IEEE Africon 2011 - The Falls Resort and Conference Centre, Livingstone, Zambia, 13 - 15 September 2011

A Critique on Previous Work in Vision Aided Navigation

Charles Murcott
UAV Research Group, Robotics Division
School of Electrical Engineering
University of Johannesburg
South Africa
Email: charles.murcott@gmail.com

Francois Du Plessis
Senior Lecturer
School of Electrical Engineering
University of Johannesburg
South Africa
Email: francoisdp@uj.ac.za

Johan Meyer
Head of UAV Research Group
School of Electrical Engineering
University of Johannesburg
South Africa
Email: johanm@uj.ac.za

***Abstract*—This paper presents a critique on previous work in the field of vision aided navigation, particularly in the fusion of visual and inertial sensors for navigation. Several improvements and updates are proposed for the existent systems. GPS receivers have allowed for accurate navigation for many vehicles and robotic platforms. GPS based navigation can, however, prove to be impractical in applications where there is no GPS reception such as underground, indoors or in some urban areas. This pertains, in particular, to many robotic applications where position must be known in global coordinates or relative to a reference point. An inertial navigation system (INS) can be used to calculate one's relative navigation state via dead-reckoning calculations. The downfall of a low-cost INS is the errors associated with the system. While these errors are initially small, integration causes large drift errors over time. To combat this problem, cameras can be used to estimate the errors present in the INS readings. These results can then be used to correct the navigation state output from the INS. While the motion estimations from the cameras are not error-free, this method is made highly effective because of the complementary nature of the errors from the cameras and INS. Several improvements are proposed for this method; algorithmically, in updates to its hardware, and with the introduction of graphics processors to improve computational performance. The overall system performance, individual steps, algorithms, and results are compared to results from similar works to those of the proposed improvements. It is shown that the accuracy, responsiveness and overall performance of the system can potentially be greatly improved.**

I. Introduction

Navigation is defined as the determination of the state of a vehicle (its position, velocity and attitude) relative to its original state. The problem addressed in this paper is the determination of a vehicle's trajectory (its navigation state through time) by the use of inertial and imaging sensors. The ability for vehicles to accurately navigate is required in many applications including military, industrial and commercial; especially those in which a vehicle or robotic platform must operate autonomously. The main, and often simplest, answer to this problem is to use GPS (Global Positioning System) sensors to navigate. GPS aided navigation presents a problem in situations where there is no available signal, such as in some urban or mountainous environments, underground (mining environments), indoors or underwater [1]. At first glance an INS (Inertial Navigation System) offers a promising solution to the problem of GPS-free navigation. INSs use accelerometers and gyroscopic sensors to determine the acceleration and rotation of the system. A strapdown INS system is attached to an object (the vehicle, in this case) allowing these readings to be used to determine the relative position, velocity and orientation of the object. This solution's downfall is that these low-cost sensors are prone to errors, which accumulate and cause the estimated position and velocity to drift from the object's actual position and velocity, making the sensors effectively unusable without GPS assistance. This problem is especially bad in consumer-grade sensors, but is still present in high-grade sensors [1].

Cameras have been used increasingly often to solve the navigation problem via methods such as visual odometry [2] and visual SLAM (Simultaneous Localisation And Mapping) [3]. By using image processing techniques on subsequent frames, one can track the motion of the cameras relative to the environment. Using cameras as primary sensors for navigation can be extremely robust, depending on the methods used. The main difficulty in this approach is the computational power required to perform these algorithms in realtime. In general, the computational requirements of the system increase as the robustness and accuracy of the algorithms increase.

This paper takes into account some established methods in vision aided navigation, identifying shortcomings and areas that can potentially be improved or updated. A new system, based upon the others that are reviewed, is then proposed, focusing on the identified points. It should also be noted that this paper presents a conceptual solution, and no physical results for the system are given. An implementation is proposed, and results from various other papers are cited to validate the proposed improvements.

II. Previous Work

The two main approaches used in vision aided navigation are based on visual odometry and visual SLAM, both of which can be supplemented by readings from an INS. Established systems based on both of these approaches will be discussed.

The first method discussed for vision aided navigation based on the visual odometry approach, is that of Veth and Raquet [1], which uses the readings from a stereoscopic camera pair to estimate the errors present in the INS readings. This is done

978-1-61284-993-5/11/$26.00 ©2011 IEEE

by first capturing images using stereoscopic cameras, then extracting features from one image in the stereo pair using the SIFT (Scale Invariant Feature Transform) algorithm [4]. These features refer to points on the image, identified by the SIFT algorithm, which can be easily tracked in subsequent frames. Using a SFM (Structure From Motion) approach, the selected features are located in the other image in the stereo pair and the absolute depths of the feature points are determined. The inclusion of the depth measurement allows for increased accuracy in the measurement of the navigation state. Integrating measurements from the INS then produces a measured update of the navigation state. This measurement is used to estimate the location of the previously identified features in the next image. Using an error model of the INS with the inertial measurements allows one to determine a window in which each feature will be found. Statistical feature matching is performed between newly extracted features and those from the previous frame. Errors in the INS readings are then estimated based on the disparities between the estimated and measured feature locations using an EKF (Extended Kalman Filter). These error estimations are then subtracted from the INS readings, providing a corrected navigation state. An overview of the system can be seen in Figure 1.

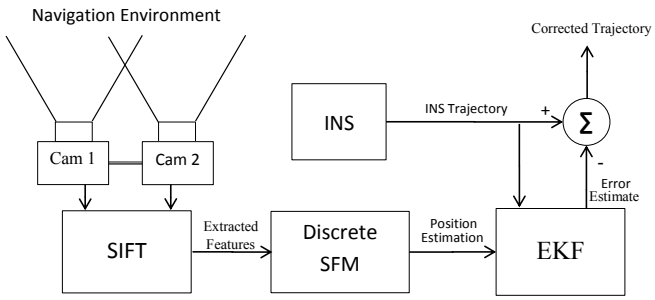

Fig. 1. Veth's System

The work of Veth and Raquet has similar goals, and has produced similar results to several related works [5], [6], [7]. Other works are largely similar in their methodology, therefore the work of Konolige et al. [5] will be briefly explained as an example. In this work features are first extracted from one image of a stereo pair. These features are given depth by locating them in the corresponding stereo image. Features are then extracted from the subsequent frame and matched with the features from the previous frame. The features are used to estimate the relative navigation state using RANSAC (RANdom SAmpling and Consensus) [8], an algorithm designed to find the best solution with respect to a large amount of potentially noisy and/or erroneous measurements. A bundle adjustment is performed using features from a number of recent frames in order to improve the accuracy of the estimate [9]. This estimated navigation state is then fused with the readings from the INS.

The method of Veth and Raquet differs from many other works in its fusion method, which allows for a deeper level of integration between the visual and inertial sensors. This deeper level of integration allows the synergistic properties of the two sensors to be better exploited, providing more accurate results. Another quality of Veth and Raquet's work, when compared to other work in this field, is that a more complete model of the INS is utilised [5]. By doing this, the INS readings can be used to their full potential in the fusion process.

A well established method associated with the visual SLAM based approach, is that of Davison et al. [3]. His method creates a sparse persistent map of landmarks extracted from the environment. Tracking these landmarks allows for the determination of the cameras trajectory. Landmarks are extracted and stored in a probabilistic 3D map. The visible map points are identified in each frame, and the position of the camera in the 3D map is calculated from the relative position of these points. The computation times involved in maintaining the map grows in order $O(n^2)$, where n is the number of landmarks in the map; because of this, landmarks must be actively maintained. This maintenance involves: keeping a maximum of 100 landmarks in the map at a given point in time, deleting landmarks that are not repeatably matched, and keeping a minimum number of landmarks in the camera's frame. As a result of the landmark maintenance, the system excels in eliminating drift in areas where previous landmarks are frequently observed by means of small loop closures. The system is at a disadvantage in large loops, where old landmarks may have been deleted; or in applications where scenes are unique, and landmarks are not often (if ever) repeated once they leave the camera's frame. These circumstances can arise in the navigation of an unknown, unconstrained, environment; therefore this method is not directly suited to the navigation system proposed in this paper. Other methods have, however, branched off from the visual SLAM based approach which are plausible in a large scale environment. For example, systems have been created that borrow from both the visual odometry and visual SLAM based approaches, which show promise [10], [11].

The work of Veth and Raquet (Figure 1) has been chosen as a basis for the proposed system due to its success in vision aided inertial navigation. Taking into account the previously discussed works, the following areas have been identified for potential improvement.

A. Image Capture

The method used by Veth and Raquet utilised two Pixelink PL-A741 Machine Vision Cameras [12] arranged to produce a stereoscopic rig. While this allows for an adjustment in the binocular disparity of the system, it also creates the opportunity for human error in measurement and alignment of the cameras. This means that the cameras must be calibrated specially for the system. Both of these points can lead to inaccuracies in measurements during the feature extraction phase. The use of stereoscopic cameras, as opposed to a single camera, allows for more reliable methods to determine depth, and can increase the accuracy of the motion estimates.

978-1-61284-993-5/11/$26.00 ©2011 IEEE

B. Processing Power and Image Processing Techniques

The overall accuracy of the method used by Veth and Raquet was severely impacted by its final operating speed of 2.5Hz (only 2.5 video frames could be processed by its navigation system per second). This was largely due to the high processing requirements of the image processing steps. All calculations were performed on the CPU, which proved to be sub-optimal for the application. Some of the previously mentioned methods improved on this operating speed, reaching true realtime capabilities. Some techniques for incorporating the other method's improvements in speed will be elaborated upon in Section IV.

C. Fusion Algorithms/Estimation Algorithms

The EKF (Extended Kalman Filter) was used to estimate the errors present in the INS in the method used by Veth and Raquet. It is used very often in visual SLAM and visual odometry based systems. This Kalman Filter variant provides a linear-approximation of the system dynamics for non-linear systems by using Jacobians to linearise the nonlinear system model; the approximation does not provide an accurate estimate for systems with more than first order nonlinearities. Its main drawbacks are its complex derivation, and the errors it introduces due to linearisation.

Civera et al. [10] used a sensor-centred EKF model in order to minimise on the linearisation errors caused by the EKF. This does improve on the filter's shortcomings, however the results were still sub-optimal.

III. The GPU

The GPU (Graphics Processing Unit) is a stream processor, designed to perform calculations on large amounts of data in parallel. GPUs were originally designed for computer graphics applications in which an array of pixels was processed (each in parallel), and output to another array of pixels. GPUs have a number of shaders, each capable of processing the same computation at the same time; these shaders are responsible for the parallel nature of GPU computations. GPGPU (General Purpose computations on Graphics Processing Units) is a technique by which the GPU is used to perform calculations which would normally be processed on the CPU. This allows the same computation to be done many times in parallel, greatly increasing the throughput of the system.

Implementing an algorithm on the GPU requires it to be parallelisable. This means that the algorithm, or parts thereof, has calculations that are repeated many times in series, but have independent outputs and identical operations. In other words, one calculation must not depend on the output of another, and each must follow the same formula (for example $Z = X + Y$).

GPGPU computations can, therefore, be performed in order to decrease the time required to perform computationally expensive steps in the navigation system. To compare GPU and CPU based systems in their potential speed, one can refer to the number of FLOPS (Floating Point Operations Per Second) they are capable of computing. Figure 2 shows the change in computing power of the CPU and GPU over time, and illustrates the great potential for current and future GPU-based computational systems.

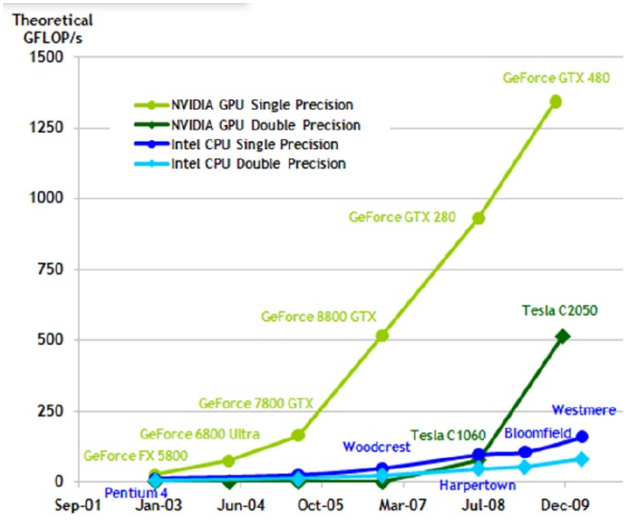

Fig. 2. GPU vs CPU Performance [13]

Image processing problems occupy a middle ground with respect to traditionally CPU-implemented algorithms and computer graphics processing, the GPU's original function. As a result, image processing techniques are particularly suited towards GPGPU implementations (e.g. [14]) and even share the same input and output data structures (a two-dimensional array of pixels). GPU implementations of image processing techniques have the potential to experience great increases in computational efficiency, as a pixel can be processed on each GPU shader in parallel, opposed to each pixel being processed individually in series on the CPU. While GPGPU implementations do come with many advantages, using the GPU as a computational platform can have disadvantages. Firstly, implementation becomes more difficult, as a different methodology must be applied when modifying an algorithm for GPGPU implementation. Secondly, the GPU can only use its own private memory, resulting in the need to copy a texture to the GPU's memory, process it, then transfer it back to CPU-readable memory for further processing. This creates additional overhead when processing on the GPU, and means that one must also consider the size and frequency of these transfers when parallelising an algorithm, as the overhead may be more computationally expensive than the algorithm itself.

IV. Proposed Methods for Improvement

As previously mentioned in Section II, a low cost INS is subject to significant measurement errors. While other methods have had success in correcting these errors, as well as in the general domain of vision aided navigation, certain areas have been identified which can be improved upon. The following subsections list these proposed improvements.

978-1-61284-993-5/11/$26.00 ©2011 IEEE

A. Image Capture

In order to improve upon the existing stereoscopic camera system, the use of a pre-calibrated stereoscopic camera system is proposed. The suggested system is Point Grey's BumbleBee XB3 stereo vision camera [15]. The cameras are pre-calibrated to a high level of accuracy, meaning that no manual calibrations need to be performed. Libraries designed specially for the cameras are also included in the package, simplifying several software-based tasks when utilising the cameras. With regards to Veth and Raquet's work, this will address difficulties in the construction and calibration of the stereo camera system; and will potentially decrease the measurement noise from the cameras, increasing overall accuracy.

B. Image Processing Techniques

For simplicity, the feature descriptors discussed here will be limited to that of SIFT, SURF and CenSurE; all of which have proven themselves in the image processing domain.

SIFT provides a scale- and rotation-invariant method for extracting features. It provides a very robust descriptor, but at a high computational cost. In general SURF (Speeded Up Robust Features) achieves slightly diminished performance when compared to SIFT (in feature matching and invariance), but it performs much faster [16], [17]. CenSurE, while less established than the other feature descriptors, provides excellent results, especially in the field of visual odometry. CenSurE produces similar results to those of SIFT and SURF, with the cost of some rotational invariance, and is faster to compute.

While the SURF and CenSurE algorithms lessen the computational load, they would still consume a considerable portion of the navigation system's computational time. As a result, the use of a GPGPU techniques are proposed to increase the system's throughput. By implementing the feature extraction algorithms on the GPU, the system can experience a significant increase in speed. Both the SIFT and SURF algorithms have been implemented on the GPU, achieving speeds of 15-20 Hz and 20-40 Hz respectively. These results pertain to the feature extraction algorithms running on an nVidia 8800GTX graphics card, at a resolution of approximately 1024×768 [18], [19] (the default resolution of the BumbleBee XB3 stereo camera); since the release of the 8800GTX, nVidia has seen three new major generations of graphics cards released, therefore one can expect these speeds to increase on more modern graphics cards. The CenSurE algorithm could also be implemented on the GPU; it has been shown to run at speeds of at least 10 Hz on a CPU implementation [5], therefore one could expect a speedup to at least 40-50 Hz in the CenSurE algorithm.

Other methods that could be used to improve the system's performance, are the use of RANSAC and bundle adjustments. These two methods were used in Howard's [2] work, and proved to increase the overall accuracy of the system. RANSAC helps to remove false feature matches by finding a consensus between groups of randomly sampled features. Bundle adjustments refer to the process of taking geometric feature information from several past frames to improve (for example) the estimated motion of the camera. In other words, one considers more frames than just the current and previous in order to improve estimations on the trajectory. Incorporating a SLAM-type methodology, in keeping persistent, or semi-persistent features in a map has proven to show improvement in vision aided navigation through small (and potentially larger) loop closures. This requires carefully maintaining the map, but has been proven computationally viable in previous works [10], [11].

Considering the previous statements, CenSurE would be recommended for a CPU-based implementation due to its proven success and speed. For a GPU-based implementation, the CenSurE or SURF algorithm would be recommended, due to CenSurE's potential speed and SURF's speed and ready implementation.

Up to this point, feature tracking methods have been discussed, which are used to extract motion information from the visual inputs; these techniques can also be described as discrete structure from motion. Another method used for motion estimation is that of optical flow. Optical flow attempts to estimate 2D (two Dimensional) velocities for areas on subsequent images; in other words, approximating a 2D motion field [20]. This can be described as continuous structure from motion. GPU implementation of optical flow has also been proven possible, and has been shown to improve the algorithm's computational speed [21]. Tick et al. [22] suggests the implementation of both discrete and continuous structure from motion techniques in parallel, fused with Kalman filtering techniques, in order to improve the overall motion estimate; it provides promising results in this application.

Veth and Raquet used a CPU-based SIFT implementation for feature extraction, which was the main limiting factor with regards to speed. An increase in system speed would allow a faster rate of correction for the INS, resulting in a more accurate trajectory estimation. The original system's speed would also not allow for adequate realtime operation; but with the speed gained by a GPU-based implementation, realtime operation is very likely. In addition to the increase in speed, the implementation of the RANSAC, batch optimisation and visual SLAM like methods will theoretically improve the trajectory estimation further. Thanks to the speedup gained from GPU implementations, the method of fusing discrete and continuous structure from motion results becomes more feasible. Implementation of this method has the potential to, once more, increase the accuracy of the motion estimations.

C. Fusion Algorithms/Estimation Algorithms

The two main candidates in the family of non-linear filtering methods that will be discussed are the UKF (Unscented Kalman Filter) and CKF (Cubature Kalman Filter) [23]. These were chosen for their performance benefits, as well as their relative ease of implementation. These filters are discussed below.

Unlike the EKF, which uses precalculated Jacobians to create a linear approximation of the system's nonlinear model, the UKF uses the unscented transformation in the approximation of the system's non-linear dynamics, improving on

many of the EKF's approximation issues. The basic premise of the UKF is that a minimal set of sample points are chosen and propagated through the nonlinear system model. The unscented transform is used to generate statistical information from the points transformed by the nonlinear model. This results in a more accurate implementation of the optimal recursive estimation equations, the basis of both filters, due to the mean and covariance of the state estimate being calculated to the second order (opposed to the first order in the EKF) [24]. Despite this higher level of accuracy, the UKF maintains an equivalent level of computational complexity. Without the need to compute linearisations using Jacobian or Hessian derivations, the implementation of the UKF becomes much simpler [25].

The CKF operates on the same principles as the UKF, where a minimal set of selectively chosen sample points are propagated through the nonlinear system model. Instead of using the unscented transform to derive the mean and covariance of the state estimate, a set of numerical integration methods known as cubature rules are used. Both the UKF and CKF are approximate Bayesian filters built in the Gaussian domain, but use different sets of deterministic weighted points in their calculations. The point set used in the CKF improves upon the EKF's in two ways; certain numerical inaccuracies present in the EKF are not present in the CKF and it eliminates filter instabilities present in the UKF. Many filtering methods are subject to huge increases in computational complexity as the number of state vectors is increased [23]. While the CKF does not solve this problem, it is less affected by it than other filters. The methods used by the CKF are also better at handling higher levels of noise and are not susceptible to the divergence problems present in other filters. [26]

Both the UKF and CKF present advantages over the EKF in ease of implementation, and potentially in performance. The CKF is, however, recommended for this system as it can potentially provide better results than the UKF.

V. Summary

The previously mentioned proposed improvements will be summarised in this section, and their potential impact on the system as a whole will be discussed. The packaged pre-calibrated BumbleBee camera has the potential to provide an increased level of accuracy in the correlation of the features to their real-world position, allowing for a higher faith in feature measurements. The speed gains of implementing the proposed feature descriptors on the GPU can increase the system's update rate, and inevitably its overall accuracy. Including the RANSAC, bundle adjustment and visual SLAM-like methods could further increase this accuracy level. Finally, the implementation of the suggested estimation algorithm can eliminate the need for difficult Jacobian derivations as well as increase accuracy of the system's estimation and its ability to handle noise.

With all of these improvements and modifications, it is suggested that the technique followed by Veth and Raquet can be used to achieve an execution speed of well beyond 2.5 Hz. If the assumption is made that the image processing steps consume the largest portion of processing time, one can tentatively expect execution speeds of 20-30 Hz; this will allow realtime operation and improved performance of the navigation system. Implementing all of the proposed improvements, as shown in Figure 3, should also enable increased accuracy and stability.

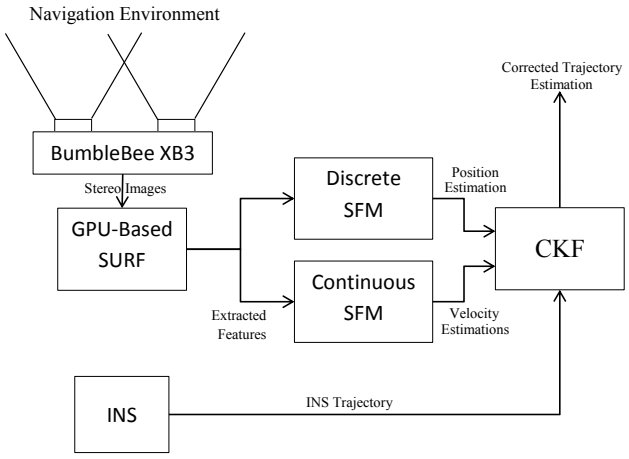

Fig. 3. The Final Proposed System

VI. Conclusion

The importance of a robust navigation solution in environments in which GPS signal is non-existent, or unreliable, have been highlighted. This paper has acknowledged the success of previous works in this area, and has proposed methods by which a system loosely based on the work of Veth and Raquet can be improved upon and brought into the state of the art. By implementing the proposed improvements mentioned in Section IV, there is great potential for increase in system performance and speed beyond that of the previous works.

References

[1] M. Veth and J. Raquet, "Fusion of low-cost imaging and inertial sensors for navigation," *Journal of the Institute of Navigation*, pp. 11–20, 2007.

[2] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," *International Conference on Intelligent Robots and Systems*, pp. 3946–3952, 2008.

[3] A. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052 –1067, june 2007.

[4] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision-Volume 2*, ser. ICCV. Washington, DC, USA: IEEE Computer Society, 1999.

[5] K. Konolige, M. Agrawal, and J. Sol, "Large scale visual odometry for rough terrain," in *Proceedings of the International Symposium on Robotics Research*, 2007.

[6] G. Bleser and D. Strickery, "Using the marginalised particle filter for real-time visual-inertial sensor fusion," in *International Symposium on Mixed and Augmented Reality*, Sept 2008, pp. 3–12.

[7] J.-P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz, "A new approach to vision-aided inertial navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, oct. 2010, pp. 4161 –4168.

[8] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, June 1981.

[9] C. Engels, H. Stewnius, and D. Nistr, "Bundle adjustment rules," in *In Photogrammetric Computer Vision*, 2006.

[10] J. Civera, O. Grasa, A. Davison, and J. Montiel, "1-point ransac for ekf-based structure from motion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, oct. 2009, pp. 3498 –3504.

[11] B. Williams and I. Reid, "On combining visual slam and visual odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, may 2010, pp. 3494 –3500.

[12] (2006) Technical datasheet for pl-b741. [Online]. Available: http://www.turnkey-solutions.com.au/cam_pixelink_pla741_series.htm

[13] *CUDA C Programming Guide - NVIDIA*, May 2011. [Online]. Available: http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/

[14] Y. Roodt, W. Visser, and W. Clarke, "Image processing on the gpu: Implementing the canny edge detection algorithm," in *PRASA*, 2007.

[15] *Bumblebee Stereo Vision Camera System*, Mar. 2011. [Online]. Available: http://www.ptgrey.com/products/bumblebee2/

[16] C. Valgren and A. Lilienthal, "Sift, surf and seasons: Long-term outdoor localization using local features," in *European Conference on Mobile Robots ECMR*, vol. 128, 2007, pp. 1–6.

[17] L. Juan and O. Gwun, "A comparison of sift, pca-sift and surf," *International Journal of Image Processing (IJIP)*, 2009.

[18] C. Wu. (2010) Siftgpu: A gpu implementation of scale invariant feature transform (sift). [Online]. Available: http://www.cs.unc.edu/ ccwu/siftgpu/

[19] T. B. Terriberry, L. M. French, and J. Helmsen, "Gpu accelerating speeded-up robust features," 2009.

[20] D. Fleet and Y. Weiss, "Optical flow estimation," in *Handbook of Mathematical Models in Computer Vision*, N. Paragios, Y. Chen, and O. Faugeras, Eds. Springer US, 2006, pp. 237–257.

[21] Y. Mizukami and K. Tadamura, "Optical flow computation on compute unified device architecture," in *14th International Conference on Image Analysis and Processing*, sept. 2007, pp. 179 –184.

[22] D. Tick, J. Shen, and N. Gans, "Fusion of discrete and continuous epipolar geometry for visual odometry and localization," in *International Workshop on Robotic and Sensors Environments (ROSE), IEEE*, oct. 2010, pp. 1 –6.

[23] I. Arasaratnam and S. Haykin, "Cubature kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, Jun. 2009.

[24] F. Daum, "Nonlinear filters: beyond the kalman filter," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 20, no. 8, pp. 57 –69, aug 2005.

[25] S. S. Haykin, *Kalman Filtering and Neural Networks*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[26] N. El-Sheimy, E.-H. Shin, and X. Niu, "Kalman filter face-off: Extended vs. unscented kalman filters for integrated gps and mems inertial," *GNSS*, 2006.

978-1-61284-993-5/11/$26.00 ©2011 IEEE