I²MTC 2008 – IEEE International Instrumentation and
Measurement Technology Conference
Victoria, Vancouver Island, Canada, May 12–15, 2008

Obstacle Detection for Low Flying Unmanned Aerial Vehicles Using Stereoscopic Imaging

E.Hanna[1], P.Straznicky[2], R.Goubran[2]
[1]Graduate Student, Department of Systems and Computer Engineering,
Carleton University, Canada. ehanna@sce.carleton.ca
[2]Professor, Department of Mechanical and Aerospace Engineering,
Carleton University, Canada. pstrazni@mae.carleton.ca
[3]Professor and Dean of Engineering, Carleton University, Canada.
SMIEEE. goubran@sce.carleton.ca

Abstract* – *This paper describes a stereoscopic imaging algorithm that is modified for obstacle detection in low flying Unmanned Aerial Vehicles (UAVs). In this type of flight, obstacle detection must be carried out quickly for the system to be effective in real time. Additionally, since the aircraft is close to the ground, the horizon is usually at the top of the field of view and obstacles must be distinguished from the clutter of the terrain. The sparse edge detection and reconstruction algorithm proposed, produces fast but partial reconstructions of the environment. One image is passed through a series of edge detectors to generate a very sparse outline of the environment. This outline is then correlated to the second image and the resulting reconstruction is added to a model of the environment. Although each individual reconstruction is incomplete, the overall result after a short initialization period is a model of the environment that is more comprehensive than a single stereoscopic correlation run with a more detailed edge detector. Simulation of the algorithm on test image patterns showed an increase in performance relative to the length of the sequence of stereo pairs. On average, the signal to noise ratio (SNR) for sparse edge reconstruction was significantly higher than that for single correlation with more detailed edge detectors. Additionally it was found that the processing speed of the sparse edge detection algorithm on a pair of stereoscopic images is faster than the processing carried out by a more detailed edge detector. A test flight was also carried out to test the algorithm in a more realistic scenario. The test confirmed that the sparse edge reconstruction algorithm resulted in a much more detailed view of the environment than if a single, more detailed edge detector had been used.

Keywords – *Obstacle Detection and Avoidance, Distance Measurement, Stereoscopic Imaging, Unmanned Aerial Vehicles (UAV), Real-time Processing, Computer Vision*

I. INTRODUCTION

The measurement of distance by way of stereoscopic analysis is limited by computational speed in real time scenarios [1]. An obstacle detection and avoidance system that uses a conventional processor and employs stereoscopy should, therefore, be streamlined for fast analysis. This is especially true in the case of a low altitude UAV where the system cannot afford to create a dense stereo disparity map from every pair of images that is captured.

A method that has been suggested to speed up and enhance the accuracy of the obstacle detection system is to first detect the horizon line in the images [2]. In the special case of low flying UAV's (10 to 60 m), the horizon is often at the very top of the field of view and is, at times, not even visible. This means that almost all of the stereoscopic analysis is performed on areas that are optically cluttered and the problem no longer lies in identifying areas of the image that are not 'sky' [2], but rather finding objects in the clutter that are not 'ground'. This paper addresses this real-time application problem.

There are many different methods for object localization [3], and ultimately the goal of the methods that are presented here is to effectively use stereoscopic imaging methods for obstacle localization. However, attempting to carry out localization with a large amount of information will result in unwanted computational expenditures. The proposed method reduces the quantity of the data from the stereoscopic cameras significantly while preserving the overall performance of the system. An algorithm is presented to quickly generate incomplete but accurate disparity maps that assume no horizon. By combining these incomplete maps over time, a comprehensive reconstruction of the environment can be accessible during the UAV's flight. In this way, it is not necessary to reconstruct the environment every time new images are acquired, but rather to add meaningful information to the system's model of it [4], [5]. The methods presented were simulated on test images and were also evaluated using data from stereoscopic cameras that were mounted on an aircraft during a test flight.

II. SPARSE EDGE RECONSTRUCTION

The proposed approach to carrying out obstacle detection onboard a low-flying UAV is to create a model of the environment over time by reading incomplete pieces of the environment sequentially. After a short initialization period, the system would have a relatively detailed description of the environment that is constantly changing as new meaningful data is added. An algorithm, (hereafter referred to as sparse edge reconstruction) is used to carry out a stereoscopic matching analysis customized for this type of quick partial analysis. An overview of this algorithm is presented in Figure 1. The procedure is heavily dependent on the idea that obstacles will have noticeable contrast with their surroundings. In many cases this may not be true for a single frame, however over the course of the UAV's flight the obstacles' contrast will invariably become defined due to change in camera angles as well as aircraft and object

location. This assumption was confirmed during the test flight that was carried out and is described in more detail in section IV.

Each individual run of the sparse edge reconstruction algorithm is asymmetric since only one of the reference images is correlated to the other (example: left image to right image). This can be repeated with the other image as the reference at the expense of doubling the computation time. A better solution is to alternate the reference camera over time. This has the advantage of increasing the accuracy of the correlation as well as removing systematic errors such as lens obstructions or lighting artifacts.

The first step in the sparse edge reconstruction is to select a reference image and apply an edge detection function to it. A detailed edge map will require more computation to generate and will also slow down the correlation process as will be shown in the next section. To avoid these problems, three linear edge detectors are combined with a bitwise AND operator to give a less dense but more accurate edge detector. Before the combination, each detector is filtered to remove edge pixels that have no neighbouring edge pixels. This is done to attenuate texture elements that have high contrast. The overall compounded edge detector is usually incomplete, sparse and accurate.

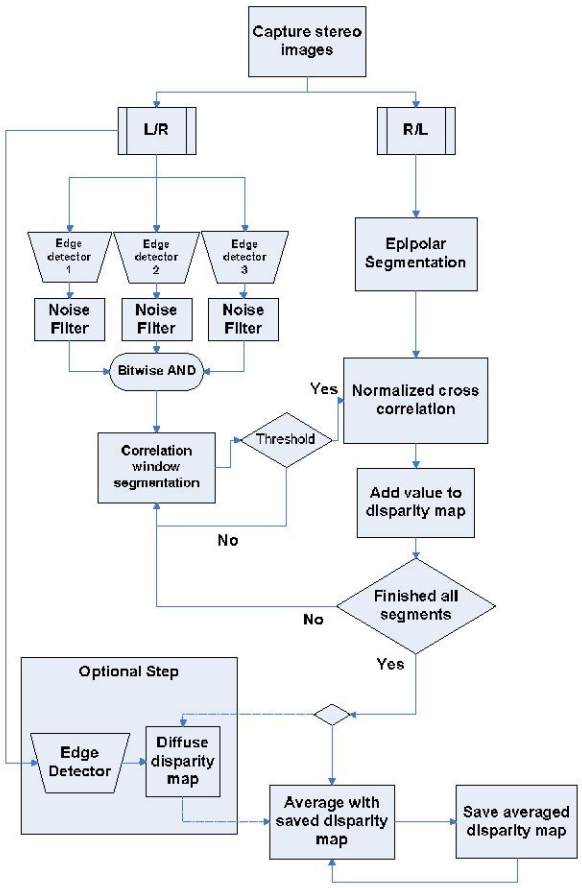

Figure 1: Sparse edge reconstruction algorithm

The resulting image is then segmented according to the size of the correlation window that will be used. The value is usually 7x7 to 9x9 pixels (comprehensive analysis) or 15x15 to 29x29 pixels (fast analysis) [6]. To further increase the speed of the system, only window segments containing a threshold number of edge pixels are correlated to the second image. As a result, the segments that are analysed are the ones containing the most contrast. Although this can be problematic in a single run, the combined reconstructions over time will compensate for this, as will be shown in the following sections. The correspondence itself is carried out using normalized cross correlation as described in [7] and summarized in the equation:

$$\gamma(u,v)=\frac{\sum_{x,y}\left[f(x,y)-\bar{f}_{u,v}\right]*\left[t(x-u,y-v)-\bar{t}\right]}{\left\{\sum_{x,y}\left[f(x,y)-\bar{f}_{u,v}\right]^2*\sum_{x,y}\left[t(x-u,y-v)-\bar{t}\right]^2\right\}^{0.5}} \quad (1)$$

Where f(x,y) is the image to which the segment t is correlated at the position (u,v). $\bar{f}_{u,y}$ is the mean of the image in the region under the window segment and $\bar{t}$ is the mean of the segment itself. Each segment that is to be correlated is evaluated at all the (u,v) positions along the epipolar line in which the segment is found [8]. In the case of parallel cameras, the epipolar lines are assumed to be horizontal.

A maximum disparity value is also set. This has the advantage of both increasing accuracy and reducing computation [8]. By correlating all the selected segments over the second image, a disparity map is generated which contains values at the areas that have been determined to be edges. All other areas are assigned a default value of zero and are interpreted as undefined distances.

An optional step can be included in the case of a reconstruction that is still relatively dense. The data can be enhanced by diffusing the disparity values over the map until an edge is encountered. In this step the edges used are not from the sparse edge map determined in the beginning of the process but rather those defined by a more complete edge detector. The purpose of this step is to make up for the increased processing time required in creating and correlating a dense map by having the step provide a higher level of information about the environment. This step is essentially an attempt to capture some of the surfaces of the objects being observed [4].

The process described above is carried out every time new images are acquired or at a predefined interval. Over time, the enhanced disparity maps are combined to filter out errors and to provide a more complete description of the obstacles being observed. To enhance this iteration process, it might be preferable to correlate the sequences of intensity values over time instead of averaging the disparity values themselves. This process is described in [10]

III. SIMULATION RESULTS

The results presented here, were generated by a MATLAB implementation of the sparse edge reconstruction algorithm. Figure 1 shows a typical image pattern that was used in simulation. Note that the textured background is a source of noise and error and is used to simulate terrain information that might be present in a low altitude UAV flight. Additionally, some of the objects of interest have low contrast with this background which may also be true in a real scenario. The baseline separation of the cameras is about 10 cm. To simulate slight vibrations and lighting changes, the cameras were placed on a running desktop computer during the simulation. A series of tests were carried out to evaluate the performance of the sparse edge reconstruction algorithm. To compare the results, the same images were put through a single edge detector and correlated using single runs of stereo matching with the same normalized cross correlation function.

In this implementation of the sparse edge detection algorithm three edge detectors are combined (*Roberts, Sobel, Prewitt [11]*). Figures 3 and 4 show the difference between a single *Prewitt* edge detector and the edge detector that results in the combination. Both of these detectors have also been filtered to remove noise (isolated edge pixels). Note that the *Prewitt* edge detector contains much more visual noise (due to the background texture) than the sparse edge detector. The threshold value for the number of edge pixels per correlation window was set to 1 such that any window containing edge pixels is correlated. This was done since these edge maps are quite sparse in general.

The correlation windows in these simulations ranged from 3x3 to 5x5 pixels. Although this is smaller than usual in stereoscopic correlation [6], they were chosen since the images are relatively small at 240x320 pixels which makes the reconstructions very coarse at larger correlation window sizes.

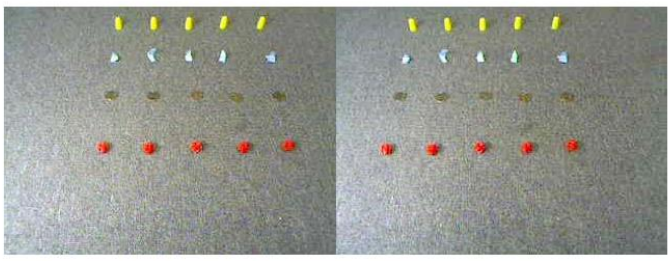

Figure 2: Stereo pair of images used in simulation

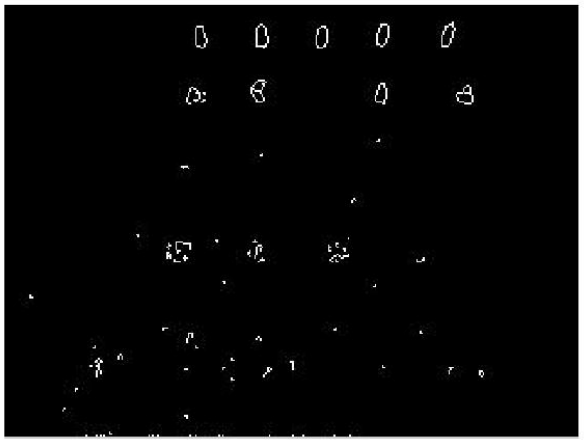

Figure 3: *Prewitt* edge detector

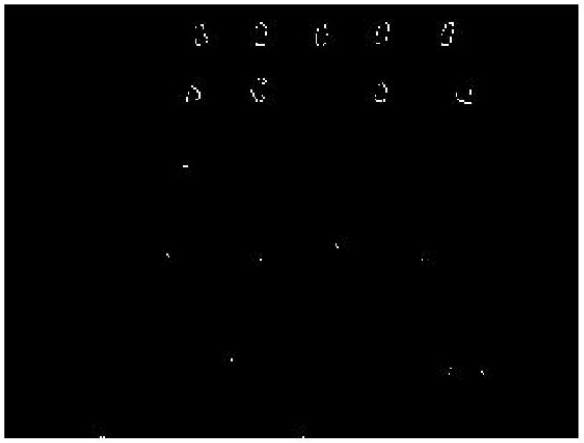

Figure 4: Sparse edge detector

Figure 5 shows a typical reconstruction from a single run of stereoscopic matching using a *Prewitt* edge detector and a noise filter. Because all the objects are at almost the same distance to the cameras, the values in the disparity map are fairly uniform. Note that the indicators where added manually to show real objects versus artifacts of the background or noise. In this case, for instance, 15 out of 20 objects where identified and there were roughly 8 error objects identified. Since the evaluation of accuracy is fairly subjective in these images, the signal to noise ratio (SNR) was estimated using

$$SNR = 20\log\left(\frac{S}{N_1 + N_2}\right) \quad (1)$$

Where S is the number of correctly identified objects, N_1 is the number of existing objects that have not been identified and N_2 is the number of incorrectly identified objects. In Figure 5, S = 15, N_1 = 5, N2 = 8 and the SNR = 1.243dB

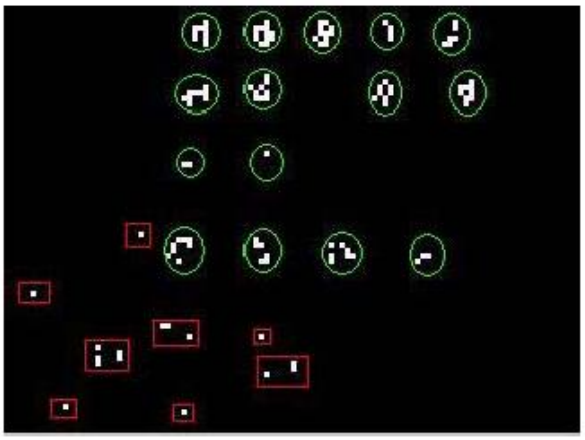

Figure 5: Reconstruction using *Prewitt* detector

Figure 6 shows a typical reconstruction using the sparse edge reconstruction algorithm after a run of 10 pairs of images. Note that typically, these reconstruction have much less noise and an equal or slightly smaller number of correctly identified objects.

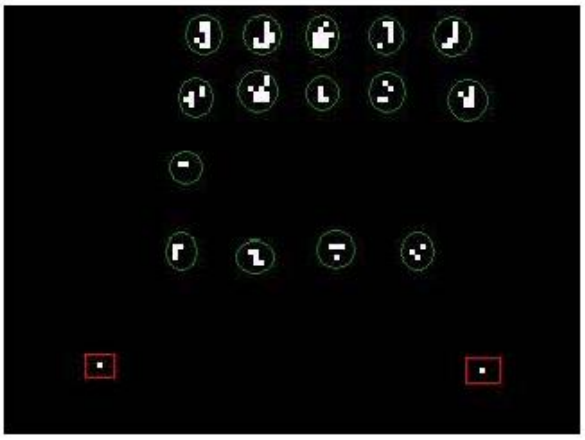

Figure 6: Reconstruction using sparse edge detector

Tables 1 and 2 show the results of a number of simulations using image patterns similar to Figure 2. On average, the SNR is significantly higher in the sparse edge reconstruction algorithm than for single runs of correlation with each individual edge detector. The results for the sparse edge reconstruction algorithm have an SNR dependant on the length of the sequence as shown in Figure 7. In a more realistic UAV scenario, the lengths of such sequences would be limited by the amount of time that the obstacles could be assumed to be static.

Table 1: SNR estimates for sparse edge reconstruction algorithm

Length of sequence	**Average SNR over 5 trials per sequence (dB)**
5	6.16
10	6.74
20	9.06

Table 2: SNR estimates for reconstructions using single edge detectors and noise removal

Edge detector	**Average SNR over 20 trials (dB)**
Prewitt	4.06
Sobel	1.87
Roberts	4.27

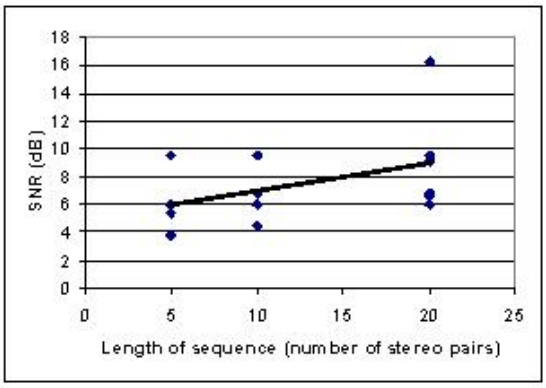

Figure 7: SNR vs. sequence length for sparse edge reconstruction algorithm

Table 3 shows the time required for the two types of reconstructions. This includes the time necessary for the creation of any edge detector as well as the normalized cross correlation and the creation of the disparity map. It does not, however, include the time required for the optional step in Figure 1 and described in section II. It is important to clarify that the SNR of the sparse edge reconstruction is dependant on the length of a sequence and although Table 3 only presents the time required for each pair of images in the sequence, the speed increase is meaningful in the larger context of the algorithm. After an initialization period, the model of the environment exists in a constant state of flux and the SNR at any given moment is enhanced by the system's memory of the sequence while maintaining the computation times listed below. It is clear that sparse edge detection is significantly faster than applying an individual filter before correlation.

Table 3: Speed of reconstruction per pair of images (2.4 GHz Intel Pentium 4 processor using a 3x3 pixels correlation window and a maximum disparity of 36 pixels.

Type of reconstruction	**Time required per pair of frames (sec)**	**Speed increase with sparse edge reconstruction (%)**
Sparse edge reconstruction	0.659	N/A
Prewitt edge detector	1.1093	40.59
Sobel edge detector	1.076	38.75
Roberts edge detector	0.8938	26.27

Section II described an optional step of diffusing the disparity maps to the nearest edge determined by a dense edge detector. Figure 8 shows a disparity map created with sparse edge reconstruction before and after this step. In this case, a *Canny* edge detector [9] was used as the border of the

diffusion. Note that the new disparity map shows a more accurate representation of the environment. The objects in the image that are correct are amplified and the noise elements remain the same. The purpose of this step is to make up for the time spent analyzing the relatively dense image by having the reconstruction provide additional information to be saved in the model.

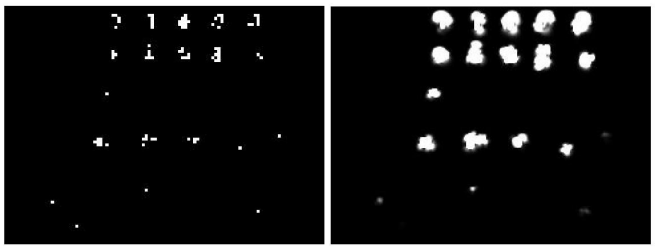

Figure 8: Sparse edge reconstruction before and after diffusion

IV. EXPERIMENTAL RESULTS

In order to test the sparse edge reconstruction algorithm in a realistic scenario as well as to determine the types of conditions that would be encountered by a low altitude UAV, a test flight was carried out with a Cessna 404 aircraft. The flight included flying relatively low altitude test lines (60-100m). Two ELMO model TSN411D colour digital CCD cameras, each equipped with a Honeywell lens: HLD5V50F13L, 5-50MM/F1.3, 1/3" CS Mount were attached to an inboard wing panel of the aircraft with a downward angle of 15º (Figures 11 and 12). After the flight, it was determined that the right camera was out of focus and so the stereo reconstruction is not optimal. The cameras recorded video over the length of the flight. Several sparse edge reconstruction tests have been performed with various sequences in the recorded video and an example is presented here.

Figure 11: Stereo camera setup

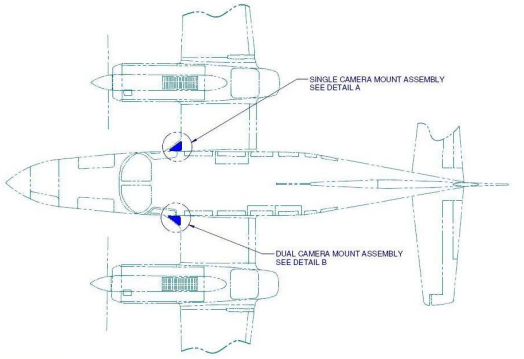

Figure 12: Camera position during test flight

The video has confirmed that for low flying aircraft, the horizon line is very high up in the field of view and that it can be very poorly defined due to haze or low resolution of the images. In this example a sequence of 10 stereo frames is analysed and the results are combined with each frame given equal weight in the reconstruction. In this analysis it was assumed that the obstacles are stationary with respect to the cameras. This is an incorrect assumption since both the cameras and the objects may be moving but it can be acceptable if the sequence of images is short enough. However, this simplification can only hold for objects that are at relatively large distances from the plane. As a result of this assumption, the position of the objects being observed cannot be exactly defined in the reconstruction.

Figure 13: First image in 10-frame sequence (right camera)

Figure 14: Last image in 10-frame sequence (right camera)

Figure 13 and 14 show the first and last frame of the sequence as viewed from the right camera. The cluttered

terrain with poorly defined obstacles is typical for the flight. It is also apparent that some obstacles lack contrast with their surroundings and may be missed in the reconstruction.

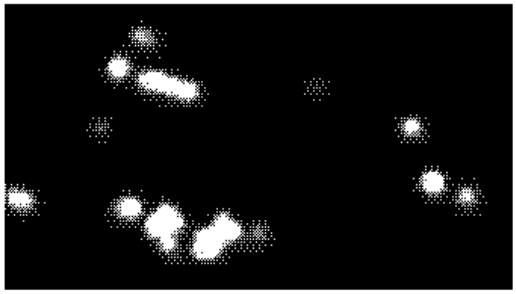

Figure 15: Prewitt edge stereo reconstruction of a single pair of images from the sequence in Figure 12 and 13

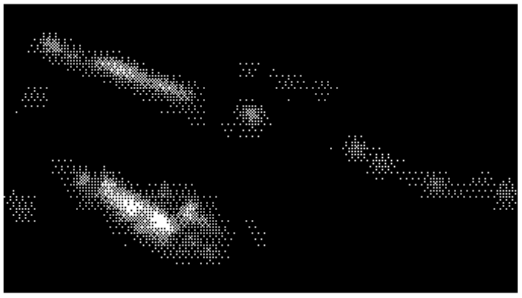

Figure 16: Sparse edge reconstruction over the entire sequence described by figure 12 and 13

Figure 15 shows the *Prewitt* edge reconstruction of a single pair of frames from the sequence and Figure 16 shows the sparse edge reconstruction over the entire sequence. In these images, the brighter areas represent objects that are closer to the cameras. Note that the second reconstruction picks out features that may have been missed or filtered out in some of the individual runs and results in a more complete and detailed representation of the environment. For instance although the average individual run easily picks out objects of high contrast, such as the houses in the lower left, a series of runs is necessary to model lower contrast objects (streets or prominent tree clusters).

V. CONCLUSION

Sparse edge reconstruction was presented as an algorithm for stereoscopic obstacle detection that is tailored for low-flying UAVs. The purpose is to accurately model the environment with a low level of computational intensity. The sparse algorithm takes incomplete snapshots sequentially such that a constant flux of information is given to the system. The superimposition of the incomplete data and the constant updating ofthe system's model of the environment gives a comprehensive representation of the obstacles present. It was shown that the sparse edge reconstruction algorithm has a higher SNR than if stereoscopic correlation is carried out once after applying a single edge detector. Additionally it was shown that the increase in the speed of the system is substantial. It is important to note that the analysis of a sequence of n frames (in sparse edge reconstruction) takes n times the computing power of a single run however the key is that each run is done sequentially in real-time. The result is a detailed model of the environment in constant flux. After a certain number of initial frames, the system has access to a detailed model of the environment even though the data analysed in any given run is incomplete. If a higher level algorithm can be developed to track the motion of the obstacles, this model could represent a comprehensive view of the environment.

VI. ACKNOWLEDGEMENT

The authors would like to thank Sander Geophysics Ltd. for providing the test flight and video data as well as the Ontario Centres of Excellence for research funding.

VII. REFERENCES

[1] J. Kaszubiak, M. Tornow, R.W. Kuhn, B. Michaelis. "Real-time 2-D-multi object position estimation and tracking," in the *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, Aug. 2004 Page(s):785 - 788 Vol.1 IEEE Reference

[2] T.G. McGee, R. Sengupta, K. Hedrick. "Obstacle Detection for Small Autonomous Aircraft Using Sky Segmentation" in the *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 2005 Page(s):4679 – 4684

[3] D. Lo, R.A. Goubran, R. M. Danserau, G. Thompson, D. Schultz. "Robust Joint Audio-Visual Localization in Video Conferencing Using Reliability Information" *IEEE Transactions on Instrumentation and Measurement*, Volume 53, Issue 4, Aug. 2004 Page(s):1132 – 1139

[4] C. Chen, P. Payeur. "Scan-Based Registration of Range Measurments" in *Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference, 2002. IMTC/2002*, 2002 page(s): 19 - 24 vol.1

[5] M. Marron, J.C. Garcia, M.A. Sotelo, D. P. Perez, I.B. Munoz. "Extraction of 3D Features from Complex Environments in Visual Tracking Applications" in *The IEEE Instrumentation and Measurement Technology Conference Proceedings*, May 2007 Page(s):1 – 5

[6] K.W. Kim, A.I. Ansar, R.D. Steele, R.C. Steinke. "Performance analysis and validation of a stereo vision system," in the *Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics*, Oct. 2005 Page(s):1409 - 1416 Vol. 2

[7] D. I. Barnea, H. F. Silverman, ``A dass of algorithms for fast digital image registration", *IEEE Trans. Computers*, 21, pp. 179-186, 1972.

[8] P. Moallem, K. Faez, L. Haddadnia. "Reduction of the search space region in the edge based stereo correspondence," in the *Proceedings of the 2001 International Conference on Image Processing*, Oct. 2001 Page(s):149 - 152 vol.2.

[9] J. Canny, "A Computational Approach to Edge Detection", IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

[10] P. Albrecht, B. Michaelis, "Improvement of the Spatial Resolution of an Optical 3-D Measurment Procedure", *IEEE Transactions on Instrumentation and Measurement*, Volume 47, Issue 1, Feb. 1998 Page(s):158 – 162

[11] S. E. Umbaugh, "Computer Imaging Digital Image Analysis and Processing", CRC,2005.