A VISION SYSTEM FOR A COGNITIVE ROBOT: COMBINATION OF MONOCULAR SLAM AND MEASURABLE MOVEMENT-CONTROL

Kai Zhou, Michael Zillich and Markus Vincze
Institute of Automation and Control,
Vienna University of Technology
Gusshausstrasse 27-29, A-1040 Vienna, Austria
{zhou, zillich,vincze}@acin.tuwien.ac.at

ABSTRACT
One essential capability of an intelligent cognitive robot is detecting objects in a scene and capturing their details based on vision. In our project, the visual SLAM (simultaneous localization and mapping) with a single camera is applied to acquire sparse landmarks for a large scale map. As we adopt the monocular SLAM with a single camera, the measurable movement-control of the mobile robot is combined, which also enhances the accuracy and the robustness of the cognition process and helps with seizing the details of some small objects.

KEY WORDS
Cognitive robot, Monocular SLAM, measurable movement-control for mobile robot, and 3D spatial point reconstruction.

1. Introduction

SLAM (Simultaneous localization and mapping) refers to the simultaneous self-localization of an intelligent robot or autonomous vehicle and the generation of a map of the observed environment. During the last two decades, the SLAM technique in robotics community has made an impressive progress. Ray laser scanning and stereo vision measurement are two major solutions for SLAM, which are developed towards maturity. With computer cartography calculations, ray laser scanning is still widely used in mobile robots and vehicle navigation [4], however, with the limitation caused by obstacles, which is unavoidable in complicated scenarios. Stereo vision systems get rid of the heavy laser devices. They turn to the calculation based on the epipolar geometric system built with two or more cameras over the scene.

Monocular SLAM using a single camera is recently becoming a popular researched solution [1-3]. This approach is in essence a variant of the stereo vision. Different view points are captured by the same camera at different positions. When the track of the camera is known, all the geometric principles used in stereo vision can be applied to monocular vision.

With monocular vision, the conventional steps for macrostructure reconstruction are: 1) image corner points are detected as features, which are recorded in the mapping database as well as the previous tracked objects; 2) the successive motion of the camera is estimated approximately for initialization; 3) the initial pose of the camera and its estimated motion offer enough information for the stereo initialization; 4) the poses of camera can then be calculated based on the tracked features using epipolar geometry; 5) meanwhile, the depth of the features can be estimated from two frames captured along the motion track with simple triangulation. Obviously this simultaneous estimation of camera motion and scene structure leads to a scale-speed ambiguity, in that fast motion in a large scene is indistinguishable from slow motion in a small scene, a typical chicken-and-egg problem.

In order to reduce the influence of inaccurate initialization and achieve robust performance, we introduce a measurable control system of the robot to refine the processing mentioned above. On one hand, we go further in the way of [2] to detect landmarks and generate a map. The mapping and tracking are split into two different threads, which makes it possible to apply different techniques to the two processes and therefore keep a low computational expense. On the other hand, a measurable control system is employed to offer the relative coordinate information when the robot is observing and moving in small scale. On the current stage, the object shape only refers to sparse 3D feature points, therefore textured surfaces of the objects and ground plane are still required. All the structures and motions are obtained in the Euclidean space, up to a global scale.

2. Related work

Classical monocular vision techniques can be divided into two categories: batch (global) or recursive (causal) approaches. Recursive approaches estimate the expected motion and optimize the observed feature points by a recursive filter, which re-uses one or more of its outputs as an input. Extended Kalman filter [5] and particle filter

[6] are two efficient filters that have been successfully employed to estimate the state of the motion in recent literature [7-9]. Efforts to improve the robustness of monocular systems have recently been made a lot by using computationally expensive recursive approach [3, 10, 11]; however the alternative batch processing technique has also been attempted. The previous work using batch processing makes many well-developed global optimization methods available [12]. Here, we will continue this batch processing based research direction in this paper.

According to the different applications, SLAM and SFM are two closely related processing methods. The on-line map, which is built by SLAM systems, usually contains the whole environment. The macrostructure of the scene is one of the important outputs of the SLAM system; therefore SLAM is often used in autonomous vehicle navigatio. In contrast, SFM is popular in the off-line microstructure reconstruction, such as 3D recovery of object surface in small-scale scenea. The computational cost of SFM is usually high and the performance of the system highly depends on the accuracy of the initialization. Both SLAM and SFM have been developed for many years with many implementations.

SLAM refers to the technique usually used by mobile robots and autonomous vehicles to build up a map within an unknown environment while keeping track of their positions. To our knowledge, EKF-SLAM [7, 8] and FAST-SLAM [3, 12] are the current state-of-the-art recursive approaches. Both of them come from the robotics community and utilize a laser rangefinder as the input sensor. Extended Kalman Filter (EKF) has been introduced into SLAM research for almost two decades and has been widely accepted. In [3, 12] the EKF was substituted by a particle filter, which was claimed to be more robust and can be applied in large environments. The robustness arises from the probabilistic independence of landmarks, which means if all the positions of the camera are known, the locations of the different landmarks become independent to each other.

More recently, conventional laser-based SLAM is reconciled with the development of the computer vision techniques and monocular visual SLAM has become an active research topic. Davison [9] shows the feasibility of real-time SLAM with a single camera system called “MonoSLAM”with the EKF-based structure, which presents the state-of-the-art to our knowledge. On the other side, the FAST-SLAM based approach was successfully applied by E. Eade and T. Drummond [1], by using a single camera. Both of these applications can also be considered as the subsets of SFM, which uses the causal or recursive estimation.

G. Klein and D. Murray [2] presented a novel method to estimate the camera pose in an unknown environment and simultaneously make a feature-points map of the scene, which is called Parallel Tracking and Mapping (PTAM). Tracking and mapping tasks have been separated into two independent threads, processed in parallel with a multi-core computer. Due to this separation, camera post tracking can run at high frame rates in one thread, while computationally expensive map estimation runs in a different thread. A brief subjective comparison between the PTAM and the MonoSLAM in [2, 13] demonstrates that PTAM offers even better performance than the long-developed EKF-based approach. According to the review of much literature, the PTAM system is considered as the state-of-the-art batch SLAM approach.

Structure from motion refers to the process of finding the 3D structure of the scene by analyzing the relative or absolute motion of one or more objects over time. C. Tomasi and T. Kanade introduce a new factorization method into the SFM at 1992 [14]. They put all the coordinates of matched features tracked through several frames into a matrix and prove that under orthographic projection this matrix is of rank 3. Then singular-value decomposition is utilised to factor this matrix into two matrices, which represent observed scene and camera motion respectively. Based on this factorization method, many researchers achieve a lot of improvements and progresses [15], which make the whole processing of SFM more precise, but also at the same time more complex and time consuming. EKF and particle filter have also been widely and successfully applied to resolve the SFM problem [16-18], while both of them need high computational cost. Although many researchers never stop to attempt to realize the real-time SFM [17, 18], to our knowledge, the real-time SFM is still not completely resolved and worth to research deeper.

In the next section, the monocular SLAM and the measurable movement-control of the robot are presented. The sections after that explain the approach in detail, give the results, analyze the limitations and point out future work.

3. Monocular SLAM

This section describes the operation of PTAM. We assume that the camera has already been calibrated and all the internal parameters are available. The system uses FAST-10 [9] corner features and an image pyramid for multi-scale feature detection. When the system is started, the five-point stereo algorithm [10] is employed to initialize the map similar to the processing described in [11-13]. For this the PTAM uses two frames of the video stream assuming a known translation between these two frames. In our system a measurable initialization is employed to provide the accurate moving distance between two view points. So the initial map has a correct scale. As a result of this initialization, the first two poses of camera are determined and can be used to calculate the coordinates of the feature points in two images grabbed at

two views. This calculation using epipolar geometry is rough and needs to be refined by the data from the subsequent image sequences.

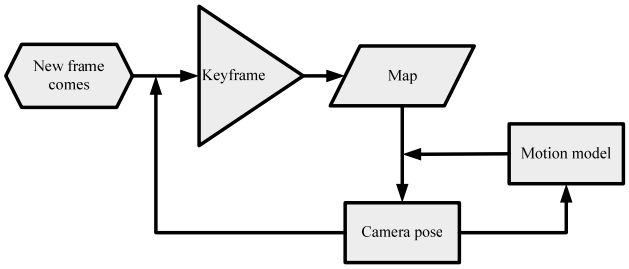

Figure 1 the processing flow of monocular SLAM, the map and keyframe generation are implemented in mapping thread, camera pose update is running in tracking thread.

Figure 1 illustrates the processing flow in the vision system. The tracking thread searches the feature points in the stored map to find the matched points in the current frame. Using the previous camera pose and a motion model, the new camera pose can be estimated. The correct and accurate camera pose can be calculated with the matched results of feature points and the estimated camera pose.

Camera pose is estimated for each new image by the tracking thread, without updating the map (circumventing the chicken-and-egg problem mentioned above). The map is only updated when a new so-called key frame comes in. Keyframes are added only when the following conditions are satisfied:

- *Minimum time duration between keyframes* The time since the last keyframe was added must exceed some frames; we use a value of twenty frames.
- *Minimum distance between new features and the features of last keyframe* The minimum distance requirement avoids the common monocular SLAM problem of a stationary camera corrupting the map, and ensures a large enough stereo baseline for robust feature triangulation.
- *Minimum umber of new features* This requirement avoids the interference of noise which can be caused by some irregular tremble of the camera, e.g. for hand-held cameras.

With the features of the new key frame a computationally expensive bundle adjustment procedure is started to calculate 3D positions of matching features from the two key frames and the 3D points are added to the map. This procedure may run for considerable time (hundreds of milliseconds) but does so in its own mapping thread, ensuring that the tracking thread continues to update camera poses at frame rate.

The map consists of a collection of M feature points located in a world coordinate frame W . Each feature point refers to a locally planar textured patch in the world. Each frame captured from the camera will be transformed to a pyramidal structure at first. The lowermost level stores the full 640×480 pixel camera snapshot, where every coordinate of the feature points will be recorded. With the larger size of the detected patches, they might be stored into the upper level of the pyramid. For example, the lowest level of the map will store all the features which only inhabit less than four pixels, the larger features will be stored into the upper level because every pixel in the upper level will be generated by four points in the lower level. The highest level stores the 80×60 pixel camera snapshot, which means each point inside represents an 8×8 pixel patch. This construction of the map allows faster searching and matching of features if the processing starts top-down.

4. Measurable Control of Robot

A simple educational mobile robot named Eddy [19] is used as the mobile platform to carry our monocular SLAM vision system. As a freely available standardized platform, the Player/Stage framework which provides basic functionalities is used to control the robot. This tool is designed around a classic proxy based architecture and usually used either to test the mechanical design or to develop higher-level algorithms. The robot is equipped with two 12 V DC ratio-motors with encoders while a caster wheel is only for support of the robot. In order to increase the flexibility of our robot, a usb Wifi pen and a usb camera are used to communicate with the server and grab the images respectively.

Our cognitive robot is designed with the ambient intelligence as the human being in everyday life; hence our vision system for the robot also has the similar functionality and processing flow. The human motion control is a good example to be imitated by our robot. For instance, if we have walked for ten minutes, it is hard to estimate even the approximate walking distance, however it is easy for us to move almost accurate ten centimetres to the left or right. The reason for this phenomenon is that humans have a better perception of short distances. That is the knowledge for the robot we call the unknown long-distance and the estimable short-distance. Obviously we are not able to know the speed of the moving camera and as a result of this, at each moment the pose of the moving camera is unknown. But we do know the direction of the motion as well as the destination. For the robot we add the so-called unknown speed, estimable destination and determinate direction as it's a priori knowledge. All the previously mentioned concepts compose the basic rules for the design of the cognitive robot.

This a priori knowledge just fits for our mobile manipulation robot. With our test, given the short moving distance (<50cm), this simple educational mobile robot Eddy can provide reasonable control accuracy (within ±5%). Once the moving distance is more than one meter,

due to the friction force of the ground and the dissonance between two motors, the control error even can accumulate to more than 20%. The application of the unknown long-distance and the estimable short-distance a priori knowledge just appropriates to our experiments. In practical applications, the curving and long-distance moving will be automatically partitioned to straight and short-distance movements.

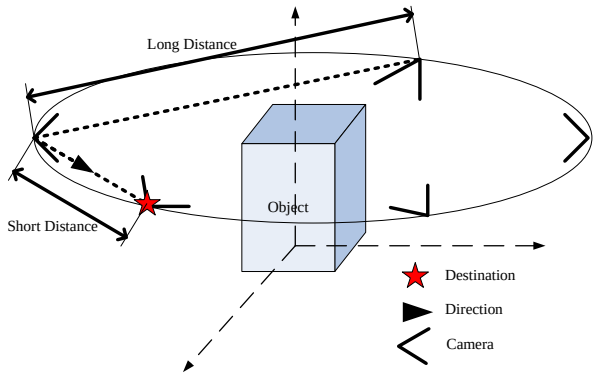

Figure 2 object exploration for the cognitive robot using monocular vision system

In figure 2, the vision system plugged on the robot takes the images from multiple views. The ellipse is just the theoretic trajectory of the camera though the robot should travel one circle around the object. Using the moving direction, destination and travel distance, the measurable control of robot can employ the epipolar geometry to calculate feature points on the object and estimate the camera poses between a pair of the starting points and destination.

Figure 3 illustrates when the system is initialized the processing flow of three threads and the invoked procedure between them. Note that the colored blocks and especially the third controlling thread constitute our additions to PTAM to fit for our mobile manipulation scenario. For every frame, the following procedure will be performed:

- A new frame is acquired from the camera, and the estimation of the prior pose is generated from the motion model.
- Map points are projected into an image according to the frame's prior pose estimate.
- All the features are searched for and matched in the image.
- The camera pose is updated from these matches.
- A final pose estimate for the frame is computed from all the matches found.
- The keyframe will be added if all conditions are satisfied.
- Check whether the loop is closed or not to justify when we should generate offline map.

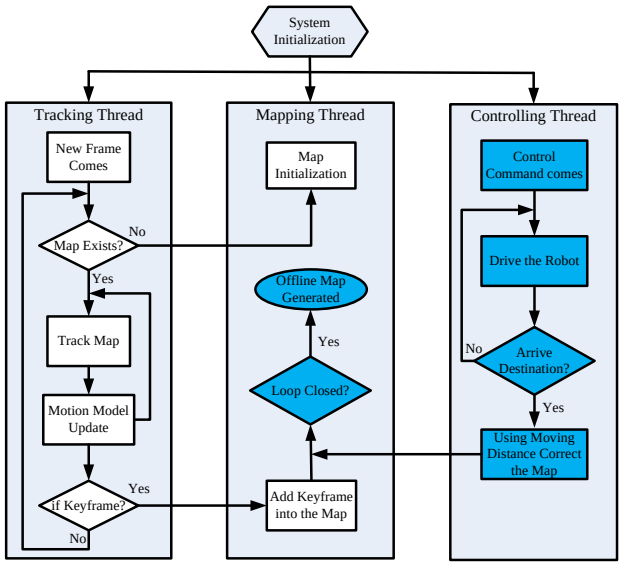

Figure 3 detailed processing flow graph of three split threads in the system

The controlling thread is initialized when the system is started, and waits for the control commands to drive the robot. If the coming command contains the precise distance which the robot is asked to move, e.g. move forward 15 centimetres, the controlling thread records the current position which comes from the estimated camera pose in the tracking thread. The robot checks whether it arrives at the destination or not by analyzing the data from its motors. When the robot finishes its movement, the new coordinates of the robot as well as the pose of the camera are calculated. These results are sent to the mapping thread to fix the estimation error and complete the loop closure.

5. Discussion

After three threads are constructed, the robot is driven by an initialization control command and then moves straight 10 centimetres to the left side. At the beginning and the end of this initialization motion, two images are grabbed and the classic epipolar geometry algorithm is employed to calculate the depth information of the feature points. Then the RANSAC algorithm is applied to compute the dominant plane. All the feature points, for which we successfully estimated the depth information, are input as a set of observed data values. Three points are selected randomly to determine a hypothetical plane and the number of supporting points is calculated. The procedure is repeated several times and the hypothesis with the largest number of supporting points is selected.

In figure 4, (a) illustrates the mobile robot Eddy, (b) shows the scene of the experiment. The control aim of the robot is shown in (c) with the white line. (d) and (e) are the snapshots of the process and the result of the

initialization respectively. (f) and (g) are the generated map. The green dots in (g) refer to the dominant plane which is the result of RANSAC. After the elimination of dominant plane, the rest of the map points are grouped to the objects considering their geometrical relationship. The comparison of the real trajectory of the robot is shown in (i), the blue one and the green one are generated by the system with and without the measurable movement-control, respectively.

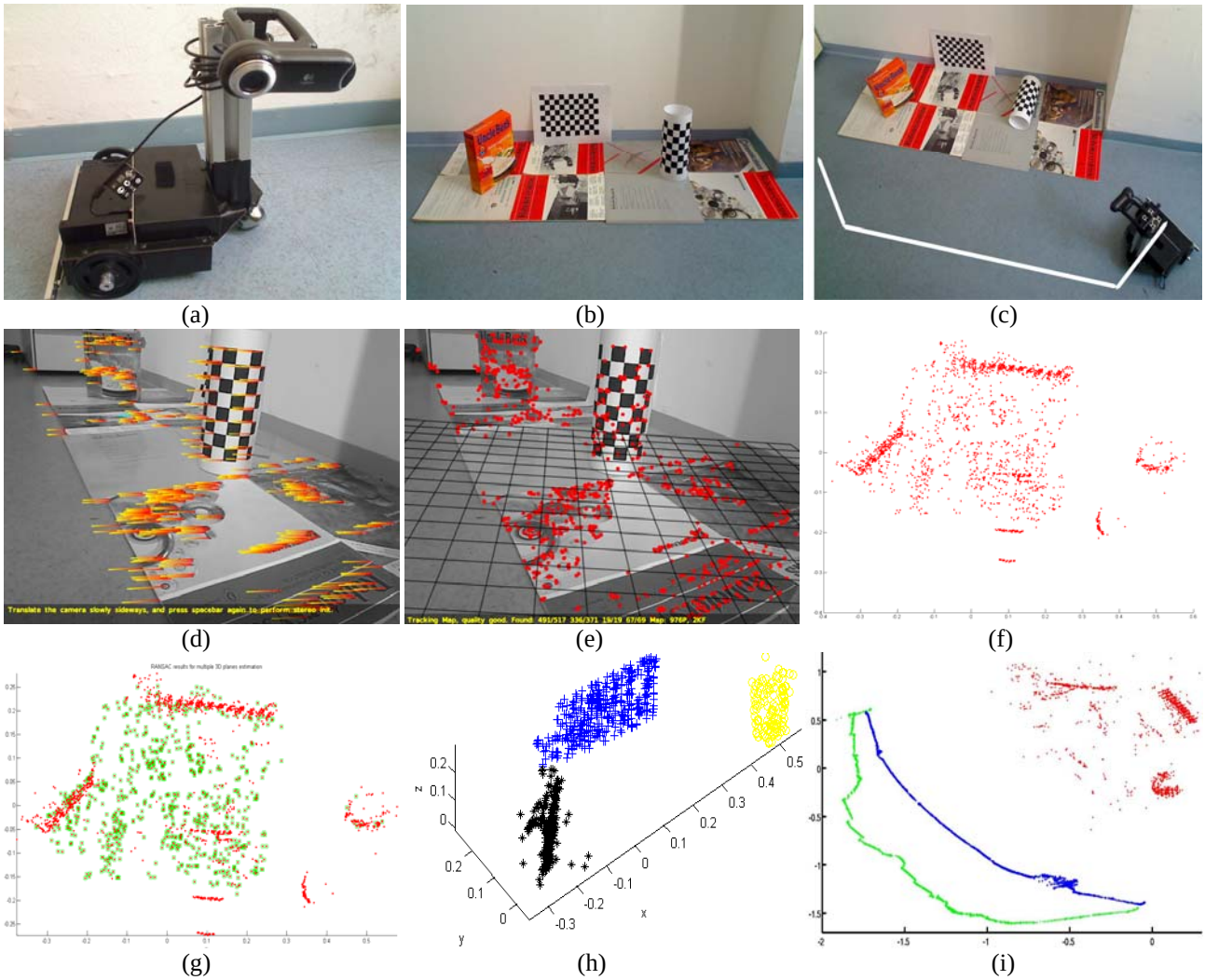

Figure 4 the experiment setting, results and comparison.

6. Conclusion

This paper has demonstrated the combination of the monocular vision system and the measurable movement-control. This solution is capable of satisfying the demand of a cognitive mobile robot to explore both the macro and micro environment. A classic batch process has been performed to implement the calculation of epipolar geometry and then the mapping, the tracking and the controlling are separated into three independent threads to enhance the accuracy and the robustness of the cognition process.

Results show that the system is able to providing excellent recovery of objects, provided the scene tracked is reasonably textured. At present we run the object extraction after the completion of loop closure by reason of the hardware limitation. Obviously this offline mode is not good for the visualization of the system. Also due to the same reason, some surface recovery techniques are not employed, which should be introduced into the system to obtain the complete 3D shape of objects in the future. Otherwise, frequently relocalization must be required. Attempts to relocalize automatically have already been investigated. Using the estimated keyframe and the camera pose to reinitialize might be a solution to this problem.

Acknowledgements

This work was supported by the EU FP7 IST Cognitive Systems Integrated Project "CogX" ICT-215181-CogX. The author Kai Zhou gratefully acknowledges the support from the China Scholarship Council (CSC).

References

[1] E. Eade, & T. Drummond, Scalable monocular slam. *Proc. 17th IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, New York, NY, 2006, 469-476.

[2] G. Klein, & D. W. Murray, Parallel tracking and mapping for small AR workspaces, *Proc. IEEE/ACM 6th Int. Symp on Mixed and Augmented Reality*, Nara, Japan, 2007, 225-234.

[3] M. Montemerlo, & S. Thrun, Simultaneous localization and mapping with unknown data association using fastslam. *Proc. IEEE Int'l Conf. on Robotics and Automation*, Taipei, Taiwan, 2003, 1985-1991.

[4] Y. Misono, Y. Goto, Y. Tarutoko, K. Kobayashi, & K. Watanabe, Development of laser rangefinder-based SLAM algorithm for mobile robot navigation. *Proc. SICE Annual Conference*, Kagawa, Japan, 2007, 392-396.

[5] B. Anderson, & J. Moore, *Optimal Filtering* (Englewood Cliffs,NJ: Prentice-Hall, 1979).

[6] A. Doucet, N. De Freitas, & N.J. Gordon, *Sequential Monte Carlo Methods in Practice* (Springer, 2001).

[7] R. Simith, M. Self, & P. Cheeseman, Estimating uncertain spatial relationships in robotics, *Autonomous Robot Vehicles*, 1990, 167-193.

[8] G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, & M. Csorba, An Experimental and Theoretical Investigation into Simultaneous Localisation and Map Building, *Proc. 6th Int'l Symposium on Experimental Robotics*, Sydney, Austrilia, 1999, 265-274.

[9] A.J. Davison, Real time simultaneous localisation and mapping with a single camera, *Proc. 9th IEEE Int'l Conf. on Computer Vision*, Nice, France, 2003, 1403-1410.

[10] D. Nister, O. Naroditsky, & J. R. Bergen, Visual odometry, *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, Washington D.C. 2004, 652-659.

[11] E. Mouragnon, F. Dekeyser, P. Sayd, M. Lhuillier, & M. Dhome, Real time localization and 3d reconstruction, *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, New York, NY 2006, 363-370.

[12] M. Montemerlo, S. Thrun, D. Koller, & B. Wegbreit, Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges, *Proc. 6th Int'l Joint Conference on Artificial Intelligence*, Acapulco, Mexico 2003.

[13] R. Castle, G. Klein, & D. Murray, Video-rate Localization in Multiple Maps for Wearable Augmented Reality, *Proc. Int'l Symposium on Wearable Computers*, Pittsburgh, PA, 2008.

[14] C. Tomasi, & T. Kanade, Shape and Motion from Image Streams: A Factorization Method, *Int'l J. Computer Vision*, 9(2), 1992, 137-154.

[15] H. Shum, K. Ikeuchi, & R. Reddy, Principal Component Analysis with Missing Data and Its Applications to Polyhedral Object Modeling, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(9), 1995, 854-867.

[16] G. Beleg, Real-Time Structure from Motion Using Kalman Filtering, Ph.D Thesis, TU Dresden, 2005.

[17] C. Tomasi, J. Zhang, & D. Redkey, Experiments with a Real-Time Structure-From-Motion System, *Experimental Robotics IV* (Springer, 1997).

[18] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, & P. Sayd, Generic and real-time structure from motion, *Proc. British Machine Vision Conference*, Warwick, UK, 2007

[19] L. Bertelli, F. Bovo, L. Grespan, S. Galvan, & P. Fiorini, Eddy: an Open Hardware Robot for Education, *Proc. 4th Int'l Symposium on Autonomous Minirobots*, Buenos Aires, Argentina, 2007.

[20] H. Stewenius, C. Engels, & D. Nister, Recent developments on direct relative orientation, *ISPRS Journal of Photogrammetry and Remote Sensing*, 2006, 284-294.