

AAM and Non-rigid Registration in Augmented Reality

Yuan Tian, Tao Guan and Cheng Wang
*Digital Engineering and Simulation Research Center,
Huazhong University of Science and Technology,
Wuhan, China*

1. Introduction

Augmented Reality copes with the problem of dynamically augmenting or enhancing the real world with computer generated virtual objects [Azuma, 1997; Azuma, 2001]. Registration is one of the most pivotal problems in augmented reality applications. Typical augmented reality applications track 2D patterns on rigid planar objects in order to acquire the pose of the camera in the scene. Although the problem of rigid registration has been widely studied [Yuan et al., 2005; Yuan et al., 2006; Guan et al., 2008a; Guan et al., 2008b; Li et al., 2008], non-rigid registration is recently receiving more and more attention. There are many non-rigid objects existing in the real world such as animated faces, deformable cloth, hand and so forth. How to overlaid virtual objects on the non-rigid objects is particular challenging.

Recently, many related non-rigid registration approaches have been reported. In many cases (e.g. human faces), only a few feature points can be reliably tracked. In [Bartoli et al., 2004], a non-rigid registration method using point and curve correspondences was proposed to solve this problem. They introduced curves into the non-rigid factorization framework because there are several curves (e.g. the hairline, the eyebrows) can be used to determine the mapping. The mapping function is computed from both point and curve correspondences. This method can successfully augment the non-rigid object with a virtual object. In [Pilet et al., 2005; Pilet et al., 2007], they presented a real-time method for detecting deformable surfaces with no need whatsoever for a prior pose knowledge. The deformable 2D meshes are introduced. With the use of fast wide baseline matching algorithm, they can superimpose an appropriately deformed logo on the T-shirt. These methods are robust to large deformations, lighting changes, motion blur and occlusions. To align the virtual objects generated by computers with the real world seamlessly, the accurate registration data should be provided. In general, registration can be achieved by solving a point matching problem. The problem is to find the correspondence between two sets of tracked feature points. Therefore, the detection of feature points and the points tracking are the two main problems. Rigid object detection and tracking have been extensively studied and effective, robust, and real-time solutions proposed [Lowe, 2004; Lepetit & Fua, 2005; Lepetit et al., 2005; Rosten & Drummond, 2005]. Non-rigid object detection and tracking is far more complex because the object is deformable and not only the registration data but also a large number of deformation parameters must be estimated.

Source: Augmented Reality, Book edited by: Soha Maad,
ISBN 978-953-7619-69-5, pp. 230, January 2010, INTECH, Croatia, downloaded from SCIYO.COM

Active Appearance Models (AAM), introduced a few years ago [Cootes et al., 1998; Cootes et al., 2001], are commonly used to track non-rigid objects, such as faces and hands. There are many methods have been proposed to track non-rigid objects using AAM. A working system for finding and tracking a human face and its features using active appearance models was presented in [Ahlberg, 2001]. A wireframe model is adapted to the face in each image. Then the model is matched to the face in the input image using active appearance algorithm. In [Sung & Kim, 2004], the previous 2D AAM is extended to 3D shape model and modified model fitting algorithm was proposed. In [Markin & Prakash, 2006], occluded images are included into AAM training data to solve the occlusion and self-occlusion problem. This approach can improve the fitting quality of the algorithm.

With known coordinates of 3D points in the world coordinates and the corresponding 2D image points, the camera pose can be estimated. For non-rigid objects, AAM algorithm is a robust method to acquire the 2D image points. It has been proven to be a useful method for matching any of the statistical models to a new image rapidly. The 3D points of the non-rigid objects can be represented by a linear combination of a set of 3D basis shapes. By varying the configuration weights and camera pose parameters, the error between the estimated 2D points (projected by the estimated 3D shapes using estimated camera pose parameters) and 2D tracking points can be minimized.

Many methods have been proposed to recover 3D shape basis from 2D image sequences. In [Tomasi & Kanade, 1992], the factorization method is used to recover shape and motion from a sequence of images under orthographic projection. The image sequence is represented as a measurement matrix. It is proved that under orthography, the measurement matrix is of rank 3 and can be factored into 3D pose and 3D shape matrix. Unfortunately this technique can not be applied to non-rigid deforming objects, since they are based on the rigidity assumption. The technique based on a non-rigid model is proposed to recover 3D non-rigid shape models under scaled orthographic projection [Bregler et al., 2000]. The 3D shape in each frame can be expressed by a linear combination of a set of K basis shapes. Under this model, the 2D tracking matrix is of rank $3K$ and can be factored into 3D pose, object configuration and 3D basis shapes with the use of SVD.

In this chapter, a novel non-rigid registration method for augmented reality applications with the use of AAM and factorization method is introduced. We focus on AAM algorithm and factorization method which can obtain the 3D shape basis, object configuration and 3D pose simultaneously. The following demonstrations are mainly based on the researches presented in [Tomasi & Kanade, 1992; Cootes et al., 1998; Bregler et al., 2000; Cootes et al., 2001; Xiao et al., 2004; Zhu et al., 2006; Tian et al., 2008]. In section 2, we will have a detailed review of active appearance models and focus on the way of how to track non-rigid objects. In section 3, we will illustrate how to compute the 3D shape basis, the camera rotation matrix and configuration weights of each training frame simultaneously from the 2D tracking data using factorization algorithm. In section 4, we will introduce how to compute the precise configuration weights and camera rotation matrix by optimization method and the experimental results are also given.

2. Tracking non-rigid objects using AAM

Tracking non-rigid objects using AAM includes five main steps. The first step is to obtain landmarks in training image set. Then establish the shape model and texture model separately. These two models are unified into one appearance model in the next step. Finally

the difference between the closest synthesized AAM model and input image is minimized to get the accurate tracking result. The flowchart is shown in Fig. 1.

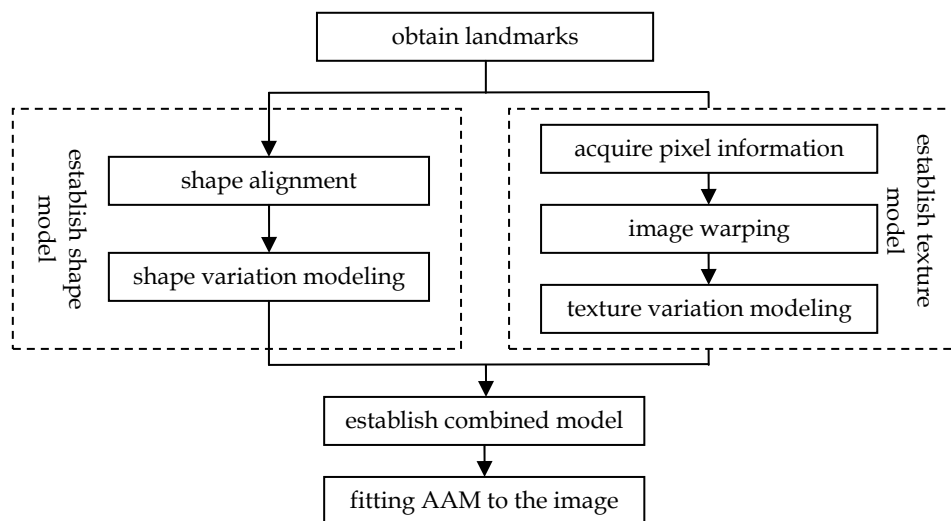


Fig. 1. The flowchart of tracking non-rigid objects using AAM.

2.1 Obtain landmarks



Fig. 2. Examples of training images manually labeled with consistent landmarks.

Before establishing the models, we should place hundreds of points in each 2D training image. The landmarks in each image should be consistent. We usually choose the points of high curve or junctions as landmark points which will control the shape of the target object strictly. The acquisition process is cumbersome especially when it is done manually. Researchers have looked for different ways to reduce the burden. Ideally, one only need to place points in one image and the corresponding points in the remain training images can be found automatically. Obviously, this is impossible. However many semi-automatic methods

have been proposed and can successfully find the correspondences across an image set. Here we focus on the tracking procedure, more details about semi-automatic placement of landmarks can be found in [Sclaroff & Pentland, 1995; Duta et al., 1999; Walker et al., 2000; Andresen & Nielsen, 2001]. The examples of training images manually labeled with consistent landmarks are shown in Fig. 2. We use 7 training images taken from different viewpoints. Each image is labeled with 36 landmarks shown as the “x” in the figures. Given a training image set with key landmark points are marked on each example object, we can establish the shape and texture variations. These approaches will be detailedly illustrated in the following sections.

2.2 Establish shape model

Suppose L denotes the number of shapes in a training image set and m is the number of key landmark points on each image. The vector representation for each shape would then be:

$$X_i = [x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_m]^T, \quad i = 1, 2, \dots, L \tag{1}$$

Then the training image set $\Omega = (X_1, X_2, \dots, X_L)$.

The shape and pose (include position, scale and rotation) of the object in each training image is different, so the shapes should be aligned to filter out the pose effects. We will first explain how to align two shapes and then extend it to a set of shapes.

If X and X' are two shape vectors, the goal is to minimize the square sum of corresponding landmarks after alignment using similarity transformation technique. That is to minimize E :

$$E = |T(X) - X'|^2 \tag{2}$$

where

$$T \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u & -v \\ v & u \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{3}$$

Choose three kinds of transformation factors: scale factor- s , rotation factor- θ and translation factor- t . And $u = s \cdot \cos \theta$, $v = s \cdot \sin \theta$. Hence $s^2 = u^2 + v^2$, $\theta = \tan^{-1}(v/u)$. Then equation (2) can be rewritten as:

$$E(u, v, t_x, t_y) = |T(X) - X'|^2 = \sum_{i=1}^m (ux_i - vy_i + t_x - x'_i)^2 + (vx_i + uy_i + t_y - y'_i)^2 \tag{4}$$

To solve the minimum value of equation (4) is equivalent to let the partial derivative equals to zero. Then the transformation parameters are:

$$u = \frac{(x \cdot x')}{|x|^2}, \quad v = \frac{\sum_{i=1}^m (x_i y'_i - y_i x'_i)}{|x|^2}, \quad t_x = \frac{1}{m} \sum_{i=1}^m x'_i, \quad t_y = \frac{1}{m} \sum_{i=1}^m y'_i \tag{5}$$

The alignment of a set of shapes can be processed by iterative approach suggested by [Bookstein, 1996]. The detailed process is shown in Table 1.

So the new training image set $\hat{\Omega} = (\hat{X}_1, \hat{X}_2, \dots, \hat{X}_L)$. The mean shape is calculated by:

- Step 1: Choose one shape as the initial estimate of the mean shape.
- Step 2: Scale the mean shape so that $|\bar{X}_0| = 1$.
- Step 3: Align all the remaining shapes to the mean shape using the method described above.
- Step 4: Re-estimate the mean shape from aligned shapes.
- Step 5: Scale the new mean shape so that $|\bar{X}_{new}| = 1$.
- Step 6: If the new calculated mean shape doesn't change significantly, convergence is declared; else return to step 3.

Table 1. The process of aligning a set of shapes.

$$\bar{X} = \frac{1}{L} \sum_{i=1}^L \hat{X}_i \tag{6}$$

The covariance matrix can thus be given as:

$$\sum_s = \frac{1}{L-1} \sum_{i=1}^L (\hat{X}_i - \bar{X})(\hat{X}_i - \bar{X})^T \tag{7}$$

Calculate the eigenvalue $\lambda_{s,i}$ of \sum_s and the corresponding eigenvector $\eta_{s,i}$:

$$\sum \eta_{s,i} = \lambda_{s,i} \eta_{s,i} \tag{8}$$

Sort all the eigenvalues in descending order:

$$\lambda_{s,i} \geq \lambda_{s,i+1}, \quad i = 1, 2, \dots, 2m - 1 \tag{9}$$

The corresponding set of eigenvectors is $H_s = [\eta_{s,1}, \eta_{s,2}, \dots, \eta_{s,2m}]$.

To reduce the computational complexity, the principal component analysis (PCA) method is used to reduce the space dimensions. Choose t largest eigenvalues which satisfy the following condition:

$$\sum_{i=1}^t \lambda_{s,i} \geq 0.98 \left(\sum_{i=1}^{2m} \lambda_{s,i} \right) \tag{10}$$

Then any shape instance can be generated by deforming the mean shape by a linear combination of eigenvectors:

$$\hat{X} \approx \bar{X} + \Phi_s b_s \tag{11}$$

where Φ_s is a matrix of dimension $2m \times t$ and contains t eigenvectors corresponding to the largest eigenvalues, $\Phi_s = (\varphi_{s,1} | \varphi_{s,2} | \dots | \varphi_{s,t})$. b_s is a t dimensional vector and the eigenvectors are mutually orthogonal, so it can be given by:

$$b_s = \Phi_s^{-1} (\hat{X} - \bar{X}) = \Phi_s^T (\hat{X} - \bar{X}) \tag{12}$$

By varying the elements of b_s , new shape instance can be generated using equation (11).

2.3 Establish texture model

To establish a complete appearance model, not only the shape but also the texture should be considered. We should firstly acquire the pixel information over the region covered by the landmarks on each training image. Then the image warping method is used to consistently collect the texture information between the landmarks on different training images. Finally establish the texture model using the PCA method.

The texture of a target object can be represented as:

$$g = [g_1, g_2, \dots, g_n]^T \quad (13)$$

where n denotes the number of pixel samples over the object surface.

In an image, the target object may occupy a small region of the whole image. The pixel intensities and global changes in illumination are different in each training image. The most important information is the texture that can reflect the characteristic of the target object. Due to the number of pixels in different target region is different and it is difficult to acquire the accurate corresponding relationship between different images, the texture model can not be established directly. We need to obtain a texture vector with the same dimension and corresponding relationship. So we warp each example image so that its control points match the reference shape. The process of image warping is described as follows:

Firstly, apply Delaunay triangulation to the reference shape to obtain the reference mesh which is consisted of a set of triangles. We choose the mean shape as the reference shape.

Secondly, suppose v_1, v_2 and v_3 are three vertices of a triangle in the example mesh, any internal point v of the triangle can be written as:

$$v = v_1 + \beta(v_2 - v_1) + \gamma(v_3 - v_1) = \alpha v_1 + \beta v_2 + \gamma v_3 \quad (14)$$

where $\alpha + \beta + \gamma = 1$. Constrain $0 \leq \alpha, \beta, \gamma \leq 1$ because v is inside the triangle. Given $v = [x, y]^T$, $v_1 = [x_1, y_1]^T$, $v_2 = [x_2, y_2]^T$, $v_3 = [x_3, y_3]^T$, then α, β, γ can be calculated by:

$$\alpha = 1 - (\beta + \gamma) \quad (15)$$

$$\beta = \frac{yx_3 - x_1y - x_3y_1 - y_3x + x_1y_3 + xy_1}{-x_2y_3 + x_2y_1 + x_1y_3 + x_3y_2 - x_3y_1 - x_1y_2} \quad (16)$$

$$\gamma = \frac{xy_2 - xy_1 - x_1y_2 - x_2y + x_2y_1 + x_1y}{-x_2y_3 + x_2y_1 + x_1y_3 + x_3y_2 - x_3y_1 - x_1y_2} \quad (17)$$

Finally, the corresponding point v' of the triangle in the reference mesh can be calculated by:

$$v' = v'_1 + \beta(v'_2 - v'_1) + \gamma(v'_3 - v'_1) = \alpha v'_1 + \beta v'_2 + \gamma v'_3 \quad (18)$$

where v'_1, v'_2 and v'_3 are the vertices of the corresponding triangle in the reference mesh.

After image warping process, each shape in the training set is warped to the reference shape and sampled. The influence from the global linear changes in pixel intensities is removed.

To reduce the effects that caused by global lighting variations, the example samples are normalized by applying a scaling a and offset b :

$$g' = (g'_1, g'_2, g'_3, \dots, g'_n) = \left(\frac{g_1 - b}{a}, \frac{g_2 - b}{a}, \frac{g_3 - b}{a}, \dots, \frac{g_n - b}{a} \right) \quad (19)$$

where

$$b = \frac{1}{n} \sum_{i=1}^n g'_i, \quad a = \sigma, \quad \sigma^2 = \frac{1}{n} \sum_{i=1}^n (g'_i - b)^2 \quad (20)$$

The establishment of the texture model is identical to the establishment of the shape model which is also analyzed by PCA approach. The mean texture is calculated by:

$$\bar{g} = \frac{1}{L} \sum_{i=1}^L g'_i \quad (21)$$

The covariance matrix can thus be given as:

$$\sum_g = \frac{1}{L-1} \sum_{i=1}^L (g'_i - \bar{g})(g'_i - \bar{g})^T \quad (22)$$

Calculate the eigenvalue $\lambda_{g,i}$ of \sum_g and the corresponding eigenvector $\eta_{g,i}$:

$$\sum_g \eta_{g,i} = \lambda_{g,i} \eta_{g,i} \quad (23)$$

Sort all the eigenvalues in descending order:

$$\lambda_{g,i} \geq \lambda_{g,i+1}, \quad i = 1, 2, \dots, 2n-1 \quad (24)$$

The corresponding set of eigenvectors is $H_g = [\eta_{g,1}, \eta_{g,2}, \dots, \eta_{g,2n}]$.

To reduce the computational complexity, the PCA method is used to reduce the space dimensions. Choose t largest eigenvalues which satisfy the following condition:

$$\sum_{i=1}^t \lambda_{g,i} \geq 0.98 \left(\sum_{i=1}^{2n} \lambda_{g,i} \right) \quad (25)$$

Then any texture instance can be generated by deforming the mean texture by a linear combination of eigenvectors:

$$g' \approx \bar{g} + \Phi_g b_g \quad (26)$$

where Φ_g is a matrix of dimension $2n \times t$ and contains t eigenvectors corresponding to the largest eigenvalues, $\Phi_g = (\phi_{g,1} | \phi_{g,2} | \dots | \phi_{g,t})$. b_g is a t dimensional vector and the eigenvectors are mutually orthogonal, so it can be given by:

$$b_g = \Phi_g^{-1} (g' - \bar{g}) = \Phi_g^T (g' - \bar{g}) \quad (27)$$

By varying the elements of b_g , new texture instance can be generated using equation (26).

2.4 Establish combined model

Since the shape model and texture model have been established, any input image can be represented using the shape parameter vector b_s and texture parameter vector b_g . Since there are some correlations between shape and texture variations, the new vector b can be generated by combining b_s and b_g :

$$b = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix} = \begin{pmatrix} W_s \Phi_s^T (\hat{X} - \bar{X}) \\ \Phi_g^T (g' - \bar{g}) \end{pmatrix} \quad (28)$$

where W_s is a diagonal matrix which adjust the weighting between pixel distances and pixel intensities. W_s is calculated by:

$$W_s = rI = \begin{bmatrix} r & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & r \end{bmatrix} \quad (29)$$

where r^2 is the ratio of the total intensity variation to the total shape variation:

$$r = \frac{\lambda_g}{\lambda_s}, \lambda_g = \sum \lambda_{g,i}, \lambda_s = \sum \lambda_{s,i} \quad (30)$$

Apply PCA on b , then

$$b = \Phi_c c \quad (31)$$

where Φ_c is the eigenvectors of covariance matrix corresponding to b :

$$\Phi_c = \begin{pmatrix} \Phi_{c,s} \\ \Phi_{c,g} \end{pmatrix} \quad (32)$$

c is a vector of appearance model parameters controlling both the shape and texture of the models.

Using the linear nature of the model, the combined model including shape X and texture g can be expressed as:

$$X = \bar{X} + \Phi_s W^{-1} \Phi_{c,s} c \quad (33)$$

$$g = \bar{g} + \Phi_g \Phi_{c,g} c \quad (34)$$

Then a new image can be synthesised using equation (33) and (34) for a given c .

2.5 Fitting AAM to the input image

Fitting a AAMs to an image is considered to be a problem of minimizing the error between the input image and closest model instance [Wang et al., 2007]:

$$\delta I = I_{image} - I_{model} \quad (35)$$



Fig. 3. Results of tracking hand using AAM fitting.

where I_{image} is the texture vector of the input image, and I_{model} is the texture vector of the model instance. The goal is to adjust the appearance model parameters c to minimize $|\delta I|^2$. The simplest way is to construct a linear relationship:

$$\delta c = R\delta I \quad (36)$$

where R can be computed by the following process:

- Suppose c_0 is the model parameter of the current image, new model parameter c can be generated by varying c_0 :

$$c = c_0 + \delta c \quad (37)$$

- Generate new shape model X and normalized texture model g_m according to equation (33) and (34).
- Deform the current image and get the corresponding texture model g_i , the difference vector can be written as:

$$\delta g = g_i - g_m \quad (38)$$

δg will change along with the variation of δc and the shape model X . The fitting procedure is shown in Table 2 and the object tracking results are shown in Fig. 3 and Fig. 4. From the tracking results we can see that when the camera moves around the scene, the tracking results are satisfying.

3. Factorization algorithm

The flowchart of the factorization algorithm is shown in Fig. 5, and the detailed demonstration will be given in the following sections.

3.1 Basic knowledge

The 3D shape of the non-rigid object can be described as a key frame basis set S_1, S_2, \dots, S_K . Each key frame basis S_i is a $3 \times P$ matrix describing P points. The 3D shape of a specific configuration is a linear combination of the basis set:

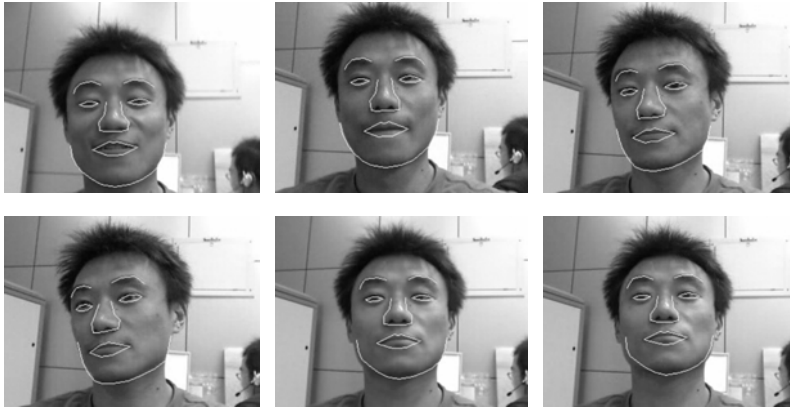


Fig. 4. Results of tracking face using AAM fitting.

-
- Step 1: Generate the normalized texture vector g_m .
- Step 2: Sample the image g_i below the model shape.
- Step 3: Evaluate the error vector $\delta_{g_0} = g_i - g_m$.
- Step 4: Evaluate the error $E_0 = |\delta_{g_0}|$.
- Step 5: Calculate the pose displacement $\delta_t = R_t \delta_{g_0}$.
- Step 6: Calculate the displacement in model parameters $\delta_c = R_c \delta_{g_0}$.
- Step 7: Set $i = 1$.
- Step 8: Update model parameters $c = c - k_i \delta_c$.
- Step 9: Transform the shape to invert the δ_t transformation.
- Step 10: Repeat step 1-4 to form a new error E_i .
- Step 11: If $E_i > E_0$ set $i = i + 1$ and go to step 8.
- Step 12: Accept the new estimate.
- where $k = [1.5, 0.5, 0.125, 0.0125, 0.0625]^T$ is the damping vector.
-

Table 2. The procedure of fitting the input image to the model instance.

$$S = \sum_{i=1}^K l_i \cdot S_i \quad S, S_i \in R^{3 \times P}, l_i \in R \quad (39)$$

where $S = \begin{bmatrix} x_1 & x_2 & \dots & x_p \\ y_1 & y_2 & \dots & y_p \\ z_1 & z_2 & \dots & z_p \end{bmatrix}$.

Under a scaled orthographic projection, the P points of S are projected into 2D image points (u_i, v_i) :

$$\begin{bmatrix} u_1 & u_2 & \dots & u_p \\ v_1 & v_2 & \dots & v_p \end{bmatrix} = R \left(\sum_{i=1}^K l_i S_i \right) + T \quad (40)$$

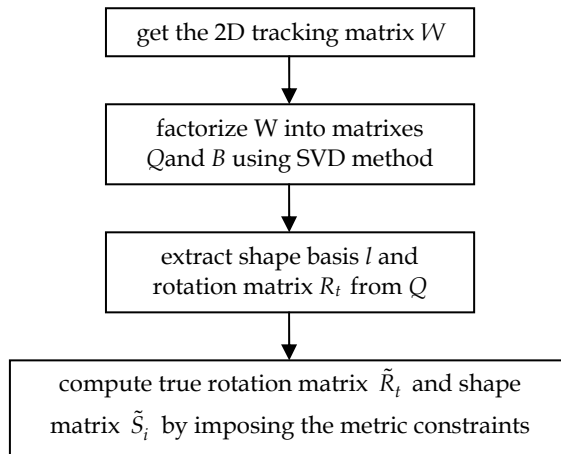


Fig. 5. The flowchart of the factorization algorithm.

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \end{bmatrix}, T = [t_1 \quad t_2 \quad t_3] \tag{41}$$

R contains the first two rows of the 3D camera rotation matrix. T is the camera translation matrix. As mentioned in [Tomasi & Kanade, 1992], we eliminate the camera translation matrix by subtracting the mean of all 2D points, and henceforth can assume that the 3D shape S is centred at the origin.

$$\begin{bmatrix} u'_1 & u'_2 & \dots & u'_p \\ v'_1 & v'_2 & \dots & v'_p \end{bmatrix} = \begin{bmatrix} u_1 - \bar{u} & u_2 - \bar{u} & \dots & u_p - \bar{u} \\ v_1 - \bar{v} & v_2 - \bar{v} & \dots & v_p - \bar{v} \end{bmatrix} \tag{42}$$

where $\bar{u} = \sum_{i=1}^p u_i$, $\bar{v} = \sum_{i=1}^p v_i$.

Therefore, we can rewrite equation (40) as:

$$\begin{bmatrix} u'_1 & u'_2 & \dots & u'_p \\ v'_1 & v'_2 & \dots & v'_p \end{bmatrix} = R \left(\sum_{i=1}^K l_i S_i \right) \tag{43}$$

Rewrite the linear combination in equation (43) as a matrix-matrix multiplication:

$$\begin{bmatrix} u'_1 & u'_2 & \dots & u'_p \\ v'_1 & v'_2 & \dots & v'_p \end{bmatrix} = [l_1 R \quad l_2 R \quad \dots \quad l_K R] \cdot \begin{bmatrix} S_1 \\ S_2 \\ \dots \\ S_K \end{bmatrix} \tag{44}$$

The 2D image points in each frame can be obtained using AAM. The tracked 2D points in frame t can be denoted as (u_{ti}, v_{ti}) . The 2D tracking matrix of N frames can be written as:

$$W = \begin{bmatrix} u'_{11} & u'_{12} & \dots & u'_{1P} \\ v'_{11} & v'_{12} & \dots & v'_{1P} \\ u'_{21} & u'_{22} & \dots & u'_{2P} \\ v'_{21} & v'_{22} & \dots & v'_{2P} \\ \dots & \dots & \dots & \dots \\ u'_{N1} & u'_{N2} & \dots & u'_{NP} \\ v'_{N1} & v'_{N2} & \dots & v'_{NP} \end{bmatrix} \tag{45}$$

Using equation (44) we can rewrite equation (45) as:

$$W = \begin{bmatrix} l_{11}R_1 & l_{12}R_1 & \dots & l_{1K}R_1 \\ l_{21}R_2 & l_{22}R_2 & \dots & l_{2K}R_2 \\ \dots & \dots & \dots & \dots \\ l_{N1}R_N & l_{N2}R_N & \dots & l_{NK}R_N \end{bmatrix} \cdot \begin{bmatrix} S_1 \\ S_2 \\ \dots \\ S_K \end{bmatrix} \tag{46}$$

where R_t denotes the camera rotation of frame t . l_{ti} denotes the shape parameter l_i of frame t .

3.2 Solving configuration weights using factorization

Equation (46) shows that the 2D tracking matrix W has rank $3K$, and can be factored into the product of two matrixes: $W=Q \cdot B$, where Q is a $2N \times 3K$ matrix, B is a $3K \times P$ matrix. Q contains the camera rotation matrix R_t and configuration weights $l_{t1}, l_{t2}, \dots, l_{tK}$ of each frame. B contains the information of shape basis S_1, S_2, \dots, S_K . The factorization can be done using singular value decomposition (SVD) method:

$$W^{2N \times P} = \tilde{U} \cdot \tilde{D} \cdot \tilde{V}^T = \tilde{Q}^{2N \times 3K} \cdot \tilde{B}^{3K \times P} \tag{47}$$

Then the camera rotation matrix R_t and shape basis weights l_{ti} of each frame can be extracted from the matrix \tilde{Q} :

$$q_t = [l_{t1}R_t \ l_{t2}R_t \ \dots \ l_{tK}R_t] = \begin{bmatrix} l_1r_1 & l_1r_2 & l_1r_3 & \dots & l_1r_4 & l_1r_5 & l_1r_6 \\ l_1r_4 & l_1r_5 & l_1r_6 & \dots & l_1r_7 & l_1r_8 & l_1r_9 \end{bmatrix} \tag{48}$$

Transform q_t into a new matrix \tilde{q}_t :

$$\tilde{q}_t = \begin{bmatrix} l_1r_1 & l_1r_2 & l_1r_3 & l_1r_4 & l_1r_5 & l_1r_6 \\ l_2r_1 & l_2r_2 & l_2r_3 & l_2r_4 & l_2r_5 & l_2r_6 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ l_Kr_1 & l_Kr_2 & l_Kr_3 & l_Kr_4 & l_Kr_5 & l_Kr_6 \end{bmatrix} = \begin{bmatrix} l_{t1} \\ l_{t2} \\ \vdots \\ l_{tK} \end{bmatrix} \cdot [r_{t1} \ r_{t2} \ r_{t3} \ r_{t4} \ r_{t5} \ r_{t6}] \tag{49}$$

here, the \tilde{q}_t can be factored using SVD method.

3.3 Solving true rotation matrix and shape basis

As mentioned in [Tomasi & Kanade, 1992], the matrix \tilde{R}_t is a linear transformation of the true rotation matrix R_t . Likewise, \tilde{S}_i is a linear transformation of the true shape matrix S_i :

$$R_t = \tilde{R}_t \cdot G, S_i = G^{-1} \cdot \tilde{S}_i \quad (50)$$

where $G^{3 \times 3}$ is found by solving a nonlinear data-fitting problem. In each frame we need to constrain the rotation matrix to be orthonormal. The constraints of frame t are:

$$[r_{t1} \ r_{t2} \ r_{t3}]GG^{-1}[r_{t1} \ r_{t2} \ r_{t3}]^T = 1 \quad (51)$$

$$[r_{t4} \ r_{t5} \ r_{t6}]GG^{-1}[r_{t4} \ r_{t5} \ r_{t6}]^T = 1 \quad (52)$$

$$[r_{t1} \ r_{t2} \ r_{t3}]GG^{-1}[r_{t4} \ r_{t5} \ r_{t6}]^T = 0 \quad (53)$$

In summary, given 2D tracking data W , we can get the 3D shape basis \tilde{S}_i , camera rotation matrix \tilde{R}_t and configuration weights l_{ti} of each training frame simultaneously using factorization method.

4. Non-rigid registration method

The 2D tracking data can be obtained using the AAM algorithm mentioned in section 2. With the use of the factorization method mentioned in section 3, we can acquire the 3D shape basis. The 3D shape is represented as 3D points in the world coordinates. Given the configuration weights, the 3D shape can be recovered by linear combination of the 3D shape basis. By projecting the 3D points to the 2D image with known camera rotation matrix (suppose the intrinsic camera matrix has been calculated), the estimated 2D points can be acquired. If the error between the 2D tracking data and the estimated 2D points is small enough, we can accept the configuration weights and the rotation matrix. Finally, the virtual object can be overlaid to the real scene using the camera rotation matrix.

The initial configuration weights and camera rotation matrix can not be precise. Optimization of the configuration weights should be done to minimize the error between the 2D tracking points detected by AAM and the estimated 2D points which is projected by the 3D shape points. This is a non-linear optimization problem which can be successfully solved by the optimization methods. Different with [Zhu et al., 2006], we use the Levenberg-Marquardt algorithm. Equation (54) shows the cost function.

$$\sum_j \|s_j - s'_j\|^2 \rightarrow \min = \sum_{j=1}^N \left\| s_j - \left(R \sum_{i=1}^K l_i S_i \right) \right\|^2 \rightarrow \min \quad (54)$$

where s_j is the 2D tracking data j , s'_j is the projected point.

The procedure of non-rigid registration is shown in Table 3.

-
- Step 1: Track the 2D points s_j using AAM.
- Step 2: Initialize the configuration weights l_i .
- Step 3: Initialize the camera rotation matrix R .
- Step 4: Calculate the 3D shape $S = \sum_{i=1}^K l_i S_i$.
- Step 5: Project the 3D points S to 2D image: $s'_j = A[R|T] \cdot S$.
- Step 6: Evaluate the projection error $E = \sum_{j=1}^N \|s_j - s'_j\|^2$.
- Step 7: If E is not small enough, improve l_i and R , then repeat step 4-6.
- Step 8: Accept the configuration weights l_i , and the camera rotation matrix R .
- Step 9: Overlaid the virtual object to the scene.
-

Table 3. The procedure of non-rigid registration.

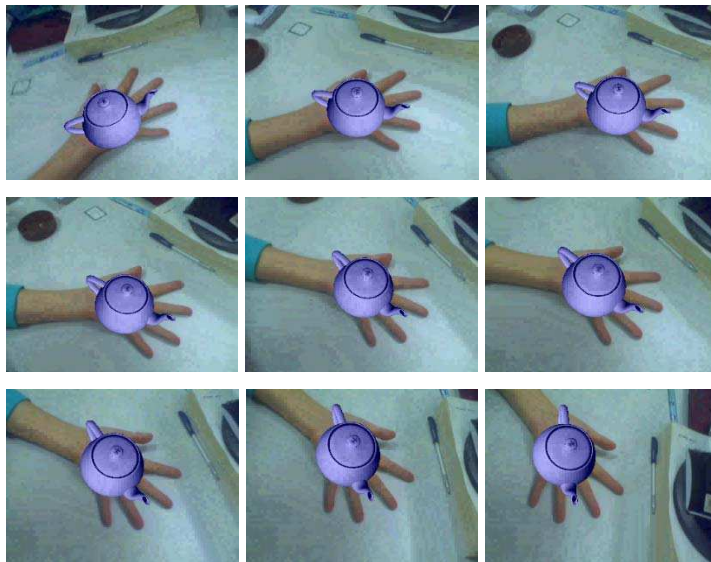


Fig. 6. Examples of augmented images using our method.

Furthermore, we should take the orthonormality of rotation matrix into consideration. The proposed method has been implemented in C using OpenGL and OpenCV on a DELL workstation (CPU 1.7G×2, RAM 1G).

In offline stage, we construct the AAM hand models using 7 training images which are manually labelled with 36 landmarks as shown in Fig. 1. We establish the hand shape basis using 300 training frames which is captured with a CCD camera. In online stage, the 2D points are tracked using AAM algorithm, and then Levenberg-Marquardt algorithm is used to optimize the parameters. Our experiment results are shown in Fig. 6. From the results we can see that the virtual teapot can be overlaid on the hand accurately when the camera moves around the scene.

5. Conclusion

In this chapter, a non-rigid registration method for augmented reality applications using AAM and factorization method is proposed. The process is divided into two stages: offline stage and online stage. In the offline stage, the 3D shape basis is constructed. To obtain the shape basis of the object, we firstly factorize the 2D data matrix tracked by the AAM into the product of two matrixes. One matrix contains the camera rotation matrix and the configuration weights, and the other matrix contains the shape basis. Then the rotation matrix and the configuration weights can be separated using SVD method. Finally the orthonormality of the rotation matrix should be the constraints to get the true rotation matrix and configuration weights. In online stage, the 3D pose parameters and the shape coefficients are estimated. The purpose is to minimize the error between the 2D tracking points detected by AAM and the estimated 2D points which is projected by the 3D shape points. The Levenberg-Marquardt method is used to solve this problem. The rotation matrix and the configuration weights are optimized. Some experiments have been conducted to validate that the proposed method is effective and useful for non-rigid registration in augmented reality applications.

6. Acknowledgement

This work is supported in part by National Natural Science Foundation of China with project No. 60903095; in part by Postdoctoral Science Foundation of China with project No. 20080440941.

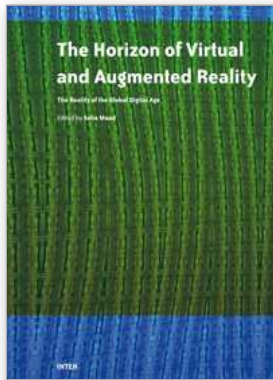
7. References

- Ahlberg, J. (2001). Using the active appearance algorithm for face and facial feature tracking. *Proceedings of the 2nd International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pp. 68-72, ISBN: 0-7695-1074-4, Vancouver, Canada, July 2001, IEEE Computer Society, Los Alamitos.
- Andresen, P. R. & Nielsen, M. (2001). Non-rigid registration by geometry-constrained diffusion. *Medical Image Analysis*, Vol. 5, No. 2, 81-88, ISSN: 1361-8415.
- Azuma, R. T. (1997). A survey of augmented reality. *Presence-Teleoperators and Virtual Environments*, Vol. 6, No. 4, 355-385, ISSN: 1054-7460.

- Azuma, R. T. (2001). Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, Vol. 21, No. 6, 34–47, ISSN: 0272-1716.
- Bartoli, A., von Tunzelmann, E. & Zisserman, A. (2004). Augmenting images of non-rigid scenes using point and curve correspondences. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 699-706, ISBN: 0-7695-2158-4, Washington, DC, June-July 2004, IEEE Computer Society, Los Alamitos.
- Bookstein, F. L. (1996). Landmark methods for forms without landmarks: localizing group differences in outline shape. *Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pp. 25–244, ISBN: 0-8186-7367-2, San Francisco, CA, June 1996, IEEE Computer Society, Los Alamitos.
- Bregler, C., Hertzmann, A. & Biermann, H. (2000). Recovering non-rigid 3d shape from image streams. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 690-696, ISSN: 1063-6919, Hilton Head Island, SC, June 2000, IEEE Computer Society, Los Alamitos.
- Cootes, T. F., Edwards, G. J. & Taylor, C. J. (1998). Active appearance models. *Lecture Notes in Computer Science*, Vol.1407, 484-498, ISSN: 0302-9743.
- Cootes, T. F., Edwards, G. J. & Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 6, 681-685, ISSN: 0162-8828.
- Cootes, T. F. & Taylor, C. J. (2001). Statistical models of appearance for medical image analysis and computer vision. *Proceedings of the Society of Photo-optical Instrumentation Engineers*, pp. 236-248, ISBN: 0-8194-4008-6, San Diego, CA, February 2001, SPIE-INT Society Engineering, Bellingham.
- Duta, N., Jain, A. K. & Dubuisson-Jolly, M.-P. (1999). Learning 2D shape models. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 8-14, ISSN: 1063-6919, Fort Collins, USA, June 1999, IEEE Computer Society, Los Alamitos.
- Guan, T., Li L. J. & Wang, C. (2008a). Registration using multiplanar structures for augmented reality systems. *Journal of Computing and Information Science in Engineering*, Vol. 8, No. 4, 041002-1~041002-6, ISSN: 1530-9827.
- Guan, T., Li, L. J. & Wang, C. (2008b). Robust estimation of trifocal tensors using natural features for augmented reality systems. *Computing and Informatics*, Vol. 27, No. 6, 891-911, ISSN: 1335-9150.
- Lepetit, V. & Fua, P. (2005). Monocular model-based 3D tracking of rigid objects: a survey. *Foundations and Trends in Computer Graphics and Vision*, Vol. 1, No. 1, 1–89, ISSN: 1572-2740.
- Lepetit, V., Laguer, P. & Fua, P. (2005). Randomized trees for real-time keypoint recognition. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 775-781, ISBN: 0-7695-2372-2, San Diego, CA, June 2005, IEEE Computer Society, Los Alamitos.
- Li, L. J., Guan, T., Ren, B. et al. (2008). Registration based on Euclidean reconstruction and natural features tracking for augmented reality systems. *Assembly Automation*, Vol. 28, Issue 4, 340–347, ISSN: 0144-5154.

- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, Vol. 20, No. 2, 91-110, ISSN: 0920-5691.
- Markin, E. & Prakash, E. C. (2006). Tracking facial features with occlusions. *Journal of Zhejiang University: Science A*, Vol. 7, No. 7, 1282-1288, ISSN: 1009-3095.
- Pilet, J., Lepetit, V. & Fua, P. (2005). Real-time non-rigid surface detection. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 822-828, ISBN: 0-7695-2372-2, San Diego, CA, June 2005, IEEE Computer Society, Los Alamitos.
- Pilet, J., Lepetit, V. & Fua, P. (2007). Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision*, Vol. 76, No. 2, 109-122, ISSN: 0920-5691.
- Rosten, E. & Drummond, T. (2005). Fusing points and lines for high performance tracking. *Proceedings of 10th IEEE International Conference on Computer Vision*, pp. 1508-1515, ISBN: 0-7695-2334-X, Beijing, China, October 2005, IEEE Computer Society, Los Alamitos.
- Scaroff, S. & Pentland, A. P. (1995). Modal matching for correspondence and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 7, 545-561, ISSN: 0162-8828.
- Sung, J. & Kim, D. J. (2004). Extension of AAM with 3D shape model for facial shape tracking. *Proceedings of International Conference on Image Processing*, pp. 3363-3366, ISBN: 0-7803-8554-3, Singapore, October 2004, IEEE, New York.
- Tian, Y., Li, Z. Y., Liu, W. et al. (2008). Non-rigid registration using AAM and factorization method for augmented reality applications. *Proceedings of the 2008 12th International Conference on Computer Supported Cooperative Work in Design*, pp. 705-709, ISBN: 978-1-4244-1650-9, Xian, China, April 2008, IEEE, New York.
- Tomasi, C. & Kanade, T. (1992). Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, Vol. 9, No. 2, 137-154, ISSN: 0920-5691.
- Walker, K.N., Cootes, T. F. & Taylor, C. J. (2000). Determining correspondences for statistical models of appearance. *Proceedings of 6th European Conference on Computer Vision*, pp. 829-843, ISBN: 3-540-67685-6, Dublin, Ireland, June 2000, Springer, Berlin.
- Wang, S. C., Wang, Y. S., & Chen, X. L. (2007). Weighted active appearance models. *3rd International Conference on Intelligent Computing*, pp. 1295-1304, ISBN:978-3-540-74170-1, Qingdao, China, August 2007, Springer, Berlin.
- Xiao, J., Baker, S., Matthews, I., & Kanade, T. (2004). Real-time combined 2D+3D active appearance models. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 535-542, ISBN: 0-7695-2158-4, Washington, DC, June-July 2004, IEEE Computer Society, Los Alamitos.
- Yuan, M. L., Ong, S. K. & Nee, A. Y. C. (2005). Registration based on projective reconstruction technique for augmented reality systems. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 11, No. 3, 254-264, ISSN: 1077-2626.

- Yuan, M. L., Ong, S. K. & Nee, A. Y. C. (2006). Registration using natural features for augmented reality systems. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 12, No. 4, 569-580, ISSN: 1077-2626.
- Zhu, J. K., Hoi, S. C. H., & Lyu, M. R. (2006). Real-time non-rigid shape recovery via active appearance models for Augmented Reality. *9th European Conference on Computer Vision*, pp. 186-197, ISBN: 3-540-33832-2, Graz, Austria, May 2006, Springer, Berlin.



Augmented Reality

Edited by Soha Maad

ISBN 978-953-7619-69-5

Hard cover, 230 pages

Publisher InTech

Published online 01, January, 2010

Published in print edition January, 2010

Virtual Reality (VR) and Augmented Reality (AR) tools and techniques supply virtual environments that have key characteristics in common with our physical environment. Viewing and interacting with 3D objects is closer to reality than abstract mathematical and 2D approaches. Augmented Reality (AR) technology, a more expansive form of VR is emerging as a cutting-edge technology that integrates images of virtual objects into a real world. In that respect Virtual and Augmented reality can potentially serve two objectives: reflecting realism through a closer correspondence with real experience, and extending the power of computer-based technology to better reflect abstract experience. With the growing amount of digital data that can be stored and accessed there is a rising need to harness this data and transform it into an engine capable of developing our view and perception of the world and of boosting the economic activity across domain verticals. Graphs, pie charts and spreadsheet are not anymore the unique medium to convey the world. Advanced interactive patterns of visualization and representations are emerging as a viable alternative with the latest advances in emerging technologies such as AR and VR. And the potential and rewards are tremendous. This book discusses the opportunities and challenges facing the development of this technology.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yuan Tian, Tao Guan and Cheng Wang (2010). AAM and Non-rigid Registration in Augmented Reality, Augmented Reality, Soha Maad (Ed.), ISBN: 978-953-7619-69-5, InTech, Available from:

<http://www.intechopen.com/books/augmented-reality/aam-and-non-rigid-registration-in-augmented-reality>

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.