area of future work in Chapter 5.

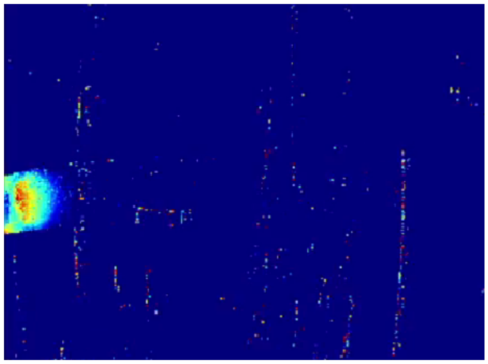
Figure 4.2 shows a correctly detected moving object and Figure 4.3 its estimated 3D position. This pair of figures shows that the method correctly estimated a bounding box for the object in the event time image, the bounding box was then correctly projected onto the depth image, and it's 3D position was correctly estimated as the object passed from in front of the UAV. Figure 4.4 shows a false positive detection, caused by the events generated from ego-motion.

4.3 APF for Dynamic Obstacles

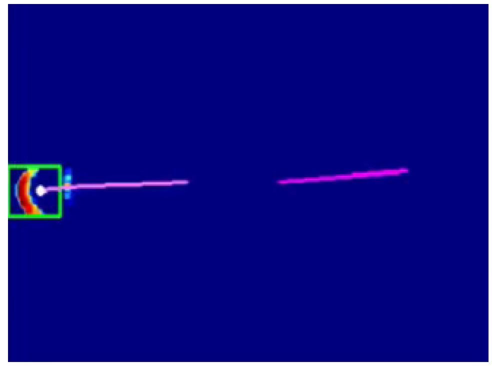
To test the avoidance capability of the APF for Dynamic Obstacles, we used obstacles at several (starting) positions with various velocities and measured the Minimum Distance (MD) between the obstacle and the UAV with and without the APF. The results are shown in Table 4.1. The APF managed to avoid the obstacle in all scenarios. Diagonal motion relative to the UAV is a potential weakness, with the smallest avoidance distance. To increase the MD values, the parameters of the APF can be adjusted.

4.4 Dynamic Obstacle Avoidance

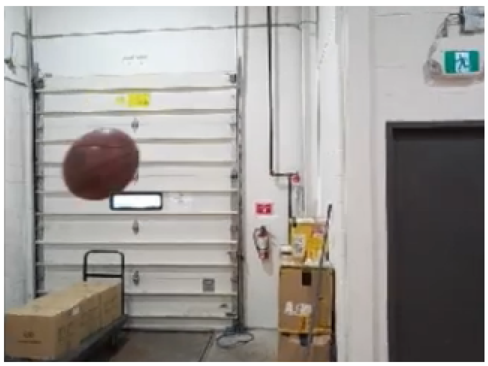
To test the combined Dynamic Obstacle Avoidance, we used the same experimental approach as 4.2. However, with the combined approach, the UAV was able to both detect and move away from the dynamic obstacles. In our experiments, we did not observe any collisions, which indicated that our method was successful.



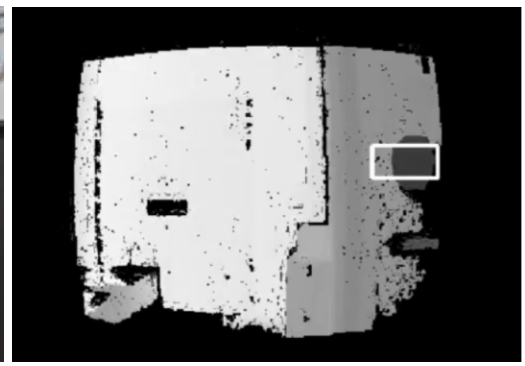
(a) Time Image



(b) Processed Time Image



(c) Color Image



(d) Depth Image

Figure 4.2: Better FAST Dynamic Vision - Dynamic Object Detection. 4.2a shows the time image of events after ego motion compensation. 4.2b shows the time image after processing including subtracting the mean timestamp and the threshold, as well as the bounding box around the detected object. 4.2c shows the dynamic object in the RGB image, and 4.2d shows the depth image. The depth image is behind by a few frames, but the figure shows that a correct bounding box was projected onto it since it includes the dynamic object.

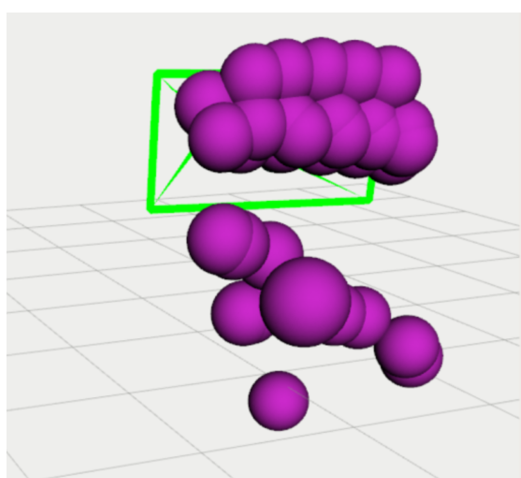
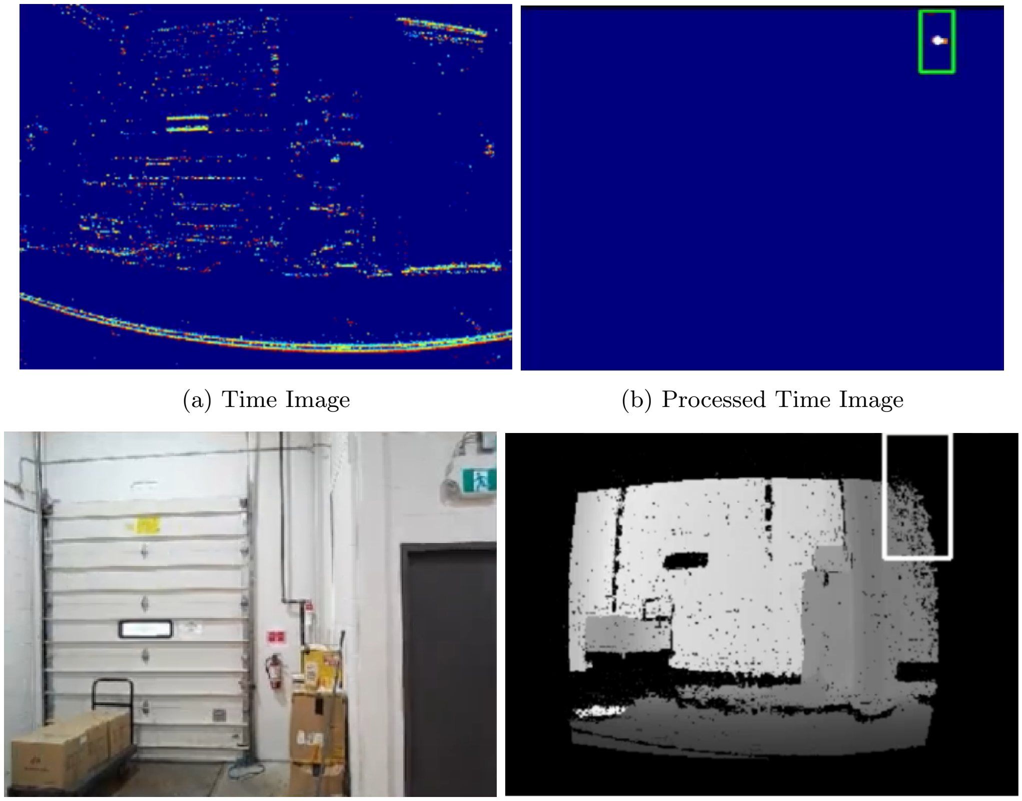


Figure 4.3: Better FAST Dynamic Vision - 3D Position Estimation. The purple spheres show the estimated 3D position of detected moving object, which was a basketball through in front of the UAV several times. The figure shows a set of detections reflecting the trajectory for each throw.



(a) Time Image

(b) Processed Time Image

(c) Color Image

(d) Depth Image

Figure 4.4: Better FAST Dynamic Vision - False Positive. 4.4a shows the time image, which contains events caused by the motion of the UAV. 4.4b shows the time image after processing, which contains a bounding box. 4.4c shows the corresponding RGB image, which has no dynamic object, confirming that this detection was a false positive. 4.4d shows the bounding box projected onto the depth image.

Position	Velocity	MD w/ APF	MD w/o APF
(5.0, 0.0, 1.0)	(-1.0, 0.0, 0.0)	1.00	0.00 (Collision)
(0.0, 5.0, 1.0)	(0.0, -1.0, 0.0)	1.04	0.00 (Collision)
(0.0, 5.0, 1.0)	(0.0, -2.0, 0.0)	0.93	0.00 (Collision)
(0.0, 5.0, 1.0)	(0.0, -3.0, 0.0)	0.68	0.00 (Collision)
(-5.0, -5.0, 1.0)	(1.0, 1.0, 0.0)	0.24	0.00 (Collision)
(5.0, -5.0, 1.0)	(-1.0, 1.0, 0.0)	0.25	0.00 (Collision)
(1.0, 5.0, 1.0)	(0.0, -3.0, 0.0)	1.34	1.00
(-5.0, 0.5, 1.0)	(1.0, 0.0, 0.0)	1.38	0.50
(-5.0, -5.5, 1.0)	(1.0, 1.0, 0.0)	0.72	0.50

Table 4.1: APF Results. All values are reported in meters and are relative to the UAV. With all starting positions and velocities of obstacles tested, the APF successfully avoided collisions, and increased the distance between itself and the dynamic obstacle. Dynamic obstacles moving diagonally towards the UAV with a path intersecting with it seemed to result in the smallest Minimum Distance (MD).

Chapter 5

Conclusions and Future Research Plans

In this project, we researched and implemented obstacle avoidance methods to allow autonomous Unmanned Aerial Vehicles (UAVs) to operate safely in dynamic indoor environments. Two obstacle avoidance methods were implemented, both of which were shown to be successful. In this chapter, we discuss our impact (Section 5.1) and possible future work based on our findings (Section 5.2).

5.1 Impact

The first part of the project utilized the 3DVFH* algorithm and adapted it for indoor environments based on experimental observations. The resulting method was shown to be safe and can be utilized by SOTI to allow its future autonomous UAVs to safely explore unmapped indoor environments. The second part of the project made improvements on an event camera based vision method to detect dynamic objects in 3D and an Artificial Potential Field (APF) to avoid them. The resulting method can be utilized by SOTI to allow its future autonomous UAVs to avoid dynamic obstacles and therefore navigate more safely. Both the vision method and the APF can also be used separately as part of other dynamic obstacle avoidance methods.

5.2 Future Work

Several areas of future work are possible based on our findings. In our testing of 3DVFH* for Indoor Environments, we observed that the algorithm is not able to successfully navigate in very tight spaces such as corridors. While it does not result in collisions, it also does not make progress towards goals in such environments. That is because it is designed to keep a minimum distance from all objects, including walls. Therefore, a space narrower than that minimum distance is not suitable for the current version of the algorithm. One possible direction of future work to address this issue is to use segmentation information to keep a variable distance from objects. For example, walls can have a smaller minimum distance while humans can have a larger minimum distance, allowing the UAV to navigate through narrower corridors while keeping a safe distance around humans.

As reported in Chapter 4, the FAST Dynamic Vision approach would benefit tremendously from a better method for ego motion compensation. With fewer false positives, the method's sensitivity to detection could be increased, resulting in more true positives as well. One possible approach to do this is to utilize vision information to perform segmentation of known-static objects such as walls and floors. The events generated from areas of the image that are static can then be filtered. We plan to investigate this approach soon.

Finally, one important question that arises from this work is how to integrate it. This includes integrating both of the methods researched, in order to avoid both static and dynamic objects in unmapped environments with one obstacle avoidance system. It also includes integrating the obstacle avoidance system with the Path Following scenario described in Chapter 2. Our future research plan includes investigating a method for integration based on generating 2D cost matrices similar to the one used by 3DVFH*, and aggregating them with configurable weights.

Bibliography

[1] J. Li, X. Xiong, Y. Yan, and Y. Yang, “A survey of indoor uav obstacle avoidance research,” *IEEE Access*, vol. 11, pp. 51 861–51 891, 2023.

[2] H. Müller, V. Niculescu, T. Polonelli, M. Magno, and L. Benini, “Robust and efficient depth-based obstacle avoidance for autonomous miniaturized uavs,” 2022.

[3] Z. Yijing, Z. Zheng, Z. Xiaoyi, and L. Yang, “Q learning algorithm based uav path learning and obstacle avoidence approach,” in *2017 36th Chinese Control Conference (CCC)*, 2017, pp. 3397–3402.

[4] X. Han, J. Wang, J. Xue, and Q. Zhang, “Intelligent decision-making for 3-dimensional dynamic obstacle avoidance of uav based on deep reinforcement learning,” in *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2019, pp. 1–6.

[5] H.-Y. Lin and X.-Z. Peng, “Autonomous quadrotor navigation with vision based obstacle avoidance and path planning,” *IEEE Access*, vol. 9, pp. 102 450–102 459, 2021.

[6] S. Kim, J. Park, J.-K. Yun, and J. Seo, “Motion planning by reinforcement learning for an unmanned aerial vehicle in virtual open space with static obstacles,” in *2020 20th International Conference on Control, Automation and Systems (ICCAS)*. IEEE, Oct. 2020. [Online]. Available: http://dx.doi.org/10.23919/ICCAS50221.2020.9268253

[7] T. Baumann, “Obstacle avoidance for drones using a 3dvfh* algorithm,” Master’s thesis, Swiss Federal Institute of Technology Zurich, Zurich, Switzerland, 2018, master’s thesis.

[8] D. Falanga, K. Kleber, and D. Scaramuzza, “Dynamic obstacle avoidance for quadrotors with event cameras,” *Science Robotics*, vol. 5, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:213193728

[9] B. He, H. Li, S. Wu, D. Wang, Z. Zhang, Q. Dong, C. Xu, and F. Gao, “Fast-dynamic-vision: Detection and tracking dynamic objects with event and depth sensing,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3071–3078.

[10] PX4DevelopmentTeam, “Px4-avoidance,” 2019, gitHub repository. [Online]. Available: https://github.com/PX4/PX4-Avoidance