ICAT 2005
December 5-8, Christchurch, New Zealand

A Vision-Based AR Registration Method Utilizing Edges and Vertices of 3D Model

Ryo Hirose, Hideo Saito
Department of Information and Computer Science, Keio University, Yokohama, Japan
{*ryo*, *saito*} @ozawa.ics.keio.ac.jp

Abstract

The most important issue in AR (Augmented Reality) is accurate geometric registration between real world and virtual objects. We propose a hybrid registration method utilizing the edges and the vertices of a 3D model of the target object. We estimate camera position and orientation by detecting the vertices and true edges every frames. The influence of the misleading edges is reduced by considering multiple edge candidates and selecting the best one from them. Furthermore the magnetic sensor or a set of the artificial vision markers as a tool to obtain approximate camera position and orientation are also used when the tracking object goes out of the user's view point or the camera movement is too fast to track natural features. By considering multiple edge candidates and utilizing a magnetic sensor or vision markers, the accuracy of registration and the robustness of rapid camera movement can be improved.

Keywords: Augmented Reality, Accurate Geometric Registration, 3D Object Model, Natural Features (Edges and Vertices), Magnetic 6DOF Sensor / Vision Markers

1. Introduction

MR (Mixed Reality) is the technique of combining real world and virtual world which is extensively studied in resent years [2, 3, 22]. In the field of MR, the technique which superimposes virtual objects onto the real world is especially called AR (Augmented Reality). MR/AR can provide the users with more effective information, because these techniques can be handling both worlds' information simultaneously. Thus MR/AR is widely used in many different fields, for example medical service [21], welfare, city or interior design, navigation systems [9], maintenance [7], work support, and so on [24].

For carrying out these AR applications, seamless superimposition between real world and virtual objects is a very important issue, such as geometric registration, temporal registration, and photometric registration. Various methods to solve these problems have been proposed. In this paper, we focus on a geometric registration, because the most important issue in AR is accurate geometric registration.

Geometric registration methods are classified into three categories.

1. Sensor-based registration method
2. Vision-based registration method
   - Artificial vision marker-based method
   - Natural feature-based method
   - Model-based method
3. Hybrid registration method

The advantages of using sensors such as magnetic sensors [14, 20], inertial sensors [1], and GPS are robustness to the movement of the user's view point, environment, and light conditions. There is no burden for the users because the position and the orientation of the user or the camera can be measured easily. However, such sensors are special and expensive, and user's moving range is limited to only sensor's effective range. Furthermore, the position and the orientation obtained by sensors are not enough accurate to achieve perfect geometric registration. Therefore this approach is often used together with other registration methods [1, 20].

The vision-based registration methods predict the position and the orientation of the user or the camera by identification of the features in the input images. The features can be either artificial vision markers or natural features. A set of the artificial vision markers which used for registration are placed in the experimental environment. Natural features exist everywhere in the environment. Sometimes, 3D models are also used when the shapes and the locations of the target objects or the target environment are known in advance.

In the artificial vision marker-based methods [4, 11, 12, 17], extracting the markers from input images is quite easy because they are designed to be easily detectable. As a result, predicting the position and the orientation is high speed and quite stable. However there are various weak points. For example, 3D positions of markers must be measured in advance. Alignment and maintenance of them involve an immense amount of time and effort to make. The user's moving range is limited to the space where they are arranged. Therefore it is quite difficult to apply this method in outdoor environments. Additionally, they give a bad visual appearance.

To solve issues of the vision markers, natural features based methods have been proposed. Generally, feature points [8, 15, 20], edges [10, 23], curves [6], planes [1, 19] and so on, are used for registration. Extraction of such natural features is more difficult than the artificial vision marker based approach. Furthermore the accuracy becomes worse because of mis-correspondence and mis-tracking. However, by using natural features, the user's movement range is not limited, and the original scenery is not broken. Therefore it is preferable to use this approach which has few limitations.

Model based approaches [13, 16] can also provide high accuracy of registration. Thus this approach is often used as taking the place of the artificial vision marker based approach. But it is hard to construct 3D models of object or environment.

These method introduced above has many advantages and weak points. However the advantage or the weak points are different from each method. Therefore hybrid method which is combined some different method is considerate. For example, combined sensor-based and marker-based [1, 20], marker-based and natural feature-based [8],

model-based and natural-feature-based [10, 23] and so on. As a result, hybrid method can be reduced the weak points and developed the advantages of each method. Additionally the registration accuracy is increased. Therefore hybrid method is used recently.

In this paper, we propose an accurate vision-based registration method that utilizes the natural features such as edges and vertices and a 3D model of the target object because natural feature-based methods have few limitations. However, if camera movement is too fast to track the natural features or the target object goes out of the user's view point, tracking will fail. Therefore, the artificial vision markers or the magnetic sensors are used in addition to the natural features to increase robustness.

2. Related Works

AR Toolkit markers are often used as the representative examples of the artificial vision markers. When multiple markers are utilized in the conventional vision based method, it is necessary that the geometric information of the marker arrangement such as their positions and orientations are known in advance. Kotake et al. [11] proposed a new AR Toolkit markers' calibration method. This approach does not require such geometric information of the markers. Though this approach uses multiple markers information to improve the accuracy, the natural features are also used to improve accuracy.

Oe et al. [16] proposed a model-based method intended for outdoor environments. In the offline procedure, they capture learning image sequences with an omni-directional camera to construct a 3D model of the target environment, extract feature points as the landmarks, and obtain template images around them. In the online procedure, the landmarks are extracted from input image and identified using a template matching method. Leptit et al. [13] proposed a high speed model-based method. They make CAD models of 3D target objects, obtain template images around feature points, and make image patches which deformed template images in advance. In the online procedure, input image correspondence with image patches in eigen space are found for a high speed matching. In both approaches, constructing a 3D model of target environment or objects and obtaining templates around feature points requires much effort. Additionally, the template matching method does not work well if the camera path of input image sequences is substantially different from the pre-captured learning images. Therefore, it is desirable not to use a template matching method which depends on the position of the view point.

Combined vision-based and model-based registration methods have also been used. Klein et al. [10] use a 3D model of a tablet, natural features, LED markers and a fixed camera. By utilizing edge information effectively, accurate registration is achieved because the tablet has many strong edges. This vision-based registration will fail when camera movement is large. Therefore, they use LED markers on the reverse side of the tablet and a fixed camera which captures and tracks the LED markers to predict the position and the orientation of the tablet. As a result, robustness to rapid camera movement is improved. However outside-in tracker is additionally needed and an offline procedure to learn LED markers' position is required. Vacchetti et al. [23] use a 3D model, edge and texture information. They considered multiple contour candidates instead of only the best one to solve edge ambiguities. By utilizing texture information in addition to edge information, the method can handle both textured and un-textured objects and the registration accuracy is more stable. However, it does not take account rapid camera movement.

3. Outline of Proposed Method

We use a video camera with pre-estimated intrinsic parameters as a device to capture image sequences. 3D object whose shape is already known is placed in the experimental environment. We also use vision markers or a magnetic sensor for the purpose of the improvement of the robustness. In the case of vision markers, they are placed on target object faces and those rough 3D coordinates are measured in advance. In the case of a magnetic sensor, it has been calibrated beforehand.

The target object, such as a box or a miniature house, has vertices and strong edges. Conventional model-based registration methods are based on template matching. However the template matching method does not work well if the camera path of the input image sequences differs substantially from the one that pre-captured learning images. Therefore we use the vertices of the object instead of texture images around the feature points, because the shapes of vertices do not depend on changing user's view point. Additionally, object edges have a lot of information. If correct edges are extracted from input images, the accuracy is improved.

Such natural feature-based method is not robust to rapid camera movement. Therefore we use the vision markers or a magnetic 6DOF sensor in addition to natural features tracking. This information is used when camera movement is too fast to track the natural features or tracking fails because the target object goes out of the field of view.

Figure 1 describes an overview of our proposed method. Input image sequences are taken with a pre-calibrated video camera. Then a 3D model is projected onto the input image plane by using the predicted extrinsic parameters $\boldsymbol{R}_t^0$ and $\boldsymbol{t}_t^0$. The predicted extrinsic parameters mean output extrinsic parameters of the previous frame. If trucking is failure in the previous frame, however, the outputs of the artificial vision markers or the magnetic sensor are used.

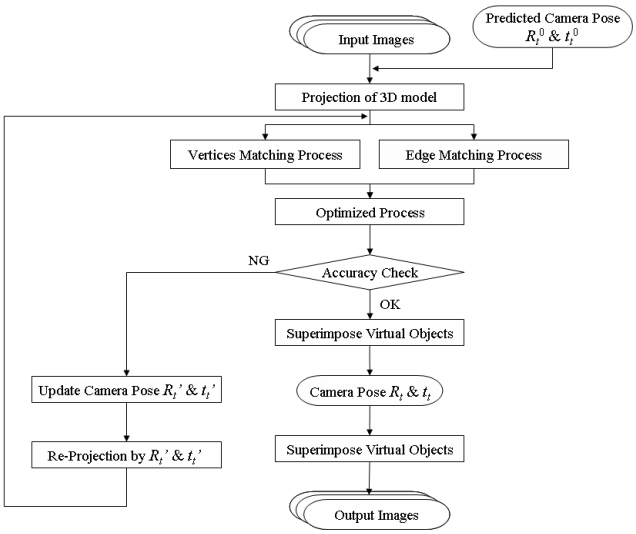


Figure 1: Overview of Our Method

In the case of vision markers, if they are detected from input image, the rotation matrix $\boldsymbol{R}_{t,i}$ and the translation vector $\boldsymbol{t}_{t,i}$ of the camera against a vision marker i is obtained in each image. Since the 3D positions of each marker are known, we can integrate $\boldsymbol{R}_{t,i}$ and $\boldsymbol{t}_{t,i}$ into one rotation matrix $\boldsymbol{R}_t^0$ and translation vector $\boldsymbol{t}_t^0$ against the world coordinates. Or if we use the magnetic sensor instead of vision markers, the rotation matrix $\boldsymbol{R}_t^0$ and the translation vector $\boldsymbol{t}_t^0$ in world coordinates is obtained in every frame.

After the projection, we search the vertices and correct edge lines from the input images, and match the natural feature points to the edges of 3D model.

Finally, we optimize an accurate rotation matrix $\boldsymbol{R}_t$ and translation vector $\boldsymbol{t}_t$ of the camera by minimizing the sum of squared projection error between the projected points of the 3D sample points and corresponding points. The camera intrinsic parameters $\boldsymbol{A}$ have been already calibrated. Therefore, virtual objects can be superimposed onto real world by using the camera intrinsic parameters $\boldsymbol{A}$ and the optimized extrinsic parameters $\boldsymbol{R}_t$, $\boldsymbol{t}_t$.

3.1. Projection of 3D Model

For corresponding some sample points on a 3D model and the natural feature points on the input image, we need to project a 3D model on the input image plane.

In the proposed method, we predict a rotation matrix $\boldsymbol{R}_t^0$ and translation vector $\boldsymbol{t}_t^0$ at first. Then 3D point $\boldsymbol{X}_{wi}$ on world coordinate can be transformed to 2D point $\boldsymbol{x}_i$ on image coordinate using the predicted extrinsic parameters $\boldsymbol{R}_t^0$, $\boldsymbol{t}_t^0$ and intrinsic parameters $\boldsymbol{A}$.

$$\lambda \tilde{\boldsymbol{x}}_i = \boldsymbol{A} \left[\boldsymbol{R}_t^0 \mid \boldsymbol{t}_t^0 \right] \tilde{\boldsymbol{X}}_{wi} \tag{1}$$

3.2. Vertex Matching Process

After projecting all vertices of the 3D model, small region areas around each projected points are set. We refer to this area as the search window in this paper. Figure 2 shows how to set the search window. We detect the corresponding point against vertex of 3D model from each search window. Experimentally, we have found a 21 × 21 search window to give good results.

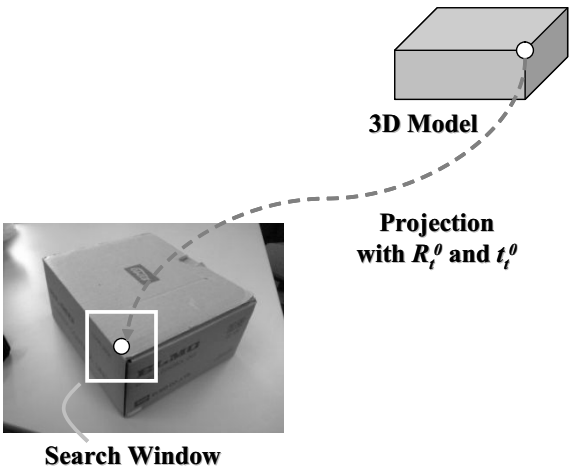


Figure 2: Search Window

First we extract the feature points from each search window and find corresponds to the vertices of the 3D model. We use the KLT-Tracker (Kanade-Lucas-Tomasi Tracker) [5, 18] to detect feature points in each search window. These feature points are extracted from not only the vertex of the object but the textures of the background or the surfaces of objects as well. Therefore, it is impossible to characterize only one of these feature points as a corresponding point. Thus further processing is required.

Figure 3 shows results of the KLT-Tracker. A white circular dot and black rectangular dots indicate the projection point of 3D model vertices and the detected feature points respectively.

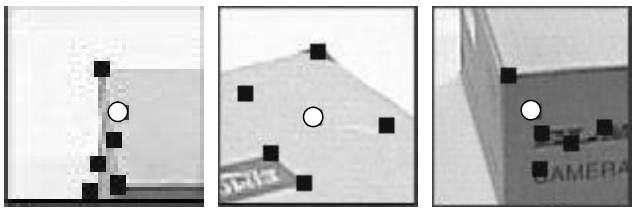


Figure 3: Results of KLT-Tracker for Detecting Feature Points

A vertex is an intersecting point of the object edges. Therefore it is determined by detecting two lines and calculating the intersecting point of them. Figure 4 shows results of Hough Transform. Input data is the edge detected binarized image. A white circular dot and a gray rectangular dot indicate the projection of 3D model vertices and an intersecting point, respectively.

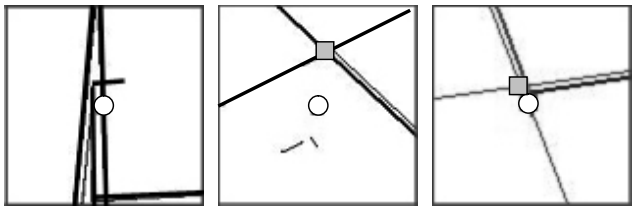


Figure 4: Results of Hough Transform for Detecting Two Lines and Calculated Intersecting Point

A pair of correct corresponding points is determined by utilizing both feature points detected by the KLT-Tracker and an intersecting point. Figure 5 shows results of the vertex matching process of various different camera view points. A white circular dot and a gray rectangular dot indicate a projection point of 3D model vertex and its corresponding point, respectively. Correct feature point are characterized as a corresponding point.

3.3. Edge Matching Process

3D target object has many strong edges. Therefore utilizing edge information is quite effective way to estimate accurate position and orientation of camera.

As described in Figure 6, the 3D model of the target object is projected in the input image at time t from the pre-

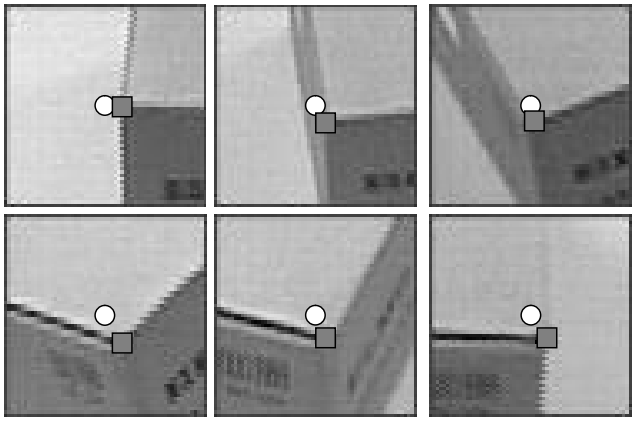


Figure 5: Results of Detecting Vertex

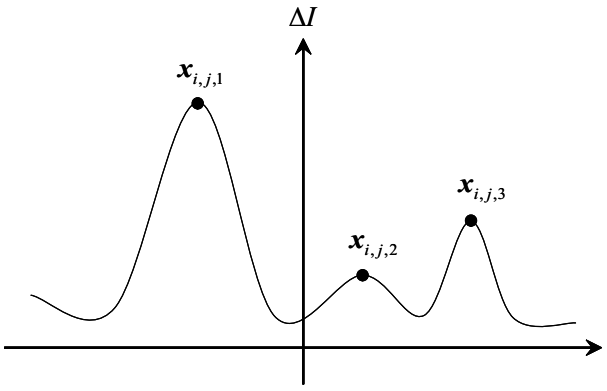


Figure 7: Distribution of Luminosity Gradient on Search Line $l_{i,j}$

dicted extrinsic parameters R_t^0 and t_t^0 of the camera. First the points are sampled along the projection of the edges E_i on the 3D model. Then, for each sample point $X_{i,j}$, the distribution of the luminosity gradient ΔI is checked for the corresponding points on the search line $l_{i,j}$ in the direction perpendicular to projection of the edge .

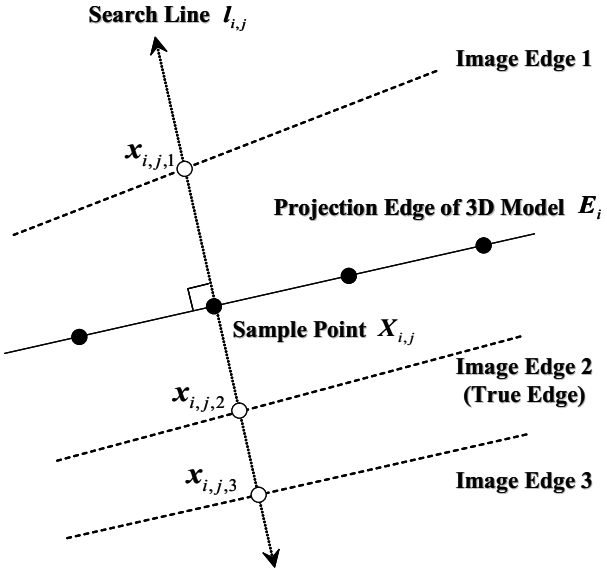


Figure 6: Searching Corresponding Point

In the conventional method [6], only one point which has the strongest luminosity gradient along the search line $l_{i,j}$ is considered as a corresponding point against the sample point $X_{i,j}$. This assumption sometimes causes miscorrespondence when other objects or the environment have strong contours such as checker patterns exist near the projection of 3D contour E_i.

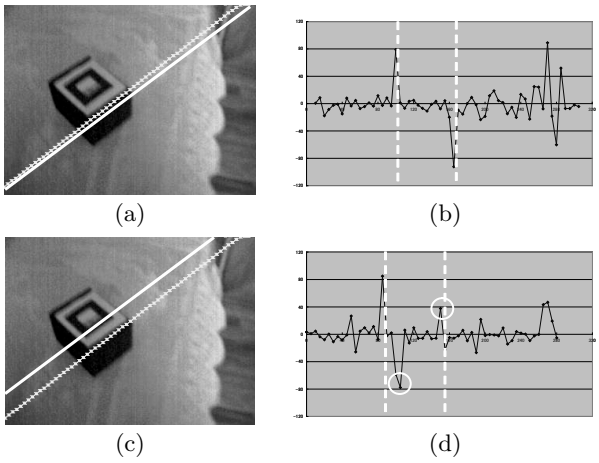
For example, Figure 7 describes a distribution of the luminosity gradient on search line $l_{i,j}$. If the conventional method is applied, $x_{i,j,1}$ is matched with sample point $X_{i,j}$.

However, $x_{i,j,1}$ actually does not exist on the real image contour, but $x_{i,j,2}$ is a real corresponding point. Thus the assumption that the point which has the strongest luminosity gradient is characterized as a corresponding point causes mis-correspondence and makes the registration accuracy worse.

By considering not only one feature point but several feature point candidates, the tracking will be more robust and the accuracy of tracking will be improved because the influence from strong contours on textured objects or the background is reduced. Therefore all local maximum points of the luminosity gradient such as $x_{i,j,1}$, $x_{i,j,2}$ and $x_{i,j,3}$ in Figure 7 are extracted as the candidates of the corresponding point. Then selecting one best corresponding point from these candidates is required. However, it is computationally expensive to select the best feature point from these points. Therefore we calculate some candidates of the corresponding edge, and select a best edge line from these candidates. First, all candidates of the corresponding point are classified into several classes depending on the distance from each sample point. Then the regression lines of each class are calculated by utilizing the linear least squares method. These lines are edge line candidates.

To choose one best edge line from some edge line candidates, the luminosity gradients in two direction, vertical direction and horizontal direction against edge candidates, should be computed. At first, the luminosity gradient in the horizontal direction is calculated. In the case of the true line, the value is always constant low, but the value of misleading edge which is made by checker pattern is not. Therefore the emphatic misleading edge is removed from edge candidates by checking horizontal luminosity gradient because the gradient of the misleading edge is differ from the one of true edge. As a result, the edge candidates with high possibility of true edge are extracted. Next, the luminosity gradient in the perpendicular direction is calculated. The highest value means the true edge. In this way, we estimate all of them using the luminosity gradient and select a best edge line.

Figure 8 shows the result of this process. Figure 8(a) & (c) are images of the observed one of the edge candidates. Figure 8(b) & (d) are the luminosity gradient of (a) and (c). The value of Figure 8(b) is constant low between the white dashed-line, so this line has high possiblity of true edge. In contrast, the high value appears between the white dashed-line in Figure 8(d). Therefore Figure 8(c) is removed from candidates.



(a) Edge candidate which has high possibility of true edge
(b) Luminosity gradient in the horizontal direction of (a)
(c) Misleading edge candidate
(d) Luminosity gradient in the horizontal direction of (c)

Figure 8: Detecting True Edge

Figure 9 describes the edge matching process. Figure 9(a) is the input image. Figure9(b) shows the result image after the extraction process of the corresponding point candidates. The white dots are the corresponding point candidates. (Local maximum points of the luminosity gradient) Most points are extracted from the boundary of the marker or another edge. Figure 9(c) shows the result image after the calculation of the edge line candidates. In this case, 5 lines are obtained. Figure 9(d) A white line regarded as a best edge line is chosen. This result shows that the edge matching process succeeds without receiving the influence of the misleading edges.

3.4. Optimization Process

If the vertices and sample points on an edge of the 3D model and the natural feature points extracted from an input image match, an accurate rotation matrix $\boldsymbol{R}_t$ and translation vector $\boldsymbol{t}_t$ for the camera are calculated.

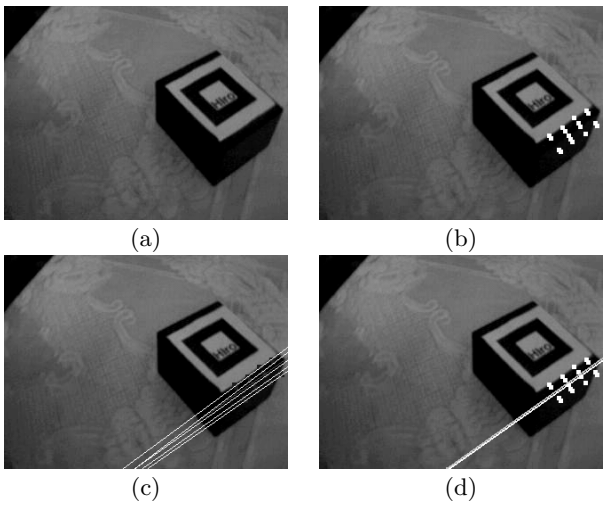
In this method, the optimized parameters are three rotation component and three translation component (six degrees of freedom in total). These parameters are expressed as a vector $\boldsymbol{s}$.

$$\boldsymbol{s} = \left[\begin{array}{cccccc} R_x & R_y & R_z & t_x & t_y & t_z \end{array} \right]^\top \tag{2}$$

We use the Steepest Descent Method which is an iterative calculation technique, to optimize parameters $\boldsymbol{s}$. We must calculate the Jacobian determinant $\boldsymbol{Jxs}$ in which we differentiate partially projected point $\boldsymbol{x} = [\ x,\ y\ ]^\top$ with respect to $\boldsymbol{s}$.

$$\boldsymbol{Jxs} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{s}} = \left[\begin{array}{cccccc} \frac{\partial x}{\partial R_x} & \frac{\partial x}{\partial R_y} & \frac{\partial x}{\partial R_z} & \frac{\partial x}{\partial t_x} & \frac{\partial x}{\partial t_y} & \frac{\partial x}{\partial t_z} \\ \frac{\partial y}{\partial R_x} & \frac{\partial y}{\partial R_y} & \frac{\partial y}{\partial R_z} & \frac{\partial y}{\partial t_x} & \frac{\partial y}{\partial t_y} & \frac{\partial y}{\partial t_z} \end{array} \right] \tag{3}$$

If the pair of the corresponding points of N pieces are obtained, the sum of the squared distance of the projected



(a) Input image
(b) Result image after the extraction process of the corresponding point candidates
(c) Result image after the calculation of the edge candidates
(d) Chosen best edge line and projected edge line

Figure 9: Edge Matching Process

point $\boldsymbol{x}_i = [\ x_i,\ y_i\ ]^\top$ of the 3D model vertex or the sample point on 3D model edge to the feature point $\boldsymbol{u}_i = [\ u_i,\ v_i\ ]^\top$ extracted from input image is

$$f(\boldsymbol{s}) = \sum_{i=0}^{N} \left\{ (u_i - x_i)^2 + (v_i - y_i)^2 \right\} \tag{4}$$

The Jacobian determinant $\boldsymbol{Jxs}$ describes how $\boldsymbol{x}_i$ changes with a small change in each component of $\boldsymbol{s}$. Therefore to reduce the sum of squared distance error $f(\boldsymbol{s})$ close to zero, parameter $\boldsymbol{s}$ is modified iteratively using $\boldsymbol{Jxs}$. Finally, the most relevant camera rotation and translation which minimize $f(\boldsymbol{s})$ obtined.

4. Implementation

The proposed method is implemented on a system using a web camera, a magnetic 6DOF sensor or AR Toolkit Markers as the vision markers. We use an ELECOM UCAM Series web camera. The image of this camera is captured with a resolution of 640 × 480 or 320 × 240 pixels. The magnetic 6DOF sensor used here is FASTRAK. The PC configuration is an Intel Pentium IV Processor 2.4GHz, and 256RAM running Windows XP. The intrinsic parameters of the camera are calibrated in advance. In addition, the magnetic sensor and the vision markers have also been calibrated in advance.

First, we show the efficacy of our method by superimposing the shape of the 3D model onto the input images with optimized extrinsic parameters $\boldsymbol{R}_t$ and $\boldsymbol{t}_t$. Figure 10 and Figure 11 show results using the magnetic sensor and the vision markers, respectively. It can be observed that accurate registration is achieved since the projected image of 3D model and the target object in the captured image are almost corresponding in both cases. In addition, in the

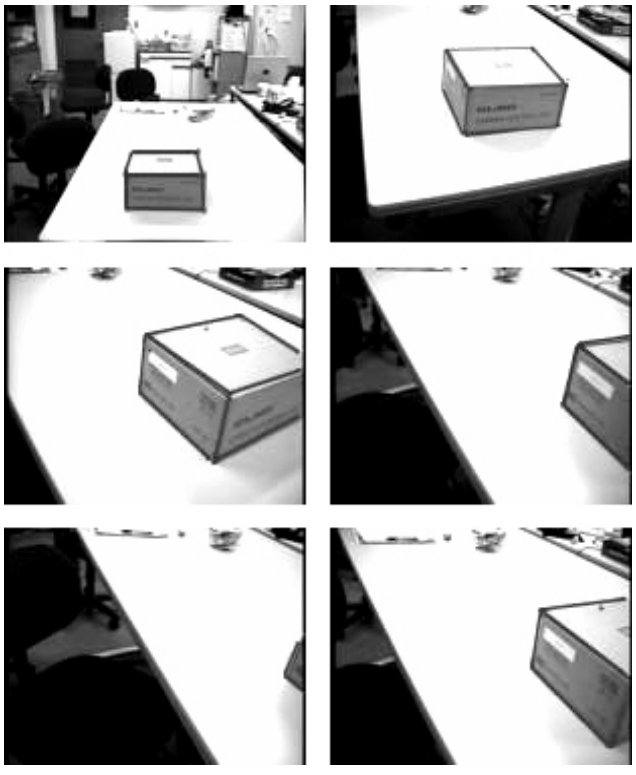


Figure 10: Result Image of Using Magnetic Sensor

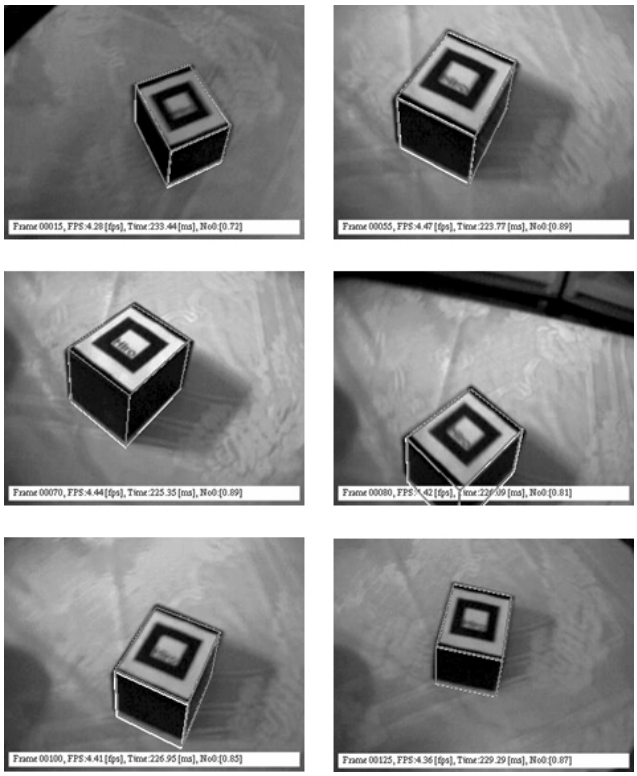


Figure 11: Reslut Image of Using Vision Marker

case of a magnetic sensor, the tracking succeeds even if the target object goes out from the user's view point. In the second experiment, we tested the advantages of considering multiple edge line candidates.

Next we calculated the projection error and frame rate in the case of Figure 11. Figure 12 shows that the distance of the ideal positions which are calculated manually and extracted corresponding points are about 2.65 pixels on an average. Additionally, tracking restart in next frame by utilizing a vision marker together, although tracking is failure in several frames because of rapid camera movement. This result shows the extrinsic parameters are corrected with high accuracy.

Figure 13(a) shows the checker patterns contours creating numerous strong misleading edges. If only one corresponding point is selected against each sample point, the selected point is corresponded to the misleading edge on the checker pattern contours as shown in Figure 13(b). In the proposed method, which considers multiple edge candidates, it can successfully correspond to correct points on real edges as shown in Figure 13(c).

Finally, a pot is superimposed onto the experimental environment using Open GL to demonstrate that the proposed method can be applied to AR applications. In Figure 14(a), one virtual pot is placed onto the target object. In Figure 14 (b), a virtual pot is placed on the box in the front and another virtual pot is placed onto the box in the back. As tracking is accurate enough, the virtual object can be superimposed stably.

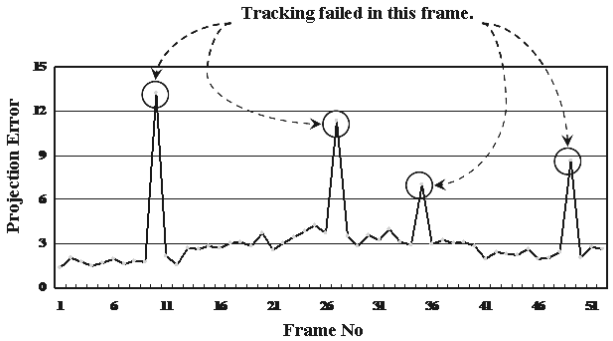


Figure 12: Projection Error

5. Conclusion

A geometric method for AR system using a 3D model, natural features, and a magnetic sensor or vision markers together to obtain an approximate camera position and orientation is proposed in this paper.

Utilizing vertices which do not depend on view point loosens restriction on camera movement. Considering not only one edge but multiple edge candidates allows the use of edge information even if strong misleading edges exist near the true edge. As a result, the accuracy of registration is improved. Additionally, by combining information of the magnetic sensor or the vision markers, the robustness

to rapid camera movement can be improved. The efficacy of this proposed method was demonstrated by experiment, and confirmed that the proposed method can be applied to the AR applications.

In the future, it is necessary to improve the accuracy of the projection of the 3D model edges to true edges in input images. In addition, it is preferable to reduce the constraints on the arrangement of the vision markers.

References

[1] Michael Aron, Gilles Simon, and Marie-Odile Berger. Handling uncertain sensor data in vision-based camera tracking. *Proc. of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR2004)*, pages 58–67, 2004.

[2] Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, Aug, 1997.

[3] Ronald T. Azuma, Y. Baillot, R. Behringer, S. Julier S. Feiner, and B. MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, 2001.

[4] Mark Billinghurst, Hirokazu Katoh, and Ivan Poupyrev. The magic book: A transitional ar interface. *Computers and Graphics*, 25:745–753, 2001.

[5] Stan Birchfield. Derivation of kanade-lucas-tomasi tracking equation. *in Technical Report*, 1997.

[6] Andrew I. Comport, Eric Marchand, and Francois Chaumette. A real time tracker for markerless augmented reality. *Proc. of the Second IEEE and ACM International Symposium on Mixed Reality and Augmented Reality (ISMAR2003)*, pages 36–45, 2003.

[7] Steven Feiner, Blair Mocintyre, and Doree Seligmann. Knowledge-based augmented reality. *ACM Communications*, pages 52–62, 1993.

[8] Y. Genc, S. Riedel, F. Souvannavong, and N. Navab C. Akinlar. Marker-less tracking for augmented reality: A learning-based approach. *Proc. of the International Symposium on Mixed and Augmented Reality (ISMAR2002)*, pages 295–304, 2002.

[9] Tobias Hollerer, Steven Feiner, Tachio Terauchi, Gus Rashid, and Drexel Hallaway. Exploring mars: Developing indoor and outdoor user interface to a mobile augmented reality system. *Computers and Graphics*, 23(6):779–785, 1999.

[10] Georg Klein and Tom Drummond. Sensor fusion and occlusion refinement for tablet-based ar. *Proc. of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR2004)*, pages 38–47, 2004.

[11] Daisuke Kotake, Shinji Uchiyama, and Hiroyuki Yamamoto. A marker calibration utilizing a priori knowledge on marker arrangement. *Proc. of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR2004)*, pages 89–98, 2004.

[12] Kiriakos N. Kutulakos and James Vallino. Affine object representations for calibration-free augmented reality. *Proc. IEEE Virtual Reality Annual International Symposium (VRAIS1996)*, pages 25–36, 1996.

[13] Vincent Lepetit, Luca Vacchetti, Daniel Thalmann, and Pascal Fua. Fully automated and stable registration for augmented reality applications. *Proc. of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR2003)*, pages 93–102, 2003.

[14] Mark A. Livingston and Andrei State. Magnetic tracker calibration for improved augmented reality registration. *Presence: Teleoperators and Virtual Environments, MIT Press*, 6(5):532–546, Oct, 1997.

[15] Ulrich Neumann and Suya You. Natural feature tracking for augmented reality. *IEEE Transaction on Multimedia*, 1(1):53–64, 1999.

[16] Motoko Oe, Tomokazu Sato, and Naokazu Yokoya. Estimating camera position and posture by using feature landmark database. *14th Scandinavian Conference on Image Analysis (SCIA 2005)*, pages 171–181, 2005.

[17] Kiyohide Satoh, Shinji Uchiyama, Hiroyuki Yamamoto, and Hideyuki Tamura. Robust vision-based registration utilizing bird's-eye view with user's view. *Proc. of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR2003)*, pages 46–55, 2003.

[18] Jianbo Shi and Carlo Tomasi. Good features to track. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR1994)*, pages 593–600, 1994.

[19] Gilles Simon and Marie-Odile Berger. Pose estimation for planar strutures. *IEEE Computer Graphics and Applications*, pages 46–53, 2002.

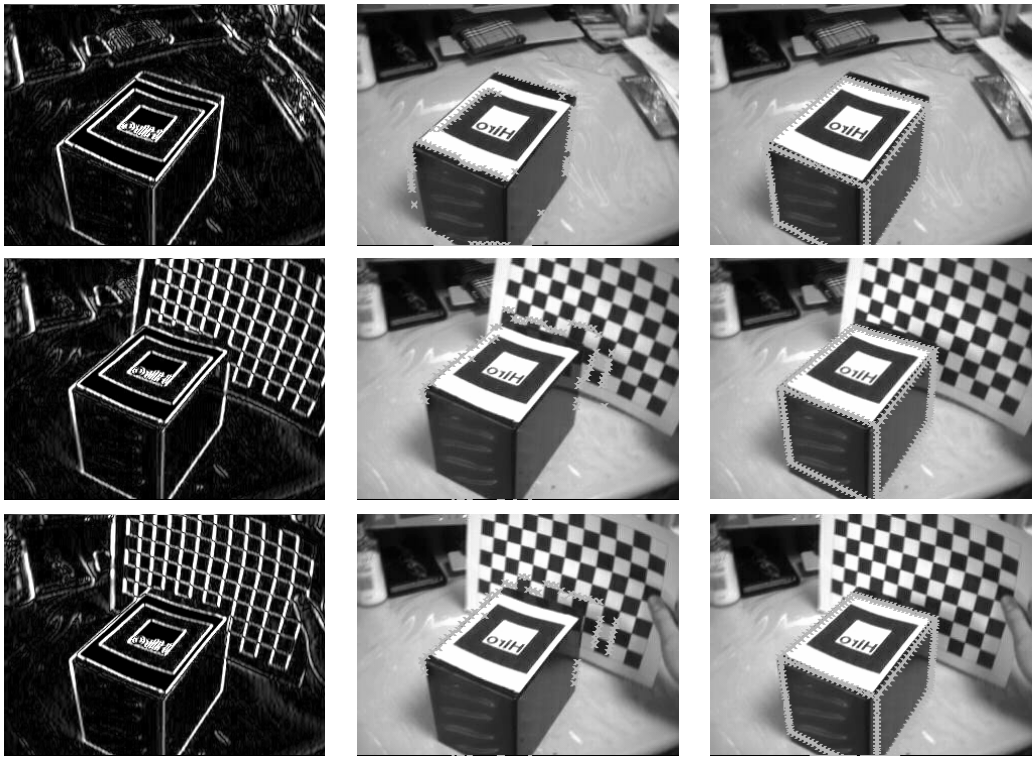
[20] Andrei State, Gentaro Hirota, David T. Chen, William F. Garrett, and Mark A. Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. *Proceedings of ACM SIGGRAPH 1996*, pages 429–438, Aug, 1996.

[21] Andrei State, Mark A. Livingston, William F. Garrett, Gentaro Hirota, Mary C. Whitton, Etta D. Pisano, and Henry Fuchs. Techniques for augmented-reality systems: Realizing ultrasound-guided needle biopsies. *Proc. SIGGRAPH 1996*, pages 439–446, 1996.

[22] Hideyuki Tamura, Hiroyuki Yamamoto, and Akihiro Katayama. Mixed reality: Feature dreams seen at the border between real and virtual worlds. *IEEE Computer Graphics and Applications*, 21(6):64–70, 2001.

[23] Luca Vacchetti, Vincent Lepetit, and Pascal Fua. Combining edge and texture information for real-time accurate 3d camera tracking. *Proc. of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR2004)*, pages 48–57, 2004.

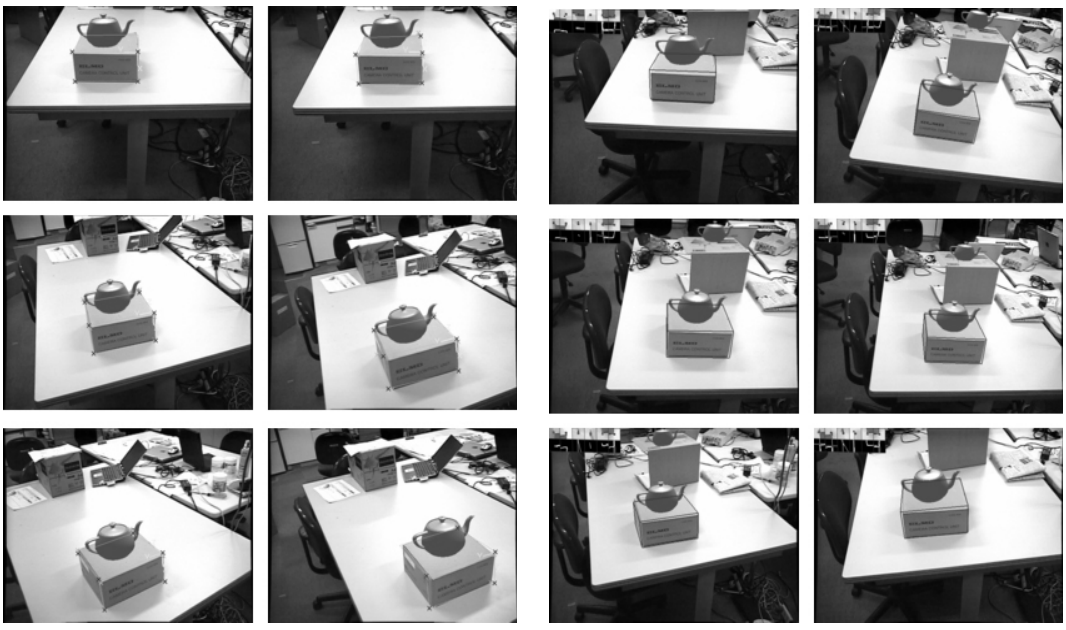
[24] Hiroyuki Yamamoto. Case studies of producing mixed reality worlds. *Proc. IEEE SMC'99*, 6:42–47, 1999.



(a) (b) (c)

(a) Results of edge detection. The checker pattern contours create numerous strong misleading edges.
(b) Results of conventional method. Corresponding points are attracted by the checker pattern.
(c) Results of proposed method. Accurate correspondence succeeds without receiving the influence of the misleading contours.

Figure 13: Comparison of Edge Matching Process



(a) (b)

(a) One virtual pot is superimposed onto the target object box.
(b) Two virtual pots are superimposed. One is placed onto the target object in the front and the other is placed onto the box in the back.

Figure 14: Superimposition of the Virtual Objects