Re-Thinking Non-Rigid Structure From Motion

Vincent Rabaud Serge Belongie
Department of Computer Science and Engineering
University of California San Diego
{vrabaud,sjb}@cs.ucsd.edu

Abstract

*We present a novel approach to non-rigid structure from motion (*NRSFM*) from an orthographic video sequence, based on a new interpretation of the problem. Existing approaches assume the object shape space is well-modeled by a linear subspace. Our approach only assumes that small neighborhoods of shapes are well-modeled with a linear subspace. This constrains the shapes to belong to a manifold of dimensionality equal to the number of degrees of freedom of the object. After showing that the problem is still overconstrained, we present a solution composed of a novel initialization algorithm, followed by a robust extension of the Locally Smooth Manifold Learning algorithm tailored to the NRSFM problem. We finally present some test cases where the linear basis method fails (and is actually not meant to work) while the proposed approach is successful.*

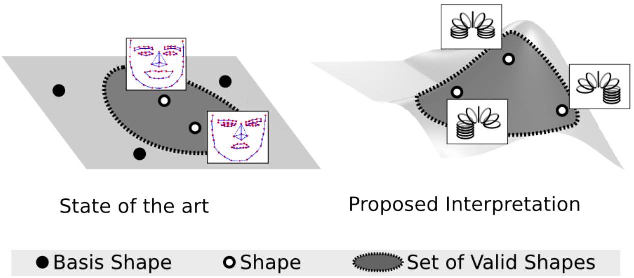


Figure 1. State of the art in non-rigid structure from motion assumes that a deformable 3D shape can be expressed as a linear combination of basis shapes. While this is a well-studied [3] and convenient assumption (*e.g.* for faces), there is no reason to believe that the manifold of the possible shapes of an object is linear (*e.g.*, as we will discover further, it is highly non-linear for a Slinky$^{\circledR}$/spring toy). In this work, the only constraint imposed on the 3D shape manifold is its dimensionality.

1. Introduction

In this paper, we place ourselves in the *Tomasi-Kanade paradigm*: features are fully tracked on a unique unknown object in an orthographic video sequence and only their 2D positions are known. We focus on the general problem where the object is non-rigid. The goal is to recover the 3D positions of the observed features over time: this is orthographic Non-rigid Structure From Motion or NRSFM.

In traditional NRSFM [21], a deforming object is assumed to adopt 3D shapes explainable by a linear combination of basis shapes. While this method can be computationally efficient and well suited to common objects of study (*e.g.* faces), there is no reason to believe that the possible 3D shapes of an object lie on a linear low-dimensional manifold (*cf*.Figure 1).

If we relax this assumption by assuming that only small neighborhoods of shapes are well-represented by a linear subspace, the set of possible 3D shapes can now be described as a smooth and low-dimensional manifold. Also, as a local neighborhood contains the different instances of how a 3D shape can deform, its dimensionality (and therefore, that of the manifold) is the number of degrees of freedom of the object. In this work, the only constraint imposed on the shape manifold is its dimensionality.

By moving away from the linear basis interpretation and adopting a manifold-learning framework to constrain the number of degrees of freedom of a deforming object, we can model more complex types of deformation and demonstrate success in cases where existing techniques fail.

The proposed method first relies on a new initialization that quantizes the non-rigidity into a temporal succession of rigid shapes. An optimization then follows to minimize at the same time the reprojection error as well as constraints on the smoothness of the 3D shape deformations. A constraint on the shape manifold dimensionality is also enforced to make sure the recovered 3D shapes have a given number of degrees of freedom.

After reviewing previous work in Section 2, we will detail our problem formulation in Section 3 and the corresponding solution in Section 4. Finally, in Section 5, weaknesses of the current state of the art and performances of the proposed approach will be highlighted on synthetic and real data.

2. Previous Work

References here are restricted to the general non-parametric case; hand-designed models are out of the scope of this paper.

Modern structure from motion started with the study of a rigid object under orthographic camera [18]. It was then extended to the projective case [16], multiple bodies [4] and articulated bodies [25]. The study of the orthographic non-rigid case started with [2, 21] and was then extended to the projective case [24, 11, 22].

Several NRSFM techniques have then improved their accuracy and efficiency by combining them with the feature tracker [21, 5], using a statistical model on top [19], including noise models [20] and even proving theoretical limitations [9].

Usually, these techniques are initialized by assuming that the observed object is globally rigid [10]. They are then optimized assuming a planar shape manifold model [1]. As this assumption is unreasonable for more complex types of deformations, we will need a more powerful manifold learning technique. Recent development in this area include kernel-PCA [13], LLE [12], ISOMAP [17], and more recently MVU [23]. *Locally Smooth Manifold Learning* or LSML [6] specifically meets our needs as it focuses on the manifold tangents, and hence its degrees of freedom, rather the manifold itself.

3. Problem

In our setup, n features are tracked over f frame under an orthographic camera: their 2D projected positions are known and can be stacked in f $2 \times n$ matrices W_t (t indexes time). The problem consists of recovering the 3D positions of these features (stacked in a $3 \times n$ shape matrix S_t at each frame) as well as the camera position: rotation R_t + translation $\mathbf{t}_t$.

These different parameters are obtained as a solution of the following *reprojection error* minimization:

$$\arg \min_{R_t, \mathbf{t}, S_t} \sum_{t=1}^{f} \left\| \Pi \left(R_t S_t + T_t \right) - W_t \right\|_F^2 \tag{1}$$

where $T_t = \mathbf{t}\mathbf{1}^\top$ and Π is the known projection matrix of a calibrated orthographic camera.

As this equation is severely underconstrained, some assumptions need to be made.

Camera Motion

First, we assume the camera motion is smooth: the camera position does not change much from one frame to the next. Therefore, we add two regularization terms to Equation (1):

$$\lambda_R \sum_{t=2}^{f} \| R_t - R_{t-1} \|_F^2 + \lambda_{\mathbf{t}} \sum_{t=2}^{f} \| \mathbf{t}_t - \mathbf{t}_{t-1} \|_F^2 \tag{2}$$

where λ_R and $\lambda_{\mathbf{t}}$ are regularization constants.

Smooth Deformation

Next, another valid assumption is that the observed object does not change much from one frame to the next: this is also a physical constraint that is usually assumed by the feature tracker prior to SFM. Hence a new term is added to Equation (1):

$$\lambda_S \sum_{t=2}^{f} \| S_t - S_{t-1} \|_2^2 \tag{3}$$

where λ_S is another regularization constant. Higher order smoothness terms could be used, but this one proved sufficient for all our experiments.

Degrees of Freedom

While previous methods only allow for linear deformations, the proposed approach only constrains the shape deformation to have at most d *degrees of freedom*, hence allowing for non-linear deformations. This means that all the possible shapes S_t lie on a d-dimensional *manifold* or that, locally, several nearby shapes lie on a d-dimensional linear subspace. We will make this constraint explicit in Section 4.2.

Complete Problem Formulation

Our problem can now be re-formulated as the following optimization:

$$\begin{aligned} \min_{R_t, \mathbf{t}, S_t} \sum_{t=0}^{f} \left\| \Pi \left(R_t S_t + T_t \right) - W_t \right\|_F^2 + \lambda_S \sum_{t=2}^{f} \| S_t - S_{t-1} \|_F^2 \\ + \lambda_R \sum_{t=2}^{f} \| R_t - R_{t-1} \|_F^2 + \lambda_{\mathbf{t}} \sum_{t=2}^{f} \| \mathbf{t}_t - \mathbf{t}_{t-1} \|_F^2 \end{aligned} \tag{4}$$

with the S_t's constrained to lie on a d-dimensional manifold.

Ambiguities

In SFM problems, there is usually an overall rigid ambiguity on the camera position. In our formulation, if the 3D shapes are modified by an overall rigid transform $(R, \mathbf{t})$, we obtain the following new unknowns: $S'_t = RS_t + \mathbf{t}$, $R'_t = R_t R^\top$ and $\mathbf{t}'_t = \mathbf{t}_t - R'_t\mathbf{t}$. As $R_tS_t + T_t = R'_tS'_t + T'_t$, the first term of Equation (4) will not change. As the Frobenius norm is rotation-invariant, the terms 2 and 3 will also be unchanged. Nonetheless, the last term becomes: $\left\|\mathbf{t}'_t - \mathbf{t}'_{t-1}\right\|_2 = \left\|\mathbf{t}_t - \mathbf{t}_{t-1} - (R'_t - R'_{t-1})\mathbf{t}\right\|_2 \neq \left\|\mathbf{t}_t - \mathbf{t}_{t-1}\right\|_2$ except if $\mathbf{t} = \mathbf{0}$.

Therefore, with our formulation, there is only a global *rotation ambiguity*, that we resolve by imposing $R_1 = \mathrm{I}_3$.

Also, the third component of the $\mathbf{t}_t$'s only matters in the last term of Equation (4) and a trivial optimum is reached by setting them to the same value. This arbitrary value is an ambiguity inherent to the orthographic model.

Over-Constrained Problem

We must recover $3f$ camera rotation angles, $2f$ camera translation parameters (2 and not 3, as there is a depth ambiguity with orthographic cameras) and $3n \times f$ 3D shape parameters. On the other hand, $2n \times f$ coordinates are given in the W_t matrices.

A d-dimensional linear subspace is parametrizable by a point and a basis of d elements, each of size $3n$. The point can be chosen anywhere in the subspace and therefore has $3n - d$ degrees of freedom. Also, the basis only has $3nd - d^2$ degrees of freedom. The subspace can therefore be explained with $3n - d + 3nd - d^2 = (d+1)(3n-d)$ parameters.

In our formulation, the shapes lie on a d-dimensional manifold, and locally on a d-dimensional linear subspace. Therefore, if the data is uniformly distributed on the manifold, there exists a neighborhood size s such that every neighborhood of s shapes approximately lies on a d-dimensional *linear* subspace. Consequently, every s-neighborhood can be explained by s linear combinations and a d-dimensional subspace. As a result, each frame can be explained, on average, by $\frac{1}{s}(sd + (d+1)(3n-d)) = d + \frac{1}{s}(d+1)(3n-d)$ parameters.

As shown in Table 1, our model requires more parameters than traditional SFM techniques but it is still over-constrained provided $5f + fd + \frac{f}{s}(d+1)(3n-d) < 2nf$ or again:

$$5 + d + \frac{1}{s}(d+1)(3n-d) < 2n \qquad (5)$$

To give a sense of magnitude, we can assume that usually, $d < 10$ and $n > 100$. Therefore, the inequality can be approximated by $s \gtrsim 1.5(d+1)$. Practically, this implies that the observed shapes have to appear in *similar* configurations at least $1.5(d+1)$ times (*similar* but not necessarily *exact*: our approach does not require a perfect repetitiveness of the motion).

	Rigid	Classical NRSFM	Proposed NRSFM
camera	$3f + 2f$		
basis	/	$(d+1)(3n-d)$	/
shape	$3n$	d	$d + \frac{1}{s}(d+1)(3n-d)$
total	$5f+3n$	$5f + fd + (d+1)(3n-d)$	$5f + fd + \frac{f}{s}(d+1)(3n-d)$

Table 1. Number of parameters defining the model in rigid SFM, traditional Non-Rigid SFM, and the proposed approach (all in the orthographic case). This statistics concerns a sequence of f frames with n features whose shape lie on a d-dimensional subspace ($d = 0$ in the rigid case). s is such that every s neighboring shapes constitute a linear subspace.

4. Method

4.1. Initialization

In order to minimize Equation (4), several techniques will be used including a partial closed-form solution, gradient descent and manifold denoising. The system is initialized by assuming that the object can be modeled at any frame as a rigid transformation of one of a collection of shape templates.

Hypergraph Interpretation

We first cluster the shapes S_t. We assume that if S_t and $S_{t'}$ are in the same cluster, they are derived from different rigid transformations of the same shape template and the corresponding reprojection error $\mathrm{err}_{tt'}$ is computed. We can then interpret the video sequence as a graph whose nodes are the frames and whose edges are weighted with the following reprojection affinity:

$$w_{tt'} = e^{-\frac{\mathrm{err}_{tt'}^2}{\sigma^2}}$$

An affinity close to 0 means the two 3D shapes are not in the same state. On the other hand, if it close to 1, it can indicate either a similar state, or an ambiguity due to a view point that "hides" the shape differences.

In order to disambiguate these cases, we compute higher order affinities using the reprojection error of triplets of frames using [18]. This method is known to be more stable than epipolar geometry but is also more expensive. Therefore, only pairs of frames that already have a high pairwise epipolar-based affinity are considered to form triplets.

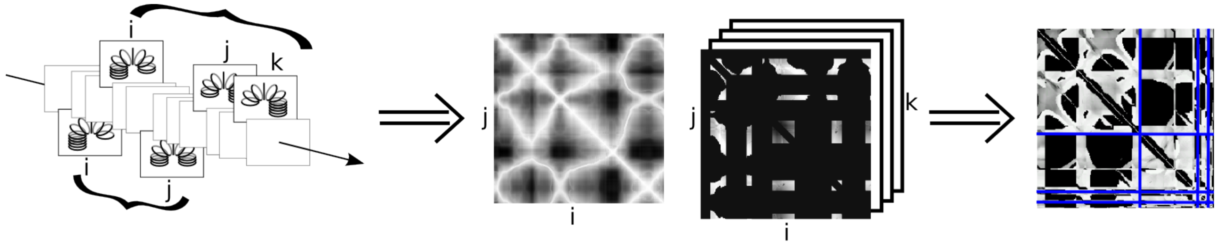


Figure 2. NRSFM is initialized by a *Rigid Shape Chain*. First of all, pairs and triplets of frames of the original video sequence are extracted. Each pair or triplet is then assumed to be projections of the same 3D view: the corresponding *reprojection error* is computed and used to define an affinity matrix/affinity tensor (only triplet composed of pairs with high affinities are considered for efficiency, hence the missing data in the image). These are then combined into a hypergraph that is flattened and clustered using *clique expansion* and *normalized cut*. The resulting clusters represent prototypical 3D shapes and they are then aligned to create a first estimate of the S_t's.

Once these dyadic and triadic affinities have been computed, we obtain a hypergraph with dyadic and triadic connections. In order to cluster the object shapes, the hypergraph is approximated by a graph using *clique expansion* [14].

Rigid Shape Chain

In order to deal with noise and outliers and to lower the dimensionality of the initialization, *quantization* is performed on the frames by separating them into clusters of equivalent 3D shape. To this end, a simple K-way clustering is applied using *normalized cuts* [15] to the previously defined frame affinity graph. K is chosen as the biggest number of clusters such that no cluster has less than 3 frames (in practice, K was between 10 and 30 for a 200-frame sequence).

Next, each cluster $\mathbf{C}^i$ is considered and its corresponding 3D shape $\mathbf{S}^i, i = 1, \ldots, K$ is computed (using [18]).

The resulting clustering is an initialization that actually explains the observed non-rigidity by a succession of rigid problems. We name this approach a *Rigid Shape Chain* (*cf*.Figure 2).

Initial Shape Alignment

The shapes obtained so far have been computed without regard to any deformation smoothness. This smoothness is enforced by applying a rigid transform $(\mathbf{R}^i, \mathbf{T}^i)$ to the $\mathbf{S}^i$'s in order to minimize the following temporal smoothness criterion:

$$\min_{(\mathbf{R}^i,\mathbf{T}^i)} \sum_{t=2}^{f} \left\| \mathbf{R}^{i(t)}\mathbf{S}^{i(t)} + \mathbf{T}^{i(t)} - \mathbf{R}^{i(t-1)}\mathbf{S}^{i(t-1)} - \mathbf{T}^{i(t-1)} \right\|_F^2 \tag{6}$$

where $i : t \mapsto i | S_t \in \mathbf{C}^i$. This minimization is simply a continuous version of the *exterior orientation* problem [7]. It can be solved by least squares optimization with random initialization and rotation constraints on the $\mathbf{R}^i$'s. In practice, we found that finding the closed form solution of the problem with no rotation constraints, and then projecting it onto $SO(3)$ [8] gave very good and fast results. The advantage of this approach is that it can rectify 3D shapes that have been flipped because of the possible *chirality* ambiguities appearing during the rigid shape chain.

Finally, the $\mathbf{R}^{i(t)}\mathbf{S}^{i(t)} + \mathbf{T}^{i(t)}$ are set to be the initial estimates of the S_t's (modulo a global rotation to ensure that $R_1 = \mathrm{I}_3$).

4.2. Minimization

After initialization, we obtain a reasonable approximation of the shapes and camera positions over time. The minimization of Equation (4) proceeds by alternating between the different unknowns, assuming the others are fixed.

Optimizing Camera Positions

If the S_t's and R_t's are fixed, finding a global optimum to Equation (4) with respect to **t** is trivial: it is the closed form solution of a sparse linear system.

Concerning the R_t's, the global optimum is as trivial, provided they are not constrained to be rotation matrices. In practice, we compute this global optimum and project it to $SO(3)$. If the result lowers the error, it is kept. Otherwise, we perform a projection-based gradient descent (at every step of gradient descent, the result is reprojected onto $SO(3)$).

LSML

When performing optimization on the S_t's, there are two criteria to take into account: the smoothness term in Equation (4) and the shape manifold dimensionality constraint. Optimizing the first one is just a least square optimization, but the second one needs a new interpretation.

The problem of imposing this dimensionality constraint can be seen as trying to force the S_t to lie on a d-dimensional manifold (as previously defined, d is the number of freedom of the observed object). After initialization, the S_t's are close to this manifold but are not on it: it is as if they formed a noisy low-dimensional manifold. As mentioned earlier, we have no reason to believe that this manifold is planar or isometric to a plane, hence our motivation for using LSML [6].

LSML is a manifold learning technique that seeks to learn from training data a smooth mapping from every point on the manifold to its local tangents. Consider:

$$\mathcal{M}: \begin{array}{l} \mathbb{R}^d \to \mathbb{R}^D \\ \mathbf{y} \mapsto \mathbf{x} \end{array}$$

a smooth mapping from a low d-dimensional space to a higher D-dimensional space (*e.g.* $\mathbf{y}$ is the coordinate on a manifold of a high dimensional point x), $d \ll D$. LSML seeks to recover the mapping:

$$\mathcal{H}: \begin{array}{l} \mathbb{R}^D \to \mathbb{R}^{D\times d} \\ \mathbf{x} \mapsto \left[\partial/\partial \mathbf{y}_1 \mathcal{M}(\mathbf{y}) \ldots \partial/\partial \mathbf{y}_d \mathcal{M}(\mathbf{y})\right] \end{array}$$

where $\mathcal{M}(\mathbf{y}) = \mathbf{x}$ and $\mathbf{y} = \left[\mathbf{y}_1 \ldots \mathbf{y}_d\right]^\top \in \mathbb{R}^d$.

The strength of this technique is that the mapping is not only learned for the training points but, by continuity, it is applicable to any new given point in $\mathbb{R}^D$.

LSML can also learn $\mathcal{H}$ from noisy data and then denoise it by making the points follow the gradient of an optimization criterion detailed in [6]. It is important to notice that LSML is limited to noise orthogonal to the manifold and cannot deal with collinear noise.

At each of our optimization steps, LSML is used to learn the manifold of noisy S_t's and recover the gradient for the LSML noise criterion.

Optimizing 3D Shapes

Optimizing the S_t's now needs to take two criteria into account - the one from Equation (4) and the one from LSML- and we have computed an optimization gradient for both, which we define as ∇_{Smooth} and ∇_{LSML}. It is an instance of *multi-objective optimization*. As, we do not want one of the constraints to be enforced more and impede the other, we decide not to use a weighted linear combination of the two criteria or a Lagrangian multiplier: we keep the constraints separate and and optimize them at the same time using *multi-level programming* to favor the dimensionality contraint.

Instead of using our two gradients as they are, we keep ∇_{LSML} responsible for any variation orthogonal to the shape manifold but only restrict ∇_{Smooth} to its projection $\nabla^{\perp}_{\text{Smooth}}$ onto a plane tangent to ∇_{LSML}: this way, $\nabla^{\perp}_{\text{Smooth}}$ does not interfere with any effect of ∇_{LSML}. Figure 3 illustrates this approach.

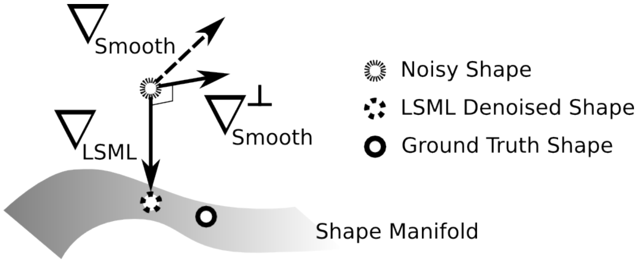


Figure 3. Two gradients are involved when optimizing the S_t's. First, there is a gradient ∇_{LSML} provided by LSML that tends to bring a noisy S_t back onto the shape manifold (but that is only orthogonal to it). Then, there is a gradient ∇_{Smooth} that minimizes the smoothness of the shape deformation. In order not to interfere with the LSML gradient, only its component orthogonal to ∇_{LSML} is considered: $\nabla^{\perp}_{\text{Smooth}}$. A linear combination of these two gradients is then searched to optimize the two criteria at the same time.

What follows is a gradient descent step following the gradient: $\nabla = a\nabla_{\text{LSML}} + b\nabla^{\perp}_{\text{Smooth}}$, where a and b are chosen so that both criteria are optimized at the same time.

Outliers

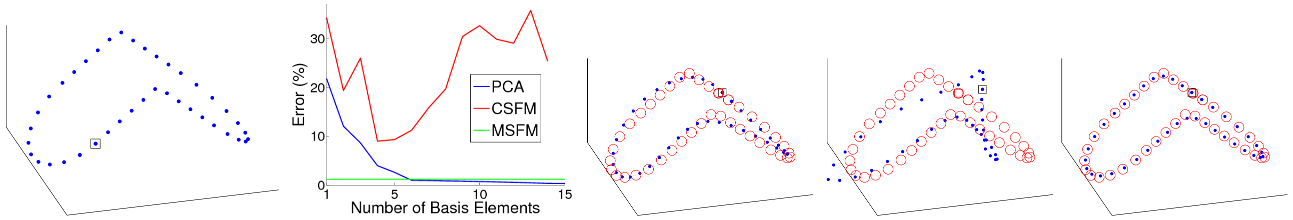
As LSML is not robust to outliers, special care is taken for any 3D shape that does not comply to the two following criteria:

- the distance to one of its neighbors is above three standard deviations (of the distribution of distances from points to their neighbors)
- the distance to its temporal predecessor is above three standard deviations.

These points are simply optimized by disregarding the manifold dimensionality constraint and by assigning them to their globally optimal value (which can be obtained in closed form, in a similar way as the $\mathbf{t}_t$'s).

Considerations

Each iteration of our optimization routine attains a lower error for Equation (4) that in the previous one, so it is bound to converge. In our experiments, we did not need to repeat the optimization (which could have been useful as it contains randomized algorithms such as Normalized Cut or LSML) but we faced a slow convergence (100 to 200 iterations were required).



(a) Frame Example (b) Reconstruction Performance (c) Reconstruction Examples (respectively PCA, CSFM, MSFM)

Figure 4. **Roller Coaster.** Figure 4(a) is a typical frame from the sequence. The black square refers to the first point of the coaster. Overall, the coaster loops 6 times. Figure 4(b) illustrates the performances of the different algorithms the percentage error refers to the average reconstruction error along the camera depth axis, normalized by the depth of the roller coaster as in [20]). PCA on the 3D points beats MSFM when the basis contains at least 6 elements which shows that the structure is not easily representable by a linear basis. Figure 4(c) shows examples of reconstruction for PCA (with 5 shapes in the basis), CSFM (with 4 shapes in the basis) and MSFM. The corner of the roller coaster presented a challenge for PCA to capture even with a bigger basis.

Also, most of the steps described previously take a few seconds to compute except for the triadic affinity computation and the S_t gradient descent involving LSML. Indeed, for each iteration, LSML needs to be retrained which can take up to a few minutes leading to an overall time of an hour or two.

5. Experiments

We experimented our method on both synthetic and real data. These experiments show the flexibility of our approach and its robustness as well as comparisons with state of the art (using code from [20]). We will refer to this *Classical NR-SFM* method as CSFM while we name ours *Manifold Structure From Motion* or MSFM.

5.1. Synthetic Data

Roller Coaster

The first data set is a synthetic roller coaster. The video sequence consists of 200 frames with 42 points moving on a fixed closed track. As seen on Figure 4(a), it looks like a closed roller coaster or a bent bike chain. The camera rotates around the object while it deforms. Both the object and the camera evolve at random speeds (no translation is involved for the camera).

This motion only has one degree of freedom as the points have to move along a fixed structure. Nonetheless, this motion causes problem for CSFM because it is not easily representable by a linear basis (for comparison, we show how hard it is for PCA to characterize the data given the full 3D points in Figure 4(b)). Also, the object does not have a main component that could be considered as rigid and be used as initialization for CSFM.

CSFM seems to fail in this case while MSFM only has 1.2% of error (the computed error is similar to [20] and detailed in Figure 4). Figure 4(b) also shows an interesting limitation of using a linear basis: as its focus is to minimize the reprojection error at any cost, more elements in the basis can help lowering it but at the cost of getting a worse 3D reconstruction.

It is also worth mentioning that the sequence of recovered S_t is also moving on itself (in addition to its intrinsic one degree of freedom): this is due to the fact that this optimizes the overall smoothness.

Bending Shark

The next experiment uses the shark data from [20]. It consists of 240 frames during which 91 points form a shark that bends its tail left/right or up/down (hence 2 degrees of freedom). In [19], they obtain errors of 1.24% and 2.5%. With our setup, we obtain 3%, which is comparable to their second best method. Several details are shown in Figure 5

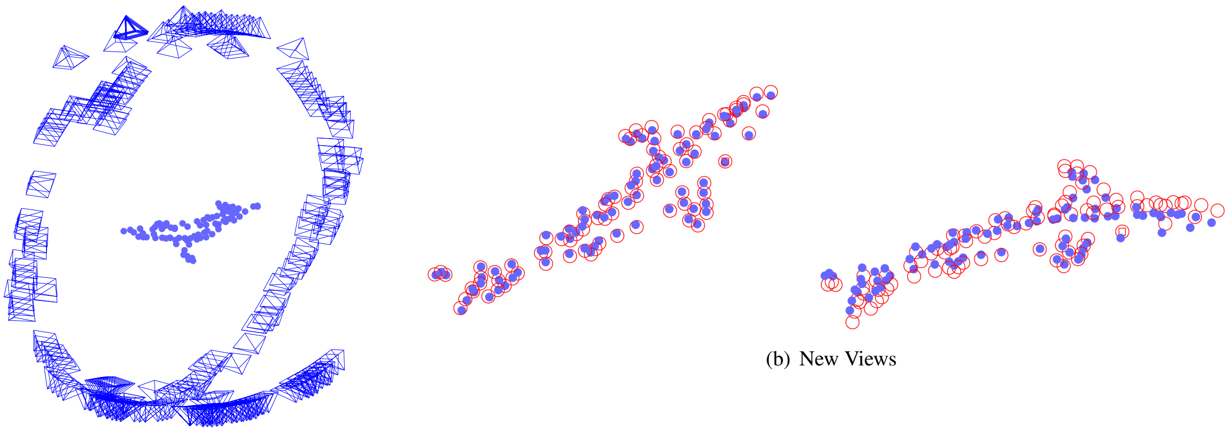
5.2. Real Data

The final round of experiments involves a calibrated video sequence of a Slinky® toy: this spring has a complex motion but, in this instance, only one degree of freedom. 27 painted features were tracked during a 300 frame long video sequence. There are approximately three periods of up/down movement that occur.

This data set is difficult for two reasons: the feature trajectories are noisy and the baseline between two extreme views is small. MSFM reconstructed a 1D-manifold of the different 3D-shapes with an average reprojection error of 1.4%. Examples are illustrated in Figure 6. CSFM fails in this case as the object does not have a main rigid part and because it undergoes non-linear deformations (like compression).

6. Conclusion

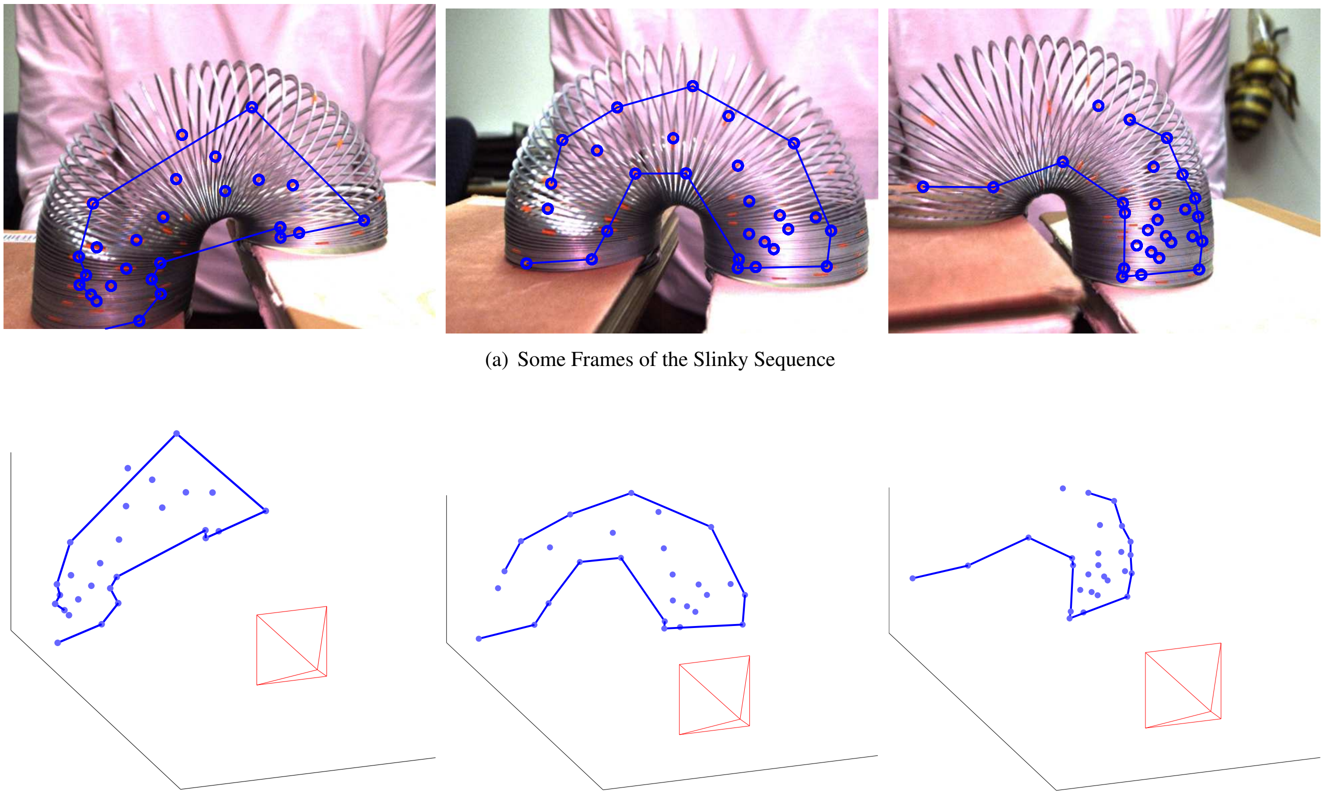
In this paper, we have presented a new approach to non-rigid structure from motion by focusing primarily on how to exploitconstraints on the degrees of freedom of the observed



(a) MSFM Initialization

(b) New Views

Figure 5. **Bending Shark.** Figure 5(a) illustrates the initial camera position estimation after the *rigid shape chain* computation. The ground truth camera movement of the camera is a view from below first followed by a full rotation around the shark. There are of course a few outliers but the overall camera trajectory is already well approximated. Figure 5(b) shows two reconstructions of the shark sequence. In the right image, the camera was pointing down leading to a lower quality reconstruction because of the depth ambiguity.



(a) Some Frames of the Slinky Sequence

(b) MSFM Reconstruction

Figure 6. **Slinky.** Figure 6(a) shows a few frames of the slinky sequence with some tracked features. The lines are just drawn to help visualize the 3D structure in the reconstruction Figure 6(b). These three frames demonstrate the compression the object undergoes: this property is difficult for a linear basis to model, even in 2D, hence the failure of CSFM. On the other hand, MSFM seems to recover correct 3D feature positions: the structure contains compression and seems to have a correct orientation.

object. By interpreting the d.o.f.'s as the dimensionality of the shape manifold, the problem boils down to performing manifold recovery and, by actually providing a good initialization via a *rigid shape chain*, it is an instance of manifold denoising.

Our method seems more intuitive and we also showed it is more flexible than a shape basis interpretation. Future work will include a better integration of LSML and analysis of the recovered shape manifold. Finally, the current initialization is still valid in the projective case but the minimization needs to be adapted.

Acknowledgment

This work was funded in part by NSF Career Grant #0448615, the Alfred P. Sloan Research Fellowship and the UCSD Division of the California Institute for Telecommunications and Information Technology, Calit2 and in part by the UCSD FWGrid Project, NSF Research Infrastructure Grant Number EIA-0303622. The authors would also like to thank S. Agarwal, K. Branson and P. Dollár for helpful discussions.

References

[1] W. Brand. Morphable 3d models from video. In *CVPR*, pages II–456– II–463, 2001.

[2] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR*, pages 690–696, 2000.

[3] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *PAMI*, number 6, pages 681–685, Jun 2001.

[4] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. In *IJCV*, volume 29, September 1998.

[5] A. del Bue, F. Smeraldi, and L. Agapito. Non-rigid structure from motion using ranklet-based tracking and non-linear optimization. In *Image and Vision Computing*, volume 25, pages 297–310, March 2007.

[6] P. Dollár, V. Rabaud, and S. Belongie. Non-isometric manifold learning: Analysis and an algorithm. In *ICML*, 2007.

[7] B. Horn. *Robot Vision*. MIT Press, 1986.

[8] B. Horn. Closed form solutions of absolute orientation using orthonormal matrices. In *Journal of the Optical Society of America*, volume 5, pages 1127–1135, 1987.

[9] J. C. Jing Xiao and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. In *IJCV*, volume 67, April 2006.

[10] T. Kim and K.-S. Hong. Estimating approximate average shape and motion of deforming objects with a monocular view. In *IJPRAI*, volume 19, pages 585–601, 2005.

[11] X. Llado, A. Del Bue, and L. Agapito. Non-rigid 3d factorization for projective reconstruction. In *BMVC*, 2005.

[12] L. K. Saul and S. T. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. In *JMLR*, 2003.

[13] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. In *NIPS*, 1998.

[14] D. G. Schweikert and B. W. Kernighan. A proper model for the partitioning of electrical circuits. In *DAC '72: Proceedings of the 9th workshop on Design automation*, pages 57–62. ACM, 1972.

[15] J. Shi and J. Malik. Normalized cuts and image segmentation. In *PAMI*, number 8, pages 888–905, Aug 2000.

[16] S. Soatto and P. Perona. Recursive estimation of camera motion from uncalibrated image sequences. In *ICIP*, pages III: 58–62, 1994.

[17] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dim. reduct. In *Science*, volume 290, 2000.

[18] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. In *IJCV*, volume 9, November 1992.

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

[20] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *PAMI*, 30(5):878–892, 2008.

[21] L. Torresani, D. Yang, E. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. In *CVPR*, pages I–493– I–500, 2001.

[22] R. Vidal and D. Abretske. Nonrigid shape and motion from multiple perspective views. In *ECCV*, pages II: 205–218, 2006.

[23] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *IJCV*, 2006.

[24] J. Xiao and T. Kanade. Uncalibrated perspective reconstruction of deformable structures. In *ICCV*, pages 1075– 1082 Vol. 2, 2005.

[25] J. Yan and M. Pollefeys. A factorization-based approach to articulated motion recovery. In *CVPR*, pages 815– 821, 2005.