See and avoidance behaviors for autonomous navigation

Dah-Jye Lee, Randal W. Beard, Paul C. Merrell, and Pengcheng Zhan
Department of Electrical and Computer Engineering
Brigham Young University, 459 CB
Provo, Utah 84602

ABSTRACT

Recent advances in many multi-discipline technologies have allowed small, low-cost fixed wing unmanned air vehicles (UAV) or more complicated unmanned ground vehicles (UGV) to be a feasible solution in many scientific, civil and military applications. Cameras can be mounted on-board of the unmanned vehicles for the purpose of scientific data gathering, surveillance for law enforcement and homeland security, as well as to provide visual information to detect and avoid imminent collisions for autonomous navigation. However, most current computer vision algorithms are highly complex computationally and usually constitute the bottleneck of the guidance and control loop. In this paper, we present a novel computer vision algorithm for collision detection and time-to-impact calculation based on feature density distribution (FDD) analysis. It does not require accurate feature extraction, tracking, or estimation of focus of expansion (FOE). Under a few reasonable assumptions, by calculating the expansion rate of the FDD in space, time-to-impact can be accurately estimated. A sequence of monocular images is studied, and different features are used simultaneously in FDD analysis to show that our algorithm can achieve a fairly good accuracy in collision detection. In this paper we also discuss reactive path planning and trajectory generation techniques that can be accomplished without violating the velocity and heading rate constraints of the UAV.

Keywords: Unmanned Air Vehicle, Unmanned Ground Vehicle, Reactive Path Planning, Feature Tracking, Feature Density Distribution

I. INTRODUCTION

Recent advances in communications, solid state devices, and battery technology have made small, low-cost fixed wing unmanned air vehicles a feasible solution for many applications in the scientific, civil and military sectors. With the use of on-board cameras this technology can provide important information for low-altitude and high-resolution applications such scientific data gathering, surveillance for law enforcement and homeland security, precision agriculture, forest fire monitoring, geological survey, and military reconnaissance.

The fixed wing UAVs that are currently used by the military (e.g., Preditor, Global Hawk) are large, expensive, special purpose vehicles with limited autonomy. At the other end of the spectrum are small UAVs (less than six foot wingspan), and micro air vehicles (MAVs) (less than one foot wingspan), which are primarily being developed by universities and research laboratories. Current FAA regulations limit the access of unmanned air vehicles in the national air space. In particular, autonomous UAVs are currently banned from flying in the national air space. The primary concern is safety. Most UAVs are not equipped with on-board sensors that detect imminent collisions.

The **M**ultiple **AG**ent **I**ntelligent **C**oordination and **C**ontrol (MAGICC) Laboratory at Brigham Young University (BYU) has recently developed innovative autopilot technologies for small UAVS. Capabilities include autonomous take-off after a hand launch, autonomous landing, and the ability to follow a pre-determined trajectory, or to fly to specified waypoints. With an on-board video camera, these autonomous UAVs can be utilized to gather useful information for many low-altitude high-resolution applications. Unfortunately, these applications are not feasible without see and avoid capability that can detect and avoid collision with other UAVs or obstacles, such as trees and buildings. Our research objective is to combine novel computer vision algorithms with innovative trajectory planning techniques for UAVs so that the see and avoid problem can be solved in real time.

An enabling technology for small and micro UAVs is see-and-avoid technology, i.e., the ability to detect imminent collisions and to maneuver the UAV to avoid them. Sensing technologies such as radar and laser range finders are available to large and medium size UAVs. However, the heavy weight and excessive power requirements of these

sensors make them infeasible for small and micro UAVs. Since most applications of small UAVs require an on-board camera, it makes sense to use the camera to solve the see-and-avoid problem as well.

The see-and-avoid problem is a unique computer vision task that does not require accurate measurement of object distance or 3-D scene reconstruction. The objective is to *see* the objects, static or moving, that are rapidly approaching the UAV and to estimate the time-to-impact. The trajectory generator is alerted if time-to-impact falls below a predetermined threshold and is required to plan a trajectory that *avoids* the obstacle.

Radar is an excellent candidate for collision detection, however for small UAVs, antenna weight and the weight of the associated power supply are prohibitive. Sonars are inexpensive and light weight but with less than thirty foot sensing radius, are inadequate to detect collisions. In addition, they have very poor lateral resolution and are sensitive to false echoes caused by specular reflections. Laser rangefinders are large and bulky and therefore not suited to small and micro UAVs.

As an alternative, small, light-weight cameras are readily available and consume minimal power. In addition, cameras are usually employed in small UAV applications as part of the payload package. Therefore the addition of computer vision for collision detection does not introduce an additional sensor package. The biggest drawback of the vision technology is computational overhead that is required to process video signals in real time. Existing computer vision algorithms are complicated and require extensive computation capacity. Our research aims precisely at the development of simple yet efficient and robust algorithms to address this drawback.

The three enabling technologies needed to solve the see-and-avoid problem are

1. Real-time vision algorithms for collision detection,
2. Real-time guidance algorithms for collision avoidance, and
3. Autopilot technologies to maneuver the UAV.

It is unlikely that the see-and-avoid problem can be solved by finding independent solutions to these three challenges: an integrated approach is essential. The vision algorithms must robustly estimate the time-to-impact even with a low quality, noisy video signal. The vision algorithms must be simple and efficient to provide real-time updates at various altitudes and ground speeds. The guidance algorithms must be able to utilize the time-to-impact and object size information from the vision algorithms to plan the new trajectories and to avoid collision. In addition, the guidance algorithms must guide the UAV around the obstacle, deviating from the original path as little as possible. The autopilot technologies must be able to track and respond to guidance algorithms and maneuver the UAV to a smooth and stable transition for temporary changes of the path.

The remainder of this paper is organized as follows. The state-of-the-art in computer vision and UAV path planning are reviewed in Section 2. This is followed, in Section 3, with a description of the vision and guidance algorithms. Results are presented in Section 4. The proposal then ends with a brief discussion in Section 5.

2. STATE-OF-THE ART

2.1 Collision detection (*See*)

Depth perception capability is what the vision system uses to detect collisions. It can be categorized into two major approaches: stereo vision and motion analysis. Stereo vision needs to solve the correspondence problem which can be a complicated and time-consuming process. Research work has been done using Dynamic Programming (DP) for finding the correspondences points for 3-D reconstruction and robotic vision applications [1-3]. The matching problem can be stated as an optimization task where an objective function, representing the constraints on the solution, is to be minimized. The optimization process can also be performed by means of Genetic Algorithms (GA) [4]. Genetic Algorithms belong to stochastic search methods. They are randomized search and optimization techniques guided by the principles of evolution and natural genetic [5-8]. Both require very complicated processing and may not be able to update the measurements in real time even with a line scan image sensor [4]. Recent work also includes using stereo vision to estimate the camera ego-motion in order to track the robot position [9-10].

Because of the payload limitation and power requirement, stereo vision techniques may not be a good candidate for UAV applications, especially for small and micro UAVs. Other *active* technologies such as radar, sonar, and laser rangefinder, etc., are also excluded for this application because of the limitations discussed in introduction section. Motion analysis, on the other hand, uses a sequence of monocular images to extract 3-D motion and structure. Motion analysis can be broadly categorized into two groups, optical flow and feature tracking. Optical flow can be seen as an

intensity-based method. Techniques based on optical flow have been developed in the past decade [11-18]. Images from the onboard camera can sometimes be very noisy, especially after going through the wireless transmission. Also, the low quality of the long range image, the nature of UAV applications, the attitude changes (rolls, pitches, and yaws), and large depth of field requirements prevent us from using intensity-based techniques.

There are works emphasizing feature tracking that are very robust and avoid the problem of optical flow [19-20]. However, because of the high-speed requirement, low image quality, and rapidly changing UAV attitude, feature tracking may not be reliable. Accurately detecting significant features and tracking them through frames is not a trivial task. Detecting significant features for tracking usually requires a priori knowledge of the scene which is not possible for UAV applications. For similar reasons, a couple of recent developments are not suitable for UAV applications [21-22]. Another very unique method worth evaluation is the measurement of defocus [23]. This approach also posts the biggest challenge, how to measure the defocus. Some alternatives include recovering depth by measuring defocus using the relative blurring between two images of the same scene [24] and using multiple images at different levels of focus and interpolating the sharpness of objects over distance [25]. Shinning illumination patterns on the scene and measuring the defocus of the patterns that have a known sharpness [26] was also studied. The method has drawbacks such as distance range limitation, recovering depth only for only one window in the image, restriction to static environments, and shallow range of depth. Other works attempt to analyze motion using focus of expansion, including using velocity field, neural networks, and connected regions [27-30]. All of these methods require complex computations and are not suitable for UAV applications.

Based on the literature review and the restrictions posed by UAV applications, we developed a simple yet efficient method to detect collisions and estimate the time-to-impact without solving a correspondence problem or accurately detecting and tracking significant features. The method is reliable and easy to use and requires minimal camera calibration. It has no intrusive component, no interference problem, and no physical anomalies such as specular reflection. Most importantly, the method works well with low image quality and is robust with respect to UAV attitude.

2.2 Collision Avoidance (*Avoid*)

In the mobile robotics literature, there are roughly two different approaches to motion planning: de*liberative* motion planning where explicit paths and trajectories are computed based on global world knowledge [31-33], and *reactive* motion planning which uses behavioral methods to react to local sensor information [34,35]. In general, deliberative motion planning is useful when the environment is known *a priori*, but can become computationally intensive in highly dynamic environments. Reactive motion planning, on the other hand, is well suited for dynamic environments, particularly collision avoidance, where information is incomplete and uncertain, but lacks the ability to specify and direct motion plans.

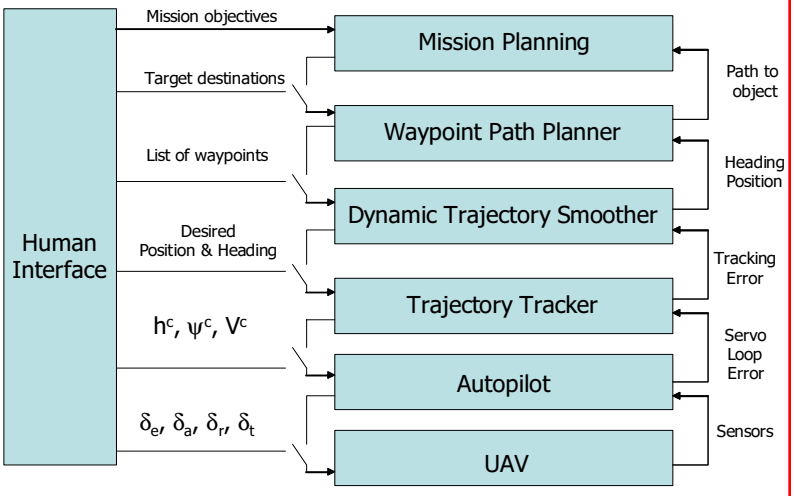

Fig. 1. Architecture of the hierarchical trajectory generation techniques.

The hierarchical trajectory generation techniques used is summarized in the architecture flowchart shown in Fig. 1. At the highest level of the architecture is the waypoint path planner (WPP), which generates waypoint paths given the UAV and the target positions and a world map. Our baseline approach to the WPP is to use a Voronoi [36, 37] algorithm to generate a graph through an obstacle field to the target, and then to search the graph using an Eppstein search [38]. This approach has been used extensively to solve cooperative timing and cooperative control problems [39-45]. The WPP generates a set of waypoints and a target velocity for the dynamic trajectory smoother (DTS). The DTS converts the set of waypoints to a time-parameterized trajectory that satisfies the kinematic, heading-rate, and velocity constraints of the UAV. Our approach to the design of the DTS is to give the trajectory generator the same kinematic structure as the UAV with similar heading-rate and velocity constraints. The trajectories are propagated on-the-fly by selecting the (constrained) reference heading rate to minimize the time deviation from the waypoint path and, optionally,

to match the path length of the waypoint path. Our approach to the DTS is documented in [40, 46-48]. The advantage of this approach is that it is computationally efficient, spreading the computation over the trajectory transition time. In fact, the DTS algorithm reduces to a three-state ODE propagation, a series of half-plane checks, and a root-finding problem for each waypoint segment. The WPP and DTS algorithms have been implemented and tested on a small UAV at BYU [49, 50].

3. ALGORITHMS

3.1 Feature density distribution analysis

As mentioned in Section 1, using computer vision techniques, depth perception can be achieved by either stereo vision or motion analysis of a sequence of monocular images. Motion analysis is more suitable for small UAV applications than stereo vision because of payload and power consumption requirements. Motion analysis can be further categorized into intensity-based (optical flow) and feature-based (feature tracking) techniques. Because of the drawbacks of both categories, we propose a novel computer vision algorithm to achieve the goal of detecting collisions and determining the time-to-impact. When using a method that involves feature point tracking, the feature points are selected based on how easily they can be tracked. However this restriction is no longer necessary when we do not have to track the feature points.

Our approach is based on feature density and distribution (FDD) analysis. It does not require accurate feature detection or feature tracking. Complicated focus of expansion (FOE) calculation is not necessary. It calculates accurate time-to-impact based on the expansion rate of feature density and distribution. Many types of feature points were considered for our approach. Possible feature points include pixels with horizontal edges, vertical edges, corners, and points of a particular color, etc. Some types of feature points may be more useful then others depending on the application. By using different types of feature points simultaneously, it should be possible to achieve better results.

There are several different ways to calculate horizontal and vertical edge pixels. The following method was chosen for its simplicity and its speed. To find pixels on the horizontal and vertical edges simply compute the gradient at each pixel. If the absolute value of gradient in the vertical direction is higher then a threshold, then that point is a horizontal line feature. If the horizontal gradient is above the threshold, then the point is a vertical line feature. Corners are points where there is a change in both the horizontal and vertical directions. Other features such as lines, curves, contours, or specific shapes and models can be used for the FDD analysis but may be complicated and computationally expensive. We chose horizontal and vertical edges for simplicity.

Once the feature points have been selected, we then analyze the distribution of the feature points in space. The see-and-avoid problem mandates that the camera face the direction of motion. We assume that the objects are large enough and close enough that they fill most of the image or the rows on the image that represent the areas in the path. This is a safe assumption as long as the camera does not move very fast compared to the frame rate and the objects are reasonably large. Based on these assumptions, we know that if the camera is approaching an object, the distribution of the feature points will expand. For collision detection, only the approaching objects will be considered. The rate of expansion can be used to estimate the speed at which we are approaching the object and can be used to estimate the time-to-impact. Another assumption is that the rotational variation is minimal between two consecutive frames compared to the translation.

3.2 Rate of Feature Distribution Expansion

To calculate the rate of expansion of the feature distribution in the *x*-direction, we first compute the number of feature points across the image at every value of *x* at a particular time value *t*, this will be denoted by *F(x,t)*. *F(x,t)* can also be viewed as the vertical projection of the feature points or the histogram of feature points on each image column. If the image expands (objects are getting closer) and there is a motion in the x-direction, then there will be a scale and an offset in the next frame. Ideally, the feature point projection of the frame at *t+dt* should be

$$F(ax + d, t + dt) = F(x, t) \tag{1}$$

, where *a* is the rate of expansion and *d* is the movement in the *x*-direction. To estimate *a* and *d* we minimize the mean-squared error over the *x*-values 0 through *N* (number of image columns):

$$E(a,d)=\sum_{x=0}^{N}[F(ax+d,t+dt)-F(x,t)]^2. \quad (2)$$

For a small value of *d* and a value of *a* near 1,

$$F(ax+d,t+dt)\approx F(x,t+dt)+\frac{\partial}{\partial x}F(x,t+dt)[(a-1)x+d] \text{ and}$$

$$E(a,d)=\sum_{x=0}^{N}[F(x,t+dt)+\frac{\partial}{\partial x}F(x,t+dt)[(a-1)x+d]-F(x,t)]^2. \quad (3)$$

For convenience, let F_t be the time-derivative and F_x be the spatial-derivative of the feature point function and let $D = a-1$:

$$E(D,d)=\sum_{x=0}^{N}[F_t(x,t)+F_x(x,t+dt)(Dx+d)]^2. \quad (4)$$

To minimize the error, we will take the derivative and set it to zero to obtain

$$\frac{\partial}{\partial d}E(D,d)=2\sum_{x=0}^{N}[F_t(x,t)+F_x(x,t+dt)(Dx+d)]\cdot F_x(x,t+dt)=0$$

$$\frac{\partial}{\partial D}E(D,d)=2\sum_{x=0}^{N}[F_t(x,t)+F_x(x,t+dt)(Dx+d)]\cdot F_x(x,t+dt)\cdot x=0$$

$$\begin{bmatrix}\sum F_x^2x^2 & \sum F_x^2x\\ \sum F_x^2x & \sum F_x^2\end{bmatrix}\begin{bmatrix}D\\ d\end{bmatrix}=\begin{bmatrix}\sum F_xF_tx\\ \sum F_xF_t\end{bmatrix}. \quad (5)$$

The quantities $a = D+1$ and d can be computed from Equation (5). This method can be iteratively refined as follows. If we define a new function $F_1(x, t + dt) = F(ax + d, t + dt)$ and then apply the same procedure but replacing the function $F(x, t + dt)$ by $F_1(x, t + dt)$, then we can calculate values of a and d for each iteration. For three iterations, we use the following calculations

$$F_1(x,t+dt)=F(a_0x+d_0,t+dt) \qquad F_2(x,t+dt)=F_1(a_1x+d_1,t+dt)$$
$$F_3(x,t+dt)=F_2(a_2x+d_2,t+dt) \quad (6)$$
$$F(x,t)\approx F_3(x,t+dt)\approx F(a_2(a_1(a_0x+d_0)+d_1)+d\approx F(ax+d),$$

where $a=a_2a_1a_0$ and $d=a_2a_1d_0+a_2d_1+d_2$.

3.3 Time-to-Impact

After calculating the expansion rate of the object, *a*, we can calculate the amount of time we have before we will impact the object at the current velocity. This is a useful calculation for obstacle avoidance. Fig. 2 shows a diagram of a point moving toward the camera and its projection onto the image plane. (It makes no difference if the object is moving toward the camera or the camera is moving towards the object, since the relative velocity is all that matters.)

For a calibrated camera we also know the focal length *f* and other intrinsic parameters. The object size, *X'* and *X*, and the object distance, *z'*, and *z* are unknown. We know that the points on the image plane, *x'* and *x* are related as $x'=ax$, where *a* is the expansion rate. From the geometry we know that

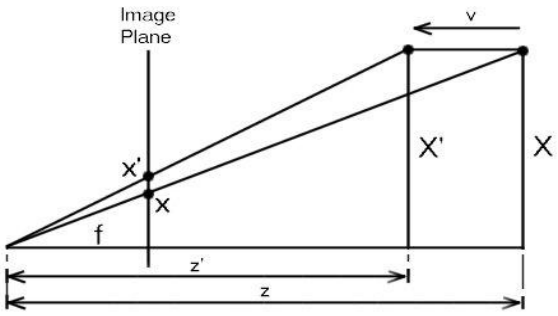

Fig. 2. Diagram of an object moving towards the camera.

$$\frac{x}{f} = \frac{X}{z} \quad \text{and} \quad \frac{x'}{f} = \frac{X'}{z'} = \frac{X}{z'}$$

$$\text{and } z' = X\frac{f}{x'} = \frac{zx}{f}\cdot\frac{f}{ax} = \frac{z}{a}. \tag{7}$$

The time to impact, τ, is the distance, z, divided by the velocity, v:

$$\tau = \frac{z}{v} = \frac{z}{z - z'} = \frac{z}{z - z/a} = \frac{a}{a-1}. \tag{8}$$

Because we know the speed that the camera is approaching the object (from onboard GPS), we can also calculate the distance to the object from the expansion rate. The expansion rate is not constant over time: for an object approaching the camera at a fixed velocity, v, $a(t)$ is given by

$$a(t) = \frac{z_0 - tv}{z_0 - (t+1)v}, \text{ where } z_0 \text{ is the value of } z \text{ at } t = 0. \tag{9}$$

3.4 Real-time Obstacle Avoidance

In this section we will describe our approach to trajectory generation for the see-and-avoid problem. The essential insight is to note that the kinematic structure of a UAV flying at a constant altitude is given by

$$\dot{x} = v\cos(\psi)$$
$$\dot{y} = v\sin(\psi)$$
$$\dot{\psi} = \omega,$$

where $-\omega_{\max} \le \omega \le \omega_{\max}$, and $0 < v_{\min} \le v \le v_{\max}$. If the UAV is flying at constant velocity, then the local reachability region up to time T is shown in grey in the Fig. 3. The turning rate constraint translates into the minimum turn radius shown in the figure. In our previous work with path planning and trajectory generation, we have used a deliberative approach to path planning and trajectory generation [40-50]. When dynamic and pop-up threats would occur, our approach has been to re-plan the associated trajectories. However, in highly dynamic environments, the computational resources associated with complete re-planning may not allow fast enough reaction time to threats and obstacles detected by the computer vision algorithms.

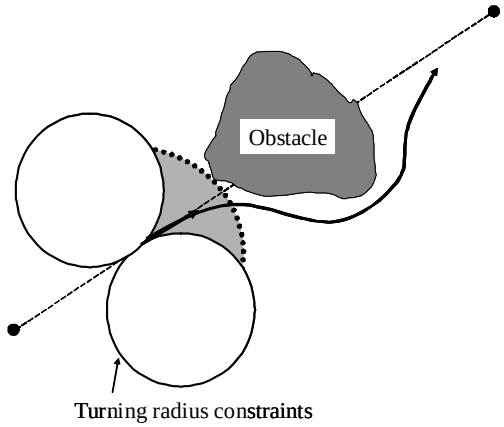

Fig 3. Turning radius constraints for path planning.

We integrated reactive path planning and trajectory generation techniques with our current (deliberative) approach. As with our deliberative approach, this will be accomplished without violating the velocity and heading rate constraints of the UAV. A reactive planning phase will be added at both the WPP (low-bandwidth) level and the DTS (high-bandwidth) level. At the low bandwidth level, our basic approach is to use the incremental Voronoi construction techniques introduced in [51] to add edges to the Voronoi diagrams as the computer vision algorithms detect obstacles and threats that were not included in an *a priori* known low resolution world map, which is assumed to be available for deliberative planning. The addition of new edges will initiate a re-plan of the waypoint paths. It the time-to-impact is large enough, then re-planning is feasible. However, if the time-to-impact is small, then the UAV needs to react instantaneously.

Our basic approach to high bandwidth reaction is shown in Fig. 4. Suppose that the UAV is tracking a waypoint path segment as shown in the figure, and that an obstacle is suddenly detected by the vision algorithm. A constant time-to-impact curve is indicated by the boarder of the shaded half circle. The trajectory generator must select a heading rate that is both dynamically feasible, and also steers the UAV around the obstacle as illustrated in the figure. If we quantize the heading rate command and let the DTS flow a time T into the future, we obtain a set of locally reachable points denoted by $R(T)$ and indicated by dots in the figure. A simple reactive scheme is to select a point in $R(T)$ that does not intersect the obstacle and minimizes the distance from the desired waypoint line segment.

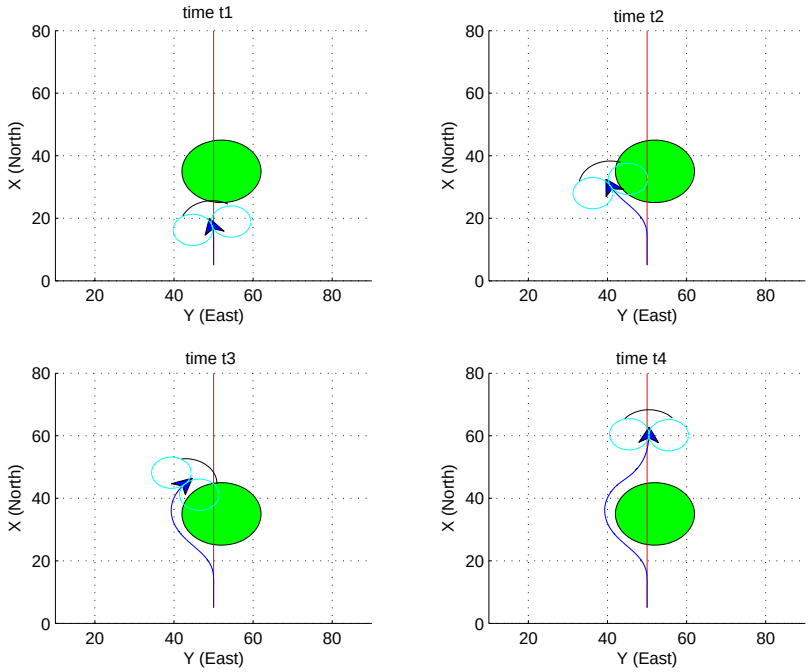

Fig. 4. Trajectory generation.

4. RESULTS

The proposed method was tested on a video sequence with total of 32 frames. The video sequence was made in a controlled environment with a bottle approaching the camera at a fixed speed. Figure 5 shows three frames (#0, #15, and #30) of the original image sequence for proof of concept. Each frame was taken at a fixed speed when approaching the camera. It can be seen that the image size increases as the object moves closer. The rate of expansion measured by the feature density and distribution as we proposed can be used to calculate the time of impact as discussed in previous sections. Figure 6 shows three types of features proposed to be used for feature density and distribution analysis for the calculation of the rate of expansion.

Fig. 5. Frames 0, 15, and 30 of the original image sequence that is used for the proof of concept.

Fig. 6. Horizontal, vertical, and corner features of Frame 0.

Figure 7 shows the horizontal projections (projected to the y-axis) of the horizontal feature points of frames 0 and 5 as an example. Because the rate of expansion between two consecutive frames is very small in this case, we show the results from frames 0 and 5 to show the effect of expansion over several frames. Similarly, Figure 8 shows the vertical projections (projected to the x-axis) of the vertical feature points of Frames 0 and 5. The feature point density and distribution has a slight expansion between the two frames.

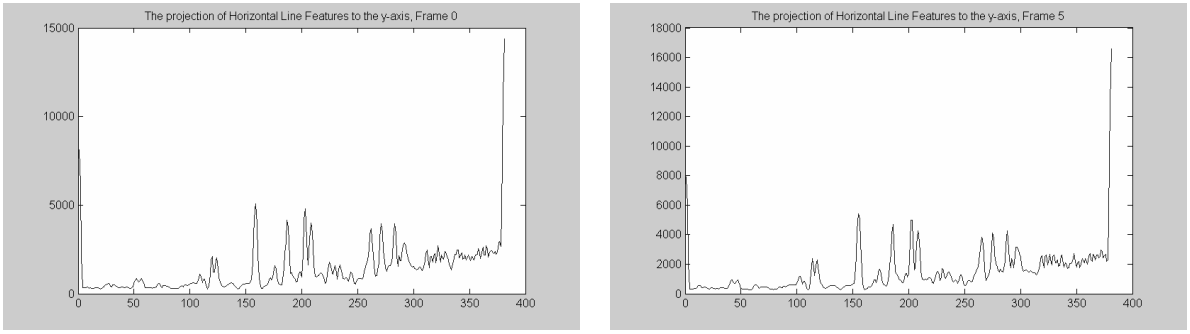

Fig. 7. The horizontal line feature points projected to the y-axis (row) of the image.

Besides the rate of feature distribution expansion (*a*), there also exists a slight shift (*d*) in the horizontal projection in Figure 7 and the vertical projection in Figure 8. This shift may be caused by several different factors. If the UAV rolls, the image will rotate. If the UAV pitches or yaws, the image will move vertically or horizontally. Each of these factors will cause a small shift in the feature distribution. The shift is calculated to reduce the inaccuracies introduced by these factors.

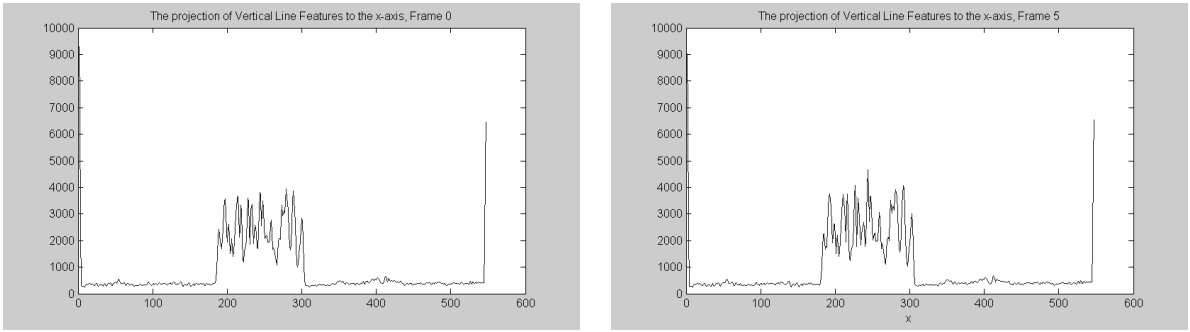

Fig. 8. The vertical line feature points projected to the x-axis (column) of the image.

The scale factor, *a*, was calculated at each frame to track the change in time-to-impact. We calculated the scale factor (or the rate of expansion) for both horizontal and vertical features to verify the accuracy. The results, after ten iterations of our method, are shown in Figure 9. The results after 3 to 5 iterations do not show any significant improvement in accuracy with further iterations. The dotted line in Figure 9 shows the expected rate of expansion and the solid curve indicates the expansion calculated using our method. Both horizontal and vertical feature expansions are very close to the expected values.

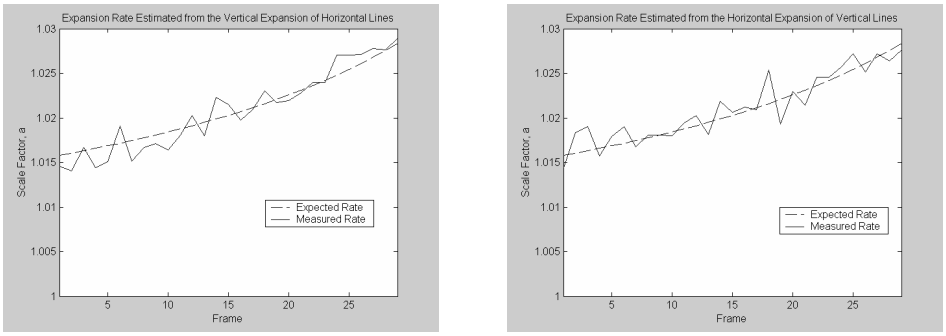

Fig. 9. The horizontal and vertical feature distribution expansion rates of all 32 frames.

From the expansion rate, the time to impact can be calculated at each frame using Equation (8). The time to impact results are shown in Figure 10. They are represented as the number of frames. Based on a 30 frames/second frame rate, the time to impact can be calculated by multiplying the number of frames by 33.3 milliseconds. If the camera moving velocity is known, which is true for our UAV applications, the actual distance to impact can be calculated. As shown in Figure 10, the numbers are fairly close to the correct value, but with a slight variation. Better results can be obtained by averaging the expansion rates over several frames.

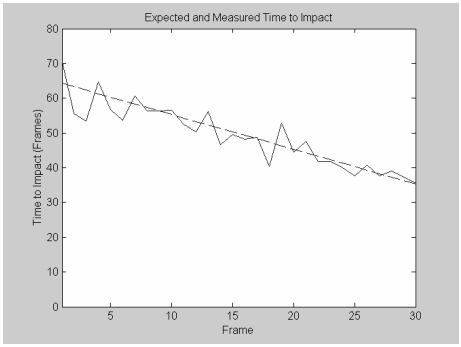

Fig. 10. Expected and measured time to impact (measured as # of frames).

Figure 11 shows an example of two image frames (Frames 103 and 108) from an image sequence of a real scene acquired with a camera on-board a small UAV. The UAV's roll, pitch, and yaw cause a small rotation as well as a slight shift in both horizontal and vertical directions between the two frames. This effect can be seen by comparing the location of the tree in two frames and the level of the horizon. Horizontal and vertical feature points of Frame 103 are shown in Figure 12. The horizontal and vertical projections of these feature points of Frame 103 are shown in Figure 13. Preliminary results show that the features of the tree standing in the path can be detected and the proposed method can be used to detect the time to impact of the tree.

Fig. 11. Examples of real scenes from camera on-board a UAV and their feature points.

The proposed reactive scheme has been implemented in Simulink, and the results at several instances of time are shown in the figure above. Note that the algorithm effectively avoids obstacles, while minimizing the deviation from the original waypoint line. The computations associated with the proposed scheme include point-intersection checks, the inversion of a 2×2 matrix, and a one-dimensional line search, and can therefore be implemented in a computationally efficient manner suitable for real-time implementation.

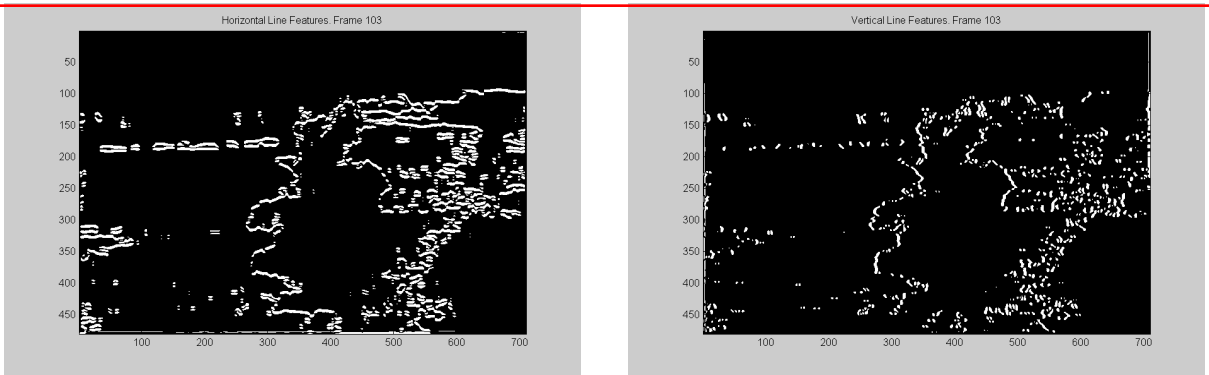

Fig. 12. Horizontal and vertical feature points of Frame 103.

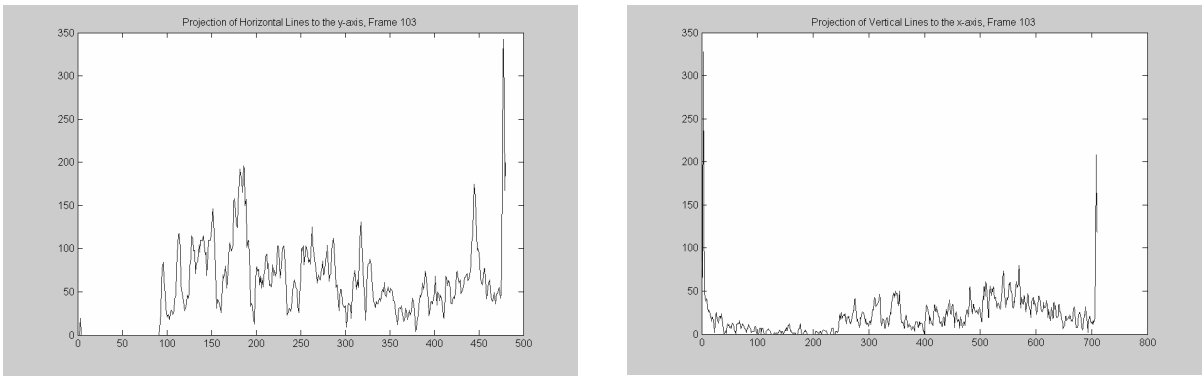

Fig. 13. Horizontal and vertical projections of the feature points of Frame 103.

The advantages of the proposed see-and-avoid scheme is that it is computationally feasible, explicitly satisfies velocity and heading rate constraints, and includes the advantages of both reactive and deliberative methods. That is, deliberative methods are used to exploit *a priori* and medium-to-long-range vision information when it is available, and reactive methods are used to quickly respond to dynamically moving and sudden pop-up obstacles.

5. CONCLUSION

The objective of this paper is to investigate a novel approach to the see-and-avoid problem for small UAVs. The essential idea is to estimate the time-to-impact from a sequence of monocular images. It is a simple but efficient method that circumvents the correspondence problem by not requiring accurate feature detection and tracking. It is robust to low image quality. Further work is needed to better estimate and compensate for the UAV attitude changes (rolls, pitches, and yaws) and consider the existence of feature points at oblique angles. Other types of feature points may be more suitable for different applications and may be worth further investigation. One significant challenge that is introduced by this new method is how to separate and effectively deal with images in which there may be objects in the foreground, close to the camera and objects in the background that are further away. The problem is that there are two different expansion rates and so it may not be possible to accurately describe the scene with a single expansion rate. Further work is needed in this area.

REFERENCES

1. Reinhard Koch, “3-D Surface Reconstruction from stereoscopic image sequences”, Proceedings of the Fifth International Conference on Computer Vision, pp. 109-114, Cambridge, MA, June 1995.
2. R. C. Gonzalez, J. A. Cancelas, J. C. Alvarez, J. A. Fernandez, and J. M. Enguita, “Fast stereo vision algorithm for robotic applications”, Proceedings of IEEE International Conference on Emerging Technologies and Factory Automation (ETFA '99), pp. 97 – 104, vol.1, Barcelona, Spain Oct. 1999.

3. G. L. Gimel'farb, "Intensity-based bi- and trinocular stereo vision: Bayesian decisions and regularizing assumptions", Proceedings of the 12th IAPR International Conference on Pattern Recognition, Conference A: Computer Vision & Image Processing, pp. 717 – 719, vol.1, Jerusalem, Israel, Oct. 1994.
4. Yassine Ruichek, Hazem Issa, and Jack-Gerard Postaire, "Approach for obstacle detection using linear stereo vision", Proceedings of the IEEE Intelligent Vehicles Symposium 2000, Dearborn, MI, October 2000.
5. Goldberg, "Genetic algorithms in search optimization and machine learning", Addison-Wesley, Reading MA, 1989.
6. Erick Cantu-Paz, "Efficient and accurate parallel genetic algorithms", Kluwer Academic Publishers, Boston, MA, 2000.
7. Lance Chambers, "The Practical handbook of genetic algorithms: applications", Chapman & Hall, Boca Raton, FL, 2001.
8. S Woo and A. Dipanda, "Matching lines and points in an active stereo vision system using genetic algorithms", Proceedings. 2000 International Conference on Image Processing, pp. 332-335, vol.3, Vancouver, BC, Canada, Sept. 2000.
9. M. S. Clark F. Olson, Larry H. Matthies and M. W. Maimone, "Rover navigation using stereo ego-motion," Robotics and Autonomous Systems, vol. 43, Jun 2003.
10. R. K. T. Gandhi, S. Devadiga and O. Camps, "Detection of obstacles on runway using egomotion compensation and tracking of significant features," Proceedings 3rd IEEE Workshop on Applications of Computer Vision, 1996. WACV 96., pp. 168–173, Dec 1996.
11. S.-C. Pei and L.-G. Liou, "What can be seen in a noisy optical flow field projected by a moving planar patch in 3d space," Pattern Recognition, vol. 30, pp. 1401–1413, Sep 1994.
12. M. H. T. Camus, D. Coombs and T.-H. Hong, "Real-time single-workstation obstacle avoidance using only wide-field flow divergence", Proceedings of the 13th International Conference on Pattern Recognition, vol. 3, pp. 323–330, Aug 1996.
13. M.W. G. Barato_, C. Toepfer and H. Neumann, "Real-time navigation and obstacle avoidance from optical flow on a space-variant map," National Aerospace and Electronics Conference, 2000. NAECON 2000. Proceedings of the IEEE 2000, pp. 289–294, Sep 1998.
14. M. H. G. Young, T. Hong and J. Yang, "Obstacle detection for a vehicle using optical flow," Proceedings of the Intelligent Vehicles '92 Symposium., pp. 185–190, Jul 1992.
15. T. H. G. Young and M. Gaithersburg, "New visual invariants for obstacle detection using optical flow induced from general motion," Proceedings, 1992., IEEE Workshop on Applications of Computer Vision, pp. 100–109, Dec 1992.
16. K. Song and J. Huang, "Fast optical flow estimation and its application to real-time obstacle avoidance," Proceedings 2001. ICRA. of IEEE International Conference on Robotics and Automation, vol. 3, pp. 2891–2896, 2001.
17. S. Sull and B. Sridhar, "Model-based obstacle detection from image sequences," 1995. Proceedings, International Conference on Image Processing, vol. 2, pp. 647–650, Oct 1995.
18. V. T. E. Micheli and S. Uras, "The accuracy of the computation of optical flow and of the recovery of motion parameters," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, pp. 434–447, May 1993.
19. S. Krishnan and D. Raviv, "2D feature tracking algorithm for motion analysis," Pattern Recognition, vol. 28, pp. 1103–1126, Aug 1995.
20. G. Desouza and A. Kak, "Vision for mobile robot navigation: a survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 237–267, Feb 2002.
21. D. R. Sridhar R. Kundur, "A vision-based pragmatic strategy for autonomous navigation," Pattern Recognition, vol. 31, pp. 1221–1239, Sep 1998.
22. S. Sull and B. Sridhar, "Runway obstacle detection by controlled spatiotemporal image flow disparity," IEEE Transactions on Robotics and Automation, vol. 15, pp. 537–547, June 1999.
23. C. T. Illah R. Nourbakhsh, David Andre and M. R. Genesereth, "Mobile robot obstacle avoidance via depth from focus," Robotics and Autonomous Systems, vol. 22, pp. 151–158, Nov 1997.
24. A. Petland, T. Darrell, M. Turk, W. Huang, "A simple real-time range camera", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, P. 256-261, 1989.
25. T. Darrell and K. Wohn, "Pyramid based depth from focus", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, P. 504-509, 1988.
26. M. Rioux and F. Blais, Compact three-dimensional cameras for robotic applications", Journal of the Optical Society of America, vol. 3, no. 9, p. 1518-1521, 1986.

27. R. Guissin and S. Ullman, "Direct computation of the focus of expansion from velocity field measurements," 1991 Proceedings of the IEEE Workshop on Visual Motion, pp. 146–155, Oct 1991.
28. A. B. G. Convertino and A. Distante, "Focus of expansion estimation with a neural network," 1996 IEEE International Conference on Neural Networks, vol. 3, pp. 1693–1697, Jun 1996.
29. K. Rangarajan and M. Shah, "Interpretation of motion trajectories using focus of expansion," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, pp. 1205–1210, Dec 1992.
30. W. Burger and B. Bhanu, "On computing a 'fuzzy' focus of expansion for autonomous navigation," 1989. Proceedings CVPR '89., IEEE Computer Society Conference on Computer Vision and Pattern Recognition,, pp. 563–568, Jun 1989.
31. D. Hsu, R. Kindel, J.C. Latombe, and S. Rock "Randomized Kinodynamic Motion Planning with Moving Obstacles," Algorithmic and Computational Robotics: New Directions, A.K. Peters, pp. 247-264, 2001.
32. F. Lamiraux, S. Sekhavat, and J.P. Laumond, "Motion Planning and Control for {H}ilare Pulling a Trailer," IEEE Transactions on Robotics and Automation, vol. 15, no. 4, August, pp. 640-652, 1999.
33. R. M. Murray and S. S. Sastry, "Nonholonomic Motion Planning: Steering Using Sinusoids," IEEE Transactions on Automatic Control, vol. 38, no. 5, May, pp. 700-716, 1993.
34. T. Balch and R.C. Arkin, "Behavior-Based Formation Control for Multirobot Teams," IEEE Transactions on Robotics and Automation, vol. 14, no. 6, December, pp. 926-939, 1998.
35. R. C. Arkin, "Behavior-based Robotics," MIT Press, 1998.
36. F. Aurenhammer, "Voronoi diagrams - a survey of fundamental geometric data struct", ACM Computing Surveys, vol. 23, pp. 345-405, September 1991.
37. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, "Introduction to Algorithms," McGraw-Hill, 1993.
38. D. Eppstein, "Finding the k shortest paths," SIAM Journal of Computing, vol. 28, no. 2, pp. 652 -673, 1999.
39. P. Chandler, S. Rasumussen, and M. Pachter, "UAV cooperative path planning," Proceedings of the AIAA Guidance, Navigation, and Control Conference, Denver, CO , August 2000, AIAA Paper No. AIAA-2000-4370.
40. R. W. Beard, T. W. McLain, M. Goodrich, and E. P. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles," IEEE Transactions on Robotics and Automation, vol. 18, pp. 911-922, December 2002.
41. T. McLain and R. Beard, "Cooperative rendezvous of multiple unmanned air vehicles," Proceedings of the AIAA Guidance, Navigation and Control Conference, Denver, CO, August 2000, AIAA-2000-4369.
42. T. W. McLain and R. W. Beard, "Coordination variables, coordination functions, and cooperative timing missions," Proceedings of the American Control Conference, pp.~296--301, 2003.
43. T. W. McLain, P. R. Chandler, S. Rasmussen, and M. Pachter, "Cooperative control of UAV rendezvous," Proceedings of the American Control Conference, Arlington, VA, pp. 2309-2314, June 2001.
44. T. McLain, P. Chandler, S. Rasmussen, and M. Pachter, "Cooperative control of UAV rendezvous, Proceedings of the American Control Conference, pp. 2309-2314, June 2001.
45. T. W. McLain, R. W. Beard, and J. Kelsey, "Experimental demonstrations of multiple robot cooperative target intercept," Proceedings of the AIAA Guidance, Navigation and Control Conference}, Monterey, CA, August 2002, AIAA Paper No. 2002-4678.
46. E. P. Anderson, "Constrained extremal trajectories and unmanned air vehicle trajectory generation," Master's thesis, Brigham Young University, Provo, Utah 84602, April 2002.
47. E. P. Anderson and R. W. Beard, "An algorithmic implementation of constrained extremal control for UAVs," Proceedings of the AIAA Guidance, Navigation and Control Conference, Monterey, CA, August 2002. AIAA Paper No. 2002-4470.
48. E. P. Anderson, R. W. Beard, and T. W. McLain, "Real time dynamic trajectory smoothing for uninhabited aerial vehicles," IEEE Transactions on Control Systems Technology, (to appear).
49. D. Kingston, R. Beard, T. McLain, M. Larsen, and W. Ren, "Autonomous vehicle technologies for small fixed wing UAVs," AIAA 2nd Unmanned Unlimited Systems, Technologies, and Operations-Aerospace, Land, and Sea Conference and Workshop & Exhibit, San Diego, CA, September 2003. Paper no. AIAA-2003-6559.
50. T. W. McLain and R. W. Beard, ``Unmanned air vehicle testbed for cooperative control experiments," Submitted for publication in the American Control Conference, Boston, MA, June 2004.
51. H. Choset, S. Walker, K. Eiamsa-Ard, and J. Burdick, "Sensor-based exploration: incremental construction of the hierarchical generalized Voronoi graph," The International Journal of Robotics Research, vol. 19, no. 2, pp. 116-148, 2000.