Non-Rigid Structure-From-Motion: Estimating Shape and Motion with Hierarchical Priors

Lorenzo Torresani, Aaron Hertzmann, *Member, IEEE*, Christoph Bregler

LT is at Microsoft Research, Cambridge. AH is at University of Toronto. CB is at New York University.

Abstract

This paper describes methods for recovering time-varying shape and motion of non-rigid 3D objects from uncalibrated 2D point tracks. For example, given a video recording of a talking person, we would like to estimate the 3D shape of the face at each instant, and learn a model of facial deformation. Time-varying shape is modeled as a rigid transformation combined with a non-rigid deformation. Reconstruction is ill-posed if arbitrary deformations are allowed, and thus additional assumptions about deformations are required. We first suggest restricting shapes to lie within a low-dimensional subspace, and describe estimation algorithms. However, this restriction alone is insufficient to constrain reconstruction. To address these problems, we propose a reconstruction method using a Probabilistic Principal Components Analysis (PPCA) shape model, and an estimation algorithm that simultaneously estimates 3D shape and motion for each instant, learns the PPCA model parameters, and robustly fills-in missing data points. We then extend the model to model temporal dynamics in object shape, allowing the algorithm to robustly handle severe cases of missing data.

Index Terms

Non-rigid Structure-From-Motion, Probabilistic Principal Components Analysis, Factor Analysis, Linear Dynamical Systems, Expectation-Maximization

I. Introduction and Related Work

A central goal of computer vision is to reconstruct the shape and motion of objects from images. Reconstruction of shape and motion from point tracks — known as structure-from-motion — is very well-understood for rigid objects [17], [26], and multiple rigid objects [10], [16]. However, many objects in the real world deform over time, including people, animals, and elastic objects. Reconstructing the shape of such objects from imagery remains an open problem.

In this paper, we describe methods for Non-Rigid Structure-From-Motion (NRSFM): extracting 3D shape and motion of non-rigid objects from 2D point tracks. Estimating time-varying 3D shape from monocular 2D point tracks is inherently underconstrained without prior assumptions. However, the apparent ease with which humans interpret 3D motion from ambiguous point tracks (e.g., [18], [30]) suggests that we might take advantage of prior assumptions about motion. A key question is: what should these prior assumptions be? One possible approach is to explicitly describe which shapes are most likely (e.g., by hard-coding a model [32]), but this can be extremely difficult for all but the simplest cases. Another approach is to learn a model from

training data. Various authors have described methods for learning linear subspace models with Principal Components Analysis (PCA) for recognition, tracking, and reconstruction [4], [9], [24], [31]. This approach works well if appropriate training data is available; however, this is often not the case. In this paper, we do not assume that any training data is available.

In this work, we model 3D shapes as lying near a low-dimensional subspace, with a Gaussian prior on each shape in the subspace. Additionally, we assume that the non-rigid object undergoes a rigid transformation at each time instant (equivalently, a rigid camera motion), followed by an weak-perspective camera projection. This model is a form of Probabilistic Principal Components Analysis (PPCA). A key feature of this approach is that we do not require any prior 3D training data. Instead, the PPCA model is used as a hierarchical Bayesian prior [13] for measurements. The hierarchical prior makes it possible to simultaneously estimate 3D shape and motion for all time instants, learn the deformation model, and robustly fill-in missing data points. During estimation, we marginalize out deformation coefficients to avoid overfitting, and solve for MAP estimates of the remaining parameters using Expectation-Maximization (EM). We additionally extend the model to learn temporal dynamics in object shape, by replacing the PPCA model with a Linear Dynamical System (LDS). The LDS model adds temporal smoothing, which improves reconstruction in severe cases of noise and missing data.

Our original presentation of this work employed a simple linear subspace model instead of PPCA [7]. Subsequent research has employed variations of this model for reconstruction from video, including the work of Brand [5] and our own [27], [29]. A significant advantage of the linear subspace model is that, as Xiao et al. [34] have shown, a closed-form solution for all unknowns is possible (with some additional assumptions). Brand [6] describes a modified version of this algorithm employing low-dimensional optimization. However, in this paper, we argue that the PPCA model will obtain better reconstructions than simple subspace models, because PPCA can represent and learn more accurate models, thus avoiding degeneracies that can occur with simple subspace models. Moreover, the PPCA formulation can automatically estimate all model parameters, thereby avoiding the difficulty of manually tuning weight parameters. Our methods use the PPCA model as a hierarchical prior for motion, and suggests the use of more sophisticated prior models in the future. Toward this end, we generalize the model to represent linear dynamics in deformations. A disadvantage of this approach is that numerical optimization procedures are required in order to perform estimation.

In this paper, we describe the first comprehensive performance evaluation of several NRSFM algorithms on synthetic datasets and real-world datasets obtained from motion capture. We show that, as expected, simple subspace and factorization methods are extremely sensitive to noise and missing data, and that our probabilistic method gives superior results in all real-world examples.

Our algorithm takes 2D point tracks as input; however, due to the difficulties in tracking non-rigid objects, we anticipate that NRSFM will ultimately be used in concert with tracking and feature detection in image sequences, such as [5], [11], [27], [29].

Our use of linear models is inspired by their success in face recognition [24], [31], tracking [9] and computer graphics [20]. In these cases, the linear model is obtained from complete training data, rather than from incomplete measurements. Bascle and Blake [2] learn a linear basis of 2D shapes for non-rigid 2D tracking, and Blanz and Vetter [4] learn a PPCA model of human heads for reconstructing 3D heads from images. These methods require the availability of a training database of the same "type" as the target motion. In contrast, our system performs learning simultaneously with reconstruction. The use of linear subspaces can also be motivated by noting that many physical systems (such as linear materials) can be accurately represented with linear subspaces (e.g., [1]).

II. Shape and motion models

We assume that a scene consists of J time-varying 3D points $\mathbf{s}_{j,t} = [X_{j,t}, Y_{j,t}, Z_{j,t}]^T$, where j is an index over scene points, and t is an index over image frames. This time-varying shape represents object deformation in a local coordinate frame. At each time t, these points undergo a rigid motion and weak-perspective projection to 2D:

$$\underbrace{\mathbf{p}_{j,t}}_{2\times 1} = \underbrace{c_t}_{1\times 1}\underbrace{\mathbf{R}_t}_{2\times 3}\left(\underbrace{\mathbf{s}_{j,t}}_{3\times 1} + \underbrace{\mathbf{d}_t}_{3\times 1}\right) + \underbrace{\mathbf{n}_t}_{2\times 1} \tag{1}$$

where $\mathbf{p}_{j,t} = [x_{j,t}, y_{j,t}]^T$ is the 2D projection of scene point j at time t, $\mathbf{d}_t$ is a 3×1 translation vector, $\mathbf{R}_t$ is a 2×3 orthographic projection matrix, c_t is the weak-perspective scaling factor, and $\mathbf{n}_t$ is a vector of zero-mean Gaussian noise with variance σ^2 in each dimension. We can also stack the points at each time-step into vectors:

$$\underbrace{\mathbf{p}_t}_{2J\times 1} = \underbrace{\mathbf{G}_t}_{2J\times 3J}\left(\underbrace{\mathbf{s}_t}_{3J\times 1} + \underbrace{\mathbf{D}_t}_{3J\times 1}\right) + \underbrace{\mathbf{N}_t}_{2J\times 1} \tag{2}$$

where $\mathbf{G}_t$ replicates the matrix $c_t\mathbf{R}_t$ across the diagonal, $\mathbf{d}_t$ stacks J copies of $\mathbf{d}_t$, and $\mathbf{N}_t$ is a zero-mean Gaussian noise vector. Note that rigid motion of the object and rigid motion of the camera are interchangeable. For example, this model can represent an object deforming within a local coordinate frame, undergoing a rigid motion, and viewed by a moving orthographic camera. In the special case of rigid shape (with $\mathbf{s}_t = \mathbf{s}_1$ for all t), this reduces to the classic rigid SFM formulation studied by Tomasi and Kanade [26].

Our goal is to estimate the time-varying shape $\mathbf{s}_t$ and motion $(c_t\mathbf{R}_t, \mathbf{D}_t)$ from observed projections $\mathbf{p}_t$. Without any constraints on the 3D shape $\mathbf{s}_t$, this problem is extremely ambiguous. For example, given a shape $\mathbf{s}_t$ and motion $(\mathbf{R}_t, \mathbf{D}_t)$ and an arbitrary orthonormal matrix $\mathbf{A}_t$, we can produce a new shape $\mathbf{A}_t\mathbf{s}_t$ and motion $(c_t\mathbf{R}_t\mathbf{A}_t^{-1}, \mathbf{A}_t\mathbf{D}_t)$ that together give identical 2D projections as the original model, even if a different matrix $\mathbf{A}_t$ is applied in every frame [35]. Hence, we need to make use of additional prior knowledge about the nature of these shapes. One approach is to learn a prior model from training data [2], [4]. However, this requires that we have appropriate training data, which we do not assume is available. Alternatively, we can explicitly design constraints on the estimation. For example, one may introduce a simple Gaussian prior on shapes $\mathbf{s}_t \sim \mathcal{N}(\bar{\mathbf{s}}; \mathbf{I})$, or, equivalently, a penalty term of the form $\sum_t ||\mathbf{s}_t - \bar{\mathbf{s}}||^2$ [35]. However, many surfaces do not deform in such a simple way, i.e., with all points uncorrelated and varying equally. For example, when tracking a face, we should penalize deformations of the nose much more than deformations of the lips.

In this paper, we employ a probabilistic deformation model with unknown parameters. In Bayesian statistics, this is known as a hierarchical prior [13]: shapes are assumed to come from a common probability distribution function (PDF), but the parameters of this distribution are not known in advance. The prior over the shapes is defined by marginalizing over these unknown parameters[1]. Intuitively, we are constraining the problem by simultaneously fitting the 3D shape reconstructions to the data, fitting the shapes to a model, and fitting the model to the shapes. This type of hierarchical prior is an extremely powerful tool for cases where the data come from a common distribution that is not known in advance. Suprisingly, hierarchical priors have seen very little use in computer vision.

In the next section, we introduce a simple prior model based on a linear subspace model of

[1]For convenience, we estimate values of some of these parameters instead of marginalizing.

shape, and discuss why this model is unsatisfactory for NRSFM. We then describe a method based on Probabilistic PCA that addresses these problems, followed by an extension that models temporal dynamics in shapes. We then describe experimental evaluations on synthetic and real-world data.

A. Linear subspace model

A common way to model non-rigid shapes is to represent them in a K-dimensional linear subspace. In this model, each shape is described by a K-dimensional vector $\mathbf{z}_t$; the corresponding 3D shape is:

$$\underbrace{\mathbf{s}_t}_{3J\times 1} = \underbrace{\bar{\mathbf{s}}}_{3J\times 1} + \underbrace{\mathbf{V}}_{3J\times K}\underbrace{\mathbf{z}_t}_{K\times 1} + \underbrace{\mathbf{m}_t}_{3J\times 1} \tag{3}$$

where $\mathbf{m}_t$ represents a Gaussian noise vector. Each column of the matrix $\mathbf{V}$ is a basis vector, and each entry of $\mathbf{z}_t$ is a corresponding weight that determines the contributions of the basis vector to the shape at each time t. We refer to the weights $\mathbf{z}_t$ as *latent coordinates*. (Equivalently, the space of possible shapes may be described by convex combinations of *basis shapes*, by selecting $K+1$ linearly independent points in the space.) The use of a linear model is inspired by the observation that many high-dimensional data-sets can be efficiently represented by low-dimensional spaces; this approach has been very successful in many applications (e.g., [4], [9], [31])

Maximum likelihood estimation entails minimizing the following least-squares objective with respect to the unknowns:

$$L_{MLE} = -\ln p(\mathbf{p}_{1:T}|c_{1:T}, \mathbf{R}_{1:T}, \mathbf{V}_{1:K}, \mathbf{d}_{1:T}, \mathbf{z}_{1:T}) \tag{4}$$

$$= \frac{1}{2\sigma^2}\sum_{j,t} ||\mathbf{p}_{j,t} - c_t\mathbf{R}_t(\bar{\mathbf{s}}_j + \mathbf{V}_j\mathbf{z}_t + \mathbf{d}_t)||^2 + JT\ln(2\pi\sigma^2) \tag{5}$$

where $\mathbf{V}_j$ denotes the row of $\mathbf{V}$ corresponding to the j-th point.

a) Ambiguities and degeneracies.: Although the linear subspace model helps constrain the reconstruction problem, many difficulties remain.

Suppose the linear subspace and motion $(\bar{\mathbf{S}}, \mathbf{V}, \mathbf{G}_t, \mathbf{D}_t)$ were known in advance, and that $\mathbf{G}_t\mathbf{V}$ is not full-rank, at some time t. For any shape represented as $\mathbf{z}_t$, there is a linear subspace of distinct 3D shapes $\mathbf{z}_t + \alpha\mathbf{w}$ that project to the same 2D shape, where $\mathbf{w}$ lies in the nullspace of $\mathbf{G}_t\mathbf{V}$ and α is an arbitrary constant. (Here we assume that $\mathbf{V}$ is full-rank; if not, redundant columns should be removed). Since we do not know the shape basis in advance, the optimal

solution may select $\mathbf{G}_t\mathbf{V}$ to be low-rank, and use the above ambiguity to obtain a better fit to the data, at the expense of very unreliable depth estimates. In the extreme case of $K = 2J$, reconstruction becomes totally unconstrained, since $\mathbf{V}$ represents the full shape space rather than a subspace. We can avoid the problem by reducing K, but we may need to make K artificially small. In general, we cannot assume that small values of K are sufficient to represent the variation of real-world shapes. These problems will become more significant for larger K. Ambiguities will be come increasingly significant when point tracks are missing, an unavoidable occurance with real tracking.

In general, we expect the linear subspace model to be sensitive to the choice of K. If K is too large for the object being tracked, then the extra degrees-of-freedom will be unconstrained by the data, and end up fitting noise. However, if K is too small, then important degrees of variation will be lost. In practice, there may not be a clear "best" value of K that will capture all variation while discarding all noise. Empirically, the eigenvalue spectrum obtained from PCA on real-world 3D shapes tends to fall off smoothly rather than being bounded at a small value of K. An example from facial motion capture data is shown in Figure 1.

An additional ambiguity occurs in the representation of the subspace; specifically, we can apply an arbitrary affine transformation $\mathbf{A}$ to the subspace (replacing $\mathbf{V}$ with $\mathbf{V}\mathbf{A}^{-1}$ and $\mathbf{z}$ with $\mathbf{A}\mathbf{z}$). However, this does not change reconstruction error or the underlying subspace, so we do not consider it to be a problem.

Although the subspace model can be made to work in simple situations, particularly with limited noise and small values of K, the above ambiguities indicate that it will scale poorly to larger problems, and become increasingly sensitive to manual parameter tuning. As the number of basis shapes grows, the problem is more likely to become unconstrained, eventually approaching the totally unconstrained case described in the previous section, where each frame may have an entirely distinct 3D shape.

Most NRSFM methods make an additional assumption that the recovered shape and motion can be obtained by transforming a low-rank factorization of the original point tracks [5], [6], [7], [34]. The main appeal of these approaches is that they decompose the problem into much simpler ones. However, this approach is only justified when measurement noise is negligible;

with non-negligible noise, these methods give no guarantee of statistical optimality,[2] and may in practice be highly biased. We do not expect noise in real NRSFM problems to be negligible, and the importance of noise modeling is borne out by our experiments.

B. Probabilistic PCA model

We propose using Probabilistic PCA (PPCA) [22], [25] to describe the distribution over shapes. In PPCA, we place a Gaussian prior distribution on the weights $\mathbf{z}_t$, and define the rest of the model as before:

$$\mathbf{z}_t \sim \mathcal{N}(0; \mathbf{I}) \tag{6}$$

$$\mathbf{s}_t = \bar{\mathbf{s}} + \mathbf{V}\mathbf{z}_t + \mathbf{m}_t \tag{7}$$

$$\mathbf{p}_t = \mathbf{G}_t(\mathbf{s}_t + \mathbf{D}_t) + \mathbf{n}_t \tag{8}$$

where $\mathbf{m}_t$ and $\mathbf{n}_t$ are zero-mean Gaussian vectors. with variance σ_m^2 and σ^2. Moreover, when estimating unknowns in PPCA, *the latent coordinates* $\mathbf{z}_t$ *are marginalized out*: we never explicitly solve for $\mathbf{z}_t$. Because any linear transformation of a Gaussian variable is Gaussian, the distribution over $\mathbf{p}_t$ is Gaussian[3]. Combining Equations 6-8 gives:

$$\mathbf{p}_t \sim \mathcal{N}(\mathbf{G}_t(\bar{\mathbf{s}} + \mathbf{D}_t); \mathbf{G}_t \left(\mathbf{V}\mathbf{V}^T + \sigma_m^2\mathbf{I}\right) \mathbf{G}_t^T + \sigma^2\mathbf{I}) \tag{10}$$

In this model, solving NRSFM — estimating motion while learning the deformation basis — is a special form of estimating a Gaussian distribution. In particular, we simply maximize the

[2]NRSFM can be posed as a constrained least-squares problem: factor the data into the product of two matrices that minimize reprojection error while satisfying certain constraints. Singular Value Decomposition (SVD) provides an optimal least-squares factorization, but does not guarantee that any constraints are satisfied. One approach has been to find a subspace transformation to the SVD solution to attempt to satisfy the constraints, but there is no guarantee that such a transformation exists. Hence, such methods cannot guarantee both that the constraints are satisfied and that the solution is optimal. For example, Tomasi and Kanade's algorithm [26] guarantees optimal affine reconstructions but not optimal rigid reconstructions. In practice, it often finds acceptable solutions. However, in the NRSFM case, the constraints are much more complex.

[3]This may also be derived by directly marginalizing out $\mathbf{z}_t$:

$$p(\mathbf{p}_t) = \int p(\mathbf{p}_t, \mathbf{z}_t) d\mathbf{z}_t = \int p(\mathbf{p}_t|\mathbf{z}_t) p(\mathbf{z}_t) d\mathbf{z}_t \tag{9}$$

where $p(\mathbf{p}_t|\mathbf{z}_t)$ is Gaussian (as given by Equations 7 and 8), and $p(\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_t|0; \mathbf{I})$, assuming that we condition on fixed values of $\bar{\mathbf{s}}$, $\mathbf{V}$, $\mathbf{G}_t$, $\mathbf{D}_t$, σ^2, and σ_m^2. Simplifying the above expression gives Equation 10.

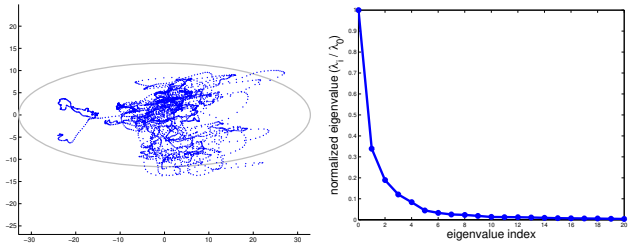

Fig. 1. *Left:* 2D coordinates obtained by applying conventional PCA to aligned 3D face shapes. The best-fit Gaussian distribution is illustrated by a gray ellipse. *Right:* Eigenvalue spectrum of the face data. (Details of the original data are given in Section IV-B)

joint likelihood of the measurements $\mathbf{p}_{1:T}$, or, equivalently, the negative logarithm of the joint likelihood:

$$\begin{aligned} L &= \frac{1}{2}\sum_t \left(\mathbf{p}_t - (\mathbf{G}_t(\bar{\mathbf{s}} + \mathbf{D}_t)\right)^T \left(\mathbf{G}_t\left(\mathbf{V}\mathbf{V}^T + \sigma_m^2\mathbf{I}\right)\mathbf{G}_t^T + \sigma^2\mathbf{I}\right)\left(\mathbf{p}_t - (\mathbf{G}_t(\bar{\mathbf{s}} + \mathbf{D}_t)\right)^T \\ &\quad + \frac{1}{2}\sum_t \ln\left|\mathbf{G}_t\left(\mathbf{V}\mathbf{V}^T + \sigma_m^2\mathbf{I}\right)\mathbf{G}_t^T + \sigma^2\mathbf{I}\right| + JT\ln(2\pi) \end{aligned} \tag{11}$$

We will describe an estimation algorithm in Section III-B.

Intuitively, the NRSFM problem can be stated as solving for shape and motion such that the reconstructed 3D shapes are as "similar" to each other as possible. In this model, shapes arise from a Gaussian distribution with mean $\bar{\mathbf{s}}$ and covariance $\mathbf{V}\mathbf{V}^T + \sigma_m^2\mathbf{I}$. Maximizing the likelihood of the data simultaneously optimizes the 3D shapes according to both the measurements and the Gaussian prior over shapes, while adjusting the Gaussian prior to fit the individual shapes. An alternative approach would be to explicitly learn a $3J \times 3J$ covariance matrix. However, this involves many more parameters than necessary, whereas PPCA provides a reduced-dimensionality representation of a Gaussian. This model provides several advantages over the linear subspace model. First, the Gaussian prior on $\mathbf{z}_t$ represents an explicit assumption that the latent coordinates $\mathbf{z}_t$ for each pose will be similar to each other; i.e., the $\mathbf{z}_t$ coordinates are not unconstrained. Empirically, we find this assumption to be justified. For example, Figure 1 shows 2D coordinates for 3D shapes taken from a facial motion capture sequence, computed by conventional PCA. These coordinates do not vary arbitrarily, but remain confined to a small region of space. In general, we find this observation consistent when applying PCA to many different types of datasets. This Gaussian prior resolves the important ambiguities described

in the previous section. Depth and scaling ambiguities are resolved by preferring shapes with smaller magnitudes of $\mathbf{z}_t$. The model is robust to large or mis-specified values of K, since very small variances will be learned for extraneous dimensions. A rotational ambiguity remains: replacing $\mathbf{V}$ and $\mathbf{z}$ with $\mathbf{V}\mathbf{A}^T$ and $\mathbf{A}\mathbf{z}$ (for any orthonormal matrix $\mathbf{A}$) does not change the likelihood. However, this ambiguity has no impact on the resulting distribution over 3D shapes and can be ignored.

Second, this model accounts for uncertainty in the latent coordinates $\mathbf{z}_t$. These coordinates are often underconstrained in some axes, and cannot necessarily be reliably estimated, especially during the early stages of optimization. Moreover, a concern with large K is the large number of unknowns in the problem, including K elements of $\mathbf{z}_t$ for each time t. Marginalizing over these coordinates removes these variables from estimation. Removing these unknowns also makes it possible to learn all model parameters — including the prior and noise terms — simultaneously without overfitting. This means that regularization terms need not be set manually for each problem, and can thus be much more sophisticated and have many more parameters than otherwise. In practice, we find that this leads to significantly improved reconstructions over user-specified shape PDFs. It might seem that, since the parameters of the PDF are not known *a priori*, the algorithm could estimate wildly varying shapes, and then learn a correspondingly spread-out PDF. However, such a spread-out PDF would assign very low likelihood to the solution and thus be suboptimal; this is a typical case of Bayesian inference automatically employing "Occam's Razor" [19]: data-fitting is automatically balanced against the model simplicity. One way to see this is to consider the terms of the log probability in Equation 11: the first term is a data-fitting term, and the second term is a regularization term that penalizes spread-out Gaussians. Hence, the optimal solution trades-off between (a) fitting the data, (b) regularizing by penalizing distance between shapes and the shape PDF, and (c) minimizing the variance of the shape PDF as much as possible. The algorithm simultaneously regularizes and learns the regularization.

b) Regularized linear subspace model: An alternative approach to resolving ambiguities is to introduce regularization terms that penalize large deformations. For example, if we solve for latent coordinates $\mathbf{z}_t$ in the above model rather than marginalizing them out, then the

corresponding objective function becomes:

$$\begin{aligned} L_{MAP} &= -\ln p(\mathbf{p}_{1:T}|\mathbf{R}_{1:T}, \mathbf{V}_{1:K}, \mathbf{d}_{1:T}, \mathbf{z}_{1:T}) && (12) \\ &= \frac{1}{2\sigma^2}\sum_{j,t} ||\mathbf{p}_{j,t} - c_t\mathbf{R}_t(\bar{\mathbf{s}}_j + \mathbf{V}_j\mathbf{z}_t + \mathbf{d}_t)||^2 \\ &\quad + \frac{1}{2\sigma_z^2}\sum_t ||\mathbf{z}_t||^2 + \frac{JT}{2}\ln(2\pi\sigma^2) + \frac{TK}{2}\ln(2\pi\sigma_z^2) && (13) \end{aligned}$$

which is the same objective function as in Equation 5 with the addition of a quadratic regularizer on $\mathbf{z}_t$. However, this objective function is degenerate. To see this, consider an estimate of the basis $\hat{\mathbf{V}}$ and latent coordinates $\hat{\mathbf{z}}_{1:T}$. If we scale all of these terms as

$$\hat{\mathbf{V}} \leftarrow 2\hat{\mathbf{V}}, \qquad \hat{\mathbf{z}}_t \leftarrow \frac{1}{2}\hat{\mathbf{z}}_t \tag{14}$$

then the objective function must decrease. Consequently, this objective function is optimized by infinitesimal latent coordinates, but without any improvement to the reconstructed 3D shapes.

Previous work in this area has used various combinations of regularization terms [5], [29]. Designing appropriate regularization terms and choosing their weights is generally not easy; we could place a prior on the basis (e.g., penalize the Frobenius norm of $\mathbf{V}$), but it is not clear how to balance the weights of the different regularization terms; for example, the scale of the $\mathbf{V}$ weight will surely depend on the scale of the specific problem being addressed. One could require the basis to be orthonormal, but this leads to an isotropic Gaussian distribution, unless separate variances were specified for every latent dimension. One could also attempt to learn the weights together with the model, but this would almost certainly be underconstrained with so many more unknown parameters than measurements. In constrast, our PPCA-based approach avoids these difficulties without requiring any additional assumptions or regularization.

C. Linear Dynamics model

In many cases, point tracking data comes from sequential frames of a video sequence. In this case, there is additional temporal structure in the data that can be modeled in the distribution over shapes. For example, 3D human facial motion shown in 2D PCA coordinates in Figure 1 shows distinct temporal structure: the coordinates move smoothly through the space, rather than appearing as random, IID samples from a Gaussian.

Here we model temporal structure with a linear dynamical model of shape:

$$\mathbf{z}_1 \sim \mathcal{N}(0; \mathbf{I}) \tag{15}$$

$$\mathbf{z}_t = \Phi \mathbf{z}_{t-1} + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(0; \mathbf{Q}) \tag{16}$$

In this model, the latent coordinates $\mathbf{z}_t$ at each time step are produced by a linear function of the previous time step, based on the $K \times K$ transition matrix Φ, plus additive Gaussian noise with covariance $\mathbf{Q}$. Shapes and observations are generated as before:

$$\mathbf{s}_t = \bar{\mathbf{s}} + \mathbf{V}\mathbf{z}_t + \mathbf{m}_t \tag{17}$$

$$\mathbf{p}_t = \mathbf{G}_t(\mathbf{s}_t + \mathbf{D}_t) + \mathbf{n}_t \tag{18}$$

As before, we solve for all unknowns except for the latent coordinates $\mathbf{z}_{1:T}$, which are marginalized out. The algorithm is described in Section III-C. This algorithm learns 3D shape with temporal smoothing, while simultaneously learning the smoothness terms.

III. Algorithms

A. Least-squares NRSFM with a linear subspace model

As a baseline algorithm, we introduce a technique that optimizes the least-squares objective function (Equation 5) with block coordinate descent. This method, which we refer to as BCD-LS, was originally presented in [29]. No prior assumption is made about the distribution of the latent coordinates, and so the weak-perspective scaling factor c_t can be folded into the latent coordinates, by representing the shape basis as:

$$\tilde{\mathbf{V}} \equiv [\bar{\mathbf{s}}, \mathbf{V}], \qquad \tilde{\mathbf{z}}_t^c \equiv c_t[1, \mathbf{z}_t^T]^T \tag{19}$$

We then optimize directly for these unknowns. Additionally, since the depth component of rigid translation is unconstrained, we estimate 2D translations $\mathbf{T}_t \equiv \mathbf{G}_t\mathbf{D}_t = [c_t\mathbf{R}_t\mathbf{d}_t, ..., c_t\mathbf{R}_t\mathbf{d}_t] \equiv [\mathbf{t}_t, ..., \mathbf{t}_t]$. The variance terms are irrelevant in this formulation and can be dropped from Equation 5, yielding the following two equivalent forms:

$$L_{MLE} = \sum_{j,t} ||\mathbf{p}_{j,t} - \mathbf{R}_t\tilde{\mathbf{V}}_j\tilde{\mathbf{z}}_t^c - \mathbf{t}_t||^2 \tag{20}$$

$$= \sum_{t} ||\mathbf{p}_t - \mathbf{H}_t\tilde{\mathbf{V}}\tilde{\mathbf{z}}_t^c - \mathbf{T}_t||^2 \tag{21}$$

where $\mathbf{H}_t$ is a $2J \times 3J$ matrix containing J copies of $\mathbf{R}_t$ across the diagonal.

This objective is optimized by coordinate descent iterations applied to subsets of the unknowns. Each of these steps finds the global optimum of the objective function with respect to a specific block of the parameters, while holding the others fixed. Except for the rotation parameters, each update can be solved in closed-form. For example, the update to $\mathbf{t}_t$ is derived by solving $\partial L_{MLE}/\partial \mathbf{t}_t = -2\sum_j(\mathbf{p}_{j,t} - \mathbf{R}_t\tilde{\mathbf{V}}_j\tilde{\mathbf{z}}_t^c - \mathbf{t}_t) = 0$. The updates are as follows:

$$\begin{aligned} \text{vec}(\tilde{\mathbf{V}}_j) &\leftarrow \mathbf{M}^+(\mathbf{p}_{j,1:T} - \mathbf{T}_t) && (22) \\ \tilde{\mathbf{z}}_t^c &\leftarrow (\mathbf{H}_t\tilde{\mathbf{V}})^+(\mathbf{p}_t - \mathbf{T}_t) && (23) \\ \mathbf{t}_t &\leftarrow \frac{1}{J}\sum_j(\mathbf{p}_{j,t} - \mathbf{R}_t\tilde{\mathbf{V}}_j\tilde{\mathbf{z}}_t^c) && (24) \end{aligned}$$

where $\mathbf{p}_{j,1:T} = [\mathbf{p}_{j,1}^T, ..., \mathbf{p}_{j,T}^T]^T$, $\mathbf{M} = [\tilde{\mathbf{z}}_1^c \otimes \mathbf{R}_1^T, ..., \tilde{\mathbf{z}}_T^c \otimes \mathbf{R}_T^T]^T$, $\otimes$ denotes Kronecker product, and the vec operator stacks the entries of a matrix into a vector[4]. The shape basis update is derived by rewriting the objective as:

$$L_{MLE} \propto \sum_j ||\mathbf{p}_{j,1:T} - \mathbf{M}\text{vec}(\tilde{\mathbf{V}}_j) - \mathbf{T}_t||^2 \quad (25)$$

and by solving $\partial L_{MLE}/\partial \text{vec}(\tilde{\mathbf{V}}_j) = 0$.

The camera matrix $\mathbf{R}_t$ is subject to a nonlinear orthonormality constraint, and cannot be updated in closed-form. Instead, we perform a single Gauss-Newton step. First, we parameterize the current estimate of the motion with a 3×3 rotation matrix $\mathbf{Q}_t$, so that $\mathbf{R}_t = \mathbf{\Pi}\mathbf{Q}_t$, where $\mathbf{\Pi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$. We define the updated rotation relative to the previous estimate as: $\mathbf{Q}_t^{new} = \Delta_{Q_t}\mathbf{Q}_t$. The incremental rotation Δ_{Q_t} is parameterized in exponential map coordinates by a three-dimensional vector $\xi_t = [\omega_t^x, \omega_t^y, \omega_t^z]^T$:

$$\Delta_{Q_t} = e^{\hat{\xi}_t} = \mathbf{I} + \hat{\xi}_t + \hat{\xi}_t^2/2! + ... \quad (26)$$

where $\hat{\xi}_t$ denotes the skew-symmetric matrix:

$$\hat{\xi}_t = \begin{bmatrix} 0 & -\omega_t^z & \omega_t^y \\ \omega_t^z & 0 & -\omega_t^x \\ -\omega_t^y & \omega_t^x & 0 \end{bmatrix} \quad (27)$$

[4]For example, vec $\left(\begin{bmatrix} a_0 & a_2 \\ a_1 & a_3 \end{bmatrix}\right) = [a_0, a_1, a_2, a_3]^T$

Dropping nonlinear terms gives the updated value as $\mathbf{Q}_t^{new} = (\mathbf{I} + \hat{\xi}_t)\mathbf{Q}_t$. Substituting $\mathbf{Q}_t^{new}$ into Equation 20 gives:

$$L_{MLE} \propto \sum_{j,t} ||\mathbf{p}_{j,t} - \Pi(\mathbf{I} + \hat{\xi}_t)\mathbf{Q}_t\tilde{\mathbf{V}}_j\tilde{\mathbf{z}}_t^c - \mathbf{t}_t||^2 \tag{28}$$

$$\propto \sum_{j,t} ||\Pi\hat{\xi}_t\mathbf{a}_{j,t} - \mathbf{b}_{j,t}||^2 \tag{29}$$

where $\mathbf{a}_{j,t} = \mathbf{Q}_t\tilde{\mathbf{V}}_j\tilde{\mathbf{z}}_t^c$ and $\mathbf{b}_{j,t} = (\mathbf{p}_{j,t} - \mathbf{R}_t\tilde{\mathbf{V}}_j\tilde{\mathbf{z}}_t^c - \mathbf{t}_t)$. Let $\mathbf{a}_{j,t} = [a_{j,t}^x, a_{j,t}^y, a_{j,t}^z]^T$. Note that we can write the matrix product $\Pi\hat{\xi}_t\mathbf{a}_{j,t}$ directly in terms of the unknown twist vector $\xi_t = [\omega_t^x, \omega_t^y, \omega_t^z]^T$:

$$\Pi\hat{\xi}_t\mathbf{a}_{j,t} = \begin{bmatrix} 0 & -\omega_t^z & \omega_t^y \\ \omega_t^z & 0 & -\omega_t^x \end{bmatrix} \begin{bmatrix} a_{j,t}^x \\ a_{j,t}^y \\ a_{j,t}^z \end{bmatrix} \tag{30}$$

$$= \begin{bmatrix} 0 & a_{j,t}^z & a_{j,t}^y \\ -a_{j,t}^z & 0 & a_{j,t}^x \end{bmatrix} \xi_t. \tag{31}$$

We use this identity to solve for the twist vector ξ_t minimizing Equation 29:

$$\xi_t = \left(\sum_j \mathbf{C}_{j,t}^T\mathbf{C}_{j,t}\right)^{-1} \left(\sum_j \mathbf{C}_{j,t}^T\mathbf{b}_{j,t}\right) \tag{32}$$

where

$$\mathbf{C}_{j,t} = \begin{bmatrix} 0 & a_{j,t}^z & a_{j,t}^y \\ -a_{j,t}^z & 0 & a_{j,t}^x \end{bmatrix} \tag{33}$$

We finally compute the updated rotation as $\mathbf{Q}_t^{new} \leftarrow e^{\hat{\xi}_t}\mathbf{Q}_t$, which is guaranteed to satisfy the orthonormality constraint.

Note that, since each of the parameter updates involves the solution of an overconstrained linear system, BCD-LS can be used even when some of the point tracks are missing. In such event, the optimization is carried out over the available data.

The rigid motion is initialized by the Tomasi-Kanade [26] algorithm; the latent coordinates are initialized randomly.

B. NRSFM with PPCA

We now describe an EM algorithm to estimate the PPCA model from point tracks. The EM algorithm is a standard optimization algorithm for latent variable problems [12]; our derivation

follows closely those for PPCA [22], [25] and factor analysis [14]. Given tracking data $\mathbf{p}_{1:T}$, we seek to estimate the unknowns $\mathbf{G}_{1:T}$, $\mathbf{T}_{1:T}$, $\bar{\mathbf{s}}$, $\mathbf{V}$, and σ^2 (as before, we estimate 2D translations $\mathbf{T}$, due to the depth ambiguity). To simplify the model, we remove one source of noise by assuming $\sigma_m^2 = 0$. The data likelihood is given by:

$$p(\mathbf{p}_{1:T}|\mathbf{G}_{1:T}, \mathbf{T}_{1:T}, \bar{\mathbf{s}}, \mathbf{V}, \sigma^2) = \prod_t p(\mathbf{p}_t|\mathbf{G}_t, \mathbf{T}_t, \bar{\mathbf{s}}, \mathbf{V}, \sigma^2) \tag{34}$$

where the per-frame distribution is Gaussian (Equation 8). Additionally, if there are any missing point tracks, these will also be estimated. The EM algorithm alternates between two steps: in the E-step, a distribution over the latent coordinates $\mathbf{z}_t$ is computed; in the M-step, the other variables are updated[5].

c) E-step.: In the E-step, we compute the posterior distribution over the latent coordinates $\mathbf{z}_t$ given the current parameter estimates, for each time t. Defining $q(\mathbf{z}_t)$ to be this distribution, we have:

$$\begin{aligned} q(\mathbf{z}_t) &= p(\mathbf{z}_t|\mathbf{p}_t, \mathbf{G}_t, \mathbf{T}_t, \bar{\mathbf{s}}, \mathbf{V}, \sigma^2) && (35)\\ &= \mathcal{N}(\mathbf{z}_t|\beta(\mathbf{p}_t - \mathbf{G}_t\bar{\mathbf{s}} - \mathbf{T}_t)); \mathbf{I} - \beta\mathbf{G}_t\mathbf{V}) && (36)\\ \beta &= \mathbf{V}^T\mathbf{G}_t^T(\mathbf{G}_t\mathbf{V}\mathbf{V}^T\mathbf{G}_t^T + \sigma^2\mathbf{I})^{-1} && (37) \end{aligned}$$

The computation of β may be accelerated by the Matrix Inversion Lemma:

$$\beta = \sigma^{-2}\mathbf{I} - \mathbf{G}_t\mathbf{V}(\mathbf{I} + \sigma^{-2}\mathbf{V}^T\mathbf{G}_t^T\mathbf{G}_t\mathbf{V})^{-1}\mathbf{V}^T\mathbf{G}_t^T\sigma^{-4} \tag{38}$$

Given this distribution, we also define the following expectations:

$$\begin{aligned} \mu_t &\equiv E[\mathbf{z}_t] = \beta(\mathbf{p}_t - \mathbf{G}_t\bar{\mathbf{s}} - \mathbf{T}_t) && (39)\\ \phi_t &\equiv E[\mathbf{z}_t\mathbf{z}_t^T] = \mathbf{I} - \beta\mathbf{G}_t\mathbf{V} + \mu_t\mu_t^T && (40) \end{aligned}$$

where the expectation is taken with respect to $q(\mathbf{z}_t)$.

[5]Technically, our algorithm is an instance of the Generalized EM algorithm, since our M-step does not compute a global optimum of the expected log-likelihood.

d) M-step.: In the M-step, we update the motion parameters by minimizing the expected negative log-likelihood:

$$\begin{aligned} Q &\equiv E[-\log p(\mathbf{p}_{1:T}|\mathbf{G}_{1:T}, \mathbf{T}_{1:T}, \bar{\mathbf{s}}, \mathbf{V}, \sigma^2)] && (41) \\ &= \frac{1}{2\sigma^2}\sum_t E[||\mathbf{p}_t - (\mathbf{G}_t(\bar{\mathbf{s}} + \mathbf{V}\mathbf{z}_t) - \mathbf{T}_t)||^2] + JT\log(2\pi\sigma^2) && (42) \end{aligned}$$

This function cannot be minimized in closed-form, but closed-form updates can be computed for each of the individual parameters (except than the camera parameters, discussed below). To make the updates more compact, we define the following additional variables:

$$\begin{aligned} \tilde{\mathbf{V}} &\equiv [\bar{\mathbf{s}}, \mathbf{V}], \quad \tilde{\mathbf{z}}_t \equiv [1, \mathbf{z}_t^T]^T && (43) \\ \tilde{\mu}_t &\equiv [1, \mu_t^T]^T, \quad \tilde{\phi}_t \equiv \begin{bmatrix} 1 & \mu_t^T \\ \mu_t & \phi_t \end{bmatrix} && (44) \end{aligned}$$

The unknowns are then updated as follows; derivations are given in the Appendix.

$$\begin{aligned} \text{vec}(\tilde{\mathbf{V}}) &\leftarrow \left(\sum_t (\tilde{\phi}_t^T \otimes (\mathbf{G}_t^T\mathbf{G}_t))\right)^{-1} \text{vec}\left(\sum_t \mathbf{G}_t^T(\mathbf{p}_t - \mathbf{T}_t)\tilde{\mu}_t^T\right) && (45) \\ \sigma^2 &\leftarrow \frac{1}{2JT}\sum_t \Big(||\mathbf{p}_t - \mathbf{T}_t||^2 - 2(\mathbf{p}_t - \mathbf{T}_t)^T\mathbf{G}_t\tilde{\mathbf{V}}\tilde{\mu}_t + && (46) \\ &\qquad \text{tr}\left(\tilde{\mathbf{V}}^T\mathbf{G}_t^T\mathbf{G}_t\tilde{\mathbf{V}}\tilde{\phi}_t\right)\Big) && (47) \\ c_t &\leftarrow \sum_j \tilde{\mu}_t^T\tilde{\mathbf{V}}_j^T\mathbf{R}_t^T(\mathbf{p}_{j,t} - \mathbf{t}_t) / \sum_j \text{tr}\left(\tilde{\mathbf{V}}_j^T\mathbf{R}_t^T\mathbf{R}_t^T\tilde{\mathbf{V}}_j\tilde{\phi}_t\right) && (48) \\ \mathbf{t}_t &\leftarrow \frac{1}{J}\sum_j \left(\mathbf{p}_{j,t} - c_t\mathbf{R}_t\tilde{\mathbf{V}}_j\tilde{\mu}_t\right) && (49) \end{aligned}$$

The system of equations for the shape basis update is large and sparse, so we compute the shape update using conjugate gradient.

The camera matrix $\mathbf{R}_t$ is subject to a nonlinear orthonormality constraint, and cannot be updated in closed-form. Instead, we perform a single Gauss-Newton step. First, we parameterize the current estimate of the motion with a 3×3 rotation matrix $\mathbf{Q}_t$, so that $\mathbf{R}_t = \mathbf{\Pi}\mathbf{Q}_t$, where $\mathbf{\Pi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$. The update is then:

$$\begin{aligned} \text{vec}(\xi) &\leftarrow \mathbf{A}^+\mathbf{B} && (50) \\ \mathbf{R}_t &\leftarrow \mathbf{\Pi}e^{\xi}\mathbf{Q}_t && (51) \end{aligned}$$

where $\mathbf{A}$ and $\mathbf{B}$ are given in Equations 70 and 71.

If the input data is incomplete, the missing tracks are filled in during the M-step of the algorithm. Let point $\mathbf{p}_{j',t'}$ be one of the missing entries in the 2D tracks. Optimizing the expected log-likelihood with respect to the unobserved point yields the update rule:

$$\mathbf{p}_{j',t'} \quad \leftarrow \quad c_{t'}\mathbf{R}_{t'}\tilde{\mathbf{V}}_{j'}\tilde{\mu}_{t'} + \mathbf{t}_{t'} \tag{52}$$

Once the model is learned, the maximum likelihood 3D shape for frame t is given by $\bar{\mathbf{s}}+\mathbf{V}\mu_t$; in camera coordinates, it is $c_t\mathbf{Q}_t(\bar{\mathbf{s}}+\mathbf{V}\mu_t+D_t)$. (The depth component of $\mathbf{D}_t$ cannot be determined, and thus is set to zero).

e) Initialization: The rigid motion is initialized by the Tomasi-Kanade [26] algorithm. The first component of the shape basis $\mathbf{V}$ is initialized by fitting the residual, using separate shapes $\mathbf{S}_t$ at each time-step (holding the rigid motion fixed), and then applying PCA to these shapes. This process is iterated (i.e., the second component is fit based on the remaining residual, etc.) to produce an initial estimate of the entire basis. We found the algorithm to be likely to converge to a good minimum when σ^2 is forced to remain large in the initial steps of the optimization. For this purpose we scale σ^2 with an annealing parameter that decreases linearly with the iteration count and finishes at 1.

C. NRSFM with Linear Dynamics

The linear dynamical model introduced in Section II-C for NRSFM is a special form of a general Linear Dynamical System (LDS). Shumway and Stoffer [15], [23] describe an EM algorithm for this case, which can be directly adapted to our problem. The sufficient statistics μ_t, ϕ_t, and $E[\mathbf{z}_t\mathbf{z}_{t-1}^T]$ can be computed with Shumway and Stoffer's E-step algorithm, which performs a linear-time Forward-Backward algorithm; the forward step is equivalent to Kalman filtering. In the M-step, we perform the same shape update steps as above; moreover, we update the $\mathbf{\Phi}$ and $\mathbf{Q}$ matrices using Shumway and Stoffer's update equations.

IV. Experiments

We now describe quantitative experiments comparing NRSFM algorithms on both synthetic and real datasets. Here we compare the following models and algorithms:[6]

[6]We are grateful to Brand and to Xiao et al. for providing the source code for their algorithms.

- **BCD-LS:** The least-squares algorithm described in Section III-A.
- **EM-PPCA:** The PPCA model, using the EM algorithm described in Section III-B.
- **EM-LDS:** The LDS model, using the EM algorithm described in Section III-C.
- **XCK:** The closed-form method of Xiao et al. [34].
- **B05:** Brand's "direct" method [6].

We do not consider here the original algorithm of Bregler et al. [7], since we and others have found it to give inferior results to all subsequent methods; we also omit Brand's factorization method [5] from consideration.

To evaluate results, we compare the sum of squared differences between estimated 3D shapes to ground truth depth: $||\hat{\mathbf{s}}^C_{1:T} - \mathbf{s}^C_{1:T}||_F$, measured in the camera coordinate system (i.e., applying the camera rotation, translation, and scale). In order to avoid an absolute depth ambiguity, we subtract out the centroid of each shape before comparing. In order to account for a reflection ambiguity, we repeat the test with the sign of depth inverted ($-Z$ instead of Z) for each instant, and take the smaller error. In the experiments involving noise added to the input data, we perturbed the 2D tracks with additive Gaussian noise. The noise level is plotted as the ratio of the noise variance to the norm of the 2D tracks, i.e., $JT\sigma^2/||\mathbf{p}_{1:T}||_F$. Errors are averaged over 20 runs.

A. Synthetic data

We performed experiments using two synthetic datasets. The first is a dataset created by Xiao et al. [34], containing six rigid points (arranged in the shape of a cube) and three linearly-deforming points, without noise. As reported previously, the XCK and B05 algorithms yield the exact shape with zero error in the absence of measurement noise. In contrast, the other methods (EM-PPCA, EM-LDS) have some error; this is to be expected, since the use of a prior model or regularizer can add bias into estimation. Additionally, we found that EM-PPCA and EM-LDS did not obtain good results in this case unless initialized by XCK. For this particular dataset, the methods of XCK and B05 are clearly superior; this is the only dataset on which Xiao et al. [34] perform quantitative comparisons between methods. However, this dataset is rather artificial, due to the absence of noise and the simplicity of the data. If we introduce measurement noise (Figure 2), EM-PPCA and EM-LDS give the best results for small amounts of noise, when initialized with XCK (this is the only example in this paper in which we used XCK for initialization).

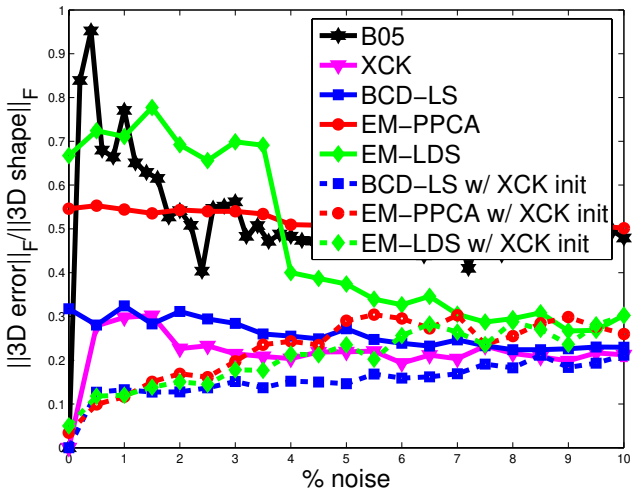

Fig. 2. Reconstruction error as a function of measurement noise for the cube-and-points data of [34].

Our second synthetic dataset is a 3D animation of a shark, consisting of 3D points. The object undergoes rigid motion and deformation corresponding to $K = 2$ basis shapes; no noise is added. Reconstruction results are shown in Figure 3, and errors plotted in Figure 4, the iterative methods (BCD-LS, EM-PPCA, and EM-LDS) perform significantly better than B05 and XCK. The ground-truth shape basis is degenerate (i.e., individual elements of the deformation are not full rank when viewed as $J \times 3$ matrices), a case that Xiao et al. [34] point to as being problematic (we have not tested their solution to this problem). Performance for BCD-LS gets significantly worse as superfluous degrees of freedom are added ($K > 2$), whereas EM-PPCA and EM-LDS are relatively robust to choice of K; this suggests that BCD-LS is more sensitive to overfitting with large K. EM-LDS performs slightly better than EM-PPCA, most likely because the very simple deformations of the shark are well-modeled by linear dynamics.

In order to test the ability of EM-PPCA and EM-LDS to estimate noise variance (σ^2), we compare the actual with estimated variances in Figure 5. The estimation is generally very accurate, and error variance across the multiple runs is very small (generally less than 0.04). This illustrates an advantage of these methods: they can automatically learn many of the parameters that would otherwise need to be set "by hand."

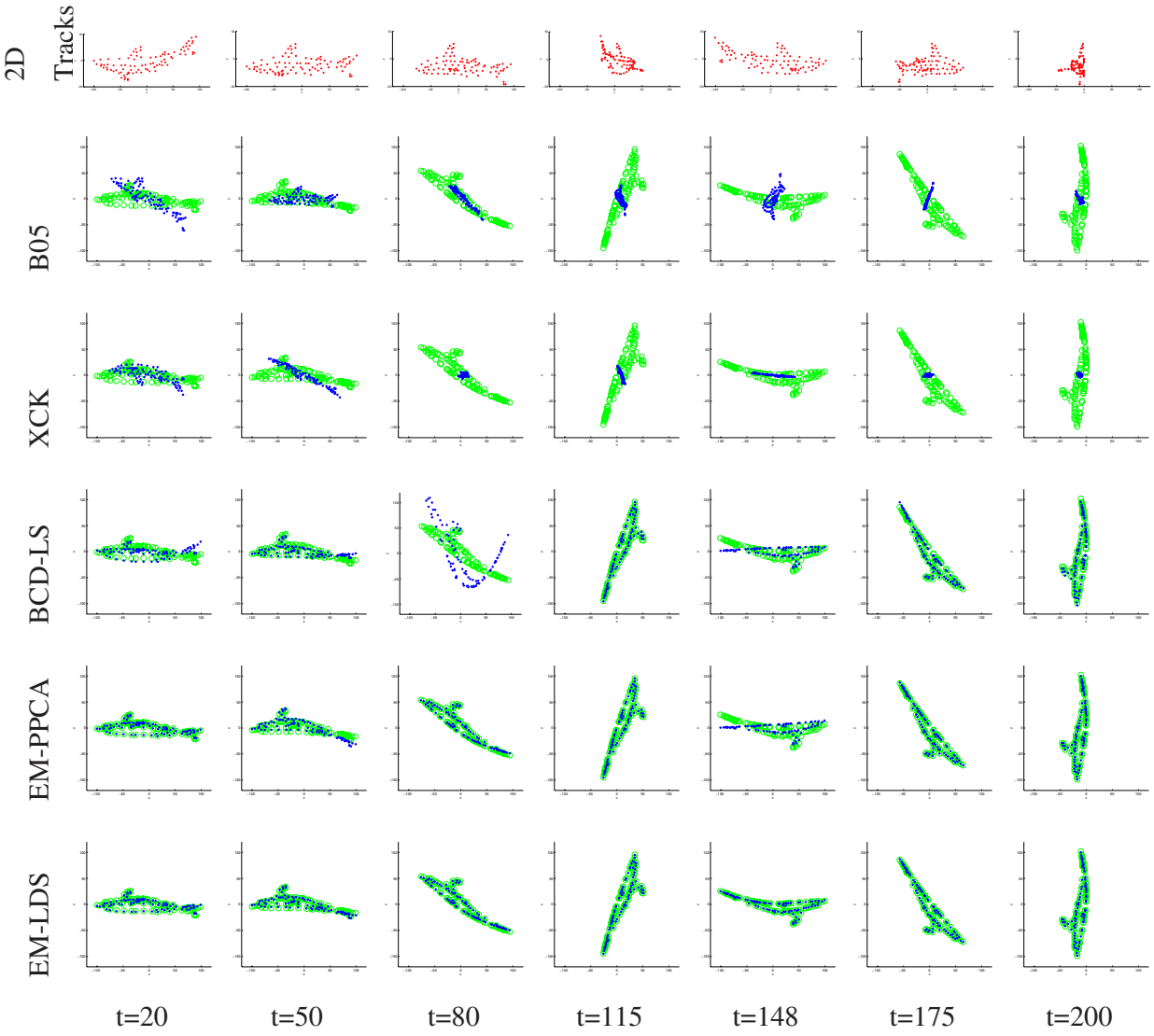

Fig. 3. Reconstructions of the shark sequence using the five algorithms. Each algorithm was given 2D tracks as inputs; reconstructions are shown here from a different viewpoint than the inputs to the algorithm. Ground-truth features are shown as green circles; reconstructions are blue dots.

B. Motion capture data

We performed experiments with two motion capture sequences. The first sequence was obtained with a Vicon optical motion capture system, with 40 markers attached to the subject's face (Figure 6). The motion capture systems tracks the markers and triangulates to estimate the 3D position of all markers. We subsampled the data to 15 Hz, yielding a sequence 316 frames long. The subject performed a range of facial expressions and dialogue. Test data is generated

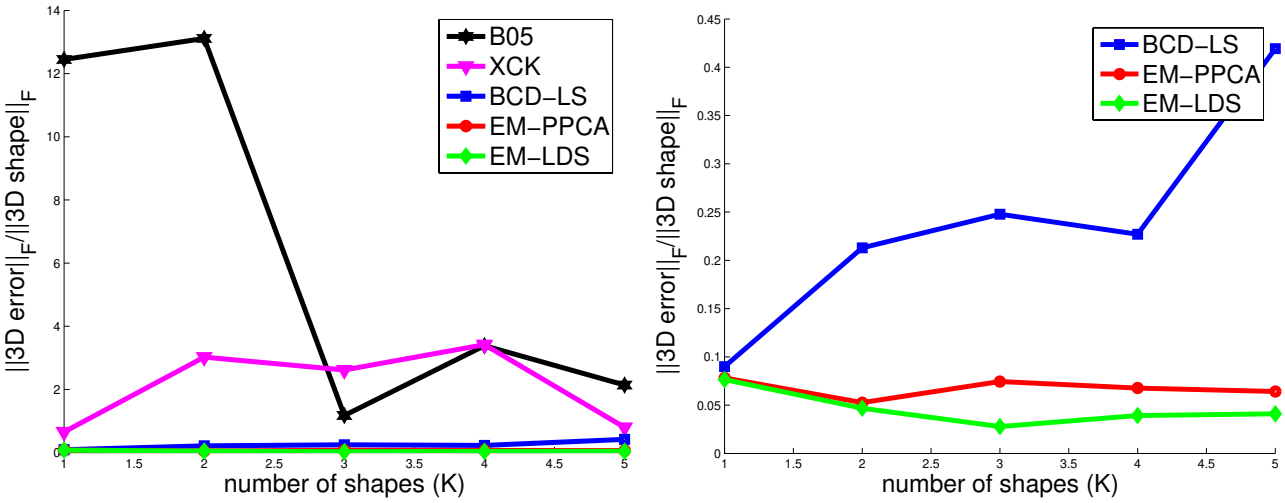

Fig. 4. Reconstruction error as a function of the number of basis shapes (K), for the synthetic shark data. The ground truth shape has $K = 2$. The plot on the left compares all methods discussed here, and the plot on the right compares only our methods.

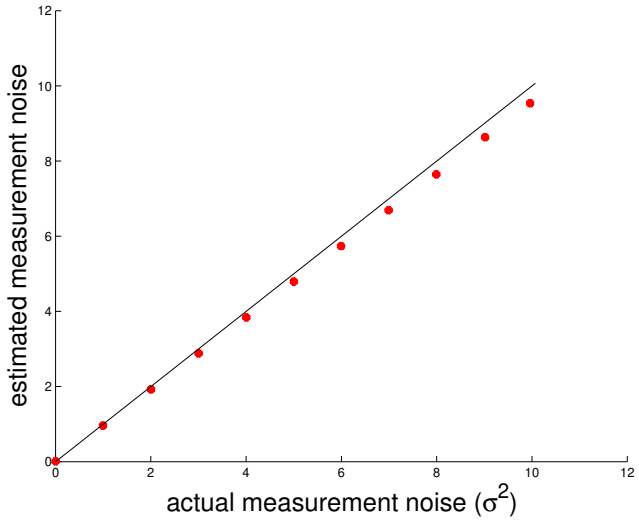

Fig. 5. Noise estimation for the synthetic shark dataset. For each true noise variance (X-axis), the variance estimated by our algorithm is shown on the Y-axis. The diagonal line corresponds to ground truth. Results are averaged over 20 runs. Error bars are not shown because the sample variance is very small.

Fig. 6. *Left:* The facial motion capture session, which provided test data for this paper. *Right:* The full-body motion capture session (from the CMU mocap database) used in this paper.

by orthographic projection.

Reconstruction results are shown in Figure 7, and reconstruction error plotted in Figure 8. As is visible in the figure, the XCK and B05 algorithms both yield unsatisfactory reconstructions,[7] regardless of the choice of K, whereas the iterative methods (BCD-LS, EM-PPCA, EM-LDS) perform significantly better. EM-PPCA yields the best results on the original data. The performance of BCD-LS degrades as K increases, suggesting an overfitting effect, whereas EM-PPCA only improves with larger K. We also performed this test with EM-PPCA using a pure orthographic projection model ($c_t \equiv 1$), and the error curve was very similar to that of scaled orthographic projection.

We tested a MAP version of the algorithm that optimizes L_{MAP} (Equation 13) plus a penalty on the Frobenius norm of $\mathbf{V}$ by block coordinate descent. We found this method to give worse results than the least-squares optimization (i.e., optimizing L_{MLE} by BCD-LS), for all regularization weights that we tested. This suggests that selecting appropriate regularization is not trivial.

We also performed experiments with noise added to the data, and with random tracks removed. The missing data case is important to test, because 3D points will necessarily become occluded in real image sequences, and may also "disappear" for other reasons, such as dropped tracks or specularities. We simulate missing data by omitting each measurement uniformly at random with a fixed probability. (In real situations, occlusions typically occur in a much more structured manner [8]). Figure 9 demonstrates the sensitivity of the different iterative estimation methods to missing data; these figures suggest that EM-PPCA and EM-LDS are more robust to missing

[7]One possible explanation would be that this data suffers from degenerate bases; however, this did not appear to be the case, as we determined by testing the PCA bases of the aligned ground truth data.

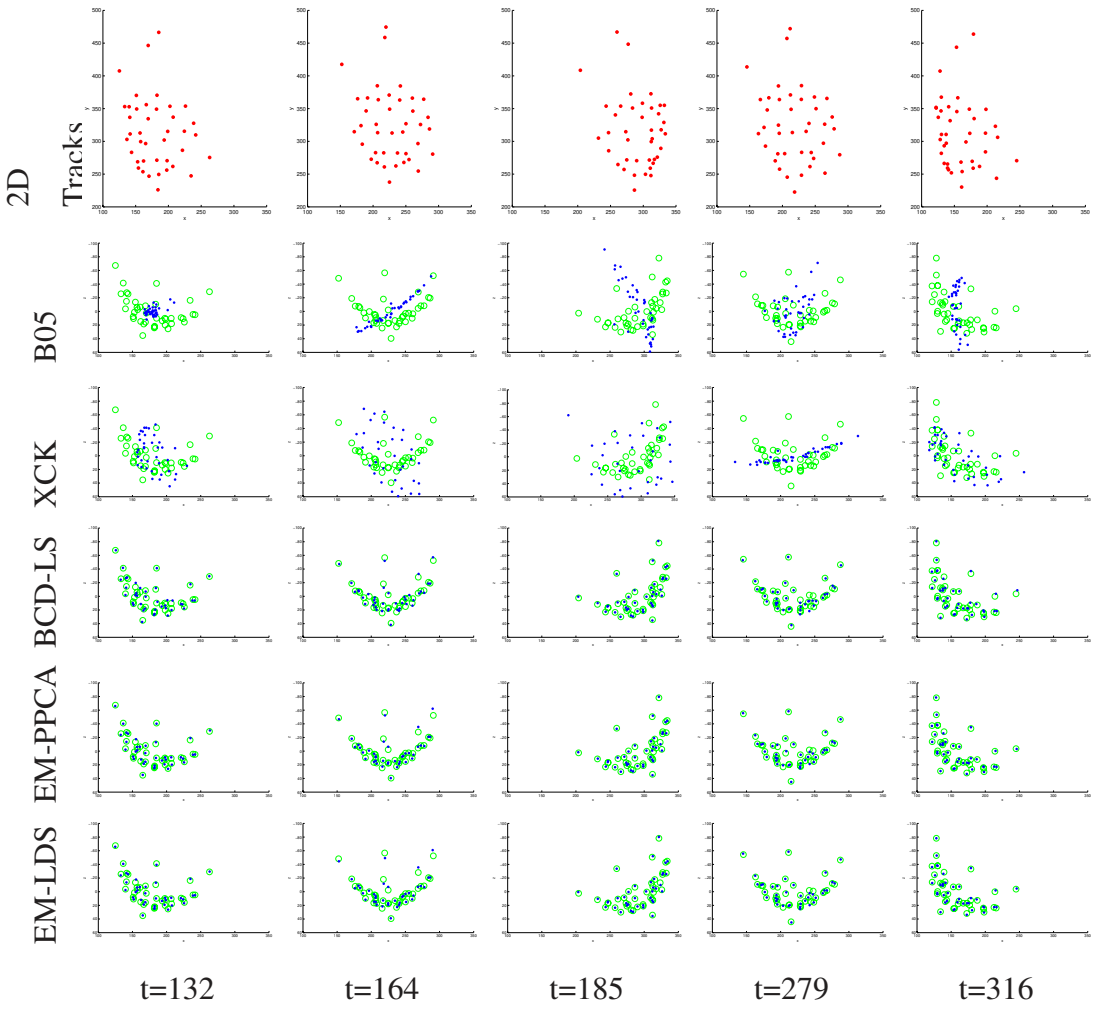

Fig. 7. Reconstruction of the facial motion capture data. The top row shows selected frames from the input data. The remaining rows show reconstruction results (blue dots), together with ground truth (green circles), viewed from below.

data, whereas BCD-LS degrades much faster. These results are computed by averaging over 30 random runs. Again, EM-LDS performs best as the amount of missing data increases. We did not test XCK and B05 on these datasets, as these methods assume that no data is missing, and will therefore depend on how this missing data is imputed in the initial factorization step.

In order to visualize the model learned by EM-PPCA, Figure 10 shows the mean shape and the modes of deformation learned with $K = 2$.

We additionally tested the algorithms' sensitivity to the size of the data set (Figure 11). Tests were conducted by sampling the face sequence at different temporal rates. (Due to local minima

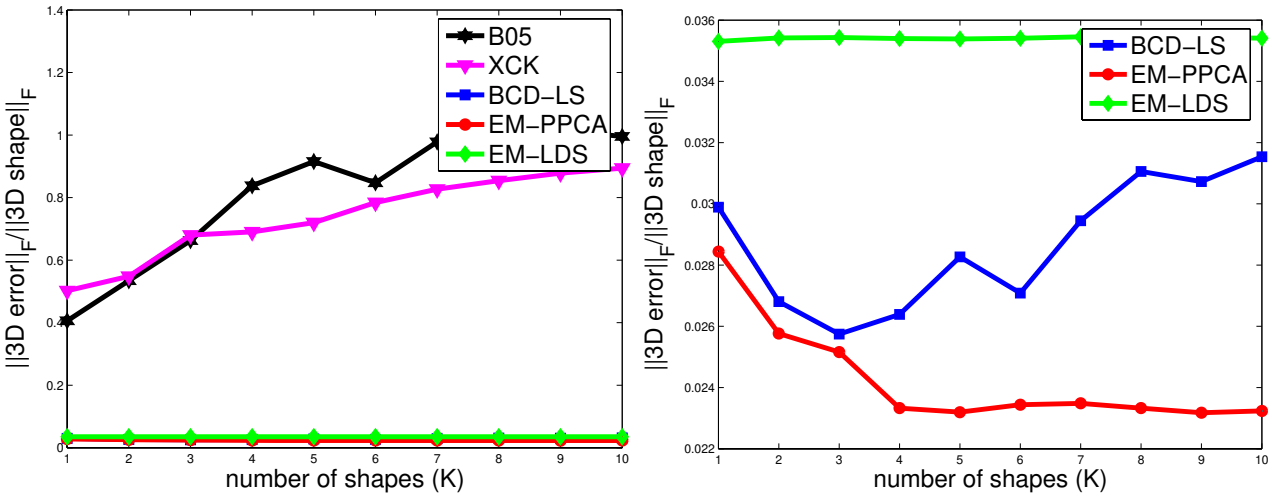

Fig. 8. Reconstruction error for the face motion capture data, varying the number of basis shapes (K) used in the reconstruction.

issues with BCD-LS, we performed 30 multiple restarts for each BCD-LS test.) We found that dataset size did not have a significant effect on the performance of the algorithm; surprisingly, reconstruction error increased somewhat with larger datasets. We suspect that this reflects non-stationarity in the data, e.g., some frames having significantly greater variations than others, or non-Gaussian behavior. We also performed the same experiment on synthetic data randomly generated from a linear-subspace model, and found the behavior to be much more as predicted, with error monotonically decreasing as the data set grew, and then levelling off.

In some applications of NRSFM, there may be significant structure in the deformations that are not represented in the model, or not known in advance. In order to explore this case, we performed experiments on full-body human motion capture data of a person walking. This human body can be approximately modeled as an articulated, rigid-body system. The articulation is not modeled by the NRSFM methods considered here, and we cannot expect perfect results from this data. However, if the simple NRSFM models work well in this case, they may provide useful initialization for an algorithm that attemps to determine the articulation structure or the kinematics. We chose walking data that includes turning (Figure 6),[8] in order to ensure adequate rotation of the body; as in rigid SFM, without rotation, there is inadequate

[8]We used sequence 16-18 from the CMU motion capture database (http://mocap.cs.cmu.edu). The sequence was subsampled by discarding every other frame, and most of the markers. The resulting data has 260 frames and 55 points per frame.

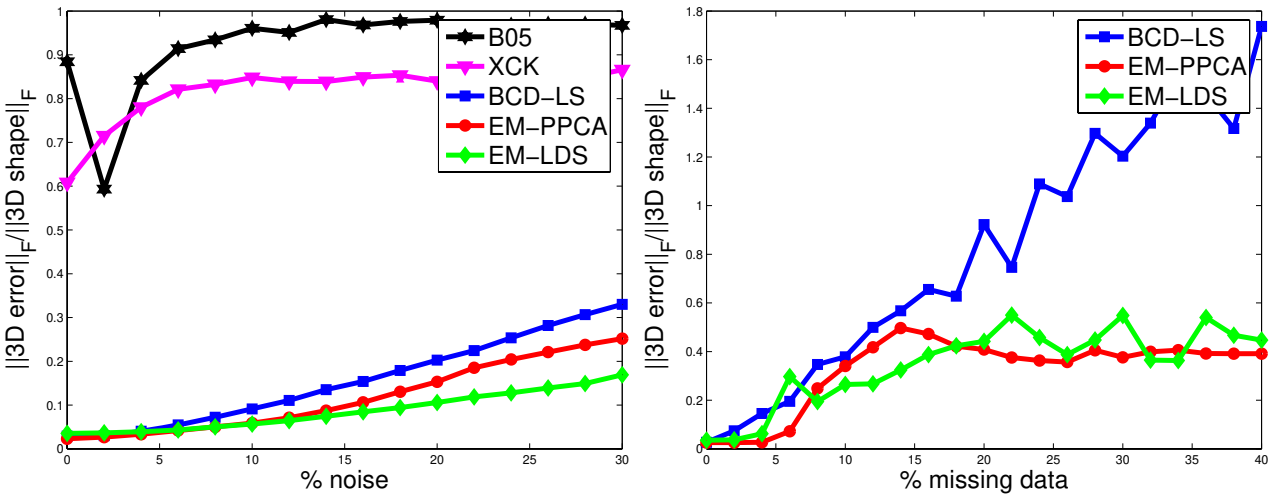

Fig. 9. Reconstruction error for the face motion capture data. The left plot shows the dependence on added measurement noise, the right plot shows increasing amounts of missing data. Note that "0%" noise corresponds to zero noise added to the data, in addition to any noise already present in the measurements.

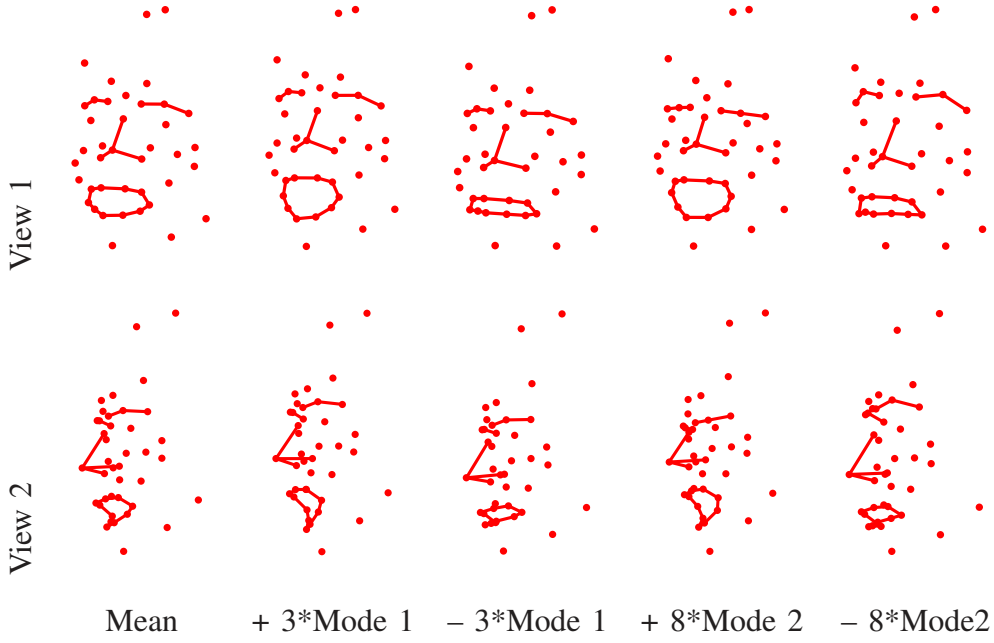

Fig. 10. 3D mean shape and modes recovered by EM-PPCA with $K = 2$. Shape modes are generated by adding each deformation vector (scaled) to the mean shape. The lines are not part of the model; they are shown for visualization purposes only.

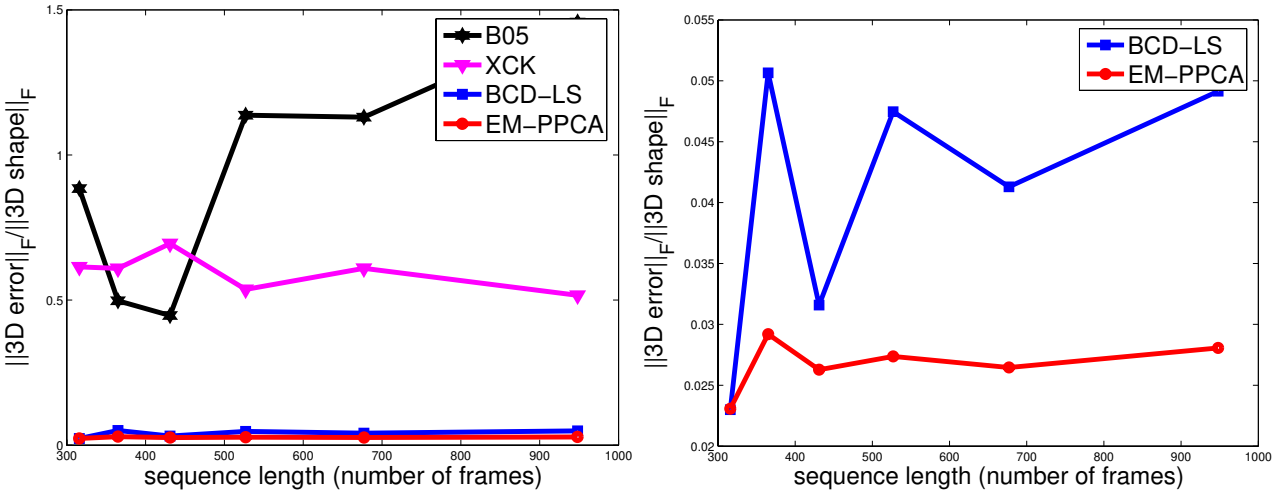

Fig. 11. Dependence on data set size for the face data. We suspect that the odd behavior of the plots is due to non-stationarity of the facial motion data; some frames are fit much better by the model than others.

information to estimate shape. The input to the algorithm is orthographic projection of 3D marker measurements. Reconstructions are shown in Figure 12. As plotted in Figure 13, all of the algorithms exhibit non-trivial reconstruction error. However, EM-PPCA gives the best results, with BCD-LS somewhat worse; XCK and B05 both yield very large errors. Additionally, B05 exhibits significant sensitivity to the choice of the number of basis shapes (K), and, as before, the reconstruction from BCD-LS degrades slowly as K grows, whereas EM-PPCA is very robust to the choice of K.

V. Discussion and Future Work

In this work, we have introduced non-rigid structure-from-motion (NRSFM). Due to the inevitable presence of measurement noise, missing data and high-dimensional spaces, we argue that NRSFM is best posed as a statistical estimation problem. This allows us to build explicit generative models of shape, to marginalize out hidden parameters and to use prior knowledge effectively. As shown by our experiments, closed-form methods — while obtaining perfect results on noiseless synthetic data — yield much higher errors on noisy data and real measurements. The superiority of EM-PPCA to BCD-LS in all of our tests illustrates the importance of marginalizing out latent coordinates. The superiority of EM-LDS over EM-PPCA for highly noisy real data illustrates the value of the use of a motion model, although a first-order linear dynamical model

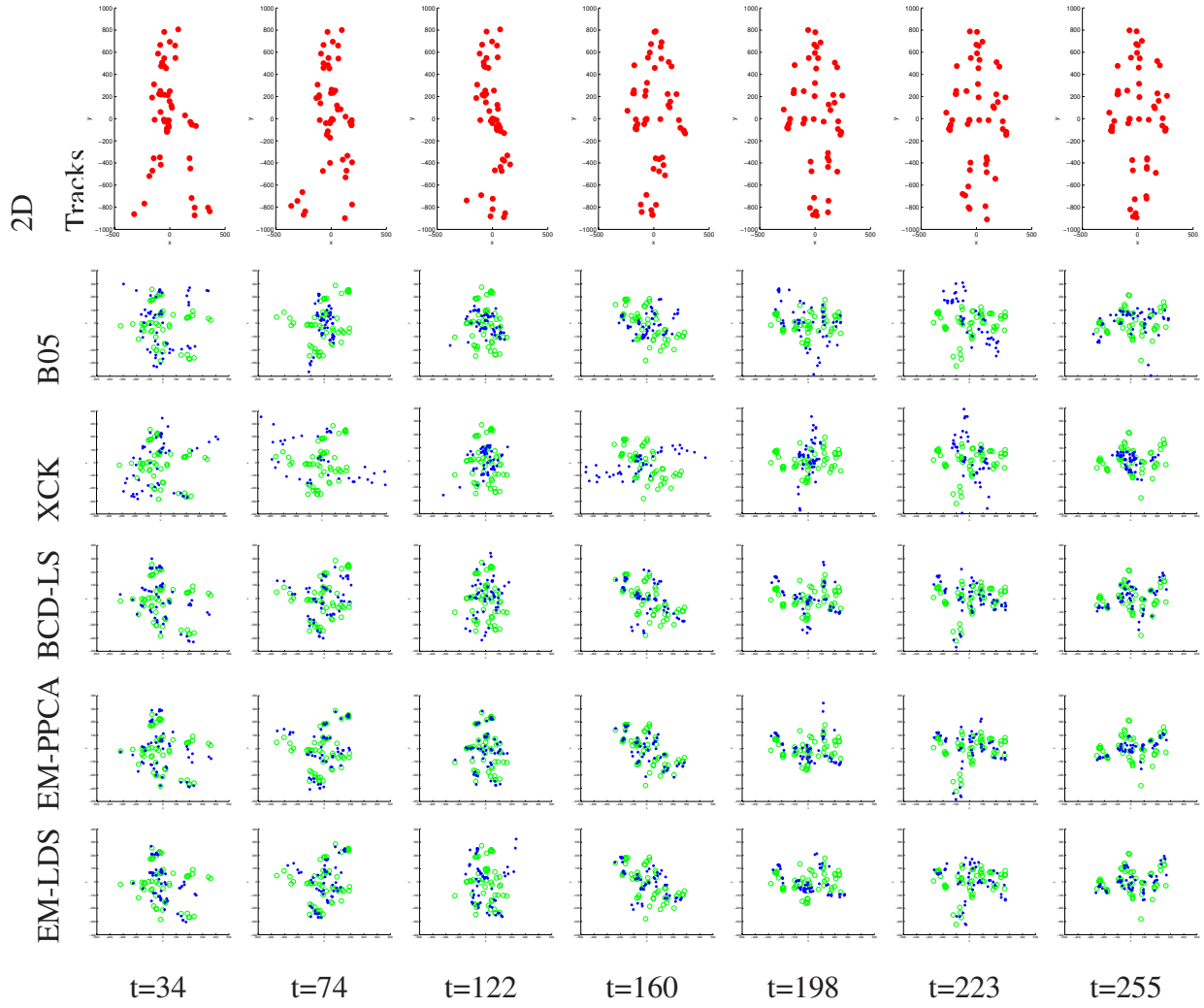

Fig. 12. Reconstruction of the walking motion capture data. The top row shows selected frames from the input data. The remaining rows show reconstruction results (blue dots), together with ground truth (green circles), viewed from below.

was too weak for our datasets.

We did find that, on synthetic, noiseless data, our methods had issues with local minima, whereas the closed-form methods performed very well on these cases. This indicates that testing on pure synthetic data, while informative, cannot replace quantitative testing on real data, and may in fact give opposite results from real data. The cube-and-points dataset is one for which our prior distribution may not be appropriate.

Linear models provide only a limited representation of shape and motion, and there is significant work to be done in determining more effective models. For example, nonlinear time-

series models (e.g., [21], [33]) can represent temporal dependencies more effectively; perspective projection is a more realistic camera model for many image sequences. However, we believe that, whatever the model, the basic principles of statistical estimation should be applied. For example, NRSFM for articulated rigid-body models will likely benefit from marginalizing over joint angles. We do not address the selection of K in this paper, although our results suggest that the methods are not extremely sensitive to this choice. Alternatively, methods such as Variational PCA [3] could be adapted in order to estimate K or integrate it out.

Another important direction is the integration of NRSFM with image data. An advantage of the statistical estimation framework is that it can be directly tied to an appearance model [27], whereas other methods must somehow extract reliable tracks without the benefit of 3D reconstruction.

Although we have chosen to use the EM algorithm for estimation, it is possible that other numerical optimization methods will give better results. For example, conjugate gradient could be applied directly to the log-posterior.

VI. Acknowledgements

Earlier versions of this work appeared in [7], [28], [29]. Portions of this work were conducted while LT was at Stanford University, New York University, and Riya, Inc., while AH was at New York University and University of Washington, and while CB was at Stanford University. We thank Matthew Brand and Jing Xiao for providing their source code, Jacky Bibliowicz for providing the face mocap data, and the CMU Motion Capture Database for the full-body data. We thank Gene Alexander, Hrishikesh Deshpande, and Danny Yang for participating in earlier versions of these projects. Thanks to Stefano Soatto for discussing shape deformation. This work was supported in part by ONR grant N00014-01-1-0890, NSF grants IIS-0113007, 0303360, 0329098, 0325715, the University of Washington Animation Research Labs, the Alfred P. Sloan Foundation, a Microsoft Research New Faculty Fellowship, the National Sciences and Engineering Research Council of Canada, the Canada Foundation for Innovation, and the Ontario Ministry of Research and Innovation.

In memory of Henning Biermann.

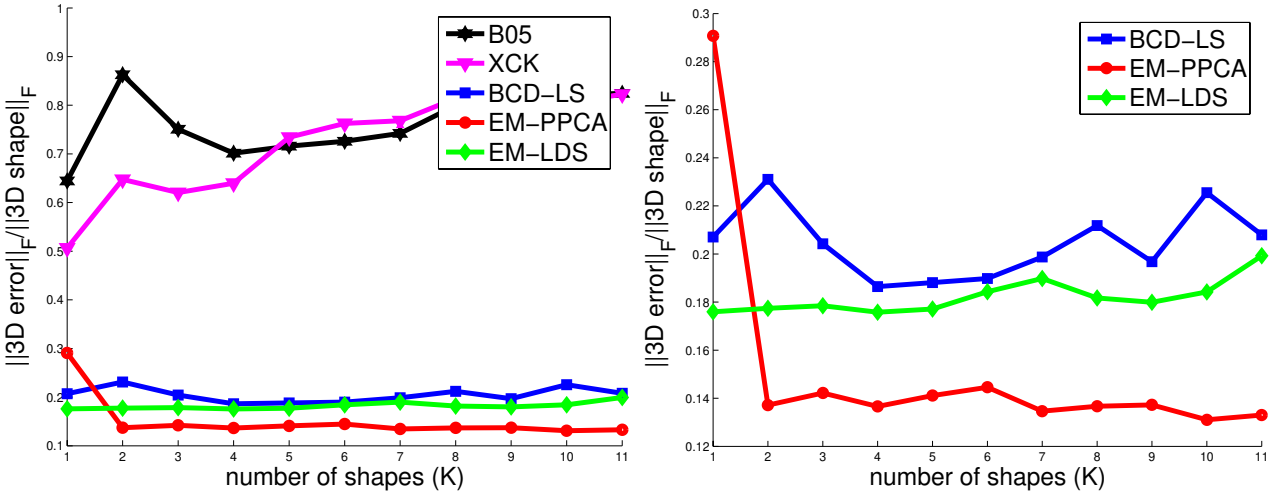

Fig. 13. Reconstruction error as a function of basis shapes (K) for full-body motion capture data. This noise is added in addition to any noise already present in the measurements.

REFERENCES

[1] J. Barbič and D. James. Real-Time Subspace Integration for St. Venant-Kirchhoff deformable Models. *ACM Trans. on Graphics*, 24(3):982–990, Aug. 2005.

[2] B. Bascle and A. Blake. Separability of pose and expression in facial tracking animation. In *Proc. ICCV*, pages 323–328, Jan. 1998.

[3] C. M. Bishop. Variational Principal Components. In *Proc. ICANN*, volume 1, pages 509–514, 1999.

[4] V. Blanz and T. Vetter. A Morphable Model for the Synthesis of 3D Faces. In *Proceedings of SIGGRAPH 99*, Computer Graphics Proceedings, pages 187–194, Aug. 1999.

[5] M. Brand. Morphable 3D models from video. In *Proc. CVPR*, volume 2, pages 456–463, 2001.

[6] M. Brand. A Direct Method for 3D Factorization of Nonrigid Motion Observed in 2D. In *Proc. CVPR*, volume 2, pages 122–128, 2005.

[7] C. Bregler, A. Hertzmann, and H. Biermann. Recovering Non-Rigid 3D Shape from Image Streams. In *Proc. CVPR*, pages 690–696, 2000.

[8] A. M. Buchanan and A. W. Fitzgibbon. Damped Newton Algorithms for Matrix Factorization with Missing Data. In *Proc. CVPR*, volume 2, pages 316–322, 2005.

[9] T. F. Cootes and C. J. Taylor. Statistical models of appearance for medical image analysis and computer vision. In *Proc. SPIE Medical Imaging*, 2001.

[10] J. P. Costeira and T. Kanade. A Multibody Factorization Method for Independently Moving Objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[11] F. Dellaert, S. M. Seitz, C. E. Thorpe, and S. Thrun. EM, MCMC, and Chain Flipping for Structure from Motion with Unknown Correspondence. *Machine Learning*, 50(1–2):45–71, 2003.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Statistical Society series B*, 39:1–38, 1977.

[13] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. CRC Press, 2nd edition, 2003.

[14] Z. Ghahramani and G. E. Hinton. The EM Algorithm for Mixtures of Factor Analyzers. Technical Report CRG-TR-96-1, University of Toronto, 1996.

[15] Z. Ghahramani and G. E. Hinton. Parameter estimation for linear dynamical systems. Technical Report CRG-TR-96-2, University of Toronto, 1996.

[16] M. Han and T. Kanade. Multiple Motion Scene Reconstruction from Uncalibrated Views. In *Proc. ICCV*, volume 1, pages 163–170, July 2001.

[17] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2003.

[18] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 14:201–211, 1973.

[19] D. J. C. MacKay. Probable Networks and Plausible Predictions - A Review of Practical Bayesian Methods for Supervised Neural Networks. *Network: CNS*, 6:469–505, 1995.

[20] F. I. Parke. Computer generated animation of faces. In *ACM'72: Proceedings of the ACM annual conference*, pages 451–457, 1972.

[21] V. Pavlović, J. M. Rehg, and J. MacCormick. Learning Switching Linear Models of Human Motion. In *Advances in Neural Information Processing Systems 13*, pages 981–987, 2001.

[22] S. T. Roweis. EM algorithms for PCA and SPCA. In *Proc. NIPS 10*, pages 626–632, 1998.

[23] R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *J. Time Series Analysis*, 3(4):253–264, 1982.

[24] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *J. Opt. Soc. Am. A*, 4(3):519–524, Mar. 1987.

[25] M. E. Tipping and C. M. Bishop. Probabilistic Principal Components Analysis. *J. Royal Statistical Society, Series B*, 61(3):611–622, 1999.

[26] 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.

[27] L. Torresani and A. Hertzmann. Automatic Non-Rigid 3D Modeling from Video. In *Proc. ECCV*, pages 299–312, 2004.

[28] L. Torresani, A. Hertzmann, and C. Bregler. Learning Non-Rigid 3D Shape from 2D Motion. In *Proc. NIPS 16*, pages 1555–1562, 2004.

[29] L. Torresani, D. Yang, G. Alexander, and C. Bregler. Tracking and Modeling Non-Rigid Objects with Rank Constraints. In *Proc. CVPR*, pages 493–500, 2001.

[30] N. F. Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *J. of Vision*, 2(5):371–387, 2002.

[31] M. Turk and A. Pentland. Eigenfaces for recognition. *J. of Cognitive Neuroscience*, 3(1):71–86, 1991.

[32] S. Ullman. Maximizing rigidity: the incremental recovery of 3-D structure from rigid and nonrigid motion. *Perception*, 13(3):255–274, 1984.

[33] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian Process Dynamical Models. In *Proc. NIPS 18*, pages 1441–1448, 2006.

[34] J. Xiao, J. Chai, and T. Kanade. A Closed-Form Solution to Non-Rigid Shape and Motion Recovery. *Int. J. of Computer Vision*, 67(2):233–246, 2006.

[35] A. J. Yezzi and S. Soatto. Deformotion: Deforming Motion, Shape Averages, and the Joint Registration and Approximation of Structures in Images. In *Int. J. Computer Vision*, volume 53, pages 153–167, 2003.

APPENDIX

We now derive the M-step step updates used in Section III-B. The expected negative log-likelihood is:

$$Q = \frac{1}{2\sigma^2}\sum_t E[||\mathbf{p}_t - (\mathbf{G}_t\tilde{\mathbf{V}}\tilde{\mathbf{z}} + \mathbf{T}_t)||^2] + JT\log(2\pi\sigma^2) \quad (53)$$

To derive updates, we solve for the minimizing value of Q with respect to each of the unknowns, holding the others fixed. Closed-form updates exist for each of the individual unknowns, aside from the rotation matrices.

To derive the shape basis $\tilde{\mathbf{V}}$ update, we solve for the stationary point:

$$\frac{\partial Q}{\partial \tilde{\mathbf{V}}} = -\frac{1}{2\sigma^2}\sum_t E[\mathbf{G}_t^T(\mathbf{p}_t - (\mathbf{G}_t\tilde{\mathbf{V}}\tilde{\mathbf{z}} + \mathbf{T}_t))\tilde{\mathbf{z}}^T] \quad (54)$$

$$= -\frac{1}{2\sigma^2}\sum_t \mathbf{G}_t^T(\mathbf{p}_t - \mathbf{T}_t)\tilde{\mu}_t^T + \frac{1}{2\sigma^2}\sum_t \mathbf{G}_t^T\mathbf{G}_t\tilde{\mathbf{V}}\tilde{\phi}_t \quad (55)$$

Applying the vec operator to both sides, and using the identities $\text{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A})\text{vec}(\mathbf{B})$ and $\text{vec}\frac{\partial Q}{\partial \tilde{\mathbf{V}}} = \frac{\partial Q}{\partial \text{vec}(\tilde{\mathbf{V}})}$ gives:

$$\frac{\partial Q}{\partial \text{vec}(\tilde{\mathbf{V}})} = -\frac{1}{2\sigma^2}\text{vec}\left(\sum_t \mathbf{G}_t^T(\mathbf{p}_t - \mathbf{T}_t)\tilde{\mu}_t^T\right) \quad (56)$$

$$+\frac{1}{2\sigma^2}\sum_t(\tilde{\phi}_t^T \otimes (\mathbf{G}_t^T\mathbf{G}_t))\text{vec}(\tilde{\mathbf{V}}) \quad (57)$$

Solving $\frac{\partial Q}{\partial \text{vec}(\tilde{\mathbf{V}})} = 0$ yields the shape basis update (Equation 45).

To solve for the variance update, we can solve $\partial Q/\partial\sigma^2 = 0$ and then simplify

$$\sigma^2 = \frac{1}{2JT}\sum_t E[||\mathbf{p}_t - (\mathbf{G}_t\tilde{\mathbf{V}}\tilde{\mathbf{z}}_t + \mathbf{T}_t)||^2] \quad (58)$$

$$= \frac{1}{2JT}\sum_t \Big(||\mathbf{p}_t - \mathbf{T}_t||^2 - 2(\mathbf{p}_t - \mathbf{T}_t)^T\mathbf{G}_t\tilde{\mathbf{V}}\tilde{\mu}_t \quad (59)$$

$$+ E[\tilde{\mathbf{z}}_t^T\tilde{\mathbf{V}}^T\mathbf{G}_t^T\mathbf{G}_t\tilde{\mathbf{V}}\tilde{\mathbf{z}}_t]\Big) \quad (60)$$

The final term in this expression is a scalar, and so we can apply a trace, and, using the identity $\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA})$, get: $E[\tilde{\mathbf{z}}_t^T\tilde{\mathbf{V}}^T\mathbf{G}_t^T\mathbf{G}_t\tilde{\mathbf{V}}\tilde{\mathbf{z}}_t] = \text{tr}(\tilde{\mathbf{V}}^T\mathbf{G}_t^T\mathbf{G}_t\tilde{\mathbf{V}}E[\tilde{\mathbf{z}}_t\tilde{\mathbf{z}}_t^T]) = \text{tr}(\tilde{\mathbf{V}}^T\mathbf{G}_t^T\mathbf{G}_t\tilde{\mathbf{V}}\tilde{\phi}_t)$.

To solve for the camera updates, we first rewrite the objective function using Equation 1 and, for brevity, drop the dependence on σ^2:

$$Q = \sum_{j,t} E[||\mathbf{p}_{j,t} - (c_t\mathbf{R}_t\tilde{\mathbf{V}}_j\tilde{\mathbf{z}}_t + \mathbf{t}_t)||^2] \quad (61)$$

where $\tilde{\mathbf{V}}_j$ are the rows of $\tilde{\mathbf{V}}$ corresponding to the j-th point (i.e., rows $3j-2$ through $3j$), and $\mathbf{t}_t$ are the x and y components of the translation in image space. The partial for translation is:

$$\frac{\partial Q}{\partial \mathbf{t}_t} = -\sum_{j,t} 2E[(\mathbf{p}_{j,t} - (c_t \mathbf{R}_t \tilde{\mathbf{V}}_j \tilde{\mathbf{z}}_t + \mathbf{t}_t))] \tag{62}$$

$$= -2\sum_{j}(\mathbf{p}_{j,t} - c_t \mathbf{R}_t \tilde{\mathbf{V}}_j \tilde{\mu}_t) + 2J\mathbf{t}_t \tag{63}$$

The update to c_t is derived as follows:

$$\frac{\partial Q}{\partial c_t} = \sum_{j} E[-2\tilde{\mathbf{z}}_t^T \tilde{\mathbf{V}}_j^T \mathbf{R}_t^T (\mathbf{p}_{j,t} - (c_t \mathbf{R}_t \tilde{\mathbf{V}}_j \tilde{\mathbf{z}}_t + \mathbf{t}_t))] \tag{64}$$

$$= -2\sum_{j} \tilde{\mu}_t^T \tilde{\mathbf{V}}_j^T \mathbf{R}_t^T (\mathbf{p}_{j,t} - \mathbf{t}_t) + 2c_t \sum_{j} \text{tr}\left(\tilde{\mathbf{V}}_j^T \mathbf{R}_t^T \mathbf{R}_t^T \tilde{\mathbf{V}}_j \tilde{\phi}_t\right) \tag{65}$$

The camera rotation is subject to a orthonormality constraint, for which we cannot derive a closed-form update. Instead, we derive the following approximate update. First, we differentiate Equation 61:

$$\frac{\partial Q}{\partial \mathbf{R}_t} = \mathbf{R}_t c_t^2 \sum_{j} \tilde{\mathbf{V}}_j \tilde{\phi}_t \tilde{\mathbf{V}}_j^T - c_t \sum_{j} (\mathbf{p}_{j,t} - \mathbf{t}_t) \tilde{\mu}_t^T \tilde{\mathbf{V}}_j^T \tag{66}$$

Since we cannot obtain a closed-form solution to $\partial Q / \partial \mathbf{R}_t = 0$, we linearize the rotation. We parameterize the current rotation as a 3×3 rotation matrix, such that $\mathbf{R}_t = \mathbf{\Pi}\mathbf{Q}_t$, parameterize the updated rotation relative to the previous estimate: $\mathbf{Q}_t^{new} = \Delta_Q \mathbf{Q}_t$. The incremental rotation Δ_Q is parameterized by an exponential map with twist matrix ξ:

$$\Delta_Q = e^{\xi} = \mathbf{I} + \xi + \xi^2/2! + ... \tag{67}$$

Dropping nonlinear terms gives the updated value as $\mathbf{Q}_t^{new} = (\mathbf{I} + \xi)\mathbf{Q}_t$. Substituting $\mathbf{Q}_t^{new}$ into Eq. 66 gives:

$$\frac{\partial Q}{\partial \mathbf{R}_t} \approx \Pi(\mathbf{I} + \xi)\mathbf{Q}_t c_t^2 \sum_{j} \tilde{\mathbf{V}}_j \tilde{\phi}_t \tilde{\mathbf{V}}_j^T - c_t \sum_{j} (\mathbf{p}_{j,t} - \mathbf{t}_t) \tilde{\mu}_t^T \tilde{\mathbf{V}}_j^T \tag{68}$$

Applying the vec operator gives:

$$\text{vec}\frac{\partial Q}{\partial \mathbf{R}_t} \approx \mathbf{A}\text{vec}(\xi) + \mathbf{B} \tag{69}$$

$$\mathbf{A} = \left(c_t^2 \sum_{j} \tilde{\mathbf{V}}_j \tilde{\phi}_t \tilde{\mathbf{V}}_j^T \mathbf{Q}_t^T\right) \otimes \Pi \tag{70}$$

$$\mathbf{B} = c_t^2 \Pi \mathbf{Q}_t \sum_{j} \tilde{\mathbf{V}}_j \tilde{\phi}_t \tilde{\mathbf{V}}_j^T - c_t \sum_{j} (\mathbf{p}_{j,t} - \mathbf{t}_t) \tilde{\mu}_t^T \tilde{\mathbf{V}}_j^T \tag{71}$$

We minimize $||\mathbf{A}\text{vec}(\xi) + \mathbf{B}||_F$ with respect to ξ for the update, giving $\text{vec}(\xi) \leftarrow \mathbf{A}^+\mathbf{B}$.

Lorenzo Torresani graduated in computer science from the University of Milan, Italy, in 1996. He received the M.S. and Ph.D. degrees in computer science from Stanford University in 2001 and 2005, respectively. He is an associate researcher at Microsoft Research Cambridge and a research assistant professor of computer science at Dartmouth College. His previous professional experience includes working as a researcher at Riya, Inc., at the Courant Institute of New York University, and at Digital Persona, Inc. He was the recipient of the best student paper prize at the IEEE Conference On Computer Vision and Pattern Recognition, 2001. His main research interests are in computer vision, machine learning, and computer animation.

Aaron Hertzmann is an Assistant Professor of Computer Science at University of Toronto. He received a BA in Computer Science and Art & Art History from Rice University in 1996, and an MS and PhD in Computer Science from New York University in 1998 and 2001, respectively. In the past, he has worked at University of Washington, Microsoft Research, Mitsubishi Electric Research Lab, Interval Research Corporation and NEC Research Institute. He serves as an Associate Editor for IEEE Transactions on Visualization and Computer Graphics, served as an Area Coordinator for SIGGRAPH 2007, and co-chaired NPAR 2004. His awards include an MIT TR100 (2004), a Ontario Early Researcher Award (2005), a Sloan Foundation Fellowship (2006), and a Microsoft New Faculty Fellowship (2007). His research interests include computer vision, computer graphics, and machine learning.

Chris Bregler received his M.S. and Ph.D. in Computer Science from U.C. Berkeley in 1995 and 1998 and his Diplom from Karlsruhe University in 1993. He is currently an Associate Professor of Computer Science at NYU's Courant Institute. Prior to NYU he was on the faculty at Stanford University and worked for several companies including Hewlett Packard, Interval, and Disney Feature Animation. He received the Olympus Prize for achievements in computer vision and AI in 2002. He was named Stanford Joyce Faculty Fellow and Terman Fellow in 1999, and Sloan Research Fellow in 2003. He was the chair for the SIGGRAPH 2004 Electronic Theater and Computer Animation Festival.