Recovery 3D Shape from a 2D Image Sequence

JIA Pei

Email: jp4work@gmail.com

Vision Open Working Group

First Edition

May 29, 2010

Contents

Abstract

Recovery of 3D face shape is one of the central goals of computer vision. Also, it's very important to face oriented HCI because the absolute orientation of the face estimated from a 3D model will always outperform the direct estimation from 2D images. Many methods have already been proposed to recover the 3D shape from 2D images including:

- those based on a single image [8–10], etc.
- those based on stereo vision [12], etc.
- those based on an image sequence by matrix factorization [2–4, 6, 7, 11, 13–15, 17, 20], etc.

Here in our application, 3D face shape recovery from an image sequence using factorization technology is adopted to synthesize the face 3D shape.

0.1 Projective Factorization

Projective factorization is first addressed by Tomasi and Kanade in [13]. Actually, Projective factorization describes a quite simple fact: every point on a 2D image $\mathbf{X} = (x, y)^T$ is a pure linear transformation from the corresponding 3D point $\hat{\mathbf{X}} = (\hat{x}, \hat{y}, \hat{z})^T$, which could be simply denoted as:

$$\mathbf{X} = \mathbf{P}(\hat{\mathbf{X}}) = a\mathbf{R}\hat{\mathbf{X}} + \mathbf{t} \tag{1}$$

where a is the scaling scalar parameter, $\mathbf{R}$ is a $2 * 3$ orthonormal projection matrix, $\mathbf{t}$ is a $2 * 1$ translation vector, and $\mathbf{P}$ describes a weak perspective transform composed by a, $\mathbf{R}$ and $\mathbf{t}$.

Concretely,

$$\begin{pmatrix} x \\ y \end{pmatrix} = a \begin{pmatrix} i_x & i_y & i_z \\ j_x & j_y & j_z \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{pmatrix} + \begin{pmatrix} {}^xO \\ {}^yO \end{pmatrix} \tag{2}$$

For plenty of points, say, v points, (1) could be extended to:

$$\begin{pmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_v \end{pmatrix} = \begin{pmatrix} \mathbf{P}(\hat{\mathbf{X}}_1) & \mathbf{P}(\hat{\mathbf{X}}_2) & \cdots & \mathbf{P}(\hat{\mathbf{X}}_v) \end{pmatrix} \tag{3}$$

Generally, for $\mathbf{P}(\hat{\mathbf{X}}_j) = a\mathbf{R}\hat{\mathbf{X}}_j + \mathbf{t}_j; j = 1, 2, \cdots, v$, the translation $\mathbf{t}_j$ could always be ignored because both the 2D and 3D shapes could always be centralized to the origin. Hence, (3) deduces to:

$$\begin{pmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_v \end{pmatrix} = \begin{pmatrix} a\mathbf{R}\hat{\mathbf{X}}_1 & a\mathbf{R}\hat{\mathbf{X}}_2 & \cdots & a\mathbf{R}\hat{\mathbf{X}}_v \end{pmatrix} \tag{4}$$

If we respectively reorganize the 2D and 3D shape defined in ASM (Refer to [**?**])

$$\mathbf{s} = \begin{pmatrix} x_1 & x_2 & \cdots & x_v \\ y_1 & y_2 & \cdots & y_v \end{pmatrix} \tag{5}$$

and

$$\hat{\mathbf{s}} = \begin{pmatrix} x_1 & x_2 & \cdots & x_v \\ y_1 & y_2 & \cdots & y_v \\ z_1 & z_2 & \cdots & z_v \end{pmatrix} \tag{6}$$

(4) could be rewritten to:

$$\mathbf{s} = a\mathbf{R}\hat{\mathbf{s}} \tag{7}$$

Considering the same 3D points in a number of image frames, say F frames, it's easy to deduce:

$$\mathbf{W} = \begin{pmatrix} \mathbf{s}^1 \\ \mathbf{s}^2 \\ \cdots \\ \mathbf{s}^F \end{pmatrix} = \begin{pmatrix} a^1\mathbf{R}^1\hat{\mathbf{s}}^1 \\ a^2\mathbf{R}^2\hat{\mathbf{s}}^2 \\ \cdots \\ a^F\mathbf{R}^F\hat{\mathbf{s}}^F \end{pmatrix} = \begin{pmatrix} a^1\mathbf{R}^1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & a^2\mathbf{R}^2 & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \cdots & a^F\mathbf{R}^F \end{pmatrix} \begin{pmatrix} \hat{\mathbf{s}}^1 \\ \hat{\mathbf{s}}^2 \\ \cdots \\ \hat{\mathbf{s}}^F \end{pmatrix} \tag{8}$$

In (8), $\mathbf{W}$ is called the measurement matrix, $\mathbf{R}^i$ defines the projection matrix for frame i, and $\hat{\mathbf{s}}^i$ defines the 3D shape in the ith frame, where $i = 1, 2, \cdots, F$.

Based on the 3D shape model, (8) could be rewritten as:

$$\begin{aligned} \mathbf{W} = \begin{pmatrix} \mathbf{s}^1 \\ \mathbf{s}^2 \\ \cdots \\ \mathbf{s}^F \end{pmatrix} = \begin{pmatrix} a^1\mathbf{R}^1\hat{\mathbf{s}}^1 \\ a^2\mathbf{R}^2\hat{\mathbf{s}}^2 \\ \cdots \\ a^F\mathbf{R}^F\hat{\mathbf{s}}^F \end{pmatrix} &= \begin{pmatrix} a^1\mathbf{R}^1(\hat{\mathbf{s}}_0 + \sum_{i=1}^N \hat{p}_i^1\hat{\mathbf{s}}_i) \\ a^2\mathbf{R}^2(\hat{\mathbf{s}}_0 + \sum_{i=1}^N \hat{p}_i^2\hat{\mathbf{s}}_i) \\ \cdots \\ a^F\mathbf{R}^F(\hat{\mathbf{s}}_0 + \sum_{i=1}^N \hat{p}_i^F\hat{\mathbf{s}}_i) \end{pmatrix} \\ = \begin{pmatrix} a^1\hat{p}_0^1\mathbf{R}^1 & a^1\hat{p}_1^1\mathbf{R}^1 & \cdots & a^1\hat{p}_N^1\mathbf{R}^1 \\ a^2\hat{p}_0^2\mathbf{R}^2 & a^2\hat{p}_1^2\mathbf{R}^2 & \cdots & a^2\hat{p}_N^2\mathbf{R}^2 \\ \cdots & \cdots & \cdots & \cdots \\ a^F\hat{p}_0^F\mathbf{R}^F & a^F\hat{p}_1^F\mathbf{R}^F & \cdots & a^F\hat{p}_N^F\mathbf{R}^F \end{pmatrix} &\begin{pmatrix} \hat{\mathbf{s}}_0 \\ \hat{\mathbf{s}}_1 \\ \cdots \\ \hat{\mathbf{s}}_N \end{pmatrix} = \mathbf{MB} \end{aligned} \tag{9}$$

where $\hat{p}_0^1 = \hat{p}_0^2 = \cdots = \hat{p}_0^F = 1$. Furthermore, if we define weight vectors $\mathbf{c}^i = a^i(\hat{p}_0^i \quad \hat{p}_1^i \quad \ldots \quad \hat{p}_N^i), i = 1, 2 \cdots F$; (9) could be rewritten to

$$\mathbf{W} = \mathbf{MB} = \begin{pmatrix} \mathbf{c}^1 \otimes \mathbf{R}^1 \\ \mathbf{c}^2 \otimes \mathbf{R}^2 \\ \cdots \\ \mathbf{c}^F \otimes \mathbf{R}^F \end{pmatrix} \begin{pmatrix} \hat{\mathbf{s}}_0 \\ \hat{\mathbf{s}}_1 \\ \cdots \\ \hat{\mathbf{s}}_N \end{pmatrix} \tag{10}$$

Apparently, $\mathbf{W}$ is a matrix of size $2F*v$, $\mathbf{M}$ is a matrix of size $2F*3(N+1)$, and $\mathbf{B}$ is a matrix of size $3(N+1)*v$.

Although one solution to (10) could easily be determined by SVD, it's not that easy to find the required solution due to the fact that there are countless solutions to this projective factorization. If we denote a pair of $\bar{\mathbf{M}}$ and $\bar{\mathbf{B}}$ as one solution, the pairs in form of $\bar{\mathbf{M}}\mathbf{G}$ and $\mathbf{G^{-1}}\bar{\mathbf{B}}$ are acceptable solutions as well, where $\mathbf{G}$ is the corrective transformation full-rank matrix of square size $3(N+1)*3(N+1)$. Our problem now comes to find the most suitable and unique $\mathbf{G}$, so that $\mathbf{M} = \bar{\mathbf{M}}\mathbf{G}$ and $\mathbf{B} = \mathbf{G^{-1}}\bar{\mathbf{B}}$ are the best solutions to recover 3D shape.

0.2 Rotation Constraints

Due to the orthogonormality of the rotation matrices and the transpose property of Kronecker product [18]:

$$\mathbf{M}\mathbf{M}^T = \begin{pmatrix} \mathbf{c}^1 \otimes \mathbf{R}^1 \\ \mathbf{c}^2 \otimes \mathbf{R}^2 \\ \cdots \\ \mathbf{c}^F \otimes \mathbf{R}^F \end{pmatrix} \begin{pmatrix} (\mathbf{c}^1)^T \otimes (\mathbf{R}^1)^T & (\mathbf{c}^2)^T \otimes (\mathbf{R}^2)^T & \cdots & (\mathbf{c}^F)^T \otimes (\mathbf{R}^F)^T \end{pmatrix}$$
$$= \begin{pmatrix} \begin{pmatrix} \sum_{k=0}^N (a^1 \hat{p}_k^1)^2 & 0 \\ 0 & \sum_{k=0}^N (a^1 \hat{p}_k^1)^2 \end{pmatrix} & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \begin{pmatrix} \sum_{k=0}^N (a^F \hat{p}_k^F)^2 & 0 \\ 0 & \sum_{k=0}^N (a^F \hat{p}_k^F)^2 \end{pmatrix} \end{pmatrix} \quad (11)$$

On the other hand, $\mathbf{M}\mathbf{M}^T = \bar{\mathbf{M}}\mathbf{G}\mathbf{G}^T\bar{\mathbf{M}}^T$, denote $\mathbf{G}\mathbf{G}^T = \mathbf{Q}$, we have $\mathbf{M}\mathbf{M}^T = \bar{\mathbf{M}}\mathbf{Q}\bar{\mathbf{M}}^T$. If we denote $\mathbf{A}(i, j)$ as the element of the ith row and jth column in matrix $\mathbf{A}$, then, we could deduce the constraints on $\mathbf{Q}$ as:

$$\begin{aligned} \bar{\mathbf{M}}\mathbf{Q}\bar{\mathbf{M}}^T(2i-1, 2i-1) &= \bar{\mathbf{M}}\mathbf{Q}\bar{\mathbf{M}}^T(2i, 2i) & (12) \\ \bar{\mathbf{M}}\mathbf{Q}\bar{\mathbf{M}}^T(2i-1, 2i) &= \bar{\mathbf{M}}\mathbf{Q}\bar{\mathbf{M}}^T(2i, 2i-1) = 0 & (13) \\ i &= 1, 2, \cdots, F \end{aligned}$$

$\mathbf{Q}$ is a symmetric matrix with $(1 + 3(N + 1)) * 3(N + 1)/2$ unknown elements. It appears that if $F \geq (1 + 3(N + 1)) * 3(N + 1)/4$, there will be adequate constraints to compute $\mathbf{Q}$, hence $\mathbf{G}$ as well. However, Xiao strictly proved in [20] that generally, no matter how many frames that you are choosing, namely, no matter how big the value F is, the solution space has a degree of freedom of $(2N+1)(N+1)$ and not all the solutions to the above rotation constraints are valid solutions. Conclusively, rotation constraints are not sufficient to afford a closed-form solution to $\mathbf{Q}$.

0.3 Triad Rotation Constraints

Apparently, $3(N+1)*3(N+1)$ matrix $\mathbf{G}$ could be rewritten as the summation of matrics $\mathbf{G}_k, k = 0, 1 \cdots, N$, which is composed by keeping the kth three columns (namely, a triad, denoted as $\mathbf{g}_k$) of $\mathbf{G}$ and replacing all other triads in $\mathbf{G}$ as $\mathbf{0}$. Namely:

$$\begin{aligned}\mathbf{G} =& \mathbf{G}_0 + \mathbf{G}_1 + \cdots + \mathbf{G}_N \\ =& \begin{pmatrix} \mathbf{g}_0 & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{g}_1 & \cdots & \mathbf{0} \end{pmatrix} + \cdots + \begin{pmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{g}_N \end{pmatrix}\end{aligned} \tag{14}$$

If we define $\mathbf{Q}_k = \mathbf{G}_k {\mathbf{G}_k}^T$, it's not hard to deduce:

$$\mathbf{Q} = \mathbf{G}\mathbf{G}^T = \sum_{k=0}^{N} \mathbf{G}_k {\mathbf{G}_k}^T = \sum_{k=0}^{N} \mathbf{Q}_k \tag{15}$$

Furthermore,

$$\mathbf{M} = \bar{\mathbf{M}}\mathbf{G} = \bar{\mathbf{M}} \sum_{k=0}^{N} \mathbf{G}_k = \sum_{k=0}^{N} \bar{\mathbf{M}}\mathbf{G}_k \tag{16}$$

Obviously,

$$\begin{pmatrix} \mathbf{0} & \cdots & \mathbf{0} & a^1 \hat{p}_k^1 \mathbf{R}^1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & a^2 \hat{p}_k^2 \mathbf{R}^2 & \mathbf{0} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \cdots & \mathbf{0} & a^F \hat{p}_k^F \mathbf{R}^F & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix} = \bar{\mathbf{M}}\mathbf{G}_k \tag{17}$$

and,

$$\bar{\mathbf{M}}\mathbf{G}_k {\mathbf{G}_k}^T \bar{\mathbf{M}}^T = \begin{pmatrix} \begin{pmatrix} (a^1 \hat{p}_k^1)^2 & 0 \\ 0 & (a^1 \hat{p}_k^1)^2 \end{pmatrix} & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \begin{pmatrix} (a^F \hat{p}_k^F)^2 & 0 \\ 0 & (a^F \hat{p}_k^F)^2 \end{pmatrix} \end{pmatrix} \tag{18}$$

Similar to (12) and (13), we have:

$$\begin{aligned} \bar{\mathbf{M}}\mathbf{Q}_k \bar{\mathbf{M}}^T (2i-1, 2i-1) &= \bar{\mathbf{M}}\mathbf{Q}_k \bar{\mathbf{M}}^T (2i, 2i) & (19) \\ \bar{\mathbf{M}}\mathbf{Q}_k \bar{\mathbf{M}}^T (2i-1, 2i) &= \bar{\mathbf{M}}\mathbf{Q}_k \bar{\mathbf{M}}^T (2i, 2i-1) = 0 & (20) \\ i &= 1, 2 \cdots F \quad k = 0, 1 \cdots N \end{aligned}$$

Certainly, if the contraints formularized in (19) and (20) are satisfied, the rotation contraints (12) and (13) will be satisfied meanwhile. Unfortunately, the rotation constraints based on triads can't afford a closed-form solution as well because the system of triad rotation constraints may have a multidimensional nullspace, which makes the solution underdetermined (Refer to [3]).

0.4 Triad Basis Constraints

Qualitatively speaking, rotation contraints only afford contraints on the projective matrix, rather than the shape basis. Namely, constraints on rotation matrix only deal with the rigid transform (the shape for every single frame is rigid) but can't cope with the nonrigid transform (the shapes over frames are nonrigid but correlated).

Hence in [20], Xiao first introduced basis constraints to construct further non-rigid constraints over frames. He even proved that the basis constraints eliminate the ambiguity lefted by the rotation constraints and produced a closed-form solution to $\mathbf{G}$.

In order to make things clearer to integrate AAM and non-rigid body matrix factorization, basis constraints are restated as in the following.

We could first find the $N+1$ independent shapes from within the F frames. Reorder the measurement matrix to ensure these $N+1$ shapes are put at the very front. If we just look on the first shape as the mean shape, namely $\hat{\mathbf{s}}_0$ and the other N shapes as the shape basis in the form of $\hat{\mathbf{s}}_0+\hat{\mathbf{s}}_i, i = 1, 2 \cdots N$, then all the rest $F-(N+1)$ shapes from the latter frames could just be represented by the former $N+1$ shapes. Thus, it's easy to have the following facts on the weighted vectors $\mathbf{c}^i, i = 1, 2 \cdots N+1$:

$$
\begin{array}{lll}
\mathbf{c}^1 & = a^1(\hat{p}_0^1 \quad \hat{p}_1^1 \cdots \quad \hat{p}_N^1) & = a^1(1 \quad 0 \cdots \quad 0) \\
\mathbf{c}^2 & = a^2(\hat{p}_0^2 \quad \hat{p}_1^2 \cdots \quad \hat{p}_N^2) & = a^2(1 \quad 1 \cdots \quad 0) \\
 & \cdots & \\
\mathbf{c}^{N+1} & = a^{N+1}(\hat{p}_0^{N+1} \quad \hat{p}_1^{N+1} \cdots \quad \hat{p}_N^{N+1}) & = a^{N+1}(1 \quad 0 \cdots \quad 1) \\
\mathbf{c}^{N+2} & = a^{N+2}(\hat{p}_0^{N+2} \quad \hat{p}_1^{N+2} \cdots \quad \hat{p}_N^{N+2}) & = a^{N+2}(1 \quad \hat{p}_1^{N+2} \cdots \quad \hat{p}_N^{N+2}) \\
 & \cdots & \\
\mathbf{c}^F & = a^F(\hat{p}_0^F \quad \hat{p}_1^F \cdots \quad \hat{p}_N^F) & = a^F(1 \quad \hat{p}_1^F \cdots \quad \hat{p}_N^F)
\end{array}
\tag{21}
$$

Refer to (17), for $k = 0$:

$$
\begin{pmatrix}
a^1\mathbf{R}^1 & \mathbf{0} & \cdots & \mathbf{0} \\
a^2\mathbf{R}^2 & \mathbf{0} & \cdots & \mathbf{0} \\
\cdots & \cdots & \cdots & \cdots \\
a^{N+1}\mathbf{R}^{N+1} & \mathbf{0} & \cdots & \mathbf{0} \\
a^{N+2}\mathbf{R}^{N+2} & \mathbf{0} & \cdots & \mathbf{0} \\
\cdots & \cdots & \cdots & \cdots \\
a^F\mathbf{R}^F & \mathbf{0} & \cdots & \mathbf{0}
\end{pmatrix}
= \bar{\mathbf{M}}\mathbf{G}_0
\tag{22}
$$

for $k = 1, 2 \cdots N$:

$$\begin{pmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \cdots & \mathbf{0} & a^{k+1}\mathbf{R}^{k+1} & \mathbf{0} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & a^{N+2}\hat{p}_k^{N+2}\mathbf{R}^{N+2} & \mathbf{0} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \cdots & \mathbf{0} & a^F\hat{p}_k^F\mathbf{R}^F & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix} = \bar{\mathbf{M}}\mathbf{G}_k \qquad (23)$$

Obviously, for $\bar{\mathbf{M}}\mathbf{G}_0$,

$$\bar{\mathbf{M}}\mathbf{G}_0\mathbf{G}_0^T\bar{\mathbf{M}}^T = \begin{pmatrix} \begin{pmatrix} (a^1)^2 & 0 \\ 0 & (a^1)^2 \end{pmatrix} & \cdots & a^1a^F\mathbf{R}^1(\mathbf{R}^F)^T \\ \cdots & \cdots & \cdots \\ a^1a^F\mathbf{R}^F(\mathbf{R}^1)^T & \cdots & \begin{pmatrix} (a^F)^2 & 0 \\ 0 & (a^F)^2 \end{pmatrix} \end{pmatrix} \qquad (24)$$

For $\bar{\mathbf{M}}\mathbf{G}_k, k = 1, 2 \cdots N$

$$\bar{\mathbf{M}}\mathbf{G}_k\mathbf{G}_k^T\bar{\mathbf{M}}^T = \begin{pmatrix} \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \cdots & \begin{pmatrix} (a^{k+1})^2 & 0 \\ 0 & (a^{k+1})^2 \end{pmatrix} & \cdots & \mathbf{0} & \mathbf{b}_{N+2,k+1}^T & \cdots & \mathbf{b}_{F,k+1}^T \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{b}_{N+2,k+1} & \cdots & \mathbf{0} & \mathbf{b}_{N+2,N+2} & \cdots & \mathbf{b}_{F,N+2}^T \\ \cdots & \cdots & \cdots & \cdots & \cdots & & & \\ \mathbf{0} & \cdots & \mathbf{b}_{F,k+1} & \cdots & \mathbf{0} & \mathbf{b}_{F,N+2} & \cdots & \mathbf{b}_{F,F} \end{pmatrix} \qquad (25)$$

where

$$\mathbf{b}_{i,k+1} = a^i\hat{p}_k^i a^{k+1}\mathbf{R}^i(\mathbf{R}^{k+1})^T, k = 1, 2 \cdots N; i = N+2, N+3 \cdots F \quad (26)$$

$$\mathbf{b}_{i,j} = a^i\hat{p}_k^i a^j\hat{p}_k^j\mathbf{R}^i(\mathbf{R}^j)^T, k = 1, 2 \cdots N; i, j = N+2, N+3 \cdots F \quad (27)$$

It won't be hard to have the following basis constraints on $\mathbf{Q}_k; k = 0, 1, 2 \cdots N$, besides the rotation constraints preserved in (19) and (20):

1. For the top left $2(N+1) * 2(N+1)$ sub block of $\bar{\mathbf{M}}\mathbf{Q}_k\bar{\mathbf{M}}^T$:

$$\bar{\mathbf{M}}\mathbf{Q}_k\bar{\mathbf{M}}^T(2k+1, 2k+1) = \bar{\mathbf{M}}\mathbf{Q}_k\bar{\mathbf{M}}^T(2k+2, 2k+2) \quad (28)$$

$$\bar{\mathbf{M}}\mathbf{Q}_k\bar{\mathbf{M}}^T(i,j) = 0 \quad \text{except} \quad (28) \quad (29)$$

$$i, j = 1, 2 \cdots 2(N+1)$$

2. For the top right $2(N+1) * 2(F-(N+1))$ sub block of $\bar{\mathbf{M}}\mathbf{Q}_k\bar{\mathbf{M}}^T$:

$$\begin{aligned} \bar{\mathbf{M}}\mathbf{Q}_k\bar{\mathbf{M}}^T(i,j) &= 0 \quad i! = 2k+1, 2k+2 \\ i &= 1, 2\cdots 2(N+1); j = 2(N+1)+1, 2(N+1)+2\cdots 2F \end{aligned} \tag{30}$$

3. For the bottom left $2(F-(N+1)) * 2(N+1)$ sub block of $\bar{\mathbf{M}}\mathbf{Q}_k\bar{\mathbf{M}}^T$:

$$\begin{aligned} \bar{\mathbf{M}}\mathbf{Q}_k\bar{\mathbf{M}}^T(i,j) &= 0 \quad j! = 2k+1, 2k+2 \\ i &= 2(N+1)+1, 2(N+1)+2\cdots 2F; j = 1, 2\cdots 2(N+1) \end{aligned} \tag{31}$$

   Since $\bar{\mathbf{M}}\mathbf{Q}_k\bar{\mathbf{M}}^T$ is symmetric, the above constriants (31) are actually duplicate of (30).

For the bottom right $2(F-(N+1)) * 2(F-(N+1))$ sub block of $\bar{\mathbf{M}}\mathbf{Q}_k\bar{\mathbf{M}}^T$, triad rotation constraints are still satisfied.

With all the above triad constraints, including both the triad basis constraints (28), (29) and (30) (alternatively (31)), and the triad rotation contraints (19) and (20), a closed-form solution to $\mathbf{Q}_k$ is able to be obtained. Consequently, $\mathbf{G}_k$ can be computed.

But, this so-called closed-form solution has its limitations as analyzed by Brand in [3]. Since this method selects the first $N+1$ independent shapes from the foremost frames as the shape basis, it can't guarantee that the selected shape basis can best represent all shapes in all frames. An assumption must be met in order to ensure its correctness: the data are noiseless so that all samples could be seamlessly represented by a full-rank (rank $N+1$) 3D shape basis. When the data is noisy so that the rank of the 3D shape basis can't be determined, this method begins to break down. Reasonably, a shape basis produced by SVD rather than deduced from the foremost frames could be chosen as the best shape basis (Please refer to (0.6)).

0.5 Solutions to Rigid Body 3D Recovery (RSFM)

Solution to rigid body recovery could simply be looked on as a special case of the general form solution of (10), where $N = 0$. This actually means there is only one shape basis – the mean shape, so that

$$\mathbf{W} = \begin{pmatrix} a^1\hat{p}_0^1\mathbf{R}^1 \\ a^2\hat{p}_0^2\mathbf{R}^2 \\ \cdots \\ a^F\hat{p}_0^F\mathbf{R}^F \end{pmatrix} \hat{\mathbf{s}}_0 = \mathbf{MB} \tag{32}$$

In this case, the classical Tomasi and Kanade (short for TK) algorithm based on SVD matrix decomposition could be adopted to find the rigid 3D shape and the rotation matrices over all frames. From the physical meaning of SVD, among all the possible spanned eigen vectors, the three eigen vectors corresponding to the three biggest eigen values represent the three axes in which the data have three biggest variances. Generally, noises of the data are relatively very small. Therefore, the three biggest eigen vectors must be corresponding to the variances caused by the projection process, rather than the noise. So, the rank-3 SVD facterization is the reasonable solution for rigid body 3D reconstruction as proposed in [13].

0.6 Solutions to Non-Rigid Body 3D Recovery (NRSFM)

Objects like human faces are actually non-rigid and seriously thorny to be resolved.

To the best of the author's knowledge, the first solution to NRSFM from a single-view video sequence was addressed by Bregler [4]. It factorizes the measurement matrix to estimate the shape bases first. Then, a second factorization is carried out to estimate poses over frames before orthonormality constraints are enforced. For this approach, the initial estimate of shape bases is required to be fairly accurate. Otherwise, there will be no remedy to correct the wrong shape bases. Hence, Torresani adopted an iterative optimization method, which can update shape bases, poses, and so-called configurations (The term "configurations" are exactly statistical shape model parameters) in turn [16]. But, both approaches only enforce rotation constraints which are insufficient to solve the corrective transformation matrix by least-square-fit. Similar failure is also found in Brand's early paper [2]. In order to remove this ambiguity, Xiao proposed a closed-form solution to NRSFM, termed as XCK, by imposing both rotation and basis constraints [20]. Yet, XCK selects a shape basis set from the foremost independent image frames, which can't guarantee the selected shape bases are able to best represent all shapes in all frames. Brand indicated this in his recent paper [3] and exposited that XCK require an assumption to ensure its correctness: the data are noiseless so that all shapes can be seamlessly represented by a full-rank 3D shape basis. He also presented a direct numerical method that aims to minimize the deviation from the required orthogonal structure of projective matrix and is not limited to XCK assumption. All the methods mentioned above share in common that an arbitrary shape is regarded as a direct linear combination of some non-rigid shape bases, without an explicit rigid component.

In fact, statistical shape models have been well studied for over ten years and describe a shape as the linear combination of both the rigid constant and non-rigid component. Active appearance model (AAM) [5] and morphable model (MM) [1] are such two influential statistical shape models used to represent 2D and 3D deformable objects respectively. Therefore, following the expression of AAM and 3DMM, Xiao and Torresani restate their shape models for 3D recovery in [19] and [15].

However, none of the above solutions paid attention to the process of alignment, namely, global shape normalization. In fact, global shape normalization in NRSFM can be taken advantage of to estimate object poses only from the object rigid component. An iterative algorithm is then proposed to recover both the rigid constant, which is exactly the primary shape for all shapes to be aligned to, and the non-rigid component, which is looked on as the main cause for shape residuals between the projective shapes of the object rigid component and the measurement shapes over frames. We have proposed our novel algorithm, borrowing the ideas from both Procrustes Analysis and n-dimentional object alignment.

Bibliography

[1] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 187 – 194, 1999.

[2] M. Brand. Morphable 3d model from video. Technical Report TR-2001-37, Mitsubishi Electric Research Laboratories, December 2001.

[3] M. Brand. A direct method for 3d factorization of nonrigid motion observed in 2d. Technical Report TR2005-052, Mitsubishi Electric Research Laboratories, December 2005.

[4] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[5] T. Cootes and C. Taylor. Statistical models of appearance for computer vision. Technical report, Imaging Science and Biomedical Engineering, University of Manchester, March 8 2004.

[6] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects,. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[7] M. Han and T. Kanade. Reconstruction of a scene with multiple linearly moving objects. In *International Conference on Computer Vision and Pattern Recognition (CVPR00)*, 2000.

[8] D.L. Jiang, Y.X. Hu, S.C. Yan, and H.J. Zhang. Efficient 3d reconstruction for face recognition. *Journal of Pattern Recognition, Special Issue on Image Understanding for Digital Photographs*, 2005.

[9] J. Lee, R. Machiraju, H. Pfister, and B. Moghaddam. Estimation of 3d faces and illumination from single photographs using a bilinear illumination model. Technical Report TR2005-045, MITSUBISHI ELECTRIC RESEARCH LABORATORIES, June 2005.

[10] Z. Lei, Q. Bai, R. He, and S.Z. Li. Face shape recovery from a single image using cca mapping between tensor spaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, 2008.

[11] C. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *Pattern Recognition and Artificial Intelligence*, 19(3):206–218, 1997.

[12] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1/2/3):7–42, April-June 2002.

[13] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[14] L. Torresani, A. Hertzmann, and C. Bregler. Learning non-rigid 3d shape from 2d motion. In *Proceedings of NIPS 2003*, 2003.

[15] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Transactions on Pattern Recognition and Artificial Intelligence*, 30:878–892, 2008.

[16] L. Torresani, D. Yang, G. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[17] B. Triggs. Factorization methods for projective structure and motion. In *International Conference on Computer Vision and Pattern Recognition (CVPR'96)*, 1996.

[18] Wikipedia. Kronecker product — wikipedia, the free encyclopedia, 2008. [Online; accessed 22-January-2009].

[19] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 535 – 542, June 2004.

[20] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. Technical Report CMU-RI-TR-03-16, Robotics Institute, Carnegie Mellon University, June 2003.