Bilinear Spatiotemporal Basis Models

Ijaz Akhter[*,‡], Tomas Simon[†,‡], Sohaib Khan[*], Iain Matthews[‡,†], and Yaser Sheikh[†]
[*]LUMS School of Science and Engineering, [†]Carnegie Mellon University, [‡]Disney Research Pittsburgh

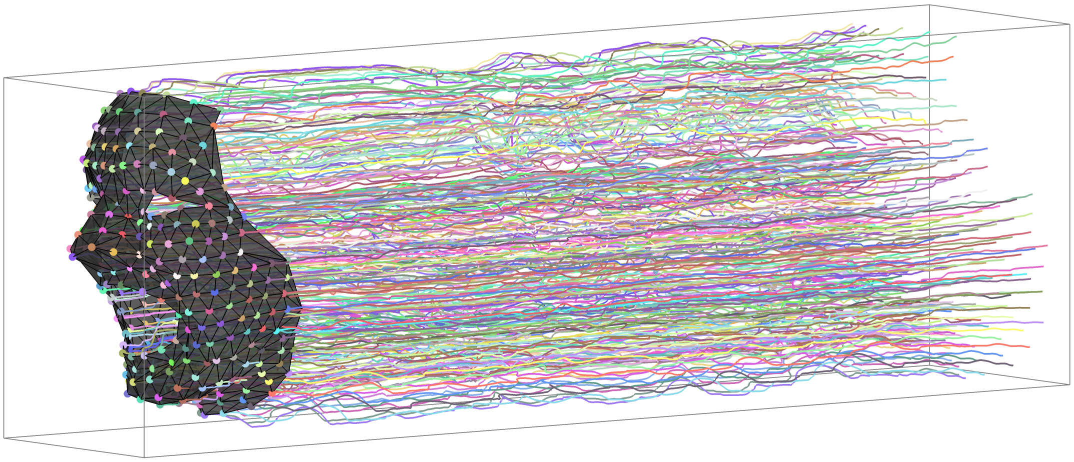


Fig. 1. We present a compact model of spatiotemporal data with the ability to generalize to unseen data and to accurately predict missing data even when it is learned from a single training sequence. This figure illustrates the model being used within an expectation maximization routine to *simultaneously* estimate labels for all points in all frames of an unlabeled time-varying point cloud.

A variety of objects, such as faces, bodies, and cloth, are represented in computer graphics as a collection of moving spatial landmarks. Spatiotemporal data is used in a number of graphics applications including performance animation, character modeling, and video-based tracking. The principal modes of variation in the spatial geometry of objects are typically modeled using dimensionality reduction techniques, while concurrently, trajectory representations like splines and autoregressive models are widely used to exploit the temporal regularity of deformation. In this paper, we present the bilinear spatiotemporal basis as a model that simultaneously exploits spatial and temporal regularity while maintaining the ability to generalize well to new sequences. The model can be interpreted as representing the data as a linear combination of spatiotemporal sequences consisting of shape modes oscillating over time at key frequencies. This factorization allows the use of analytical, predefined functions to represent temporal variation (e.g., B-Splines or the Discrete Cosine Transform) resulting in more efficient model estimation. We apply the model to natural spatiotemporal data, including face, body, and cloth data, and compare it in terms of compaction, generalization ability, predictive precision, and efficiency to existing models. We demonstrate the direct application of the model for a number of graphics applications including labeling, gap-filling, de-noising, and motion touch-up.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Animation*; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Physically based modeling*

Additional Key Words and Phrases: bilinear models, spatiotemporal, motion data

1. INTRODUCTION

We present a compact and generalizable model of time-varying spatial data that can simultaneously capture the spatial and the temporal correlations inherent in the data while remaining computationally efficient in its requirements of training data and memory. Time-varying spatial data is widely used to represent animated characters in computer games, marker data in motion capture, and surface meshes in physical simulators. A variety of analysis tasks are performed on this type of data such as performance animation [Chai and Hodgins 2005], gap-filling [Liu and McMillan 2006], motion editing [Gleicher 2001], correspondence [Wand et al. 2007], and compression [Arikan 2006]. In theory, many of these tasks are highly under-constrained, and estimation algorithms for these tasks

Email: `akhter@lums.edu.pk, tsimon@andrew.cmu.edu`

DOI 10.1145/XXXXXXX.YYYYYYY
http://doi.acm.org/10.1145/XXXXXXX.YYYYYYY

exploit the natural regularity that exists as a cloud of points moves over time.

Spatial regularity, i.e., the correlation between the spatial locations of nearby points, has been captured using dimensionality reduction techniques in several contexts, including statistical shape modeling [Cootes et al. 1995], cloth animation [de Aguiar et al. 2010], face animation [Li et al. 2010], and nonrigid structure from motion [Bregler et al. 2000]. Typically, each instantaneous shape is represented as a linear combination of a compact set of basis shapes. Temporal regularity, i.e., the correlation between the location of a point at two successive time instances, has also been used to compactly represent the motion of a point using autoregressive models, via splines [Gleicher 1998], and through dimensionality reduction over trajectories [Torresani and Bregler 2002; Akhter et al. 2008]. When dimensionality reduction is used to capture the principal modes of variation of the shape geometry, the correlation between temporally successive points is ignored; when it is performed on trajectories, the correlation between spatially adjacent trajectories is ignored. Discarding these correlations leads to an over-parameterization of the data and ignores relationships that are useful for performing analysis tasks.

To utilize spatiotemporal regularity, a number of researchers have proposed directly learning a linear basis that spans the space of fixed-duration spatiotemporal sequences [Hamarneh and Gustavsson 2004; Min et al. 2009]. As the joint dimensionality of such a linear basis is high, a large amount of training data is required to avoid learning spatiotemporal correlations that are sequence specific. Filtering approaches, such as Kalman or particle filters, are also popular approaches to analyze time-varying spatial data [Thrun et al. 2006]. They represent the instantaneous configuration of spatial data by state variables and represent the temporal variation in terms of a dynamical function that relates the state at a time instant in terms of the state at the preceding time instant (or instances). Linear dynamical models are widely used and extensions to nonlinear dynamical models (e.g., [Wang et al. 2008]) have emerged. Due to the incremental form of dynamical models, they are inherently online models that are designed to produce estimates of the state variables as data arrives sequentially.

In this paper, we present a new model of time-varying spatial data as a linear combination of spatiotemporal sequences, each of which may be intuitively interpreted as shape modes oscillating over time at key frequencies. We demonstrate that such a model can be expressed in a simple bilinear form, which separately but simultaneously exploits both the spatial and the temporal regularities that exist in data. The separation between the spatial and the temporal modes, allows us to leverage analytical trajectory bases to condition the model such as the discrete cosine transform (DCT) and B-splines. Such conditioning allows the model to generalize well to sequences of arbitrary length from a small number of training sequences while remaining tractable and highly compact. We analyze the form thoroughly, providing bounds on reconstruction error and experimental validation on four measures of performance: compaction, generalization ability, computational efficiency, and predictive precision. Using these measures we compare our model to linear dynamical models, shape basis models, splines, trajectory basis models, and linear spatiotemporal basis models. We demonstrate the broad applicability of the model by directly embedded it in standard algorithms, such as expectation maximization, to perform a number of useful analysis tasks such as labeling (Figure 1), de-noising, gap-filling, and editing for face, body, and cloth motion data.

2. RELATED WORK

The representation of time-varying spatial data is a well-studied problem in computer graphics, computer vision, and applied mathematics; an overview of representation and analysis techniques has been covered by Bronstein and colleagues [2008]. A widely used approach due to its simplicity and effectiveness is to represent the data as a compact linear combination of basis vectors. Principal Component Analysis (PCA) or a similar dimensionality reduction technique is used on a training set to determine the most significant modes of deformation, and data samples are then described as a linear combination of these modes. This general approach has subsequently been extended using nonlinear dimensionality reduction techniques, in particular through the use of kernel methods in Kernel PCA [Schölkopf et al. 1997] and Gaussian Process Latent Variable Models (GPLVMs) [Lawrence 2004]. For spatial data, the linear model is commonly called a point distribution model and was established through the work of Mardia and Dryden [1989], Le and Kendall [1993], and Cootes and colleagues [1995]. For temporal data, dimensionality reduction has also been applied to learn a compact linear basis of trajectories [Sidenbladh et al. 2000; Torresani and Bregler 2002; Akhter et al. 2008].

Linear models that jointly span both space and time have been used to track shapes deforming over time and to describe their principal modes of spatiotemporal variation [Hamarneh and Gustavsson 2004], for registration in both space and time [Perperidis et al. 2004], for spatiotemporal segmentation [Mitchell et al. 2002], for motion synthesis [Urtasun et al. 2004; Min et al. 2009], and for denoising [Lou and Chai 2010]. Typically, joint spatiotemporal models are a direct application of linear dimensionality reduction where each spatiotemporal sequence is vectorized and represents one sample. These models are often specific to the particular sequence length that was fixed during training and correlations between points at different space-time locations are explicitly learned. These are most prominent in periodic motions. To generalize beyond specific spatiotemporal correlations, joint linear spatiotemporal models therefore require a large training set, as we show in this paper.

Multilinear methods have been used in a number of computer graphics applications, including expression retargeting [Chuang and Bregler 2005; Vlasic et al. 2005], approximating multi-array visual data [Wang et al. 2005], factoring temporal variations from time-lapse videos [Sunkavalli et al. 2007], and representing textures [Vasilescu and Terzopoulos 2004]. In particular, bilinear models have been applied to separate style and content by Tenenbaum and Freeman [2000] and has been applied to cardiac data by Hoogendoorn and colleagues [2009]. The principal difference is that while Tenenbaum and Freeman's symmetric model factors the *coefficients* into bilinear style and content terms which are combined by a shared mixing basis, our model factors the *basis* into spatial and temporal variations and unifies the coefficients. That is, the Tenenbaum and Freeman approach computes bilinear factorizations of the coefficients of each sample, while our model is linear in coefficients and a bilinear factorization of the basis. From a practical perspective, this switch allows for least squares estimation of the coefficients rather than requiring nonlinear iterative minimization. When conditioned using a predefined trajectory basis, our approach also allows a closed form solution for model estimation. From a conceptual perspective, our form, when conditioned using DCT, encodes a spatiotemporal sequence as a linear combination of spatial modes of variation oscillating at key frequencies. This inter-

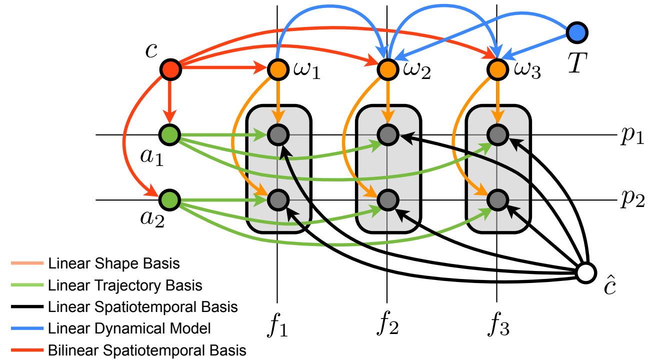


Fig. 2. Graphical model showing parameter dependencies for various models of time-varying spatial data. In this figure, p_i refers to a particular point index, f_i refers to a particular time frame, ω_i is a shape coefficient at time f_i, and a_i is a trajectory coefficient associated with p_i. $\hat{c}$ is a coefficient of a linear spatiotemporal basis, and c is a coefficient of the bilinear spatiotemporal basis. $\mathbf{T}$ refers to the transition matrix of a linear dynamical system.

pretation of our DCT-conditioned bilinear model also finds empirical support in studies of receptive field dynamics [DeAngelis et al. 1995] and biological motion [Troje 2002; Sigal et al. 2010], where a similar sine-wave decomposition of the PCA modes was shown to capture the most prominent features of human gaits. In Troje's original approach, a truncated Fourier-related transform of the PCA modes was shown to retain the most relevant information for gait classification, analysis, and synthesis.

Time-varying spatial data has also been modeled as a dynamical system, where a fixed rule describes transitions across time [Thrun et al. 2006]. Compared to basis representations, dynamical systems model the evolution of a process as transitions between time-steps, making them especially attractive for online processing. Conversely, because it is a model of the process rather than a direct model of the data, operations affecting the entire sequence are usually more costly. Li and colleagues [2009] model marker trajectories as a Linear Dynamical System (LDS) to infer missing markers. Nonlinear dynamical systems have also been successfully applied to motion data, most notably Gaussian Process Dynamical Models (GPDMs) [Wang et al. 2008], which have been shown to be an excellent model for synthesis and inference. The main drawback of Gaussian process models is significant computational and memory cost, making them impractical for very large datasets. Inference is usually iterative in the case of missing data. Model estimation is costly as well, and typically accomplished using a nonlinear optimization (expectation-maximization) that requires adequate initialization. In comparison, these operations have efficient closed form solutions for our DCT-conditioned bilinear model.

Spatiotemporal models have been widely applied in analyzing, editing, synthesizing, compressing, and denoising of time-varying spatial data. For motion capture data in particular, missing markers, occlusions, and broken trajectories are often significant issues, and spatiotemporal models are used to infer marker data across long occlusions and during dropouts. For full-body motion capture applications, the models used often constrain spatial variation using an articulated skeletal model [Herda et al. 2001; Hornung et al. 2005], or bone-length constraints incorporated into an LDS framework [Li et al. 2010]. The focus of this work is on spatial data where an articulated model is not appropriate, such as dense facial motion capture, where most of the motion is due to non-rigid deformations. A data driven approach is that of Liu and McMillan [2006], who learn piece-wise linear models from large datasets of motion capture examples for inference. Other work in motion capture of skin deformations includes Park and Hodgins [2006] and Anguelov and colleagues [2005] who use sparse motion capture supplemented by a detailed skin model to label and impute missing data. Current practice in the industry is for animation houses to use motion capture clean-up professionals who create a marker-set, label the points, reconstruct the missing points, and finally retouch the data, smoothing out noisy points.

Spatiotemporal representations have also been considered for the tasks of motion editing and motion adaptation. Methods related to spacetime constraints [Witkin and Kass 1988] aim to globally modify the character motion to meet certain requirements; these methods commonly aim to produce physically-realistic motions by minimizing an energy function. Most approaches focus on carefully formulating the optimization function to enforce the characteristics of spatiotemporal data and not on the representation itself. Typically, the representation is based on keyframe interpolation of joint angles (for articulated characters) or rig parameters (for facial animation). For body motion, a related approach is the *per-frame inverse kinematics + filtering* method of Gleicher [2001], who offers an extensive review of this method and related techniques. Other approaches to motion transformation use dimensionality reduction in the configuration space of the character to constrain the motion optimization process [Safonova et al. 2004; de Aguiar et al. 2010], or to match the parameters of an animation rig [Lewis and Anjyo 2010].

3. METHOD

The time-varying structure of a set of P points sampled at F time instances can be represented as a concatenated sequence of 3D points:

$$\mathbf{S}_{F\times 3P} = \begin{bmatrix} \mathbf{X}_1^1 & \dots & \mathbf{X}_P^1 \\ \vdots & & \vdots \\ \mathbf{X}_1^F & \dots & \mathbf{X}_P^F \end{bmatrix}, \tag{1}$$

where $\mathbf{X}_j^i = \left[X_j^i, Y_j^i, Z_j^i\right]$ denotes the 3D coordinates of the j-th point at the i-th time instance[1], denoted by one of the gray nodes in Figure 2. Thus, the time-varying structure matrix $\mathbf{S}$ contains $3FP$ parameters. This representation of the structure is an over-parametrization because it does not take into account the high degree of regularity generally exhibited by motion data.

One way to exploit the regularity in spatiotemporal data is to represent the 3D shape at each time instance as a linear combination of a small number of *shape basis* vectors $\mathbf{b}_j$ weighted by coefficients ω_j^i [Cootes et al. 1995; Bregler et al. 2000][2],

$$\mathbf{s}^i = \sum_j \omega_j^i \mathbf{b}_j^T. \tag{2}$$

[1] As a matter of standard notation, we indicate row-index as superscript and column-index as subscript.

[2] The rigid component of deformation is typically compensated for using Procrustes analysis [Dryden and Mardia 1998]. For clarity of exposition, we do not include the transformation explicitly in our development.

Thus, the complete structure matrix, $\mathbf{S}$, which is a row-wise concatenation of F 3D shapes, can be represented as

$$\mathbf{S} = \mathbf{\Omega}\mathbf{B}^T, \tag{3}$$

where $\mathbf{B}$ is a $3P \times K_s$ matrix containing K_s shape basis vectors, each representing a 3D structure of length $3P$, and $\mathbf{\Omega}$ is an $F \times K_s$ matrix containing the corresponding shape coefficients ω_j^i, shown as orange nodes in Figure 2 representing all points at a particular time frame. The number of shape basis vectors used to represent a particular instance of motion data is $K_s \leq \min\{F, 3P\}$.

An alternate representation of the time-varying structure is to model it in the trajectory subspace, as a linear combination of *trajectory basis* vectors θ_i [Torresani and Bregler 2002; Akhter et al. 2008],

$$\mathbf{s}_j = \sum_i a_i^j \theta_i \,, \tag{4}$$

where a_i^j is the coefficient weighing each trajectory basis vector (denoted by a green node in Figure 2 representing a particular point across all frames). In this case, the structure matrix $\mathbf{S}$ may be considered as the column-wise concatenation of P 3D trajectories, as

$$\mathbf{S} = \mathbf{\Theta}\mathbf{A}^T, \tag{5}$$

where $\mathbf{\Theta}$ is an $F \times K_t$ matrix containing K_t trajectory basis as its columns, and $\mathbf{A}$ is a $3P \times K_t$ matrix of trajectory coefficients. Here, $K_t \leq \min\{F, 3P\}$ is the number of trajectory basis vectors spanning the trajectory subspace. Note that if orthonormal bases are used in both representations, then $\mathbf{B}^T\mathbf{B} = \mathbf{I}_{K_s \times K_s}$ and $\mathbf{\Theta}^T\mathbf{\Theta} = \mathbf{I}_{K_t \times K_t}$, because the basis vectors are arranged along the columns of $\mathbf{B}$ and $\mathbf{\Theta}$.

3.1 Bilinear Spatiotemporal Basis

The key insight of this paper is the observation that using a shape basis or a trajectory basis independently fails to exploit the full range of generalizable spatiotemporal regularities. In the shape basis representation, the temporal regularity of trajectories is ignored; removing temporal regularity by shuffling the frames in time to a random arrangement only results in a corresponding shuffling of the coefficients. The same is true for the trajectory basis representation, in which case each spatial location is treated independently; hence, their spatial ordering becomes immaterial. Thus, both representations are over-parameterizations because they do not capitalize on either the spatial or the temporal regularity.

This paper presents a bilinear representation of $\mathbf{S}$ linking both shape and trajectory bases in a single model.

Theorem 1: If $\mathbf{S}$ can be expressed exactly as $\mathbf{S} = \mathbf{\Omega}\mathbf{B}^T$ and $\mathbf{S} = \mathbf{\Theta}\mathbf{A}^T$, then there exists a factorization,

$$\mathbf{S} = \mathbf{\Theta}\mathbf{C}\mathbf{B}^T, \tag{6}$$

where $\mathbf{C} = \mathbf{\Theta}^T\mathbf{\Omega} = \mathbf{A}^T\mathbf{B}$ is a $K_t \times K_s$ matrix of spatiotemporal coefficients[3].

Proof: Equating the two forms of $\mathbf{S}$ in Equations 3 and 5 yields $\mathbf{\Omega}\mathbf{B}^T = \mathbf{\Theta}\mathbf{A}^T$. It follows that $\mathbf{A}^T = \mathbf{\Theta}^T\mathbf{\Omega}\mathbf{B}^T$. Substituting this into Equation 5 yields $\mathbf{S} = \mathbf{\Theta}\mathbf{\Theta}^T\mathbf{\Omega}\mathbf{B}$. If we define $\mathbf{C} = \mathbf{\Theta}^T\mathbf{\Omega}$, then we have $\mathbf{S} = \mathbf{\Theta}\mathbf{C}\mathbf{B}^T$. The same result can be derived in a dual fashion by substituting $\mathbf{\Omega} = \mathbf{\Theta}\mathbf{A}^T\mathbf{B}$ in Equation 3 and yielding $\mathbf{C} = \mathbf{A}^T\mathbf{B}$. □

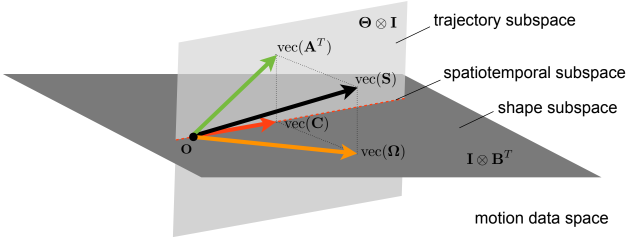


Fig. 3. Illustration of the relationship of the spatiotemporal coefficients $\mathbf{C}$ with shape and trajectory coefficients $\mathbf{\Omega}$ and $\mathbf{A}$ respectively. Spatiotemporal coefficients may be interpreted as either the projection of shape coefficients into trajectory space or as projection of trajectory coefficients into shape space.

Equation 6 describes the *bilinear spatiotemporal* basis, which contains both shape and trajectory bases linked together by a common set of coefficients. These coefficients can be visualized in two equivalent ways as indicated by the two definitions of $\mathbf{C}$ given above: (1) $\mathbf{C} = \mathbf{\Theta}^T\mathbf{\Omega}$ implies the projection of shape coefficients $\mathbf{\Omega}$ onto the trajectory basis, $\mathbf{\Theta}$, and (2) $\mathbf{C} = \mathbf{A}^T\mathbf{B}$ implies the projection of trajectory coefficients $\mathbf{A}$ onto the shape basis $\mathbf{B}$. This dual interpretation of $\mathbf{C}$ is illustrated in Figure 3.

For an intuitive understanding of the bilinear spatiotemporal model, consider the coefficient c_j^i at the i-th row and the j-th column in $\mathbf{C}$ (denoted by the red node in Figure 2). This coefficient represents the weight of the outer product of the i-th trajectory basis vector, θ_i, and the j-th shape basis vector, $\mathbf{b}_j$. This outer product will result in a time-varying structure sequence in which all points of a single shape mode (as defined by the j-th shape basis) will vary over time (as defined by the i-th trajectory basis). The sum of all such outer products $\theta_i\mathbf{b}_j^T$, weighted by the corresponding coefficient, c_j^i, results in the bilinear representation of $\mathbf{S}$, equivalent to Equation 6:

$$\mathbf{S} = \sum_i \sum_j c_j^i \theta_i \mathbf{b}_j^T. \tag{7}$$

This is best illustrated as an animation of each shape basis vector $\mathbf{b}_j$ modulated over time according to each trajectory basis vector θ_i , as shown in the accompanying video (in the vignette titled 'Bilinear Spatiotemporal Modes'). Under our bilinear basis model, spatiotemporal data is represented as linear combination of each of these modulated spatiotemporal sequences.

3.2 Bounds on Reconstruction Error

In Theorem 1, the bilinear spatiotemporal model is derived for the case of perfect representation of time-varying structure. We can also use the bilinear basis (Equation 6) with a reduced number of basis vectors. In the following theorem, we describe bounds on the bilinear spatiotemporal model error as a function of approximation errors of the shape and trajectory models.

Theorem 2 If the reconstruction error of the trajectory model is $\epsilon_t = \|\mathbf{S} - \mathbf{\Theta}\mathbf{A}^T\|_F$, and the error of the shape model is $\epsilon_s = \|\mathbf{S} - \mathbf{\Omega}\mathbf{B}^T\|_F$, then the error of the bilinear spatiotemporal model

[3] For clarity, this result is stated and proven assuming orthogonal bases. An equivalent proof for non-orthogonal bases can be derived by using the pseudo-inverses of $\mathbf{\Theta}$ and $\mathbf{B}$ instead of transposes.

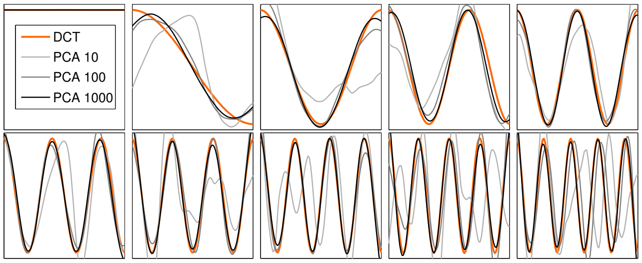


Fig. 4. Ordered left-to-right, top-to-bottom, comparison of the first 10 DCT basis vectors (orange) with the first 10 data-specific PCA trajectory basis vectors learned on a varying number of facial motion capture training sequences: 10 sequences (light gray), 100 sequences (dark gray), and 1000 sequences (black). Each sequence and each vector depicted here is 100 frames in length. For large training sets, the PCA-learned trajectory basis approaches DCT.

$\epsilon = \|\mathbf{S} - \mathbf{\Theta}\mathbf{C}\mathbf{B}^T\|_F$ is upper bounded by $\epsilon_t + \epsilon_s$ and lower bounded by $\max(\epsilon_t, \epsilon_s)$, where $\|\cdot\|_F$ is the Frobenius norm.

Proof: The approximate model may be expressed as,

$$\mathbf{S} \; = \; \mathbf{\Theta}\mathbf{A}^T + \mathbf{\Theta}^{\perp}\mathbf{A}^{\perp}, \tag{8}$$

$$\mathbf{S} \; = \; \mathbf{\Omega}\mathbf{B}^T + \mathbf{\Omega}^{\perp}\mathbf{B}^{\perp T}, \tag{9}$$

where the columns of $\mathbf{\Theta}^{\perp}$ and $\mathbf{B}^{\perp}$ form a basis for the nullspaces of $\mathbf{\Theta}^T$ and $\mathbf{B}^T$ respectively. $\mathbf{A}^{\perp}$ and $\mathbf{\Omega}^{\perp}$ are the coefficients of these nullspaces. Here $\epsilon_t = \|\mathbf{\Theta}^{\perp}\mathbf{A}^{\perp}\|_F$ and $\epsilon_s = \|\mathbf{\Omega}^{\perp}\mathbf{B}^{\perp T}\|_F$. Setting Equations 8 and 9 equal we get,

$$\mathbf{\Theta}\mathbf{A}^T + \mathbf{\Theta}^{\perp}\mathbf{A}^{\perp} = \mathbf{\Omega}\mathbf{B}^T + \mathbf{\Omega}^{\perp}\mathbf{B}^{\perp T}. \tag{10}$$

This can be rearranged to give,

$$\mathbf{A}^T = \mathbf{\Theta}^T\mathbf{\Omega}\mathbf{B}^T + \mathbf{\Theta}^T\mathbf{\Omega}^{\perp}\mathbf{B}^{\perp T}, \tag{11}$$

as $\mathbf{\Theta}^T\mathbf{\Theta}^{\perp} = \mathbf{0}$. Substituting back into Equation 8, we get,

$$\mathbf{S} = \mathbf{\Theta}\mathbf{C}\mathbf{B}^T + \mathbf{\Theta}\mathbf{\Theta}^T\mathbf{\Omega}^{\perp}\mathbf{B}^{\perp T} + \mathbf{\Theta}^{\perp}\mathbf{A}^{\perp}. \tag{12}$$

By inspection we see that $\epsilon = \|\mathbf{\Theta}\mathbf{\Theta}^T\mathbf{\Omega}^{\perp}\mathbf{B}^{\perp T} + \mathbf{\Theta}^{\perp}\mathbf{A}^{\perp}\|_F$. From the triangle inequality we get $\epsilon \leq \|\mathbf{\Theta}\mathbf{\Theta}^T\mathbf{\Omega}^{\perp}\mathbf{B}^{\perp T}\|_F + \epsilon_t$. As $\mathbf{\Theta}\mathbf{\Theta}^T$ is an orthogonal projection matrix onto the range of $\mathbf{\Theta}$, it follows that $\|\mathbf{\Theta}\mathbf{\Theta}^T\mathbf{\Omega}^{\perp}\mathbf{B}^{\perp T}\| \leq \|\mathbf{\Omega}^{\perp}\mathbf{B}^{\perp T}\| = \epsilon_s$. Therefore,

$$\epsilon \leq \epsilon_t + \epsilon_s. \tag{13}$$

A dual equality can be written where $\epsilon = \|\mathbf{\Theta}^{\perp}\mathbf{A}^{\perp}\mathbf{B}\mathbf{B}^T + \mathbf{\Omega}^{\perp}\mathbf{B}^{\perp T}\|$ can be derived using $\mathbf{\Omega} = \mathbf{\Theta}\mathbf{A}^T\mathbf{B}^{T+} + \mathbf{\Theta}^{\perp}\mathbf{A}^{\perp}\mathbf{B}$ instead of Equation 11. As ϵ must be greater than ϵ_s and ϵ_t it follows that

$$\epsilon \geq \max(\epsilon_t, \epsilon_s). \tag{14}$$

□

Theorem 2 states that the bilinear spatiotemporal model error cannot exceed the sum of errors of the shape and trajectory models. This error, however, is reached with far fewer coefficients for the bilinear model as compared to the shape or trajectory models; this parsimony will be demonstrated presently.

3.3 Comparison with Linear Spatiotemporal Basis

It is instructive to compare the bilinear spatiotemporal basis with a linear spatiotemporal representation. In the latter case, $\mathbf{S}$ is vectorized into a single column vector which can be represented by a linear spatiotemporal basis,

$$\text{vec}\,(\mathbf{S}) = \mathbf{\Phi}\hat{\mathbf{c}}, \tag{15}$$

where vec $(\mathbf{S})$ is a column-wise vectorized version of $\mathbf{S}$, $\mathbf{\Phi}$ is a $3FP \times K$ matrix representing K *linear spatiotemporal* basis vectors, and $\hat{\mathbf{c}}$ is a $K \times 1$ column vector of coefficients (denoted by the white node in Figure 2). We can write the bilinear spatiotemporal basis as a linear model,

$$\text{vec}\,(\mathbf{S}) = (\mathbf{B} \otimes \mathbf{\Theta})\,\text{vec}\,(\mathbf{C})\,, \tag{16}$$

where $\otimes$ denotes the Kronecker Product [Jain 1989]. Hence, for K_s shape and K_t trajectory bases in the bilinear model, an equivalent linear model will have $K = K_sK_t$ columns in $\mathbf{\Phi}$. It must be noted, however, that not all linear spatiotemporal basis can be factored into bilinear spatiotemporal basis.

While the linear and the bilinear spatiotemporal models can model both spatial and temporal regularity, linear spatiotemporal bases do not generalize well, unless trained on substantial amounts of data. For limited data, the model may learn sequence-specific correlations, thus making them inapplicable to new sequences. In bilinear models, sequence-specific correlations are limited because trajectory and shape dimensions are separated, improving generalization.

3.4 Conditioned Bilinear Bases

We observe that while appropriate shape basis will often have to be learned to suit particular datasets, the high degree of temporal smoothness in natural motions allows a predefined analytical trajectory basis to be used for a wide variety of datasets without significant loss in representation. The *conditioned* bilinear spatiotemporal representation is thus a special case of Equation 6,

$$\mathbf{S} = \widetilde{\mathbf{\Theta}}\mathbf{C}\mathbf{B}^T + \epsilon, \tag{17}$$

where $\widetilde{\mathbf{\Theta}}$ contains the first K_t vectors of the predefined trajectory basis arranged along its columns, each of length F. The ability to use a predefined trajectory basis yields closed form and numerically stable solutions, for both the estimation of the shape basis and coefficients in Equation 6. The benefit of using a trajectory basis for which an analytical expression exists is that the same model can represent time-varying structures of arbitrary durations.

A particularly suitable choice of a conditioning trajectory basis is the Discrete Cosine Transform (DCT) basis. Figure 4 shows that the optimal PCA basis learned from a large number of varied facial motion capture sequences converges to the DCT basis. This result is consistent with Akhter and colleagues [2010] who have demonstrated a similar experiment for human body sequences, and with that of Arikan [2006] who use the DCT basis to compress motion-capture data. Indeed, it is well known that the DCT basis approaches the optimal PCA basis if the data is generated from a stationary first-order Markov process [Rao and Yip 1990]. Given the high temporal regularity present in almost all human motions, it is not surprising that we empirically find DCT to be an excellent basis for trajectories of varied types.

Other choices of a conditioning trajectory basis are possible and may be preferable in specific applications. While DCT shows compaction that is close to optimal, the support of each basis vector is global and each coefficient affects the entire sequence. This may be undesirable in some cases, and therefore overlapped-block versions of the DCT are often used in online signal processing tasks. A practical alternative with localized basis support is the B-spline basis [Deboor 1978], commonly used to approximate smooth functions while offering local control over the shape of the curve. The B-spline basis is not orthogonal, which results in a slightly more expensive solution for estimating the coefficients, as will be shown in Section 3.5.

Using a predefined trajectory basis is a major strength of the bilinear representation, which not only reduces the complexity of estimating bilinear bases to being nearly identical to shape-only models, but also provides good generalization capabilities, and the ability to handle sequences of arbitrary duration. In contrast, for the linear spatiotemporal model given in Equation 15, the spatial and the temporal components do not factor out separately, and hence it is not possible to use a predefined basis for one mode of variation and a data-driven basis for the other.

3.5 Parameter Estimation

An important strength of the conditioned bilinear model is that the estimation of coefficients and basis have closed form solutions requiring only linear least squares and SVD routines. Hence, the estimation is efficient, optimal, and numerically stable. Learning the bilinear basis given a set of example sequences has been studied for the more general cases of bilinear and multi-linear models [Magnus and Neudecker 1999]. While several competing tensor decompositions exist, one possibility is to iteratively project and take the SVD in each of the two subspaces, analogous to the process of estimating the bilinear coefficients in [Tenenbaum and Freeman 2000]. Another option is to stack the sequences as a third order tensor and factorize it using Higher Order SVD (HOSVD), a generalization of SVD for tensors. However, the conditioned bilinear representation results in simplified estimation, because in this case, one of the three matrices in the model is already known. In the following subsections, we first discuss the problem of computing the coefficients, $\mathbf{C}$, given known bases $\mathbf{\Theta}$ and $\mathbf{B}$. Subsequently, in Section 3.5.2 we address the problem of estimating the shape basis.

3.5.1 *Estimating the Coefficients of a Bilinear Model.* Given a shape basis $\mathbf{B}$ and a trajectory basis $\mathbf{\Theta}$, we wish to compute the bilinear model coefficients $\mathbf{C}$, that minimize the reconstruction error for a given $\mathbf{S}$. The solution may be estimated by minimizing the squared reconstruction error:

$$\mathbf{C} = \arg\min_{\mathbf{C}} \left|\left|\left(\mathbf{S} - \mathbf{\Theta}\mathbf{C}\mathbf{B}^T\right)\right|\right|_F^2 . \tag{18}$$

For any bases $\mathbf{\Theta}$ and $\mathbf{B}$, the general solution for optimal $\mathbf{C}$ is in terms of the pseudo-inverses

$$\mathbf{C} = \mathbf{\Theta}^+\mathbf{S}\left(\mathbf{B}^T\right)^+ , \tag{19}$$

where superscripted $+$ denotes the Moore-Penrose pseudo-inverse. For the case when both $\mathbf{\Theta}$ and $\mathbf{B}$ have full column-rank, the above solution is unique. If the bases are orthogonal, then the solution simplifies to $\mathbf{C} = \mathbf{\Theta}^T\mathbf{S}\mathbf{B}$, which implies simply projecting the structure $\mathbf{S}$ onto each of the bases sequentially. This simplification applies to the DCT basis, but not to the B-spline basis, since that basis is not orthonormal.

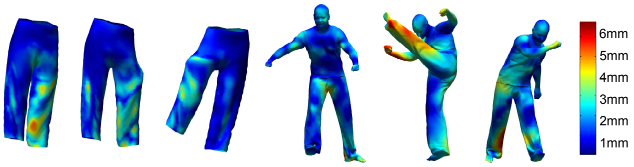


Fig. 5. The model is applied to represent cloth and full body motion. The heat map on the meshes denote the reconstruction error induced by projecting the data onto the bilinear spatiotemporal basis and then reconstructing the data from the projection.

3.5.2 *Conditioned Shape Basis Estimation.* While HOSVD or iterative SVD may be used to estimate bilinear bases in general, the estimation of the conditioned bilinear bases is significantly simpler. This is because the trajectory basis are already known. Hence, given a set of training examples, the appropriate shape basis for the conditioned bilinear model may be estimated using the following theorem.

Theorem 3: Given a trajectory basis $\mathbf{\Theta}$ and a set of training sequences of time-varying structure, $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_N$, the optimal shape basis which minimizes the squared reconstruction error is given by the row-space computed through SVD of the matrix,

$$\mathbf{\Pi} = \left[\hat{\mathbf{S}}_1^T, \hat{\mathbf{S}}_2^T, \ldots, \hat{\mathbf{S}}_N^T\right]^T ,$$

where $\hat{\mathbf{S}}_i = \mathbf{\Theta}\mathbf{\Theta}^+\mathbf{S}_i$.

Proof: We wish to find the shape basis which will minimize the reconstruction error. Given a single sequence, since the estimated coefficients are given by $\mathbf{C} = \mathbf{\Theta}^+\mathbf{S}\left(\mathbf{B}^T\right)^+$, the reconstructed structure can be written as $\hat{\mathbf{S}} = \mathbf{\Theta}\mathbf{\Theta}^+\mathbf{S}\left(\mathbf{B}^T\right)^+\mathbf{B}^T$. Hence, the optimal shape basis can be computed by minimizing the reconstruction error:

$$\arg\min_{\mathbf{B}} \left|\left|\mathbf{S} - \mathbf{\Theta}\mathbf{\Theta}^+\mathbf{S}\left(\mathbf{B}^T\right)^+\mathbf{B}^T\right|\right|_F^2 .$$

Expanding $\mathbf{S}$ into its components that span the trajectory basis and its null space, this may be written as,

$$\arg\min_{\mathbf{B}} \left|\left|\mathbf{\Theta}\mathbf{\Theta}^+\mathbf{S} + \mathbf{\Theta}^\perp\mathbf{A}^\perp - \mathbf{\Theta}\mathbf{\Theta}^+\mathbf{S}\left(\mathbf{B}^T\right)^+\mathbf{B}^T\right|\right|_F^2 ,$$

where $\mathbf{\Theta}^\perp$ is the null space of $\mathbf{\Theta}$ and $\mathbf{A}^\perp$ contains the coefficients of this null space. Defining $\hat{\mathbf{S}} = \mathbf{\Theta}\mathbf{\Theta}^+\mathbf{S}$, which denotes the reconstruction of $\mathbf{S}$ from its trajectory projection, and observing that, for a fixed $\mathbf{\Theta}$, $\mathbf{\Theta}^\perp\mathbf{A}^\perp$ does not depend on the choice of $\mathbf{B}$, it is equivalent to minimize,

$$\arg\min_{\mathbf{B}} \left|\left|\hat{\mathbf{S}} - \hat{\mathbf{S}}\left(\mathbf{B}^T\right)^+\mathbf{B}^T\right|\right|_F^2 ,$$

which, for a rank-K_s reconstruction with orthogonal $\mathbf{B}$, is computed as the row space of $\hat{\mathbf{S}}$ via SVD.

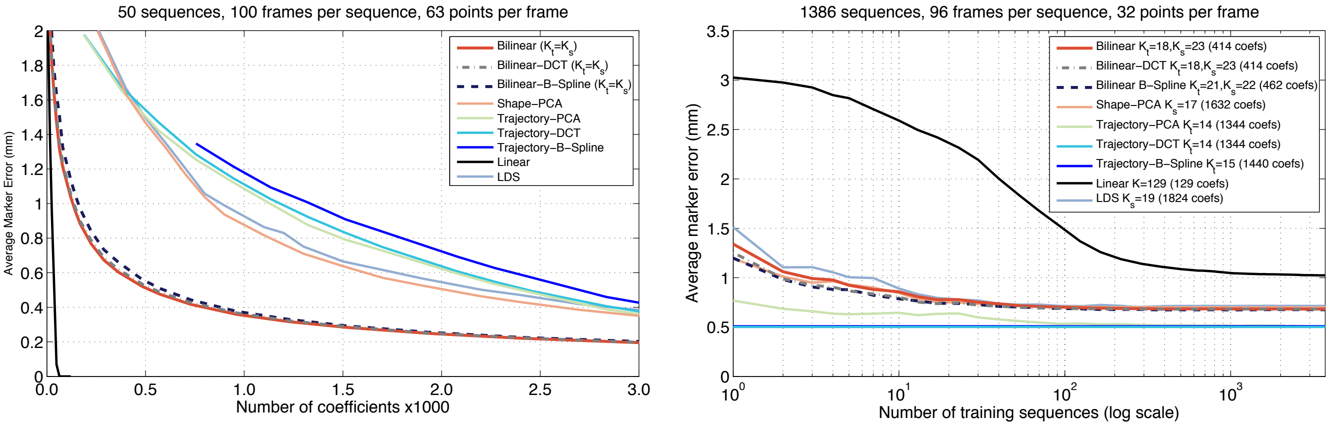
For more than one structure sequence, each sequence will have to be first reconstructed from its trajectory projection, and the optimal shape basis $\mathbf{B}$ will result from the SVD of the matrix formed by stacking the sequences into an $FN \times 3P$ matrix $\mathbf{\Pi}$, defined above. This is because the error to be minimized,

$$\arg\min_{\mathbf{B}} \sum_i \left|\left|\hat{\mathbf{S}}_i - \hat{\mathbf{S}}_i\left(\mathbf{B}^T\right)^+\mathbf{B}^T\right|\right|_F^2 ,$$

is equivalent to,

$$\arg\min_{\mathbf{B}} \left|\left|\mathbf{\Pi} - \mathbf{\Pi}\left(\mathbf{B}^T\right)^+\mathbf{B}^T\right|\right|_F^2 ,$$

if $\left(\mathbf{B}^T\right)^+\mathbf{B}^T$ is factored out from each of the summation terms. □



(a) Reconstruction Error on Training Data

(b) Generalization Error on Testing Data

Fig. 6. Compactness and Generalization Ability. (a) Average reconstruction error in marker displacement for a varying number of model parameters. The reconstruction error is the average marker displacement computed across all frames on 50 distinct dense facial motion capture sequences of 100 frames each. The bilinear spatiotemporal model achieves the same average error as the shape and trajectory models with much fewer coefficients. (b) Reconstruction error on unseen test sequences for varying amounts of training data. Here, the models are learned on training sequences and fit on an unseen testing set. For fair comparison, the sizes of the latent spaces for all models were chosen such that the reconstruction error on the full training set was less than 0.5 mm with minimal number of coefficients. The relative rate of improvement rather than the final error is the more relevant measure in this graph.

Theorem 3 implies that the estimation of the shape basis for the conditioned bilinear model is based on first projecting each structure sequence onto the trajectory basis, then reconstructing structure back from that projection, and finally computing the row space via SVD of the matrix formed by stacking all these reconstructions together.

3.6 Properties of Bilinear Spatiotemporal Models

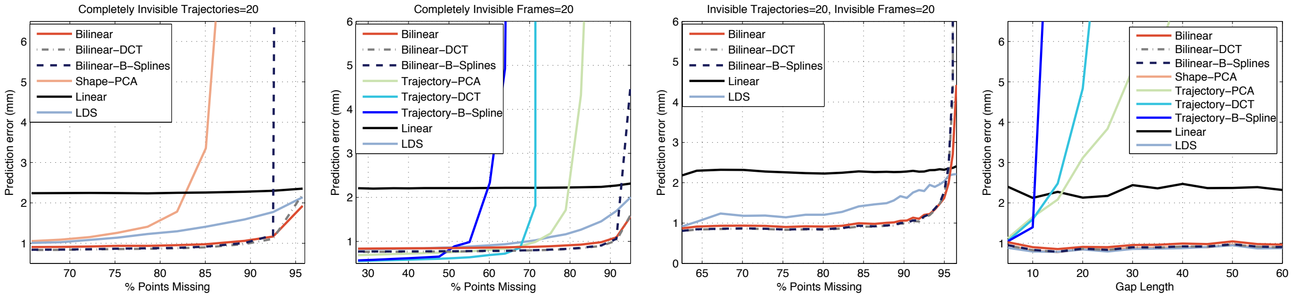
We now analyze the properties of the bilinear representation in terms of compactness, generalization ability, predictive precision, and computational efficiency compared to five other commonly used representations of time-varying spatial data. We contrast the properties of the bilinear spatiotemporal models (Equations 6 and 17), and compare them to the shape model (Equation 3), the trajectory model (Equation 5), and the linear spatiotemporal model (Equation 15). In addition, we also compare our model to a Linear Dynamical System (LDS) approach, where reconstruction is implemented as Kalman smoothing and the model is trained using Expectation Maximization (EM), and to a B-spline basis representation of the time-varying point-clouds[4]. The conceptual relationship between these approaches can be compared through Figure 2.

3.6.1 *Compactness.* Compactness, or parsimony, is the ability of a model to represent data with fewer parameters. Since the bilinear representation exploits both spatial and temporal regularity, it requires far fewer coefficients for the same reconstruction error compared to shape or trajectory representations. The number of coefficients in Equations 6 or 17 is $K_t \times K_s$. Note that for a subspace representation to be useful, typically $K_t \ll F$ and $K_s \ll 3P$. Hence, the $K_t \times K_s$ coefficients in $\mathbf{C}$ are far fewer than the $F \times K_s$ coefficients in $\mathbf{\Omega}$ or the $K_t \times 3P$ coefficients in $\mathbf{A}$. Figure 5 shows cloth data and full body scans reconstructed by re-projecting the data from the bilinear spatiotemporal basis.

Empirically, the conditioned bilinear spatiotemporal models are shown to demonstrate a reduction of nearly an order of magnitude for the same reconstruction error when compared to the shape or trajectory models. Figure 6(a) shows the compaction performance of several models on dense facial motion capture data. We use 50 temporally non-overlapping face sequences extracted from 18 different facial motion-capture sentence sequences of a single actor. The plots show reconstruction error in average marker displacement (millimeters) when varying the number of coefficients in each model. The results indicate that an average reconstruction error of about 0.5mm is achieved with approximately 500 bilinear-DCT or bilinear-B-spline coefficients, but requires around 2,000 shape coefficients (approximately 20 coefficients per frame) or 2,500 trajectory coefficients (approximately 40 coefficients per point) to reach similar error values.

The conditioned bilinear spatiotemporal models is also compared against a bilinear spatiotemporal model that uses a data-driven trajectory basis. The two methods perform similarly, indicating no significant loss of representation by choosing the generic DCT or B-spline basis over a data-driven trajectory basis. The DCT-conditioned model has similar, though slightly improved, compaction ability when compared to a B-spline conditioned bilinear spatiotemporal model, indicating that the bilinear models require roughly similar number of coefficients for a reasonable choice of predefined basis. However, if a B-spline basis is used as a trajectory-only model without combining them in a bilinear representation, compaction is significantly reduced. Our model is also compared with the Linear Dynamical System (LDS) model, where the number of latent variables required to reach the same recon-

[4] For LDS we use the Kalman filtering toolbox by Kevin Murphy, available at `http://www.cs.ubc.ca/~murphyk/Software/Kalman/kalman.html`. As a B-Spline basis, we used a cubic B-spline basis defined by an open uniform knot vector, where the number of coefficients for this trajectory basis was $K_t = n + k$, with n the number of internal knots and $k = 4$ for cubic degree splines.



(a) Spatial Imputation (b) Temporal Imputation (c) Spatiotemporal Imputation (d) Clustered Imputation

Fig. 7. Predictive precision in synthetically generated occlusion scenarios. In (a-c) randomly selected points were occluded whereas in (d) a contiguous chunk of 15 randomly selected trajectories were occluded. For training, 25 sequences were used and another 25 were used for testing. Each sequence has 100 frames and 63 points. The number of bases was selected to ensure 0.5mm reconstruction error on the training data. Bilinear models show remarkably better performance compared to other methods in all four scenarios. The trajectory models cannot be applied in (a) and (c) because some of the trajectories are completely missing. Similarly, the shape model cannot be applied in (b) and (c) because some of the frames are completely missing.

struction error is an order of magnitude more than for bilinear models.

In comparison to the bilinear spatiotemporal model, the linear spatiotemporal model requires even fewer coefficients. In this experiment, 50 coefficients will always suffice to represent the training data with zero error, because that equals the number of training sequences. However, such training constitutes an exact overfit and does not generalize well to other sequences. The length of each linear basis vector becomes prohibitively large even for sequences of moderate size. For the experiment reported above, each basis vector in the linear representation will contain $3 \times 100 \times 63 = 18,900$ terms, compared to each DCT basis of length 100 and each shape basis is of length $3 \times 63 = 189$ in the bilinear spatiotemporal model.

3.6.2 *Generalization Ability.*

Generalization is the ability of a model to describe instances outside its training set. For the same number of coefficients, the bilinear basis has far fewer model parameters than the linear spatiotemporal basis and consequently they tend to generalize better. The separation of spatial and temporal modes of variation reduces the sequence-specific correlations that can be learned, and due to the smaller sizes of the bases in the bilinear model, the estimation error in computing the bases suffers relatively less from the curse of dimensionality.

Figure 6(b) empirically validates these observations. For this experiment, we took 18 motion-capture sequences of an actor and extracted around 5,000 overlapping sub-sequences of 96 frames, each offset by 2 frames. For comparison with the linear spatiotemporal model and LDS, it was necessary to subsample the spatial resolution to only 32 points in each frame due to the large memory and computational footprint of these methods. Of these sequences, roughly two thirds were set aside for training and the remaining third was used for testing. By varying the number of training examples used for training the models, we computed the reconstruction error on the testing dataset. The results, plotted on log-scale in Figure 6(b), confirm that bilinear spatiotemporal models have superior generalization ability than the linear spatiotemporal model, showing smaller error on test sequences for the same number of training sequences. The bilinear basis estimated through iterative SVD generalizes very similarly to the conditioned bilinear basis. We observe that the learned trajectory basis approaches the DCT, as discussed earlier. A large number of training sequences is necessary for the linear model to generalize.

The properties of compaction and generalization ability compete: better compaction often comes at the cost of generalization ability. Studying Figures 6(a) and 6(b) together shows that bilinear models provide a good tradeoff between these two properties. The linear spatiotemporal model is highly compact (at least in number of coefficients), but extremely poor in the ability to generalize. All trajectory-only models using predefined basis generalize very well because the basis is chosen to suit a large number of sequences, but, for the same reason, have significantly lower compaction. LDS and shape-only models have roughly equivalent generalization ability as bilinear models at the cost of significantly poorer compaction.

3.6.3 *Predictive Precision.*

Predictive precision is the ability of a model to impute missing data accurately. To compare predictive precision of different models, we took 50 face sequences, split 25-25 into training and testing sequences. As in the generalization ability experiment, the bases were learned from the training sequences. However, instead of estimating the coefficients from a complete test sequence, random gaps in the data were synthetically introduced to simulate missing data. Four types of occlusion simulations were conducted to compare predictive precision across both space and time in Figure 7: (a) randomly occluded points with some trajectories completely occluded across all time instances, (b) randomly occluded points with some frames completely occluded at some time instances, (c) randomly occluded points with some trajectories completely occluded across all time instances *and* some frames completely occluded at some time instances, and (d) random occlusion of spatiotemporally contiguous chunks of data to simulate practical scenarios.

Figure 7 shows greater predictive precision for the bilinear model compared to other models. In the spatial imputation experiment (Figure 7(a)), the bilinear spatiotemporal model, the DCT-conditioned model, and the B-spline conditioned model show markedly improved performance compared to the shape model, LDS, and the linear spatiotemporal model. The trajectory models are not used because it is impossible to use them to impute completely occluded trajectories. The results of the temporal imputation experiment (Figure 7(b)) are similar, the three bilinear models show better imputation with significant percentage of missing data.

Table I. Memory requirements and computational cost for the Bilinear, conditioned Bilinear, Shape, Trajectory, Linear, and LDS.

Method	Model storage	Coefficients	Reconstruction cost
Bilinear	$3PK_s + FK_t$	K_sK_t	$O(FK_s\,[3P + K_t])$ or $O(3PK_t\,[F + K_s])$
Bilinear (conditioned)	$3PK_s$	K_sK_t	$O(FK_s\,[3P + K_t])$ or $O(3PK_t\,[F + K_s])$
Shape	$3PK_s$	FK_s	$O(3PFK_s)$
Trajectory	FK_t	$3PK_t$	$O(3PFK_t)$
Linear	$3PFK_tK_s$	K_sK_t	$O(3PFK_sK_t)$
LDS	$3PK_s + K_s^2$	FK_s	$O(3PFK_s)$

For the purposes of comparison we assume that the number of coefficients of the linear model is $K = K_sK_t$. For LDS, we assume that the size of the latent space is K_s, and we have disregarded the storage cost for the noise covariance matrices.

Shape models could not be used in this experiment as they cannot impute completely missing frames. In the spatiotemporal imputation experiment (Figure 7(c)), neither shape-only, nor trajectory-only models could be used. Again, bilinear models show significantly better performance for a range of missing percentage of points. Finally, for the last experiment in which space-time chunks of data were occluded (Figure 7(d)), the bilinear models and the LDS showed comparable performance, superior to other models.

3.6.4 *Efficiency.* Finally, we discuss the computational efficiency of the two most common operations when working with data models: data reconstruction given the coefficients, and coefficient estimation given the data. In terms of storage, the bilinear models require almost the same order of magnitude storage for the basis as the shape or trajectory models, which is a significant saving over the linear spatiotemporal representation. In terms of coefficients, they are much more compact than shape or trajectory models, and their cost of reconstruction is not significantly higher.

The computational complexity of reconstruction for the bilinear model (Equation 6) is due to multiplication by the shape basis and trajectory basis matrices. Depending on the order of operations, this will be $O(FK_s\,[3P + K_t])$[5] when multiplying by the trajectory basis first, or $O(3PK_t\,[F + K_s])$ when multiplying by the shape basis first. Reconstruction for the corresponding shape-only and trajectory-only models is $O(3PFK_s)$ and $O(3PFK_t)$ respectively, which is on the order of FK_sK_t (or $3PK_sK_t$) fewer operations at the expense of a larger memory footprint. Reconstruction for a corresponding linear spatiotemporal model with $K_s \times K_t$ basis vectors is $O(3PFK_sK_t)$ which usually will be the most expensive method.

Similarly, the cost of estimating the bilinear coefficients given a spatiotemporal sequence is $O(3PK_t\,[F + K_s])$ (multiplication by the trajectory basis first), or $O(FK_s\,[3P + K_t])$ (shape basis first). The cost of coefficient estimation for the shape-only, trajectory-only, and linear spatiotemporal models is the same as that of the corresponding reconstruction computation. In comparison, the cost of the Kalman filter implementation in our experiments was dominated by a term $O(F(3P)^3)$ related to the observation covariance[6]. Coupled with the need for several iterations during learning made the LDS model substantially slower than all other methods.

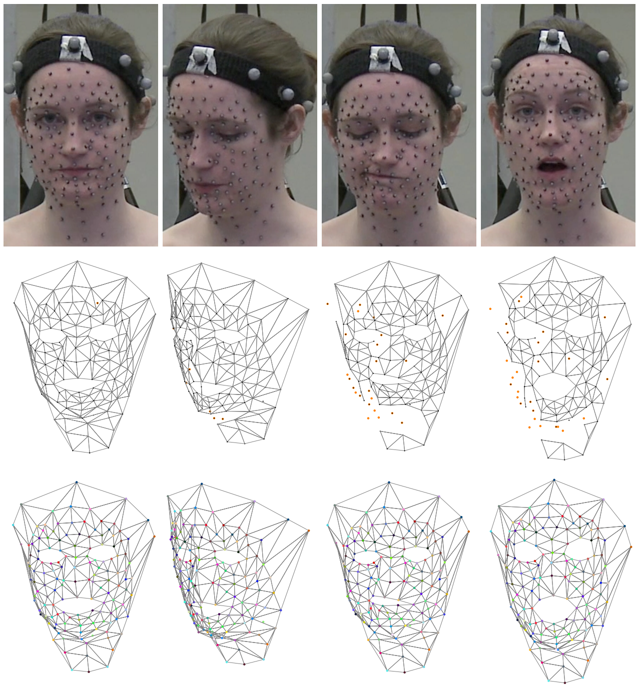


Fig. 8. Point clouds reconstructed from motion capture systems usually consists of broken trajectories and mislabeled points. The figure shows frame numbers 56, 84, 320, and 560 of a motion capture session (first row), raw motion capture data (second row) and our labeling results (third row). In raw data, as time progresses, errors propagate and more points are mislabeled.

4. APPLICATIONS

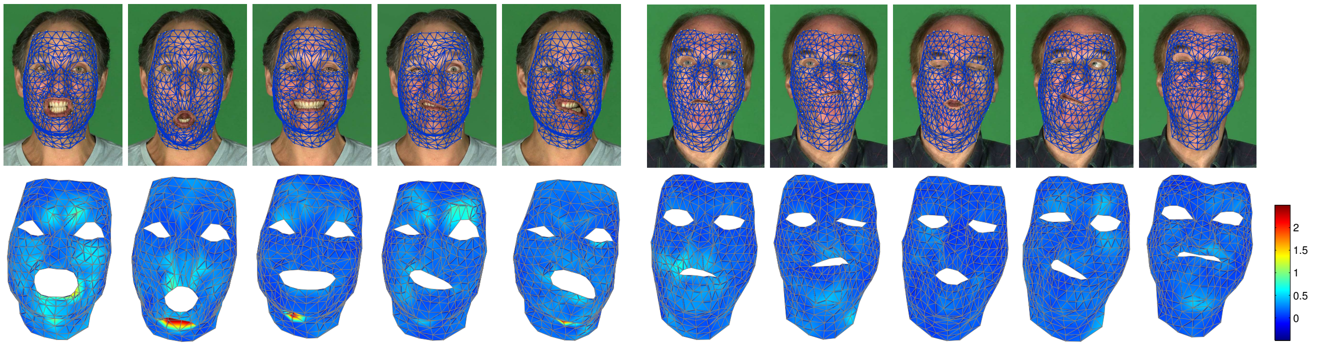
The conditioned bilinear model is applicable to a range of problems which require a compact representation of motion data. We demonstrate its efficacy for analysis tasks in the motion capture pipeline: denoising and marker labeling of raw data, gap-filling, and motion touch-up. The following subsections demonstrate: (1) marker labeling and denoising for dense facial motion capture, requiring just a few minutes of cleanup compared to the current standard of several hours of professional time, (2) gap-filling and imputation on face sequences given appropriately learned bases, and (3) a motion touch-up tool which allows plausible deformations of an entire motion capture sequence by moving only a few points and without employing any kinematic or skeletal model. Each of these applications exploits the DCT-conditioned bilinear model.

4.1 Motion Capture Labeling and Denoising

Reconstruction using motion capture systems often requires tedious post-processing for data cleanup, to connect broken trajectories, impute missing markers, correct mislabeled markers and denoise trajectories, as illustrated in Figure 8. We have developed a semi-automatic tool which simultaneously labels, denoises and imputes missing points, and drastically reduces the time required for cleanup while generating reconstructions qualitatively and quantitatively similar to those by industry professionals. The process often generates error-free labels, but when it does not, our semi-automated tool allows a few user-identified corrections to automat-

[5] The cost of matrix multiplication for matrices $m \times p$ by $p \times n$ is assumed $O(mpn)$.

[6] Alternative implementations of the Kalman filter exist specifically to improve on this cost.



(a) KB-ROM

(b) DC-ROM

Fig. 9. Labeling results of Bilinear-DCT based EM algorithm. Sub-figures (a) and (b) show our labeling results projected on images (first row) and Bilinear-DCT reconstruction (second row). Colors indicate error in reconstruction with respect to the ground truth. Results shows that Bilinear-DCT EM algorithm achieves similar accuracy as labeling by a professional, while significantly reducing manual effort.

ically propagate temporally, hence reducing cleanup time. Our approach is based on using the DCT-conditioned bilinear representation to compute marker labels. Given the bases, the estimation of bilinear spacetime coefficients and marker labels is interdependent and are iteratively estimated using the Expectation Maximization (EM) algorithm.

4.1.1 *Expectation Maximization.* We model the marker data using the DCT-conditioned bilinear basis. The observed 3D coordinates of the p^{th} marker in frame f is $\hat{\mathbf{X}}_f^p = \mathbf{X}_f^p + e$, where $e \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ is measurement error, and $\mathbf{X}_f^p$ is the true value of $\hat{\mathbf{X}}_f^p$ and σ denotes the standard deviation of the error. We want to assign a label $l_f^p \in \{1, \dots, P\}$ to each marker $\mathbf{X}_f^p$ associating it to a unique trajectory, such that the rearranged matrix $\mathbf{S} = \mathbf{\Theta}\mathbf{C}\mathbf{B}^T$. The goal of the EM algorithm is to estimate both the set of hidden variables l_f^p as well as the model parameters, $\mathbf{C}$ and σ.

In the expectation step, we estimate the probabilistic labeling of the spacetime points given an initialization of the bilinear coefficients $\mathbf{C}$. In the maximization step, the probabilistic labeling is used to estimate the maximum likelihood estimate of $\mathbf{C}$. We found that the running time of the algorithm can be significantly improved by making a hard assignment of the unlabeled points, instead of doing this probabilistically. This is often referred to as the hard-EM algorithm[7]. This simplification reduces the expectation step to estimating imputation using equation, $\hat{\mathbf{S}} = \mathbf{\Theta}\mathbf{C}\mathbf{B}^T$ and assigning labels to raw data points based on the label of the closest imputed point in each frame. In the maximization step, the raw points are arranged into the structure matrix $\mathbf{S}$. The coefficients are then estimated as $\mathbf{C} = \mathbf{\Theta}^T\mathbf{S}\mathbf{B}$.

To initialize the marker labels for the EM algorithm, we exploit the smoothness of trajectories to propagate labels from one frame to the next. We do this by estimating the model coefficients using the first N frames, and imputing the marker positions at frame $N + 1$ by using the analytical expression of the DCT trajectory basis to extend the sequence length during reconstruction. The first frame is initialized by assigning arbitrary labels to each point in a user selected frame containing all markers. Once an initial estimate of the marker labels is known, we can estimate the shape basis $\mathbf{B}$ and the coefficients $\mathbf{C}$. These estimates will be used to initialize the EM algorithm as described above.

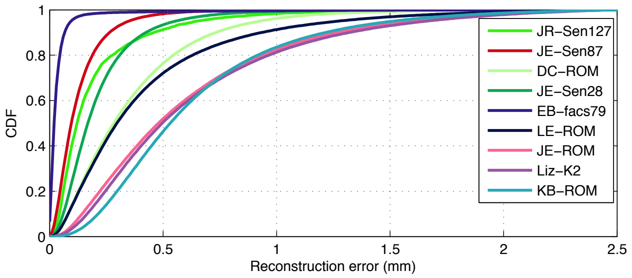


Fig. 10. Accuracy plots for the Bilinear-DCT based EM algorithm. We compute the per frame average distance between our reconstruction and ground truth for six sequences. The figure shows close resemblance between the two reconstructions in terms of the cumulative distribution function (CDF) of reconstruction errors.

To guard against errors in the labeling biasing the estimate of the shape basis, we use an ordering constraint to find mislabeled data. Errors in the initial labeling can often be identified because for each mislabeled point, at least one of the triangles in the triangulated mesh containing that point as a vertex switches the direction of its normal vector. Therefore by comparing the direction of a normal vector of a triangle in the current frame with the one in the previous frame, the consistency of labeling can be determined. This filtering can be used to identify frames which do not have labeling errors; to ensure correct estimation, the shape basis, $\mathbf{B}$, is learned using only these frames.

The labels computed by the EM algorithm may not always be correct. In our setup, an interactive tool is used to review the triangulated mesh output overlaid on raw motion capture data. Errors in labeling can be corrected by clicking on the mesh and the corresponding raw data point in one frame. This correction is propagated automatically over time by recomputing the initialization procedure described above. When the user is satisfied with all corrections we reconstruct the final imputation by projecting the labeled data onto the bilinear basis.

[7] http://ttic.uchicago.edu/~dmcallester/ttic101-07/lectures/em/em.pdf

4.1.2 *Quantitative and Qualitative Evaluation.* We extensively test our labeling algorithm on more than 1000 dense facial motion capture sequences of five different actors. Each actor performed roughly 200 sequences of varying expressions and emotions. The average length of a sequence was about 1000 frames. Completely automated labeling of these sequences was carried out using the algorithm described above. About 10% of the sequences were found to violate the ordering constraints; the rest showed consistent labels, some of which were qualitatively verified manually. The sequences which did violate the ordering constraints were corrected manually using our interactive tool. The average time to correct the labels of one such sequence was roughly 10 to 15 minutes, depending on the number of errors. Labeling such a large dataset with currently available industry standard software would take prohibitive amounts of time.

For a quantitative comparison, we selected 9 face sequences at random and had them labeled by an industry professional. Figure 9 shows five frames selected from two range-of-motion (ROM) sequences with our labeling results compared against the professional reconstruction. While both results are qualitatively similar, our method takes about 10 minutes per sequence compared to approximately 2-3 hours to label a 1000 frames long sequence with roughly 300 markers. Figure 10 shows the quantitative comparison in terms of the Euclidean distance of each labeled point with its ground truth location, as determined by the industry professional. The cumulative distribution function (CDF) of the error shows that almost all markers exhibit less than 2 mm displacement, attesting to the quality of reconstruction and labeling. Figure 11 shows a close-up example of a portion of the X- and Y-trajectories, showing that our model inherently filters out high frequency noise due to the use of DCT as a trajectory basis, and adapts well to the raw data trajectories.

4.2 Gap Filling and Imputation

Missing data in both space and time can be reconstructed well through the DCT-conditioned bilinear model. Since the representation is compact, a few points may be enough to reconstruct the entire sequence provided that a good shape basis is learned. In several experiments shown in the accompanying video, we estimated a shape basis on range-of-motion (ROM) sequences, because they capture much of the variability of human expression. In Figure 12, the conditioned bilinear model trained on the second half of a single ROM sequence and used to impute missing data on the first half on the sequence. We randomly discard marker observations from the unseen first half of the sequence, and estimate the coefficients from the remaining points. The model yields a convincing reconstruction with an average error of around 1 mm for up to 99% missing observations.

4.3 Motion Touch-up

Motion capture data often requires touch-ups or more extensive editing to adapt the recorded motions to new situations. Examples include preventing mesh interpenetrations after insertion into new environments and matching up the motions of several characters. In these scenarios, we require that the adapted motions meet new constraints in spacetime, but we would like to retain most of the original motion's dynamics and spatial features [Gleicher 1997; 2001].

Tasks of this type fit into the framework of constraint-based motion adaptation [Gleicher and Litwinowicz 1998]. The bilinear spacetime formulation is readily applicable to this framework: user-specified constraints can be directly incorporated into a system of linear equations involving the model parameters. The solution of this system yields globally modified marker trajectories while ensuring a smooth, close fit to the original data. Because the fitting is performed in the model's compact parameterization, the resulting motions match the statistics of the original data or that of the training data used to build the bilinear model.

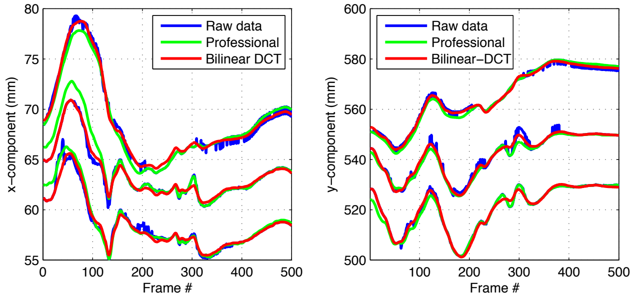


Fig. 11. Illustration of Denoising: Comparison of raw point trajectories (blue), clean-up by a professional (green) and clean-up by our method (red). Our method adapts well to the raw trajectories while filtering out high-frequency noise.

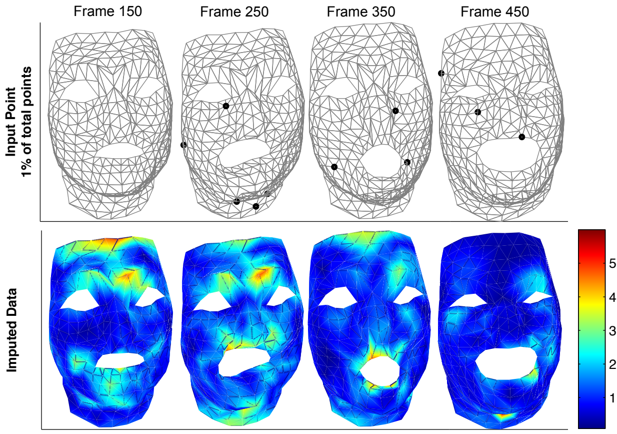


Fig. 12. To demonstrate gap-filling through DCT-conditioned bilinear basis, 315 face points varying over 600 frames are reconstructed from around 1% of the original points (marked on the selected frames by black dots in the top row) using the spatiotemporal model learned from a single training sequence. The colors correspond to reconstruction error.

Perhaps the closest formulation of this optimization problem is that of direct manipulation blendshapes [Lewis and Anjyo 2010]. In that work, the user constrains the position of individual points and the optimization process solves for the coefficients of a shape basis (blendshapes of a facial rig). In our work, the constraints are spacetime events—a point constrained to a location at a particular moment in time—and the coefficients are those of the bilinear basis. Formally, given a matrix $\mathbf{S}_c$ with the desired position of certain points at specific frames, we can solve for the global motion parameters, $\mathbf{C}$, that deviate as little as possible from the original motion parameters, $\mathbf{C}_0$, but satisfy soft-constraints on points in $\mathbf{S}_c$:

$$\min_{\mathbf{C}} \lambda ||(\mathbf{S}_c - \mathbf{\Theta}_c \mathbf{C} \mathbf{B}_c^T)||^2_{W_c} + ||(\mathbf{C}_0 - \mathbf{C})||^2_{W_0}.$$

The parameter λ is chosen to be a high value so that constraints are approximately met. Vectorizing and expanding the matrix norms in the above expression results in a linear system of equations with an efficient least-squares solution that can typically be solved in real-time to allow for interactive editing.

$$e(\mathbf{c}) = \lambda(\mathbf{s} - \mathbf{\Phi c})^T W_c (\mathbf{s} - \mathbf{\Phi c}) + (\mathbf{c}_0 - \mathbf{c})^T W_0 (\mathbf{c}_0 - \mathbf{c}),$$

where $\mathbf{c} = \text{vec}(\mathbf{C})$, and as before, $\mathbf{\Phi} = \mathbf{B} \otimes \mathbf{\Theta}$. Additionally, diagonal weighting matrices have been introduced. W_c controls which points in $\mathbf{s}$ should be constrained (e.g., if the entry corresponding to point p in frame f has weight 0, that point is unconstrained). The diagonal matrix W_0 allows for non-equal penalization of changes to different coefficients. For example, by increasing the weight corresponding to higher-frequency DCT components in W_0, changes in low-frequency coefficients will be preferred, resulting in smoother changes to the trajectories. In our experiments, the weight assigned to low-energy shape basis vectors was also increased.

5. DISCUSSION

Gabiax and Laibson [2008] recently postulated properties of good models of data: parsimony, generalizability, tractability, empirical consistency, predictive precision, conceptual insightfulness, and falsifiability. The bilinear spatiotemporal model is highly compact and when conditioned with DCT is shown to generalize well to new data. The model is also tractable: the basis can be estimated using singular value decomposition and the coefficients can be estimated using least squares estimation. We empirically demonstrate that our model is consistent with motion capture data, and that it can accurately impute large portions of missing data. The model also provides valuable insights into the data when visualized as decomposed fundamental frequencies of principal shapes. Finally, while the model does satisfy falsifiability in the original sense of the term, it is not falsifiable in one sense: the model is statistical—learned from training data—and does not use spacetime physical constraints as developed by Witkin and Kass [1988]. For instance, in editing marker positions (see Figure 13), bone length will not necessarily be kept constant by this model, nor does the model ensure force-coherence of the motion. This limitation restricts the usage to touch-ups where the linear approximation is valid. An important direction of research lies in marrying the desirable properties obtained from statistical modeling with the correctness of a physical grounding, especially if this can be done while maintaining the numerical efficiency of the model.

A question not addressed in the present work is model selection. It is unclear in what way the number of basis K_t and K_s should be chosen. While cross-validation is an obvious first choice, it is often the case that there is insufficient data to do this, or that the properties of the data change too much from sequence to sequence to make fixed assumptions about the distribution of the data. This issue is related to choosing the amount of regularization, especially when imputing large amounts of missing data.

We have presented a compact, generalizable model for motion data that captures and exploits the dependencies across both the spatial and temporal dimensions, and shown that it is an empirically faithful model for various types of motion data. The bilinear spatiotemporal basis model makes it possible to quickly and efficiently label and denoise large databases of dense facial motion capture data, and we have shown its application in gap-filling, key-frame interpolation and motion adaptation. Motion data is crucial to applications in animation, robotics, and visual intelligence, and a good representation is fundamental to any process to be built upon it.

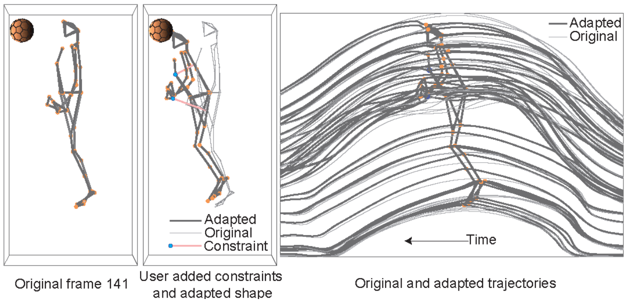


Fig. 13. Motion touch up of a motion captured jump. On the leftmost panel, the character's original motion fails to make contact with the ball. The user can constrain points at the desired time (middle panel) in order to satisfy the spacetime event. The bilinear model is then used to obtain the modified trajectories of all points. No kinematic model is used, yet non-rigid deformations such as bending the arms (middle panel) are possible with only two point constraints. The rightmost panel shows the original and globally modified trajectories with time on the x-axis. The source data is a cloud of motion capture markers from the CMU Motion Capture Database.

REFERENCES

AKHTER, I., SHEIKH, Y., KHAN, S., AND KANADE, T. 2008. Nonrigid structure from motion in trajectory space. In *Advances in Neural Information Processing Systems*.

AKHTER, I., SHEIKH, Y., KHAN, S., AND KANADE, T. 2010. Trajectory space: A dual representation for nonrigid structure from motion. *IEEE Transaction on Pattern Analysis and Machine Intelligence.*.

ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., RODGERS, J., AND DAVIS, J. 2005. Scape: shape completion and animation of people. *ACM Transactions on Graphics 24,* 3 (Aug.), 408–416.

ARIKAN, O. 2006. Compression of motion capture databases. *ACM Transactions on Graphics 25,* 3 (July), 890–897.

BREGLER, C., HERTZMANN, A., AND BIERMANN, H. 2000. Recovering non-rigid 3d shape from image streams. *IEEE Conference on Computer Vision and Pattern Recognition*, 690 – 696.

BRONSTEIN, A., BRONSTEIN, M., AND KIMMEL, R. 2008. *Numerical Geometry of Non-Rigid Shapes*, 1 ed. Springer Publishing Company, Incorporated.

CHAI, J. AND HODGINS, J. K. 2005. Performance animation from low-dimensional control signals. *ACM Transactions on Graphics 24,* 3 (Aug.), 686–696.

CHUANG, E. AND BREGLER, C. 2005. Mood swings: expressive speech animation. *ACM Transactions on Graphics 24,* 2 (Apr.), 331–347.

COOTES, T. F., TAYLOR, C. J., COOPER, D. H., AND GRAHAM, J. 1995. Active shape models — their training and application. *Computer Vision and Image Understanding*.

DE AGUIAR, E., SIGAL, L., TREUILLE, A., AND HODGINS, J. K. 2010. Stable spaces for real-time clothing. *ACM Transactions on Graphics 29,* 4 (July), 106:1–106:9.

DEANGELIS, G. C., OHZAWA, I., AND FREEMAN, R. D. 1995. Receptive-field dynamics in the central visual pathways. *Trends in Neurosciences 18,* 10, 451 – 458.

DEBOOR, C. 1978. *A practical guide to splines*. Springer.

DRYDEN, I. L. AND MARDIA, K. V. 1998. *Statistical Shape Analysis*. Wiley.

GABAIX, X. AND LAIBSON, D. 2008. The seven properties of good models. *The Foundations of Positive and Normative Economics*.

GLEICHER, M. 1997. Motion editing with spacetime constraints. In *1997 Symposium on Interactive 3D Graphics*. 139–148.

GLEICHER, M. 1998. Retargetting motion to new characters. In *Proceedings of SIGGRAPH 98*. Computer Graphics Proceedings, Annual Conference Series. 33–42.

GLEICHER, M. 2001. Comparing constraint-based motion editing methods. *Graphical Models 63,* 2.

GLEICHER, M. AND LITWINOWICZ, P. 1998. Constraint-based motion adaptation. *The Journal of Visualization and Computer Animation*, 65–94.

HAMARNEH, G. AND GUSTAVSSON, T. 2004. Deformable spatio-temporal shape models: extending active shape models to 2d+time. *Image and Vision Computing 22,* 6, 461 – 470.

HERDA, L., FUA, P., PLANKERS, R., BOULIC, R., AND THALMANN, D. 2001. Using skeleton-based tracking to increase the reliability of optical motion capture. *Human movement science 20,* 3, 313–341.

HOOGENDOORN, C., SUKNO, F., ORDS, S., AND FRANGI, A. 2009. Bilinear models for spatio-temporal point distribution analysis. *International Journal of Computer Vision 85*, 237–252.

HORNUNG, A., SAR-DESSAI, S., AND KOBBELT, L. 2005. Self-calibrating optical motion tracking for articulated bodies. In *Proceedings of Virtual Reality Conference (VR)*. IEEE, 75–82.

JAIN, A. 1989. *Fundamentals of digital image processing*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

LAWRENCE, N. D. 2004. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in Neural Information Processing Systems*.

LE, H. AND KENDALL, D. G. 1993. The riemannian structure of euclidean shape spaces: A novel environment for statistics. *The Annals of Statistics 21,* 3, pp. 1225–1271.

LEWIS, J. P. AND ANJYO, K.-I. 2010. Direct manipulation blendshapes. *IEEE computer graphics and applications 30,* 4, 42–50.

LI, H., WEISE, T., AND PAULY, M. 2010. Example-based facial rigging. *ACM Transactions on Graphics 29,* 4 (July), 32:1–32:6.

LI, L., MCCANN, J., FALOUTSOS, C., AND POLLARD, N. 2010. Bolero: A principled technique for including bone length constraints in motion capture occlusion filling. *The ACM SIGGRAPH / Eurographics Symposium on Computer Animation*.

LI, L., MCCANN, J., POLLARD, N. S., AND FALOUTSOS, C. 2009. Dynammo: mining and summarization of coevolving sequences with missing values. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 507–516.

LIU, G. AND MCMILLAN, L. 2006. Estimation of missing markers in human motion capture. *The Visual Computer 22*, 721–728.

LOU, H. AND CHAI, J. 2010. Example-based human motion denoising. *IEEE Transactions on Visualization and Computer Graphics 16*, 870–879.

MAGNUS, J. R. AND NEUDECKER, H. 1999. *Matrix Differential Calculus with Applications in Statistics and Econometrics, 2nd Edition*. John Wiley & Sons.

MARDIA, K. V. AND DRYDEN, I. L. 1989. Shape distributions for landmark data. *Advances in Applied Probability 21,* 4, pp. 742–755.

MIN, J., CHEN, Y.-L., AND CHAI, J. 2009. Interactive generation of human animation with deformable motion models. *ACM Transactions on Graphics 29,* 1 (Dec.), 9:1–9:12.

MITCHELL, S., BOSCH, J., LELIEVELDT, B., VAN DER GEEST, R., REIBER, J., AND SONKA, M. 2002. 3-d active appearance models: segmentation of cardiac mr and ultrasound images. *IEEE Transactions Transactions on Medical Imaging 21,* 9, 1167–1178.

PARK, S. I. AND HODGINS, J. K. 2006. Capturing and animating skin deformation in human motion. *ACM Transactions on Graphics 25,* 3 (July), 881–889.

PERPERIDIS, D., MOHIADDIN, R., AND RUECKERT, D. 2004. Spatio-temporal free-form registration of cardiac mr image sequences. In *Medical Image Computing and Computer-Assisted Intervention*, C. Barillot, D. R. Haynor, and P. Hellier, Eds. Lecture Notes in Computer Science, vol. 3216. Springer Berlin / Heidelberg, 911–919.

RAO, K. AND YIP, P. 1990. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic, New York, NY.

SAFONOVA, A., HODGINS, J. K., AND POLLARD, N. S. 2004. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions on Graphics 23,* 3 (Aug.), 514–521.

SCHÖLKOPF, B., SMOLA, A. J., AND MÜLLER, K.-R. 1997. Kernel principal component analysis. In *Artificial Neural Networks - ICANN '97, 7th International Conference, Lausanne, Switzerland, October 8-10, 1997, Proceedings*. 583–588.

SIDENBLADH, H., BLACK, M. J., AND FLEET, D. J. 2000. Stochastic tracking of 3d human figures using 2d image motion. In *European Conference on Computer Vision*.

SIGAL, L., FLEET, D., TROJE, N., AND LIVNE, M. 2010. Human attributes from 3d pose tracking. In *Proceedings of the European Conference on Computer Vision*.

SUNKAVALLI, K., MATUSIK, W., PFISTER, H., AND RUSINKIEWICZ, S. 2007. Factored time-lapse video. *ACM Transactions on Graphics 26,* 3 (July), 101:1–101:10.

TENENBAUM, J. B. AND FREEMAN, W. T. 2000. Separating style and content with bilinear models. *Neural Computation 12*, 1247–1283.

THRUN, S., BURGARD, W., AND FOX, D. 2006. *Probabilistic Robotics*. Cambridge University Press.

TORRESANI, L. AND BREGLER, C. 2002. Space-time tracking. In *Proceedings of the European Conference on Computer Vision*.

TROJE, N. F. 2002. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of Vision 2,* 5 (9), 371–387.

URTASUN, R., GLARDON, P., BOULIC, R., THALMANN, D., AND FUA, P. 2004. Style-based motion synthesis. *Computer Graphics Forum 23,* 4 (Dec.), 799–812.

VASILESCU, M. A. O. AND TERZOPOULOS, D. 2004. Tensortextures: multilinear image-based rendering. *ACM Transactions on Graphics 23,* 3 (Aug.), 336–342.

VLASIC, D., BRAND, M., PFISTER, H., AND POPOVIĆ, J. 2005. Face transfer with multilinear models. *ACM Transactions on Graphics 24,* 3 (Aug.), 426–433.

WAND, M., JENKE, P., HUANG, Q., BOKELOH, M., GUIBAS, L., AND SCHILLING, A. 2007. Reconstruction of deforming geometry from time-varying point clouds. In *Fifth Eurographics Symposium on Geometry Processing*. 49–58.

WANG, H., WU, Q., SHI, L., YU, Y., AND AHUJA, N. 2005. Out-of-core tensor approximation of multi-dimensional matrices of visual data. *ACM Transactions on Graphics 24,* 3 (Aug.), 527–535.

WANG, J., FLEET, D., AND AARON, H. 2008. Gaussian process dynamical models for human motion. *IEEE Transaction on Pattern Analysis and Machine Intelligence. 30*, 283–298.

WITKIN, A. AND KASS, M. 1988. Spacetime constraints. In *Computer Graphics (Proceedings of SIGGRAPH 88)*. 159–168.