A Natural Feature-Based 3D Object Tracking Method for Wearable Augmented Reality

Takashi Okuma
Columbia University / AIST
Email: okuma@cs.columbia.edu

Takeshi Kurata
University of Washington / AIST
Email: kurata@ieee.org

Katsuhiko Sakaue
AIST
Email: k.sakaue@aist.go.jp

Abstract- **In this paper, we describe a novel natural feature based 3-D object tracking method. Our method determines geometric relation between known 3-D objects and a camera, not using fiducial markers. Since our method only uses a camera to determine this geometric relation, it is suitable for wearable augmented reality (AR) systems. Our method combines two different types of approaches for tracking: a bottom up approach (BUA) and a top down approach (TDA). We mainly use a BUA, because it acquires accurate results with small calculation cost. When BUA cannot output an accurate result, our method starts TDA to avoid mistracking. An experimental result shows an accuracy and integrity of our method.**

I. INTRODUCTION

Tracking methods are one of the most important issues in the field of Augmented Reality (AR). AR systems overlay virtual objects onto the real world to help their user do an activity in the real world. In many cases, AR systems need to know accurate geometric relations between real objects and users' viewing position to locate virtual objects onto suitable position of the real world. Tracking methods determine these geometric relations. Various tracking methods have been developed in the field of AR [1][2], and it is important to select the appropriate tracking method for the application requirements.

For some application systems that have cameras in their configuration, like a video see-through AR system or a wearable AR system, a vision-based tracking method is appropriate. Vision-based tracking methods use cameras that capture real scene as sensing devices. Some of them don't need to set any sensing devices in the environment unlike ultra-sonic or magnetic trackers do. Video see-through AR systems have one or two cameras, and capture real image sequences to be used as background for synthetic AR images [3][4][5][6]. Therefore, video see-through AR systems can use a vision-based tracking method without changing their system configuration. We have been developing wearable AR systems that have video see-through configuration and use a vision-based tracking method [7][8]. We believe our design could give a practical platform for AR applications to consumer.

In these systems, the camera position corresponds to the user's viewing position. In this case, the geometric relations that are required by these AR systems are called external camera parameters. Vision-based tracking methods determine these geometric relations by matching a known model and image sequences.

Figure 1 : Output stills of a 3-D online manual.

However, it is difficult to match captured image sequences and known models that have general shape and color because of image noise, occlusions of the object, and so on. Therefore, some vision-based tracking methods set fiducial markers on real objects to help the matching process [3][4][5][6]. We call these methods fiducial-based tracking methods. For example, multiple color circle fiducials are used to detect known points in the method described in ref. [3] and [4]. Another method defines matrix codes to recognize objects and use the four corners of a matrix code as known points [5]. The ARToolKit [6] uses black square regions with black and white patterns as fiducials. Fiducial-based tracking methods are also very practical and appropriate for some AR application; for example, video conferencing applications, interior design applications, and so on [2].

However, an environment with too many fiducials is unnatural and may limit applications. Some applications require tracking methods that don't use fiducials. A three-dimensional online manual is a good example of such applications. Figure 1 shows the output examples of a prototype three-dimensional online manual. Fiducial markers cannot be set for annotation on all objects in this application. Therefore, tracking methods that don't use fiducials have been attracting attention.

II. OBJECT TRACKING METHODS

We call tracking methods that don't use fiducials natural feature based tracking methods. Some research groups have been developing natural feature-based tracking methods [7][8][9]. There are two types of approaches to determine

the camera parameters:

- Methods based on the bottom-up approach (BUA),
- Methods based on the top-down approach (TDA).

In the field of augmented reality, BUA have mainly been used not only in natural feature based methods but also in fiducial-based methods. BUA-based methods have the following two dominant steps:

1. Reference points tracking step: This step outputs two-dimensional image coordinates of reference points. We define reference points as points of which three-dimensional positions and templates for image matching are registered in a database. Usually, BUA-based methods use points that can be easily detected such as corner points or points on edges as reference points. In other words, BUA-based methods match the captured image sequences and known models of reference points in this step.
2. Camera parameters calculation step: This step calculates camera parameters using the known three-dimensional positions and the obtained image coordinates of reference points. When at least three reference points are detected or tracked, this step can calculate camera parameters.

BUA-based methods can calculate accurate camera parameters with low calculation cost. However, it is difficult to detect and track the two-dimensional position of reference points. One important reason for this difficulty is a problem of reference points changing the appearances. Appearances of local areas around reference points change as the camera moves, so reference points in a captured image and templates in the database do not match. This causes mis-tracking of the reference points. Another important reason is occlusion of the reference points. When reference points are occluded by another real object, templates of reference points are probably matched to incorrect points. This also causes mis-tracking of reference points. Mis-tracking of the reference points results in an inaccurate estimation of the camera parameters. Therefore, BUA based tracking methods have to deal with these two problems.

On the other hand, TDA-based methods can robustly estimate camera parameters using context and multiple hypotheses. One well-known TDA-based method is the ConDensation [10] framework. The ConDensation framework has multiple hypotheses represented with a discrete probability density of parameters to be determined. We can use the ConDensation framework to estimate the camera parameters [8]. In this case, we represent the discrete probability density of the camera parameters with a set of samples of each frame. A sample shows possible camera parameters. TDA-based methods can track an object even if the object is occluded or is in clutters. However, to track the target in real time, we have effectively to limit the extent of sampling area and the number of samples.

Inertial orientation sensors can help vision-based tracking methods, because they can give camera orientation data even if camera moves quickly. Quick camera motion causes an image motion blur that makes vision-based tracking difficult. Some tracking methods use inertial orientation sensors with vision-based tracking. Inertial orientation sensors can be easily added to wearable AR systems and video see-through AR systems because these sensors do not require external sensing devices. However, output error of these sensors accumulates during tracking process because they don't have any references. We have to treat this problem, which is called as drift, when we combine inertial orientation sensors with vision-based tracking.

III. OUR METHOD

We propose a novel tracking method based on hybrid framework of a BUA-based estimation and a TDA-based estimation. Our method also uses an inertial orientation sensor. Figure 2 shows a diagram of our method. The thick arrows indicate flow of the processes, and the thin arrows indicate flow of data

First, our method predicts camera parameters using the camera position and velocity of the previous frame. To predict the parameters effectively, our method uses data from an inertial orientation sensor. Subsection A describes the detail of this prediction step.

Then, our method estimates the camera parameters using a BUA-based estimation. The BUA-based estimation calculates a number of potential camera parameters using the BUA and the predicted parameters, and then it determines the camera parameters using robust statistics. When the error in the estimated parameters is sufficiently small, the method outputs the parameters. Subsection B shows the details of the BUA-based estimation.

When the error in the parameters estimated by BUA-based estimation is not sufficiently small, our method starts estimating the camera parameters using a TDA-based estimation. Our method uses the estimated parameters by the BUA-based estimation to create an initial discrete probability density. While the error in the output parameters from BUA-based estimation is not sufficiently small, the TDA-based estimation works. The detail of the TDA-based estimation is described in Subsection C. Our method compares the error of the TDA-based estimation with the error of the BUA-based estimation. Then, our method outputs the parameters with the smaller error.

A database used in the proposed method has the following information:

- reference images: images that capture objects being tracked,
- camera parameters of reference images: the camera parameters when each reference image is captured,
- position of reference points: points on the object being tracked that are detected as feature points on the reference images.

Reference points are defined as a local areas that have big eigenvalues in the following matrix [11].

$$\mathbf{M}_{tp} = \begin{pmatrix} \sum \left(dI/dx \right)^2 & \sum \left(dI/dx \right)\left(dI/dy \right) \\ \sum \left(dI/dx \right)\left(dI/dy \right) & \sum \left(dI/dy \right)^2 \end{pmatrix} \quad (1)$$

The two-dimensional image coordinates on the reference image and the three-dimensional positions of reference points are measured in advance. This information is used in

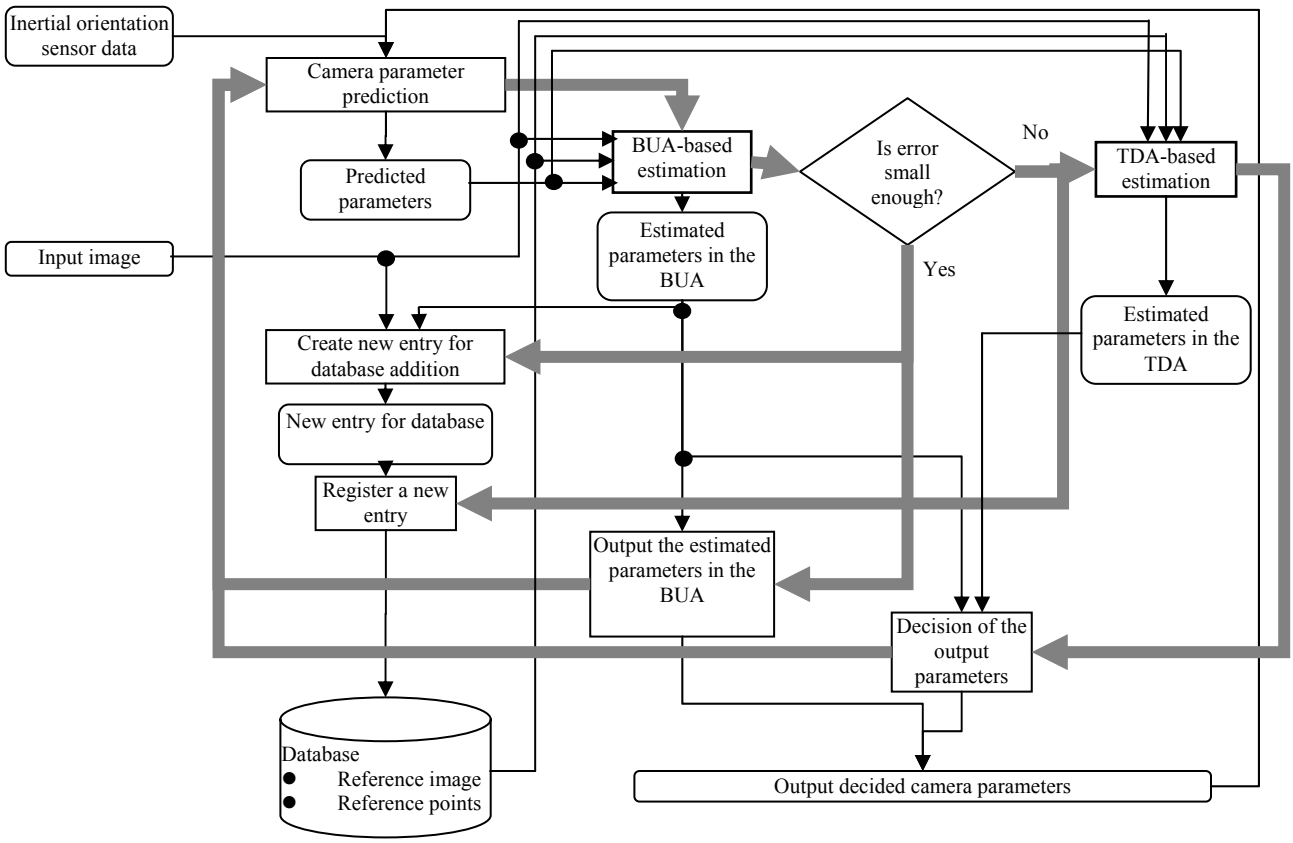


Figure 2: Flow diagram of the proposed method.

both of the BUA-based estimation and the TDA-based estimation. To deal with the problem of the feature points changing the appearances, the Automatic Database Addition (ADA) step automatically adds the appearance data of the object being tracked to the database as appropriate while it is tracking the object. We describe the detail of the ADA step in Subsection D.

A. Parameter prediction

To track an object effectively, our method predicts the camera parameters using the parameters of the previous frame, the velocity of the camera, and the inertial orientation sensor data.

Because inertial orientation sensors measure only the three orientation parameters, we assume three types of movement to predict the six camera parameters; three position parameters and three orientation parameters. To eliminate effects of drift and error accumulation, the prediction step uses the difference between the data of the sensor in the previous frame and that in the current frame. The three types of movements are these:

- Movements of the user when he/she moves the object in the field of view. These movements do not affect the rotation sensor data.
- Movements of the user when he/she is moving around the object to observe it (object-centered rotation). The object appears to be rotating around its center. The rotation is in the direction opposite to that of the rotation of the sensor.
- Movements of the user when he/she is looking around (user-centered rotation). The object appears to be rotating around the viewing position. The rotation angle and the direction of rotation are the same as those of the rotation sensor.

Our method calculates six sets of predicted camera parameters PCPn:

$PCP1: CP_{t-1}$

$PCP2: CP_{t-1}+VC_{t-1}$

$PCP3: T_{OCR}(CP_{t-1})$

$PCP4: T_{OCR}(CP_{t-1}+VC_{t-1})$

$PCP5: T_{UCR}(CP_{t-1})$

$PCP6: T_{UCR}(CP_{t-1}+VC_{t-1})$

where CP_{t-1} is the camera parameters in the previous frame, and VC_{t-1} is the velocity of the camera which can be calculated as $CP_{t-1} - CP_{t-2}$. $T_{OCR}(CP)$ and $T_{UCR}(CP)$ indicates the transformation of the object-centered rotation and that of the user-centered rotation.

B. Bottom-up approach (BUA) based estimation

A BUA-based estimation is a primary part of our tracking method. As we mentioned, a BUA-based approach has to deal with mis-tracking of reference points. To reduce mis-tracking by a problem of reference points changing appearance, our BUA-based estimation uses multiple reference images and a reference image rotation. In addition, our BUA-based estimation step uses LMedS framework with the method for solving P3P problems. The LMedS framework makes BUA-based estimation possible to calculate the camera parameters from a reference points tracking result that includes mis-tracking. The following subsections describe the details.

1) Reference points tracking step using multiple reference images and reference image rotation.

In this step, feature points that correspond to reference points are detected in input image. Our method adopts the Lucas-Kanade method, which tracks feature points by iterative calculation using gradient information about local areas[11]. Since the Lucas-Kanade method assumes only the translations of feature points on an image plane, it cannot deal with the problem of the reference points changing appearances.

Therefore, our method prepares multiple reference images. To minimize the calculation cost, only two reference images are used in the tracking for each set of the predicted parameters; PCP1 to PCP6. One reference image is the image that is used in the previous frame. The other one is selected by comparing the predicted parameters with registered camera parameters of the reference images. (When these two reference images are the same, the only one image is used.) Our BUA-based estimation uses the predicted parameters to calculate the initial values of the iterative calculation in the Lucas-Kanade method. As a result, the reference points detection is processed twelve times in maxim.

We also use an image rotation to deal with the problem of the reference points changing the appearances. The image rotation can approximate the appearance of the object when the object rotates around the optic axis of the camera, even if the object has a three-dimensional shape. The rotation angles are calculated as the difference between the rotation parameters of reference image and that of the predicted parameters. This image rotation step does not need to be done with the viewpoint of accuracy. However, it can prevent too much addition of entries into the database from the ADA step described in subsection D. It can contribute the total efficiency of the system.

2) Parameter estimation using the method for solving P3P problems and the LMedS framework.

Our BUA-based estimation uses a method for solving P3P problems in the LMedS framework. Methods for solving PnP problems can calculate accurate camera parameters using two-dimensional image coordinates and the three-dimensional position of the *n* points. However, as we mentioned, the results of tracking all the *n* points are rarely accurate because of mis-tracking of reference points. We thus use the LMedS framework to estimate the camera parameters because the LMedS framework can acquire accurate camera parameters at very high possibility if more than a half of all reference points are correctly tracked. After the LMedS framework acquires the camera parameters, mis-tracked reference points can be eliminated as the outliers. Therefore, our method optimizes the parameters with Levenberg-Marquadt method using reference points that are tracked correctly. The following shows the detail steps.

Step 1. This step randomly selects three points from *n* tracked reference points, and then calculates potential camera parameters using a method for solving P3P problems [12].

Step 2. This step calculates an error err_{LMedS} defined by the following equation.

$$err_{LMedS}{}^2 = \mathrm{med}\left(\left(x_i - \tilde{x}_i\right)^2 + \left(y_i - \tilde{y}_i\right)^2 \right) \qquad (2)$$

where (x_i, y_i) are the image coordinates of the tracked *i*-th reference point, $(\tilde{x}_i, \tilde{y}_i)$ are the image coordinates onto which the three-dimensional position of the reference point is projected with the calculated camera parameters, and med(f(i)) indicates the median of the f(i) for all i.

Step 3. Steps 1 and 2 are repeated *m* times. The smallest number of times, *m,* is determined by the following inequality:

$$p < 1 - (1 - r^3)^m \qquad (3)$$

where *p* is the assumed probability that the camera parameters are calculated with correctly tracked reference points, and *r* is a rate at which reference points are assumed to be tracked correctly.

Step 4. To detect inliers, the three-dimensional positions of the reference points are projected onto image plane using the camera parameters with the smallest err_{LMedS}.

Step 5. This step acquires the camera parameters that have the smallest err_{BUA} using Levenberg-Marquadt method. err_{BUA} is defined by the following equation.

$$err_{BUA} = \left(x_i - \tilde{x}_i\right)^2 + \left(y_i - \tilde{y}_i\right)^2 \qquad (4)$$

C. Top-down approach (TDA) based estimation

When the BUA-based estimation cannot obtain camera parameters with sufficiently small err_{BUA}, our method uses the TDA-based estimation. In this study, we design the TDA-based estimation to process just like the ConDensation algorithm. Our TDA-based estimation tracks the target object by repeating three steps; Sampling, Observation, and Decision step.

1) The sampling step.

As we mentioned before, TDA-based estimation represents a discrete probability density of the camera parameters at each frame. The sampling step generates a new sample set for the discrete probability density. To create effective samples, the sampling step uses the predicted parameters and the parameters estimated by the BUA-based estimation. The following paragraphs describe the sampling steps that create sample set $\left\{s_t^{(1)}, s_t^{(2)}, \cdots, s_t^{(N)}\right\}$ of time *t*, where s_t denotes a sample that indicates the camera parameters, and *N* denotes the number of samples.

First, the sampling step creates a set comprising 1/7 of all the samples $\left\{s_t^{(1)}, s_t^{(2)}, \cdots, s_t^{(N/7)}\right\}$ using random sampling. The center of distribution used in the random sampling is set to the parameters estimated by the BUA-based estimation. The sampling step selects the method for creating the rest of the samples depending on whether the BUA-based estimation or the TDA-based estimation was used to estimate the camera parameters of time *t-1*.

Case 1) the output of time *t-1* was estimated by using the BUA-based estimation: Each set comprising 1/7 of all the samples $\left\{s_t^{(jN/7+1)}, s_t^{(jN/7+2)}, \cdots, s_t^{((j+1)N/7)}\right\}$, $(1 \le j \le 6)$ is generated by random sampling. The center of distribution used in the random sampling is set to PCP*j*.

Case 2) the output of time *t-1* was estimated by using the

TDA-based estimation: The sampling step creates the rest of samples using the following three sub-steps;

- First sub-step selects the $6N/7$ samples from a set of samples with time *t-1* $\{s_{t-1}^{(1)}, s_{t-1}^{(2)}, \cdots, s_{t-1}^{(N)}\}$ according to their weights $\{\pi_{t-1}^{(i)}\}, (i = 1,2,..., N)$, which are calculated in the previous observation step. The selected samples are denoted by $\{s_t'^{(1)}, s_t'^{(2)}, \cdots, s_t'^{(6N/7)}\}$.
- Second sub-step sets $s_t''^{(k)}$, $(k = 1,..,6N/7)$ by the following equation:

$$s_t''^{(k)} = \begin{cases} s_t'^{(k)} & (1 \le k \le N/7) \\ s_t'^{(k)} + VC_{t-1} & (N/7+1 \le k \le 2N/7) \\ T_{OCR}(s_t'^{(k)}) & (2N/7+1 \le k \le 3N/7) \\ T_{OCR}(s_t'^{(k)} + VC_{t-1}) & (3N/7+1 \le k \le 4N/7) \\ T_{UCR}(s_t'^{(k)}) & (4N/7+1 \le k \le 5N/7) \\ T_{UCR}(s_t'^{(k)} + VC_{t-1}) & (5N/7+1 \le k \le 6N/7) \end{cases}$$

- The last sub-step generates samples $s_t^{(h)}$, from $s_t''^{(h-N/7)}$ $(h = N/7+1,..., N)$ using random walk.

2) The observation step.

Our TDA-based estimation step calculates an evaluation value of each sample based on the image observation. The observation step detects natural feature points of input frame as local areas that have big eigenvalues in the matrix M_{fp} of the equation (1). The observation step uses the image coordinates of the nearest detected point (x_i', y_i') in place of the (x_i, y_i) coordinates to calculate the error.

$$err_{TDA}^{(i)\,2} = \mathrm{med}\left(\left(x_i' - \tilde{x}_i \right)^2 + \left(y_i' - \tilde{y}_i \right)^2 \right).$$

Then, the observation step calculates a weight $\pi^{(i)}$ of each *i*-th sample by the following equation:

$$\pi^{(i)} = e^{-err_{TDA}^{(i)\,2}/2}.$$

Finally, the observation step normalize $\pi^{(i)}$ so that

$$\sum_{n=1}^{N} \pi^{(n)} = 1$$

.

3) The decision step.

The decision step calculates the weighted average of a sample set using $\pi^{(i)}$ as a weight of each sample, and outputs these average camera parameters as the representation of the estimated probability density of the frame.

D. Automatic Database Addition.

To track the object from any viewpoint, the proposed method requires a database that has the multiple reference images captured from different viewpoints. To simplify the preparation of a database, our tracking method uses Automatic Database Addition step. This step automatically adds new entry during the tracking process. An entry of the database consists of three components; 1) the frame image, 2) calculated parameters, and 3) image coordinate values and three-dimensional positions of reference points.

When the TDA-based estimation starts, the automatic database addition step adds these three components of the previous frame as a new entry if the number of tracked reference points is larger than threshold and the err_{BUA} is smaller than threshold. This step can reduce the number of reference images in the initial database and the necessity of complicated preparation.

IV. EXPERIMENT

To evaluate the tracking competence of our method, we input real image sequences to our method. The initial database had only one entry. The method was implemented on a PC (Intel Xeon, dual CPU, 1.7 GHz). Figure 3 shows the change in the min(err_{BUA}, err_{TDA}) and the number of reference images in the database as well as some images that the axes of the object coordinates are overlaid onto. These tracking results show that the proposed method can track the real object in real image sequences. These results also show the proposed method can track even if the user's hand occludes the object to be tracked. The time of processing each frame was, on average, 33 ms when only the BUA-based estimation was used, and 252 ms when both the BUA-based estimation and the TDA-based estimation were used.

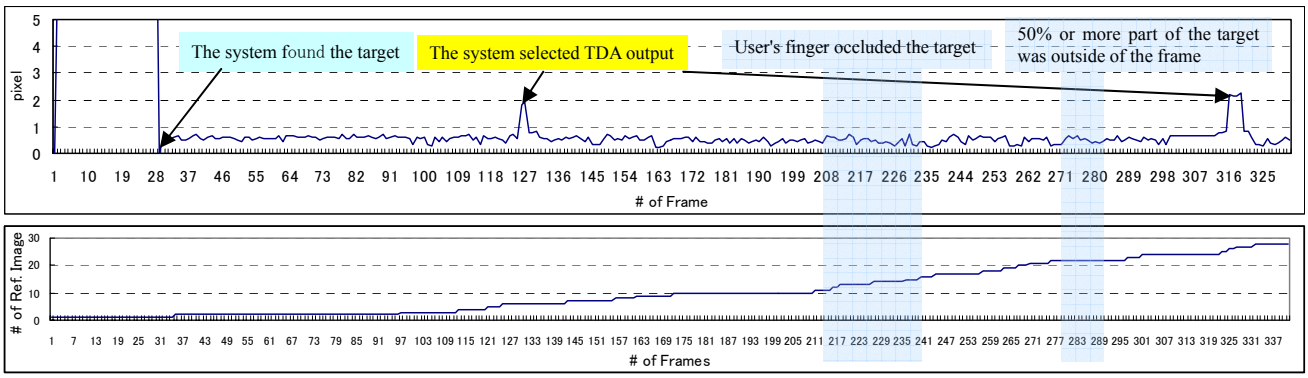
V. CONCLUSION

In this paper, we described a natural feature based 3-D object tracking method for video see-through and/or wearable augmented reality systems. This method is based on a combination of the BUA, TDA, and Automatic Database Addition.

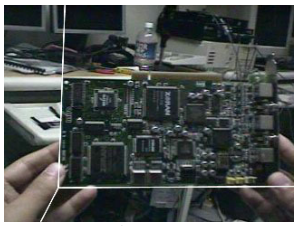
Basically, we use the BUA-based estimation to track objects. The BUA-based estimation can track the object even if miss-tracking of the reference points occurs because it is based on the LMedS framework. In the LMedS framework, method for solving P3P problems calculates each potential external camera parameters using the two-dimensional coordinates and three-dimensional object coordinates of the tracked reference points. In the BUA-based estimation, the reference points tracking step use multiple reference images and image rotation to deal with the problem of the reference points changing the appearance.

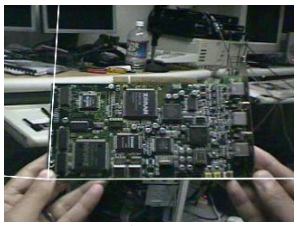
The TDA-based estimation is effective when more than half of the reference points are occluded. The TDA-based estimation tracks the target even when the BUA-based estimation cannot, but it cannot track the target for a long time. In the proposed method, TDA-based estimation works when the BUA-estimation cannot acquire accurate camera parameters. In the TDA-based estimation, the number of samples is not enough and observation is too simple to track objects for long term, because of the requirement for real time tracking.

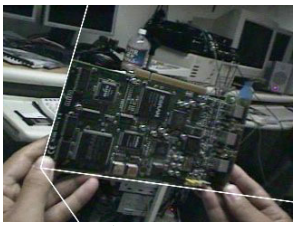
In addition, Automatic Database Addition step extends the area of tracking. The Automatic Database Addition step adds new entry when the BUA-based tracking fails to estimate the camera parameter. The proposed method can robustly track objects because these two approaches were effectively combined. An experimental result shows an accuracy and integrity of our method.

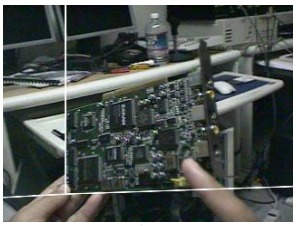


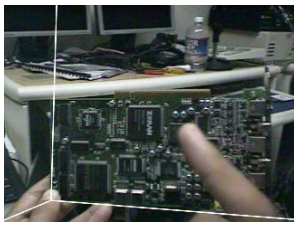

10th frame


28th frame


126thframe


144th frame


185th frame


230th frame


283rd frame

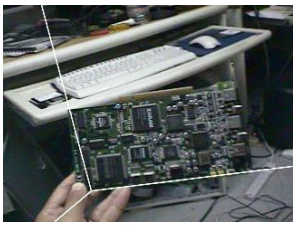

316th frame

Figure 3: error value and output stills of the experiment.

Currently, our method cannot add new reference points into the database entry because it doesn't estimate three-dimensional coordinates of new reference points. In the future, we will combine the method for shape from motion with our method to make the automatic database addition step possible to add new reference points into a database.

REFERENCES

[1] Azuma, R.T., "A survey of augmented reality," Presence, vol.6, No.4, pp.355-385, 1997.

[2] Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., and MacIntyre, B., "Recent Advances in Augmented Reality", IEEE Computer Graphics and Applications, Vol. 21, No. 6, pp. 34-47, 2001.

[3] Neumann, U., and Cho, Y., "A self-tracking augmented reality system," In Proc. VRST 96, pp. 109-115, 1996.

[4] Neumann, U., You, S., Cho, Y., Lee, J. and Park, J., "Augmented Reality Tracking in Natural Environments," Mixed Reality – Merging Real and Virtual Worlds, Ohmsha & Springer-Verlag, pp. 101-130, 1999

[5] Rekimoto, J., "Matrix: A Realtime Object Identification and Registration Method for Augmented Reality," APCHI'98, 1998.

[6] Kato, H., and Billinghurst, M., "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System," In Proc. the 2nd IEEE and ACM International Workshop on Augmented Reality '99, pp.85-94, 1999.

[7] Okuma, T., Kurata, T., and Sakaue, K., "Real-Time Camera Parameter Estimation for 3-D Annotation on a Wearable Vision System," IEICE Trans. Inf. Syst., Vol.E84-E, No. 12, pp.1668-1675, 2001.

[8] Okuma, T., Kurata, T., and Sakaue, K., "VizWear-3D: A Wearable 3-D Annotation System Based on 3-D Object Tracking using a Condensation Algorithm," In Proc. IEEE Virtual Reality 2002, pp.295-296, 2002.

[9] Simon, G. and Berger, M-O., "Reconstructing while registering: A novel approach for markerless augmented reality, In Proc. IEEE and ACM International Symposium on Mixed and Augmented Reality, pp.285-294, 2002.

[10]Isard, M. A., "Visual Motion Analysis by Probabilistic Propagation of Conditional Density," Ph.D. thesis, Department of Engineering Science, University of Oxford, 1998.

[11]Lucas, B.D. and Kanade, T., "An Iterative Image Registration Technique with an Application to Stereo Vision," Proc. DARPA Image Understanding Workshop, pp.121-130, 1981

[12]Haralick, R.M., Lee, C.-N., and Ottenberg, K. "Analysis and solutions of the three point perspective pose estimation problem," Proc. CVPR '91, pp.592-598, 1991.