AN UNMANNED AIRCRAFT "SEE AND AVOID" ALGORITHM DEVELOPMENT PLATFORM USING OPENGL AND OPENCV[*]

Andrew Morgan, Zachary Jones, Richard Chapman, and Saad Biaz
Department of Computer Science and Software Engineering
Auburn University
Auburn, AL 36849
chapmro@auburn.edu

ABSTRACT

This work presents a see-and-avoid (SAA) system for autonomous Unmanned Aircraft Systems (UAS) and a testbed for evaluating SAA capabilities using simulated cockpit video generated using the computer graphics library OpenGL. The computer vision library OpenCV is used to construct algorithms for background filtering, obstacle detection, and collision avoidance. A distance-based and a distance-agnostic algorithm were developed and evaluated on the proposed testbed. Preliminary results with scenarios with two to four planes are promising.

INTRODUCTION

A major obstacle to the integration of UAS into the national airspace (NAS) has been their inability to comply with FAA regulations for human pilots. According to 14 CFR part 91.113, aircraft should be able to "see and avoid" other aircraft to avoid collisions. While current rulemaking attempts to accommodate the lack of this capability for UAS, it would enhance safety if UAS could implement 91.113.

Background

Previous work by the authors created a fixed-node control unit for GPS-enabled UAS on a ground-based air traffic control computer running Robot Operating System (ROS). Several collision avoidance algorithms including inverse proportional navigation and artificial potential fields were implemented and evaluated [4,6,7]. Dayton et.al. [2]

---

developed a collision detection system for fixed-wing UAS capable of handling both moving and non-moving objects. The OpenCV library was used to process a video stream from a front-facing, externally mounted camera. Shi-Tomasi corner detection was used to identify interesting features to track, and the Lucas-Kanade optical flow method was used to track moving objects and predict collisions. Ortiz and Neogi [10] proposed a similar method, noting that many other object tracking algorithms are too computationally expensive. Carnie et al.[1] use a morphological operation known as a "Close-Minus-Open" (CMO) in order to extract point-like masses from large-scale clutter. Lyu et. al. [9] addressed this problem by first performing a sky-ground segmentation, then following up with a morphological CMO operation to detect points of interest. . Reed et. al. [11] use a technique called "3-D matched filtering" to extract target points from background noise. Deshpande et al [3] use max-mean and max-median filters on infrared sensor images to detect small targets and distinguish them from clutter. Approaches involving stereo vision systems are limited by the computational capacity of the UAS. Attempts to mitigate the high computational load include using techniques such as dynamic programming[5] or transforming the image matching problem into an optimization problem and using a genetic algorithm[12]. Hrahar et al. [8] combine color optic flow and stereo vision techniques Yu [13] implements vision based planning, avoidance, and target tracking on UAS using a depth map of an environment obtained by computer vision methods, and an Extended Kalman Filter to estimate range and sizes of obstacles.

Project Goals

The primary objective of this research is to build a robust fixed-wing UAS SAA system using visible light computer vision techniques. Camera-based vision systems are light, scalable, and easy to deploy. While information exchange systems such as ADS-B can provide collision avoidance capabilities, they rely on all aircraft having specific pieces of hardware. SAA is capable of detecting all obstacles.

A secondary objective of the research is a testbed to simulate cockpit video from a camera fed to a computer vision-based avoidance algorithm. With a realistic simulation of cockpit video, vision-based collision-avoidance algorithms can be prototyped and evaluated more cheaply than possible with live testing. The test environment will consist of a 3-dimensional scene simulating the view of a front-facing camera mounted to an aircraft. The camera's orientation will change as the aircraft moves. The simulated airspace is given a realistic background and contains moving obstacle aircraft.

METHODOLOGY

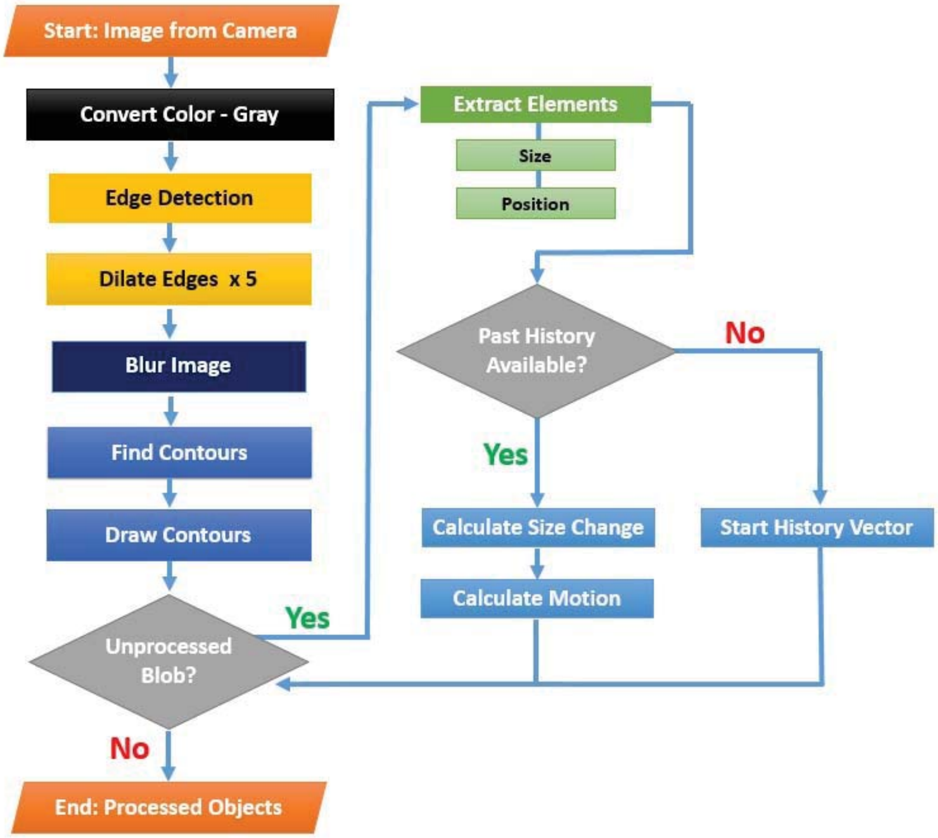
The OpenGL simulation passes images to the OpenCV-based SAA system. The SAA system is responsible for recognizing any potential obstacles and, in response, issuing avoidance commands to the UAS. The avoidance commands will be simulated in the cockpit video to provide a realistic feedback loop. The SAA system can be broken down into vision processing and collision avoidance.

Care must be taken to distinguish objects that could be obstacles from background noise such as cloud and terrain clutter. The vision processing algorithm must be efficient enough to process video from the camera in near-real-time. The SAA system first converts images to grayscale for improved edge detection, then Canny edge detection is applied, and the detected edges are dilated by 500% to removes gap in the outlines of objects. The image is slightly Gaussian blurred to remove texture information and reduce cloud clutter. Finally, OpenCV's findContours function is used to detect closed shapes and draw bounding circles around the shapes. The output at the end of these steps is an array of detected objects with the on-screen location and size of each detected object (Figure 1).

We implement two different avoidance techniques. One (distance-based) involves using image size to estimate the actual size of an identified object. If the size is known, then distance and velocity can be estimated. The other technique (distance-agnostic) makes no assumptions about distance, and an avoidance maneuver is chosen based only on information available from captured by camera.

The distance-based technique uses a reference frame and triangle similarity. This technique requires that the tracked object's actual size be known. The object's size p (in pixels) is measured from a reference frame, and the camera's focal length F is computed as $F = p/(DS)$, where D is the known distance to the object in the reference frame, and S is the actual size of the object. Then for any future image containing the object, the distance d to the object can be computed as $d = SF/p'$ where p' is the new measured size in pixels. The present system includes reference frames for a large, commercial airliner, a mid-size fighter, and a small plane. The problem of identifying the object detected by the vision processing algorithm is a separate research problem. By tracking the objects position over several frames, the object's velocity and projected flight path can be estimated. To determine if a collision between the aircraft and the object is likely, the distance between the projected path and the object's path can be estimated as a function of time, and minimized using elementary techniques from calculus. If the minimum distance is below some threshold, then the object is identified as a collision threat, and an avoidance maneuver is chosen, to safely avoid the object with minimal deviation from the intended course. To avoid the object, the deflection between the current flight path, and the projected path of the object must increase.

The design of the distance agnostic algorithm is dependent on the hardware used, including the camera FOV and resolution. The algorithm receives as input the "information from the vision processor about each blob including about its current size, position, and change in size/position over the past 30 frames. A blob must meet two criteria to be considered a threat: (1) It must have obtained a sufficiently high "danger value" by remaining near the center or moving closer to the center of the camera's FOV during the past few frames. We define a bounded weight function based on the distance from the blob's center to the center of the FOV. A running total is kept for the last 30 frames. If the danger value is above a threshold, then the blob has passed the first criteria. (2) The blob must be sufficiently large before an avoidance maneuver is attempted. This prevents the algorithm from being too aggressive and ensures an avoidance maneuver is made only when necessary. If a blob is sufficiently large, an avoidance maneuver is immediately attempted. After the threat criteria are met, the algorithm must determine an avoidance maneuver. This algorithm was designed to prefer avoidance maneuvers that involve changes in elevation, as that is a common ATC instruction for an aircraft at risk of collision - it allows the aircraft to maintain its heading. In close cases the algorithm will also add a horizontal component to the avoidance maneuver when necessary. The combination of elevation change and horizontal path modification is nearly always sufficient to avoid a collision, provided that the maneuver is completed in a timely fashion. However, a horizontal translation nearly always results in a larger deviation from the intended path than an elevation change, so when the horizontal velocity of a blob is not significant, no horizontal component will be present in the avoidance maneuver. Additionally, if further "dangerous" blobs are detected in the middle of a maneuver, it will be aborted and a new one will be planned.



RESULTS

Table 1	2-plane	tests
Bearing	Distance-agnostic	Distance-based
0 deg	Avoided	Avoided
60 deg	Avoided	Avoided
90 deg	Avoided	Avoided
160deg	Collided	Collided

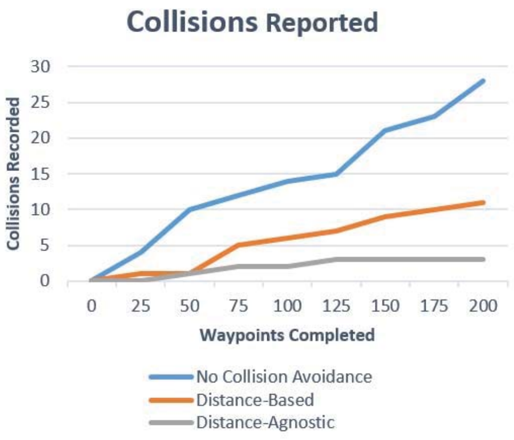
The OpenGL test bed allows for rapid construction and simulation of various scenarios. The simplest possible collision scenario involves two planes - the plane being autonomously controlled and some other "obstacle" plane. The two-plane collision scenarios are summarized in Table 1,where the Bearing indicates the angle at which the obstacle plane was approaching. In reality, most aircraft will never encounter more than one "obstacle plane" at a time unless the aircraft was near a crowded airspace such as an airport. With this consideration in mind, a scenario was devised in which many planes were maneuvering in an airspace similar to that of an airport. The autonomous plane is then instructed to fly a course that crosses through the airport space several times at a variety of angles. While unrealistic for legal reasons, this scenario provides a robust test for an autonomously guided UAS. As a control, the simulation was first run with no collision avoidance, and then with the distance-based and distance-agnostic algorithms.

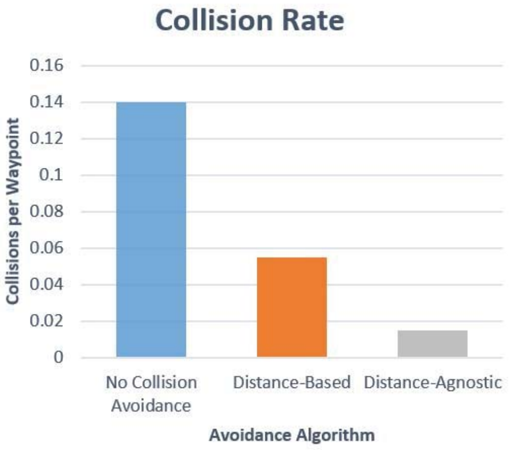
Table 2	3-plane	tests
Bearing	Agnostic	Distance
0,45	Avoided	Avoided
0,90	Avoided	Avoided
0,160	Avoided	Avoided
45,90	Avoided	Avoided
45,160	Avoided	Collided
90,160	Avoided	Collided

Table 3	4-plane	tests
Bearing	Agnostic	Distance
0,45,90	Avoided	Avoided
0,45,160	Avoided	Avoided
0,90,160	Avoided	Avoided

45,90,160	Avoided	Collided

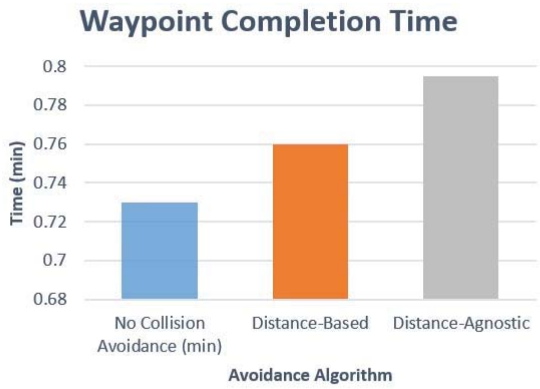
The most important measure of a collision avoidance system is its ability to avoid collisions. Success of the avoidance algorithm can be measured by the number of collisions per completed waypoint. Naturally, lower numbers indicate more successful collision avoidance. Figure 3 shows collisions detected versus waypoints completed. The bar chart shows the average number of collisions per waypoint. It is also desirable to minimize deviation from the intended course. For all simulations, the deviation from the intended course was measured by the average time to complete a waypoint. Figure 4 shows flight time versus waypoints completed. The bar chart shows the average time (in minutes) required to complete one randomly generated waypoint.





CONCLUSIONS

Using the test environment, both algorithms were able to avoid many collision events while maintaining efficient flight patterns. The distance-agnostic avoidance outperforms the distance-based avoidance algorithm in terms of collision avoidance in this simulation, while the distance-based algorithm outperforms the distance-agnostic algorithm in terms of flight efficiency, but this is because the distance-based algorithm made fewer avoidance maneuvers. In both cases, it should be noted that the average waypoint completion time is close to that of no avoidance, indicating that the algorithms yield a good efficiency. Simulation also illustrated that the CV algorithm is quite robust. Nearly all background noise is filtered, and collision threats are nearly always identified in plenty of time to execute an avoidance maneuver. However, a single camera lacks a 360-degree field of view. Distance estimation is a significant challenge. Future work will focus on improving the ability of the SAA system to identify the obstacle aircraft type, direction, and speed. More sophisticated avoidance algorithms should be considered.



REFERENCES

[1] Carnie, Walker, and Corke. Computer-vision based collision avoidance algorithm for UAVs. *Proc. of 11th Australian Int'l Aerospace Cong.*, Melbourne, Australia, 2005.

[2] Dayton, Enriquez, Gan, Liu, Quintana, and Richards. Obstacle avoidance system for UAV's using computer vision, www.cpp.edu/~cppsrc/documents/Richards.pdf, retrieved May 25, 2016.

[3] Deshpande, Venkatessarlu, and Chan. Max-mean and max-median filters for detection of small targets. *Conf. on Signal and Data Processing of Small Targets*, 1999.

[4] David Fish, Eric Westman, James Holt, and Saad Biaz. An inverse proportional navigation algorithm for collision avoidance among multiple unmanned aerial vehicles. *Emory Undergraduate Research Journal*, 12(3):7p, 2013. ?

[5] Gonzalez, Cancelas, Alvarez, Fernandez, and Enguita. Fast stereo vision algorithm for robotic applications. *7th IEEE Int'l Conf on Emerging Technologies and Factory Automation*, 1999.

[6] Holt, Aji, and Biaz. Comparison of UAS collision avoidance algo- rithms in a simulated environment. *Journal of Guidance, Control, and Dynamics*, 36(3):881-883, 2013. ?

[7] Holt, Biaz, Yilmaz, and Aj. A symbiotic simu- lation architecture for evaluating UAVs collision avoidance techniques. *Journal of Simulation*, 8(1):64-75, April 2014.

[8] Hrabar, Sukhatme, Corke, Usher, and Roberts. Combined optic-flow and stereo-based navigation of urban canyons for a UAV. *IEEE Int'l Conf. on*

Intelligent Robots & Systems, 2005.

[9] Lyu, Pan, Zhao, Zhu,Tang, and Zhang. A vision based sense and avoid systems for small unmanned helicopter. *2015 Int'l Conference on Unmanned Aircraft Systems.*.

[10] Ortiz, Neogi. Color optic flow: a computer vision approach for object detection on UAVs. *25th Digital Avionics System Conference*, October 15, 2006.

[11 Reed, Gagliardi, and Stotts. Optical moving target detection with 3-d matched filtering. *IEEE Trans on Aerospace and Electronic Sys.*, 24,(4),pp.327-336, July 1988.

[12] Woo and Dipanda. Matching lines and points in an active stereo vision system using genetic algorithms. *Proc., Int'l Conf. on Image Processing*, 3, 332-335, 2000.

[13] Yu. *Vision-based Path Planning, Collision Avoidance, and Target Tracking for Unmanned Air and Ground Vehicles in Urban Environments*. Ph.D. dissertation, Brigham Young University, Provo, Utah, 2011.