

Real-Time 3D Path and Map Estimation Using a Multi-Camera System and a FastSLAM Algorithm

¹Chayakorn NETRAMAI, ²Hubert ROTH, ³Anatoly SACHENKO

¹The Sirindhorn International Thai-German Graduate School of Engineering,
King Mongkut's University of Technology North Bangkok,
1518 Pracharat 1 Road, Bangsue, Bangkok, 10800, Thailand

²Institute of Automatic Control Engineering,
Department of Electrical Engineering and Computer Science,
University of Siegen, Hölderlin str. 3, 57076, Siegen, Germany

³Department of Computer Science and Econometrics,
Silesian University of Technology,
Roosevelta St. 26-28, Zabrze, 41-800, Poland
E-mail: chayakorn.n.sse@tggs-bangkok.org

Received: 15 May 2013 / Accepted: 16 August 2013 / Published: 30 August 2013

Abstract: This work proposed a real-time 3D map building system based on the visual odometry information derived from a Multi-Camera hardware. The accurate odometry information and images derived from the proposed intuitive Multi-Camera hardware, so called Multi-Camera Unit (MCU), are used as an input for the 3D map building process which cooperates the FastSLAM algorithm as a map and path estimator. Experiment results show that the proposed system yield good estimation of 3D path and map in real-time. Using the estimated 3D path and map, a showcase application of 3D photorealistic map building can be implemented with ease where it can be run on top of all process in real-time manner using a standard PC. *Copyright © 2013 IFSA.*

Keywords: Visual odometry, Multi-camera systems, 3D map building, FastSLAM.

1. Introduction

Real-time 3D path and map estimation using only visual data are two challenging problems which have been intensively studied by the research community in the past years [1-3]. A 3D path is a set of pose information that normally describe the location and the orientation of the sensor or agent within a certain time frame while a 3D map is an abstract representation of the real environment usually describe by a set of landmarks or in some cases as a fully 3D map with complete geometric description

and texture information that replicates those of the real environment.

In order to successfully estimate the 3D path and 3D map, the accurate 3D motion estimation of the sensor is needed. This is prove to be a difficult task for the case where only the visual information from the sensor, e.g. stereo camera setup, is used mainly due to its limited accuracy [4, 5]. While is it true that the accuracy of such sensor can be improved by using high resolution images and realizing better software algorithm but both solutions usually end up hurting the real-time performance of the system [6].

This work introduces a multi-camera hardware to be used as a 3D visual sensing device for real-time 3D path and map estimation tasks. By utilizing an efficient multi-camera arrangement, a very robust visual odometry can be realized and it can be shown that the solution is able to perform in real-time manner [7]. On top of that, a probabilistic approach for real-time path and map estimation is implemented to cooperate the odometry information and the visual information from the sensing device in order to estimate the 3D path and map in the most efficient way. A sample application of 3D photorealistic map building is also given at the end of the paper.

This work is organized as follow: Section 0 explains the visual odometry technique in detail including the problems commonly found in case of single camera systems and the solution realized using multi-camera systems, Section 0 introduces the multi-camera hardware and its visual odometry implementation, Section 0 contains an information about the real-time 3D path and map estimation using a probabilistic approach, Section 0 contains the experiment procedures and results and lastly Section 0 contains the conclusion of the work.

2. Visual Odometry

Visual odometry is a technique that estimates the ego motion of the camera using a sequence of images acquired during the movement. It can be used in autonomous systems in order to safely and accurately navigate itself around an unknown environment [8-10]. Commonly, the input of the visual odometry process is a set of 3D feature points extracted from the scene images using a 3D visual sensor, e.g. a stereo camera. It is clear that the accuracy of the estimated ego motion relies on the accurate 3D location of the extracted feature points. If the 3D location of the feature points contains large uncertainty then the estimated ego motion is poor and unusable.

Much of the work in the past decades has been done in order to improve the accuracy and robustness of the visual odometry system based on software algorithms used, e.g. sub-pixel accuracy feature detection [11, 12], improvement of the 3D motion estimation based on probabilistic approach [13, 14], etc. While these techniques improve the accuracy and robustness of the visual odometry, the performance of such system will always be hindered by the limitation of the visual hardware, namely the motion ambiguity problems.

This work proposed a new multi-camera hardware as an input sensor for visual odometry tasks. It can be shown that, by using the proposed multi-camera system, the motion ambiguity which is the main hindrance of the single camera visual odometry system can be eliminated and therefore high accuracy ego motion estimation can be obtained.

2.1. Visual Odometry Using Single Camera Systems

Normally one stereo camera is used on an autonomous system as an input sensor for the visual odometry task. In most case, this camera is usually pointing forward, i.e. looking in the direction of travelling. This configuration leads to two situations where the limitation of the stereo camera hardware can cause poor motion estimation result: (1) the *translational ambiguity* and (2) the *rotational ambiguity*. These two situations are explained in detail in the following text.

2.1.1. Translational Ambiguity

It is well known that the estimated depth value of the stereo camera with small baseline contains some uncertainty that grows larger when the distance between the camera and object increases (Fig. 1). As can be seen, this uncertainty is most biased in the direction along the optical axis of the camera. The so called *translational ambiguity* problem occurs when the motion to be estimated, e.g. a small translation, is aligned with the camera optical axis. This is illustrated in Fig. 2, where the error e_z represents the uncertainty of the feature point's location along the direction of the camera's optical axis.

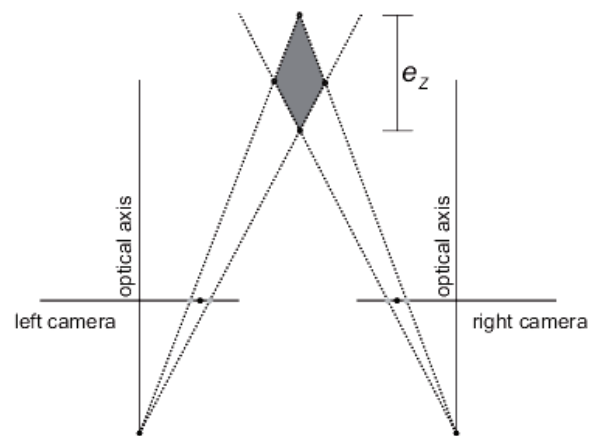


Fig. 1. The uncertainty of the depth calculation of a small baseline stereo camera.

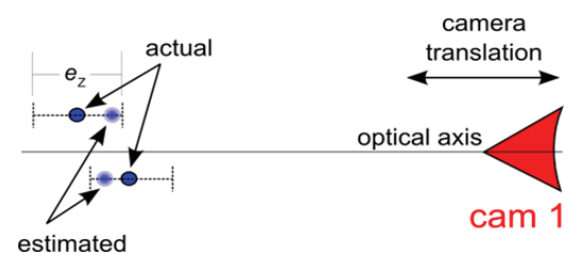


Fig. 2. The translational ambiguity caused by the uncertainty of the stereo depth calculation. The solid circles and dotted circles indicate the actual and estimated location of the feature point, respectively.

The translational ambiguity takes place when the motion to be estimated is smaller or equal to the error e_z . This is due to the obvious fact that it is not possible to use the feature points as reference points when the magnitude of the uncertainty is larger than the magnitude of the motion to be estimated itself.

Although this problem can be solved by increasing the resolution of the image sensor or by applying an estimation technique, it will be shown in the later section that multi-camera systems can be used to solve this problem easily by addressing the ambiguity problem directly at the root of the problem.

2.1.2. Rotational Ambiguity

The rotational ambiguity occurs when a small rotation around the axis which is perpendicular to the optical axis take place. In this case, the motion field obtained from a small rotation is similar to the motion field obtained from a small translation perpendicular to the optical axis, which leads to the wrong motion estimation result. This situation is illustrated in Fig. 3. The red triangle represents the camera and the arrows-filled area presents the image and the motion field of the feature points within the image. When the camera moves to the left, the position of the feature points shift to the right parallel to the camera's movement direction (Fig. 3a). However, when a small rotation around the axis that is perpendicular to the optical axis of the camera is introduced, the similar motion field is obtained (Fig. 3b). Therefore such a small rotation can be confused as a translation from the camera's point of view.

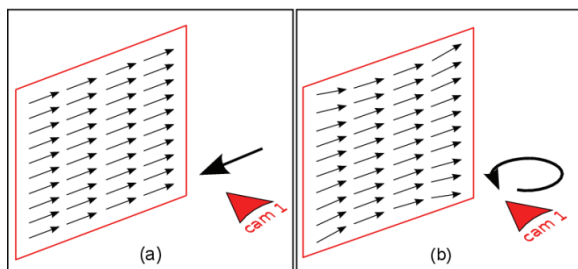


Fig. 3. Rotational ambiguity between a small translation (left) and a small rotation around an axis perpendicular to the optical axis (right). The motion fields obtained from both cases are similar.

Traditionally this problem can be solved by increasing the field of view of the optical lenses but the main drawback would be the degrading of the stereo image quality since the camera resolution remains the same while larger portion of the environment is seen. However, it will be shown in the next sections that a multi-camera system can be used to solve this problem without sacrifice of the stereo image quality.

2.2. Elimination of Motion Ambiguity Using Multi-Camera Systems

By using the multi-camera systems, i.e. a system with two or more cameras, it can be shown that the motion ambiguity problem can be eliminated. The key solution is the placement of the auxiliary camera whose optical axis is perpendicular to the direction of movement in order to maximize the movement detection. Some solutions to the motion ambiguities are given below.

2.3. Eliminate Translational Ambiguity Using Three Cameras

By using three cameras, the information from the cameras that are looking in the direction perpendicular to the translation vector can be used to eliminate the translational ambiguity. This is illustrated in Fig. 4. The estimated feature point location from the camera that points in the movement direction (indicated as cam1 in red) is corrupted by the stereo depth error. However, the other two cameras can detect the fine movement with less uncertainty and therefore the accurate translation estimation can be obtained.

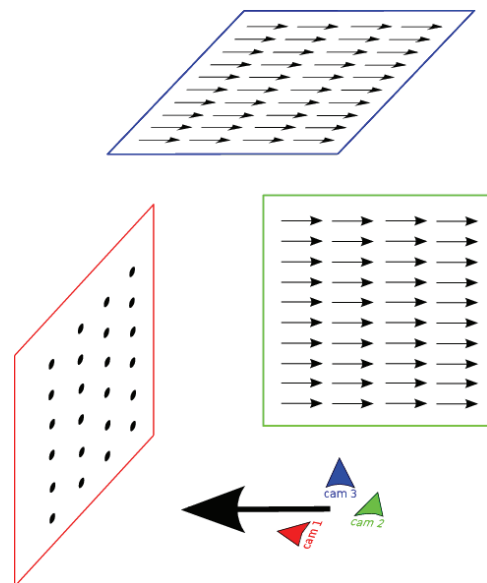
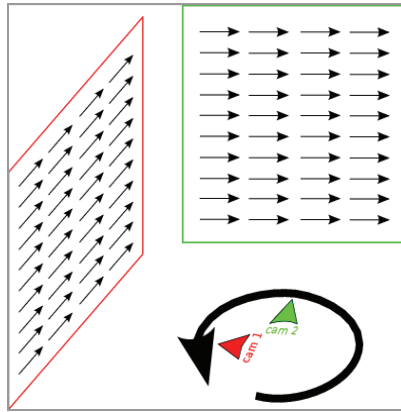


Fig. 4. Elimination of translational ambiguity using three cameras.

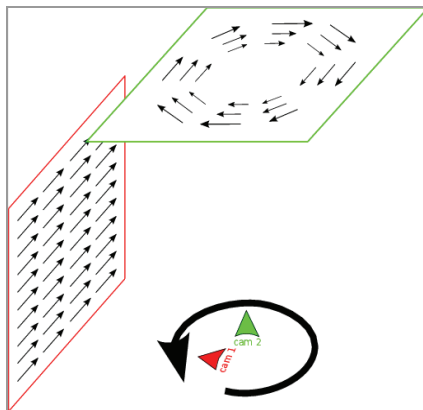
2.3.1. Eliminate Rotational Ambiguity Using Two Cameras

The multi-camera system solves the rotational ambiguity problem by introducing a second camera whose optical axis points in the direction perpendicular to the first camera's optical axis. For example, in Fig. 5a, a second camera (indicated as cam2 in green) which points to the right direction of

the first camera is introduced which provides one more image to the system. Although the optical axes of both cameras are pointing in the perpendicular direction to the rotation of the observer, it is now easier to distinguish a rotation from a translation. That is, if the motion field on both cameras has similar magnitudes and directions then the detected motion can be confirmed as a rotation.



(a)



(b)

Fig. 5. Elimination of the rotational ambiguity using two cameras: (a) 1st solution and (b) 2nd solution.

Another solution is to point a second camera along the axis of rotation as shown in Fig. 5b. In this case the rotation can be directly detected by the second camera and hence by using the motion field from both cameras the rotational ambiguity can be completely eliminated.

3. Visual Odometry Using Multi-Camera Hardware

This work proposed a multi-camera unit (MCU) for high accuracy visual odometry tasks. An MCU (Fig. 6) is used as an input sensor to demonstrate the improvement over the conventional single camera system. It consists of three stereo cameras whose

optical axes are all arranged orthogonally to each other, i.e. in x-, y- and z-directions. More information about the MCU can be found in [15, 16]. The main benefits of the proposed multi-camera system are:

- **Elimination of the motion ambiguity problems;** auxiliary cameras can be used to eliminate the ambiguity problems as explained in the previous section.
- **Increased reliability;** since more images from multiple views are available therefore even if one camera is corrupted by noise the other cameras can be used to support the system.
- **Higher acquisition rate;** multiple acquisition rate compared to single camera system. This is a big benefit for some applications such as remote exploration and map building where high acquisition rate means more terrain can be explored within a given time.
- **Easy and robust outlier rejection;** by using the proposed multi-camera system, a simple yet effective outlier rejection scheme can be implemented.

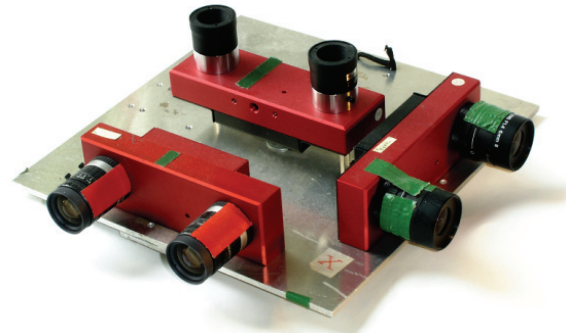


Fig. 6. The MCU.

The overview of the multi-camera system is shown in Fig. 7.

3.1. Feature Extraction and Feature Matching

The MCU uses three stereo cameras to acquire 2D and 3D images from the environment. These images are then used for the feature detection process and motion estimation process. Currently, the feature extraction is done using a corner detector to extract corners from the environment. After each acquisition the corners or feature points are then matched across two successive frames using normalized cross correlation (NCC) [17]. The NCC coefficient between two feature points f and t is defined by:

$$\frac{\sum_{x,y} [f_{(x,y)} - \bar{f}][t_{(x-u,y-v)} - \bar{t}]}{\left\{ \sum_{x,y} [f_{(x,y)} - \bar{f}]^2 \sum_{x,y} [t_{(x-u,y-v)} - \bar{t}]^2 \right\}^{0,5}}$$

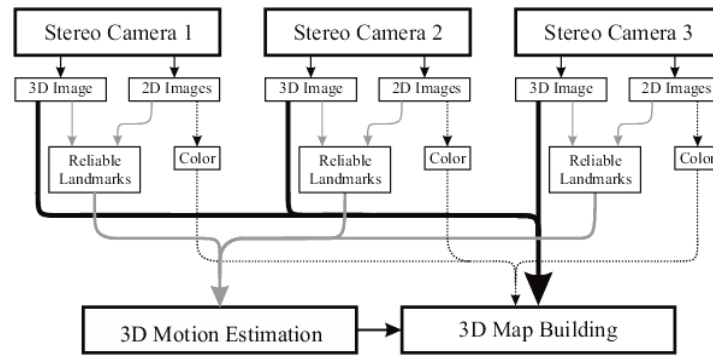


Fig. 7. System overview.

$$\frac{\Sigma_{x,y} [f_{(x,y)} - \bar{f}] [t_{(x-u,y-v)} - \bar{t}]}{\left\{ \Sigma_{x,y} [f_{(x,y)} - \bar{f}]^2 \Sigma_{x,y} [t_{(x-u,y-v)} - \bar{t}]^2 \right\}^{0.5}} \quad (1)$$

where $f_{(x,y)}$ and $t_{(x,y)}$ are the pixel intensity at correspond locations within previous and current image frame and \bar{f} and \bar{t} are the mean intensity in the 9×9 pixels region under point f and t , respectively. The result of these processes is the matched feature points that can be used to estimate the motion of the camera at each time step.

3.2. Elimination of Motion Ambiguity Using Three Cameras

By using the 2D images from all three stereo cameras equipped on the MCU, a very effective motion detection scheme can be applied. This is done by first using 2D movement from all three cameras and then use a look up table to conclude the real 3D movement of the multi-camera hardware. Fig. 8 illustrates the direction of such 2D movement and Table 1 gives some examples of the resultant 3D motion according to the detected 2D movements. This method is simple yet effective and not prone to motion ambiguity and therefore accurate real-time motion estimation results can be obtained.

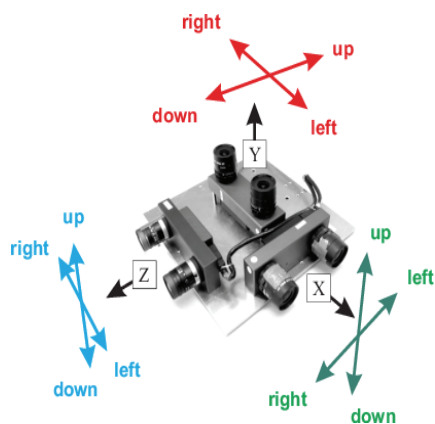


Fig. 8. Direction of the 2D movement of all three cameras on the MCU.

Table 1. 3D motion detection using 2D motion information.

X-camera	Z-camera	Y-camera	Motion
up	up	-	Up
down	down	-	Down
left	-	up	Left
right	-	down	Right
-	left	left	Forward
-	right	right	Backward
-	down	down	Rotate CW around x-axis
-	up	up	Rotate CCW around x-axis
up	-	right	Rotate CW around y-axis
down	-	left	Rotate CCW around y-axis
left	left	-	Rotate CW around z-axis
right	right	-	Rotate CCW around z-axis
up, right	up, right	-	Up and rotate CCW around z-axis

note: CW = clockwise, CCW = counterclockwise

3.3. Estimate Six-DOF Motion Using ICP Algorithm

Currently, the six DoF motion ($T_x, T_y, T_z, roll, pitch, yaw$) is estimated using the iterative closest point (ICP) algorithm [18]. Since the correspondence of the feature points are available from the matching process the ICP algorithm can be applied directly by iteratively minimize the Euclidean distance between two set of matched feature points $\min \|p - p'(R, t)\|$, where p and p' are the feature point from the previous time step and current time step, respectively. More details about these procedures can be found in [16].

4. Real-Time 3D Map Building Using FastSLAM Algorithm

The 3D path and map estimation is a task where both path and map are to be incrementally built using

only the information from the input sensor, usually installed on a moving agent (e.g. an autonomous robot), to determine the travelled path and the feature point map of the environment. This is a well known problem called simultaneous localization and map building (SLAM) problem. The solution to SLAM usually implement some probabilistic approaches in order to deal with the noisy measurement of the sensor by cooperating models of uncertainties of the measurement and of the feature points within the map. Several SLAM solutions are available [13, 19, 20] but in this work the FastSLAM solution was used in order to deal with the 3D path and map building task in real-time.

4.1. FastSLAM Algorithm

The objective of the FastSLAM algorithm is to integrate the new feature point detected using the MCU into a consistent 3D feature point map and at the same time maintain the location of the MCU related to the map coordinate system. The main components of the FastSLAM algorithm are the path of the MCU denoted by $s^t = \{s_1, s_2, \dots, s_t\}$ and the 3D feature point map denoted by $\Theta = \{\theta_1, \theta_2, \dots, \theta_N\}$. Both are being maintained by a set of particles within the FastSLAM algorithm. The posterior probability of the feature point map given the MCU path is expressed as:

$$p(\Theta, s^t | z^t, u^t, n^t), \quad (2)$$

where z^t, u^t and n^t are set of feature points measurement $\{z_1, z_2, \dots, z_t\}$, set of MCU odometry measurement $\{u_1, u_2, \dots, u_t\}$, and set of data associate variables $\{n_1, n_2, \dots, n_t\}$, respectively. The probabilistic model of the feature points measurement is $p(z_t | s_t, \Theta, n_t)$ and the probabilistic motion model of the MCU is $p(s_t | u_t, s_{t-1})$. The FastSLAM algorithm solves the feature points location estimation independently from the MCU pose estimation. By assuming that the path of the MCU is known, the location of each feature point can be estimated independently. Using this assumption, the factorization of the posterior distribution in Eq. 2 becomes:

$$p(\Theta, s^t | z^t, u^t, n^t) = p(s^t | z^t, u^t, n^t) \prod_{n=1}^N p(\theta_n | s^t, z^t, u^t, n^t) \quad (3)$$

The FastSLAM algorithm samples the path using a particle filtering technique where a number of m particles are used to maintained m unique MCU paths and m feature point maps. For more information regarding the FastSLAM algorithm please refer to [13, 19, 21].

4.2. FastSLAM Implementation

The software implementation of the FastSLAM algorithm using MCU consists of the following process:

- Update the current MCU pose and the new MCU measurement;
- Data association of known feature points;
- Adding new feature points to the map;
- Compute weight of the particles;
- Resample the particles;
- Calculate sample proposal;
- Update known feature points using EKFs.

The relationship of these processes is shown in Fig. 9.

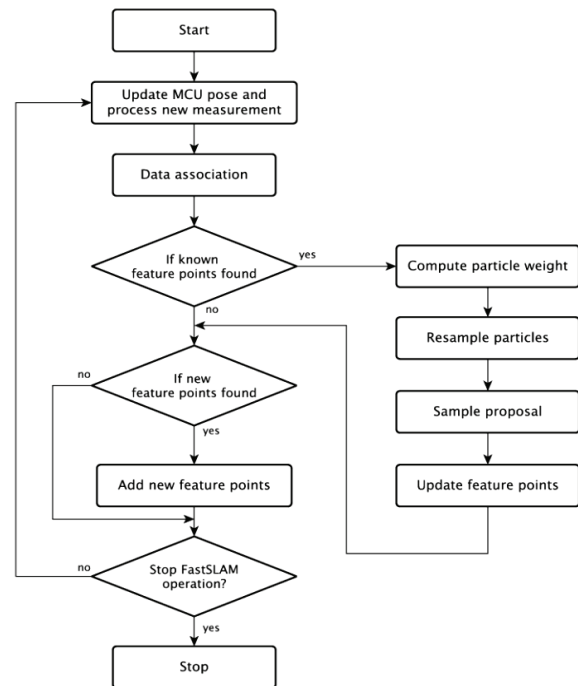


Fig. 9. Relationship between processes within FastSLAM algorithm.

All software modules in this work were implemented using C++ Language. The MCU software is running on a host computer with Intel Core 2 Quad processor Q6600 running at 3.0 GHz and 4 GB of RAM. The selected operating is a 32-bit Linux with kernel version 2.6.27.

5. Experiments and Results

Extensive real-world experiments were done to compare the performance of the visual odometry process using single camera system and multi-camera systems. The tests were done using the proposed multi-camera hardware as shown in Fig. 6. To simulate the single camera systems, only images from one selected camera are used. For multi-camera systems, images from all three cameras are used. In

both cases the visual odometry process are identical, i.e. same corner detector, same motion estimation algorithm. Fig. 10 shows the test environment to be used for the experiments. The test environment has an approximate dimension of $4 \times 5 \times 3$ meters and it contains no artificial landmarks.

5.1. Experiment Procedures

The following procedures are carried out during each test:

- Initializing the MCU hardware and software (i.e. map, path)
- Move the MCU along the given test path
- While moving, the MCU hardware constantly acquires new images and calculates visual odometry output. The odometry output is then used to estimate the 3D path and map of the environment.

The experiments are divided into three parts: (1) *estimation of a rotation movement*, (2) *estimation of a translation movement*, and (3) *estimation of the MCU path and feature point map*. Some significant results including the analysis are presented in the following section.



Fig. 10. The test environment.

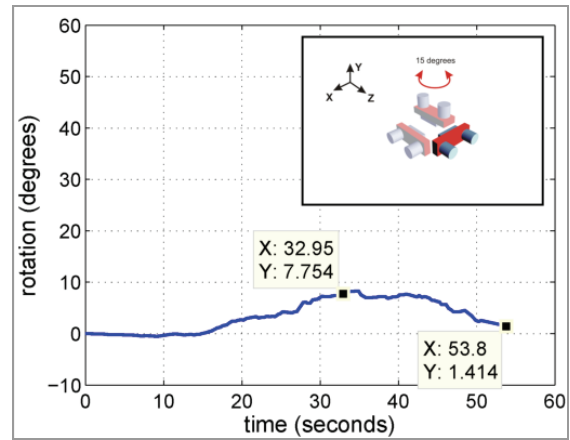
5.2. Estimation of a Small Rotation

In this test, a small rotation of 15 degrees back and forth is given to the MCU. By using the images from only one camera, the estimated rotation is poor due to the rotational ambiguity problem (Fig. 11a). On the other hand, when using three cameras the estimated rotation is better (Fig. 11b).

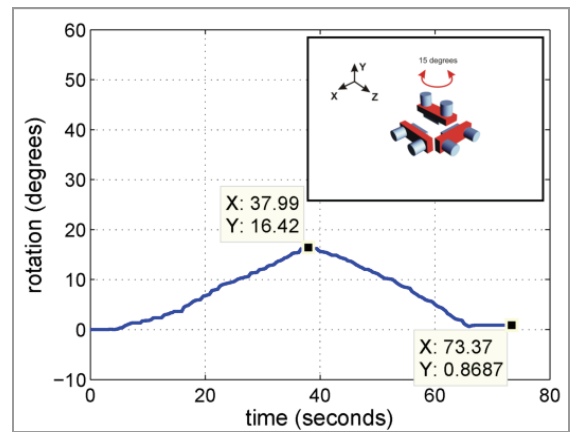
5.3. Estimation of a Small Translation

In this test, a small translation of 50 mm back and forth is given to the camera. By using the input images from only one camera, the estimated

translation is poor due to the translational ambiguity problem (Fig. 12a). On the other hand, when using three cameras the estimated translation is better (Fig. 12b).



(a)

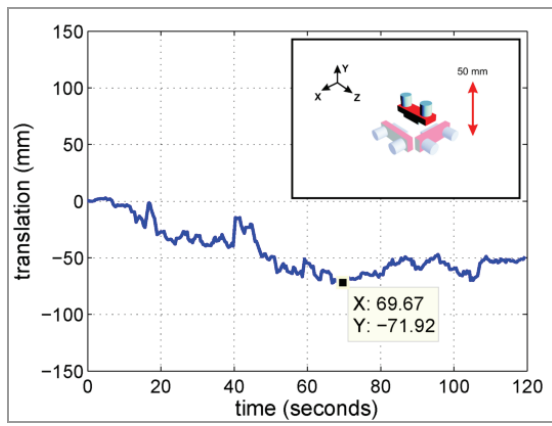


(b)

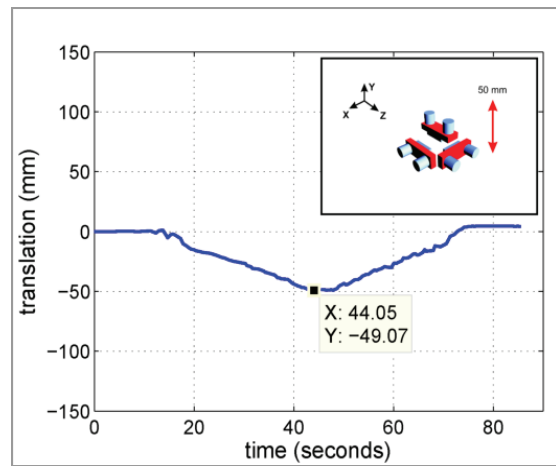
Fig. 11. Estimate 15 degrees rotation around the y-axis using (a) one camera and (b) three cameras. By using one camera, the estimated rotation is about 7 degrees which is far from the actual rotation (15 degrees). By using three cameras, the estimated rotation is about 16 degrees which is close to the actual rotation. The estimated motion also correctly indicates the movement of the MCU back to 0 degree at the end of the test.

5.4. Estimation of the MCU Path and Feature Point Map

In this test, the MCU was moved along the 2-shape trajectory as described in Fig. 13a. The estimated 3D path closely resembled the real path, which can be confirmed by checking the geometric dimension of the estimated path shown in Fig. 13b to the dimension of the real path. During the test, the 3D feature point map is also incrementally built and maintains by the FastSLAM algorithm, a screen capture of such feature point map is shown in Fig. 14a.

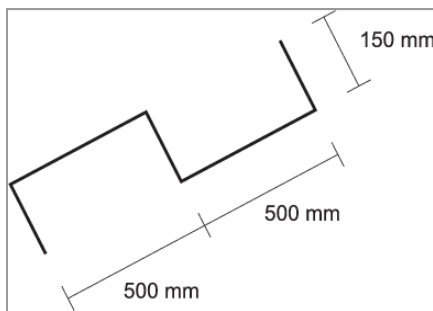


(a)

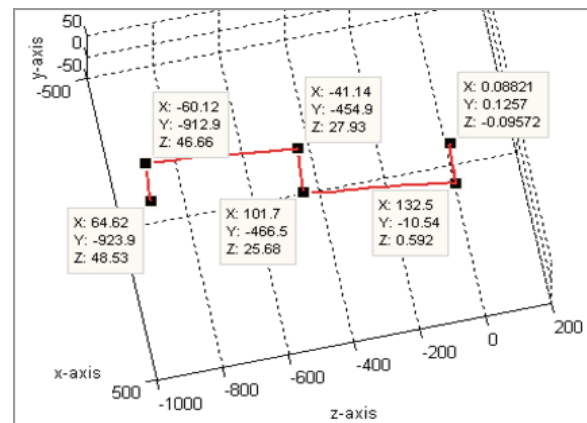


(b)

Fig. 12. Estimate 50 mm translation along the y-axis using (a) one camera and (b) three cameras. By using one camera, the estimated translation is about 70 mm which is far from the actual translation (50 mm). By using all three cameras, the estimated translation is about 49 mm which is close to the actual translation (50 mm). The estimated motion also correctly indicates the movement of the MCU back to start position at the end of the test.

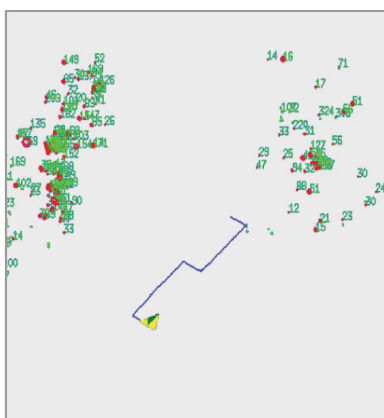


(a)

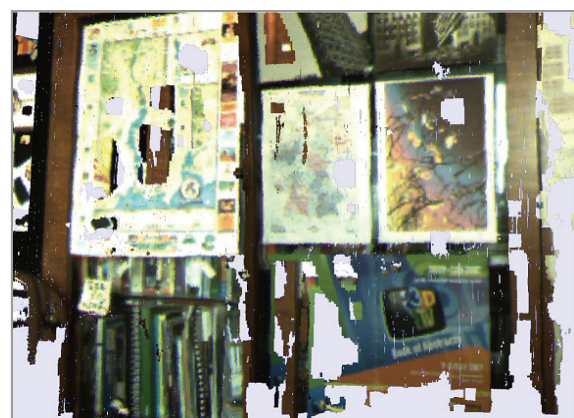


(b)

Fig. 13. Estimate a 2-shapetrajectory using three cameras: (a) the real path and its dimensions, (b) the estimated 3D path using the odometry information from the MCU and the FastSLAM algorithm.



(a)



(b)

Fig. 14. (a) A screen captures of the estimated 3D path of the 2-shapetest trajectory and (b) a 3D photorealistic model of the test environment obtained by combining multiple 3D images from the MCU along the test path.

5.4. Sample Application: 3D Photorealistic Map Building

In addition to the feature point map, a simple application of the 3D photorealistic map building is also implemented. By using the exact location of the MCU during each movement, the captured 3D images from all MCU stereo cameras can be combined to generate a 3D photorealistic model of the real world scene as shown in Fig. 14b.

6. Conclusions

This work introduces a novel multi-camera hardware as a way to improve the visual odometry process. By using the MCU the motion ambiguity commonly found in single camera system is eliminated and the resulting odometry is improved. A probabilistic approach for 3D path and map estimation, so called, FastSLAM algorithm is also implemented to cope with the uncertainty of the MCU measurement and thus a good result in path and map estimation is obtained. This has been shown by the results of the real-world experiments where the motion ambiguities are effectively eliminated when the proposed multi-camera hardware is used. Finally, a test application of real-time 3D photorealistic map building that make use of the accurate odometry output and the FastSLAM algorithm was implemented and good results were obtained.

References

- [1]. H. Surmann, A. Nüchter, K. Lingemann, J. Hertzberg, 6D SLAM – mapping outdoor environment, *Journal of Field Robotics*, Vol. 24(8/9), 2007, pp. 699-722.
- [2]. A. J. Davison, Y. G. Cid, Real-time 3D SLAM with wide-angle vision, in *Proceedings of Symposium on Intelligent Autonomous Vehicle (IAV '2004)*, Lisboa, Portugal, 2004.
- [3]. P. Firoozfam, S. Negahdaripour, A multi-camera conical imaging system for robust 3D motion estimation, positioning and mapping from UAVs, in *Proceedings of IEEE International Conference on Advance Video and Signal Based Surveillance*, 2003, Miami, USA.
- [4]. M. A. Garcia, A. Solanas, 3D simultaneous localization and modeling from stereo vision, in *Proceedings of International Conference on Robotics & Automation*, 2004.
- [5]. C. Knöppel, Stereobasierte und spurgenaue Erkennung von Straßenfahrzeugen im Rückraum eines Straßenfahrzeuges, Ph.D. Thesis, *Otto-von-Guericke-University*, Magdeburg.
- [6]. K. Konolige, D. Beymer, SRI small vision system: user manual for software version 4.4.
- [7]. C. Netramai, H. Roth, A. Sachenko, High accuracy visual odometry using multi-camera systems, in *Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS '2011)*, 2011, Prague, Czech Republic, pp. 263-268.
- [8]. J. M. Sáez, F. Escolano, A global 3D map-building approach using stereo vision, in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004, pp. 1197-1202.
- [9]. M. Maimone, Y. Cheng, L. Matthies, Two years of visual odometry on the mars exploration rovers, *Journal of Field Robotics*, Vol. 24, No. 3, 2007, pp. 169-186.
- [10]. A. Howard, Real-time stereo visual odometry for autonomous ground vehicles, in *Proceedings of the International Conference on Intelligent Robots and Systems, IROS 2008*, 2008.
- [11]. L. Lucchese, S. Mitra, Using saddle points for subpixel feature detection in camera calibration targets, in *Proceedings of the 2002 Asia-Pacific Conference on Circuits and Systems (APCCAS '02)*, Vol. 2, 2002.
- [12]. X. Zhang, L. Wang, X. Sun, Q. He, Sub-pixel accuracy for extracting corner point based on support vector regression, in *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation (ICMTMA '09)*, Vol. 3, 2009, pp. 330-333.
- [13]. M. Montemerlo, S. Thun, Simultaneous localization and mapping with unknown data association using FastSLAM, in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '03)*, Vol. 2, 2003, pp. 1985-1991.
- [14]. J. L. Blanco, Derivation and implementation of a full 6D EKF-based solution to bearing-range SLAM, *University of Malaga*, Tech. Report, 2008.
- [15]. C. Netramai, H. Roth, Real-time photorealistic 3D map building using mobile multiple stereo cameras setup, in *Proceedings of the 3DTV Conference*, Kos Island, Greece, 2007.
- [16]. C. Netramai, Using mobile multi-camera unit for real-time 3D motion estimation and map building of indoor environment, Ph.D. Dissertation, *University of Siegen*, Germany, 2011.
- [17]. J. P. Lewis, Fast normalized cross-correlation, *Industrial Light and Magic*, Tech. Report, 1995.
- [18]. P. J. Besl, N. D. McKay, A method for registration of 3-D shapes, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992.
- [19]. M. Montemerlo, FastSLAM: a factored solution to the simultaneous localization and mapping problem with unknown data association, Ph.D. Thesis, *Carnegie Mellon University*, 2003.
- [20]. Z. Kurt-Yavuz, S. Yavuz, A comparison of EKF, UKF, FastSLAM2.0 and UKF-based FastSLAM algorithms, in *Proceedings of the IEEE 16th International Conference on Intelligent Engineering Systems (INES' 2012)*, Lisbon, Portugal, 2012.
- [21]. M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, FastSLAM2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges, in *Proceedings of the International Conference on Artificial Intelligence*, 2003, pp.1151-1156.



Handbook of Laboratory Measurements and Instrumentation

Maria Teresa Restivo
Fernando Gomes de Almeida
Maria de Fátima Chouzal
Joaquim Gabriel Mendes
António Mendes Lopes

The Handbook of Laboratory Measurements and Instrumentation presents experimental and laboratory activities with an approach as close as possible to reality, even offering remote access to experiments, providing to the reader an excellent tool for learning laboratory techniques and methodologies. Book includes dozens videos, animations and simulations following each of chapters. It makes the title very valued and different from existing books on measurements and instrumentation.



International Frequency Sensor Association Publishing

Order online:

http://www.sensorsportal.com/HTML/BOOKSTORE/Handbook_of_Measurements.htm

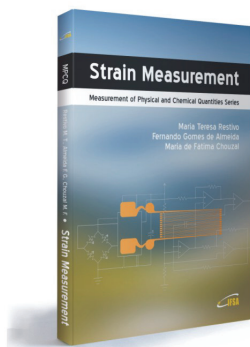


International Frequency Sensor Association (IFSA) Publishing

Maria Teresa Restivo, Fernando Gomes de Almeida, Maria de Fátima Chouzal

Strain Measurement

Measurement of Physical and Chemical Quantities Series



Formats: printable pdf (Acrobat)
and print (hardcover), 106 pages

ISBN: 978-84-616-0067-0,
e-ISBN: 978-84-615-9897-7

'*Strain Measurement*' deals with measurement of stresses and strains in mechanical and structural components. This topic is related to such diverse disciplines as physical and mechanical sciences, engineering (mechanical, aeronautical, civil, automotive, nuclear, etc.), materials, electronics, medicine and biology, and uses experimental methodologies to test and evaluate the behaviour and performance of all kinds of materials, structures and mechanical systems.

The material covered includes:

- Introduction to the elementary concepts of stress and strain state of a body;
- Experimental extensometry measurement techniques;
- Basic instrumentation theory and techniques associated with the use of strain gauges;
- Optical fibre based extensometry;
- Uncertainty estimation on the measurement of mechanical stress;
- Supplemented multimedia components such as animations, simulations and video clips.

The different subjects exposed in this book are presented in a very simple and easy sequence, which makes it most adequate for engineering students, technicians and professionals, as well as for other users interested in mechanical measurements and related instrumentation.

http://sensorsportal.com/HTML/BOOKSTORE/Strain_Measurement.htm