

Registration Using Projective Reconstruction for Augmented Reality Systems

M.L. Yuan*, S.K. Ong** and A.Y.C. Nee*

*Singapore-MIT Alliance

**Mechanical Engineering Department

10 Kent Ridge Crescent, Singapore 119260

National University of Singapore

Abstract — In AR systems, registration is one of the most difficult problems currently limiting their applications. In this paper, we proposed a simple registration method using projective reconstruction. This method consists of two steps: embedding and tracking. Embedding involves specifying four points to build the world coordinate system on which a virtual object will be superimposed. In tracking, a projective reconstruction technique in computer vision is used to track the four specified points to compute the modelview transformation for augmentation. This method is simple as only four points need to be specified at the embedding stage, and the virtual object can then be easily augmented in a real video sequence. In addition, it can be extended to a common scenario using a common projective matrix. The proposed method has three advantages: (1) It is fast because the linear least square method can be used to estimate the related matrix in the algorithm and it is not necessary to calculate the fundamental matrix in the extended case; (2) A virtual object can still be superimposed on a related area even if some parts of the specified area are occluded during the augmentation process; and (3) This method is robust because it remains effective even when not all the reference points are detected during the augmentation process (in the rendering process), as long as at least six pairs of related reference point correspondences can be found. Several projective matrices obtained from the authors' previous work, which are unrelated with the present AR system, were tested on this extended registration method. Experiments showed that these projective matrices can also be utilized for tracking the specified points.

Index Terms—Augmented reality, registration, projective reconstruction, tracking.

I. INTRODUCTION

Augmented reality (AR) is the visual enhancement of a real scene through the integration of computer-generated information, such as computer graphics, annotations

Manuscript received October 31, 2003.

M.L. Yuan is with Singapore-MIT Alliance, the National University of Singapore, Singapore, 119260, email: smayml@nus.edu.sg.

S.K. Ong is with the Department of Mechanical Engineering, The National University of Singapore, email: mpeongsk@nus.edu.sg.

A.Y.C. Nee is with Singapore-MIT Alliance, the National University of Singapore, Singapore, 119260, email: mpeneeyc@nus.edu.sg.

and other modalities, superimposed onto the images of the real world. Although AR technology is still in its infancy, it has already been demonstrated to be a solution in many application domains. Details can be found in comprehensive surveys [7].

Registration is one of the most difficult problems currently limiting AR applications [2-6]. The virtual objects must be properly aligned with the objects in the real world, or the virtual world and the real world coexistence will be compromised. Many applications demand accurate registration. In general, registration methods can be divided into two kinds: sensor-based registration and computer vision-based registration. Conventional AR systems combine real and virtual objects using sensors to determine the relationships between the viewer (or camera), the display and the real world. These techniques often require special equipments such as magnetic field detectors, radio frequency and acoustic signals, etc., to track and measure the positions of different objects involved in the whole process. This kind of registration method often needs bulky equipments. Computer vision-based approaches, on the other hand, offer a potential for accurate registration without the need of costly equipments. This technique is able to estimate the motion and structure by analyzing the image sequences obtained from cameras at different positions, however, the main hurdle is the real-time performance.

The main purpose of the research reported in this paper is to superimpose virtual objects onto video sequences in real time. The virtual objects in the AR system reported in this paper are generally 3D graphical models. In this AR system, they are OpenGL-generated 3D models or 3DS format files that can be exported from different CAD software, such as Pro/E, AutoCAD2000, etc.

Camera calibration is generally a prerequisite for embedding virtual objects into video sequences. In most cases, the camera intrinsic parameters can be assumed to be known in advance. However, if the camera intrinsic parameters are changing, such as the focal length during 3D motion, it is very difficult to calibrate the camera at every frame. Self-calibration techniques could be considered. However, the main hurdle is the real-time performance [11, 12]. There are some reported AR systems

that do not need calibration [2, 4, 8]. Kiriakos and Kutulakos [4] proposed a calibration-free AR method based on affine object representation. Seo and Chen [2, 8] proposed another calibration-free registration method. They used the projective motion of the real camera recovered from the image matches to obtain a virtual camera containing the related intrinsic and extrinsic parameters, and the virtual camera moves according to the motion of the real camera. They used a perspective projection to generate the virtual camera for augmentation. However, the virtual camera will not always exist because there is a possibility that no solution can be obtained from the perspective matrix.

Generally, the camera intrinsic parameters will not change in most cases. Most of the reported AR systems assumed that they are known in advance [1]. Kato et al [1] provided a common AR platform (ARToolkit) including registration algorithms based on designed size-known markers, given that the intrinsic parameters are known. However, in ARToolkit, the virtual objects will be augmented over one of the square markers. If a part of the marker is occluded, the virtual object will not be augmented. Hence, the issue is whether any virtual objects can be superimposed over any planar structure or object in the real scene. The problem, in fact, is the tracking of a planar structure during the entire augmentation process. In computer vision, it is very difficult to track points located on a surface because of the aperture problem. Seo et al [2] reported a projective method and algorithm to obtain the updated image points during augmented simulation. Our work is motivated by the work of Seo et al [2] and Kato et al [1]. There are two differences between our method and the method reported by Seo et al [2]. Firstly, the method reported by Seo et al [2] is calibration-free while we assume that the intrinsic parameters are known in advance as it is logical in most cases. For example, in AR-assisted assembly and augmented NC simulation, calibration-free augmentation is not necessary. In addition, the method proposed by Seo et al [2] cannot guarantee the existence of the virtual camera. Secondly, only four points are specified at the initial stage to define the X -axis and the Y -axis of the graphical world coordinate system. However, the method proposed by Seo et al [2] requires five points to specify the coordinate system in the control images. It is easier and more accurate to specify four points as an approximate square than five points as a cube in the images. Meanwhile, Seo et al [2] used these points to compute a virtual perspective matrix and not a real perspective matrix. Thus, there is a homography between this matrix and the real perspective matrix. Our method uses Euclidean Space directly to calculate the perspective matrix. Furthermore, our method can be extended to a more common case of using general projective matrices without computing the related fundamental matrix, which is time-consuming. After we have tracked these specified points, the method reported by Kato et al. [1] is used for computing the

modelview transformation in our AR system.

Our proposed method has three advantages: (1) It is fast because only the linear least square method is used to estimate the related matrix in the algorithm, and it is not required to calculate the fundamental matrix in the extended case; (2) The virtual object can still be superimposed on a related area even if some parts of the specified area are occluded during the entire augmentation process; and (3) the proposed method is robust because it remains effective if at least six pairs of reference point correspondences can be found, when not all the reference points are detected during the augmentation process (rendering process).

The remaining of this paper is organized as follows. Section 2 provides some background and notations. Section 3 presents the embedding method. Section 4 presents in details the proposed new registration method using projective reconstruction. The extension of this method to a general case is discussed in Section 5. Section 6 shows some experimental results of this proposed method and the general case. Section 7 discusses its performance and some applications in manufacturing. Finally, conclusions are given in the last section.

II. BACKGROUND

A. Camera Model

An image point is denoted by $m = (u, v)^T$. A 3D point is denoted by $M = (X, Y, Z)^T$. In this paper, homogeneous vectors are used to represent 2D and 3D points respectively as follows:

$$m = (u, v, 1)^T, M = (X, Y, Z, 1)^T \quad (1)$$

The relationship between a 3D point M and its image projection m is generally given as follows:

$$\rho m = A[R | t]M \quad (2)$$

where ρ is an arbitrary factor, (R, t) is the rotation and translation that relates the world coordinate system to the camera coordinate system, and is generally called the extrinsic parameters. A , which is called the camera intrinsic matrix, is given by

$$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where (u_0, v_0) is the coordinates of the principal point, α and β are the scale factors in the image u and v axes, and γ is the skew factor of the image axes. In this paper, this parameter is assumed to be zero. Based on the requirement that M should have rank 3, A is non-singular.

B. Projective Reconstruction

As commonly known, there is a geometric constraint in the correspondence problem between two images, which is

the epipolar geometry [9]. Let M_k be a 3D point, m_{ik} and m_{jk} be its projections to image I and image J , respectively. It is well known that at least eight point matches are needed for computing an essential matrix. The essential matrix is called a fundamental matrix in the case of two uncalibrated images. The fundamental matrix between image I and image J is a 3×3 matrix satisfying equation (3):

$$m_{jk}^T F_{ij} m_{ik} = 0, k = 1, \dots, n \quad (3)$$

for any matched pair (m_{ik}, m_{jk}) in the two images I and J . The fundamental matrix F_{ij} indicates the epipolar geometry of the two cameras.

Now, let F be the fundamental matrix between two images. There are infinite numbers of projection bases that satisfy the epipolar geometry. One possibility is to factor F as a product of an anti-symmetric matrix $[e']_x$ and a matrix M , i.e., $F = [e']_x M$. In fact, e' is the epipole in the second image. Next, two projective camera matrices can be represented canonically as follows:

$$P = [I | 0], P' = [M | e'] \quad (4)$$

The factorization of F into $[e']_x M$ is in general not unique because if M is a solution, then $M + e' v^T$ is also a solution for any vector v^T . M is defined as $[e']_x F$.

Given a pair of matched points in two images: (m_{1i}, m_{2i}) , let $M_i = [X, Y, Z, 1]^T$ be the corresponding projective 3D point. Following the pinhole model, the following equations can be obtained:

$$s m_{1i} = P M_i \quad (5)$$

$$s' m_{2i} = P' M_i \quad (6)$$

where s and s' are two arbitrary scalars. Let p_i and p'_i be the vectors corresponding to the i^{th} row of P and P' respectively. The two scalars can then be computed as follows:

$$s = p_3^T M_i \quad (7)$$

$$s' = p'_3{}^T M_i \quad (8)$$

Using equations (5) to (8), M_i can be reconstructed from its image matches (m_{1i}, m_{2i}) using the linear least square technique.

On the other hand, for the k^{th} image, its corresponding projective matrix P_k can be computed if there are enough number of pairs of (m_{ki}, M_k) of the image points and their corresponding 3D projective points are given. We can assume the value of P_{23} of P_k to be 1, where P_{23} is the element of P_k at row 3 and column 4. There are 11

unknown elements in P_k . Given that for one pair of (m_{ki}, M_k) , there are two equations. Hence, at least six pairs of (m_{ki}, M_k) are needed to compute the projective matrix using the linear least square method. Conversely, if a 3D projective point is known, the estimated projective matrix can be used to compute the image coordinates. Hence, if a 3D projective point is known at the initial stage using two images, its projection is thus determined using the linear least square method. Therefore, the image projections of 3D projective points can be obtained from the updated projective matrix during the augmentation process. During the complete video sequence, this method can be used to track the specified points on which the virtual object can be augmented.

III. EMBEDDING

Before embedding, two control images are first selected to detect the feature points and find their matches in the two control images. In our AR system, the marker detection method in ARToolkit is used to find the related markers and the feature points. These features points will be the reference points in the entire augmentation process. It is noted that it is not necessary to obtain all the reference points to compute the updating projective matrix. Only a minimum of six points need to be detected. Of course, if more reference points are detected, the robustness of the algorithm will be improved. Next, the Hartley 8-point method [9, 10] is used to compute the fundamental matrix. Projective camera matrices P and P' of the two control images are then computed respectively. Based on P and P' , the 3D projective coordinates $M_i (i = 1, \dots, N)$ of the reference points can be obtained.

The next step is to specify four corresponding points $\{x_{0i}\} (i = 1, \dots, 4)$ in the two control images respectively, in order to insert the X, Y -axes of the graphical world coordinate system where the virtual object will be superimposed. These four points will form an approximate square. The origin of the world coordinate system will be the center of the square formed by the four specified points. The Z -axis will be the vertical direction of the XY -plane [1] (Figure 1).

After four point matches have been specified in the two control images, the related projective 3D coordinates $\{X_i\} (i = 1, \dots, 4)$ of these specified square points are computed to determine their locations in the other images in the video sequence. Next, these projective 3D points are further used in the tracking process using the re-projection technique as discussed in Section 2. The location and pose of a virtual 3D object are next defined with respect to the specified world coordinate system that is now embedded in the camera system.

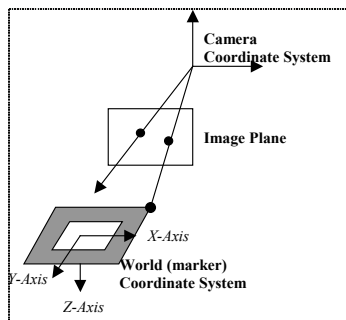


Figure 1. The relationship between the world coordinate system and the camera coordinate system

IV. TRACKING USING PROJECTIVE RECONSTRUCTION

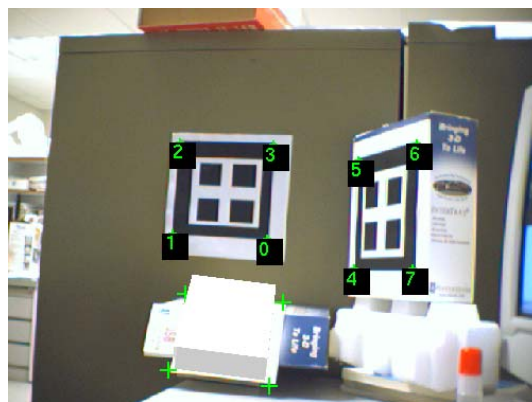
After the initial embedding process, the next step is to track the camera position when it is moving. Once the 3D projective coordinates of the points on a virtual object are known, the projection of these 3D points can be computed easily using the updated projective matrix, which could be computed based on the tracked reference points.

The algorithm is as follows:

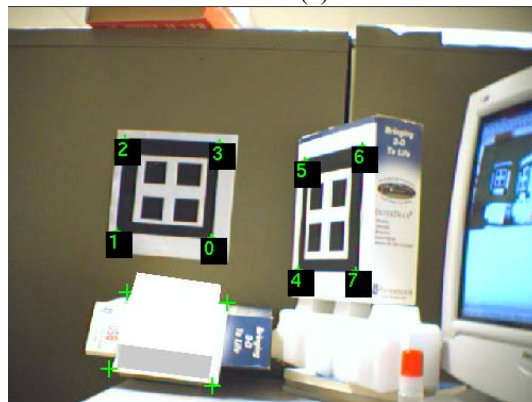
- (1) For the k^{th} image, find the reference points $\{m_{ki}\} (i = 1, \dots, N)$ using the marker detection method in ARToolkit;
- (2) Using these detected reference image points $\{m_{ki}\} (i = 1, \dots, N)$ and the computed 3D projective points $M_i (i = 1, \dots, N)$ at the embedding stage, compute the projective matrix P_k of the k^{th} image;
- (3) Using the computed P_k in Step 2 and the 3D projective coordinates $\{X_i\} (i = 1, \dots, 4)$ of the four specified points at the embedding stage, compute their corresponding projections $\{x_{ki}\} (i = 1, \dots, 4)$;
- (4) After obtaining these image points $\{x_{ki}\} (i = 1, \dots, 4)$, use the method in [1] to calculate the modelview transformation, i.e., R_k and t_k , which are the rotation and translation between the camera coordinate system and the inserted world coordinate system;

Since the camera intrinsic matrix A is known in advance, using the above computed R_k and t_k , the graphics rendering procedure can be performed using OpenGL. The virtual object will then be aligned with the video sequence.

In the above algorithm, the intrinsic camera parameters determine the perspective viewing volume and (R_k, t_k) determines the model transformation when using OpenGL to render in a graphics system, as shown in Figure 2.



(a)



(b)

Figure 2. An example to track the four specified points using projective reconstruction techniques; (a) is the 57th Image, and (b) is the 108th image in the whole video sequence with the virtual object.

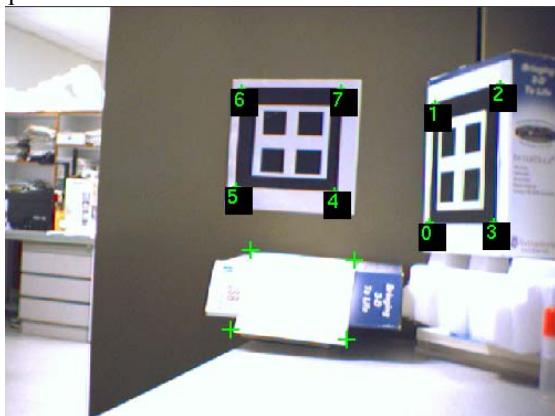
V. EXTENSIONS

Careful analysis and several experiments have shown that this method can be extended to a general case. As long as $M_i (i = 1, \dots, N)$ and $\{X_i\} (i = 1, \dots, 4)$ can be solved using equations (5) and (6) from any projective 3x4 matrix P' at the initial stage, this matrix can be used for the registration algorithm presented in this paper. Several examples have been tested with some matrices that have been obtained from two other images that are unrelated with our AR system. The results showed that it is also applicable using the general projective matrix. Hence, users do not need to calculate the fundamental matrix, and only need to load one of the projective matrices from different resources. In these examples, only the tracked points are shown, which form a square and the virtual object will be superimposed on it. One of general perspective matrix, which had been obtained from two images that are completely unrelated with this video sequence, is as follows:

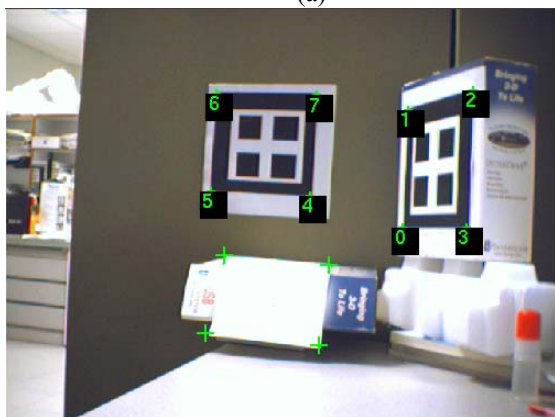
$$P = \begin{bmatrix} 0.001406 & -1.359584e-005 & -0.409455 & 291.535602 \\ -16.354082 & 133.594019 & -291.585260 & 37.7215207 \\ -0.409481 & -0.049456 & 121.243935 & 1 \end{bmatrix}$$

Figure 3 shows several examples of the results using a general projective matrix that has been obtained from two images that are completely unrelated with this video

sequence.



(a)



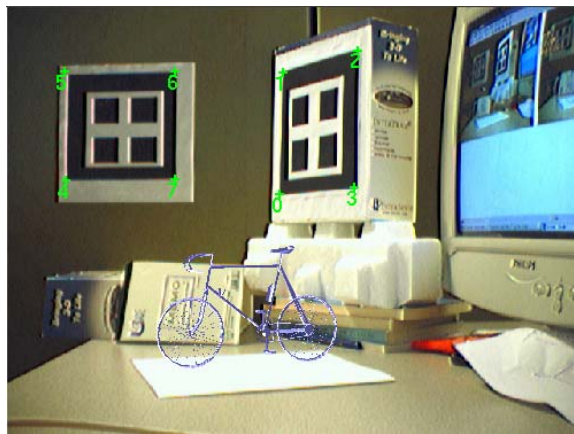
(b)

Figure 3. The results of tracking specified points (R_0 , R_1 , R_2 and R_3) using a general projective matrix; (a) is the 45th image and (b) is the 78th image in the whole video sequence.

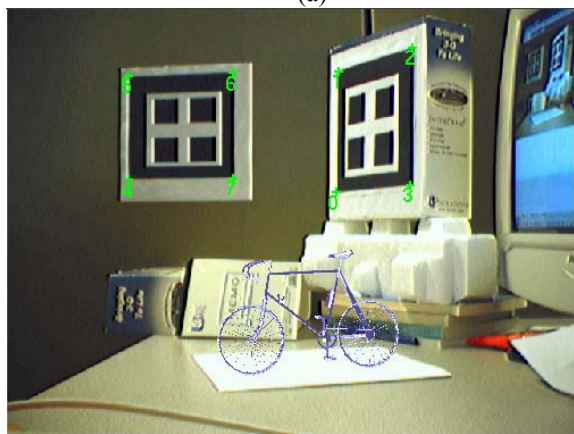
VI. EXPERIMENTS

The proposed registration method has been implemented using visual C++ and some experiments have been conducted to verify the algorithm described in this paper. The video sequence was captured with an IEEE 1394 FireFly camera in which 40 reference points were tracked to estimate the projective structure of the moving camera. The image size is 640x480. Two control images of the video sequence were first selected. In this section, two experiments are described, namely (1) to test the method described in Sections 3 and 4; and (2) to test the extended method.

In the first experiment, the Hartley 8-point algorithm [9, 10] was used to compute the fundamental matrix. Projective camera matrices for the two views were computed and the 3D projective coordinates were computed for the 40 reference points. Some examples can be seen in Figure 4. In these AR examples, four points were picked, which are the four corners of the white paper and form a world coordinate system. These four points were tracked using the projective matrix.



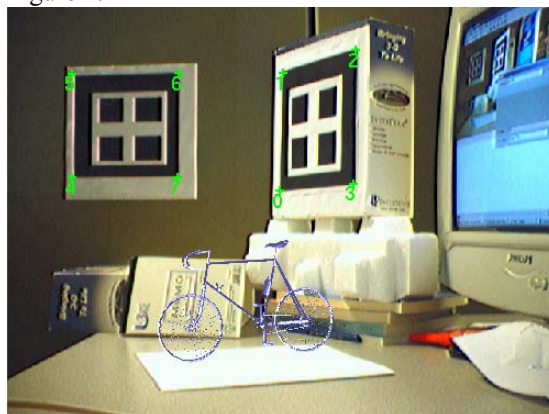
(a)



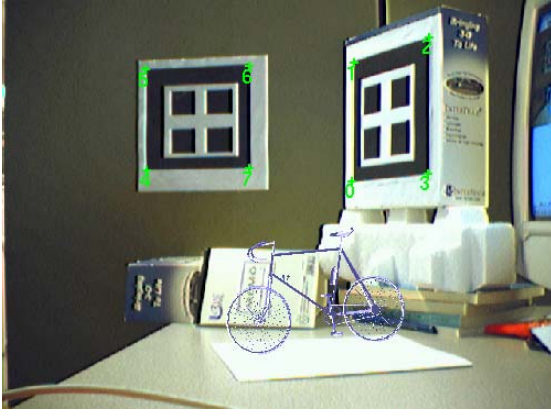
(b)

Figure 4. Examples using the fundamental matrix. This figure shows the results of tracking specified points (the four corner points of the white paper) using the method described in Section 4; (a) is the 47th image and (b) is the 99th image in the whole video sequence.

In the second experiment, the extension of the registration algorithm was tested without computing the fundamental matrix. A projective matrix from other unrelated resources was loaded for the proposed registration method. The projective matrix used here is the same as described in Section 5. Some examples can be seen in Figure 5. The results of these examples are similar to Figure 4.



(a)



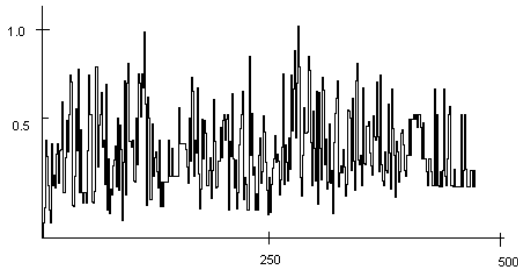
(b)

Figure 5. Examples using a general projective matrix which is unrelated with this video sequence.

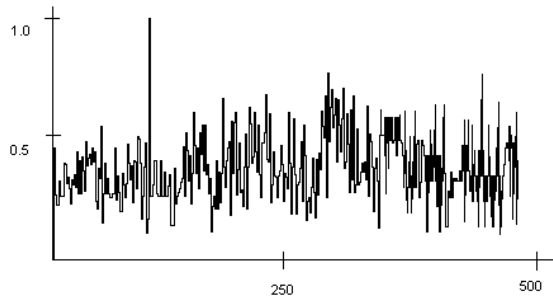
For these two experiments, the errors between the real reference points and their re-projections using projective reconstruction were compared. The errors are described as follows:

$$error = \frac{\sum_{i=1}^N \|m_{ki} - \hat{m}_{ki}\|}{N} \quad (9)$$

where \hat{m}_{ki} are the estimated points (tracked points) obtained using equations (5) and (6), and $\|\bullet\|$ is the Euclidean distance between two image points. Figure 6 shows the graphs of the error descriptions between the real reference points and their re-projections. All the errors are below 1.0 pixel error. These experiments demonstrated the accuracy of the tracked points.



(a)



(b)

Figure 6. Error estimations of the first and second experiments. The abscissa axis denotes the image frame of the video sequence and the ordinate axis denotes the magnitude of errors in the unit of pixel.

VII. DISCUSSIONS AND POTENTIAL APPLICATIONS IN MANUFACTURING

In this paper, a registration method for AR systems has been proposed based on the projective reconstruction technique in computer vision. This method can also be extended to a general case where it is not necessary to compute the fundamental matrix for determining the camera projective matrix. These matrices can be loaded from a designed projective matrix database.

Real-time implementation is an important issue for AR systems. The proposed method can meet real-time performance because the fast linear least square method can be used to estimate the related matrices for augmentation. For detecting the reference points, the fast marker detection method in ARToolkit is used. Of course, when the virtual CAD models are very large, it will affect the real-time performance of the augmented simulation, especially if the virtual objects rotate or translate along some directions. Hence, the related geometric information filtering techniques for display and operation should be further researched and used in order to improve the speed of the system. For example, in Delmia, it uses a special simulation data structure that is different from the traditional CAD/CAM software data structures. This area of study could form our future work.

From the experimental observation and the analysis of the projective matrices that are used for the tracking process, the results of some of the tested projective matrices are not good because the new projective matrix has to be determined for computing the tracked image points. From equations (7) and (8), if $s = p_3^T M_i$ and $s' = p_3'^T M_i$ are very small, it is apparent that the tracked points will be prone to errors, which will propagate and affect the registration results. Further work will be conducted to determine how projective matrices affect the results of the tracking points, and the registration results.

This proposed method could be used in AR-assisted systems in manufacturing, such as assembly, product evaluation and augmented NC simulation. If some components could be recognized, virtual CAD models could then be superimposed on the related components automatically without specifying a square at the embedding stage. This is very useful for AR applications in these areas. In assembly, this visualization method can be used to guide the assembly and provide some instructions on the following steps even if the operator's hand occludes the related parts of the assembly components. In product evaluation, when a component is recognized, the proposed registration method can be used to align the virtual CAD model with the real object to evaluate the properties of the design prototype. In augmented NC simulation, the proposed method can be used for augmentation. In AR NC simulation, one would see the real setting of the machine and the cutting tool, and the virtual workpiece will be

simulated and superimposed on the worktable using the proposed registration method, where the user will be asked to specify four points. The real cutting tool will remove material from the virtual workpiece, following the pre-set tool paths. All the fixturing and clamping arrangements will also be superimposed on the virtual workpiece using this proposed method. These are our on-going projects using AR techniques. For AR applications in manufacturing, another important issue is real-time object recognition. This issue is beyond the scope of this paper.

VIII. CONCLUSIONS

In this paper, a simple registration algorithm is proposed based on the projective reconstruction technique in computer vision, and it is useful for realizing the augmented simulation between the views of a real scene and a virtual object in an image-based rendering framework. This method can also be extended to a more general case. It assumes that the intrinsic camera parameters are known in advance since they can be found using various methods and they will not change in most cases, especially in many manufacturing areas. The method is effective for augmentation by simply specifying four planar points on which the virtual objects will be superimposed. If the planar structures can be recognized using some computer vision techniques, the virtual object can be superimposed automatically without having to specify the four basic points. In future work, this method will be applied in AR applications in manufacturing, which are our on-going projects, including AR-assisted assembly and augmented NC simulation. Meanwhile, our future work will further this method without using the markers as a reference.

REFERENCES

- [1] Hirokazu Kato, Mark Billinghurst, "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System", *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, Oct, 1999, San Francisco, pp 85-94
- [2] Yongduek Seo, Ki Sang Hong, "Calibration-free Augmented Reality in Perspective", *IEEE Trans on Visualization and Computer Graphics*, Vol. 6, No. 4, pp 346-359, Oct-Dec. 2000
- [3] Simmon J.D. Prince, Ke Xu, Adrian David Cheok. "Augmented Reality Camera Tracking with Homographies", *IEEE Computer Graphics and Applications*, Vol. 22, No. 6, pp 39-45, Nov-Dec 2002
- [4] Kiriakos N. Kutulakos, "Calibration-Free Augmented Reality", *IEEE Trans on Visualization and Computer Graphics*, Vol. 4, No. 1, pp 1-20, Jan-March. 1998
- [5] Vijaimukund R, Molineros J, Sharma R, "Interactive Evaluation of Assembly Sequence Using Augmented Reality", *IEEE Trans. On Robotics and Automation*, Vol. 15, No. 3, pp 435-449, June 1999
- [6] Satoh K, Anabuki M, Yamamoto H, Tamura H, "A hybrid registration method for outdoor augmented reality". *Proceedings of IEEE and ACM International Symposium on Augmented Reality*, 29-30 Oct. 2001, pp 67-76
- [7] Azuma, Ronald T. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 1997, Vol. 6, No. 4, pp 355 – 385
- [8] Chu-Song Chen, Chi-Kuo Yu, Yi-Ping Hung, "New Calibration-free approach for augmented reality based on parameterized cuboid structure", *Proc. 7th Int'l conf. Computer vision*, 1999, pp 30-37

- [9] Zhengyou Zhang, Rachid Deriche, Olivier Faugeras, Quang-Tuan Luong, "A Robust Technique for Matching Two Uncalibrated Images through the Recovery of the Unknown Epipolar Geometry", *Artificial Intelligence J.*, Vol. 78, pp 87-119, 1995
- [10] R. I. Hartley, "In Defense of the Eight-Point Algorithm", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 19, no. 6, pp.580-593, June 1997
- [11] Q.-T. Luong and O. Faugeras, "Self-calibration of a moving camera from point correspondences and Fundamental matrices", *International Journal of Computer vision*, Vol. 22, No. 3, pp 261-289, 1997
- [12] Maybank, S. J, O. Faugeras, "A theory of self-calibration of a moving camera", *International Journal of Computer vision*, Vol. 8, No. 2, pp 123-151, 1992

M.L. Yuan is a research fellow in the National University of Singapore under the Singapore-MIT Alliance (SMA). He received his PhD degree from Huazhong University of Science and Technology (HUST) in 1997. His current research interests are augmented reality and its applications in manufacturing.

S.K. Ong is an assistant professor in the Manufacturing Division, Department of Mechanical Engineering, National University of Singapore since 1995. She is the program manager for the Virtual Manufacturing Program in the Laboratory for Concurrent Engineering and Logistics (LCEL) in the Faculty of Engineering. Her research interests are VR and AR applications in manufacturing, intelligent and distributed manufacturing systems, computer-aided set-up planning, life cycle engineering, and environment impact assessment. She has published 2 books, and over 65 international refereed journals and conference papers.

A.Y.C. Nee is a professor of manufacturing engineering, Department of Mechanical Engineering, National University of Singapore since 1989. He received his PhD and DEng from UMIST in 1973 and 2002 respectively. He is presently the Co-Director of the Singapore-MIT Alliance (SMA) Program and CEO of the Design Technology Institute (DTI). His research interest is in computer applications to tool, die, fixture design and planning, intelligent and distributed manufacturing systems, and application of VR and AR techniques in manufacturing. He has published 4 books and over 450 papers in refereed journals and conference presentations. He currently held regional and associate editorship, and member of editorial board of 14 international journals in the field of manufacturing engineering. He is an active member of CIRP and a Fellow of the Society of Manufacturing Engineers, both elected in 1990.