# Tracking and Modeling Non-Rigid Objects with Rank Constraints

Lorenzo Torresani[†] Danny B. Yang[†] Eugene J. Alexander[‡] Christoph Bregler[†]
[†]{ltorresa, dbyang, bregler}@cs.stanford.edu
Computer Science Department
Stanford University, Stanford, CA 94305
[‡]Gene.Alexander@stanford.edu
Mechanical Engineering Department
Stanford University, Stanford, CA 94305

## Abstract

*This paper presents a novel solution for flow-based tracking and 3D reconstruction of deforming objects in monocular image sequences. A non-rigid 3D object undergoing rotation and deformation can be effectively approximated using a linear combination of 3D basis shapes. This puts a bound on the rank of the tracking matrix. The rank constraint is used to achieve robust and precise low-level optical flow estimation without prior knowledge of the 3D shape of the object. The bound on the rank is also exploited to handle occlusion at the tracking level leading to the possibility of recovering the complete trajectories of occluded/disoccluded points. Following the same low-rank principle, the resulting flow matrix can be factored to get the 3D pose, configuration coefficients, and 3D basis shapes. The flow matrix is factored in an iterative manner, looping between solving for pose, configuration, and basis shapes. The flow-based tracking is applied to several video sequences and provides the input to the 3D non-rigid reconstruction task. Additional results on synthetic data and comparisons to ground truth complete the experiments.*

## 1. Introduction

This paper addresses the problem of 3D tracking and model acquisition of non-rigid motion in video sequences. We are specifically concerned with human motion, which is a challenging domain. Standard low-level tracking schemes usually fail due to local ambiguities and noise. Most recent approaches overcome this problem with the use of a model. In those techniques optical flow vectors or the motion of feature locations can be constrained by a low degree-of-freedom parametric model. For instance, to track joint-angles of human limb segments an approximate kinematic chain model can be used. The models lose many details that cannot be recovered by simple cylinder or sphere shape models and fixed axis rotations. Non-rigid torso motions, deforming shoe motions, or subtle facial skin motions are problem areas. Alternatively, such non-rigid motions can be captured with basis-shape models that are learned from example data. Most of the previous work is based on PCA techniques applied to 2D or 3D training data. For example, human face deformations have been tracked in 2D and 3D with such models. For 3D domains, prior models are aquired using stereo cameras or cyber-scan hardware. Carefully labeled data have to be provided to derive the PCA based models.

We are interested in cases where no such 3D models are available, or existing models are too restricted and would not be able to recover all subtleties. The input to our technique is a single-view video recording of an arbitrary deforming object, and the output is the 3D motion AND a 3D shape model parameterized by its modes of non-rigid deformation.

We are facing three very challenging problems:

1. Without a model, how can we reliably track ambiguous and noisy local features in this domain?
2. Without point feature tracks or robust optical flow, how can we derive a model?
3. Given reliable 2D tracks, how can we recover 3D nonrigid motion and shape structure?

We have previously demonstrated that single-view 2D point tracks are enough to recover 3D non-rigid motion and structure by exploiting low-rank constraints [7]. Based on the same assumption, we show in this paper that it is also possible to constrain the low-level flow estimation and to handle occlusion without any model-assumption. Irani [14] has demonstrated that model-free low-rank constraints can be applied to overcome local ambiguities in flow-estimation for rigid scenes. We show that this can be extended to 3D non-rigid tracking and model-acquistion. Our new techniques do not need 2D point tracks, can deal with ambigous and noisy local features, and can handle occlusion. By exploiting the low-rank constraints in low-level tracking and in 3D non-rigid model acquisition we are able to solve all three challenges mentioned above in one unified manner.

We demostrate the technique on tracking several video sequences and on deriving 3D deformable models from those measurements.

## 2 Previous Work

Many non-rigid tracking solutions have been proposed previously. As mentioned earlier, most techniques use an a-priori model. Examples are [16, 5, 9, 19, 3, 4]. Most of these techniques model 2D non-rigid motion, but some of these approaches also recover 3D pose and deformations based on a 3D model. The 3D model is obtained from 3D scanning devices [6], stereo cameras [10], or multi-view reconstruction [18, 11]. The multi-view reconstruction is based on the assumption that for a specific deformed configuration all views are sampled at the same time. This is equivalent to the structure from motion problem, that assumes rigidity between the different views [22]. Extensions have been proposed, such as the multi-body factorization method of Coseira and Kanade [8] that relaxes the rigidity constraint. In this method, $K$ independently moving objects are allowed, which results in a tracking matrix of rank $3K$ and a permutation algorithm that identifies the submatrix corresponding to each object. More recently, Bascle and Blake [1] proposed a method for factoring facial expressions and pose during tracking. Although it exploits the bilinearity of 3D pose and nonrigid object configuration, it requires again a set of basis images selected before factorization is performed. The discovery of these basis images is not part of their algorithm.

In addition, most techniques treat low-level tracking and 3D structural constraints independently. In the following section we describe how we can track and reconstruct non-rigid motions from single views without prior models.

## 3 Technical Approach

The central theme in this paper is the exploitation of rank-bounds for recovering 3D non-rigid motion. We first describe in general why and under what circumstances 3D non-rigid motion puts rank bounds on 2D image motion (section 3.1). We then detail how these bounds can be used to constrain low-level tracking in a model-free fashion (section 3.2). We then describe how this technique can also be used for prediction of occluded features (section 3.3), and we then introduce three techniques that are able to reconstruct 3D deformable shapes and their motion from those 2D measurements (section 3.4.1, 3.4.2, and 3.4.3).

### 3.1 Low-rank constraints for non-rigid motion

Given a sequence of $F$ video frames, the optical flow of $P$ pixels can be coded into two $F \times P$ matrices, $\mathbf{U}$ and $\mathbf{V}$. Each row of $\mathbf{U}$ holds all x-displacements of all $P$ locations for a specific time frame, and each row of $\mathbf{V}$ holds all y-displacements for a specific time frame. It has been shown that if $\mathbf{U}$ and $\mathbf{V}$ describe a 3D rigid motion, the rank of $[\frac{\mathbf{U}}{\mathbf{V}}]$ has an upper bound, which depends on the assumed camera model (for example, for an orthographic camera model the rank is $r \leq 4$, while for a perspective camera model the rank is $r \leq 8$) [22, 14]. This rank constraint derives from the fact that $[\frac{\mathbf{U}}{\mathbf{V}}]$ can be factored into two matrices: $Q \times S$. $Q^{2F \times r}$ describes the relative pose between camera and object for each time frame, and $S^{r \times P}$ describes the 3D structure of the scene which is invariant to camera and object motion.

Previously we have shown that non-rigid object motion can also be factored into 2 matrices [7] but of rank $r$ that is higher than the bounds for the rigid case. Assuming the 3D non-rigid motion can be approximated by a set of $K$ modes of variation, the 3D shape of a specific object configuration can be expressed as a linear combination of $K$ basis-shapes $(S_1, S_2, \dots S_k)$. Each basis-shape $S_i$ is a $3 \times P$ matrix describing $P$ points. The shape of a specific configuration is a linear combination of this basis set:

$$S = \sum_{i=1}^{K} l_i \cdot S_i \qquad S, S_i \in \mathbb{R}^{3 \times P}, l_i \in \mathbb{R} \tag{1}$$

Assuming weak-perspective projection, at a specific time frame $t$ the $P$ points of a configuration $S$ are projected onto $2D$ image points $(u_{t,i}, v_{t,i})$:

$$\begin{bmatrix} u_{t,1} & \dots & u_{t,P} \\ v_{t,1} & \dots & v_{t,P} \end{bmatrix} = R_t \cdot \left( \sum_{i=1}^{K} l_{t,i} \cdot S_i \right) + T_t \tag{2}$$

$$R_t = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \end{bmatrix} \tag{3}$$

where $R_t$ contains the first two rows of the full $3D$ camera rotation matrix, and $T_t$ is the camera translation. The weak perspective scaling $(f/Z_{avg})$ of the projection is implicitly coded in $l_{t,1}, \dots l_{t,K}$. As in [22], we can eliminate $T_t$ by subtracting the mean of all $2D$ points, and henceforth can assume that $S$ is centered at the origin.

Weak perspective projection is in practice a good approximation if the perspective effects between the closest and furthest point on the object surface are small. Extending this framework to full-perspective projection is straight-forward using an iterative extension. All experiments reported here assume weak perspective projection.

We can rewrite the linear combination in (2) as a matrix multiplication:

$$\begin{bmatrix} u_{t,1} & \dots & u_{t,P} \\ v_{t,1} & \dots & v_{t,P} \end{bmatrix} = \begin{bmatrix} l_{t,1}R_t & \dots & l_{t,K}R_t \end{bmatrix} \cdot \begin{bmatrix} S_1 \\ S_2 \\ \dots \\ S_K \end{bmatrix} \tag{4}$$

We stack all point tracks from time frame 1 to $F$ into one large measurement $2F \times P$ matrix W. Using (4) we can write:

$$W = \underbrace{\begin{bmatrix} l_{1,1}R_1 & \dots & l_{1,K}R_1 \\ l_{2,1}R_2 & \dots & l_{2,K}R_2 \\ & \dots & \\ l_{F,1}R_F & \dots & l_{F,K}R_F \end{bmatrix}}_{Q} \cdot \underbrace{\begin{bmatrix} S_1 \\ S_2 \\ \dots \\ S_K \end{bmatrix}}_{B} \tag{5}$$

Since $Q$ is a $2F \times 3K$ matrix and $B$ is a $3K \times P$ matrix, in the noise free case $W$ has a rank $r \leq 3K$.

In the following sections we describe how this rank bound on $W$ can be exploited for 1) constrained low-level tracking 2) recovery of occluded feature locations 3) 3D reconstruction of pose, non-rigid deformations, and key-shapes.

## 3.2 Basis Flow

The previous analysis tells us why $W$ is rank bounded and how $W$ can be factored. In this section we discuss how to derive the optical flow matrix $W$ from an image sequence and how the rank-bound can be used to disambiguate the local flow.

Features can usually be tracked reliably with local methods, such as Lucas-Kanade [17] and extensions [21, 2], if they contain a distinctive high contrast pattern with 2D texture, such as corner features. For traditional rigid shape reconstruction, only a few feature locations are necessary. Non-rigid objects go through much more severe motion variations, hence many more features need to be tracked. In the extreme case it might be desirable to track every pixel location. Unfortunately, many objects that we are interested in, including the human body, do not have many of those very reliable features.

Our solution to the tracking dilemma builds on a technique introduced in [14] that exploits rank constraints for optical flow estimation in the case of rigid motion.

Since $W$ is assumed to have rank $r$, all $P$ columns of $W$ can be modeled as a linear combination of $r$ "basis-tracks", $Q$. The basis is not uniquely defined, but if there are more than $r$ points whose trajectories over the $F$ frames can be reliably estimated, then we can compute with SVD the first $r$ eigenvectors $\hat{Q}$ of the reduced tracking matrix $W_{reliable}$. $\hat{Q}^{2F\times r}$ is an initial estimate of the basis for all $P$ tracks. Our next task is to estimate all $P$ tracks (the entire $W$) using this eigenbase $\hat{Q}$ and additional local image constraints.

As in the original Lucas-Kanade tracking, we assume that a small image-patch centered at a track-point location will not change its appearance drastically between two consecutive frames. Therefore the local patch flow $[u, v]$ can be computed by solving the following well known equation [17]:

$$[u_{t,p} v_{t,p}] \cdot \begin{bmatrix} c & d \\ d & e \end{bmatrix} = [g, h] \tag{6}$$

where $\begin{bmatrix} c & d \\ d & e \end{bmatrix} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$ is the second moment matrix of the local image patch in the first frame, $g = \sum I_x I_t$, and $h = \sum I_y I_t$. (for further details see [17, 21, 2]).

If all $F \times P$ flow-vectors across the entire image sequence are coded relative to one single image template, the following equation system can be written [14]:

$$[U|V] \cdot \begin{bmatrix} C & D \\ D & E \end{bmatrix} = [G|H] \tag{7}$$

where $C$, $D$, $E$ are diagonal $P \times P$ matrices that contain the corresponding $c$, $d$, and $e$ values for each of the $P$ local image patches. Accordingly, $G$ and $H$ are $F \times P$ matrices, that contain the $g$ and $h$ values for all $P$ local patches across all $F$ time frames. This system of equations is a rewriting of the Lucas-Kanade linearization for every flow vector, with no additional constraints yet applied. The number of free variables is equal to the number of constraints. If a local patch has no 2D texture, the single equation describing its motion in the system will only provide an accurate estimate of its normal flow (aperture problem).

Now we split $\hat{Q}$ into $\hat{Q}_U$ that contains all even rows of $\hat{Q}$, and $\hat{Q}_V$ that contains all odd rows of $\hat{Q}$. Since $Q$ is a basis for $W$, there must exist some $r \times P$ matrix $\hat{B}$ for which the following equations hold:

$$\hat{Q}_u \cdot \hat{B} = U \qquad \hat{Q}_v \cdot \hat{B} = V \tag{8}$$

Using (7) we can write [14]:

$$[\hat{Q}_u \cdot \hat{B} | \hat{Q}_v \cdot \hat{B}] \cdot \begin{bmatrix} C & D \\ D & E \end{bmatrix} = [G|H] \tag{9}$$

This is a system with $r \times P$ unknowns (the entries in $\hat{B}$) and $2F \times P$ equations. For long tracks ($F >> P$) the system is very over-constrained (in contrast to (7)). We can exploit this redundancy to derive the optical flow for points difficult to track and for features along 1D edges.

Since $[G|H]$ is computed based on the Lucas-Kanade linearization, the resulting flow $[U|V] = [\hat{Q}_u \cdot \hat{B} | \hat{Q}_v \cdot \hat{B}]$ will only be a first approximation. We rewarp all images of the sequence using the new flow and then iterate equation (9).

## 3.3 Dealing with Occlusion

By reordering the elements of $\hat{B}$ into a $r \cdot P$-dimensional vector $\hat{b}$, equation (9) can be rewritten in the form:

$$L^{2PN\times rP} \cdot \hat{b}^{rP\times 1} = m^{2PN\times 1} \tag{10}$$

where now each row describes one point in one particular frame. If we have occlusion, or the tracker used for initialization has lost some points at certain time frames, then the corresponding entries in the $m$ vector will not be measurable. We eliminate those rows from the $L$ matrix and the $m$ vector. If the number of missing points is not overly large, we are still left with an overconstrained system that can give us an accurate solution for $\hat{b}$. As long as the disappearing features are visible in enough frames, the product $\hat{Q} \cdot \hat{B}$ provides also a good prediction of the displacements for the missing points.

## 3.4 3D Reconstruction

As mentioned earlier, the factorization of $W$ into $Q$ and $B$ is not unique. Any invertible $r \times r$ matrix $A$ applied to $Q$ and $B$ in the following way leads to an alternative factorization:

$$Q_a = Q \cdot A, \qquad B_a = A^{-1} \cdot B \tag{11}$$

$Q_a$ and $B_a$ multiplied together approximate $W$ with the same sum-of-squared error as $Q$ and $B$.

Using SVD, we compute a $\hat{Q}$ (with orthonormal columns) and $\hat{B}$. In general $\hat{Q}$ will not comply to the structure we described in (5):

$$Q = \begin{bmatrix} Q_1 \\ Q_2 \\ \dots \\ Q_F \end{bmatrix} \quad \text{with} \quad Q_t = [l_{t,1}R_t | \dots l_{t,k}R_t] \tag{12}$$

For the general case, transforming $\hat{Q}$ into a $Q$ that complies to those constraints can not be done with a linear least-squares technique. For the specific case of rigid scenes, each sub-block is equal to the first 2 rows of a rotation matrix ($Q_t = R_t$). Tomasi-Kanade [22] suggested a linear approximation schema to find an $A$ that enforces the sub-blocks of $Q$ to comply to rotation matrices.

### 3.4.1 Sub-block factorization

For the non-rigid case, we previously proposed a second factorization step on each sub-block that transforms every $\hat{Q}_t$ onto a $Q_t$ that complies to the constraints (5) [7]. $\hat{Q}_t$ can be rewritten as:

$$\begin{aligned} Q_t &= \begin{bmatrix} l_1R_t & \dots & l_KR_t \end{bmatrix} \\ &= \begin{bmatrix} l_1r_1 & l_1r_2 & l_1r_3 & \dots & l_Kr_1 & l_Kr_2 & l_Kr_3 \\ l_1r_4 & l_1r_5 & l_1r_6 & \dots & l_Kr_4 & l_Kr_5 & l_Kr_6 \end{bmatrix} \end{aligned}$$

We reorder the elements of $Q_t$ into a new matrix $\bar{Q}_t$:

$$\begin{aligned} \bar{Q}_t &= \begin{bmatrix} l_1r_1 & l_1r_2 & l_1r_3 & l_1r_4 & l_1r_5 & l_1r_6 \\ l_2r_1 & l_2r_2 & l_2r_3 & l_2r_4 & l_2r_5 & l_2r_6 \\ & & & \dots & & \\ l_Kr_1 & l_Kr_2 & l_Kr_3 & l_Kr_4 & l_Kr_5 & l_Kr_6 \end{bmatrix} \\ &= \begin{bmatrix} l_1 \\ l_2 \\ \dots \\ l_K \end{bmatrix} \cdot \begin{bmatrix} r_1 & r_2 & r_3 & r_4 & r_5 & r_6 \end{bmatrix} \end{aligned}$$

which shows that $\bar{Q}_t$ is of rank 1 and can be factored into the pose $R_t$ and configuration weights $l_i$ by SVD.

After the second factorization step is applied to each of the individual sub-blocks $\hat{Q}_t$, a non-linear optimization over the entire time sequence is performed to find one invertible matrix $A$ that orthonormalizes all of the sub-blocks. The result is that each sub-block is a scaled rotation matrix. In the presence of noise and ambiguities, the second and higher eigen-values of many sub-blocks do not vanish. In those cases, it results in bad rank-1 approximations, and bad estimates for $R_t$. We therefore propose a second alternative in the next section that overcomes this limitation.

### 3.4.2 Iterative Optimization

Instead of local factorizations on the sub-blocks, we propose a new iterative technique that solves (5) directly.

Many non-rigid objects have a dominant rigid component and we take advantage of this to get an initial estimate for all pose matrices ($R_1$, ... , $R_F$). Given an initial guess of the pose at each time frame, we can solve for the configuration weights and the basis shapes.

To initialize the pose, we factor $W$ into a $2F \times 3$ rigid pose matrix $\hat{Q}_{rig}$ and a $3 \times P$ matrix $\hat{B}_{rig}$ (as originally done by Tomasi-Kanade). As usual, we transform $\hat{Q}_{rig}$ into a matrix $Q_{rig}$, whose sub-blocks have all weak-perspective rotation matrices (as outlined in section 3.4.1).

Using $Q_{rig}$ as an initial guess for the pose of the non-rigid shape, we solve for the non-rigid $l_{t,i}$ and $B$ terms in (5). We do this iteratively by first initializing $l_{t,i}$ randomly and then iterating between solving for $B$, then for $l_{t,i}$, and then refining $R_t$ again[1].

1. Given all $R_t$ and $l_{t,i}$ terms (the $Q$ matrix), equation (5) can be used to find the linear least-square-fit of $B$.

2. Given $B$ and all $R_t$, we can solve for all $l_{t,i}$ with linear least-squares.

3. Given $B$ and $L$, we can rewrite (5) to:

$$W_t = R_t \sum_k l_{t,k} S_k \tag{13}$$

Solving for all $R_t$ such that they fit this equation and remain rotation matrices can be done by parameterizing $R_t$ with exponential coordinates. A full rotation

[1] Alternatively we can use the sub-block factorization described in section 3.4.1 for initialization

matrix can be described by 3 variables $[\omega_x, \omega_y, \omega_z]$ as:

$$R(\omega) = \exp \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{14}$$

Assume $\bar{\omega}$ is the estimate of $R_t$ at the previous iteration, we can then linearize (13) around the previous estimate to:

$$W_t = \begin{bmatrix} 1 & -\omega_z' & \omega_y' \\ \omega_z' & 1 & -\omega_x' \end{bmatrix} R(\bar{\omega}) \sum_k l_{t,k} S_k \tag{15}$$

and solve for a new $\omega$. We then update $R(\omega) := R(\omega')R(\bar{\omega})$ and iterate[2]

We iterate all 3 steps until convergence.

Similar to the technique described in section 3.3 we can easily handle missing entries in $W$ when points are occluded or are lost by the tracker. $B$ and $L$ are over-constrained, so we leave out the missing data points and solve the linear fit as before.

### 3.4.3 Multi-View Input

Another extension of this factorization technique is the incorporation of multi-view inputs from $M$ cameras.

This enlarges the input matrix $W$ to size $2FM \times P$.

$$W = \begin{bmatrix} W_1 \\ W_2 \\ \ldots \\ W_F \end{bmatrix}, W_t = \begin{bmatrix} W_{t,1} \\ W_{t,2} \\ \ldots \\ W_{t,M} \end{bmatrix}, W_{t,c} = \begin{bmatrix} u_{c,1} \ldots u_{c,P} \\ v_{c,1} \ldots v_{c,P} \end{bmatrix} \tag{16}$$

As before, we assume that $W_{t,c}$ can be described by a $2 \times 3$ pose matrix $R_{t,c}$, by $k$ deformation coefficients $l_{t,1}, l_{t,2}, \ldots l_{t,k}$, and a $3K \times P$ key-shape matrix $B$. Assuming the cameras are synchronized, an additional constraint for the multi-view case is that all $M$ views share the same deformation coefficients for a particular time frame $t$:

$$W_t = [l_{t,1} \cdot R_t | l_{t_2} \cdot R_t | \ldots | l_{t,K} \cdot R_t] \cdot B \tag{17}$$

$$R_t = \begin{bmatrix} R_{t,1} \\ R_{t,2} \\ R_{t,M} \end{bmatrix} \tag{18}$$

Similar to our previous 2-step factorization, we can factor $W$ into $Q$ and $B$ complying to this new structure. Furthermore we can enforce another constraint if we assume that all $M$ cameras remain fixed relative to each other: The relative rotation between all $R_{t,c}$'s in the $R_t$ sub-block of Q is constant over time. This is enforced with a nonlinear iterative optimization after the 2-step factorization.

[2]A future extension of this algorithm will deal with an iterative version for true perspective models. However, we like to point out, that for the orthographic case, there exist also several closed-form solutions including Horn's technique [12, 13], and a SVD based method proposed by Ruderman [20] that we will include in an extendet technical report.

### 3.4.4 Shape Regularization

If there is not enough out-of-plane rotation, the $Z$ values of $B$ can be ill-conditioned. For instance, a small non-rigid deformation in $X$ and $Y$ can also be explained by a small out-of-image-plane rigid rotation of a shape with large $Z$ values. Another problem area is that if the low-level features have almost no image texture, the corresponding 3D point in the shape matrix $B$ is not defined.

We can overcome these problems by regularizing the shape matrix $B$ during the iterative optimization. A simple term can be added to the least-squares-fit of $B$ in section 3.4.2, but also to the least-squares-fit of $\hat{B}$ in section 3.2. If point location $i$ and $j$ are neighbors (we can determine that with a local nearest neighbor algorithm or Delaunay triangulation), we add the following term for each neighbor pair $\alpha_{ij}^2 \sum_r (b_{r,i} - b_{r,j})^2$. This pulls ambiguous points closer to the average of their neighbors. It is similar to the smoothness terms in snake-based contour tracking optimizations [15]. $\alpha_{ij}$ can be inversely proportional to the distance between the 2 points $i$ and $j$. The global least-square-fit remains linear, since we only need to add additional linear equations of the form $\alpha_{ij} b_{r,i} - \alpha_{ij} b_{r,j} = 0$ to the least square system in section 3.2 and 3.4.2.

## 4 Experimental Results

### 4.1 Rank-Constrained Tracking

We tested the rank-constraint technique for optical flow estimation on a 500 frame long video sequence of a deforming shoe (Figure 1). The recordings are challenging due to changes in the object appearance that are caused by the large rotations and deformations as well as by variations in illumination.

In our examples a set of 30 reliable features were initially tracked using an implementation of the technique of Lucas-Kanade employing affine transformations for the patches centered at these points. We updated the reference patch of each point every 10 frames in order to accommodate the changes in feature appearance of our long sequence. We assumed our multi-frame approach capable of recovering the possible drifting introduced on some of the tracks by the frequent update of the template for the points. 80 additional features were then selected along 1D edges in the reference frame. We produced a first approximate initialization of their displacements by linear extrapolation from the motion of the reliable points. We used the resulting $W$ matrix as initialization for our tracking technique based on rank constraints. We experimented with different values for the rank and achieved the best solution by setting rank $r = 9$. We employed the classic pyramidal approach in smoothing the images and ran several iterations of the multi-frame method.

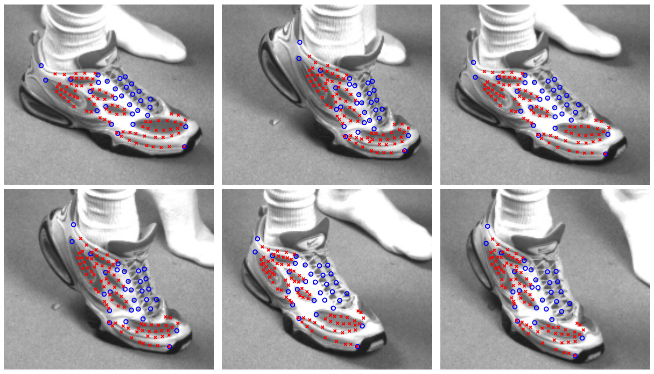

**Figure 1. Example tracks of the shoe sequence. The blue circles are reliable points, the red crosses are features with 1D texture that have been recovered using rank constraints.**

We could track robustly and very accurately most of the 110 points throughout the whole sequence. In order to correct drifting of some of the edge features we incorporated the additional regularization described in section 3.4.4. Figure 1 shows the features tracked for several frames.

In the last part of the image sequence many of the distinctive points that we have used to derive the optical flow field are progressively occluded while some disappear and then become visible again with the variations in the motion. These difficult frames were used to demonstrate the ability of our solution to cope with lost features and occlusion. For this experiment we manually labeled in each frame features that were not visible, incorrectly tracked or lost by the Lucas-Kanade initialization and then used the approach described in section 3.3 to recover their position. Figure 2 shows the estimated position of some of these features before, during and after their temporary occlusion. The algorithm is successful in reconstructing their complete trajectories. Future experiments will also use an automatic track termination test.

## 4.2 3D Reconstruction

Given the estimated $\hat{Q}$ and $\hat{B}$ of those 500 tracked monocular image frames, we then applied our reconstruction technique described in section 3.4.2. Figure 3 shows some example frames and the reconstructed non-rigid 3D shapes overlayed. See http://movement.stanford.edu/nonrig for mpeg or quicktime video showing the entire sequence reconstructed.

We also applied the new reconstruction technique to a video recording of a deforming human torso (Figure 4). The included video **841_tyab.mpg** (also at

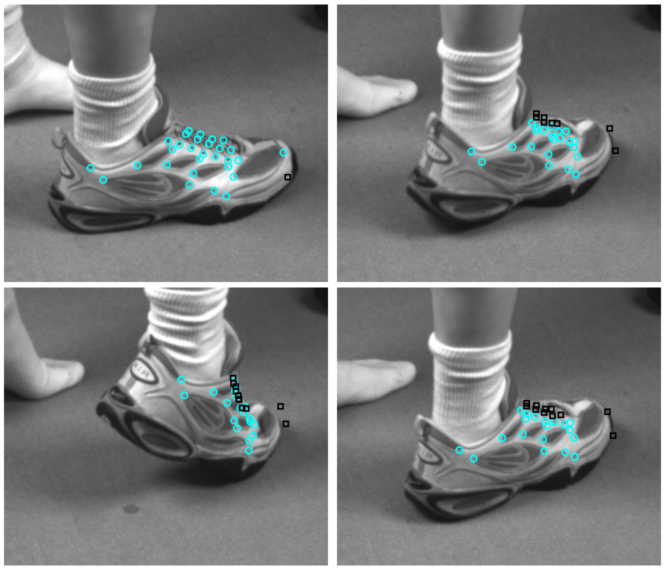

**Figure 2. The black rectangular markers are predicted locations of disappearing features. The algorithm can recover the complete trajectories of the temporarily occluded points.**

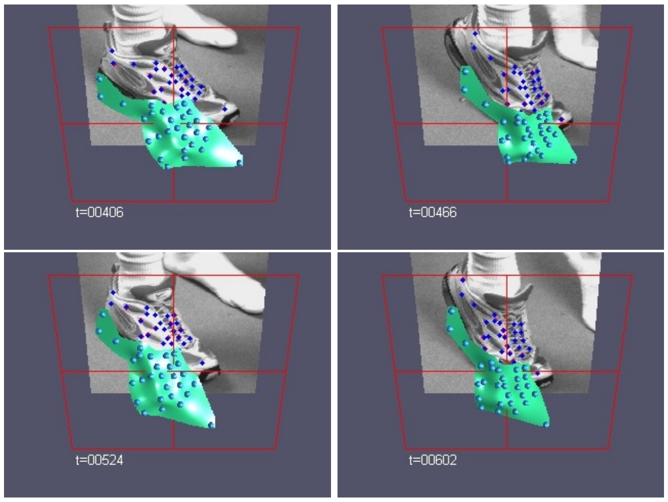

**Figure 3. 3D reconstruction of corresponding 2D tracks from monocular video sequence. Please check video to see all details.**

http://movement.stanford.edu/nonrig) shows the deforming shapes in 3D. As you can tell, again the pose changes and the shape deformations were recovered successfully.

We applied this technique and the previously reported solution [7] to several other video recordings, including a giraffe recording and human face tracking. All those reconstructions recovered the pose and shape deformations

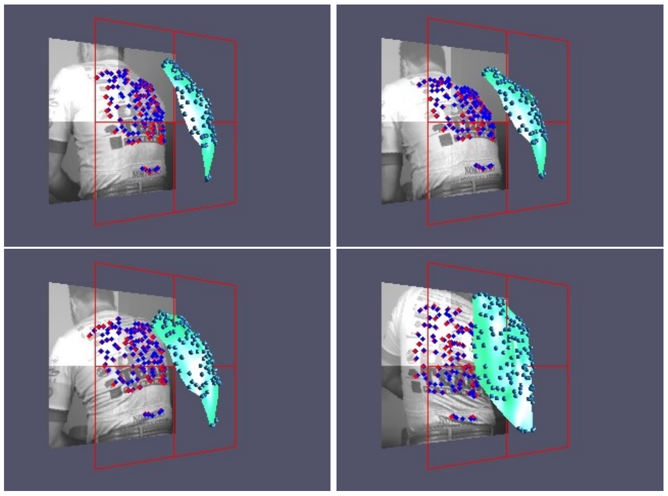

**Figure 4. 3D reconstruction of corresponding 2D tracks from monocular video sequence. Please check video to see all details.**

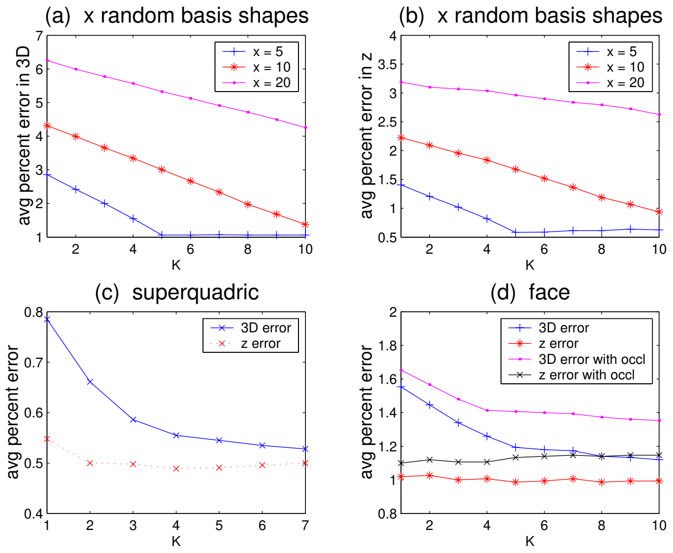

**Figure 5. Iterative optimization on monocular artificial data. Plots show how 3D and z errors vary as the number of basis shapes ($K$) used in the iterative optimization is increased. (a,b) random basis shapes, (a) shows 3D error and (b) shows z error. $x$ is the number of basis shapes used to generate the data. (c) superquadric. (d) deforming face.**

nicely. Since we do not have ground-truth data available, we can only speculate how good the quantitative performance is. Therefore we also tested this technique on several artificial datasets with ground truth and multi-view recordings with 2 calibrated cameras that allowed us to get a good estimate of the ground truth by using triangulation.

## 4.3 Performance on Artificial Data

The artificial data sets were based on random basis shapes, artificial superquadrics, 3D data from 2-view recordings of a deforming face, and 2-view recordings of the shoe sequence. All error reported here are computed in percentage points: the average distance of the reconstructed point to the correct point divided by the size of the shape.

The random basis shapes were generated by sampling points uniformly inside a unit cube. The first basis shape was given the largest weight so that the overall shape has a strong rigid component. Artificial data were generated from 5, 10, and 20 random basis shapes rotating over 300 frames. The overall maximum rotation in any axis was 90 degrees and gaussian noise was added to the final tracking matrix $W$. The results of running the iterative optimization are shown in figure 5(a,b). Both 3D error and z error decrease as the number of basis shapes ($K$) used in the optimization increase. For the data generated from 5 basis shapes, the 3D error and z error level off after $K = 5$. This makes sense because only 5 basis shapes are required to describe the data and the iterative optimization finds the 5 basis shapes.

The superquadric data were generated with three of the octants deforming independently. The same rotation from the random basis shapes was applied to it to generate 300 frames for the tracking matrix $W$. The average 3D and z errors are plotted in figure 5(c).

The 3D face data were taken from a stereo reconstruction of a deforming face. The same rotation was applied to this to generate the 300 frames for the tracking matrix $W$. The average 3D and z errors are plotted in figure 5(d). The iterative optimization was also tested with occlusion. Those points on the face which should be occluded in a certain pose were labelled as such and not included in $W$. As a result, 15% of the points in $W$ were removed. The average 3D and z errors for the reconstruction with occlusion are also plotted in figure 5(d).

## 4.4 Performance on Real Data

Table 1 shows the monocular view reconstruction errors for the shoe sequence. A second camera was used for triangulation to get the ground truth so that we could compare our monocular reconstruction. This reconstruction is the most challenging task, since it tests the entire system from video input to 3D output. In the artificial experiments we ran the algorithms for each $K$ 30 times and reported the median error, for the shoe recording we only ran it once for each $K$. This explains the larger random fluctuations in the

| $K$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 3D error | 3.25 | 2.19 | 3.80 | 2.84 | 2.24 | 2.62 | 2.62 |
| z error | 2.33 | 1.69 | 1.91 | 2.49 | 2.02 | 2.34 | 2.34 |

| $K$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 3D error | 3.28 | 2.95 | 2.65 | 2.46 | 2.26 |

**Table 1. Top: 3D reconstruction performance on monocular shoe sequence. Bottom: 3D reconstruction performance on 2-view face sequence.**

errors. Overall the small reconstruction errors tell us that this technique is indeed able to accurately recover nonrigid deformations from monocular image sequences.

We also ran the multi-view reconstruction on the face data, and achieved again reasonable error rates as can be seen in Table 1.

## 5 Discussion

We have shown how to exploit low-rank constraints for low-level tracking, for prediction of missing low-level features and for non-rigid reconstruction. We have demonstrated those techniques on several video sequences and have shown that good 3D non-rigid reconstructions can be achieved. We further quantified the performance of single-view reconstructions with the use of additional calibrated views and simulations of artificial data.

We have not yet addressed the issue of discovering how many basis shapes are needed. One possible solution is that the user defines an upper treshold on how much reprojection error is allowed. $K$ is increased until the error is below threshold.

Another interesting aspect that is currently under investigation is the bias of this technique. In many cases 3D rotation can be compensated with some degrees of freedom of the basis shape set. Despite this ambiguity, our technique has a strong bias towards representing as much as possible with the rotation matrix, but we like to further study this ambiguities.

Reconstructing non-rigid models from single-view video recordings has many potential applications. For example, we intend to apply this technique to our image-based facial and full-body animation system and to a model based tracking system.

## Acknowledgements

We would like to thank James Davis, Erika Chuang, and Ajit Chaudhari for providing advice and data. This research was funded in part by the National Science Foundation and the Stanford BIO-X program.

## References

[1] B. Bascle and A. Blake. Separability of pose and expression in facial tracking and animation. In *ICCV*, 1998.

[2] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV*, 1992.

[3] M. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *ICCV*, 1995.

[4] M. Black, Y.Yacoob, A.D.Jepson, and D.J.Fleet. Learning parameterized models of image motion. In *CVPR*, 1997.

[5] A. Blake, M. Isard, and D. Reynard. Learning to track the visual motion of contours. In *J. Artificial Intelligence*, 1995.

[6] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, 1999.

[7] C. Bregler, A. Hertzmann, and H. Biermann. Recovering Non-Rigid 3D Shape from Image Streams. In *CVPR*, 2000.

[8] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. *Int. J. of Computer Vision*, pages 159–180, Sep 1998.

[9] D. DeCarlo and D. Metaxas. Deformable model-based shape and motion analysis from images using motion residual error. In *ICCV*, 1998.

[10] S. Gokturk, J. Bouget, and R. Grzeszocuk. A data-driven model for monocular face tracking. In *ICCV*, 2001.

[11] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making faces. In *SIGGRAPH*, 1998.

[12] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4), 1987.

[13] B. K. P. Horn, H. M. Hildne, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7), 1988.

[14] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *ICCV*, 1999.

[15] M. Kass, A. Witkin, and D. Terzopoulus. Snakes: Active contour models. *Int. J. of Computer Vision*, 1(4):321–331, 1987.

[16] A. Lanitis, T. C.J., C. T.F., and A. T. Automatic interpretation of human faces and hand gestures using flexible models. In *International Workshop on Automatic Face- and Gesture-Recognition*, 1995.

[17] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proc. 7th Int. Joint Conf. on Artif. Intell.*, 1981.

[18] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing realistic facial expressions from photographs. In *SIGGRAPH*, 1998.

[19] F. Pighin, D. H. Salesin, and R. Szeliski. Resynthesizing facial animation through 3d model-based tracking. In *ICCV*, 1999.

[20] D. L. Ruderman. Private communication.

[21] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994.

[22] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. of Computer Vision*, 9(2):137–154, 1992.