2008 IEEE/RSJ International Conference on Intelligent Robots and Systems
Acropolis Convention Center
Nice, France, Sept, 22-26, 2008

Robust Vision-based Autonomous Navigation against Environment Changes

Jungho Kim, Yunsu Bok and In So Kweon

Abstract— **Recently, many vision-based navigation methods have been introduced as an intelligent robot application. However, many of these methods mainly focus on finding an image in the database corresponding to a query image. Thus, if the environment changes, for example, objects moving in the environment, a robot is unlikely to find consistent corresponding points with one of the database images. To handle these problems, we propose a novel motion-based navigation method in contrast with appearance-based approaches. This algorithm is based on motion estimation by a camera to plan the next movement of a robot and robust feature matching to recognize home and destination locations. Experimental results demonstrate the capability of the vision-based autonomous navigation against environment changes.**

I. INTRODUCTION

An autonomous mobile robot requires a relation between the perception of the environment and a low-level robot operation. Vision sensors are especially efficient for this task in that they provide more information on scene interpretation than range scanning sensors such as LRF and sonar sensors. For this reason, many navigation methods adopt various vision sensors.

Autonomous navigation necessitates spatial reasoning that relies on some kind of internal environment representation.

In the model-based approach, the environment is represented by landmarks and descriptions of the corresponding image features. One of possible solutions for autonomous navigation involves map building using natural or artificial landmarks [1].

The alternative appearance-based approach employs a sensor-centered representation of the environment by sensor readings. For vision-based navigation methods, the representation usually contains a set of key images that are acquired during a learning state, and these images compose a graph. In [2], Yagi et al. proposed a new iconic memory-based navigation method that synthesized a corresponding image pattern from an omnidirectional route panorama (ORP) that can be acquired by arranging points on the horizontal plane for environment representation. Gaspar et al. [3] proposed a method for visual-based navigation of a mobile robot in indoor environments using a single omnidirectional camera. In this approach, a bird-eye view [4] was considered to simplify the solution for navigation problems. Chen and Birchfield [5] presented a simple algorithm for mobile robot navigation that only compares feature coordinates in the image sequence with those computed previously in the teaching step. Booij et al. [6] proposed a navigation method that used the appearance-based topological map and the estimated heading direction by feature correspondences between a current image and one of the database images in the topological map, thereby assisting a robot to navigate the destination. Another appearance-based navigation approach was proposed using image collections and an efficient image matching scheme by Fraundorfer [7]. In [8], Ŝegvić proposed a solution based on a hierarchical environment representation with a graph of key images at the top and local 3D reconstructions at the bottom.

All of the previous approaches only consider a well-organized environment that is not influenced by moving objects or environment changes. Some previous studies that cope with dynamic environments have been proposed in the robotics community [9][10][11]. In this paper, we present a novel vision-based autonomous navigation framework to handle problems including environment changes and moving objects; this method requires significantly less database images. We also adopt the teaching and replay strategy in which a robot is manually led through the path once during a teaching step; subsequently the robot follows the path autonomously during the replay step. This is a relevant technique for autonomous robot navigation.

This paper is organized as follows. As a prerequisite step for autonomous navigation, the details of the visual SLAM approach are described in Section II. In Section III, we introduce a novel motion-based navigation strategy and a robust feature matching method using the prediction-optimization approach to recognize home and destination locations. Section IV shows various experimental results on vision-based navigation under environment changes and moving objects to demonstrate feasibility of the proposed method. Finally, Section V concludes this paper.

II. TEACHING STEP

A. Feature Extraction

In each frame, we detect corner features [12] in the left image. Corner features have been found to give detections that are relatively stable under small to moderate image distortion and are identified by the intersection of two strong edges. By using MMX programming, we observe that the computational time for extracting corners from the 320×240 image is less than 5 *ms*.

In So Kweon is with Faculty of Electrical Engineering and Computer Science, KAIST, 373-1, Guseong-dong, Yuseong-gu, Daejeon, Korea iskweon@kaist.ac.kr

Jungho Kim and Yunsu Bok are with the Department of Electrical Engineering and Computer Science, KAIST, Korea jhkim@rcv.kaist.ac.kr, ysbok@rcv.kaist.ac.kr

978-1-4244-2058-2/08/$25.00 ©2008 IEEE.

B. Stereo Matching

From stereo matching of detected feature points, we can infer information on the 3-D structures and distances of a scene. If we assume that stereo images are rectified, then pairs of conjugate epipolar lines become collinear and parallel to one of the image axes, and this eases the stereo matching problem because it is reduced to a 1-D search on a trivially indentified scanline.

We use the modified KLT feature tracker to obtain correspondences between stereo images. Because the KLT tracker [13] gives the sub pixel locations of matched points, we can reconstruct more realistic structures, as shown in Fig. 1. To reduce the original KLT feature tracker that involves a 2-D search into a 1-D search along the horizontal line, we assume that a point in the left image I moves to point $x - d_{min} - d_x$ in the right image J, and we linearize $J(x - d_{min} - d_x)$ by Taylor expansion, as given in (1)

$$I(x,y) = J(x - d_{min}, y) - g_x d_x \tag{1}$$

Here, d_{min} is a minimum disparity threshold, and g_x is an intensity gradient along the horizontal axis. d_x is the disparity that minimizes the following dissimilarity defined by SSD in (2).

$$\varepsilon = \int\int_A [h(x,y) - g_x d_x]^2 w dA \tag{2}$$

Here, $h(x,y) = J(x - d_{min}, y) - I(x,y)$.
To find the disparity d_x, we set the derivative to zero

$$\frac{\partial \varepsilon}{\partial d_x} = \int\int_A [-2h(x,y)g_x + 2g_x^2 d_x] w dA = 0$$

Finally, d_x is computed by (3)

$$d_x = \frac{\int\int_A h(x,y) g_x w dA}{\int\int_A g_x^2 w dA} \tag{3}$$

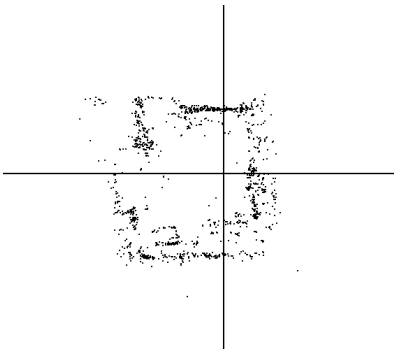
C. Motion Estimation and Map Generation

As a prerequisite for autonomous navigation, visual SLAM in the teaching step plays important roles including topological place recognition and navigation strategy. Among many vision-based SLAM approaches [14][15], Nister introduced a method that estimates the movement of a stereo head or a single camera in real time using only vision sensors [16]. The overall process based on Fig. 2 is as follows.

- Match feature points between left and right images of the stereo pair and triangulate the matched points into 3D points after camera calibration. The 3D points are stored in a local map as landmarks.

- Track features between the incoming left image and landmarks in the local map using the KLT feature tracker and estimate the robot pose using 3-point algorithm followed by RANSAC [17] until it satisfies the following criteria. In 3-point algorithm [18][19], the images of three known world points provide the possible camera poses of up to four solutions and more than three points are required to obtain one solution automatically.

(a) Map building using NCC stereo matching



(b) Map building using modified KLT stereo matching

Fig. 1. Map building comparison

  - The number of inliers after RANSAC is more than the predefined number.
  - The variance of tracked features on an incoming image is above a variance threshold to ensure the large field of view.
- If it does not satisfy the above criteria, store the current left image in the database as a key-frame image used for autonomous navigation. Subsequently, all the landmarks in the local map are stored in the global map, and generate a new local map from step 1.

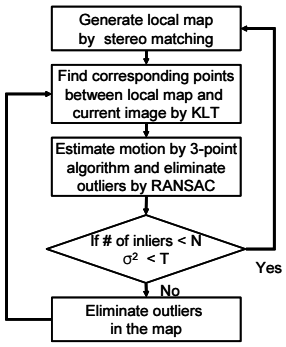


Fig. 2. Flow chart of visual SLAM

The key-frame images are stored in a database according to the number of corresponding corners between an incoming

image and the previous key-frame image.

III. Replay Step

A. Navigation Strategy

Two geometrical views, which are extracted from key-frame images in the database and incoming images, provide inference on the next movement of a robot to reach the destination. For home to destination locations, the epipole between two views can provide good reasoning, i.e., if a camera faces a destination point, the epipole must be located near a principal point in the image, as shown in Fig. 2. Therefore, a robot automatically changes its operation based on the epipole coordinates.

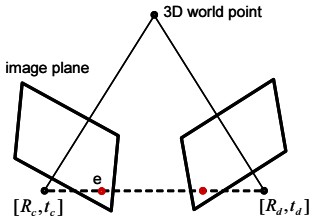


Fig. 3. Epipole coordinate between two views

The fundamental matrix is the algebraic representation of epipolar geometry [20]. Suppose we have two images captured by camera with non-coincide centers, then the fundamental matrix F is a unique 3×3 rank 2 homogenous matrix that satisfies (4)

$$x'^T F x = 0 \tag{4}$$

The epipole is the point of intersection of the line joining the camera centers (the baseline) with the image plane. Equivalently, the epipole is the image in one view of the camera center of the other view. The epipole is a null vector of the fundamental matrix F, as given in (5).

$$Fe = 0 \tag{5}$$

When a camera is located at the destination location, the centers of the two views coincide. Therefore, the epipole cannot be computed by the fundamental matrix, and the fundamental matrix requires minimal 8 true correspondences. However, the epipole is simply a re-projected point of one camera center into another image plane. Because we know camera poses of two views, we compute the epipole in the alternative way, as given in (6) [20].

$$e = K[R_c \quad t_c]X_d \tag{6}$$

Here, X_d is the camera center of the destination location and R_c and t_c represent the current camera pose.

If the x-coordinate of the epipole e_u is located at the right side of the principal point fc_x, then a robot turns right, as shown in Fig. 4. We can summarize the navigation strategy,

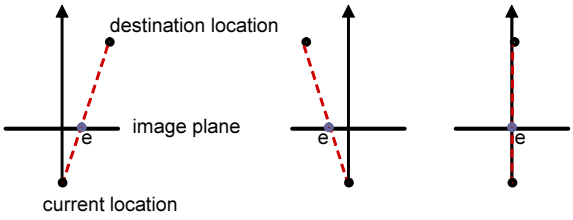


Fig. 4. Navigation strategy based on epipole coordinate

as stated in (7) and (9). If a camera is not close to the destination location, a robot moves according to (7).

$$\begin{array}{lll} if \quad e_u > fc_x + T & : & turn \quad right \\ else \quad if \quad e_u < fc_x - T & : & turn \quad left \\ else & : & go \quad straight \end{array} \tag{7}$$

Here, T is a positive value.

To determine whether a robot reaches a destination location, we compare the locations of the robot using (8).

$$d = |R_d^T t_d - R_c^T t_c| \tag{8}$$

Here, $-R_d^T t_d$ reprsents the destination location with respect to the global coordinate and $-R_c^T t_c$ is computed by the current camera pose.

When a robot reaches the destination location, we adjust the robot orientation by feature matching to satisfy the relation given in (9). To find corresponding points between a current image and a corresponding database image, we use a feature matching method that utilizes a camera pose and 3D structures introduced in the next section.

$$\begin{array}{lll} if \quad \hat{x}_c > \hat{x}_d + T_a & : & turn \quad left \\ else \quad if \quad \hat{x}_c + T_a < \hat{x}_d & : & turn \quad right \\ else & : & stop \end{array} \tag{9}$$

Here, T_a is a marginal angle and $\hat{x}_d$ and $\hat{x}_c$ are means of x coordinates among correspondences from destination and current images, respectively.

When a robot stops according to the conditions stated in (9), we regard this operation as home to the destination location and designate the next key image as the destination image. To reduce the pose error, we recompute the camera pose with 3D coordinates in the given map and their corresponding images obtained by topological place recognition using the 3-point algorithm.

B. Topological Place Recognition by Feature Matching

Topological place recognition is the task of deciding whether a robot has returned to a previously visited area in the teaching step. Invariant features such as SIFT [21] and maximally stable regions [22] are designed for this purpose. However, heavy computational complexity prevents them from real-time applications and full invariance of descriptors is futile, i.e, in that we can utilize a camera pose in the replay step and 3D structures which were computed in the teaching step. By using a camera pose and 3D locations of

world points, we can predict possible corresponding points that near true corresponding points from (10).

$$\hat{x} = K[R_c \quad t_c]X \tag{10}$$

Here, $\hat{x} = [u_p \quad v_p \quad 1]^T$ is a re-projected coordinate of 3D world point $X = [X \quad Y \quad Z \quad 1]^T$ according to the current pose, R_c, t_c.

Starting from the predicted points, the KLT feature tracker provides robust corresponding points without distinctive features and their descriptors.

$$I(u_p - d_x, v_p - d_y) = I(u_p, v_p) - [g_x \quad g_y][d_x \quad d_y]^T. \tag{11}$$

By using re-projected points, we considerably reduce d_x and d_y , thus consequently, satisfy the approximation of first-order Taylor expansion in (11).

$$\varepsilon = \int\int_A [h(x,y) - g_x d_x - g_y d_y]^2 w dA \tag{12}$$
$$Here, \quad h(x,y) = I(u_p, v_p) - J(x,y)$$

g_x and g_y represent the intensity gradients along x and y directions, respectively. In addition, I and J represent a key image and a current image, respectively. From (14), we finally compute displacements that minimize SSD (sum of squared difference) defined in (12).

$$\frac{\partial \varepsilon}{\partial d_x} = \int\int_A [-2h(x,y)g_x + 2g_x^2 d_x + 2g_x g_y d_y] w dA = 0$$
$$\frac{\partial \varepsilon}{\partial d_y} = \int\int_A [-2h(x,y)g_y + 2g_y^2 d_y + 2g_x g_y d_x] w dA = 0 \tag{13}$$

$$d = Z^{-1} e \tag{14}$$

Here $e = \begin{bmatrix} \int\int_A [h(x,y)g_x] w dA \\ \int\int_A [h(x,y)g_y] w dA \end{bmatrix}$, $d = \begin{bmatrix} d_x \\ d_y \end{bmatrix}$ and $Z = \begin{bmatrix} \int\int_A g_x^2 w dA & \int\int_A g_x g_y w dA \\ \int\int_A g_x g_y w dA & \int\int_A g_y^2 w dA \end{bmatrix}$

Fig. 5 shows the feature matching results from the proposed method under large image deformations as a role of topological place recognition.

C. Topological Place Recognition by Motion Estimation

Topological place recognition is likely to fail when an environment changes or when objects move in an environment because a robot has difficulty in deciding whether it reaches the destination location by feature matching. If an estimated pose has a large difference in the location of a destination node due to false matches or if we cannot find sufficient correspondences with a key-frame image, we recognize a topological node by comparing a current pose and a key-frame node. This signifies that we compare not only the heading directions of a locally estimated pose and a key-frame pose (θ_g, θ_l in Fig. 6) but also their locations, as defined in (8).

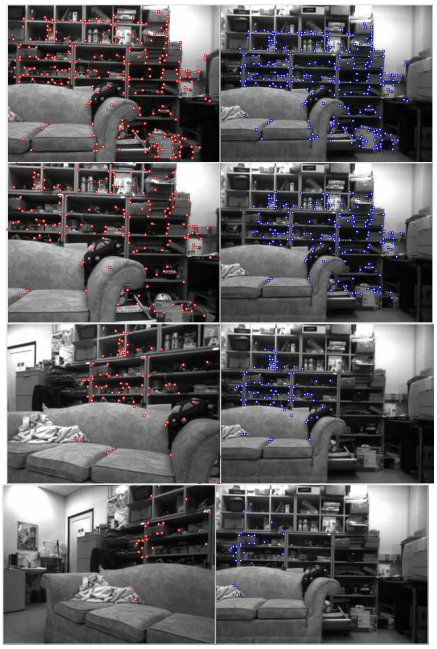


Fig. 5. Feature tracking for topological node recognition

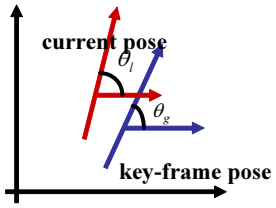


Fig. 6. Node recognition by motion

IV. EXPERIMENTAL RESULTS

The proposed algorithm was implemented on a laptop (1.6 GHz CPU and 1GB RAM) that controlled a Pioneer 2 mobile robot mounted with a Bumblebee stereo camera.

A. Navigation Under Normal Environment

Fig. 7 shows the visual SLAM result in real time approximately 10 frames/s after a robot moved approximately 20 m in an indoor environment, as a prerequisite for an autonomous mobile robot navigation.

Here, black points represent the map at the top-down view and blue circles represent the locations of a camera where key images were obtained. 39 key-frame images were automatically selected among 2500 frames. If there exist abundant visual features, two consecutive key-frame images might have large image deformations and narrow or homogenous regions have more key-frame images, as shown in Fig. 7. Some approaches require many key-frame images that have significantly overlapped regions and less image variations to find the correspondences with incoming images. However, for the proposed approach, only a few key-frame images are required because we can compute correspondences using a robot pose under large image variations.

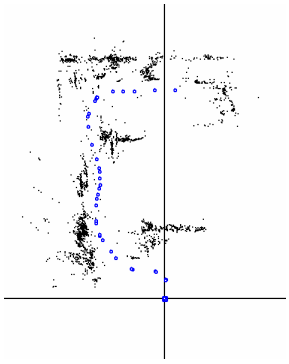


Fig. 7. Visual SLAM result

Fig. 8 reveals the navigation path computed in the replay step. Comparing with key-frame locations, a robot can autonomously follow the path that it followed during the teaching step. Fig. 9 shows some figures of autonomous navigation.

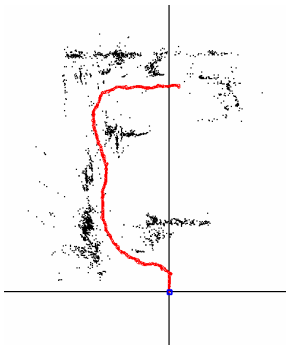


Fig. 8. Navigation path in a static environment

B. Navigation Under Environment Changes and Dynamic Environment

To evaluate the efficiency of the proposed method, a robot was manually driven in an environment during the teaching step when there were no people. After a few hours, when some people arrived at the office environment and some objects were moved, the robot autonomously navigated through the environment.

Fig. 10 shows some of the key-frame images automatically stored from the office environment in the teaching step, and fig. 12 shows the generated global map and key-frame locations. Fig. 13 depicts the navigation path under environment changes.

When a camera is disturbed by moving objects, some parts of the images are occluded; a robot is unlikely to have robust correspondences between an incoming image and some database images. However, from the proposed method, in spite of visual occlusion, a robot can estimate its pose by fast motion estimation, approximately 10 frames/s, introduced in section II, and the robot determines its operation based on its pose when it cannot match with the database images.

Fig. 9. Some figures in a video

Fig. 11 shows the navigation results when there exist moving objects and environment changes.

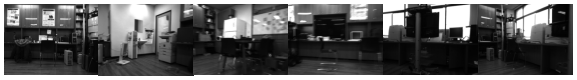


Fig. 10. Key-frame images obtained when there are no people

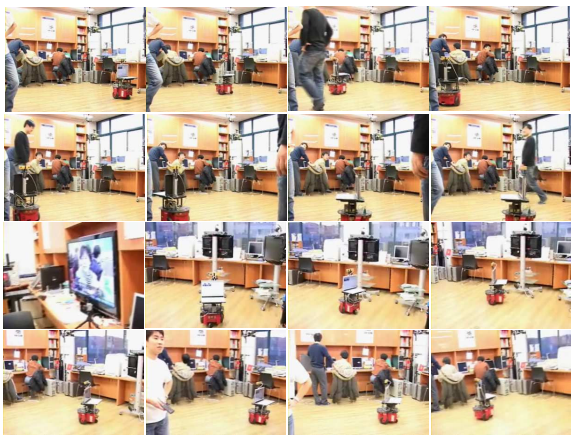


Fig. 11. Navigation under environment changes and moving objects

C. Long-distance Navigation

Fig. 15 depicts the path from long-distance navigation of approximately 60 *m* in the first floor of the EE building in KAIST and some image sequences are shown in Fig. 14.

V. CONCLUSIONS

We have presented a robust navigation strategy under environment changes and moving objects. In this paper, we

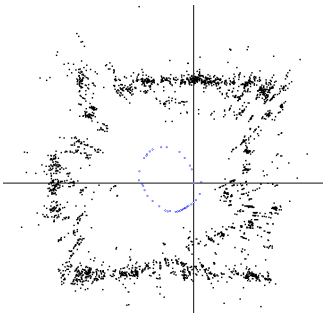


Fig. 12. Key-frame locations overlapped with the map

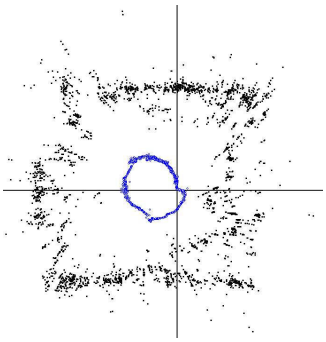


Fig. 13. Navigation path in a dynamic environment

have proposed an autonomous navigation method combined with visual SLAM and a motion-based navigation strategy to overcome the failure of image matching. Compared to appearance-based navigation methods, this proposed approach is more robust to environment variations and moving objects. We demonstrate the feasibility of the proposed method through the various navigation experiments.

VI. ACKNOWLEDGMENTS

This work was supported in part by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea goverment (MOST) (No. M1-0302-00-0064).

REFERENCES

[1] D.Burschka and G.D.Hager, Vision-based control of mobile robots, IEEE International Conference on Robotics and Automation, 2001.
[2] Yasushi Yagi, Kousuke Imai, Kentaro Tsuji and Masahiko Yachida, Iconic Memory-Based Omnidirectional Route Panorama Navigation, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.27, No.1, 2006.
[3] José Gaspar, Niall Winters and José Santos-Victor, Vision-based Navigation and Environmental Representations With an Omni-directional Caemra, IEEE Transactions on Robotics and Automation, Vol.16, No.6, 2000.

Fig. 14. Image sequences for long-distance navigation

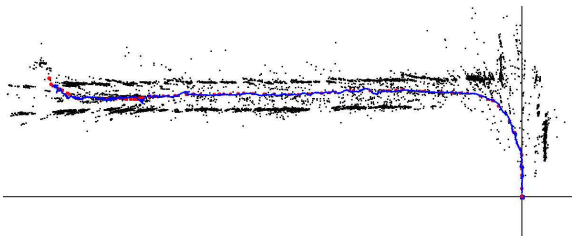


Fig. 15. Long-distance navigation path

[4] J.S.Chahl and M.V.Srinivasan, Reflective surfaces for panoramic imaging, Applied Optics, Vol.36, No.31, 1997.
[5] Zhichao Chen and Stanley T.Birchfield, Qualitative Vision-based Mobile Robot Navigation, IEEE International Conference on Robotics and Automation, 2006.
[6] O. Booij, B. Terwijn, Z. Zovkovic and B. Krőse, Navigation using an appearance based topological map, IEEE International Conference on Robotics and Automation, 2007.
[7] Friedrich Faundorfer, Christopher Engles and David Nistér, Topological mapping, localization and navigation using image collections, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007.
[8] Sinîša Ŝegvić, Anthony Remazeilles, Albert Diosi and François Chaumette, Large scale vision-based navigation without an accurate global reconstruction, IEEE Conference on Computer Vision and Pattern Recognition, 2007.
[9] M.Cummins and P.Newman, Probabilistic appearance based navigation and loop closing, IEEE International Conference on Robotics and Automation, 2007.
[10] Denis F.Wolf and Gaurav S.Sukhatme, Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments, Autonomous Robots, Vol.19, No.1, 2005.
[11] Chieh-Chih Wang, Charles Thorpe and Sebastian Thrun, Online Simultaneous Localization and Mapping with Detection and Tracking of Moving Objects, IEEE International Conference on Robotics and Automation, 2003.
[12] C. Harris and M.J.Stephen, A combined corner and edge detector, In Alvey Vision Conference, page 147-152, 1988.
[13] Jianbo Shi and Carlo Tomasi, Good Features to Track, IEEE Conference on Computer Vision and Pattern Recognition, 1994.
[14] Andrew J. Davison, "Real-Time Simultaneous Localisation and Mapping with a Single Camera", IEEE International Conference on Computer Vision, 2003.
[15] Chris McCarthy and Nick Barnes, Performance of Optical Flow Techniques for Indoor Navigation with a Mobile Robot, IEEE International Conference on Robotics and Automation, 2004.
[16] David Nister, Oleg Naroditsky and James Bergen, Visual Odometry, IEEE Society Conference on Computer Vision and Pattern Recognition, 2004.
[17] M.Fischler and R.Bolles, Random Sample Consensus : a Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography, Communications ACM, 24:381-395. 1981.
[18] R.Haralick, C.Lee, K.Ottenberg and M.Nölle, Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem, International Journal of Computer Vision, 13(3):331-356, 1994.
[19] David Nister, A Minimal Solution to the Generalised 3-Point Pose Problem, IEEE Conference on Computer Vision and Pattern Recognition, 2004.
[20] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, ISBN 0-521-62304-9, 2000.
[21] D.Lowe, Distinctive image features from scale-invariant Keypoints, International Journal of Computer Vision, Vol.60, No.2, 2004.
[22] J.Matas, O.Chum, U.Martin, and T.Pajdla, Robust wide baseline stereo from maximally stable extremal regions, British Machine Vision Conference, 2002.