Multimed Tools Appl (2009) 41:235–252
DOI 10.1007/s11042-008-0227-y

Marker-less registration based on template tracking for augmented reality

Liang Lin · Yongtian Wang · Yue Liu ·
Caiming Xiong · Kun Zeng

Published online: 22 October 2008

Abstract Accurate 3D registration is a key issue in the Augmented Reality (AR) applications, particularly where are no markers placed manually. In this paper, an efficient markerless registration algorithm is presented for both outdoor and indoor AR system. This algorithm first calculates the correspondences among frames using fixed region tracking, and then estimates the motion parameters on projective transformation following the homography of the tracked region. To achieve the illumination insensitive tracking, the illumination parameters are solved jointly with motion parameters in each step. Based on the perspective motion parameters of the tracked region, the 3D registration, the camera's pose and position, can be calculated with calibrated intrinsic parameters. A marker-less AR system is described using this algorithm, and the system architecture and working flow are also proposed. Experimental results with comparison quantitatively demonstrate the correctness of the theoretical analysis and the robustness of the registration algorithm.

Keywords Augmented reality · Marker-less registration · Virtual tracking · Motion estimation

L. Lin (✉) · Y. Wang · Y. Liu
School of Information Science and Technology, Beijing Institute of Technology,
Beijing 100081, China
e-mail: linliang@bit.edu.cn

Y. Wang
e-mail: wyt@bit.edu.cn

Y. Liu
e-mail: bithxm@bit.edu.cn

C. Xiong · K. Zeng
Institute for Pattern Recognition and Artificial Intelligence,
Huazhong University of Science and Technology,
Wuhan 430074, China

C. Xiong
e-mail: Cmxiong.lhi@gmail.com

K. Zeng
e-mail: zengkun@gmail.com

1 Introduction

Augmented Reality (AR) is a new computer technology that combines virtual computer-generated 3D graphics with realistic environments in natural visual perception. It has been widely used in industrial applications, such as virtual medical surgery, virtual computer aided education, military, industry, and entertainment [2, 22, 25]. The key problem in AR system is the accurate and robust 3D registration, which entails aligning virtual objects with a real environment in 3D coordinates. In general, the registration process can be realized in the following three steps: positioning, rendering, and merging [18]. Positioning entails transforming and rotating the virtual objects relative to the observer's location. Rendering means computing the projected 2D image from the 3D model, which is the real image observed by the user. Merging is an image processing procedure to merge the virtual objects with the real environments in order to make them look like real parts of the scene.

A real-time registration method for a markerless AR system is proposed in this paper, which combines fixed region tracking and perspective motion estimation. A texture-full region is first selected manually and stored as the reference template for initialization. When the camera undergoes freely motion, the region can be tracked in real time, and its position and pose can be estimated, combining the camera intrinsic parameters. Considering the computing efficiency, a set of sparse points is randomly sampled to represent the region.

Hence the proposed registration process can be divided into two parts. First the tracking is achieved by minimizing the sum-of-squared differences (SSD) between the reference template and the target region. The next step is to compute the homography of the tracked region and estimate its position and pose. With robust 3D registration, the virtual generated model can be rendered with correct pose and orientation and merged to the real scene seamless. The AR system and its working flow are also proposed.

The main contributions of the presented registration algorithm and AR system are: (1) using the informative tracked region to replace manual markers in previous AR system [1, 2, 11, 17, 19, 25], which enhances the system applicability and robustness. (2) Discuss the illumination insensitive tracking by incorporating illumination parameters estimation with tracking process. (3) Propose the integrated solution of 3D registration based on template tracking and the camera intrinsic parameters.

The rest of this paper is organized as follows. A number of related works are introduced in section 2. Section 3 then explains the principle of 3D registration, including illumination insensitive template tracking and perspective motion estimation, and section 4 describes the AR system with this registration approach. The experimental results with comparisons are showed in section 5, and the final conclusion is drawn in the last section 6.

2 Related works

3D registration for AR system is a challenging problem, particularly for lacking manual markers in the scenes. There are three categories of registration methods, e.g. registration based on direction-finding equipments, such as GPS (Global Positioning System) and gyrometer for computing the 6DOF (Degree of Freedom) of a user [4, 16, 24], registration based on computer vision methods [3, 7, 11, 18, 19, 23] and hybrid registration to combine the virtue of both hardware and computer vision [7, 16, 24]. All of these registration methods have their own advantages and shortcomings. For example, registration with GPS and gyrometer is fast and robust but generally has low accuracy. Many computer vision registration methods used in AR systems require placing markers in the scene beforehand.

These markers are designed for easily detection, through giving them distinctive photometric characteristics (e.g. colored markers [16], LEDs [19], or special shapes (like squares [17, 19], dot codes [11] or even 2D barcodes [1])). Such approaches have been successfully applied in some AR systems [14, 23]. However, manual markers narrow the applicability of the AR systems, and the easy-designed markers are always not robust enough against environment noise. In addition, the systems sometimes crash when some of markers are occluded.

In last a few years, the researches attempt to track patches of the natural scene as landmarks, to achieve markerless registration [5, 10, 13, 16, 17, 20].

The traditional region tracking approaches can be divided into two categories: approaches using local independent correspondences (feature-based approaches [11, 16, 17, 20]) and those approaches using template correspondences (template-based approaches [5, 13]). The former uses the local features, such as the key points, line segments, and structure primitives, and the latter uses the texture-full image patches as a whole. Although feature-based methods have the advantage of fast computing, the strength of these global methods lies in their ability to handle complex patterns that cannot be modeled by local features.

For the template-based approaches, an L_2 norm is generally used to measure the error between a reference template and a candidate region [9]. Historically a brute force search was used to match the template [16, 18]. However, this strategy is impractical under perspective transformations, a situation that involves higher dimensional parameter spaces than simple 2D translation. More recent methods treat the problem as a nonlinear optimization problem using Newton type or Levenberg–Marquardt based algorithms [12]. However, they can not be implemented in real time due to their expensive nonlinear computations.

The landmarks correspond to regions that are "good feature" according to the criteria of Shi and Tomasi [21]. Recently, some region tracked based registration methods [9, 18–20] were proposed for both outdoor and indoor AR system, and they achieve the real-time performance with fixed small regions. However, they have two non-trivial weaknesses: (1) The tracking error is estimated by the gray intensity of tracked region (template), and thus the registration is sensitive to illumination; (2) They only perform well for 2D virtual object augmentation, due to solving the registration by estimating the affine motion transformation of the tracked region.

3 Principle of 3D registration

We first introduce the fixed template tracking method, the projective motion estimation via region tracking then is described, and the 3D environment registration is proposed finally.

3.1 Fixed template tracking

Since a weak displacement of the camera should result in a change in pixel intensities, we can determine the motion of the moving region from these intensity variations, assuming that the capture rate of the camera is high enough.

We define some notations first. Let $I(\mathbf{X},t)$ be the brightness value at the location $\mathbf{X}=(x, y)^t$ in an image acquired at time t and $\nabla I_{\mathrm{X}}(\mathrm{X}, t)$ be its intensity gradient. Let the set $R=\{X_1,X_2,X_3...X_N\}$ be the set of N image locations which define a target region. $\mathbf{I}(R,t)=(I(X_1,t),I(X_2,t,),...,I(X_N,t))$ is a vector of brightness values of the target region at time t and $\mathbf{I}(R,t_0)$ is referred as the reference template, which is the template to be tracked. t_0 is the initial time (t=0). In the process of tracking, the relative motion between the camera and the scene results in a deformation of the tracking target. Therefore, a model $f(\mathbf{X};\boldsymbol{\mu})$ is adopted to describe the motion of the target, where $\boldsymbol{\mu}$ is a vector modeled by n parameters, obviously, $f(\mathbf{X};0)=\mathbf{X}$ and $N>n$. Thus the tracking process can be reduced to motion vector computation in every frame related to the variations in intensity. Suppose $\boldsymbol{\mu}^*(t)$ is the true value of the motion vector at time t and $\boldsymbol{\mu}^*(t)=\boldsymbol{\mu}^*(t)=0$ and at arbitrary time $t>t_0$ we have:

$$\mathbf{I}(\mathrm{X},t_0)=\mathbf{I}(f(x,\mu*(t)),t) \tag{1}$$

Eq. (1) is the equation denotes gray-scale invariable attribute in region tracking. Least-squares can be used to estimate the motion parameters at time $t>t_0$ as:

$$O(\boldsymbol{\mu})=\|(\mathbf{I}(f(\mathbf{X},\boldsymbol{\mu}),t)-\mathbf{I}(\mathbf{X},t_0))\|^2 \tag{2}$$

In order to simplify notations, $\mathbf{I}(f(\mathbf{X},\boldsymbol{\mu}),t)$ is denoted by $\mathbf{I}(\boldsymbol{\mu},t)$, which describes the intensity of the target at time $t>t_0$. Assuming $\boldsymbol{\mu}=0$ at time t_0, Eq. (2) can be simplified as:

$$O(\boldsymbol{\mu})=||(\mathbf{I}(\boldsymbol{\mu},t)-\mathbf{I}(0,t_0))||^2 \tag{3}$$

In general, Eq. (3) is a non-convex objective function. Lacking a good starting point, this problem will usually require some type of time-consuming global optimization procedure [12]. However, in the case of the tracking problem, an exact value of target position and orientation can be obtained before tracking, and at time $t>t_0$, the target motion can be described as $\boldsymbol{\mu}(t)$. Therefore, the problem can be recast to the process of computing $\delta\boldsymbol{\mu}$, which denotes the variable of motion parameters, where $\boldsymbol{\mu}(t+\tau)=\boldsymbol{\mu}(t)+\delta\boldsymbol{\mu}$. In this case, Eq. (3) can be transformed to

$$O(\delta\boldsymbol{\mu})=\|\mathbf{I}(\boldsymbol{\mu}(t)+\delta\boldsymbol{\mu},t+\tau)-\mathbf{I}(0,t_0)\|^2 \tag{4}$$

Using a high-capture-rate camera can ensure the deformation between frames is small, which means the reduction of $\delta\boldsymbol{\mu}$ is also small. Thus, the linearization can be carried out by expanding $\mathbf{I}(\boldsymbol{\mu}(t)+\delta\boldsymbol{\mu},t+\tau)$ using a Taylor series as:

$$\mathbf{I}(\boldsymbol{\mu}+\delta\boldsymbol{\mu},t+\tau)=\mathbf{I}(\boldsymbol{\mu},t)+\mathbf{M}(\boldsymbol{\mu},t)\delta\boldsymbol{\mu}+\tau\times\mathbf{I}_t(\boldsymbol{\mu},t)+\text{h.o.t} \tag{5}$$

where h.o.t denotes higher-order terms of the expansion and $\mathbf{M}(\boldsymbol{\mu},t)$ is the Jacobian matrix of the captured image, which corresponds to motion parameters and intensity variables for the target. $\mathbf{M}(\boldsymbol{\mu},t)$ is an $N\times n$ matrix of partial derivatives, because the number of sparse points is N and the dimension of motion parameters are n. Each element of this matrix is given by

$$m_{ij}=\mathbf{I}_{\boldsymbol{\mu}_j}(f(\mathbf{X}_i;\boldsymbol{\mu}),t)=\nabla_f\mathbf{I}(f(\mathbf{X}_i;\boldsymbol{\mu}),t)^t f_{\boldsymbol{\mu}}(\mathbf{X}_i;\boldsymbol{\mu}) \tag{6}$$

where $\nabla_f\mathbf{I}$ is the gradient of the target with respect to the motion model.

Combining Eq. (5) with Eq. (4) and neglecting the higher-order terms, we have:

$$O(\delta\boldsymbol{\mu})=\|\mathbf{I}(\boldsymbol{\mu},t)+\mathbf{M}\delta\boldsymbol{\mu}+\tau\times\mathbf{I}_t-\mathbf{I}(0,t_0)\|^2 \tag{7}$$

Letting $\tau\times\mathbf{I}_t(\boldsymbol{\mu},t)\approx\mathbf{I}(\boldsymbol{\mu},t+\tau)-\mathbf{I}(\boldsymbol{\mu},t)$, Eq. (7) can be simplified to:

$$O(\delta\boldsymbol{\mu})\approx\|\mathbf{I}(\boldsymbol{\mu},t+\tau)+\mathbf{M}\delta\boldsymbol{\mu}-\mathbf{I}(0,t_0)\|^2 \tag{8}$$

To maximize the right side of Eq. (8), $\delta\boldsymbol{\mu}$ can be expressed as

$$\delta\boldsymbol{\mu} = -(\mathbf{M}^t\mathbf{M})^{-1}\mathbf{M}^t[\mathbf{I}(\boldsymbol{\mu}, t+\tau) - \mathbf{I}(0, t_0)] \tag{9}$$

Eq. (9) is the basic model for target tracking. From this model, the Jacobian constraint should be computed in every frame for $\delta\boldsymbol{\mu}$. Because $\mathbf{M}(\boldsymbol{\mu},t)$ depends on time-varying quantities, it may appear that it must be recomputed at each time step, which is a computationally expensive procedure. Therefore, Eq. (6) should be analyzed in another way to obtain the constant part of $\mathbf{M}(\boldsymbol{\mu},t)$ and compute it before the tracking process.

The gradient expressions are used to analyze $\nabla_f\mathbf{I}$ for the purpose of simplifying $\mathbf{M}(\boldsymbol{\mu},t)$, as shown in Eq. (10):

$$\nabla_{\mathrm{X}} I(\mathbf{X}, t_0) = \mathrm{f}_{\mathrm{X}}(\mathbf{X};\boldsymbol{\mu})^t \nabla_f \mathbf{I}(\mathrm{f}(\mathbf{X};\boldsymbol{\mu}), t) \tag{10}$$

By substituting Eq. (10) into Eq. (6), we have:

$$\mathbf{M}(\boldsymbol{\mu}) = \begin{pmatrix} \nabla_{\mathrm{X}}\mathbf{I}(\mathbf{X}_1, \boldsymbol{\mu}_0)^t \mathrm{f}_{\mathrm{X}}(\mathbf{X}_1, \boldsymbol{\mu})^{-1} \mathrm{f}_{\mu}(\mathbf{X}_1, \boldsymbol{\mu})^{-1} \\ \nabla_{\mathrm{X}}\mathbf{I}(\mathbf{X}_2, \boldsymbol{\mu}_0)^t \mathrm{f}_{\mathrm{X}}(\mathbf{X}_2, \boldsymbol{\mu})^{-1} \mathrm{f}_{\mu}(\mathbf{X}_2, \boldsymbol{\mu})^{-1} \\ \cdots\cdot \\ \nabla_{\mathrm{X}}\mathbf{I}(\mathbf{X}_N, \boldsymbol{\mu}_0)^t \mathrm{f}_{\mathrm{X}}(\mathbf{X}_N, \boldsymbol{\mu})^{-1} \mathrm{f}_{\mu}(\mathbf{X}_N, \boldsymbol{\mu})^{-1} \end{pmatrix} \tag{11}$$

where $f_X(\mathbf{X}_i;\boldsymbol{\mu})$ is the partial derivative of the motion model by with respect to $\mathbf{X}_\mathrm{i}$ and $f_{\boldsymbol{\mu}}(\mathbf{X}_i;\boldsymbol{\mu})$ is the partial derivative of the motion model with respect to $\boldsymbol{\mu}$. Obviously, $\nabla_{\mathrm{X}}\mathbf{I}(\mathbf{X}_i, \boldsymbol{\mu}_0)^t$ is constant over time while $f_{\boldsymbol{\mu}}(\mathbf{X}_i;\boldsymbol{\mu})$ is the time-varying part. However, $f_X(\mathbf{X}_i;\boldsymbol{\mu})$ is partially related to time and the constant part is supposed to be $\Gamma(\mathbf{X}_i)$. Eq. (11) can be rewritten as:

$$\mathbf{M}(\boldsymbol{\mu}) = \begin{pmatrix} \nabla_{\mathrm{X}}\mathbf{I}(\mathbf{X}_1, \boldsymbol{\mu}_0)^t \Gamma(\mathbf{X}_1) \\ \nabla_{\mathrm{X}}\mathbf{I}(\mathbf{X}_2, \boldsymbol{\mu}_0)^t \Gamma(\mathbf{X}_2) \\ \cdots\cdot \\ \nabla_{\mathrm{X}}\mathbf{I}(\mathbf{X}_N, \boldsymbol{\mu}_0)^t \Gamma(\mathbf{X}_N) \end{pmatrix} \sum(\boldsymbol{\mu}) = \mathbf{M}_0 \sum(\boldsymbol{\mu}) \tag{12}$$

According to Eq. (12), $\mathbf{M}_0$ is the constant part, which denotes the prior information of the tracking target and describes every pixels' change in gray value due to movement. This part can be computed in the initialization phase, while $\sum(\boldsymbol{\mu})$ definitely depends on motion vector $\boldsymbol{\mu}$ and will be recomputed in every frame. Thus, we bring this adjusted $\mathbf{M}(\boldsymbol{\mu},t)$ into (9)

$$\delta\boldsymbol{\mu} = -\sum{}^{-1}\left(\mathbf{M}_0^t\mathbf{M}_0\right)^{-1}\mathbf{M}_0^t[\mathbf{I}(f(\mathbf{X}, \boldsymbol{\mu}), t_n) - \mathbf{I}(\mathbf{X}, t_0)] \tag{13}$$

By the above analysis, the tracking algorithm can be divided into two parts: one can be completed in the initialization step and the other can be executed in real time during tracking. The flow of the tracking algorithm is shown in Fig. 1. The motion model of tracking will be described in section 3.3.

3.2 Illumination insensitive tracking

As the incremental estimation step is effectively computing a structured optical flow, and optical flow methods are well-known to be sensitive to illumination changes. In real environments brightness or contrast changes are unavoidable phenomena that cannot always be controlled. It follows that modeling light changes is necessary for visual trackers

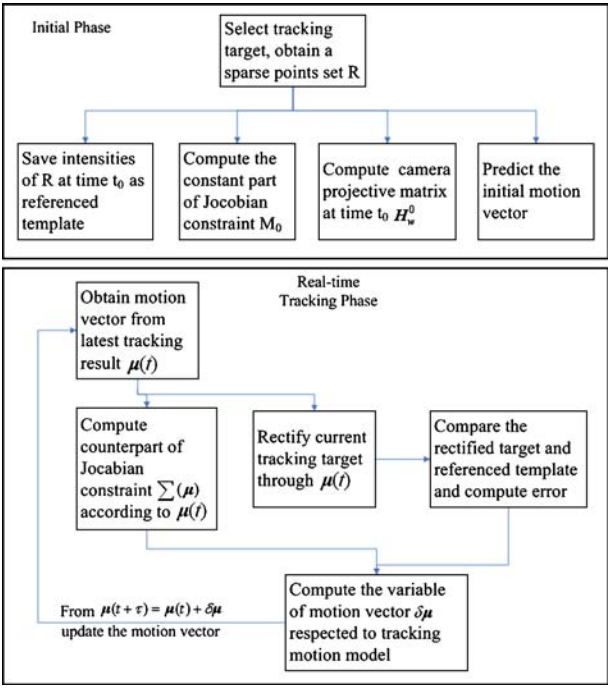

Initial Phase
Select tracking target, obtain a sparse points set R
Save intensities of R at time t₀ as referenced template
Compute the constant part of Jocobian constraint M₀
Compute camera projective matrix at time t₀ H_w^0
Predict the initial motion vector
Real-time Tracking Phase
Obtain motion vector from latest tracking result $\mu(t)$
Compute counterpart of Jocabian constraint $\sum(\mu)$ according to $\mu(t)$
Rectify current tracking target through $\mu(t)$
Compare the rectified target and referenced template and compute error
Compute the variable of motion vector $\delta\mu$ respected to tracking motion model
From $\mu(t+\tau)=\mu(t)+\delta\mu$ update the motion vector

Fig. 1 The work flow of template tracking algorithm

to operate in a general situation. As described in [10, 15], we can estimate the photometric parameters in each frame to ensure the tolerance for illumination changes.

Consider a light source L in the 3-D space and suppose we are observing a smooth surface S, as shown in Fig. 2. The intensity value of point P on the image plane depends on the portion of incoming light from the source L that is reflected by the surface S, and is described by the Bidirectional Reflectance Distribution Function (BRDF). Assuming that

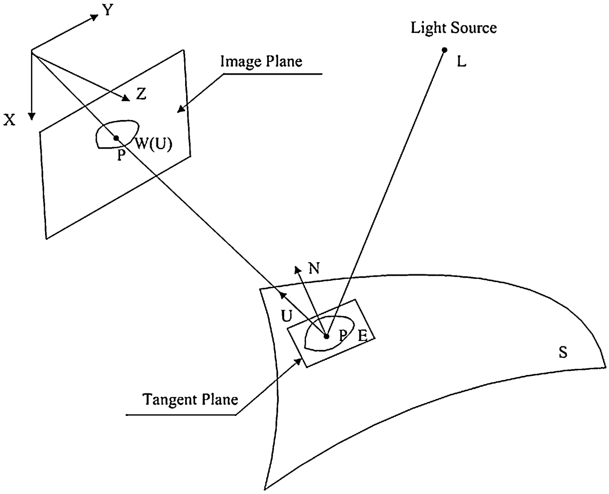


Fig. 2 Image formation process when illumination is taken into account: photometric parameters via Lambertian assumption

the surface S is Lambertian and is a plane in range of U, the BRDF simplifies considerably and the intensity observed at the point $P(\mathbf{X})$ can be modeled as:

$$\mathbf{I}(\mathbf{x}) = \lambda E(\mathbf{X}) + \delta, \quad \forall \mathbf{x} \in W(U) \tag{14}$$

where $E(\mathbf{X})$ is an albedo function of surface S and W(U) is the image region related with the surface in range U, and λ and δ can be thought as parameters that represent respectively the contrast and brightness changes of image. Therefore, we can take the illumination changes into our model and rewrite Eq. (13) as:

$$\delta\boldsymbol{\mu} = -\sum\nolimits^{-1}\left(\mathbf{M}_0^t\mathbf{M}_0\right)^{-1}\mathbf{M}_0^t[\lambda(t_n)\mathbf{I}(f(\mathbf{X},\boldsymbol{\mu}),t_n) - \mathbf{I}(\mathbf{X},t_0) + \delta(t_n)] \tag{15}$$

3.3 Projective motion model of region tracking

The purpose of the proposed algorithm is to obtain the registration information of the environment and thus camera motion recovery is necessary. The motion model is defined as a projective transformation during tracking [6].

Let $X=(\mu,v)^t$ be the Cartesian coordinate and $X_h=(r,s,t)^t$ be the corresponding projective coordinate as shown in Fig. 3.

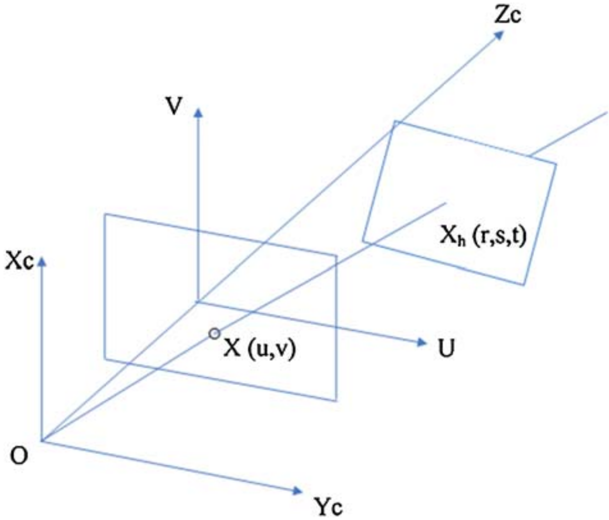
The relationship between them is:

$$\mathrm{X}_h = \begin{pmatrix} r \\ s \\ t \end{pmatrix} \rightarrow \mathrm{X} = \begin{pmatrix} r/t \\ s/t \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} \forall \mathbf{X} \in \mathbf{I}(\mathrm{X}) \tag{16}$$

Because we assume the tracking target is coplanar, the motion model for tracking can be defined as a projective transformation [6], as shown in Eq. (17)

$$f_{\mathrm{X}}(\mathbf{X},\boldsymbol{\mu}) = \mathbf{P}\mathrm{X} = \begin{pmatrix} a & d & g \\ b & e & h \\ c & f & 1 \end{pmatrix}\begin{pmatrix} r \\ s \\ t \end{pmatrix} \tag{17}$$

Fig. 3 Projective model of target in camera coordinate



In this case, the motion parameter vector $\boldsymbol{\mu}$ can be defined as $\mu = (a, b, c, d, e, f, g)^t$. After substituting it into the Jacobian matrix, we have:

$$\nabla_{\mathrm{X}}\mathbf{I}(\mathbf{X},\boldsymbol{\mu})^t = \left(\frac{\partial \mathbf{I}}{\partial \mu}, \frac{\partial \mathbf{I}}{\partial v}, -\left(u\frac{\partial \mathbf{I}}{\partial u} + v\frac{\partial I}{\partial v}\right)\right) \tag{18}$$

$$f_{\mathrm{X}}(\mathbf{X},\boldsymbol{\mu})^{-1} = \mathbf{P}^{-1} \tag{19}$$

$$f_{\boldsymbol{\mu}}(\mathbf{X},\boldsymbol{\mu}) = \begin{pmatrix} r\,0\,0\,s\,0\,0\,t\,0 \\ 0\,r\,0\,0\,s\,0\,0\,t \\ 0\,0\,r\,0\,0\,s\,0\,0 \end{pmatrix} \tag{20}$$

After combining Eq. (18) with Eq. (19), the result is:

$$f_{\mathrm{X}}(\mathbf{X},\boldsymbol{\mu})^{-1} f_{\boldsymbol{\mu}}(\mathbf{X},\boldsymbol{\mu}) = \left(r\mathbf{P}^{-1}\middle|s\mathbf{P}^{-1}\middle|t\mathbf{P}_{12}^{-1}\right) = \Gamma(\mathbf{X})\sum(\boldsymbol{\mu}) \tag{21}$$

where $\mathbf{P}_{12}^{-1}$ denotes the first two columns of $\mathbf{P}^{-1}$ and $\Gamma(\mathbf{X}_i)$ is a constant. Thus, we have:

$$\Gamma(\mathbf{X}) = \left(r\mathbf{I}_{3\times3}\middle|s\mathbf{I}_{3\times3}\middle|t\mathbf{I}_{3\times3}\right) \tag{22}$$

$$\sum(\boldsymbol{\mu}) = \begin{pmatrix} \mathbf{P}^{-1} & 0 & 0 \\ 0 & \mathbf{P}^{-1} & 0 \\ 0 & 0 & \mathbf{P}_{12}^{-1} \end{pmatrix} \tag{23}$$

$\sum(\boldsymbol{\mu})$ is an invertible 9×8 matrix, and by combining with Eq. (15), the target tracking model based on projective transformation can be written as:

$$\delta\boldsymbol{\mu} = -\left(\sum{}^t\mathbf{M}_0^t\mathbf{M}_0\sum\right)^{-1}\sum{}^t\mathbf{M}_0^t[\lambda(t_n)\mathbf{I}(f(\mathbf{X},\boldsymbol{\mu}),t_n) - \mathbf{I}(\mathbf{X},t_0) + \delta(t_n)] \tag{24}$$

3.4 3D environment registration

The correct tracking for target motion results in accurate positions for each sparse point, which together compose the target, and the correspondences between every frame can be computed. Thus we have:

$$(\mathbf{X}_{1..N}, t) = \mathbf{H}_0^n(\mathbf{X}_{1..N}, t_0) \tag{25}$$

where $\mathbf{H}_0^n$ denotes the homography of sparse points set respectively at time t_n and t_0. Therefore, the target's position and orientation in camera coordinates can be estimated from homography. The relationship between camera model and tracking target is shown in Fig. 4.To simplify the projection equation, the world coordinate is defined as the tracking target's coordinate.

Let (x_{pp}) be the target's true coordinates in the world frame, (x_0,y_0) be its coordinates at time t_0 in the camera's projective plane, and (x_n,y_n) be its coordinates at time t_n in the camera's projective plane, as shown in Fig. 3. The relationships among them can be written as:

$$\begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix} = \mathbf{P}_w^0 \begin{pmatrix} x_P \\ y_P \\ 1 \end{pmatrix} \tag{26}$$

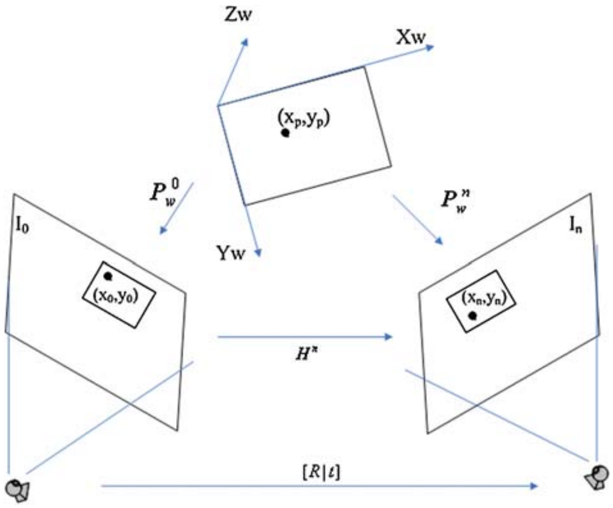


Fig. 4 The relationship between target and camera in tracking process

$$\begin{pmatrix} x_n \\ y_n \\ 1 \end{pmatrix} = \mathbf{P}_w^n \begin{pmatrix} x_P \\ y_P \\ 1 \end{pmatrix} \tag{27}$$

$$\begin{pmatrix} x_n \\ y_n \\ 1 \end{pmatrix} = \mathbf{H}^n \begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix} \tag{28}$$

The projective matrix $\mathbf{P}_w^0$ at time t_0 can be computed in the initialization phase. In the literature, the projection matrix can be computed in the case of four known correspondence features [8], and more points can be used for higher accuracy. The vision projection equation [6] is as follows,

$$\begin{pmatrix} x_n \\ y_n \\ 1 \end{pmatrix} = \lambda K[R|t] \begin{pmatrix} x_P \\ y_P \\ z_P \\ 1 \end{pmatrix} \tag{29}$$

where λ is the scale factor and K is the intrinsic parameters of the camera, $[R|t]$ denotes the rotation and translation related to the real scene, which compose the extrinsic parameters of the camera. $[R|t]$ is also the pose of the environment in camera coordinates, because the world coordinate has been defined with respect to the tracking target. Let the target plane be the Z plane in the world coordinate and z_p=0, then by substituting Eqs. (26), (27), (28) into Eq. (29) we have:

$$\mathbf{H}_0^n = P_w^n \left(P_w^0\right)^{-1} = \lambda K[R|T]\left(P_w^0\right)^{-1} \tag{30}$$

From Eq. (25), $\mathbf{H}_0^n$ can be computed using the result of the tracking process. Thus, the camera pose $[R|t]$ can be completely registered by computing Eq. (30).

4 Architecture of marker-less AR system

In the literature, an AR system is composed of the following key components: real scene capture module, registration, rendering, merging, and stereo vision. Based on our registration algorithm presented in section 3, a marker-less AR system can be designed and its structure is shown in Fig. 5.

A high-capture-rate calibrated camera is adopted for real scene capture and it is fixed with the user's head. The real scene video is transferred to a portable computer, where the registration algorithm can be executed. Then, rendering and merging based on registration can be achieved and, finally, the augmented image sequences are transferred to the head mounted display (HMD).

The whole working flow of the proposed marker-less AR system is:

1. Initialization
   - (a) Establish the computer generated 3D models.
   - (b) Calibrate the camera and obtain the intrinsic parameters *K*.
   - (c) Manually select a region as the tracking target and store it as the reference template.
   - (d) Compute initial parameters.
2. Registration, rendering and merging
   - (a) Track target with camera and estimate its pose and position.
   - (b) Render 3D model to virtual image with respect to the target's pose.
   - (c) Merge the virtual image with the real-time image sequences.
3. Transfer the augmented scene to the HMD.

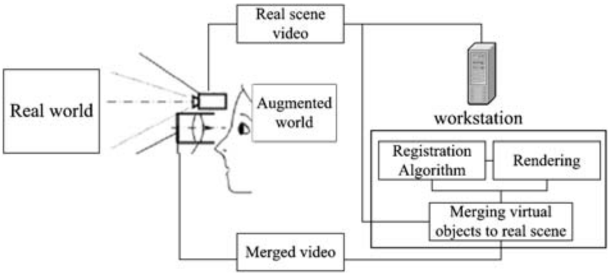
5 Experiment

The tracking-based registration algorithm has been implemented to run on a common workstation (Pentium 4 3.0 GHz CPU) in real time. The experiments are discussed with two phases: (1) Planar region tracking in the scene. We evaluate the tracking accuracy with pixel residues of tracked region and template; (2) 3D registration and augmentation reality via fixed region tracking.

5.1 Tracking evaluation

In phase one, a complex-texture plane moves and an interesting region is randomly defined as the target region. We test 400 frames sequence and the tracked region comes back to

Fig. 5 Architecture of AR system



original position in the end. In the tracking sequence, we use a "intrude" object (a pen) and adjust environment illumination conditions to test the algorithm robustness. The errors between the rectified target and template are recorded. During the experiment, 120 sparse points are selected to represent the target region. The results are shown as follows: the template shown in Fig. 6(a) is defined during the initialization phase and stored, while Fig. 6 (b) (c) (d) (e) illustrate the registration process. The illumination condition changing is shown in Fig. 6(c).

To quantitatively illustrate the algorithm accuracy, the tracking region is rectified with motion parameters and compared with the template to obtain tracking residua. Following [10], we define the residue as

$$\hat{\mathrm{Re}} = \frac{1}{\mathrm{Z}}\sum_{\mathrm{X}} \|\mathbf{I}_{\mathrm{X}}(\boldsymbol{\mu}(\mathrm{t}) + \delta\boldsymbol{\mu}, \mathrm{t}+\tau) - \mathbf{I}_{\mathrm{X}}(0, \mathrm{t}_0)\|^2 \tag{31}$$

where Z is normalize term and X denotes the tracked point. We test three indoor video sequences (400 frames for each), which are similar with scenes in Fig. 6. The plot shown in Fig. 7 (red curve) illustrates the error between the registered patch and our target can be

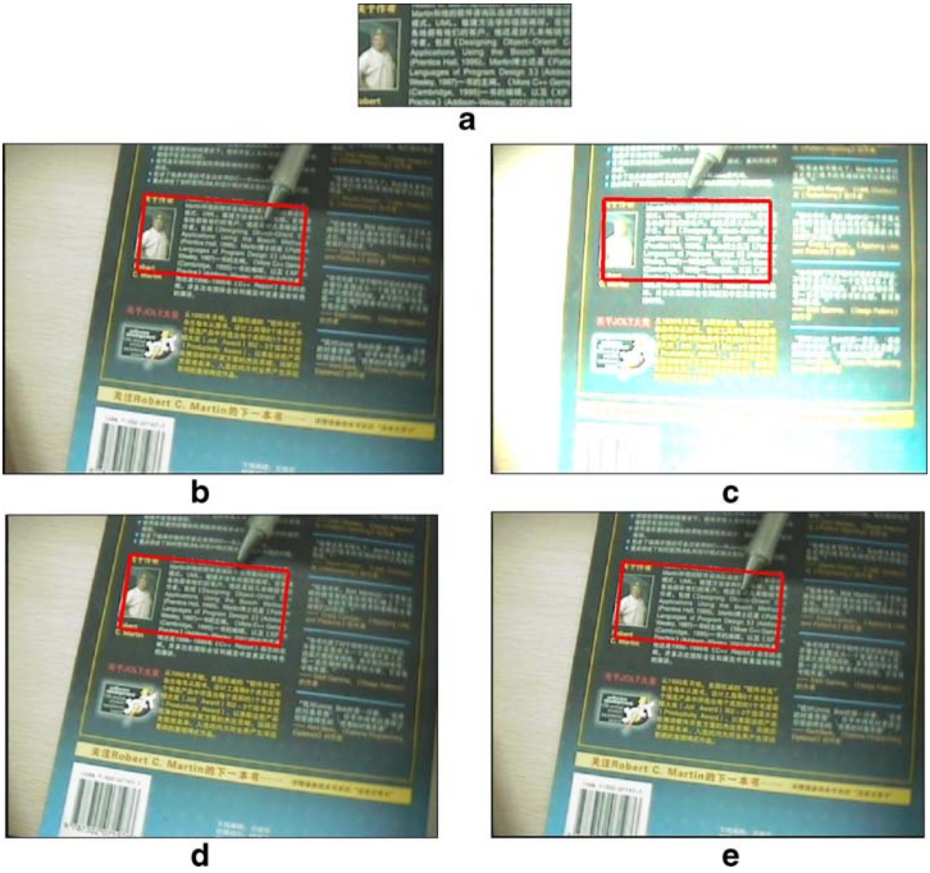


Fig. 6 Tracking a fixed region in real-time. (**a**) Template of Target. (**b**) Camera moving at 70th frame. (**c**) Camera moving at 120th frame with illumination changing. (**d**) Camera moving at 180th frame. (**e**) Camera moving at 380th frame

control well and the region can be matched exactly. We also test Shi–Tomasi tracker [10], as shown in Fig. 7 (blue curve), and it can not track the region after approximately 90 frames.

5.2 3D Registration evaluation

In phase two, the environment registration is achieved based on the tracking step, which helps us to merge the virtual object with the real scene.

Two representative experimental images are shown in Fig. 8 and the image resolution is 320×240. Figure 8(a) and (b) are the augmented scene observed with the HMD. Intuitively, it shows that the augmented effect is natural and seamless due to the high registration accuracy. Our registration method can also be used for outdoor AR solution. The experimental results in our other project about mobile augmented reality (MAR) system for historic-site navigation are shown in Fig. 9. The detailed of MAR technology will be discussed in future works.

In order to further demonstrate the system advance on the registration accuracy and efficiency, we compare with two stat of the art methods of AR registration [16, 18]. For evaluate the accuracy of 3D registration, we utilize the VICON system (http://www.vicon.com/support), one well-known pose and motion capture platform. We assume the output of VICON system is the ground truth benchmark, and registration accuracy is thus normalized via the average errors of 6 freedom degrees, following the definition in [11, 17].

We first test the registration performance on randomly illumination changing, and compare with the color-markers based method. Note that it is difficult to measure the illumination condition precisely, and thus we manually adjust the lighting condition, including the spot light and global light. As shown in Fig. 10 (a), the red curve and green curve denote the 3D registration accuracy

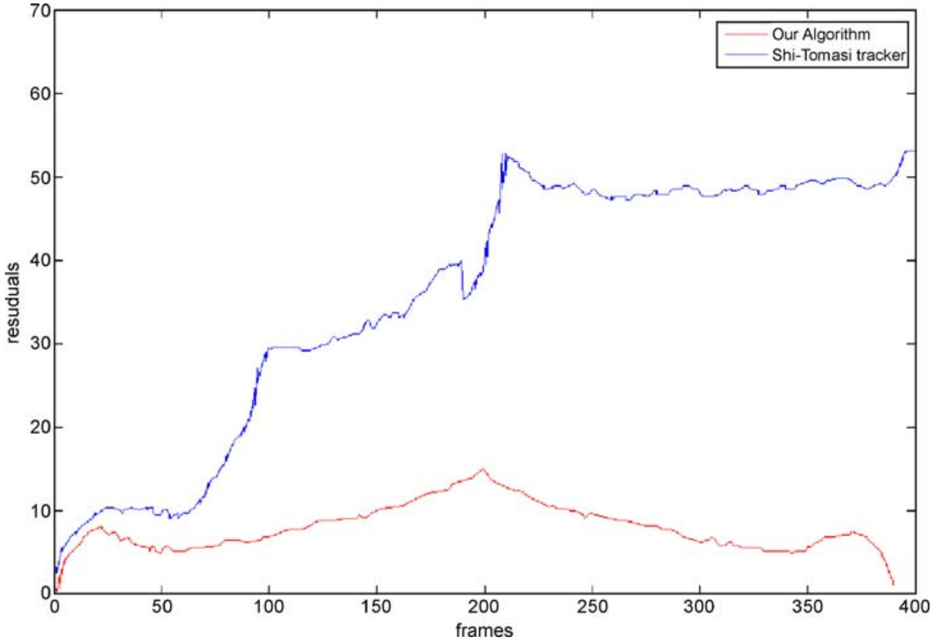


Fig. 7 Curve of residue errors between tracked regions and template. *Red curve*: our algorithm can control the illumination changing and "intrude" object well and the region can be matched exactly. *Blue curve*: the Tomasi Tracker (*blue curve*) failed to track the region, at around 90 frame

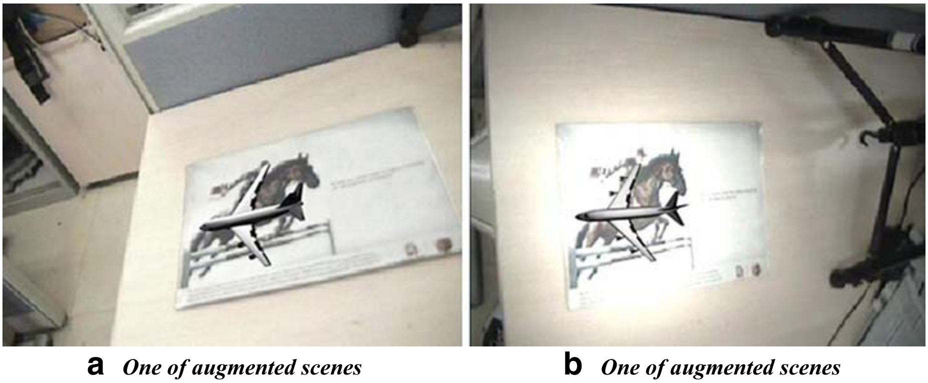


Fig. 8 (**a**) One of augmented scenes; (**b**) one of augmented scenes

of our method and color markers based method [16] respectively. Along with the horizontal axe, the illumination changing became drastic gradually. The dashed marked nodes in the curves denote the critical value of visual perception, that is, the virtual objects cannot be matched well with the real objects (scene) below the critical registration accuracy. The following experiment shows the registration performance when the random occlusion increases. As shown in Fig. 10 (b), the red curve denotes our result and the blue denotes the result of FFT-based template matching method [18]. The two curve illustrate the accuracy reduces with the random occlusion increases, and the dashed marked nodes in the curves are the critical point of human visual perception as well.

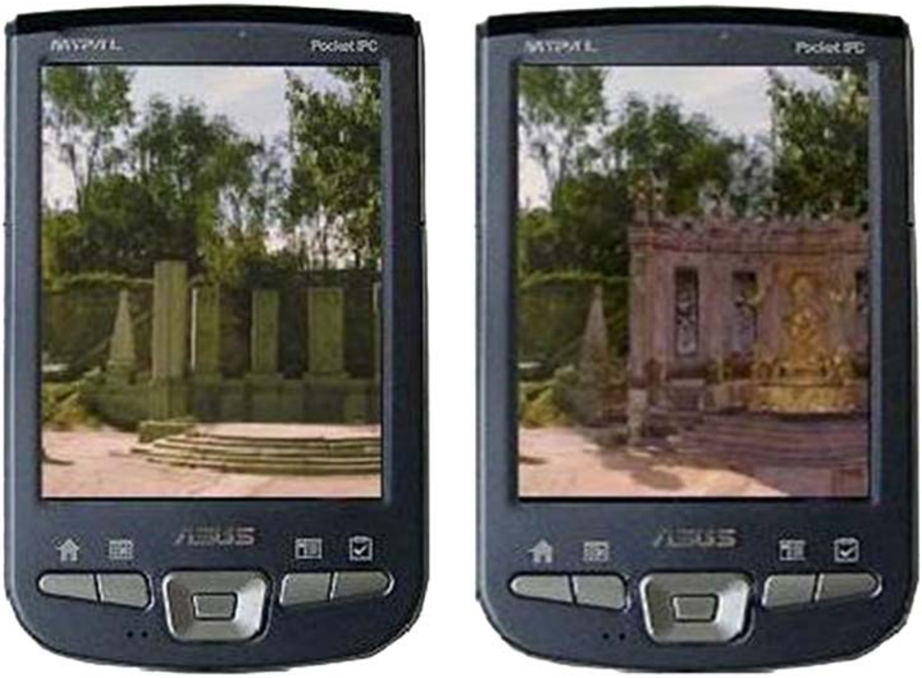


Fig. 9 Mobile augmented reality based on the proposed registration method

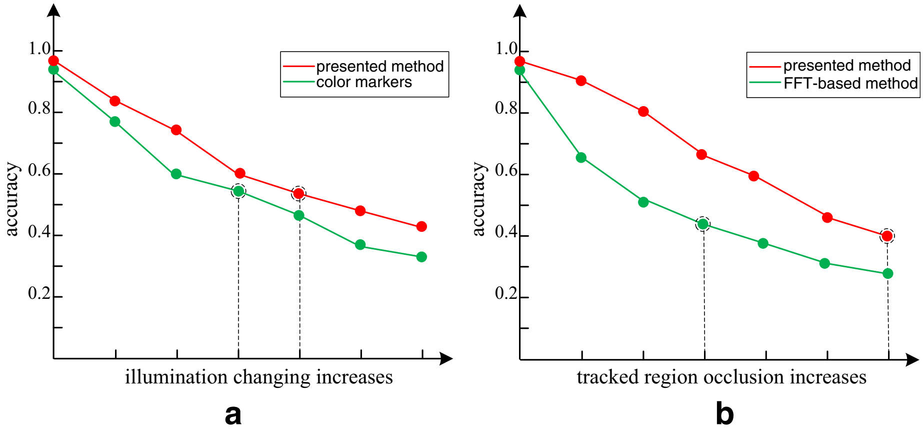


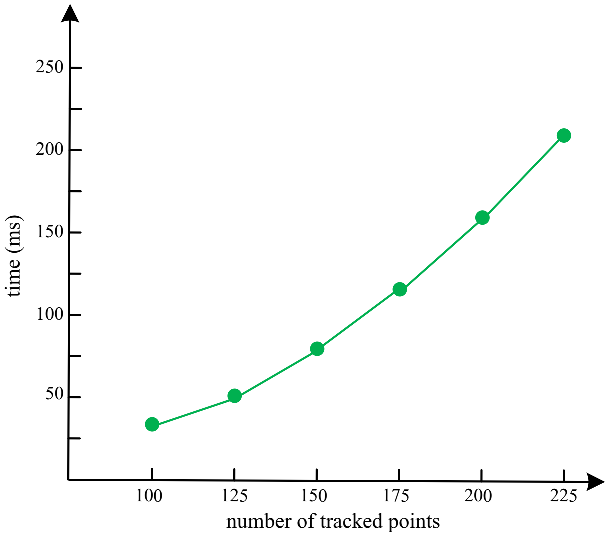
Fig. 10 Curves of 3D registration accuracy with comparison. (**a**) Shows the performance under illumination randomly changing; (**b**) Shows the performance under random occlusion occurring

In practice, to enhance the robustness of 3D registration, multiple fixed regions can be tracked simultaneously. Since we randomly sample points from fixed regions as tracking correspondences, the multiple regions tracking is solved equally as single. Hence the system efficiency should be evaluated via sampled points for tracking. The consuming time curve is illustrated in Fig. 11 it shows the cost time increase with point number in each iterative step. In the proposed AR system, we fix around 150 sampled points from three tracked regions empirically.

6 Summary

In this paper, a novel 3D registration approach based on planar region tracking has been proposed. This algorithm combines fixed region tracking and perspective motion estimation. A texture-full region is first selected manually and stored as the reference

Fig. 11 Consuming time curve for each computing step



template for initialization. When the camera undergoes freely motion, the region can be tracked in real time, and its position and pose can be estimated, combining the camera intrinsic parameters. Based on this registration method, an AR system is proposed.

However, one major limitation with this algorithm is that the template needs to be defined manually. That can be improved by adopting some planar region localization algorithms. In addition, we can not only define template with the intensity of sampled point set, but also use those discriminative features, such as object shape, patch gradient histogram.

Acknowledgements This project is supported by National Basic Research Program of China (National 863 Program, Grant No. 2006AA01Z339), National Natural Science Foundation of China (Grant No. 60673198), and China Postdoctoral Science Foundation funded project (Grant No. 20080430313). The author would like to thank Ke Yang for contributive comments and assistance in experiments.

References

1. Azuma R, Hoff B, Neely H (1999) A motion-stabilized outdoor augmented reality system. Proc IEEE Virtual Reality, Los Alamitos. IEEE, California, pp 252–259
2. Azuma R, Baillot Y et al (2001) Recent advances in augmented reality, computer graphics and applications. IEEE Comput Graph Appl 21(6):34–47 doi:10.1109/38.963459
3. Bajura M, Henry F, Ohbuchi R (1992) Merging virtual reality with the real world: seeing ultrasound imagery within the patient. Proceedings of SIGGRAPH'92 (Chicago, IL). Comput Graph (ACM) 26(2):203–210 doi:10.1145/142920.134061
4. Behringer R (1999) Registration for outdoor augmented reality applications using computer vision techniques and hybrid sensors. Proc IEEE Virtual Real 13–17:244–251 (March)
5. Black MJ, Jepson AD (1998) Eigentracking: robust matching and tracking of articulated objects using a view-based representation. Int J Comput Vis 26(1):63–84 doi:10.1023/A:1007939232436
6. Bobick AF, Wilson AD (1995) A state-based technique for the summarization of recognition of gesture. Proc. Proceedings of International Conference on Computer Vision, 382–388
7. Chen J, Shi Q, Wang Y (2001) AR technology and applications. Comput Engineer Application 37(21):55–57
8. Darrell T, Moghaddam B, Pentland A (1996) Active face tracking and pose estimation in an interactive room. Proceedings of IEEE Conf. Computer Vision and Pattern Recognition 67–72
9. Dorfmuller K (1999) Robust tracking for augmented reality using retroreflective markers. Comput Graph 23(6):795–800 doi:10.1016/S0097-8493(99)00105-3
10. Hager GD, Bellumeur PN (1996) Real-time tracking of image regions with changes in geometry and illumination. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 403–410
11. Hu X, Liu Y, Wang Y (2005) Autocalibration of an electronic compass for augmented reality. Proc of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR) 1:182–183
12. Hutchinson S, Hager GD, Corke P (1996) A tutorial introduction to visual servo control. IEEE Trans Robot Autom 12(5):651–670 doi:10.1109/70.538972
13. Jafari S, Jarvis R (2005) Robotic eye-to-hand coordination: implementing visual perception to object manipulation. Int J Hybrid Intell Syst 2(4):269–293
14. Kutulakos KN, Vallino JR (1998) Calibration-free augmented reality. IEEE Trans Vis Comput Graph 4(1):1–20 doi:10.1109/2945.675647
15. La Cascia M, Sclaroff S, Athitsos V (2000) Fast, reliable head tracking under varying illumination: an approach based on registration of textured-mapped 3D models. IEEE Trans Pattern Anal Mach Intell 22(4):322–336 doi:10.1109/34.845375
16. Li X, Liu Y, Wang Y et al (2005) An improved colored-marker based registration method for AR applications. Lect Notes Comput Sci 3482:266–273
17. Li Y, Wang Y, Liu Y (2007) Fiducial marker based on projective invariant for augmented reality. J Comput Sci Technol 22(6):890–897 doi:10.1007/s11390-007-9100-0
18. Lin L, Liu Y, Zheng W, Wang Y (2006) Registration algorithm based on image matching for outdoor AR system with fixed viewing position. IEE Proc, Vis Image Signal Process 153(1):57–62 doi:10.1049/ip-vis:20045181
19. Okuma T, Sakaue K, Takemura H, Yokoya N (2000) Real-time camera parameter estimation from images for a mixed reality system. Proc Int Conf Pattern Recognit 4:4482–4486

20. Ribo M, Pinz A, Fuhrmann A (2001) A new optical tracking system for virtual and augmented reality applications. Instrum Meas Tech Conf 3:1932–1936
21. Shi J, Tomasi C (1994) Good features to track. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94), 593–600
22. Tang S-L, Kwoh C-K et al (1998) Augmented reality systems for medical applications. Eng Med Biol Mag 17(3):49–58 doi:10.1109/51.677169
23. Uenohara M, Kanade T (1995) Vision-based object registration for real-time image overlay. Int J Comput Biol Med 25(2):249–260 doi:10.1016/0010-4825(94)00045-R
24. You S, Neumann U (2001) Fusion of vision and gyro tracking for robust augmented reality registration. Proc IEEE Virtual Real 2001:71–78 doi:10.1109/VR.2001.913772
25. Zagoranski S, Divjak S (2003) Use of augmented reality in education, EUROCON 2003, Computer as a Tool, The IEEE Region 8, 22–24 Sept. 2003, Vol. 2: 339–342

Liang Lin was born in 1981. He graduated from Beijing Institute of Technology (BIT) in 2003 with the major, “electronic science and technology” and was awarded as excellent undergraduate. He received the Ph.D degree in BIT in 2008. He worked in Lab of Optoelectronic tech and information system for 3 years and studied in center for image and vision science (CIVS) of UCLA, USA, as a visiting scholar. His research interests are computer vision, image understanding, and augmented reality. He has published a number of papers in relevant international/national journals and conferences.

Yongtian Wang received his B.Sc. degree in precision instrumentation from Tianjin University, China, in 1982, and his Ph.D. degree in optics from the University of Reading, England, in 1986. He is currently a professor of optics and the director of the Center for Research on Optoelectronic Technology and Information System in Beijing Institute of Technology. His research interests include optical design and CAD, optical instrumentation, image processing, virtual reality (VR) and augmented reality (AR) technologies and applications. Dr. Wang is a Fellow of SPIE and a director of the Chinese Optics Society.

Yue Liu received his M.Sc. degree in Telecommunication and Electronic System from Jilin University of Technology, China, in 1996, and his Ph.D. degree in Telecommunication and Information System from Jilin University, China, in 2000. He is currently a professor of optics in Beijing Institute of Technology. His research interests include computer vision, image processing, virtual reality (VR) and augmented reality (AR) technologies and applications. Dr. Liu is a director of the China Society of Image and Graphics.

Xiong Caiming received the BS and MS degree from the Huazhong University of Science and Technology in 2005 and 2007. His current research interests are multilevel graphical model, optimization algorithm, machine learning and non-photorealistic rendering.

Kun Zeng was born in Hunan, China in 1978. He received the Ph.D degree from National Laboratory of Pattern Recognition Institute of Automation, Chinese Academy of Sciences in 2008. His research interests are in computer vision, image understanding and non-photorealistic Rendering.