Online Scene Modeling for Interactive AR Applications

Jaesang Yoo, Kyusung Cho, Jinki Jung, Hyun S. Yang

Online Scene Modeling for Interactive AR applications

Jaesang Yoo[1] , Kyusung Cho[2], Jinki Jung[2], and Hyun S. Yang[2]

[1] Electronics and Telecommunications Research Institute, 138 Gajeongno, Yuseong-gu, Daejeon, 305-700, Republic of Korea
jsyoo@etri.re.kr
[2] Department of Computer Science, Korea Advance Institute of Science and Technology, 373-1 Guseong-dong, Yuseong-gu, Daejeon, 305-701, Republic of Korea
{qtboy, jk, hsyang}@paradise.kaist.ac.kr

Abstract. Augmented reality applications require 3D model of environment to provide even more realistic experience. Unfortunately, however, most of researches on 3D modeling have been restricted to an offline process up to now, which conflicts with characteristics of AR such as realtime and online experience. In addition, it is barely possible not only to generate 3D model of whole world in advance but also trasfer the burden of 3D model generation to a user, which limits the usability of AR. Thus, it is required to draw the 3D model generation to an online stage from an offline stage. In this paper, we propose an online scene modeling method to generate 3D model of a scene, based on the keyframe-based SLAM which supports AR experience even in an unknown scene by generating a map of 3D points. The scene modeling process in this paper is a little computationally expensive in terms of real-time but it doesn't restrict real-time property of AR because it is executed on a background process. Therefore, a user will be provided with an interactive AR applications that support interactions between the real and virtual world even in an unknown environment.

Keywords: Keyframe-based SLAM, higher level structure, online modeling, augmented reality

1 Introduction

Recently Augmented Reality (AR) technology is gaining attention by the spread of mobile devices. The application of AR is expanding to Xbox Project Natal, PS3, and Nintendo DSi beyond the mobile platforms. By broadening the environment for AR applications, the importance of AR contents is getting bigger.

It is still difficult to create AR contents in spite of a growth on development tools and circumstances for AR applications. Simple applications such as Layar 2.0 and Sekai camera to develop AR contents have been released. However, it is hardly possible to build 3D models of arbitrary unknown scenes for AR applications and moreover it takes a lot of time to build a 3D model of a scene. Thus, we propose a method of an online scene modeling for interactive AR applications in real-time.

Recently many researches related to 3D modeling for interactive AR contents have been published. Bergig et al.'s work [1] has presented a 3D model authoring tool by

hand sketching using normal pen and paper. Moreover Hengel et al.'s work [2] has presented a tool to easily generate 3D models from a video clip in which objects for modeling are appear. However, [1] has limitation because they are only capable of simple modeling. Also [2] is not fully online: the user should build models in offline.

In this paper, we propose an online scene modeling method to generate 3D model of an environment for interactive AR applications with minimum interaction with a user. The system requires a user to do nothing but moving the camera to capture the scene of which the user wants to generate a 3D model. This paper focused on an online scene modeling to recognize geometry of a scene and incorporate it into the system for interactive AR applications.

2 Related Work

2.1 Keyframe-based SLAM

Simultaneous Localization and Mapping was begun in Robotics to localize a robot in an unknown environment. The early days, it used LRF and sonar sensors[4,5] but recently, vision sensor was also used [6,7]. The first stage of vision-based SLAM, only the sensor was substituted to a camera from LRF or sonar, keeping the overall framework of SLAM using probabilistic models such as EKF(Extended Kalman Filter) and particle filter. Davison et al.'s work [6] was based on EKF and constructed a map of 3D sparse points so that it contained uncertainties on the camera's position by nature. Later, PTAM(Parallel Tracking and Mapping) so-called keyframe-based SLAM appeared [7]. To localize a moving camera in an unknown scene, tracking and mapping are separated and run in two parallel threads. PTAM constructs a map of 3D points based on keyframes collected during tracking process using batch techniques, Bundle Adjustment and the map can contain thousands of points, which allows precise camera localization more proper for AR applications.

2.2 Scene Modeling

The detection of higher level structure in a real world has been progressed using both visual and non-visual sensors. Planes were fitted to clouds of points reconstructed using LRF(Laser Ranging Finder)[8,9]. The discovered planes were, however, not incorporated into a map.

Afterward, many have attempted to discover planes in vision-based SLAM as in [10,11,12]. Rachmielowski et al.'s work [10] attempted to get 3D reconstruction of a scene during SLAM tracking. They connected sparse 3D points to compose a mesh using a Delaunay triangulation and reconstructed a scene model with the meshes. Thus, the reconstruction is unlikely a good estimation of the real surface because the 3D points used for a mesh were sparse and involved uncertainties by nature of the SLAM approach. Chekhlov et al.'s work [11] also tried 3D reconstruction of a scene during SLAM tracking. However, to find planes they approached with RANSAC process instead of the Delaunay triangulation used in [10]. In view of using RANSAC

process, our system is similar to their system but we go beyond their work by incorporating the discovered planes into the map and using them during the tracking process.

3 Online Plane Fitting for Interactive AR

A map of PTAM usually consists of thousands of 3D points and it is impossible to consider the all points to generate a plane hypothesis. Thus, we face a problem of a working set selection, which will be discussed in 3.1. A method how to discover a planar structure in a scene will be presented in 3.2 and how to incorporate the discovered plane into a map will be presented in 3.3.

3.1 Region of Interest

Because considering all 3D points results in an inaccurate result as well as takes a lot of time when a plane hypothesis is generated, the points for the plane detection need to be classified. We call the classified region as ROI(region of interest).
In Hengel et al.'s work [13], a user selected the ROI by dragging a mouse. This was possible in their system because it worked in offline, not online. In our system, a user can select a working set by moving a camera view to the direction in which the user wants to generate a model, which is even more proper process for an online modeling system.

3.2 Detection of Planar Structure

In this section, it is introduced how to discover planes existing in real world. At first, it is described how to discover a plane from a 3D point cloud within ROI(region of interest) using RANSAC process in 3.2.1. However, it doesn't guarantee the discovered plane using RANSAC is a real plane in the real world because RANSAC process only considers a geometric distribution of 3D points. We call the discovered plane at this stage as a geometric consistent plane. Therefore, it is required a process to verify whether or not the generated plane exists in reality. This verification process is introduced in section 3.2.2 by testing if the geometric consistent plane is also photometric consistent.

3.2.1 Plane hypothesis using RANSAC

To discover a potential planar structure using RANSAC process, a similar approach to that used in [12] is used in this paper.

Step 1 : RANSAC robust estimation
A minimum of three 3D points is required to generate a plane hypothesis. Assuming X_1, X_2, X_3 are three 3D points, an origin p_0 and a normal vector n of the plane hypothesis π including those three points can be generated as followings.

$$p_0 = X_1 \quad n = c_1 \times c_2 \qquad (3.1)$$
$$\text{where } c_1 = X_2 - X_1 \quad c_2 = X_3 - X_1$$

As soon as a plane hypothesis is generated with the minimum three points, consensus for the hypothesis among other points is determined using a perpendicular distance from the plane. The perpendicular distance $d_\perp$ for a 3D point X_i from the plane $\pi[p_0, n]$ can be calculated as a following.

$$d_\perp = (X_i - p_0) \cdot n \qquad (3.2)$$

To minimize effects of the ouliers, we impose a criterion that the distance must be less than or equal to some threshold d_{max}, that is, although $d_\perp > d_{max}$, we set $d_\perp = d_{max}$.

Step 2 : Optimal estimation

To compute the best-fit plane from inliers S_i resulting from RANSAC process, we use a principal component analysis approach similar to [3].

The origin of the plane hypothesis is set to the mean of the inliers and normal vector is set to the eigenvector of $M^T M$ corresponding to the smallest eigenvalue where M is a $l \times 3$ matrix stacked with position vectors for l inlying points w.r.t the mean. The smallest eigenvalue gives a measure of quality of the fit because it is the variance of the inliers in the normal direction.

3.2.2 Photo-consistent plane

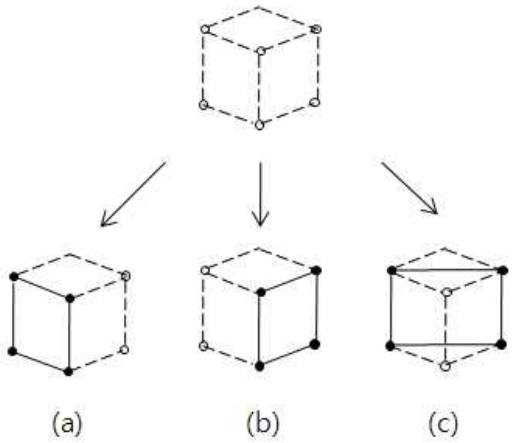

Fig 1. Possibly detectable planes solely considered a geometric distribution of 3D points. (c) is false detection

So far, given a map of point clouds, we introduced how to discover a plane using RANSAC. The mainly considered property is the geometric distribution of the 3D

points so that it is often likely to result in false detections of planes as shown in (c) of Fig 1. Therefore, a verification process is required to remove the false detections and the process called photo-consistency test is described below in details and summarized in algorithm 1.

Step 1 : Finding the visibility images $\mathcal{V}$

Given a set of keyframes I and a geometric consistent plane with geometric supports Π belonging to the plane, project 3D points within Π into every keyframe within I. If over 70 percent of overall points within Π appear on the keyframe, it is said that the plane is "visible" in the keyframe. To find a visibility image set $\mathcal{V}$, test if the plane is visible every keyframe.

$$\mathcal{V} \leftarrow \{i \mid \text{Ratio}(q, \Pi) > 0.7\} \quad \text{for } q \in \{\hat{q}_{i,j} \mid \hat{q}_{i,j} = P_i Q_j \quad \text{for } \forall\, i \in I, j \in \Pi\} \tag{3.3}$$

Step 2 : Computing the photo-consistency κ

Among visibility images $\mathcal{V}$, choose reference image I_{i_0} in which a plane appears dominant.

$$i_0 \leftarrow \text{argmax}_{i \in \mathcal{V}}\big(\text{Ratio}(q, \Pi)\big) \quad \text{for } q \in \{\hat{q}_{i,j} \mid \hat{q}_{i,j} = P_i Q_j \quad \text{for } \forall\, i \in I, j \in \Pi\} \tag{3.4}$$

Using 2D point correspondences between reference image I_{i_0} and the others in $\mathcal{V}$, compute homography $\text{H}_{i_0,i}$ satisfying the following relation.

$$\hat{q}_{i_0,j} \approx \text{H}_{i_0,i} \hat{q}_{i,j} \quad \text{for } \forall\, i \in \mathcal{V} \tag{3.5}$$

Once homography $\text{H}_{i_0,i}$ is computed, warp all images except reference one to the reference one.

$$\tilde{I}_i[q] \leftarrow I_i\big[\text{H}_{i_0,i}(q)\big] \tag{3.6}$$

$\tilde{I}_i$ is a warped image of image I_i to the reference one and q is a set of 2D points projected into an image I_i.
Ideally the warped image $\tilde{I}_i$ is identical to the reference image I_i However, because they are not identical due to image noises, we find the most suitable match in terms of a pixel value within radius r. Therefore, the photometric consistency score κ is induced by square root of the photometric error between the warped image and the reference one as a following.

$$\kappa^2(\pi) = \frac{1}{\#(\mathcal{V}-1)\#(q)} \sum_{i \in \mathcal{V}, i \neq i_0} \left(\sum_q \left(\min_{q', \|q'-q\| \leq r} \left| I_{i_0}[q] - \tilde{I}_i[q'] \right| \right)^2 \right) \tag{3.7}$$

If the photometric consistency score κ is greater than some threshold ε, reject the plane because it is not photometric consistent, that is, not exist in real world.

OBJECTIVE

Compute a photometric consistency κ for the plane with geometric support Π given the cloud of reconstructed 3D points Q_j, keyframe images I_i and cameras P_i and reject the plane if $\kappa(I_0) > \varepsilon$.

ALGORITHM

Step 1: Find the visibility images $\mathcal{V}$.

$$\mathcal{V} \leftarrow \{i \mid \text{Ratio}(q, \Pi) > 0.7\} \text{ for } q \in \{\hat{q}_{i,j} \mid \hat{q}_{i,j} = P_i Q_j \text{ for } \forall\, i \in I, j \in \Pi\}$$

where $\text{Ratio}(q, \Pi)$ is the ratio of the visibility measurements q to $\#(\Pi)$

Step 2: Compute the photometric consistency κ.

- Choose reference image i_0 :

$$i_0 \leftarrow \text{argmax}_{i \in \mathcal{V}}\big(\text{Ratio}(q, \Pi)\big)$$
$$\text{for } q \in \{\hat{q}_{i,j} \mid \hat{q}_{i,j} = P_i Q_j \text{ for } \forall\, i \in I, j \in \Pi\}$$

- Compute a homography $\text{H}_{i_0,i}$ induced by the point correspondences $\hat{q}_{i_0,j} \leftrightarrow \hat{q}_{i,j}$ such that

$$\hat{q}_{i_0,j} \approx \text{H}_{i_0,i} \hat{q}_{i,j} \text{ for } \forall\, i \in \mathcal{V}$$

- Warp all images I_i for $i \in \mathcal{V}$ and $i \neq i_0$ to the reference one :

$$\tilde{I}_i[q] \leftarrow I_i\big[\text{H}_{i_0,i}(q)\big] \text{ for } q \in \{\hat{q}_{i,j} \mid \hat{q}_{i,j} = P_i Q_j \text{ for } \forall\, i \in I, j \in \Pi\}$$

- Reject the plane if the consistency check fails, $i.e.$ if

$$\kappa(I_0) > \varepsilon,$$

where the consistency $\kappa(I_0)$ is given by

$$\kappa^2(I_0) = \frac{1}{\#(\mathcal{V}-1)\#(q)} \sum_{i \in \mathcal{V}, i \neq i_0} \left(\sum_{q \in I_0} \left(\min_{q', \|q'-q\| \leq r} \left| I_{i0}[q] - \tilde{I}_i[q'] \right| \right)^2 \right)$$

Algorithm 1. Photo-consistency test

3.3 Insertion of Planar Features

This section describes how to incorporate the discovered plane in 3.2 into the map of PTAM. The parameterization of a plane, addition of new point using a plane, and finally the way to reduce the computation using a plane are discussed in order.

3.3.1 Parameterization

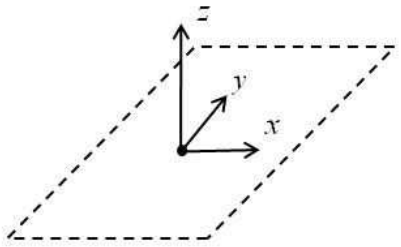

Fig 2. Local plane coordinate $\boldsymbol{\mathcal{E}}$

To represent a plane, a center position and normal vector are necessary. They are 3 DOF each, and thus, a plane has total 6 DOF. This is over-parameterized into a transformation matrix $\boldsymbol{E_{W\mathcal{E}}}$ from the local plane coordinate $\boldsymbol{\mathcal{E}}$ to the world coordinate $\boldsymbol{\mathcal{W}}$ with 12 internal parameters as in. A center of a plane is located on an origin and the normal vector is parallel to the z axis as in Fig 2. We set the x and y axis of the local plane coordinate parallel to the x and y axis of the world coordinate.

$$E_{W\mathcal{E}} = \begin{bmatrix} R & t \\ \mathbf{0} & 1 \end{bmatrix} \tag{3.8}$$

3.3.2 New mappoint addition using a plane

To reconstruct 3D points, triangulation from correspondences between two views is used in PTAM. The number of iterations and final converging position of a bundle adjustment depend on the initial position of the reconstructed point. That is, the more accurate the initial position of the reconstructed point, the more accurate the adjusted position after the bundle adjustment as well as the faster the converging speed. Therefore we propose a method taking advantage of the plane information, homography, when a point is reconstructed as shown in Fig 3.

1. Find the closest keyframe to the new keyframe among keyframes in a map and set it as a target.
2. Find matching correspondences between new and target keyframes.
3. Compute a homography H_1 from the matching correspondences.
4. Compute H_2 from 2D point correspondences between the target keyframe and a plane coordinate.
5. With H_1 and H_2, the position on the plane coordinate p_{new} of a new point x_{new} is determined as below.
$$p_{new} = H_2 H_1 x_{new} \tag{3.12}$$
6. A 3D point on a plane X_{new} is reconstructed from a 2D point on a plane coordinate p_{new} using transformation matrix $E_{W\mathcal{E}}$ from the plane

coordinate to the world coordinate.

$$X_{new} = E_{W\mathcal{E}} P_{new} \quad (3.9)$$

$$where \ p_{new} = (x, y)^{\mathrm{T}} \text{ and } P_{new} = (x, y, 0, 1)^{\mathrm{T}}$$

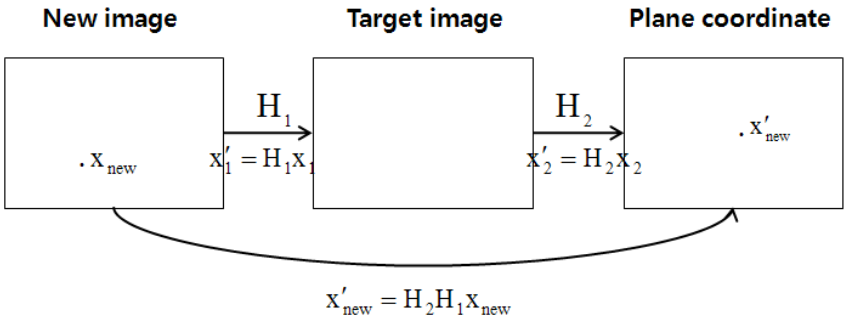

Fig 3. A 3D point reconstruction using homography given a plane

3.3.3 Selective measurement

Without any prior knowledge of a scene, all mappoints are used to estimate the pose of the camera. To be frank, thousands of 3D points are not necessary and a few hundred are enough to estimate the pose of the camera, using planes. As a result, it is likely to save the computational time by knowing the scene information and actually it is as showed in the experiment below.

4 Experiment

Fig 4. Man-made environment with three orthogonal planes

For the experiment, we use a laptop of 2.83 GHz Quad CPU with 2GB memory. Logitech Ultra webcam is attached to the laptop to obtain a 640×480 image.

Experiments were performed in a man-made environment in which three orthogonal planes appeared with a lot of textures as in Fig 4.

Fig 5 shows how this system discovers and incorporates plane structures in a scene into PTAM map. Points in white indicate 3D reconstructed points from stereo analysis and colored points indicate points that belong to the discovered plane by method in this paper. Different color shows different planes. As a result three orthogonal planes are discovered and incorporated into the map.

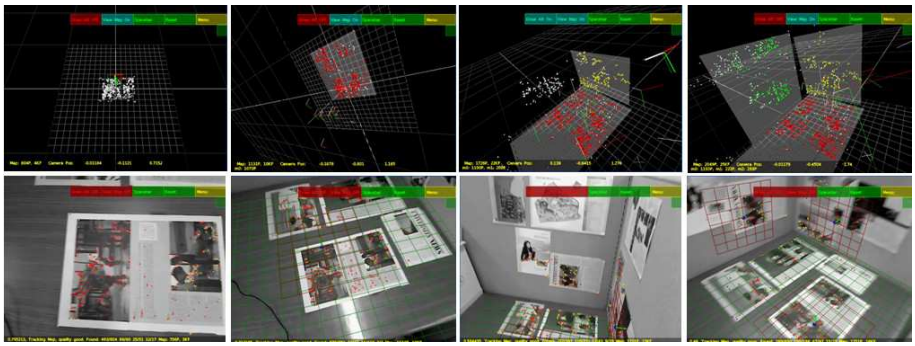

Fig 5. Discovery of a scene geometry (left to right)

Fig 6 shows changes in a matching rate and time consumption as the number of measurements per a plane varies from 50 to 200 with video clip of 1800 frames. As a result, the test shows that tracking with higher level structures like a plane can potentially reduce the time consumption compared with the tracking only with points, resulting in similar tracking accuracy. As an example, with 150 points per a plane, tracking time consumed to track a camera in same video clip decreased 13ms to 9ms, which meant 30% reduction in time.

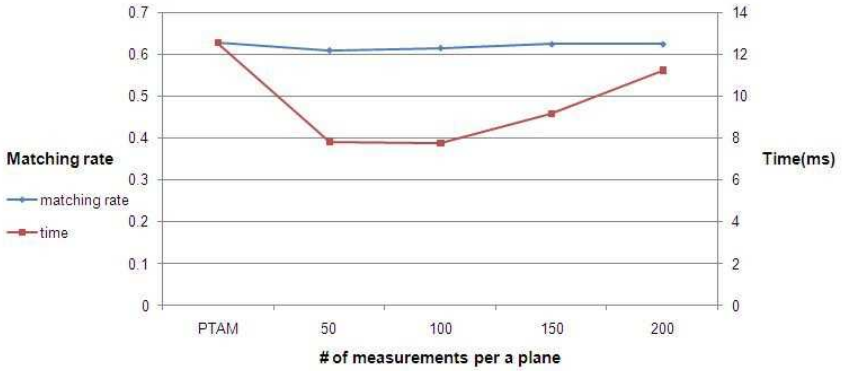

Fig 6. Changes in a matching rate(blue, left axis) and tracking time(red, right axis) as the number of measurements per a plane varies

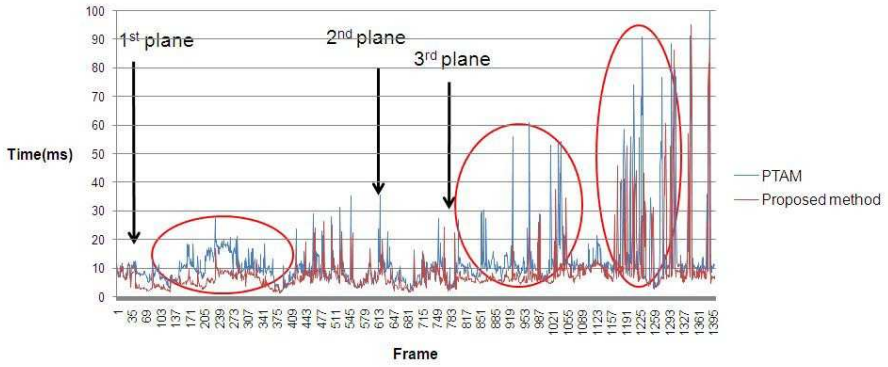

Fig 7. Tracking time comparison of PTAM and proposed method of this paper on an identical video clip of 1400 frames

To compare proposed method with PTAM in terms of overall performance in time consumption, we plotted the tracking time of the PTAM(blue line) and proposed method(red line) on an identical video clip of 1400 frames in Fig 7. 150 points per a plane was chosen in the test and first, second, and third planes were discovered and incorporated into a map on 35th, 620th, and 780th frame. On the beginning of the test before 35 frames when first plane was discovered and incorporated into a map, both showed similar time consumption. However, since 35th frame, it showed the proposed method was superior to PTAM in terms of time consumption. The frames within red circles in Fig 7 indicated such superiority.

Fig 8 shows a simple interactive AR application that interacts with a real world. A white ball is a virtual gravity ball and it interacts with a scene, two walls and ground, by rebounding when it collides with them. A full video clip is attached with this paper.

Fig 8. A virtual gravity ball rebounding from real ground and walls

5 Conclusion and Future Work

For interactive AR applications, this paper presented a method to discover and incorporate a higher level structure, a plane from point clouds on online. However, a plane is just a little bit higher level structure than a point cloud. There are a lot higher level structures than a plane in our environment where AR applications will set up. Therefore, for the future work, we need to discover and incorporate other higher level structures such as sphere, cube, cylinder, and polyhedral to describe a real world environment so that ultimately interactive AR application can be achieved. As a result of the future work, a real interaction not only with a plane such as a wall and ground but with objects such as a box, ball, cup and any other objects existing in real world can come true.

6 Acknowledgement

"This research is supported by the Ubiquitous Computing and Network (UCN) Project, Knowledge and Economy Frontier R&D Program of the Ministry of Knowledge Economy (MKE) in Korea and a result of subproject UCN 10C2-J2-11T"

References

1. Bergig, O., Hagbi, N., El-Sana, J., Billinghurst, M.: In-Place 3D Sketching for Authoring Mechanical Systems. 8th IEEE International Symposium on Mixed and Augmented Reality, pp. 87-94 (2009)
2. Hengel, A.V.D., Hill, R., Ward, W., Dick, A.: In Situ Image-based Modeling. 8th IEEE International Symposium on Mixed and Augmented Reality, pp. 107-110 (2009)
3. Weingarten, J., Gruener, G., Siegwart, R.: Probabilistic plane fitting in 3D and an application to robotic mapping. IEEE International Conference on Robotics and Automation, pp. 927-932 (2004)
4. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM : A Factored Solution to Simultaneous Localization and Mapping Problem. National Conference on Artificial Intelligence, (2002)
5. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. International Joint Conference on Artificial Intelligence, (2003)
6. Davison, A., Reid I., Molton, N. D., Stasse, O.: MonoSLAM : Real-time single camera SLAM. IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 29, No. 6, pp. 1052-1067 (2007)
7. Klein, G., Murray, D.: Parallel Tracking and Mapping for Small AR Workspaces. 6th IEEE International Symposium on Mixed and Augmented Reality, pp. 225-234 (2007)
8. Nevado, M. M., García-Bermejo J. G., Zalama, E.: Obtaining 3D models of indoor environments with a mobile robot by estimating local surface directions. Robotics and Autonomous Systems, vol. 48, no. 2-3, pp. 131-143 (2004)
9. Viejo, D., Cazorla, M.: 3D plane-based egomotion for SLAM on semi-structured environment. IEEE International Conference on Intelligent Robots and Systems (2007)
10. Rachmielowski, A., Jägers, M., Cobzas, D.: Realtime visualization of monocular data for 3D reconstruction, Canadian Conference on Computer and Robot Vision, pp. 196-202 (2008)
11. Chekhlov, D., Gee, A., Calway, A., Mayol-Cuevas, W.: Ninja on a plane: Automatic Discovery of Physical Planes for Augmented Reality Using Visual Slam. 6th IEEE International Symposium on Mixed and Augmented Reality, pp. 1-4 (2007)
12. Gee, A., Chekhlov, D., Calway, A., Mayol-Cuevas, W.: Discovering Higher Level Structure in Visual SLAM, IEEE Transactions on Robotics, pp. 980-990 (2008)
13. Hengel, A.V.D., Dick, A., Thormählen, T., Ward, B., Philip H. S. Torr.: Building Models of Regular Scenes from Structure-and-Motion. In Proc. 17th British Machine Vision Conference, pp. 197-206 (2006)