International Micro Air Vehicle Conference and Flight Competition (IMAV2013)
17-20 September 2013, Toulouse, France

Obstacle mapping module for quadrotors on outdoor Search and Rescue operations

Andrew Nolan[1], Daniel Serrano[2], Aura Hernandez Sabaté[1], Daniel Ponsa Mussarra[1] and Antonio M. López Peña[1]

[1] Centre de Visió per Computador, Universitat Autònoma de Barcelona, Bellaterra, Spain
andrewpeternolan@gmail.com
[2] Ascamm Private Foundation, Barcelona Cerdanyola del Vallès, Spain
dserrano@ascamm.com

Abstract

Obstacle avoidance remains a challenging task for Micro Aerial Vehicles (MAV), due to their limited payload capacity to carry advanced sensors. Unlike larger vehicles, MAV can only carry light weight sensors, for instance a camera, which is our main assumption in this work. We explore passive monocular depth estimation and propose a novel method Position Aided Depth Estimation (PADE). We analyse PADE performance and compare it against the extensively used Time To Collision (TTC). We evaluate the accuracy, robustness to noise and speed of three Optical Flow (OF) techniques, combined with both depth estimation methods. Our results show PADE is more accurate than TTC at depths between 0-12 meters and is less sensitive to noise. Our findings highlight the potential application of PADE for MAV to perform safe autonomous navigation in unknown and unstructured environments.

1 Introduction

The use of Unmanned Aerial Vehicles (UAV) has become an important part of search and rescue operations in recent years. UAV can be used to assist human teams with multiple operations; from exploring a disaster area, to dealing with the difficult task of finding human survivors. The European Union project, Integrated Components for Assisted Rescue and Unmanned Search operations (ICARUS EU-FP7) [5] aims to improve unmanned air vehicles to integrate them into existing emergency service systems. As part of the ICARUS project, Aerospace technology centre of Fundació ASCAMM (Spain), is developing a small quadrotor platform to aid in Urban Search and Rescue (USAR) operations. ASCAMM's quadrotor is considered a Micro Aerial Vehicle (MAV) due to its size and limited payload capacity. USAR scenarios require MAV to perform safe autonomous navigation in unknown and unstructured outdoor environments. In order to autonomously navigate MAV needs to plan paths that avoid collisions with any obstacles.

Path planning can be described in two steps; firstly, to use sensors to generate a description of the environment and identify any potential obstacles. Then, to find a feasible path between two points using the environment description that does not result in a collision [27]. A description of the environment can be achieved through depth estimation using a variety of active and passive sensors. Active depth sensing uses physical sensors such as laser or structured light patterns [25] to obtain depth information from natural scenes [11]. For large quadrotors, collision avoidance has been solved using active sensors such as laser range finders [22] or depth cameras [10]. However, on MAV with limited payloads, active sensors remain prohibitive because they are power consuming and heavy. In

contrast to active sensors, passive visual sensors such as cameras present a light weight option for collision avoidance on MAV.

Passive sensors such as stereo or monocular cameras estimate depth using stereopsis methods that mimic the human visual system by calculating depth from two slightly different viewpoints of the same scene [20]. Stereo cameras use distance between each viewpoint to recover depth by measuring the relative displacement of an object on two image planes [14]. Similarly, depth is recovered from a monocular camera by estimating motion on the image plane using Optical Flow (OF) methods and measuring camera translation and rotation [18, 27]. Monocular cameras are best suited for visual navigation on MAV due to their light weight and low power requirements.

The primary focus of our work is the investigation of monocular depth estimation techniques using stereopsis methods. For robotic (monocular) depth estimation, the most extensively used technique is called Time To Collision (TTC) [23, 17, 4]. TTC estimates, for each pixel, the number of frames until a collision occurs. The TTC algorithm starts by removing the effects of camera rotation from the flow field by either estimating camera rotation or using measurements obtained from an Inertial Measurement Unit (IMU). Next, the location of the flow field's Focus of Expansion (FOE) is estimated and used to calculate the time to collision for each flow vector. Using the vehicle's translational velocity, each TTC value is converted to a depth estimate [23, 28]. TTC relies on an accurate FOE estimation, which in practice is not a trivial task for arbitrary camera motion and especially difficult under noisy conditions [2].

In our work, we investigate possible methods for obtaining a description of the environment using depth estimation in the context of a quadrotor MAV. Many OF based depth estimation techniques have been presented for MAV [16, 26] operating indoors and outdoors, although few have incorporated camera position to improve depth estimation. Our state of the art review has shown a lack of research combining OF techniques with tightly coupled Global Positioning System (GPS) and Inertial Navigation System (INS) measurements for depth estimation. Fusing GPS/INS measurements with an Extended Kalman filter (EKF) [6] improves the accuracy of relative position estimates [1]. In our work we propose a novel Position Aided Depth Estimation (PADE) method and compare its accuracy and computational performance against TTC. We also evaluate the robustness of both methods under noise.

2 Methodology

Our general processing pipeline for monocular depth estimation, shown in Figure 1, is constructed of two processing stages: Data Acquisition and PADE. The Data Acquisition stage uses two consecutive images to calculate optical flow and also calculates the relative camera translation and rotation from absolute position and orientation measurements. The PADE stage uses OF and camera rotation and translation to calculate a Depth Image.

Motion observed on the image plane is known as Image Flow. Image Flow is the result of the projection of an object's 3D velocity on the image plane [8] generated by objects in the scene moving or through camera movement. To approximate image flow, OF methods detect motion between two consecutive images for each pixel, described by a 2D vector $[uv]^T$. If we assume the scene is static and the observed motion is generated by camera translation and rotation, the computed OF can be used to reconstruct 3D structure of a scene through flow divergence techniques [18].

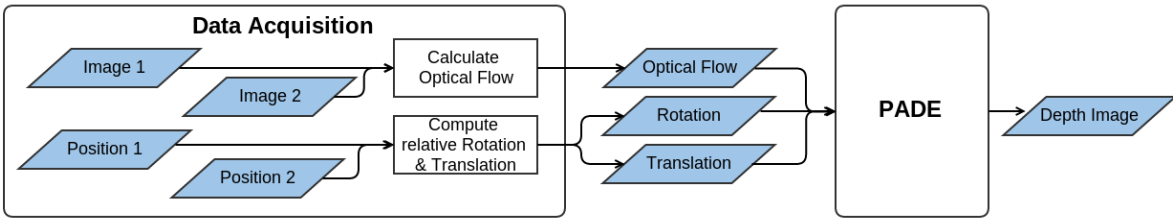


Figure 1: Monocular depth estimation pipeline

2.1 Optical Flow

The relationship between camera motion (translation and rotation) and OF can be defined as the velocity of a 3D point in space with its corresponding velocity on the image plane, shown in the equation:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{Z}\mathbf{A}\mathbf{t} + \mathbf{B}\omega \tag{1}$$

Matrices $\mathbf{A}$ and $\mathbf{B}$ are constructed using the camera focal length f and pixel coordinates x and y.

$$\mathbf{A} = \begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix}, \mathbf{B} = \begin{bmatrix} -\frac{xy}{f} & f + \frac{x^2}{f} & -y \\ -f - \frac{y^2}{f} & \frac{xy}{f} & x \end{bmatrix}$$

Camera translation and rotation are defined by vectors $\mathbf{t} = [t_t, t_y, t_z]^T$ and $\omega = [r_x, r_y, r_z]^T$ and are expressed relative to the first image frame. In our work we use the convention t_z as forward, t_x as right, t_y as down for camera translation and rotation is represented as using typical avionics notation r_x for roll, r_y for pitch, r_z for yaw.

Flow divergence generated by pure camera translation indicates the 3D location of stationary objects in a scene. Flow divergence methods assume that an object increases in size on the image plane as the distance to the object decreases, thus flow magnitude is inversely proportional to the depth of a point to the camera. Using the spatial arrangement cues generated by the relative motion between a scene and a viewer we can reconstruct a model of the observed scene.

2.2 Compute Relative Translation and Rotation

Equation (1) defines OF as the result of objects 3D velocity on the image plane [8]. If we assume that the scene is static, then we can suppose that motion observed on the image plane is the result of camera translation and rotation. The camera's relative translation $\mathbf{t} = [t_t, t_y, t_z]^T$ is the position displacement between the position p at time t and the position at time $t + 1$, shown in the following equation:

$$\mathbf{t} = \mathbf{p}_{t+1} - \mathbf{p}_t \tag{2}$$

Similarly, relative camera rotation $\omega = [r_x, r_y, r_z]^T$ is the difference between the orientation o at time t and the orientation at time $t + 1$, shown in the following equation:

$$\omega = \mathbf{o}_{t+1} - \mathbf{o}_t \tag{3}$$

2.3 Position Aided Depth Estimation (PADE)

Monocular depth estimation consists of reconstructing a 3D scene using depth cues extracted from consecutive image frames after camera translation and rotation. The relative depth image of the environment can be computed from OF and position information. Our PADE method uses relative camera position to improve OF depth estimation. To estimate depth, it is not necessary to compute the FOE since we already have the camera's translation and rotation $[\mathbf{t}, \omega]$. Figure 2.3 shows our PADE pipeline as two steps: Average Optical Flow and Calculate Depth Image.

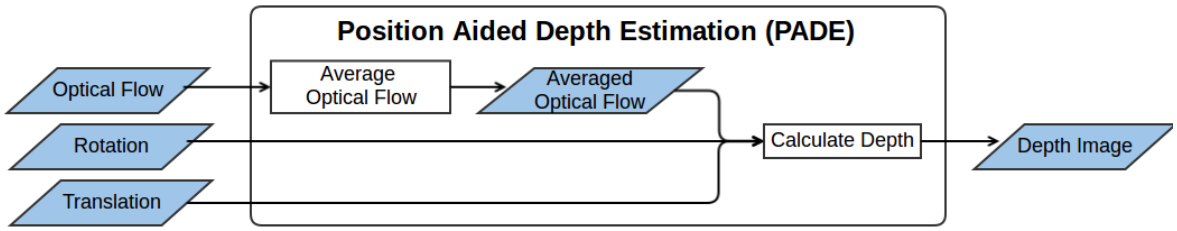


Figure 2: Position Aided Depth Estimation (PADE) pipeline

2.3.1 Average Optical Flow

Computing a dense depth image from a dense OF field is computationally expensive because a depth value is computed for each flow vector. We aim to improve depth estimation performance by reducing the number of calculations. We calculate the average of all flow vectors within a $n \times n$ window and use it to compute a single depth value for the whole window. By reducing the number of depth calculations we can improve computational performance by $(n \times n) - 1$. Averaging OF also has the advantage of smoothing noisy vectors.

2.3.2 Calculate Depth

We isolate Z from equation (1) to compute depth using OF $[u, v]^T$, translation and rotation $[\mathbf{t}, \omega]$. By rearranging the terms in equation (1) we obtain the distance to a point Z, as follows:

$$Z = \mathbf{At} \cdot (\begin{bmatrix} u \\ v \end{bmatrix} - \mathbf{B}\omega)^{-1} \tag{4}$$

The camera's extrinsic parameters $\mathbf{t}$ and ω between image frames plus matrices $\mathbf{A}$ and $\mathbf{B}$ (defined in Section 2.1) allow us to solve for Z using an estimate of OF. Equation 4 requires translation $|\mathbf{t}| > 0$ for depth estimation.

3 Experiments

3.1 Measures

In this study, we have investigated multiple methodologies for obstacle mapping on small UAVs. Our primary research focus is the comparison of two monocular depth estimation techniques: TTC and PADE. To provide a quantitative assessment of both depth estimation techniques we used Robot

Operating System (ROS) [21] with the robotic simulator Gazebo [12] to obtain ground truth depth images. We generated ten image sequences captured from a virtual quadrotor flying through a 3D scene. Our sequences, totalling 1038 frames at a resolution of 640×480, isolate many combinations of translation and rotation in order to explore the robustness of each depth estimation technique.

OF based solutions are typically a trade off between accuracy and speed. Our work focuses on evaluating methods with the potential of real-time speed at the expense of accuracy. We evaluate two dense global OF methods (CLG [3], Farnebäck [7]) and one local method (Lucas-Kanade [13]).

Absolute Depth Error (ADE)

We evaluate two depth estimation methods TTC and PADE combined with each OF method, using Absolute Depth Error (ADE). ADE is the difference between the estimated distance and the ground truth distance.

$$ADE = |Z_C - Z_T| \tag{5}$$

3.2 Results

In order to assess the accuracy at different depth ranges we have calculated ADE at 4 meter intervals (0-4m, 4-8m, 8-12m, 12-16m). Table 1 is divided into two blocks, TTC and PADE, and for each depth estimation method we evaluate the contribution of each OF method. For each 4 meter interval (columns), we report the average error and standard deviation. The minimum error for each depth range is highlighted in bold. Lucas-Kanade provides the lowest average error in the 0-8 meter range. TTC and PADE are inseparable for both Farnebäck and Lucas-Kanade, in the 0-4 meter range, although between 4-12 meters the difference between the two methods grows in favour of PADE.

	Absolute Depth Error (ADE)			
	0-4m	**4-8m**	**8-12m**	**12-16m**
TTC				
CLG	0.53 ±0.17	1.99 ±0.46	2.69 ±0.36	**2.15** ±0.47
Farnebäck	0.38 ±0.20	1.84 ±0.42	2.76 ±0.46	2.36 ±0.47
Lucas-Kanade	**0.37** ±0.18	1.76 ±0.36	2.53 ±0.37	2.41 ±0.42
PADE				
CLG	0.45 ±0.20	1.57 ±0.20	**2.47**±0.33	2.98±0.44
Farnebäck	0.40±0.16	1.53±0.28	2.54±0.32	3.23±0.47
Lucas-Kanade	0.39±0.15	**1.39**±0.28	2.51±0.35	3.47±0.45

Table 1: Absolute Depth Error for each distance range.

Position Measurement Noise Analysis

To analyse how dependent each depth estimation technique is on the accuracy of relative position measurements, we add noise to the translation, rotation and linear velocity measurements of up to 20%. If we consider a camera operating at 10 fps and a quadrotor with a translational velocity of 2 meters/second, an image will be captured approximately every 20 centimetres, producing translational errors of up to 4 cm in each direction. Figure 3 shows ADE for TTC and PADE using Lucas-Kanade with noise added to position measurements, PADE (green dotted), PADE with noise (green solid), TTC (red dotted) and TTC with noise (red solid). Lucas-Kanade results are representative of all OF methods, for simplicity they are the OF method shown in Figure 3. Consistent with our previous experiments, PADE (green dotted) has the lowest ADE between 0-12 meters and TTC (red dotted) is slightly better between 12-16 meters. With the addition of noise, PADE (green solid) ADE slightly increase, although it is stable from 0-16 meters. In contrast, TTC (red solid) ADE measurements

increase by 10-30 centimetres between 0-12 meters and up to 65 centimetres between 12-16 meters. Our data suggests that PADE (green solid) is more robust to noise at 0-12 meters than TTC (red solid) by approximately 30%. TTC (red solid) is sensitive to noise because it is dependant on accurate linear velocity measurements to scale from frames into meters.

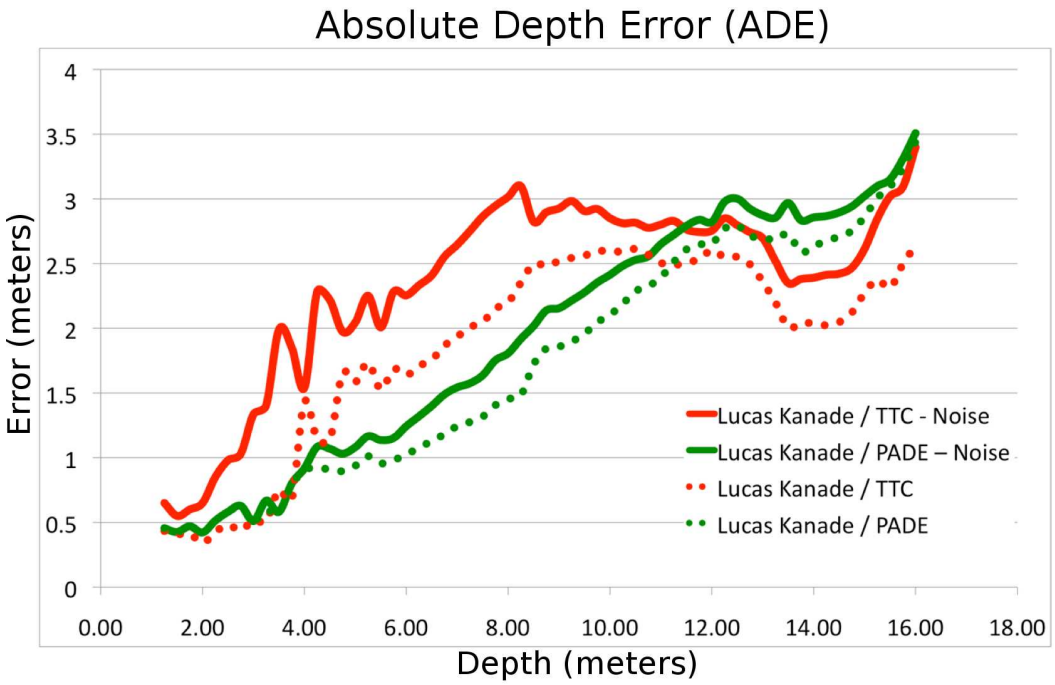


Figure 3: Absolute Depth Error - TTC and PADE with noise added.

Depth Estimation performance

To evaluate the computational speed of each step in our processing pipeline, Table 2 is divided into three main columns: OF, depth estimation and overall performance. We have evaluated the performance of each combination of OF and depth estimation. In both the OF and depth estimation sections we list the methods used and average processing time per frame in milliseconds. The final column (performance) shows the average processing time for the complete pipeline per frame. We have also included the approximate frames per second (FPS) measurement to indicate if each pipeline is appropriate for real time performance. The computational performance of OF methods vary substantially based on their implementations. Computations have been performed on a 2.5 GHz Intel Core processor executing C/C++ code.

Although Lucas-Kanade has out performed CLG and Farnebäck (see Table 1), its average processing time of 2800 milliseconds makes it inappropriate for real time performance. However, ADE measurements indicate that Farnebäck's accuracy is comparable to Lucas-Kanade in the range of 0-12 meters and has an average processing time of 100 milliseconds, 3% of Lucas-Kanade's. The fastest computational pipeline is Farnebäck combined with PADE, which achieves an average pipeline processing time of 136 milliseconds per frame or 7.35 fps.

Optical Flow		**Depth Estimation**		**Performance**	
Method	Time(ms)	Method	Time(ms)	Total(ms)	FPS
CLG	880	TTC	190	1070	0.93
		PADE	39	919	1.09
Farnebäck	**100**	TTC	178	278	3.60
		PADE	**36**	**136**	**7.35**
Lucas-Kanade	2800	TTC	181	2981	0.34
		PADE	39	2839	0.35

Table 2: Optical flow / depth estimation density and processing time comparison.

4 Applications

4.1 Relative Positioning System

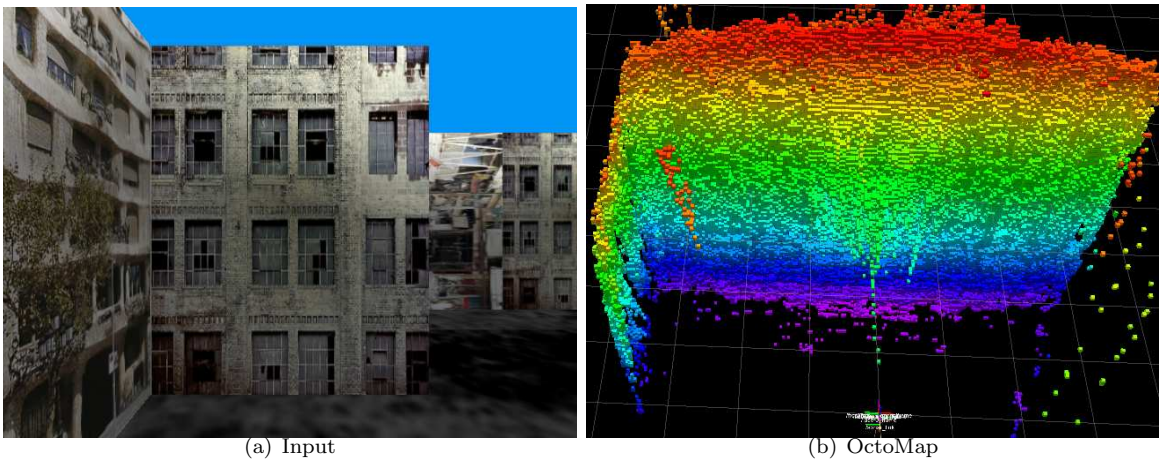
PADE requires accurate relative camera position and orientation to aid in depth estimation. Global Positioning System (GPS) currently provides position measurements with meter level accuracy. Operating on the micro scale compared to GPS there is Inertial Navigation System (INS). INS provides accurate dead reckoning, based on changes in linear and angular acceleration, although position uncertainty grows over time as errors accumulate. INS are typically implemented using an Inertial Measurement Unit (IMU) consisting of gyroscopes for angular measurements and accelerometers for linear acceleration. GPS and INS measurements are inherently noisy, although, by taking advantage of the strengths of both systems GPS and IMU signals can be fused together using an Extended Kalman Filter (EKF) to achieve greater precision [19]. GPS/INS can be combined by loose (integration at the position, velocity and/or attitude level) or tight integration (integration at the pseudorange, Doppler, or carrier phase level) [1] to correct measurements and even provide position estimates during short GPS outages.

We propose using tightly coupled GPS/INS to improve the computing the camera's relative position and orientation measurements to aid in monocular depth estimation. If the observed scene is generally static, the displacement of pixels on the image plane is due to camera translation and rotation. Two consecutive frames can be treated as a stereo pair using the translation and rotation obtained from the GPS/INS as the camera baseline. The relative position and orientation between images depends on the frame rate of the camera and the UAVs speed. If we consider a camera operating at 10 fps and a UAV with a translational velocity of 2 meters/second, an image will be captured approximately every 20 centimetres. The accuracy of GPS/INS relative measurements for small position changes, and within short time periods is in the order of centimetres.

4.2 Occupancy Grids

Although it is possible to map obstacles using dense point clouds, it is more practical to create a coarse and fast representation of the surrounding scene by using an alternative representation, like an occupancy grid. An occupancy grid is a binary map representation of obstacles in the surrounding environment. Space is evenly divided into cells and each cell is determined to be occupied or empty. The simplicity of occupancy grids have made them the dominant spatial representation used in mobile robotics [24]. 2D occupancy grids are common for ground vehicle and 3D occupancy grids for air vehicles. Taking advantage of this mapping simplification we can use point clouds projected from depth images to update a probabilistic 3D occupancy map using a framework called OctoMap [9]. The

occupancy grid when updated with a new point cloud takes into account the probabilistic uncertainty of noisy measurements, producing a low noise 3D obstacle map of the local environment. OctoMap uses a volumetric representation (voxel) with configurable resolutions. For our research we've used a voxel resolution of 10 centimetres. The occupancy grid is updated with reference to a global frame taking into account changes in the quadrotors position. The map's volumetric representation enables the quadrotor's trajectory planner to avoid collisions and ignore obstacles outside its planned path.



(a) Input (b) OctoMap

Figure 4: Image processing pipeline. Quadrotor flying towards a building.

5 Conclusions

In this work we explore monocular depth estimation techniques using stereopsis methods. We evaluate the extensively used Time To Collision (TTC) method against our novel Position Aided Depth Estimation (PADE) method and we compare both methods ability to measure distances up to 16 meters with camera translation and rotation. Our study includes an evaluation of the influence of OF methods on depth estimation and we evaluate three popular OF methods Lucas-Kanade, Farnebäck and Combined Local Global.

By adding noise to translation, rotation and velocity measurements, the difference between the two methods is more noticeable. Our study shows that TTC is sensitive to noise due to its reliance on linear velocity to scale its initial collision estimate from frames to meters. Small errors in the linear velocity estimation result in large depth errors. In contrast, the accuracy of PADE is only slightly reduced across all evaluated depth ranges, indicating that it is robust to noise.

As part of our evaluation of OF and depth estimation techniques we also evaluate the processing time for each step of the pipeline. Calculating motion using Lucas-Kanade produces the most accurate depth estimation. However, Lucas-Kanade's processing time makes it prohibitive for real time performance. Farnebäck on the other hand, provides comparable depth estimation results and requires only 3% of the time. The combination of Farnebäck with our PADE method provides a depth estimation pipeline with close to real time performance.

Our evaluation of depth estimation techniques in the context of MAV has shown that monocular depth estimation is improved when combined with relative camera position.

6 Future work

Our initial results are encouraging, although future work will involve testing, evaluation and optimisation using real flight data before deployment on a quadrotor. Currently, the majority of image pipeline processing time is consumed by OF. However, using Graphic Processing Units (GPU) have been shown to compute OF at 25 fps [15], which could provide better pipeline throughput.

7 Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 285417. This work was supported by the Spanish projects TRA2011-29454-C03-01 and TIN2011-29494-C03-02.

References

[1] Santiago Alban, D Akos, Stephen M Rock, and Demoz Gebre-Egziabher. Performance analysis and architectures for ins-aided gps tracking loops. *Institute of Navigation–NTM*, pages 611–622, 2003.

[2] Christof Born. Determining the focus of expansion by means of flow field projections. 1994.

[3] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[4] Jeffrey Byrne and Camillo J Taylor. Expansion segmentation for visual collision detection and estimation. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 875–882. IEEE, 2009.

[5] Geert De Cubber, Daniela Doroftei, Yvan Baudoin, Daniel Serrano, Keshav Chintamani, Rui Sabino, and Stephane Ourevitch. Icarus: Providing unmanned search and rescue tools. 2012.

[6] Weidong Ding, Jinling Wang, Songlai Han, Ali Almagbile, Matthew A Garratt, Andrew Lambert, and Jack Jianguo Wang. Adding optical flow into the gps/ins integration for uav navigation. In *Proc. of International Global Navigation Satellite Systems Society Symposium*, pages 1–13, 2009.

[7] Gunnar Farneback. Fast and accurate motion estimation using orientation tensors and parametric motion models. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 1, pages 135–139. IEEE, 2000.

[8] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1):185–203, 1981.

[9] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013. Software available at `http://octomap.github.com`.

[10] Albert S Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *Proc. IEEE International Symposium of Robotics Research (ISRR)*, 2011.

[11] Sung-Yeol Kim, Eun-Kyung Lee, and Yo-Sung Ho. Generation of roi enhanced depth maps using stereoscopic cameras and a depth camera. *Broadcasting, IEEE Transactions on*, 54(4):732–740, 2008.

[12] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2149–2154. IEEE, 2004.

[13] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

[14] David Marr, Tomaso Poggio, Ellen C Hildreth, and W Eric L Grimson. A computational theory of human stereo vision. In *From the Retina to the Neocortex*, pages 263–295. Springer, 1991.

[15] Paul Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, J-M Frahm, Ruigang Yang, David Nistér, and Marc Pollefeys. Real-time visibility-based fusion of depth maps. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[16] Paul C Merrell, Dah-Jye Lee, and Randal W Beard. Obstacle avoidance for unmanned air vehicles using optical flow probability distributions. In *Optics East*, pages 13–22. International Society for Optics and Photonics, 2004.

[17] Laurent Muratet, Stephane Doncieux, Yves Briere, and Jean-Arcady Meyer. A contribution to vision-based autonomous helicopter flight in urban environments. *Robotics and Autonomous Systems*, 50(4):195–209, 2005.

[18] Randal C. Nelson and Jhon Aloimonos. Obstacle avoidance using flow field divergence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(10):1102–1106, 1989.

[19] MG Petovello, D Sun, G Lachapelle, and ME Cannon. Performance analysis of an ultra-tightly integrated gps and reduced imu system. In *ION GNSS*, pages 1–8, 2007.

[20] Gian F Poggio and Tomaso Poggio. The analysis of stereopsis. *Annual review of neuroscience*, 7(1):379–412, 1984.

[21] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, 2009.

[22] Sebastian Scherer, Sanjiv Singh, Lyle Chamberlain, and Srikanth Saripalli. Flying fast and low among obstacles. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2023–2029. IEEE, 2007.

[23] Kahlouche Souhila and Achour Karim. Optical flow based robot obstacle avoidance. *International Journal of Advanced Robotic Systems*, 4(1):13–16, 2007.

[24] Sebastian Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous robots*, 15(2):111–127, 2003.

[25] Michael Waschbüsch, Stephan Würmlin, Daniel Cotting, and Markus Gross. Point-sampled 3d video of real-world scenes. *Signal Processing: Image Communication*, 22(2):203–216, 2007.

[26] Dong-Wan Yoo, Dae-Yeon Won, and Min-Jea Tahk. Optical flow based collision avoidance of multi-rotor uavs in urban environments. *International Journal of Aeronautical and Space Sciences*, 12:252–259, 2011.

[27] Huili Yu, Randy Beard, and Jeffrey Byrne. Vision-based navigation frame mapping and planning for collision avoidance for miniature air vehicles. *Control Engineering Practice*, 18(7):824–836, 2010.

[28] Simon Zingg, Davide Scaramuzza, Stephan Weiss, and Roland Siegwart. Mav navigation through indoor corridors using optical flow. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3361–3368. IEEE, 2010.