Structure and motion of nonrigid object under perspective projection

Guanghui Wang a,b,*, Hung-Tat Tsui b, Zhanyi Hu c

a Department of Control Engineering, Aviation University of Airforce, No. 2222, Nanhu Road, Changchun, Jilin Province 130022, PR China
b Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N.T. Hong Kong
c National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, PR China

Received 7 August 2005; received in revised form 5 July 2006
Available online 28 November 2006
Communicated by S. Dickinson

Abstract

The paper focuses on the problem of structure and motion of nonrigid object from image sequence under perspective projection. Many previous methods on this problem utilize the extension technique of SVD factorization based on rank constraint to the tracking matrix, where the 3D shape of nonrigid object is expressed as a weighted combination of a set of shape bases. All these solutions are based on the assumption of Affine camera model. This assumption will become invalid and cause large reconstruction errors when the object is close to the camera. In this paper, we propose two algorithms, namely the linear recursive estimation and the nonlinear optimization, to extend these methods to general perspective camera model. Both algorithms are based on the shape and motion of weak perspective projection. The former one updates the solutions from weak perspective to perspective projection by refining the scalars corresponding to the projective depths recursively. The latter one is based on nonlinear optimization by minimizing the perspective reprojection residuals. Extensive experiments on simulated data and real image sequences are performed to validate the effectiveness of our new algorithms and noticeable improvements over the previous solutions are observed.

Keywords: Structure from motion; Nonrigid object; Perspective projection; Weak perspective projection; Singular value decomposition; Geometrical optimization

1. Introduction

The problem of recovering both structure and motion of an object from a sequence of its images is an important and essential task of computer vision. During the last two decades, many approaches have been proposed for different applications. Among them, factorization based methods are widely studied and attracted much attention in the computer vision society. Since this kind of approaches deal with all the data in all images uniformly, good robustness and accuracy can usually be achieved.

The factorization method was first proposed by Tomasi and Kanade (1992) in the early 1990s. The main idea of this algorithm is the use of rank constraint on the tracking matrix, which are composed of all the features tracked across the entire sequence, to factorize it into the motion and structure matrices simultaneously via singular value decomposition (SVD). The algorithm assumes an orthographic projection model. It was then extended to paraperspective and weak perspective projection model by Poelman and Kanade (1997). Christy and Horaud (1996) proposed an iterative method to estimate the Euclidean structure under perspective camera model by incrementally performing the reconstruction process with either a weak perspective or a paraperspective projection. A similar method to attain perspective solution from paraperspective reconstruction was also studied by Fujiki and Kurata

* Corresponding author. Address: Department of Control Engineering, Aviation University of Airforce, No. 2222, Nanhu Road, Changchun, Jilin Province 130022, PR China.
E-mail address: ghwanghk@gmail.com (G. Wang).

0167-8655/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2006.09.006
The above methods assume the camera is calibrated. In case of uncalibrated camera, Triggs (1996) proposed to achieve the projective reconstruction via a similar factorization scheme, where the projective depths of the features are computed via epipolar geometry. The method was further studied by Heyden and Berthilsson (1999) and Mahamud and Hebert (2000).

The above methods work only for rigid object and static scenes. While in real world, many objects are nonrigid and the scenes are dynamic. Examples include human faces carrying different expressions, lip movements, a walking person, moving vehicles, etc. Many extensions stemming from the factorization algorithm were proposed to relax the rigidity constraint. Costeira and Kanade (1998) first discussed how to recover the motion and shape of several independent moving objects via factorization under orthographic projection. The same problem was further discussed by Han and Kanade (2000). Bascle et al. (1998) proposed a method for factoring facial expressions and poses based on a set of preselected basis images.

Recent works on nonrigid factorization by Bregler et al. (2000) demonstrated that the 3D shape of nonrigid objects may be expressed as a weighted combination of a set of shape bases. Then the shape bases, weighting coefficients, and camera motions were factorized simultaneously under the rank constraint of the tracking matrix. Following this idea, Torresani et al. (2001) introduced an iterative algorithm to optimize the recovered shape and motion. Brand (2001) generalized the method and proposed nonrigid subspace flow for the searching of correspondences. The above methods use only the orthonormal (rotation) constraint to the Euclidean reconstruction after SVD factorization. This may cause ambiguity to the combination of shape bases. To solve this ambiguity, Xiao et al. (2004) introduced the basis constraint to ensure the uniqueness of solutions. They also studied the degenerate cases of nonrigid deformation in (Xiao and Kanade, 2004). However, it is difficult to select the shape bases automatically in real applications. Bue and Agapito (2004) extended the method to the stereo camera setup and introduced a nonlinear optimization step to select the correct shape and motion. There are also some other methods for the nonrigid structure from motion problem. Torresani et al. (2003) modeled the shape motion as a rigid component combined with a nonrigid deformation and presented an algorithm for learning the time-varying shape. More recently, Brand (2005) proposed a method for 3D factorization of nonrigid motion by directly minimizing deviation from the required orthogonal structure of the projection matrix.

To our best knowledge, most previous factorization methods for nonrigid shape and motion recovery are based on Affine camera model. This is a zero-order (for weak perspective) or first-order (for paraperspective) approximation to the real imaging conditions and is only valid when the depth variation of the object is small compared to the distance between the object and the camera (Hartley and Zisserman, 2000). Therefore, large reconstruction errors may appear when the object is close to the camera. In this paper, we will extend the algorithm in (Christy and Hornaud, 1996) to the nonrigid case so as to upgrade the solutions under Affine projection to those under general perspective projection. Two kinds of algorithms, namely the linear recursive approximation and the nonlinear optimization, are proposed. Experiments demonstrate the effectiveness of the algorithms and improvements to the final solutions.

The remaining parts of the paper are organized as follows. We first give some background on nonrigid factorization under weak perspective assumption in Section 2. Then we present the linear recursive algorithm by analyzing the relationship between perspective and weak perspective projection in Section 3. In Section 4, we give the nonlinear optimization algorithm that upgrades the weak perspective solution to perspective case. Some experiment results on synthetic data and real image sequences are given in Section 5 and Section 6 respectively. Finally, the conclusions of this paper are given in Section 7.

2. Background on nonrigid factorization

Suppose the camera is calibrated and we are working with normalized image coordinates. Given a sequence of m video frames and n tracked feature points across the sequence. Let \( \{x_i^{(j)} \in \mathbb{R}^2\}_{j=1}^{m} \) be the normalized coordinates of these features. The aim is to recover the shape \( S_j^{(j)} \in \mathbb{R}^{3\times n} \) (i.e., the corresponding 3D coordinates of the features) and the motion \( R_j \in \mathbb{R}^{3\times 3} \) associated with each frame.

Suppose the nonrigid shape can be represented as a weighted combination of k shape bases \( \{\omega_j^{(j)} \in \mathbb{R}\}_{j=1}^{k} \), where, \( S_j \in \mathbb{R}^{3\times n} \) is one of the shape bases, \( \omega_j^{(j)} \in \mathbb{R} \) is the deformation weight (a perfect rigid object would correspond to the case of \( k = 1 \) and \( \omega_j^{(j)} = 1 \)). Under the weak perspective assumption, we have

\[
\begin{bmatrix}
    x_1^{(j)} \\
    x_2^{(j)} \\
    \vdots \\
    x_n^{(j)}
\end{bmatrix}
= R_w^{(j)} \left[ \sum_{i=1}^{k} \omega_i^{(j)} S_i \right] + T_w^{(j)}, \quad (j = 1, \ldots, m)
\]

(1)

where \( R_w^{(j)} \) stands for the first two rows of the rotation matrix corresponding to the jth frame. \( T_w^{(j)} \) is the corresponding translation of the camera. If we register the coordinates of all image features to their centroid in each frame, then the translation term disappears (Poelman and Kanade, 1997). Thus we can obtain the factorization expression of the tracking matrix by stacking all equations in (1) frame by frame.

\[
\begin{bmatrix}
    x_1^{(1)} & \cdots & x_1^{(m)} \\
    \vdots & \ddots & \vdots \\
    x_n^{(1)} & \cdots & x_n^{(m)}
\end{bmatrix}_{2mn \times n}
= \begin{bmatrix}
    \omega_1^{(1)} R_w^{(1)} & \cdots & \omega_k^{(1)} R_w^{(1)} \\
    \vdots & \ddots & \vdots \\
    \omega_1^{(m)} R_w^{(m)} & \cdots & \omega_k^{(m)} R_w^{(m)}
\end{bmatrix}_{2mn \times 3k} \times
\begin{bmatrix}
    S_1 \\
    \vdots \\
    S_k
\end{bmatrix}_{3k \times n}
\]

(2)

The above equation can be written in short as \( W_{2mn \times n} = M_{2mn \times 3k} B_{3k \times n} \). It is easy to see from the right side of (2) that the rank of the tracking matrix \( W \) is at most 3k (usually 2n
and n are both larger than 3k). By performing singular value decomposition (SVD) on the tracking matrix and imposing the rank constraint, W may be factored as $W = M_{2m \times 3k} B_{3k \times n}$. However, the decomposition is not unique since it is only defined up to a nonsingular $3k \times 3k$ linear transformation as $W = MB = (MG)$ ($G^{-1}B$). If the transformation $G$ is known, the shape bases can be recovered from $B = G^{-1}B$, while the rotation matrix $R^{(i)}$ and the weighting coefficient $\omega^{(i)}$ may be recovered from $M = MG$ by Procrustes analysis, such as sub-block factorization (Brand, 2001; Bregler et al., 2000) or a further iterative optimization step proposed in (Torresani et al., 2001). For the computation of the transformation matrix $G$, many researchers (Brand, 2001; Bregler et al., 2000; Torresani et al., 2001) utilize the rotation constraint to the motion matrix. However, as proved by Xiao et al. (2004), only the rotation constraints may be insufficient when the object deforms at varying speed, and the basis constraint (Xiao et al., 2004) was introduced to solve the ambiguity in computing the correct transformation.

We can now obtain the solutions of the shape and motion corresponding to each frame. Nevertheless, due to the inherent property of the Affine camera model, a reversal ambiguity still remains for the recovered shapes and motions (Christy and Horaud, 1996). As one may see from (2), if we denote $(S^{(i)}, R^{(i)})$ as “positive” solution, then a “negative” solution $(-S^{(i)}, -R^{(i)})$ is also hold true to the algorithm.

3. Linear recursive estimation with perspective projection

Most previous algorithms are based on the assumption of Affine camera model. The assumption will become invalid and bring large reconstruction errors when the object is close to the camera. In this section, we will introduce a linear recursive algorithm to upgrade the solutions to general perspective projection.

Under the perspective projection, the mapping between a space point $X_i \in \mathbb{R}^3$ and its normalized image $x_i \in \mathbb{R}^2$ can be expressed as

$$s_i \hat{x}_i = RX_i + T,$$

where $s_i = [x_i^T, 1]^T$ is the homogeneous form of $x_i$, $R = [r_1^T, r_2^T, r_3^T]^T$ is the rotation matrix with $r_i$ the ith row of the matrix, $T = [t_x, t_y, t_z]^T$ is the translation vector of the camera, and $s_i$ is a nonzero scalar. Suppose the normalized image coordinates of $x_i$ is $[u_i, v_i]^T$, then we have

$$\begin{align*}
u_i &= \frac{r_1 X_i + t_z}{1 + r_3 X_i / t_z} = \frac{r_1 X_i / t_z + t_z / t_z}{1 + r_3 X_i / t_z}, \\
v_i &= \frac{r_2 X_i + t_y}{1 + r_3 X_i / t_z} = \frac{r_2 X_i / t_z + t_y / t_z}{1 + r_3 X_i / t_z}. \tag{4}
\end{align*}$$

Under the weak perspective assumption, the depth variation of the object is assumed small compared with the distance to the camera. This is equivalent to a zero-order approximation ($v_i = 0$) of the perspective projection. Let $\lambda_i = 1 + v_i$, then the relationship between the perspective and the weak perspective projections can be modeled as $\hat{x}_i = \lambda_i x_i$. For the feature points in the jth frame, the relationship can be expressed as

$$\hat{x}_i^j = \lambda_i^j x_i^j, \quad \lambda_i^j = 1 + v_i^j, (i = 1, \ldots, n, j = 1, \ldots, m). \tag{5}$$

Let us define a weighted tracking matrix as

$$\hat{W} = \begin{bmatrix}
\lambda_1^{(j)} x_1^{(j)} & \cdots & \lambda_n^{(j)} x_n^{(j)} \\
\vdots & \ddots & \vdots \\
\lambda_1^{(m)} x_1^{(m)} & \cdots & \lambda_n^{(m)} x_n^{(m)}
\end{bmatrix}_{2m \times n} \tag{6}$$

then if the scalars $\{\lambda_i^{(j)} \in \mathbb{R}\}_{i=1,\ldots,n}^{j=1,\ldots,m}$ are recovered consistently with (5), then the shape and motion obtained by the factorization of the weighted tracking matrix $\hat{W}$ would correspond to the solutions of the perspective projection. Here we adopt a similar method as (Christy and Horaud, 1996) (which is for rigid case) to estimate the scalars iteratively. The algorithm is summarized as follows:

Algorithm: Linear recursive estimation with perspective projection

Given the tracking matrix $W \in \mathbb{R}^{2m \times n}$, construct the weighted tracking matrix $\hat{W}$ according to (6) and set the initial value of $\lambda_i^{(j)} = 1$ for $i = 1, \ldots, n, j = 1, \ldots, m$. Repeat the following steps until the convergence of $\lambda_i^{(j)}$:

1. Update the weighted tracking matrix according to (6).
2. Recover the shape and motion via factorization of $\hat{W}$.
3. Estimate the new value of $\lambda_i^{(j)}$ according to (4) and (5).

A theoretical proof of the convergence of this algorithm is still an open problem. However, extensive simulations show that the algorithm converges rapidly if reasonable initial values are present. DeMenthon and Davis (1995) and Christy and Horaud (1996) also give a qualitative analysis on the convergence of this kind of recursive algorithm. In practice, one convenient way to determine the convergence of the algorithm is to check the scalar variations. Let us arrange all the scalars at the rth iteration into a matrix $A_r = [\lambda_i^{(j)}]_{i,n,j,m}^{r}$, the scalar variation is defined as $\delta = \left|\|A_r - A_{r-1}\|_F\right|$ (i.e., the Frobenius norm of the difference of the scalar matrix). Experiments show that the algorithm usually converges after 4 to 5 iterations.

Geometrical explanation of the algorithm: The geometrical explanation of the algorithm is clearly shown in Fig. 1. The initial tracking matrix is composed of the image point $x_0$, which is actually formed by the perspective projection. The weighted tracking matrix of (6), which may be composed of the point $x_i = \lambda_i x_0$, varies in accordance with the updated scalars of step 3. Upon convergence, the image points are modified to $\hat{x}_i$ such that they fit the weak perspective projection. Thus, the process of the algorithm is to recursively adjust the image coordinates from the
position of the perspective projection to that of the weak perspective by adjusting the value of $\hat{x}^{(i)}$. While the image of the centroid ($x_i$) remains untouched during iterations.

Dealing with the reversal ambiguity: In the algorithm, we do not take the reversal ambiguity as mentioned in Section 2 into consideration. However, this ambiguity may be easily solved out here. After recovering the shape and motion in step 2, we can reproject the “positive” and the “negative” solutions to each frame and form two new tracking matrices, say $W^+$ for the “positive” and $W^-$ for the “negative”. Check the Frobenius norms of $\|W - W^+\|_F$ and $\|W - W^-\|_F$. The one with the smaller error would be the correct solution.

4. Nonlinear optimization with perspective projection

One may have noted that the solution obtained by the linear recursive algorithm is just an approximation to the fully perspective projection. Here we will present a nonlinear optimization algorithm to upgrade the reconstruction from Affine to perspective projection.

Suppose the camera rotation, translation, shape bases and deformation weights under the perspective projection are $R^{(i)}$, $T^{(i)}$, $S_i$, $\omega_i^{(j)}$, respectively. Our goal is to recover these parameters by minimizing the image reprojection residual of the following cost function:

$$\left( R^{(i)}, T^{(i)}, S_i, \omega_i^{(j)} \right) = \|W - \hat{W}\|_F = \sum_{j=1}^m \left\| \hat{x}^{(j)} - N\left( \tilde{x}^{(j)} \right) \right\|_F^2,$$

$$s_i^{(j)} \hat{x}_i^{(j)} = R^{(i)} \sum_{l=1}^k \omega_l^{(j)} S_i + T^{(i)},$$

where $W$ denotes the reprojected tracking matrix, $\hat{x}^{(j)} = \left[ (x^{(j)})^T, 1 \right]^T$ is the homogeneous coordinates of the initial tracked feature points, $N(\bullet)$ denotes the normalization of a homogeneous vector so as to make its last element unit, $\tilde{x}$ is the homogeneous coordinates of the reprojected image point under the perspective projection, $s_i^{(j)}$ is a nonzero scalar.

The minimization process is also termed as bundle adjustment in computer vision society that can be solved via Newton iteration or other gradient descent methods. Here we employ the sparse Levenberg–Marquardt iteration method as given in (Hartley and Zisserman, 2000). During computation, the rotation matrix is parameterized by three parameters $\mathbf{R}^{(j)} = [R^{(i)1}, R^{(i)2}, R^{(i)3}]^T$ using the exponential map as

$$\mathbf{R}^{(j)} = \exp \left[ \begin{array}{ccc} 0 & l_2^{(j)} & -l_1^{(j)} \\ -l_2^{(j)} & 0 & l_3^{(j)} \\ l_1^{(j)} & -l_3^{(j)} & 0 \end{array} \right].$$

The method presented here is similar to that in (Bue et al., 2004). However, the work in (Bue et al., 2004) is designed to optimize the solutions of Affine reconstruction, while ours is to minimize the perspective reprojection error and take the Affine solutions as the initial values of iterations, thus the cost function (7) is much more complicated. In this paper, we assume the camera is calibrated, however, the camera parameters can also be optimized if we combine these parameters into the minimization process of (7). Suppose $K^{(j)}$ is the camera matrix of the $j$th frame, $\hat{m}^{(j)}$ is the homogeneous coordinates of a feature point in the frame measured in pixels, $\tilde{x}^{(j)}$ is the corresponding normalized image point, then we have $\tilde{x}^{(j)} = N((K^{(j)})^{-1} \hat{m}^{(j)})$, where $N(\bullet)$ is the normalization operator.

Compared with the linear recursive algorithm, the nonlinear method may converge to more accurate solutions of the perspective projection, since the algorithm minimizes a geometrical meaningful cost function. Nevertheless, the nonlinear method may lead to a local minimum when the initial values are not good. In practice, we can combine the two algorithms together, use the linear method to obtain initial values and then further refine them by the nonlinear method, or we may start with the solutions from the paraperspective assumption (Christy and Horaud, 1996; Fujiki and Kurata, 2000), since this assumption is a first-order approximation of the perspective projection. The nonlinear algorithm is usually computationally intensive compared with the linear recursive algorithm, interested readers are refereed to (Christy and Horaud, 1996, Hartley and Zisserman (2000) for a comparison of their complexities.

5. Experiments with synthetic data

During the simulations, we generate a cube in the space, whose dimension is of $20 \times 20 \times 20$ with 21 evenly distributed points on each side. The origin of the world coordinate system is set at the center of the cube. There are three sets of points ($33 \times 3$ points) on the adjacent three surfaces of the cube that move along the axes at constant speed toward outside as shown in Fig. 2. We generate 30 cubes (together with the three moving parts) in space with
randomly selected poses, then project each cube to an image by perspective projection, and there are 351 image points (252 points belong to the cube and the rest 99 points belong to the three moving parts) in each frame. During shooting, the distance of the camera to the object is set at about 14 times of the object size such that the imaging condition is quite close to the weak perspective assumption.

*Reconstruction results:* We recover the shape and motion corresponding to each frame using the algorithm of Section 2, and then upgrade the solutions from Affine to perspective projection according to the proposed algorithms in Sections 3 and 4. Fig. 2(a) and Fig. 2(b) show the recovered 3D shapes of two frames, respectively. We can see from the results that both the linear recursive and the nonlinear algorithms can achieve very good results, and they are both almost the same as the ground truths visually. But the errors of Affine reconstructions are obvious, as we can see from Fig. 2(a2) and Fig. 2(b2), the reconstructed shapes are not exactly cubes and the reconstructed lines are curved a little bit due to the perspective effect, although the camera setup in the simulation is very close to the weak perspective assumption.

One may have noted that each reconstructed shape in Fig. 2 is defined up to a 3D similarity transformation with the ground truth. For convenience of evaluation, we first compute the transformation matrix by virtue of the point correspondences between the recovered structure and its ground truth, then transform each reconstructed shape to the coordinate system of the associated ground truth, and calculate the distances between all the corresponding point pairs. Table 1 shows the mean and standard deviation of the distances associated with each recovered structure in Fig. 2. We can see from Table 1 that the structures recovered by the two proposed algorithms are more accurate than those by previous method based on weak perspective assumption.

*Convergence property of the proposed algorithms:* We upgrade the 3D shapes and motions of all the 30 frames to perspective projection by the two proposed algorithms. At each iteration, we record the scalar variation and the relative reprojection error which is defined as

$$\delta_i = \|W - W_i\|_F / \|W\|_F \times 100\%,$$

where, $W$ is the initial tracking matrix, $W_i$ is the projected tracking matrix at the $i$th iteration. The results are shown in Fig. 3.

It can be seen from Fig. 3 that the convergence speed of the two algorithms are very fast. The reprojection error of previous weak perspective factorization method corresponds to that at the first iteration as shown in Fig. 3(b) and Fig. 3(c), the error is reduced greatly by the proposed upgrading algorithms. There is no added noise in the test, however, a small residual error still exists after convergence, this is because the images are produced by general perspective projection rather than weak perspective projection. It is obviously that the residual error will increase with the increase of noise level. For the computation time

![Fig. 2. Synthetic data and reconstruction results of different algorithms corresponding to different frames. (a1) and (b1) the first and the 10th generated 3D shapes of the synthetic cubes (in dot) and the three sets of moving points (in circle); (a2) and (b2) the corresponding recovered shapes by weak perspective factorization of Section 2; (a3) and (b3) the upgraded perspective shapes by the linear recursive algorithm; (a4) and (b4) the upgraded shapes by the nonlinear optimization algorithm.](image-url)
of 10 iterations, the nonlinear algorithm takes 158.3 s, while the linear recursive method only takes 11.2 s on a Dell PC with Intel Xeon 2.6 GHz CPU programmed with Matlab 6.5.

More performance comparison of the algorithms: In this test, we use the same data to study the performance of different algorithms with respect to the relative distances (i.e., the ratio of the distance of the object to the camera over the object size). We vary the relative distance from 7 to 14 with a step of 1 so as to produce different sets of images. At each position, we first recover the Affine shape and motion of each frame by the method of Section 2, then upgrade the solution to perspective projection by the two proposed algorithms, and check the relative reprojection error of each solution according to (9). We also transform the reconstructed shape corresponding to the fifth frame to the coordinate system of its associated ground truth, and calculate the mean and standard deviation of the error distances between the transformed points and the associated ground truths. The results are shown in Table 2.

We can see from Table 2 that the reconstruction error would increase when we move the camera close to the object. The two proposed upgrading algorithms are all based on the initial estimation under weak perspective assumption, thus they may not converge to the correct solutions when the errors of initial values increase to a certain extent. As shown in Table 2, when the relative distance is set at 7, the two algorithms converge in 6 and 8 iterations respectively, however, they converge to false solutions due to bad initial values. Generally speaking, Good solutions can be guaranteed when the relative distance is greater than 8. We also find that the nonlinear method is more sensitive to the initial values than the linear recursive algorithm.

6. Experiments with real sequences

We tested the proposed methods on several real image sequences. Here we will only report the results of two sequences due to the space limitation.

Test on Franck sequence: The sequence is downloaded from the European working group on face and gesture recognition (www-prima.inrialpes.fr/fgnet/), whose resolution is of $720 \times 576$. We select 60 frames with various facial expressions for the experiment. The tracking data, which contain 68 automatically tracked feature points using the

![Fig. 3. Performance of the two upgrading algorithms. (a) The scalar variations of the linear recursive algorithm with respect to the iteration steps; (b) the relative reprojection error of the linear recursive algorithm at each iteration; (c) the relative reprojection error of the nonlinear optimization algorithm at each iteration.](image)

**Table 2**

<table>
<thead>
<tr>
<th>Relative distance</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error (%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear</td>
<td>1.2670</td>
<td>0.3186</td>
<td>0.2561</td>
<td>0.2193</td>
<td>0.1985</td>
<td>0.1758</td>
<td>0.1689</td>
<td>0.1538</td>
</tr>
<tr>
<td>Nonlinear</td>
<td>1.2576</td>
<td>0.3137</td>
<td>0.2512</td>
<td>0.2145</td>
<td>0.1924</td>
<td>0.1697</td>
<td>0.1618</td>
<td>0.1492</td>
</tr>
<tr>
<td>Mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weak</td>
<td>0.7128</td>
<td>0.6226</td>
<td>0.5510</td>
<td>0.4899</td>
<td>0.4487</td>
<td>0.4159</td>
<td>0.3841</td>
<td>0.3617</td>
</tr>
<tr>
<td>Linear</td>
<td>1.0685</td>
<td>0.0417</td>
<td>0.0361</td>
<td>0.0337</td>
<td>0.0312</td>
<td>0.0268</td>
<td>0.0184</td>
<td>0.0153</td>
</tr>
<tr>
<td>Nonlinear</td>
<td>1.0659</td>
<td>0.0396</td>
<td>0.0350</td>
<td>0.0225</td>
<td>0.0297</td>
<td>0.0254</td>
<td>0.0173</td>
<td>0.0148</td>
</tr>
<tr>
<td>Std</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weak</td>
<td>0.1685</td>
<td>0.1457</td>
<td>0.1332</td>
<td>0.1215</td>
<td>0.1124</td>
<td>0.1069</td>
<td>0.1017</td>
<td>0.0986</td>
</tr>
<tr>
<td>Linear</td>
<td>0.1670</td>
<td>0.0895</td>
<td>0.0639</td>
<td>0.0374</td>
<td>0.0269</td>
<td>0.027</td>
<td>0.0095</td>
<td>0.0079</td>
</tr>
<tr>
<td>Nonlinear</td>
<td>0.1658</td>
<td>0.0812</td>
<td>0.0625</td>
<td>0.0354</td>
<td>0.0251</td>
<td>0.0106</td>
<td>0.0086</td>
<td>0.0067</td>
</tr>
<tr>
<td>Iteration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Nonlinear</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

Where ‘Weak’ stands for the previous weak perspective factorization method, ‘Linear’ for the linear recursive algorithm, ‘Nonlinear’ for the nonlinear optimization algorithm, ‘Error’ for the relative reprojection error, ‘Mean’ and ‘Std’ for the mean and standard deviation of the distances respectively, ‘Iteration’ for the iteration times of convergence.
active appearance model (AAM) method, are also downloaded from the site. In this test, we use a simplified camera model with square pixel and the principal point at the image center, the intrinsic parameters of the camera is assumed to
be constant during shooting. The only unknown parameter (i.e., the focal length) is estimated by the method of Pollefeys et al. (1999) with rigid approximation.

Fig. 4 shows the reconstructed VRML models with texture mapping and the triangulated wireframes corresponding to five frames, which are obtained by the linear recursive algorithm with 10 iterations. The reconstruction results by other methods are omitted here, since there is no much difference visually. For a comparison, the relative reprojection errors of the weak perspective factorization, the linear recursive and the nonlinear optimization algorithm are 6.42%, 0.27%, 0.21%, respectively. We may see from the results that the improvements of the proposed algorithms are obvious. The models associated with different frames and different facial expressions are correctly recovered. The results could be used for visualization and recognition. However, the positions of some reconstructed features may not be very accurate. This is mainly caused by the errors of the tracked features and the camera parameters.

Test on scarf sequence: The sequence is taken by Canon PowerShot G3 with a resolution of 1024 × 768. The camera is pre-calibrated by Zhang’s method (Zhang, 2000). There are 15 frames in the sequence and the scarf is pressed during shooting so as to make it deforms a little bit. We utilize the method proposed by Yao and Cham (2005) to seek the correspondences between these frames and delete the outliers interactively. There are 2986 features tracked across the sequence as shown in Fig. 5. We recover the affine shape and motion of each frame by weak perspective factorization in Section 2, then upgrade the solution to that under general perspective projection by the linear recursive algorithm. Fig. 5 shows the reconstructed VRML models and wireframes corresponding to two frames of the sequence. The recovered 3D structures of the scarf with deformations are visually plausible and realistic.

As a comparison, the relative reprojection errors of the three algorithms are compared as shown in Table 3. The background of this sequence is two orthogonal sheets with square grids. We take this as a ground truth of evaluation, and calculate the reconstructed angle between the two diagonals of each square and the angle formed by the two diagonals.

The means of the length ratios and the formed angles are also shown in Table 3, from which we may see the noticeable improvements of the proposed algorithms over previous method based on Affine assumption.

7. Conclusions

In this paper, we have introduced two algorithms to update previous nonrigid factorization methods from Affine to fully perspective camera model. Experiment results with synthetic data and real images validate the algorithms and show the improvements over the previous methods. The test data and reconstructed models can be downloaded from the first author’s homepage. We are currently working towards an extension of the method to more challenging cases that usually happen in practice. These include the problems of how to deal with the outliers and the uncertainty in image measurement, and how to accurately self-calibrate the cameras from nonrigid sequences.

Acknowledgements

The authors would like to thank the reviewers for their valuable comments and suggestions. The work is supported by the National Natural Science Foundation of China under Grant no. 60575015 and the Hong Kong RGC Grant CUHK 4378/02E.

References


---

<table>
<thead>
<tr>
<th>Reprojection error (%)</th>
<th>Mean error of reconstructed angle (°)</th>
<th>Mean error of length ratios</th>
<th>Mean error of diagonal angles (°)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weak</td>
<td>4.946</td>
<td>3.455</td>
<td>0.126</td>
</tr>
<tr>
<td>Linear</td>
<td>0.409</td>
<td>1.736</td>
<td>0.085</td>
</tr>
<tr>
<td>Nonlinear</td>
<td>0.342</td>
<td>1.384</td>
<td>0.079</td>
</tr>
</tbody>
</table>