International Journal of Computer Vision
The original publication is available at www.springerlink.com.

Learning Articulated Structure and Motion

David A. Ross · Daniel Tarlow · Richard S. Zemel

Received: 21 July 2008 / Accepted: 9 Feb 2010 / Published 2 March 2010.
DOI 10.1007/s11263-010-0325-y

Abstract Humans demonstrate a remarkable ability to parse complicated motion sequences into their constituent structures and motions. We investigate this problem, attempting to learn the structure of one or more articulated objects, given a time series of two-dimensional feature positions. We model the observed sequence in terms of "stick figure" objects, under the assumption that the relative joint angles between sticks can change over time, but their lengths and connectivities are fixed. The problem is formulated as a single probabilistic model that includes multiple sub-components: associating the features with particular sticks, determining the proper number of sticks, and finding which sticks are physically joined. We test the algorithm on challenging datasets of 2D projections of optical human motion capture and feature trajectories from real videos.

Keywords structure from motion · graphical models · non-rigid motion

1 Introduction

An important aspect of analyzing dynamic scenes involves segmenting the scene into separate moving objects and constructing detailed models of each object's motion. For scenes represented by trajectories of features on the objects, structure-from-motion methods are capable of grouping the features and inferring the object poses when the features belong to multiple independently moving rigid objects. Recently, however, research has been increasingly devoted to more complicated versions of this problem, when the moving objects are articulated and non-rigid.

In this article we investigate the problem, attempting to learn the structure of an articulated object while simultaneously inferring its pose at each frame of the sequence, given a time series of feature positions. We propose a single probabilistic model for describing the observed sequence in terms of one or more "stick figure" objects. We define a "stick figure" as a collection of line segments (bones or sticks) joined at their endpoints. The structure of a stick figure—the number and lengths of the component sticks, the association of each feature point with exactly one stick, and the connectivity of the sticks—is assumed to be temporally invariant, while the angles (at joints) between the sticks are allowed to change over time. We begin with no information about the figures in a sequence, as the model parameters and structure are all learned. An example of a stick figure learned by applying our model to 2D feature observations from a video of a walking giraffe is shown in Figure 1.

Learned models of skeletal structure have many possible uses. For example, detailed, manually constructed skeletal models are often a key component in full-body tracking algorithms. The ability to learn skeletal structure could help to automate the process, potentially producing models more flexible and accurate than those constructed manually. Additionally, skeletons are necessary for converting feature point positions into joint angles, a standard way to encode motion for animation. Furthermore, knowledge of the skeleton can be used to improve the reliability of optical motion capture, per-

D.A. Ross · D. Tarlow · R.S. Zemel
University of Toronto, 10 King's College Road, Toronto, ON, M5S 3G4, Canada
E-mail: {dross, dtarlow, zemel}@cs.toronto.edu

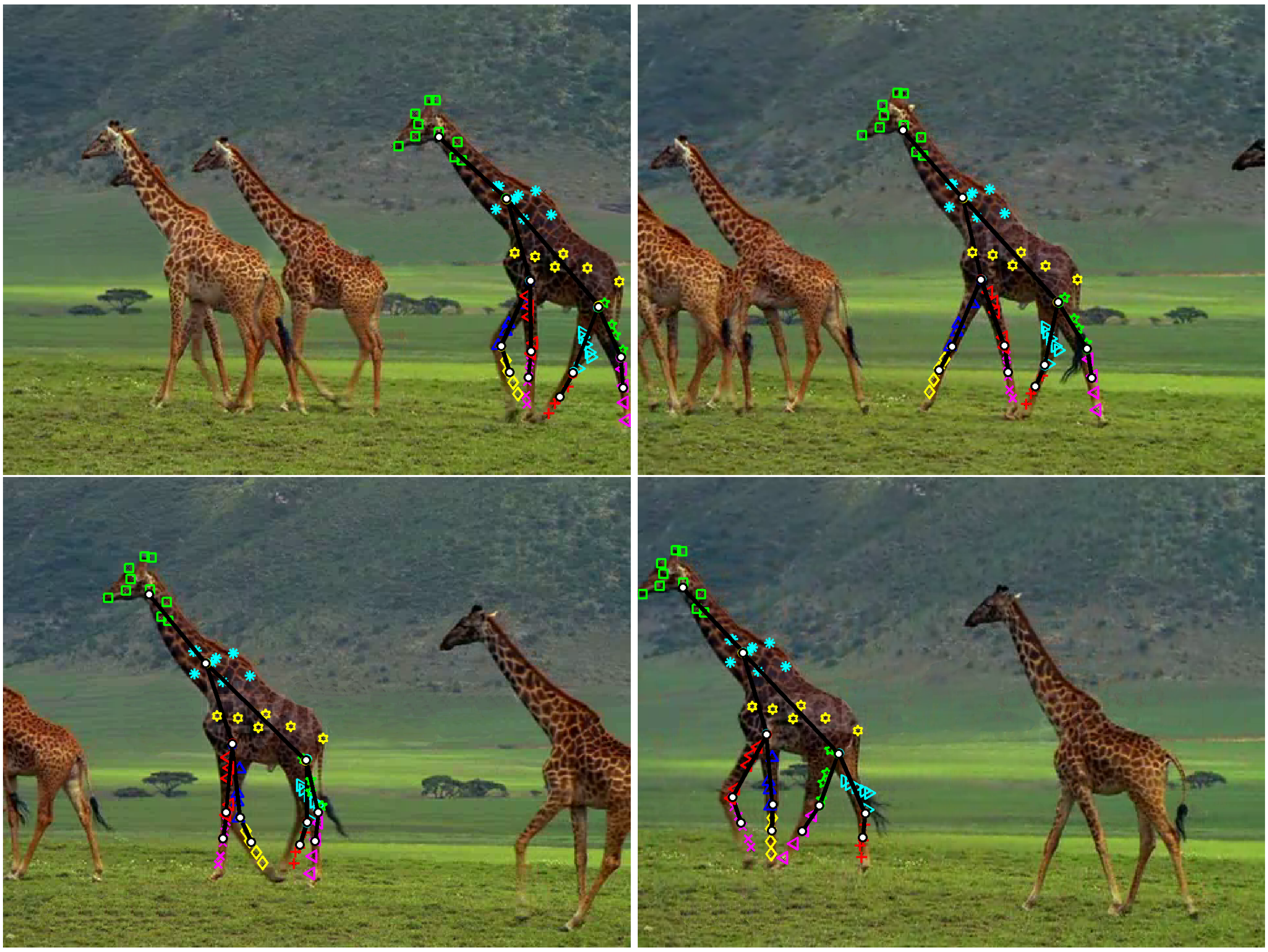

Fig. 1 Four frames from a video of a walking giraffe, with the articulated skeleton learned by our model superimposed. Each black line represents a stick, and each white circle a joint between sticks. The tracked features, which serve as the only input, are shown as coloured markers. Features associated with the same stick are assigned markers of the same colour and shape.

mitting disambiguation of marker correspondence and occlusion (Herda et al., 2001). Finally, a learned skeleton might be used as a rough prior on shape to help guide image segmentation (Bray et al., 2006).

In the following section we discuss other recent approaches to modelling articulated figures from tracked feature points. In Section 3 we formulate the problem as a probabilistic model, and in Section 4 we propose an algorithm for learning the model from data. Learning proceeds in a stage-wise fashion, building up the structure incrementally to maximize the joint probability of the model variables.

In Section 5 we test the algorithm on a range of datasets. In the final section we describe assumptions and limitations of the approach, and discuss future work.

Research presented in this paper is a continuation of Ross et al. (2008), and includes results from Ross (2008a).

2 Related Work

Humans demonstrate a remarkable ability to parse complicated motion sequences, even from apparently sparse streams of information. One field where this is readily apparent is in the study of human response to point light displays. A point light display (PLD), as depicted in Figure 2, is constructed by attaching a number of point light sources to an object, then recording (only) the positions of these lights as the object moves. The canonical example is to instrument a human's limbs and body with lights, then to record their positions as he or she performs motions such as walking, running, or swinging a golf club. PLDs have received considerable attention in psychology research (*e.g.* Johansson, 1973) due to one remarkable property. Despite the apparently limited information they contain, biological motion depicted in PLDs is almost instantly recognizable by humans. From a PLD of a person or animal, humans are

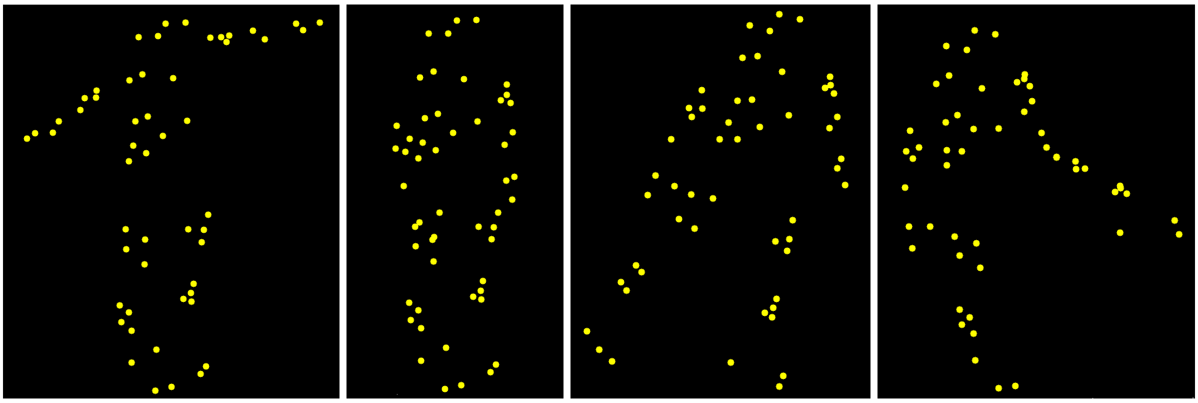

Fig. 2 A point light display of a human, in four different poses.

able to understand the structure of the display (how the lights are connected via the performer's underlying skeleton), and the motions that are performed.

Point light displays are also common in several domains of computer science research. The field of motion capture, in essence, is the study of recording and analyzing PLDs. In computer animation, PLDs obtained via motion capture are used to animate synthetic character models. Finally, in computer vision many applications choose to represent digital images sequences in terms of feature point trajectories. When the original image data is discarded, the feature points locations are equivalent to a PLD.

What follows is a discussion of three recent approaches to modelling articulated figures from tracked feature points. Each of these approaches addresses the problem from a different viewpoint: the first as structure from motion, the second as geometrical constraints in motion capture data, and the third as learning the structure of a probabilistic graphical model.

2.1 Articulated Structure From Motion

The first work we will consider is "Automatic Kinematic Chain Building from Feature Trajectories of Articulated Objects" by Yan and Pollefeys (2006b, 2008). This work builds on a history of solutions for the *structure from motion* (SFM) problem, extending them to handle articulated objects. We begin with a brief overview of this evolution, before describing the Yan and Pollefeys approach.

2.1.1 Standard Structure From Motion

Given a set of feature points observed at a number of frames, the goal of SFM is to recover the *structure*—the time-invariant relative 3D positions of the points—while simultaneously solving for the *motion*—the per-frame pose of the object(s) relative to the camera—that produced the observations. Generally, the input for SFM is assumed to be two-dimensional observations (image coordinates) of points on an inherently three-dimensional object. However most algorithms, including the ones presented here, work equally well given 3D inputs.

When the trajectories come from one rigid object (or equivalently, the scene is static and only the camera moves), and the camera is assumed to be orthographic, Tomasi and Kanade (1992) have shown that structure and motion can recovered by using the singular value decomposition (SVD) to obtain a low-rank factorization of the matrix of feature point trajectories.

Suppose we are given a matrix $\mathbf{W}$ where each column contains the x and y image coordinates of one of the observed points, at all time frames. Thus, given P points and F frames, the size of $\mathbf{W}$ is $2F \times P$ (or $3F \times P$ for three-dimensional observations). Considering the generative process that produced the observations (and disregarding noise), $\mathbf{W}$ is the product of a motion matrix and a structure matrix,

$$\mathbf{W} = \mathbf{MS},$$

both of which are rank 4. The structure $\mathbf{S}$ is a $4 \times P$ matrix containing the time-invariant (homogeneous) 3D coordinates of the points. At each frame f, the observations are produced by applying a rigid-body motion—a rotation $\mathbf{R}_f$ and a translation $\mathbf{t}_f$—to $\mathbf{S}$, and projecting the points onto the image plane:

$$\begin{bmatrix} x_{f,1} & \dots & x_{f,P} \\ y_{f,1} & \dots & y_{f,P} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_f & \mathbf{t}_f \end{bmatrix} \mathbf{S}.$$

Hence, $\mathbf{M}$ is formed by stacking the first two rows of each of these F motion matrices. From $\mathbf{W}$, $\mathbf{M}$ and $\mathbf{S}$

can be recovered by taking the singular value decomposition[1]:

$$\mathbf{W} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top \quad \Rightarrow \quad \mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}^{\frac{1}{2}} \qquad \mathbf{S} = \boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{V}^\top.$$

In practice, feature trajectories will be contaminated by noise, giving $\mathbf{W}$ a rank larger than 4. In this case Tomasi and Kanade suggest retaining only the columns of $\mathbf{U}$, $\boldsymbol{\Sigma}$ and $\mathbf{V}$ corresponding to the four largest singular values, which is the optimal rank-4 approximation to W (under squared error).

Despite the elegance and popularity of this solution, Tomasi and Kanade (1992) assume a rather unrealistic camera model—scaled orthography—for the projection of three-dimensional points down to two dimensions. As such, this does not represent a complete solution to rigid-body SFM.[2] However, when the input consists of three-dimensional points (*e.g.* obtained from a motion capture system), scaled orthography is perfectly reasonable assumption.

2.1.2 Multibody SFM

Recovering structure and motion when the scene contains multiple objects moving independently is more challenging. Consider the case in which the point trajectories arise from two independent rigid objects. If the columns of $\mathbf{W}$ are sorted so that all points from object 1 come first, and the points from object 2 come second, the low-rank factorization can be written as follows:

$$\mathbf{W} = \mathbf{M}\mathbf{S} = \begin{bmatrix}\mathbf{M}_1 \; \mathbf{M}_2\end{bmatrix} \begin{bmatrix}\mathbf{S}_1 & 0 \\ 0 & \mathbf{S}_2\end{bmatrix}. \tag{1}$$

In this case the ranks of the motion and structure matrices (and hence, of $\mathbf{W}$) have increased to 8, or 4× the number of objects. If the grouping of point trajectories into objects was known, the structure and motion of each object, $\mathbf{M}_i$ and $\mathbf{S}_i$, could be recovered independently, using the method described earlier. The problem now becomes, how to group the points?

The solution proposed by Costeira and Kanade (1996, 1998) involves considering what they term the *shape-interaction matrix*, $\mathbf{Q} \equiv \mathbf{V}\mathbf{V}^\top$. When the columns of $\mathbf{W}$ are correctly sorted, as in (1), $\mathbf{Q}$ assumes a distinctive block-diagonal structure[3]

$$\mathbf{Q} \equiv \mathbf{V}\mathbf{V}^\top = \mathbf{S}^\top\boldsymbol{\Sigma}^{-1}\mathbf{S} = \begin{bmatrix}\mathbf{S}_1{}^\top\boldsymbol{\Sigma}_1^{-1}\mathbf{S}_1 & 0 \\ 0 & \mathbf{S}_2{}^\top\boldsymbol{\Sigma}_2^{-1}\mathbf{S}_2\end{bmatrix},$$

where $\mathbf{V}$ and $\boldsymbol{\Sigma}$ again arise from the SVD of $\mathbf{W}$. Regardless of the sorting of the points, $\mathbf{Q}_{i,j}$ is nonzero if points i and j are part of the same rigid object, and 0 otherwise. The shape-interaction matrix has the advantage of being invariant to object motion, image scale, and choice of coordinate system.

Costeira and Kanade suggest that grouping point trajectories can now be accomplished by reordering the points to make $\mathbf{Q}$ block-diagonal. This problem, however, is NP-complete, thus the greedy algorithm they propose obtains only sub-optimal solutions. Interestingly, $\mathbf{Q}$ can be interpreted as a pairwise affinity matrix. In fact, $\mathbf{V}\mathbf{V}^\top$ is simply a weighted version of the inner product matrix $\mathbf{W}^\top\mathbf{W}$. This interpretation suggests that other ways of normalizing the shape-interaction matrix are possible, and that points could be grouped by any clustering algorithm which takes as input an affinity matrix, such as spectral clustering (Shi and Malik, 2000; Culverhouse and Wang, 2003; Weiss, 1999) or Affinity Propagation (Frey and Dueck, 2007).

The primary disadvantage to this approach is that the shape-interaction matrix is highly sensitive to noise in the observations (Gruber and Weiss, 2004). First of all, in the presence of noise $\mathbf{Q}_{i,j}$ is no longer zero when i and j come from different objects. Furthermore, computing $\mathbf{Q}$ requires knowing the rank of $\mathbf{W}$, which is the number of columns of $\mathbf{V}$ retained after the SVD. (Note that if we retain all columns of $\mathbf{V}$, then $\mathbf{Q} = \mathbf{V}\mathbf{V}^\top = \mathbf{I}$.) In the simplest case, this rank is 4× the number of objects, but it can be less when an object does not express all its degrees of mobility. Noise makes the rank of $\mathbf{W}$ difficult to determine, requiring an often-unreliable analysis of the eigenspectrum. One approach for dealing with this, in the presence of noise, is described by Gear (1998).

2.1.3 Probabilistic SFM

Gruber and Weiss (2003) have noted that the approach of Tomasi and Kanade can be reinterpreted as a probabilistic graphical model, specifically *factor analysis.* In factor analysis, each observed data vector is generated by taking a linear combination of a set of basis vectors, and adding diagonal-covariance Gaussian noise. In the context of single-body SFM each row $\mathbf{w}_i$ of $\mathbf{W}$, the x or

[1] In most cases, although the columns of $\mathbf{U}$ and $\mathbf{V}$ span the correct subspaces, they are actually linear transformations of the columns of $\mathbf{M}$ and $\mathbf{S}$ respectively. This can be corrected by solving, via nonlinear optimization, for a transformation that satisfies the constraints on the rotational components of $\mathbf{M}$ Tomasi and Kanade (1992).

[2] Solutions based on the more-realistic projective camera, perhaps using the above method as an initialization, can be obtained via an algorithm for *bundle adjustment* (Hartley and Zisserman, 2003).

[3] $\mathbf{V}\mathbf{V}^\top$ is also block-diagonal if we allow $\mathbf{V}^\top$ to more generally be an invertible linear transformation of the true structure: $\mathbf{S} = \mathbf{A}^{-1}\boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{V}^\top$ (Costeira and Kanade, 1998).

y coordinates of all feature points in one frame, is generated by taking a linear combination $\mathbf{m}_i$ of the rows of $\mathbf{S}$. Including a standard Gaussian prior on the rows of the motion matrix produced the following model:

$$\begin{aligned}
\mathbf{w}_i &= \mathbf{m}_i\mathbf{S} + \mathbf{n}_i, \quad \text{where} \\
\mathbf{n}_i &\sim \mathcal{N}(\mathbf{0}, \operatorname{diag}(\boldsymbol{\psi}_i)) \\
\mathbf{m}_i &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}).
\end{aligned}$$

Structure and motion can be recovered by fitting the model using the standard Expectation Maximization (EM) algorithm for factor analysis (Ghahramani and Hinton, 1996a). An advantage of this formulation is that missing observations can be dealt with easily; setting the corresponding variances to ∞ has the effect of eliminating them from the calculations (Gruber and Weiss, 2003).

Another key innovation of the Gruber and Weiss approach is to assume temporal coherence of motions. This allows them to take advantage of the fact, when estimating motions, that motions for adjacent frames should be similar. In the graphical model, temporal coherence is incorporated easily through the use of a Markov chain prior (a Kalman filter) over the latent motion variables. The result is closely related to the EM algorithm for learning linear dynamical systems (Ghahramani and Hinton, 1996b).

Multibody factorization

The probabilistic approach has also been extended to handle multiple independent rigid objects (Gruber and Weiss, 2004). Structure and motion are modeled in much the same way as (Costeira and Kanade, 1996): one independent factor analyzer of dimension 4 for each object. However, the approach of Gruber and Weiss to grouping point trajectories is quite different.

Instead of grouping points by clustering a pairwise affinity matrix, Gruber and Weiss incorporate additional discrete latent variables that assign each of the points to one of the motions. With this addition, the grouping, together with the structures and motions, can be estimated jointly using EM. This provides a distinct advantage over the method of Costeira and Kanade which, once it has grouped the points, is unable to reestimate the grouping based on subsequent information. Although fitting with EM often leads to local minima, in the presence of noise it outperforms Costeira and Kanade.

The core of this model is the same as *Multiple Cause Factor Analysis* (Ross and Zemel, 2006), independently proposed for simultaneous segmentation and appearance modelling of images.

2.1.4 Articulated Structures

The motion of an articulated object can be described as a collection of rigid motions, one per part, with the added constraint that the motions of connected parts must be spatially coherent. Yan and Pollefeys (2005a) have shown that this constraint causes the motion subspaces of two connected objects to intersect, making them linearly dependent. In particular, for each pair of connected parts, the motion subspaces share one dimension (translation) if they are joined at a point and two dimensions (translation and one angle of rotation) if they are joined at an axis of rotation. As a result of this dependence, the method of Costeira and Kanade (1996) for grouping points is no longer applicable.

To illustrate this, consider two parts that are connected by a rotational joint. Without loss of generality the shape matrices of the objects, $\mathbf{S}_1$ and $\mathbf{S}_2$ (dropping the homogeneous coordinate) can be adjusted to place this joint at the origin. Now, because the objects are connected at the joint, at each frame the translation components of their motions must be identical. Thus the ranks of $\mathbf{W}$, $\mathbf{M}$, and $\mathbf{S}$ have been reduced to at most 7 (Yan and Pollefeys, 2005a,b).

$$\mathbf{W} = \mathbf{MS} = \begin{bmatrix}\mathbf{R}_1 & \mathbf{R}_2 & \mathbf{t}\end{bmatrix} \begin{bmatrix}\mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_2 \\ \mathbf{1} & \mathbf{1}\end{bmatrix}$$

From this equation, we can see that the off-diagonal blocks of the shape interaction matrix, $\mathbf{V}\mathbf{V}^\top = \mathbf{S}^\top\boldsymbol{\Sigma}^{-1}\mathbf{S}$, are no longer zero, so clustering it will not effect the grouping of point trajectories.

Recognizing this, Yan and Pollefeys (2006a,b, 2008) propose an alternative affinity matrix to use for grouping points, and an approach for recovering the full articulated structure and motion of the sequence. Their method consists of four key steps: (1) segmenting the feature point trajectories into a number of rigid parts, (2) computing an affinity measure indicating the likelihood that each pair of parts is connected by a joint, (3) obtaining a spanning tree that connects parts while maximizing affinity, and finally (4) solving for the locations of joints.

When specifying the affinity between a pair of features, instead of relying on the dot product (angle) between rows $\mathbf{v}_i$ and $\mathbf{v}_j$ of $\mathbf{V}$, they suggest that a more robust measure could be obtained by comparing the subspace spanned by $\mathbf{v}_i$ and its nearest neighbors with that of $\mathbf{v}_j$ and its neighbors. Given these two subspaces, they compute the principal angles $\theta_1, \ldots, \theta_m$ between them, and define the affinity between i and j to be

$$\exp\left(-\sum_n \sin^2(\theta_n)\right).$$

The affinity is used as input for spectral clustering (Shi and Malik, 2000), thereby producing a grouping of feature point trajectories.

Principal angles are also used as a basis for learning the articulated structure. Noting that the four-dimensional motions (and hence shape subspaces) of parts connected by an articulated joint will have at least one dimension in common, at least one of the principal angles between the parts should be zero. Using minimum principal angle as an edge weight, Yan and Pollefeys set up a fully connected graph and solve for the articulated structure by finding the minimum spanning tree. The method can be extended to finding multiple articulated objects in a scene simply by disallowing edges with weight exceeding a manually specified threshold.

Finally, the locations of the joints can be obtained from the intersections of the motion subspaces of connected parts, as described in (Yan and Pollefeys, 2005a)

Due to the reliance on estimating subspaces, this method requires each body part to have at least as many feature points as the dimensionality of its motion subspace. (In practice, segmenting two independent objects requires at least five points per object, using at least three neighbors to estimate the local subspace, in the noise-free case.) However, relying on subspaces provides an additional advantage: the approach is able to deal with non-rigid body parts–single subpaces with rank higher than four.

Alternative approaches to articulated structure from motion are presented by Tresadern and Reid (2005) and Sminchisescu and Triggs (2003).

2.2 Geometric Analysis of Motion Capture Data

When observations are the 3D world locations of feature points, rather than 2D projections, the geometry of recovering 3D skeletal structure becomes easier. Based on a simple analysis of the distance between feature points, and following roughly the same four steps as Yan and Pollefeys (2006b), Kirk et al. (2005) are able to automatically recover skeletal structure from motion capture data. This is an improvement upon existing methods of fitting a skeleton to motion capture data (*e.g.* Silaghi et al., 1998; Abdel-Malek et al., 2004), which often require a user to manually associate markers with positions on a generic human skeleton.

The key property motivating the approach of Kirk et al. (2005) is, if two feature points are attached to the same rigid body part, then the distance between these points is constant. Furthermore, if two body parts are connected by a rotational joint, then the distances between the joint and the points belonging to both parts should also be constant. Feature points are grouped, to obtain body parts, by computing the standard deviation of the distance between each pair of points and using that as the (negative or inverse) affinity matrix for spectral clustering (Ng et al., 2002). The number of body parts is chosen manually, or again by analysis of the eigenspectrum.

When determining the skeletal connectivity of the body parts, Kirk et al. define a *joint cost*, which is the average variance in the distance from a putative joint to each of the points in the two parts it connects. Joint costs are computed for each pair of body parts. Evaluating the joint cost requires non-linear conjugate gradient minimization, but also returns the optimal joint location at each frame. Note that joint locations can be estimated as long as one stick has at least two observed markers and the other stick has at least one. Finally, the skeletal structure is obtained by running a minimum spanning tree algorithm, using the joint costs as edge weights.

This method has a few drawbacks. First, it is only able to work on 3D observations–none of the distance constraints it relies upon apply when points are projected into 2D. Second, like (Yan and Pollefeys, 2006b), it consists of a sequence of steps without feedback or reestimation. Finally, beyond computing the positions of joints in each frame, the method does not produce a time-invariant model of structure or a set of motion parameters. As such, filling in missing observations or computing joint angles would require further processing.

One further caveat regarding this method is that, contrary to the images included in the paper (Kirk et al., 2005), its output is not actually a “stick figure”—a collection of line segments (bones or sticks) joined at their endpoints. Instead, in the learned graph, parts of the body are nodes and joints are edges, which is a more-difficult structure to visualize.

2.3 Learning a Graphical Model Structure

Another approach to the analysis of PLDs is to model the relationships between feature point locations with a probabilistic graphical model. In this setting, recovering the skeleton is a matter of learning the graph structure and parameters of the model. This is the approach taken by Song et al. (2001, 2003), with a goal of automatically detecting human motion in cluttered scenes.

Treating each frame as an independent, identically distributed sample, Song et al. construct a model in which each variable node represents the position and velocity of one of the observed points. No latent variables are included, instead each feature point is treated as a unique part of the body. This presumes a much sparser set of features than (Yan and Pollefeys, 2006b) and (Kirk et al., 2005), which require each part to give rise to multiple feature point trajectories. The set of graphs considered is restricted to a particular class, *decomposable triangulated graphs*, in which all cliques are of size three. The limitation placed on the structure ensures that, although these graphs are more complicated than trees, efficient exact inference is still possible. The clique potentials, over triplets of nodes, are multivariate Gaussian distributions over the velocities and relative positions of the parts.

The maximum likelihood (ML) graph is the one that minimizes the empirical entropy of each feature point given its parents. Unfortunately no tractable algorithm exists for computing the ML graph, so Song et al. propose the following approximate greedy algorithm. Assuming all nodes are initially disconnected, choose the first edge in the graph by connecting the nodes B and C that minimize the joint entropy $h(B, C)$. Then, for all possible ways of choosing an pair of connected parents (B, C) already in the graph, find the child A that minimizes the conditional entropy $h(A|B, C)$ and connect it to the graph. Continue connecting child nodes to the graph until it has reached the desired size, or the entropy of the best putative child exceeds a threshold. The cost of this algorithm is $O(n^4)$, where n is the number of feature points.

Note that if the class of graphical models considered is restricted to trees, the graph structure can be found efficiently, by calculating the mutual information between each pair of body parts and solving for the maximum spanning tree (Taycher et al., 2002; Song et al., 2003).

Song et al. further extend their approach to handle cluttered scenes, obtained by automatically tracking features in video. Since the results of tracking are invariably noisy, this requires solving the correspondence problem at each frame (identifying which feature points are body parts, which come from the background, and which body parts are occluded). Learning can now be accomplished via an EM-like algorithm, which alternates optimizing the feature correspondence with learning the graphical model structure and parameters.

Although the authors are able to show some interesting results, this approach has a number of drawbacks. First, learned models are specific to the 3D position and orientation of the subject, accounting only for invariance to translation parallel to the image plane. Thus a model trained on a person walking from left to right is unable to detect a person walking from right to left (Song et al., 2003). Secondly, a single time-invariant model is learned on the data from all frames, thereby confounding structure and motion. Instead of trying to model these two latent factors separately, the presence of motion serves only to increase uncertainty in the graphical model.

3 Model

Here we formulate a probabilistic graphical model for sequences generated from articulated skeletons. By fitting this model to a set of feature point trajectories (the observed locations of a set of features across time), we are able to parse the sequence into one or more articulated skeletons and recover the corresponding motion parameters for each frame. The observations are assumed to be 2D, whether tracked from video or projected from 3D motion capture, and the goal is to learn skeletons that capture the full 3D structure. Fitting the model is performed entirely via unsupervised learning; the only inputs are the observed trajectories, with manually tuned parameters restricted to a small set of thresholds on Gaussian variances.

The observations for this model are the locations $\mathbf{w}_p^f$ of feature points p in frames f. A discrete latent variable $\mathbf{R}$ assigns each point to one of S sticks. Each stick s consists of a set of time-invariant 3D local coordinates $\mathbf{L}_s$, describing the relative positions of all points belonging to the stick. $\mathbf{L}_s$ is mapped to the observed world coordinate system by a different motion matrix $\mathbf{M}_s^f$ at every frame f (see Figure 3). For example, in a noiseless system, where $r_{p,1} = 1$, indicating that point p has been assigned to stick 1, $\mathbf{M}_1^f \mathbf{l}_{1,p} = \mathbf{w}_p^f$.

If all of the sticks are unconnected and move independently, then this model essentially describes multibody SFM (Costeira and Kanade, 1998; Gruber and Weiss, 2004), or equivalently an instance of *Multiple Cause Factor Analysis* (Ross and Zemel, 2006). However, for an articulated structure, with connections between sticks, the stick motion variables are not independent (Yan and Pollefeys, 2006a). Allowing connectivity between sticks makes the problems of describing the constraints between motions and inferring motions from the observations considerably more difficult.

To deal with this complexity, we introduce variables to model the connectivity between sticks, and the (unobserved) locations of stick endpoints and joints in each frame. Every stick has two endpoints, each of which is assigned to exactly one *vertex*. Each vertex can correspond to one or more stick endpoints (vertices assigned two or more endpoints are joints). We will let $\mathbf{k}_i$ specify the coordinates of endpoint i relative to the local coordinate system of its stick, $s(i)$, and $\mathbf{v}_j^f$ and $\mathbf{e}_i^f$ represent the world coordinate location of vertex j and endpoint

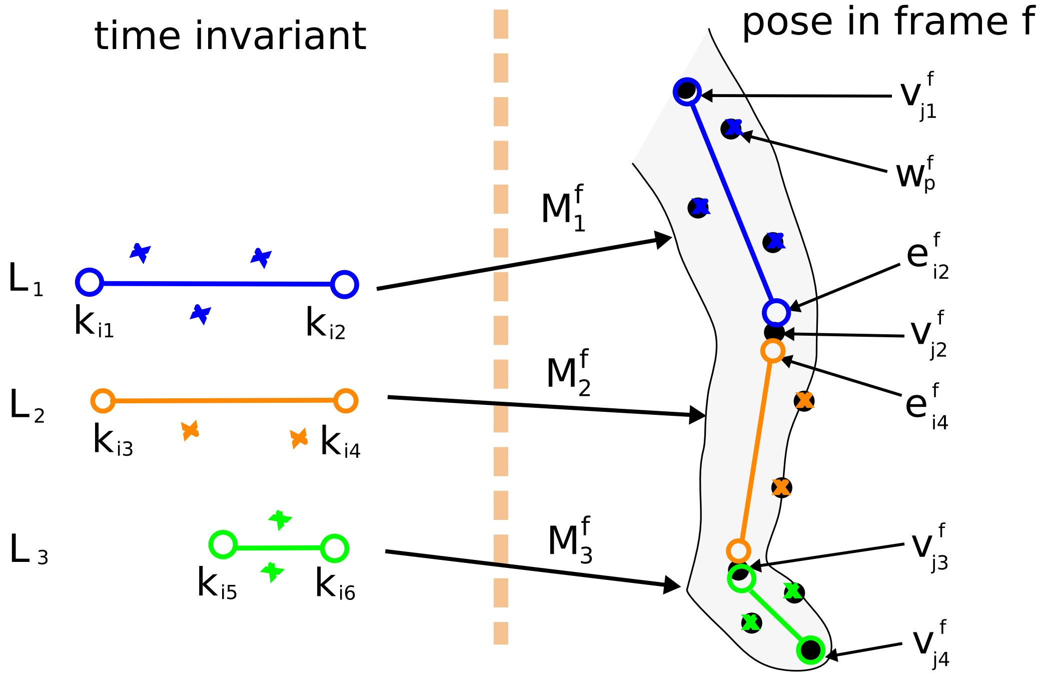

Fig. 3 The generative process for the observed feature positions, and the imputed positions of the stick endpoints. For each stick, the relative positions of its feature points and endpoints are represented in a time-invariant local coordinate system (left). For each frame in the sequence (right), motion variables attempt to fit the observed feature positions (*e.g.* $\mathbf{w}_P^f$) by mapping local coordinates to world coordinates, while maintaining structural cohesion by mapping stick endpoints to inferred vertex (joint) locations.

i in frame f, respectively. Again, in a noiseless system, $\mathbf{e}_i^f = \mathbf{M}_{s(i)}^f \mathbf{k}_i$ for every frame f. Noting the similarity between the $\mathbf{e}_i^f$ variables and the observed feature positions $\mathbf{w}_p^f$, these endpoint locations can be interpreted as a set of pseudo-observations, inferred from the data rather than directly observed.

Vertices are used to enforce a key constraint: for all the sticks that share a given vertex, the motion matrices should map their local endpoint locations to a consistent world coordinate. This restricts the range of possible motions to only those resulting in appropriately connected figures. For example, in Figure 3, endpoint 2 (of stick 1), is connected to endpoint 4 (of stick 2); both are assigned to vertex 2. Thus in every frame f both endpoints should map to the same world location, the location of the knee joint, *i.e.* $\mathbf{v}_2^f = \mathbf{e}_2^f = \mathbf{e}_4^f$.

The utility of introducing these additional variables is that, given the vertices $\mathbf{V}$ and endpoints $\mathbf{E}$, the problem of estimating the motions and local geometries ($\mathbf{M}$ and $\mathbf{L}$) factorizes into S independent structure-from-motion problems, one for each stick. Latent variable $\mathbf{g}_{i,j} = 1$ indicates that endpoint i is assigned to vertex j; hence $\mathbf{G}$ indirectly describes the connectivity between sticks. The assumed generative process for the feature observations and the vertex and endpoint pseudo-observations is shown in Figure 3, and the corresponding probabilistic model is shown in Figure 4.

The complete joint probability of the model can be decomposed into a product of two likelihood terms, one for the true feature observations and the second for the endpoint pseudo1-observations, and priors over the remaining variables in the model:

$$\mathbb{P} = \mathrm{P}(\mathbf{W}|\mathbf{M},\mathbf{L},\mathbf{R})\mathrm{P}(\mathbf{E}|\mathbf{M},\mathbf{K},\mathbf{V},\boldsymbol{\phi},\mathbf{G}) \qquad (2)$$
$$\mathrm{P}(\mathbf{V})\mathrm{P}(\boldsymbol{\phi})\mathrm{P}(\mathbf{M})\mathrm{P}(\mathbf{L})\mathrm{P}(\mathbf{K})\mathrm{P}(\mathbf{R})\mathrm{P}(\mathbf{G})$$

Assuming isotropic Gaussian noise with precision (inverse variance) τ_w, the likelihood function is

$$\mathrm{P}(\mathbf{W}|\mathbf{M},\mathbf{L},\mathbf{R}) = \prod_{f,p,s} \mathcal{N}(\mathbf{w}_p^f|\mathbf{M}_s^f\mathbf{l}_{s,p},\tau_w^{-1}\mathbf{I})^{r_{p,s}} \qquad (3)$$

where $r_{p,s}$ is a binary variable equal to 1 if and only if point p has been assigned to stick s. This distribution captures the constraint that for feature point p, its predicted world location, based on the motion matrix and its location in the local coordinate system for the stick to which it belongs ($r_{p,s} = 1$), should match its observed world location. Note that dealing with missing observations is simply a matter of removing the corresponding factors from this likelihood expression.[4]

[4] This likelihood is applicable if the observations $\mathbf{w}_p^f$ are 2D or 3D. In the 2D case, we assume an affine camera projection. How-

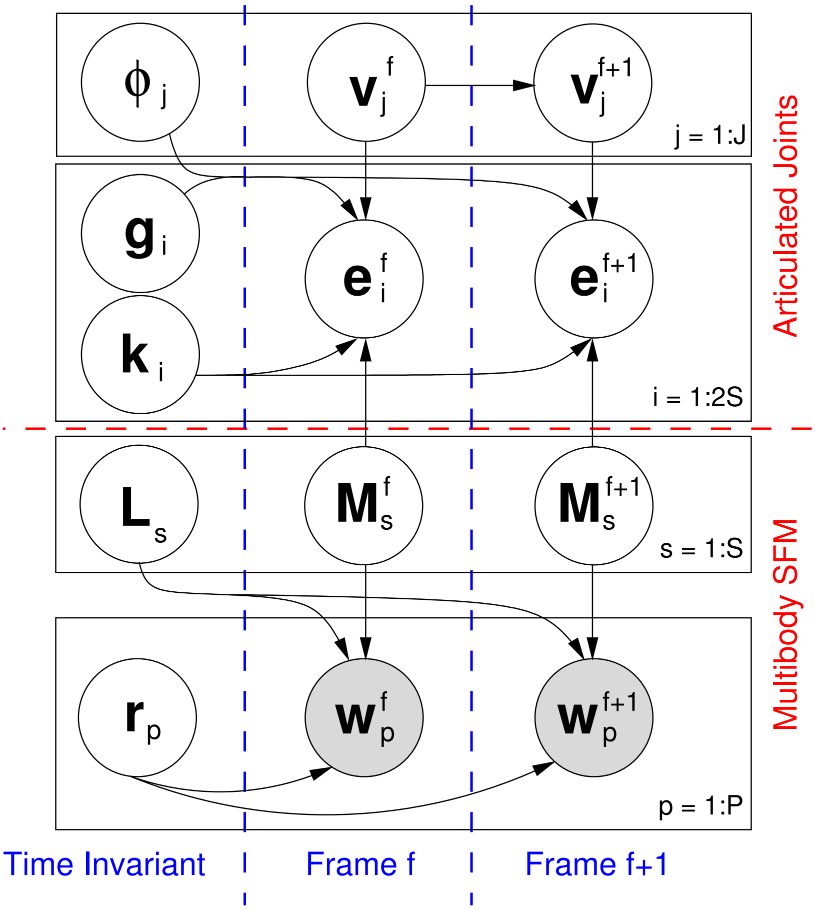

Fig. 4 The graphical model. The bottom half shows the model for independent multibody SFM; the top half describes the vertices and endpoints, which account for motion dependencies introduced by the articulated joints.

Each motion variable consists of a 2 × 3 rotation matrix $\mathbf{R}_s^f$ and a 2 × 1 translation vector $\mathbf{t}_s^f$: $\mathbf{M}_s^f \equiv [\mathbf{R}_s^f \quad \mathbf{t}_s^f]$. The motion prior $\mathrm{P}(\mathbf{M})$ is uniform, with the stipulation that all rotations be orthogonal: $\mathbf{R}_s^f \mathbf{R}_s^{f\top} = \mathbf{I}$.

We define the missing-data likelihood of an endpoint location as the product of two Gaussians, based on the predictions of the appropriate vertex and stick:

$$\mathrm{P}(\mathbf{E}|\mathbf{M}, \mathbf{K}, \mathbf{V}, \boldsymbol{\phi}, \mathbf{G}) \propto \prod_{f,i} \mathcal{N}(\mathbf{e}_i^f | \mathbf{M}_{s(i)}^f \mathbf{k}_i, \tau_m^{-1} \mathbf{I}) \prod_{f,i,j} \mathcal{N}(\mathbf{e}_i^f | \mathbf{v}_j^f, \phi_j^{-1} \mathbf{I})^{g_{i,j}} \tag{4}$$

Here τ_m is the precision of the isotropic Gaussian noise on the endpoint locations with respect to the stick, and $g_{i,j}$ is a binary variable equal to 1 if and only if endpoint i has been assigned to vertex j. The second Gaussian in this product captures the requirement that endpoints belonging to the same vertex should be coincident. Instead of making this a hard constraint, connectivity is softly enforced, allowing the model to accommodate a certain degree of non-rigidity in the underlying structure, as illustrated by the mismatch between endpoint and vertex positions in Figure 3. The vertex precision variables ϕ_j capture the degree of "play" in the joints, and are assigned Gamma prior distributions:

$$\mathrm{P}(\boldsymbol{\phi}) = \prod_j \mathrm{Gamma}(\phi_j | \alpha_j, \beta_j). \tag{5}$$

The prior on the vertex locations incorporates a temporal smoothness constraint, with precision τ_t:

$$\mathrm{P}(\mathbf{V}) = \prod_{f,j} \mathcal{N}(\mathbf{v}_j^f | \mathbf{v}_j^{f-1}, \tau_t^{-1} \mathbf{I}) \tag{6}$$

The priors for feature and endpoint locations in the local coordinate frames, $\mathbf{L}$ and $\mathbf{K}$, are zero-mean Gaussians, with isotropic precision τ_p.

$$\mathrm{P}(\mathbf{L}) = \prod_{s,p} \mathcal{N}(\mathbf{l}_{s,p} | 0, \tau_p^{-1} \mathbf{I}) \quad \mathrm{P}(\mathbf{K}) = \prod_i \mathcal{N}(\mathbf{k}_i | 0, \tau_p^{-1} \mathbf{I})$$

Finally, the priors for the variables defining the structure of the skeleton, $\mathbf{R}$ and $\mathbf{G}$, are multinomial. Each

ever, it would be possible to extend this to a projective camera by making the mean depend non-linearly on $\mathbf{M}_s^f \mathbf{l}_{s,p}$.

point p selects exactly one stick s (enforced mathematically by the constraint $\sum_s r_{p,s} = 1$) with prior probability c_s, and each endpoint i selects one vertex j (similarly $\sum_j g_{i,j} = 1$) with probability d_j:

$$\mathrm{P}(\mathbf{R}) = \prod_{p,s} (c_s)^{r_{p,s}} \qquad \mathrm{P}(\mathbf{G}) = \prod_{i,j} (d_j)^{g_{i,j}}.$$

4 Learning

Given a set of observed feature point trajectories, we propose to fit this model in an entirely unsupervised fashion, by maximum likelihood learning. Conceptually, we divide learning into two challenges: recovering the skeletal structure of the model, and given a structure, fitting the model's remaining parameters. Structure learning involves grouping the observed trajectories into a number of rigid sticks, including determining the number of sticks, as well as determining the connectivity between them. Parameter learning involves determining the local geometries and motions of each stick, as well as imputing the locations of the stick endpoints and joints — all while respecting the connectivity constraints imposed by the structure.

Both learning tasks seek to optimize the same objective function—the expected complete log-likelihood of the data given the model—using different, albeit related, approaches. Given a structure, parameters are learned using the standard variational expectation maximization algorithm. Structure learning is formulated as an "outer loop" of learning: beginning with a fully disjoint multibody SFM solution, we incrementally merge stick endpoints, at each step greedily choosing the merge that maximizes the objective. Finally the expected complete log-likelihood can be used for model comparison and selection.

A summary of the proposed learning algorithm is provided in Figure 4.2.1.

4.1 Learning the model parameters

Given a particular model structure, indicated by a specific setting of $\mathbf{R}$ and $\mathbf{G}$, the remaining model parameters are fit using the variational expectation-maximization (EM) algorithm (Neal and Hinton, 1998; Dempster et al., 1977). This well-known algorithm takes an iterative approach to learning: beginning with an initial setting of the parameters, each parameter is updated in turn, by choosing the value that maximizes the expected complete log-likelihood objective function, given the values (or expectations) of the other parameters.

The objective function—also known as the negative *Free Energy* (Neal and Hinton, 1998)—is formed by assuming a fully factorized *variational posterior* distribution $\mathbb{Q}$ over a subset of the model parameters, then computing the expectation of the model's log probability (2) with respect to $\mathbb{Q}$, plus an entropy term:

$$\mathcal{L} = \mathrm{E}_{\mathbb{Q}}[\log \mathbb{P}] - \mathrm{E}_{\mathbb{Q}}[\log \mathbb{Q}]. \tag{7}$$

For this model, we define $\mathbb{Q}$ over the variables $\mathbf{V}$, $\mathbf{E}$, and $\boldsymbol{\phi}$, involved in the world-coordinate locations of the joints. The variational posterior for $\mathbf{v}_j^f$ is a multivariate Gaussian with mean parameter $\mu(\mathbf{v}_j^f)$ and precision parameter $\tau(\mathbf{v}_j^f)$, for $\mathbf{e}_i^f$ is also a Gaussian with mean $\mu(\mathbf{e}_i^f)$ and precision $\tau(\mathbf{e}_i^f)$, and for $\boldsymbol{\phi}$ is a Gamma distribution with parameters $\alpha(\phi_j)$ and $\beta(\phi_j)$:

$$\begin{aligned}
\mathbb{Q} &= Q(\mathbf{V})\, Q(\mathbf{E})\, Q(\phi) \\
Q(\mathbf{V}) &= \prod_{f,j} \mathcal{N}(\mathbf{v}_j^f | \mu(\mathbf{v}_j^f), \tau(\mathbf{v}_j^f)^{-1}) \\
Q(\mathbf{E}) &= \prod_{f,i} \mathcal{N}(\mathbf{e}_i^f | \mu(\mathbf{e}_i^f), \tau(\mathbf{e}_i^f)^{-1}) \\
Q(\boldsymbol{\phi}) &= \prod_{j} \mathrm{Gamma}(\phi_j | \alpha(\phi_j), \beta(\phi_j)).
\end{aligned}$$

The EM update equations are obtained by differentiating the objective function $\mathcal{L}$, with respect to each parameter, and solving for the maximum given the other parameters. We now present the parameter updates, with detailed derivation of $\mathcal{L}$ and the updates appearing in (Ross, 2008b). As a reminder, the constants appearing in these equations denote: D_o the dimensionality of the observations, generally 2 but 3 will also work; F the number of observation frames; J the number of vertices; P the number of data points; S the number of sticks.

$$\begin{aligned}
\tau_w^{-1} &= \frac{\sum_{f,p,s} r_{p,s} \|\mathbf{w}_p^f - \mathbf{M}_s^f \mathbf{l}_{s,p}\|^2}{FPD_o} \\
\tau_m^{-1} &= \frac{\sum_{f,i} \|\mu(\mathbf{e}_i^f) - \mathbf{M}_{s(i)}^f \mathbf{k}_i\|^2}{2FSD_o} + \frac{\sum_{f,i} \tau(\mathbf{e}_i^f)^{-1}}{2FS} \\
\tau_t^{-1} &= \frac{\sum_{f=2}^{F} \sum_j \|\mu(\mathbf{v}_j^f) - \mu(\mathbf{v}_j^{f-1})\|^2}{(F-1)JD_o} \\
&\quad + \frac{\sum_{f,j} \tau(\mathbf{v}_j^f)^{-1} 2^{h(f)}}{(F-1)J},
\end{aligned}$$

where $h(f) = 1$ if $1 < f < F$ and 0 otherwise.

$$\begin{aligned}
\tau(\mathbf{e}_i^f) &= \sum_j g_{i,j} \frac{\alpha(\phi_j)}{\beta(\phi_j)} + \tau_m \\
\mu(\mathbf{v}_j^f) &= \Big(\frac{\alpha(\phi_j)}{\beta(\phi_j)} \sum_i g_{i,j} \mu(\mathbf{e}_i^f) \\
&\quad + [f > 1]\tau_t \mu(\mathbf{v}_j^{f-1}) + [f < F]\tau_t \mu(\mathbf{v}_j^{f+1}) \Big) \\
&\quad / \Big(\frac{\alpha(\phi_j)}{\beta(\phi_j)} \sum_i g_{i,j} + \tau_t 2^{h(f)} \Big) \\
\tau(\mathbf{v}_j^f) &= \frac{\alpha(\phi_j)}{\beta(\phi_j)} \sum_i g_{i,j} + \tau_t 2^{h(f)}
\end{aligned}$$

$$\begin{aligned}
\alpha(\phi_j) &= \alpha_j + \frac{F D_o}{2} \sum_i g_{i,j} \\
\beta(\phi_j) &= \beta_j + \frac{1}{2} \sum_{f,i} g_{i,j} \|\mu(\mathbf{e}_i^f) - \mu(\mathbf{v}_j^f)\|^2 \\
&\quad + \frac{D_o}{2} \sum_{f,i} g_{i,j} [(\tau(\mathbf{e}_i^f))^{-1} + (\tau(\mathbf{v}_j^f))^{-1}] \\
\alpha_j &= \alpha(\phi_j) \\
\beta_j &= \beta(\phi_j)
\end{aligned}$$

The update for the motion matrices is slightly more challenging due to the orthogonality constraint on the rotations. A straightforward approach is to separate the rotation and translation components of the motion and to solve for each individually. The update for translation is obtained simply via differentiation:

$$\begin{aligned}
\mathbf{M}_s^f &= \left[\mathbf{R}_s^f \ \mathbf{t}_s^f\right] \\
\mathbf{t}_{s,f} &= \Big(\tau_w \sum_p r_{p,s} (\mathbf{w}_p^f - \mathbf{R}_s^f \mathbf{l}_{s,p}) \\
&\quad + \tau_m \sum_{\{i|s(i)=s\}} (\mu(\mathbf{e}_i^f) - \mathbf{M}_s^f \mathbf{k}_{s,i}) \Big) \\
&\quad / \Big(\tau_w \sum_p r_{p,s} + 2\tau_m \Big)
\end{aligned}$$

To deal with the orthogonality constraint on $\mathbf{R}_s^f$, its update can be posed as an *orthogonal Procrustes problem* (Golub and Van Loan, 1996; Viklands, 2006). Given matrices $\mathbf{A}$ and $\mathbf{B}$, the goal of orthogonal Procrustes is to obtain the matrix $\mathbf{R}$ that minimizes $\|\mathbf{A} - \mathbf{R}\mathbf{B}\|^2$, subject to the constraint that the rows of $\mathbf{R}$ form an orthonormal basis. Computing the most likely rotation involves maximizing the likelihood of the observations (3) and of the endpoints (4), which can be written as the minimization of $\sum_p \|(\mathbf{w}_p^f - \mathbf{t}_{s,f}) - \mathbf{R}_s^f \mathbf{l}_{s,p}\|^2$ and $\sum_{\{i|s(i)=s\}} \|(\mu(\mathbf{e}_i^f) - \mathbf{t}_{s,f}) - \mathbf{R}_s^f \mathbf{k}_{s,i}\|^2$ respectively. Concatenating the two problems together, weighted by their respective precisions, allows the update of $\mathbf{R}_s^f$ to be written as a single orthogonal Procrustes problem $\operatorname{argmin}_{\mathbf{R}_s^f} \|\mathbf{A} - \mathbf{R}_s^f \mathbf{B}\|^2$, where

$$\begin{aligned}
\mathbf{A} &= \Big[\left[\sqrt{\tau_w}\, r_{p,s} (\mathbf{w}_p^f - \mathbf{t}_{s,f})\right]_{p=1..P} \ \left[\sqrt{\tau_m}\, (\mu(\mathbf{e}_i^f) - \mathbf{t}_{s,f})\right]_{\{i|s(i)=s\}} \Big] \\
\mathbf{B} &= \Big[\left[\sqrt{\tau_w}\, r_{p,s} \mathbf{l}_{s,p}\right]_{p=1..P} \ \left[\sqrt{\tau_m}\, \mathbf{k}_i\right]_{\{i|s(i)=s\}} \Big] .
\end{aligned}$$

The solution is to compute the singular value decomposition of $\mathbf{B}\mathbf{A}^\top \overset{SVD}{=} \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$, and let $\mathbf{R} = \mathbf{V}\, \mathbf{I}_{m \times n} \mathbf{U}^\top$, where m and n are the numbers of rows in $\mathbf{A}$ and $\mathbf{B}$ respectively.

Given $\mathbf{R}_s^f$ and $\mathbf{t}_s^f$, the updates for the local coordinates are:

$$\begin{aligned}
\mathbf{l}_{s,p} &= \big(\sum_f \mathbf{R}_s^{f\top} \mathbf{R}_s^f + \frac{\tau_p}{\tau_w} \mathbf{I} \big)^{-1} \sum_f \mathbf{R}_s^{f\top} (\mathbf{w}_p^f - \mathbf{t}_{s,f}) \\
\mathbf{k}_i &= \big(\sum_f \mathbf{R}_{s(i)}^f{}^\top \mathbf{R}_{s(i)}^f + \frac{\tau_p}{\tau_m} \mathbf{I} \big)^{-1} \sum_f \mathbf{R}_{s(i)}^f{}^\top (\mu(\mathbf{e}_i^f) - \mathbf{t}_{s(i)}^f)
\end{aligned}$$

The final issue to address for EM learning is initialization. Many ways to initialize the parameters are possible; here we settle on one simple method that produces satisfactory results. The motions and local coordinates, $\mathbf{M}$ and $\mathbf{L}$, are initialized by solving SFM independently for each stick (Tomasi and Kanade, 1992). The vertex locations are initialized by averaging the observations of all sticks participating in the joint: $\mu(\mathbf{v}_j^f) = (\sum_{i,p} g_{i,j}\, r_{p,s(i)}\, \mathbf{w}_p^f) / (\sum_{i,p} g_{i,j}\, r_{p,s(i)})$. The endpoints are initially coincident with their corresponding vertices, $\mu(\mathbf{e}_i^f) = \sum_j g_{i,j}\, \mu(\mathbf{v}_j^f)$, and the $\mathbf{K}$s by averaging the backprojected endpoint locations: $\mathbf{k}_i = \frac{1}{F} \sum_f \mathbf{R}_{s(i)}^f{}^\top (\mu(\mathbf{e}_i^f) - \mathbf{t}_{s(i)}^f)$. All precision parameters are initialized to constant values, as discussed in Section 5.1.

4.2 Learning the skeletal structure

Structure learning in this model entails estimating the assignments of feature points to sticks (including the number of sticks), and the connectivity of sticks, expressed via the assignments of stick endpoints to vertices. The space of possible structures is enormous. We therefore adopt an incremental approach to structure learning: beginning with a fully disconnected multibody-SFM model, we greedily add joints between sticks by merging vertices. After each merge the model parameters are updated via EM, and the assignments of observations to sticks are resampled. After performing the desired number of merges, model selection—that is, choosing the optimal number of joints—is guided by comparing the expected complete log-likelihood of each model.

The first step in structure learning involves hypothesizing an assignment of each observed feature trajectories to a stick. This is accomplished by clustering the trajectories using the *Affinity Propagation* algorithm of Frey and Dueck (2007). Affinity Propagation takes as input an affinity matrix, for which we supply the affinity measure from Yan and Pollefeys (2006a,b, 2008) as presented in Section 2.1.4 (or for 3D data, Kirk et al. (2005) discussed in 2.2). During EM parameter learning, the stick assignments $\mathbf{R}$ are resampled every 10 iterations using the posterior probability distribution

$$\mathrm{P}(r_{p,s}) \propto c_s \exp\left(-\frac{\alpha_w}{2}\sum_f \|\mathbf{w}_p^f - \mathbf{M}_s^f \mathbf{l}_{s,p}\|^2\right)$$
$$\text{s.t.} \sum_{s'} r_{p,s'} = 1.$$

Instead of relying only on information available before model fitting begins (Costeira and Kanade, 1998; Kirk et al., 2005; Yan and Pollefeys, 2006b), resampling of stick assignments allows model probability to be improved by leveraging current best estimates of the model parameters.

The second step of structure learning involves determining which sticks endpoints are joined together. As discussed earlier, connectivity is captured by assigning stick endpoints to vertices; each endpoint must be associated to one vertex, and vertices with two or more endpoints act as articulated joints. (Valid configurations include only cases in which endpoints of a given stick are assigned to different vertices.) We employ an incremental greedy scheme for inferring this graphical structure $\mathbf{G}$, beginning from an initial structure that contains no joints between sticks. Thus, in terms of the model, we start with $J = 2S$ vertices, one per stick-endpoint, so $g_{i,j} = 1$ if and only if $j = i$. Given this initial structure, parameters are fit using variational EM.

A joint between sticks is introduced by merging together a pair of vertices. The choice of vertices to merge is guided by our objective function $\mathcal{L}$. At each stage of merging we consider all valid pairs of vertices, putatively joining them and estimating (via 20 iterations of EM) the change in log-likelihood if this merge were accepted. The merge with the highest log-likelihood is performed, by modifying $\mathbf{G}$ accordingly, and the model parameters are re-optimized with 200 additional iterations of EM, including resampling of the stick assignments $\mathbf{R}$. This process is repeated until no valid merges remain, or the desired maximum number of merges has been reached.

4.2.1 Computational Cost

By examining the updates presented in Section 4.1, in can be seen that the cost of each iteration of EM parameter learning scales linearly in the following quantities: F the number of frames, J the number of joints, P the number of observed feature point trajectories, and S the number of sticks. (Note that since the number of rows in $\mathbf{A}$ and $\mathbf{B}$ are fixed, each orthogonal Procrustes update of $\mathbf{R}_s^f$ has a cost that is linear in P—the initial multiplication $\mathbf{AB}^\top$—in addition to a constant-cost SVD and final multiplication.)

Each stage of greedy merging requires computing the expected log-likelihood for all of the possible pairs of vertices to be merged. The number of possible merges scales with $O(J^2)$, which, since $J = 2S$ during the first stage, can be as high as $4S^2$. In practice, however, it is possible to reduce the number that must be considered. Savings can be obtained by noting the symmetry of the merge operation, reducing the number of unique merges by a factor of two, as well as by disallowing self-merges between the two endpoints of a stick. A less obvious savings can be realized by avoiding duplication when merging with a stick that has two free endpoints, since the change in probability from merging to either of these otherwise unconstrained endpoints will be identical. During the initial stage, when the structure contains no joints, this reduces the number of unique merges by an additional factor of four. During later stages, there are fewer possible merges to consider since J, the number of vertices, decreases by one for each stage, and our previously mentioned restriction—that the endpoints of a stick cannot be assigned to the same vertex—eliminates a greater proportion of potential merges.

In our experiments these optimizations are sufficient to yield acceptable runtimes, however given much larger models the number of possible merges could be reduced to $O(J)$ by allowing each stick to merge with only a fixed number (*e.g.* five) of its nearest neighbors. It may also be possible to achieve further savings through caching—approximating the expected change in log-likelihood of a merge with its value from the previous stage, without recomputing (Ross et al., 2007).

5 Experimental Results and Analysis

We now present results of the proposed algorithm on a range of different feature point trajectory datasets. This includes data obtained by automatically tracking features in video, from optical motion capture (both 2D and 3D), as well as a challenging artificially generated sequence. In each experiment a model was learned on

1. Obtain an initial grouping $\mathbf{R}$ by clustering the observed trajectories using Affinity propagation. Initialize $\mathbf{G}$ to a fully disconnected structure.
2. Optimize the parameters $\mathbf{M}$, $\mathbf{L}$, $\mathbf{K}$, $\mathbf{V}$, $\boldsymbol{\phi}$, $\mathbf{E}$, using 200 iterations of the variational EM updates, resampling $\mathbf{R}$ every 10 iterations.
3. For all vertex-pair merges, estimate gain resulting from the proposed structure by updating the parameters with 20 EM iterations and noting the change in expected log-probability.
4. Choose the merge with the largest gain, modifying $\mathbf{G}$ accordingly. Re-optimize parameters using another 200 EM iterations, resampling $\mathbf{R}$ every 10^{th}.
5. Go to Step 3 and repeat. Exit when there are no more valid merges, or the maximum number of merges has been reached.

Fig. 5 A summary of the learning algorithm

the first 70% of the sequence frames, with the remaining 30% held out as a test set used to measure the model's performance. Learning was performed using the algorithm summarized in Figure 4.2.1, with greedy merging continuing (generally) until no valid merges remained. After each stage of merging, we saved the learned model and corresponding expected complete log-likelihood—the objective function learning maximizes. The likelihoods were plotted for comparison, and used to select the optimal model.

The learned model's performance was evaluated based on its ability to impute (reconstruct) the locations of missing observations. For each test sequence we generated a set of missing observations by simulating an occluder that sweeps across the scene, obscuring points as it passes. We augmented this set with an additional 5% of the observations chosen to be "missing at random", to simulate drop-outs and measurement errors, resulting in a overall occlusion rate of 10-15%. The learned model was fit to the un-occluded points of the test sequence, and used to predict the location of the missing points. Performance was measured by computing the root-mean-squared error between the predictions and the locations of the heldout points. We compared the performance of our model against similar prediction errors made by single-body and multibody structure from motion models.

This section begins with a brief analysis of the effect of precision parameters during learning, followed by experimental results on five datasets: a video of an excavator, a video of a walking giraffe, 2D feature trajectories obtained from human motion capture, an synthetic dataset of a jointed ring, and an additional set of human motion data in 3D. Finally we conclude with a brief comparison against two related methods (Yan and Pollefeys, 2008; Kirk et al., 2005).

Videos of the experimental results may be found at http://www.cs.toronto.edu/~dross/articulated/.

5.1 Setting Precision Parameters During Learning

As presented, the model contains a number of precision parameters to be determined during learning: τ_w, τ_m, τ_t, τ_p, $\tau(\mathbf{v}_j^f)$, $\tau(\mathbf{e}_i^f)$, as well as the parameters of the prior distribution on the joint prior, α_j and β_j. In practice, simply initializing these precisions to arbitrary values and allowing them to adapt freely during EM leads to poor results. Some of the precisions—particularly α_w, α_m, $\tau(\mathbf{v}_j^f)$, and $\tau(\mathbf{e}_i^f)$—tend to grow unbounded, thus we have found it useful to specify a maximum precision of 50 (a standard trick during EM). In contrast, the joint precisions ϕ_j (given by $\alpha(\phi_j)/\beta(\phi_j)$) tend towards relatively small values, resulting in a model that has very little cohesion in the joints. To counteract this we specify a very strong prior on ϕ_j encouraging it towards large values: $\alpha_j = 2 \times 10^5 \times$ *maximum precision* and $\beta_j = 10^5$, resulting in an expected value of $2 \times$ *maximum precision* with limited variance. When fitting the motion of a stick, assuming other precisions saturate at the maximum, this means that keeping an endpoint near its vertex is at least twice as important as keeping a feature point near its observed location.

In our experiments, we have found temporal smoothing of the vertices, governed by precision τ_t to be a disadvantage during learning. Particularly at the beginning, when the structure contains no joints, smoothing causes the unconnected vertices and endpoints to drift away from the actual observations at each frame, towards their temporal mean. Thus, in all of the following experiments we disable smoothing during learning. However, when measuring test performance it's not uncommon for one or more adjacent sticks to be entirely occluded during a frame. When this happens, smoothed locations of the vertices provide the only source of information about the location of the stick, and thus temporal smoothing is essential for limiting test error. When measuring test performance, therefore, we enable smoothing and set $\tau_t = 2000$.

Finally, the precisions play an important role in determining the optimal number of joints. During model selection we seek the model with the largest expected complete log-likelihood, hoping that this will include as many plausible joints as possible. However most terms in this objective function favour a disconnected model, with more vertices and fewer joints. To understand this, consider the problem of estimating the motion of an unconnected stick. Since there are no constraints on the vertices, they can be trivially placed to be coincident with endpoints, thus the motion variable needs only focus on maximizing the probability of the observations. However when two sticks are joined together, perfect placement of the vertices is generally not possible, requiring modelling compromises that introduce slight reductions in observation probability. The one term in the objective function that does not decrease as merges are performed is the entropy of the vertices $\mathrm{E}_{Q(\mathbf{V})}[\log Q(\mathbf{V})]$.

Assuming each precision parameter in $Q(v)$ is equal to the maximum precision, p, this entropy is $(FJD_o/2)\log(2\pi e/p)$. If p is greater than $2\pi e \approx 17.08$, then the differential entropy is negative[5]. The result is that decreasing the number of vertices J causes the log-likelihood to increase. In fact a fixed cost of $(FD_o/2)\log(2\pi e/p)$ is paid for each vertex in the model, giving us the desired bias towards connectivity. Plots of the relevant log-likelihood terms are included for the datasets presented below.

5.2 Excavator

Our first dataset consisted of a video clip of an excavator. We used a Kanade-Lucas-Tomasi tracker (Shi and Tomasi, 1994) (with manual assistance to correct for frequent loss-of-track) to obtain 35 feature trajectories across 176 frames. Our algorithm processed the data in 4 minutes on a 2.8 GHz processor. The learned model at each stage of greedy merging is depicted in Figure 6. The optimal structure was chosen by comparing the log-likelihood at each stage, as plotted in Figure 7 (left). The four most significant terms comprising this objective function are plotted individually in Figure 7 (right). As can be seen, joining sticks adds additional constraints that reduce the expected probability of the observations (top left), the endpoints given vertices (top right), and the endpoints given **Mk** (bottom left). In contrast the vertex entropy term (bottom right) acts as a per-vertex penalty, which decreases as we merge vertices, favoring more highly connected models. Figure 7 (bottom) shows that the system's prediction error for occluded data was significantly better than either multibody or single-body SFM.

As can be seen in Figure 6, the model does a good job at recovering the structure—the grouping and connectivity—of the observed trajectories. The reconstruction shows some deviation between the inferred locations of the joints and their intuitive positions. The probable source of this inaccuracy is that the small range of motion exhibited by the excavator's arm permits a range of possible joint positions, while the Gaussian prior says that the joints should be near the center of mass of each stick. Apparently, while mathematically convenient, the Gaussian prior is not always the best choice. Nevertheless, the model is fully able to capture the observed motion of the excavator's arm, despite the inaccurate joints.

Using the excavator data, we also examined the model's robustness to learning with occlusions in the training data. When the occlusion scheme described earlier was employed to generate a training set with missing observations, and the learning algorithm was applied to this data, it was still able to recover the correct structure. Similarly, when training observations were randomly withheld during training, rather than using structured occlusion, the correct structure was reliably recovered with up to 75% of the training observations missing.

5.3 Giraffe

Our second dataset consisted of a video of a walking giraffe. As before features were tracked, producing 60 trajectories across 128 frames. Merging results are depicted in Figure 8. Using the objective function to guide model selection (Figure 9), the best structure corresponded to stage 10, and this model is shown superimposed over the original video in Figure 1, appearing at the start of this article.

5.4 2D Human

Our third dataset consisted of optical human motion capture data (courtesy of the Biomotion Lab, Queen's University, Canada), which we projected from 3D to 2D using an isometric projection. The data contained 53 features, tracked across a 1018-frame range-of-motion exercise (training data), and 318 frames of running on an inclined plane (test data). The structures learned during greedy merging are shown in Figure 10, of which stage 11 most closely matches human intuition.

By examining the plots in Figure 11, it can be noted that the expected log-likelihood of the various models forms a plateau, roughly between stages 8 and 11,

[5] Although unintuitive, negative differential entropies are perfectly acceptable (Cover and Thomas, 1991).

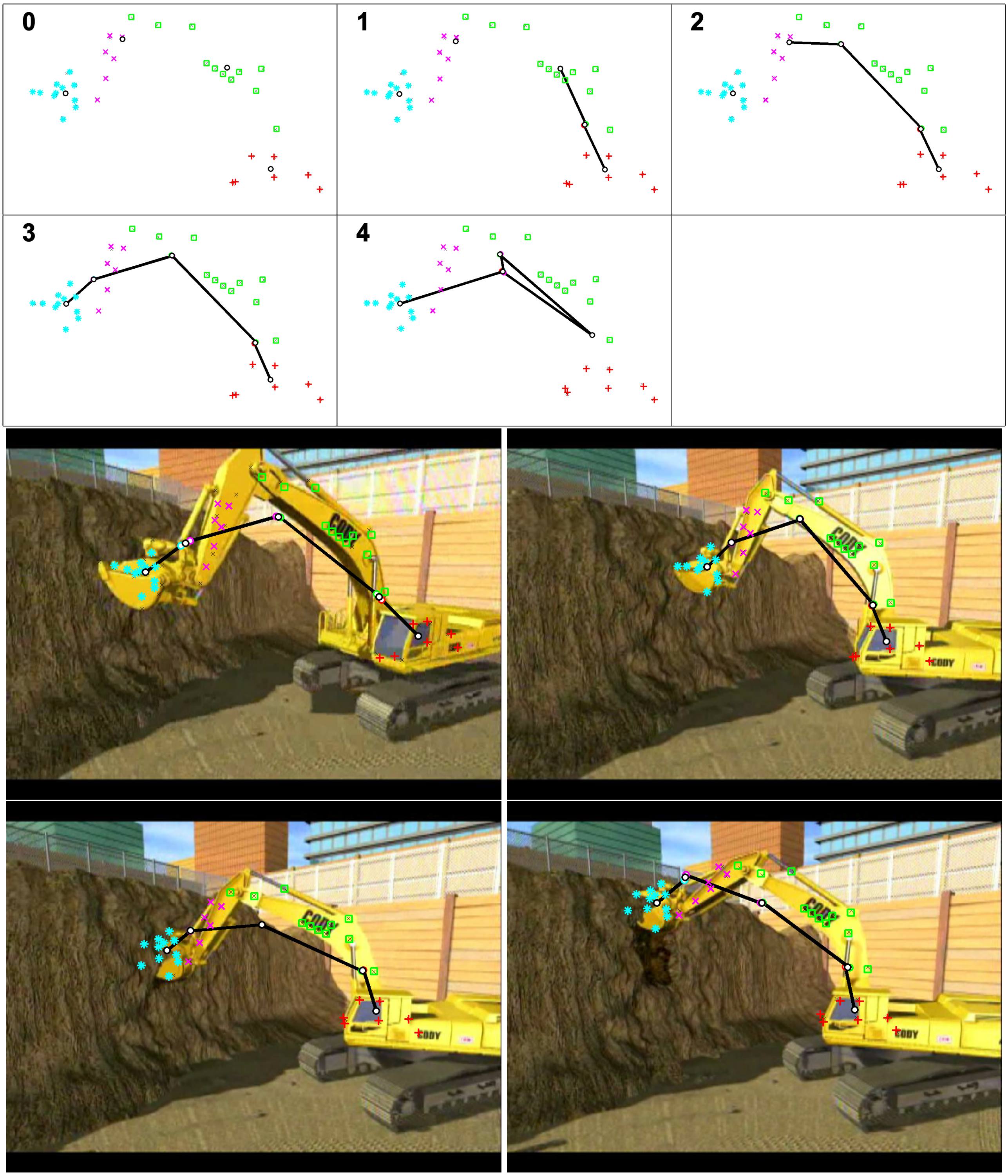

Fig. 6 Excavator Data: Shown at the top are the models learned in each of the five successive stages of greedy learning. Reconstructions of the observed markers are shown with different symbols depending on their stick assignments. The locations of vertices are shown as black o's, and black lines are drawn to connect each stick's pair of vertices. At the bottom, the selected model (stage 3) is used to reconstruct the observed feature trajectories, and the results are superimposed over the corresponding frames of the input video.

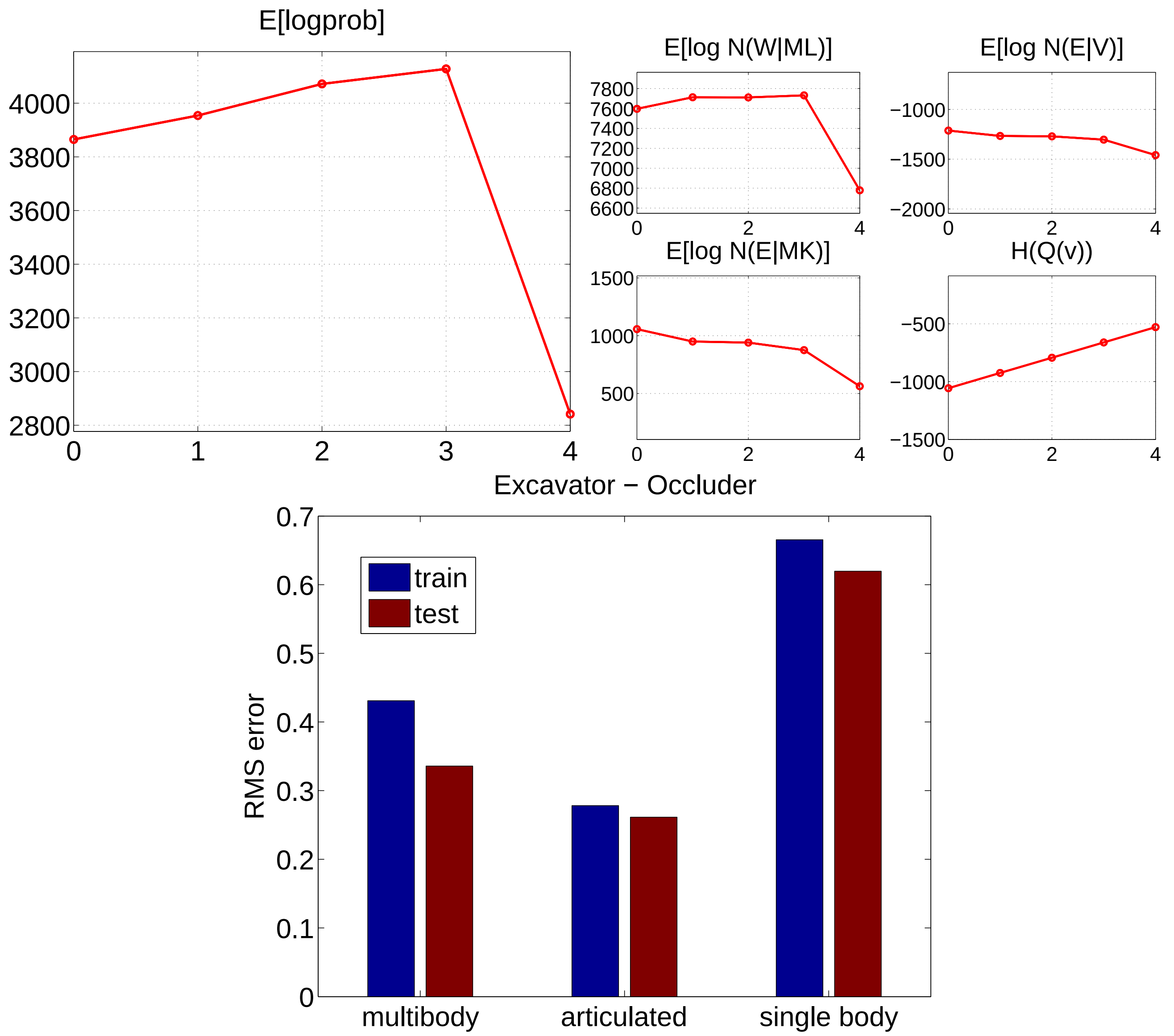

Fig. 7 Excavator Log-likelihood and Error. At the top-left we see that stage 3 of merging produces the model with the highest log-probability. At the top-right are individual plots of the four most significant terms comprising the log probability. At the bottom, we can see that the learned model exhibits less reconstruction error than either single or multibody SFM models.

rather than a sharp peak as seen for the Excavator data. Although stage 11 is not actually the most likely model (stage 8 is slightly higher), the log-likelihood decreases rapidly after stage 11. This suggests that having too many joints—and thereby hampering the ability of sticks to move so as to fit the observations—is a bigger disadvantage to the model than simply having too few joints. Theoretically it may be possible to encourage a global maximum in log-likelihood at stage 11 by simply increasing the maximum precision (thereby penalizing stage 8 which has more vertices). However, recognizing our preference for models with as many plausible joints as possible, selecting the stage at the edge of the plateau—stage 11—seems a reasonable choice.

Again, the articulated model achieved a lower test error than either SFM or multibody SFM.

5.5 Synthetic Ring

In order to evaluate the performance of the model on data which contains significant out-of-plane motion, we created a challenging synthetic dataset depicting a segmented ring deforming in space. The generated sequence consisted of 100 features across 300 frames, to which independent Gaussian noise of standard deviation 0.05

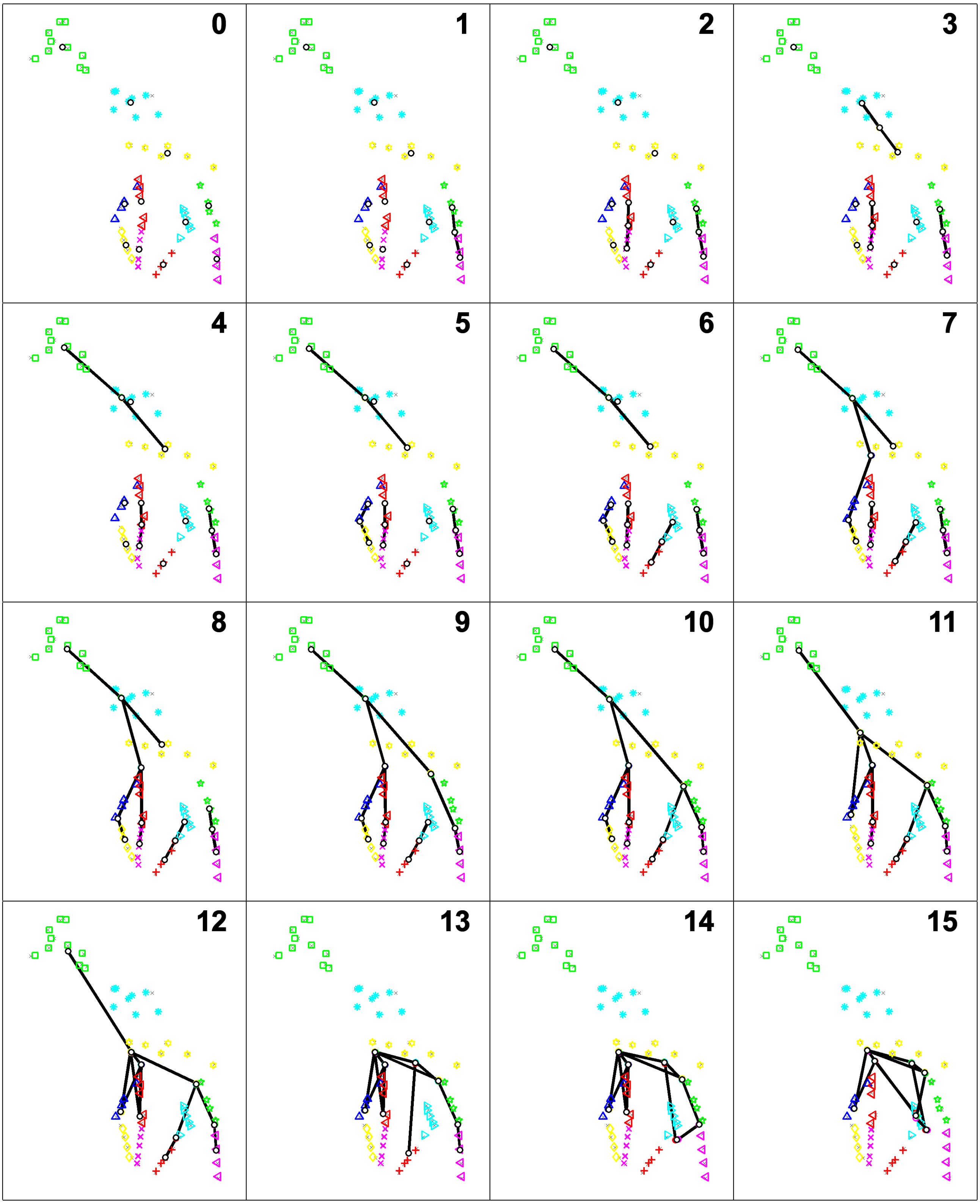

Fig. 8 Giraffe structures learned during greedy merging. Stage 10 has the highest expected log-likelihood.

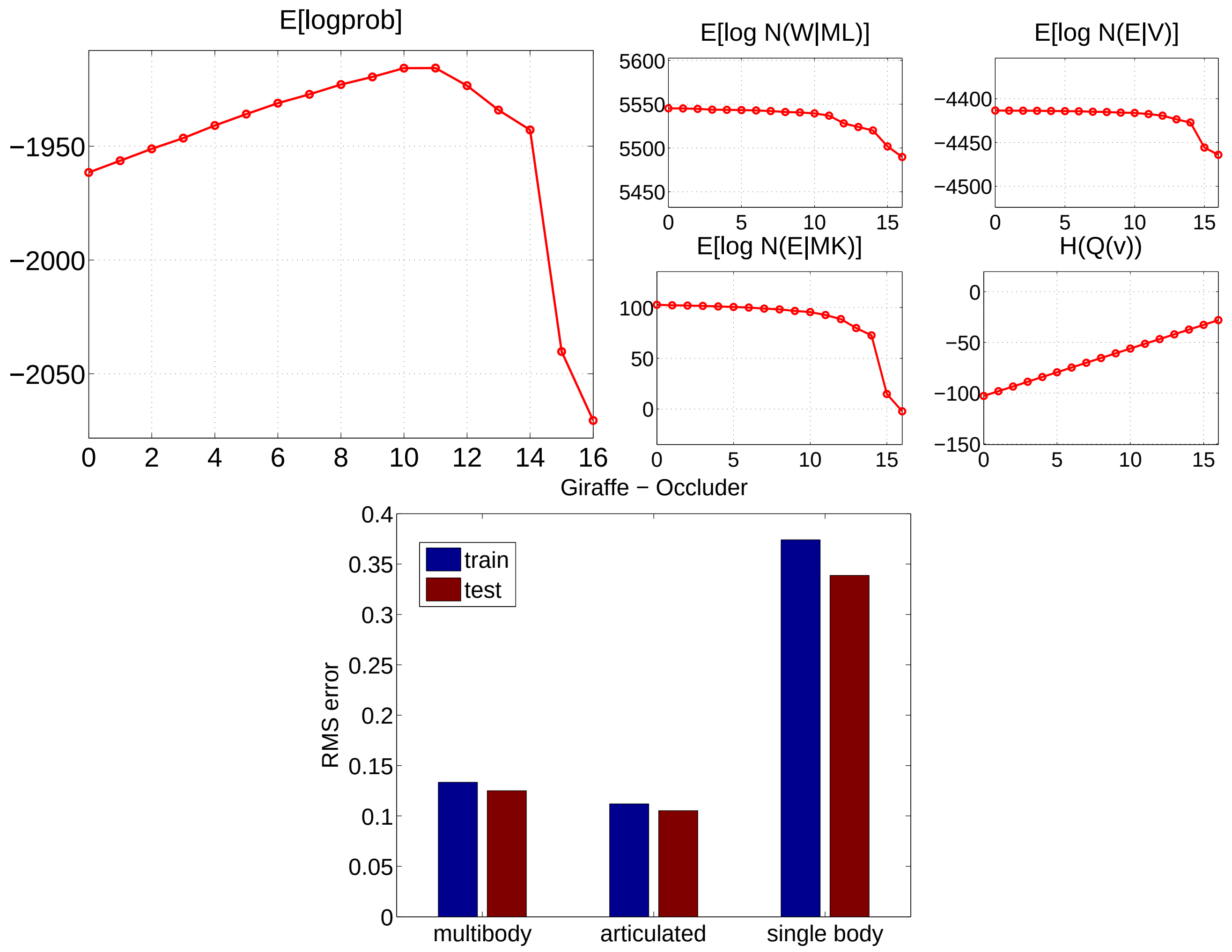

Fig. 9 Giraffe Log-likelihood and Error.

was added. (For comparison, each stick was approximately 0.5 units wide and 5 units long.) Six frames from the sequence are depicted in Figure 12.

The models learned for the successive stages of merging are shown in Figure 13. The sharp downturn in log-likelihood between stages 8 and 9, shown in Figure 14, suggests selecting stage 8 as the best model. (Note that although stage 0, which is equivalent to multibody SFM, has a higher expected log-probability, stage 8 has the lower test error.) Unlike methods based on spanning trees, our approach was able to recover the correct closed ring structure.

Interestingly, all of the learned structures chose to group the feature points into eight sticks, three more than were in the true grouping used to generate the data, as illustrated in the bottom-right of Figure 13. Examination of the results show that these extra groups arise from splitting three of the true sticks each into a pair sticks connected by a joint. Although the learned structure is an over-segmentation of the ground truth structure, it still provides a perfectly acceptable model of the data.

As a further analysis of the algorithm's inability to identify the correct number of sticks and joints, an experiment was performed in which the correct ground-truth segmentation for the ring data was provided as an initialization. From this starting point, the learning algorithm was able to recover the connectivity, joint locations, and parameters correctly. This suggests that the problem is not inherent in the representative capability of the model, rather that the greedy/EM optimization algorithm has difficulty escaping from a poor initial segmentation, thereby impairing the ability to identify the correct number of sticks.

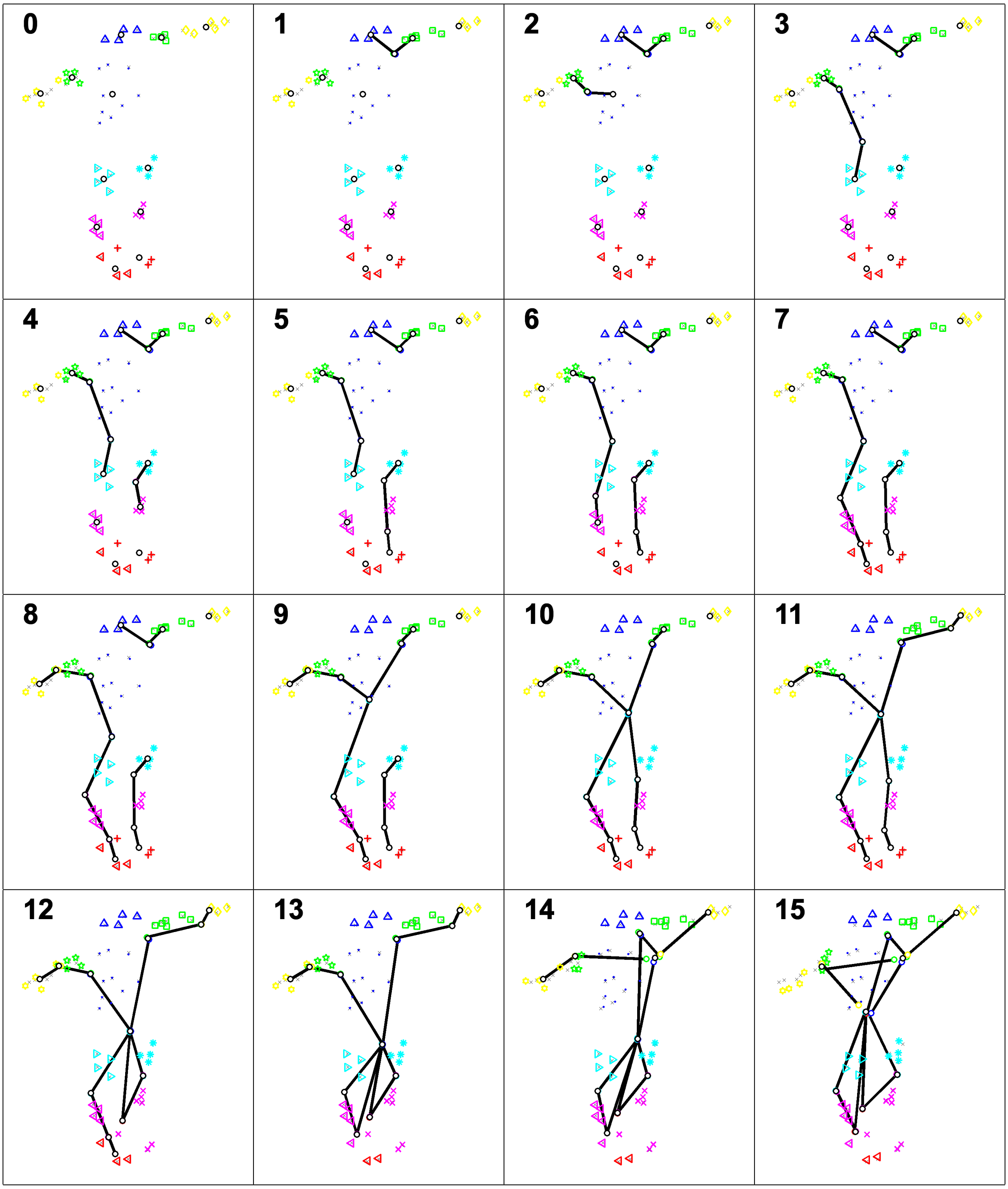

Fig. 10 2D Human structures learned during greedy merging, of which stage 11 most closely matches human intuition.

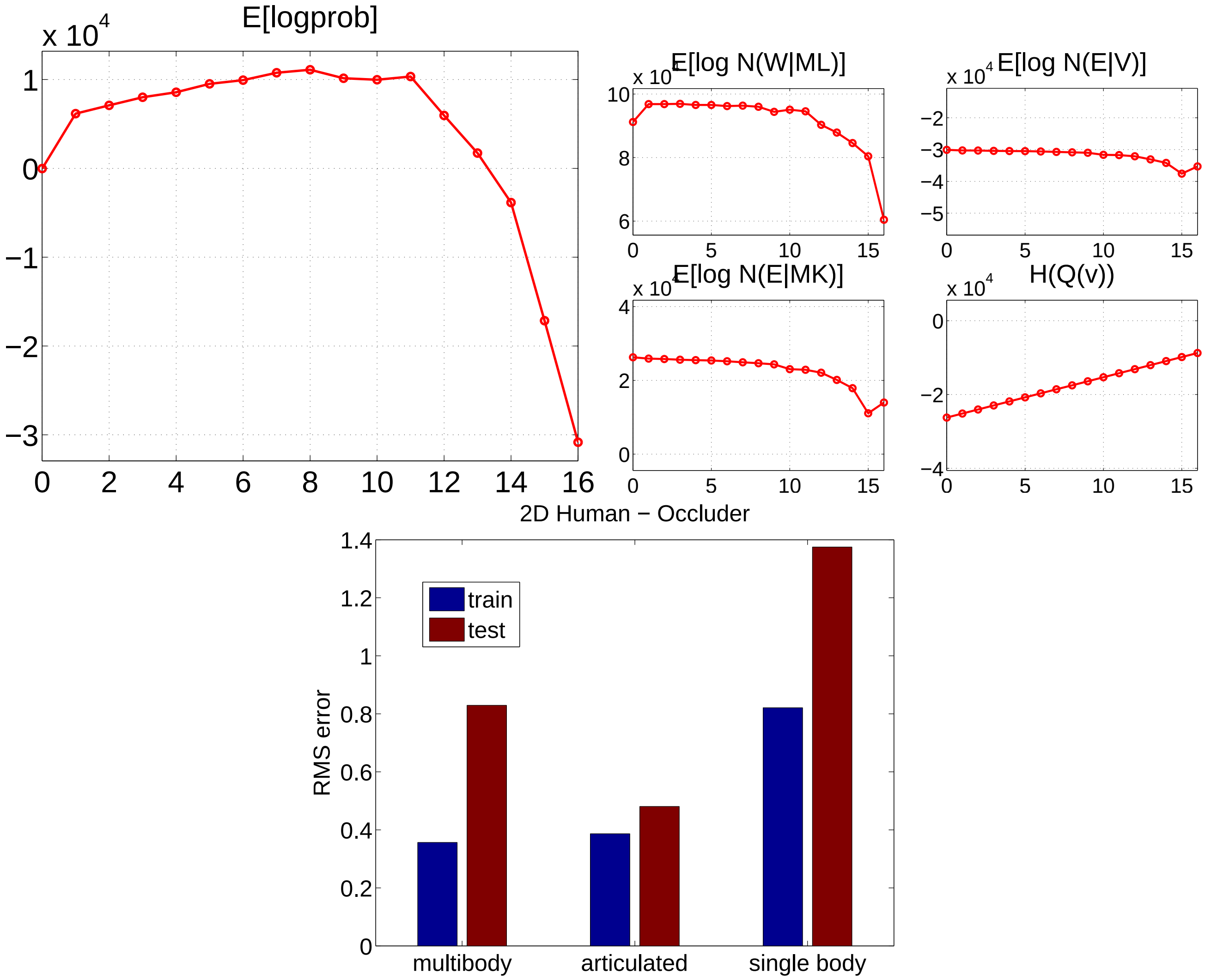

Fig. 11 2D Human Log-likelihood and Error.

5.6 3D Human

Although recovering 3D structure from 3D observations is much simpler than from 2D data, it also receives attention in the literature. As mentioned previously, our model easily extends to 3D observations, so we include an additional experiment demonstrating this ability. Here we trained our model on optical human motion capture data obtained from the Carnegie Mellon University Motion Capture Database. The data consisted of 174 feature points tracked across 732 frames (downsampled by a factor of three from the original framerate). The results of greedy merging are shown in Figure 15, and the corresponding log-likelihoods in Figure 16. Since learning from 3D observations is an easier problem, the most likely structure—stage 15—is visually more appealing than the structure learned earlier on the 2D human data.

5.7 Comparisons with Related Methods

Finally, as an additional qualitative comparison, we ran our method on two sequences from Yan and Pollefeys (2008), and ran a re-implementation of Kirk et al. (2005) on our 3D datasets.

The results of our method on Yan and Pollefeys's "puppet" and "dancing" sequences are shown in Figure 17, at the top left and top right respectively. (Please compare with Figures 10 and 11 in Yan and Pollefeys (2008).) As can be seen, our method does a good job recovering the structure, including segmentation and joints, of the puppet. In contrast with Yan and Pollefeys's, our approach finds more segments: the arms and legs are split into two segments each, instead of only one; and the head, neck, and chest are subdivided, instead of being combined into one segment. An unintuitive choice made by our algorithm was to place a joint

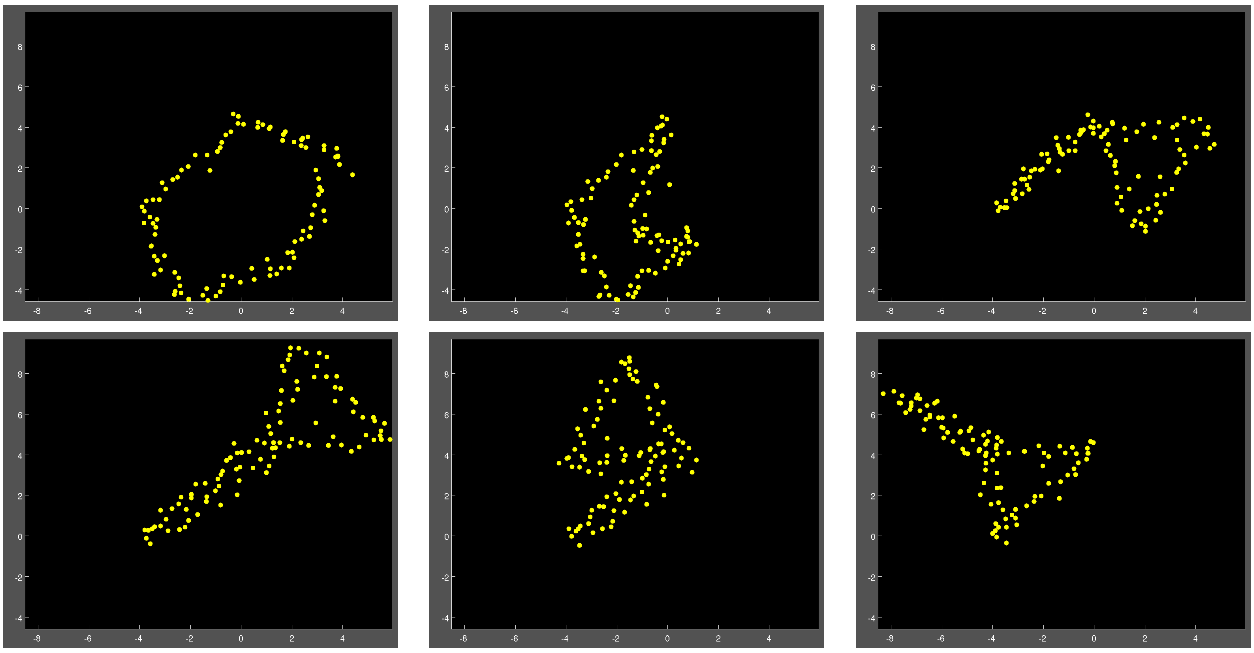

Fig. 12 Synthetic Ring Data: Six frames selected from a synthetic data sequence depicting the motion of a 5-segmented ring. The ring undergoes significant out-of-plane motion.

connecting the legs, above the knees. This placement makes more sense upon watching the entire sequence, and noticing how little the legs appear to move. Specifically, the left leg is stationary, and the right leg moves only slightly, back and forward perpendicularly to the image plane. In this case the algorithm favours a simpler model which still adequately captures the visible motion.

The results on the "dancing" sequence are similar. Again our method find more segments. In particular the chest is divided into four segments instead of one. Interestingly, the algorithm learned interconnections between these chest segments, producing a near-fully connected graph. This shows that the model does a good job capturing the near-rigidity of the chest segments. However it also suggests there is a limitation in the extent to which initial segments which are actually tightly coupled can be combined into a single segment during learning. The segments which show the most motion, the forearms and head, each are reasonably modeled by sticks which extend from the main body.

As described earlier, the method of Kirk et al. is designed to work on 3D optical motion capture data, thus we trained it on the 3D Human dataset used in Section 5.6, as well as on the 3D feature locations that gave rise to the 2D Human dataset from Section 5.4. In the original paper, Kirk et al. focus on fitting their model to "calibration" sequences, in which the actor fully flexes each of his individual joints. Indeed, as shown in Figure 17, (bottom right), the method does a good job at recovering the structure from the range-of-motion sequence. (For comparison, the results of our method trained on the 2D-projection of the same sequence is shown in Figure 10.) In contrast, on the other 3D Human sequence which depicts walking and sitting rather than range-of-motion exercises, Kirk's method fares much more poorly (Figure 17 (bottom left), *c.f.* our method Figure 15).

6 Discussion

We have demonstrated a single coherent model that can learn the structures and motion of articulated skeletons. This model can be applied to a variety of structures, requiring no input beyond the observed feature trajectories, and a minimum of manually adjusted precision parameters.

Our model makes a number of contributions to the state of the art. First, it is based optimizing a single global objective function, which details how all aspects of learning—grouping, connectivity, and parameter fitting—contribute to the overall quality of the model. Having this objective function permits iteration between updates of the structure and parameters, allowing information obtained from one stage to assist learning in the other. Moreover, the value of the objective function proves useful for model selection, determining the optimal number of joints. Also, the noise

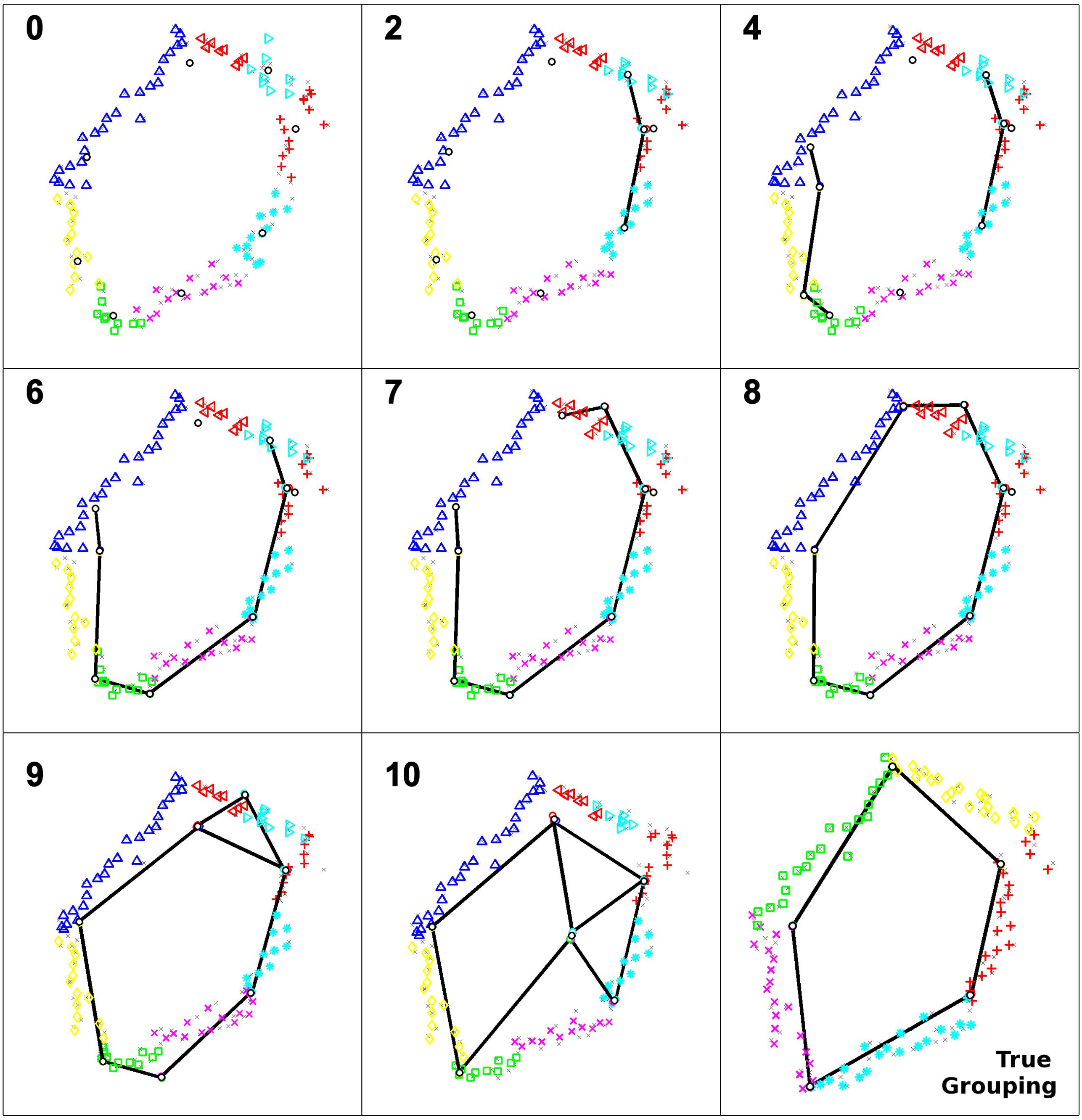

Fig. 13 Synthetic Ring Structures learned during greedy merging, of which stage 8 is the best. In comparison to the ground-truth structure, shown in the lower-right, the learned model over-segments the data into 8 sticks, rather than 5. However, since this involves splitting three of the true sticks in half, the learned model still provides a good fit to the data.

in our generative model plays an important role, allowing a degree of non-rigidity in the motion with respect to the learned skeleton. This not only allows a feature point to move in relation to its associated stick, but also permits complexity in the joints, as the stick endpoints joined at a vertex need not coincide exactly. In addition we presented a method for quantitative comparison, based on imputing the locations of occluded observations, and were able to demonstrate that our model performs measurably better than single-body or multibody structure from motion.

Our model has some limitations. First, as illustrated in the “excavator” example (Section 5.2) the choice of a Gaussian prior for joint locations, while mathematically convenient, is not ideal, since it encourages the model to place joints in the middle of each stick, rather

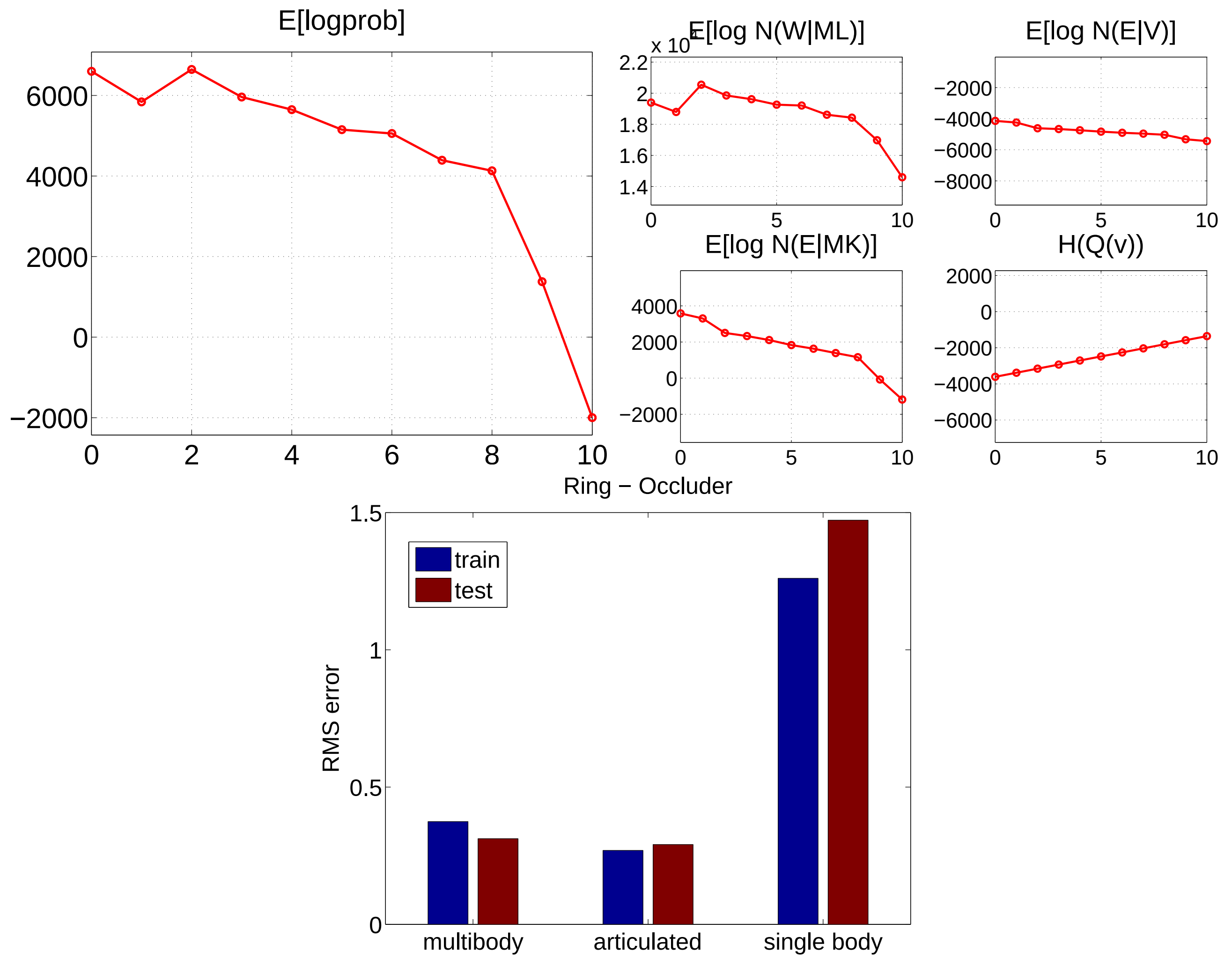

Fig. 14 Synthetic Ring Log-likelihood and Error. The sharp downturn in log-likelihood at stage 9 suggests selecting the structure learned during stage 8.

than at its endpoints. In many cases the effect of the prior is minor, and the problem does not arise. However, when the range of observed motion in a particular joint is quite limited, the prior can be more pronounced, moving the inferred joint away from its true location. Secondly, when starting from a poor initial segmentation, the greedy/EM learning algorithm can have difficulty identifying the correct number of sticks, due to challenges escaping from local minima. This problem occurs in the synthetic ring experiment (Section 5.5), and suggests that alternative optimization procedures be investigated.

To obtain good results, the model requires a certain density of features, in particular because the affinity matrix used for initialization (Yan and Pollefeys, 2006a, 2008) requires at least 4 points per stick. In addition, the flexibility of learned models is limited to the degrees of freedom visible in the training data; if a joint is not exercised, then the body parts it connects cannot be distinguished. Finally, our model requires that the observations arise from a scene containing roughly articulated figures; it would be a poor model of an octopus, for example.

An important direction for future study is the ability of learned skeletal structures to generalize: applying them to new motions not seen during training, and to related sequences, such as using a model trained on one giraffe to parse the motion of another.

Acknowledgements The motion capture data used in this project was provided by the Biomotion Lab, Queen's University, Canada, and the Carnegie Mellon University Motion Capture Database http://mocap.cs.cmu.edu/ (created with funding from NSF EIA-0196217). We would like to acknowledge funding from the Natural Science and Engineering Research Council of Canada, the Canadian Institute for Advanced Research, and a Microsoft/LiveLabs-University Support Agreement.

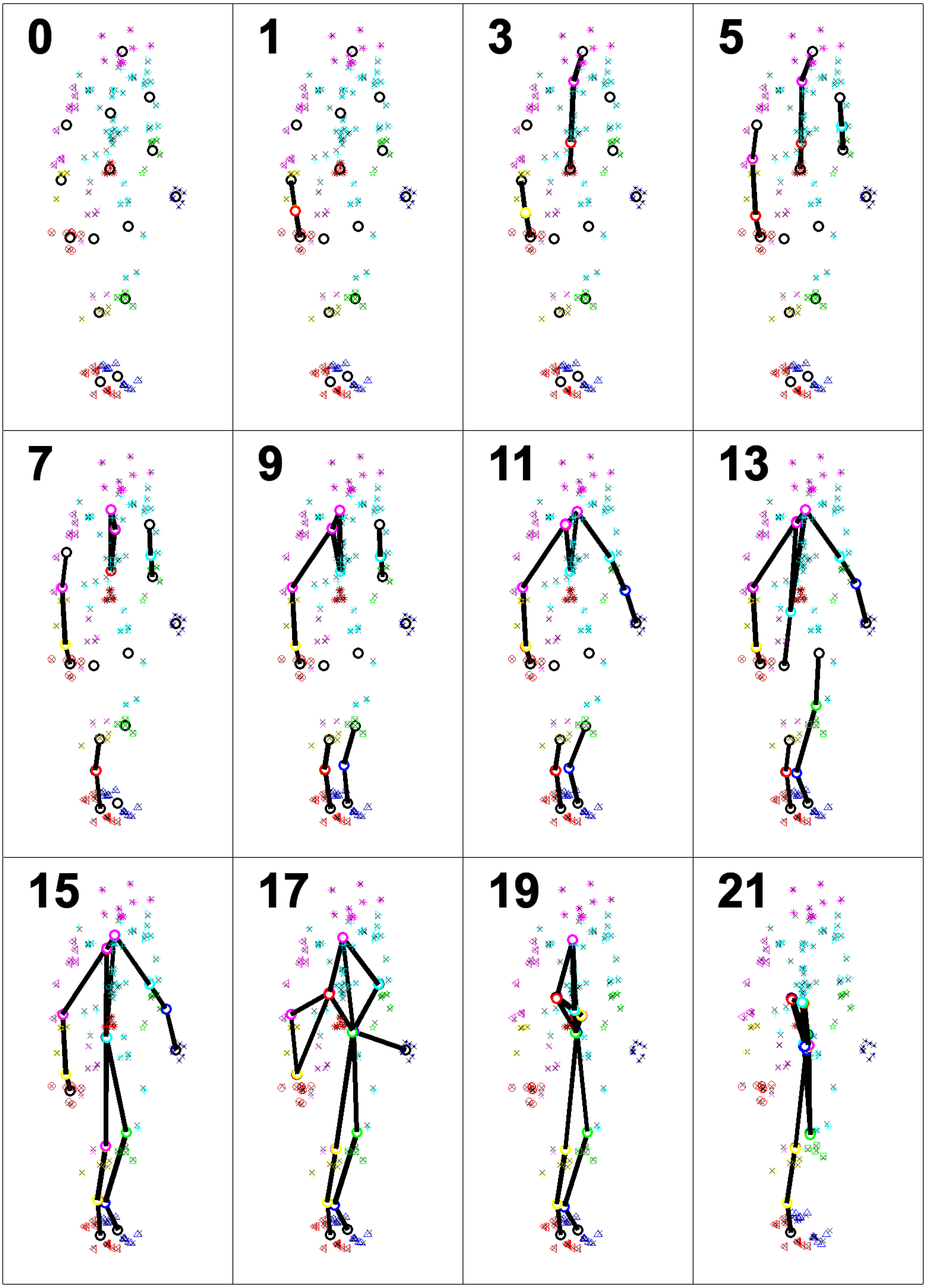

Fig. 15 3D Human structures learned during greedy merging. Stage 15 has the highest log-likelihood.

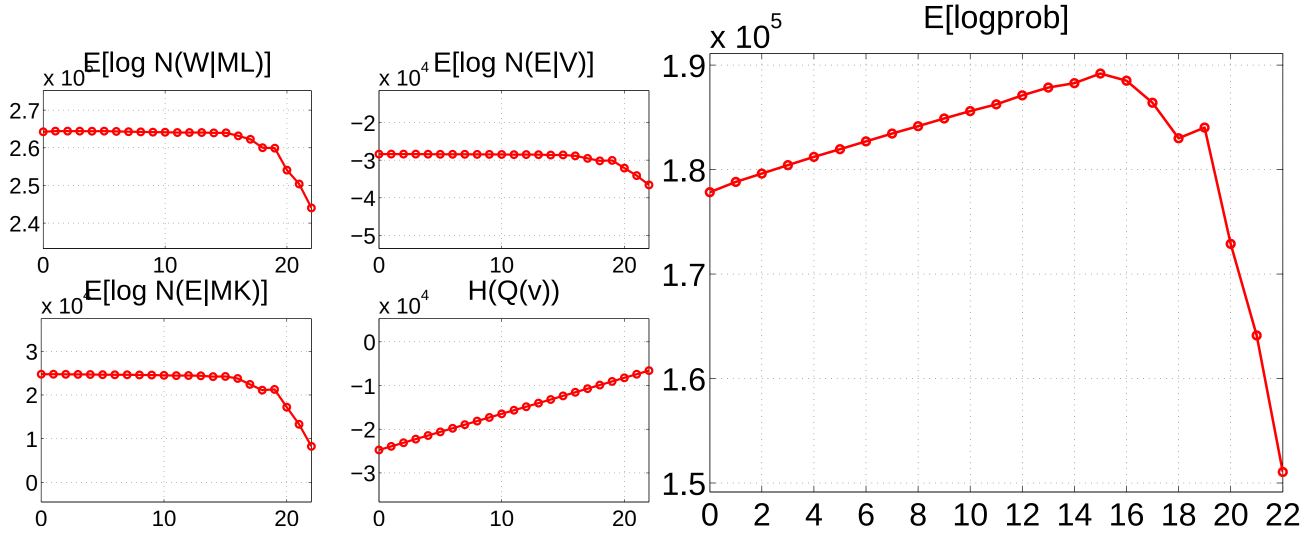

Fig. 16 3D Human Log-likelihood.

References

K. Abdel-Malek, J. Arora, S. Beck, M. Bhatti, J. Carroll, T. Cook, S. Dasgupta, N. Grosland, R. Han, H. Kim, J. Lu, C. Swan, A. Williams, and J. Yang. Digital human modeling and virtual reality for FCS. Technical Report VSR-04.02, The Virtual Soldier Research (VSR) Program, Center for Computer-Aided Design, College of Engineering, The University of Iowa, October 2004.

M. Bray, P. Kohli, and P. Torr. Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In *ECCV (2)*, pages 642–655, 2006.

J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *Image Understanding Workshop*, pages 1013–1026, 1996.

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

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

P.F. Culverhouse and H. Wang. Robust motion segmentation by spectral clustering. In *British Machine Vision Conference*, pages 639–648, 2003.

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39: 1–38, 1977.

B.J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, February 2007.

C. W. Gear. Multibody grouping from motion images. *Int. J. Comput. Vision*, 29(2):133–150, 1998. ISSN 0920-5691. doi: http://dx.doi.org/10.1023/A:1008026310903.

Z. Ghahramani and G.E. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, University of Toronto, 1996a.

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

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.

Amit Gruber and Yair Weiss. Factorization with uncertainty and missing data: Exploiting temporal coherence. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems*. MIT Press, 2003. ISBN 0-262-20152-6.

Amit Gruber and Yair Weiss. Multibody factorization with uncertainty and missing data using the EM algorithm. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 707–714, 2004.

R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2003.

L. Herda, P. Fua, R. Plankers, R. Boulic, and D. Thalmann. Using skeleton-based tracking to increase the reliability of optical motion capture. *Human Movement Science Journal*, 20(3):313–341, 2001.

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

Adam G. Kirk, James F. O'Brien, and David A. Forsyth. Skeletal parameter estimation from optical

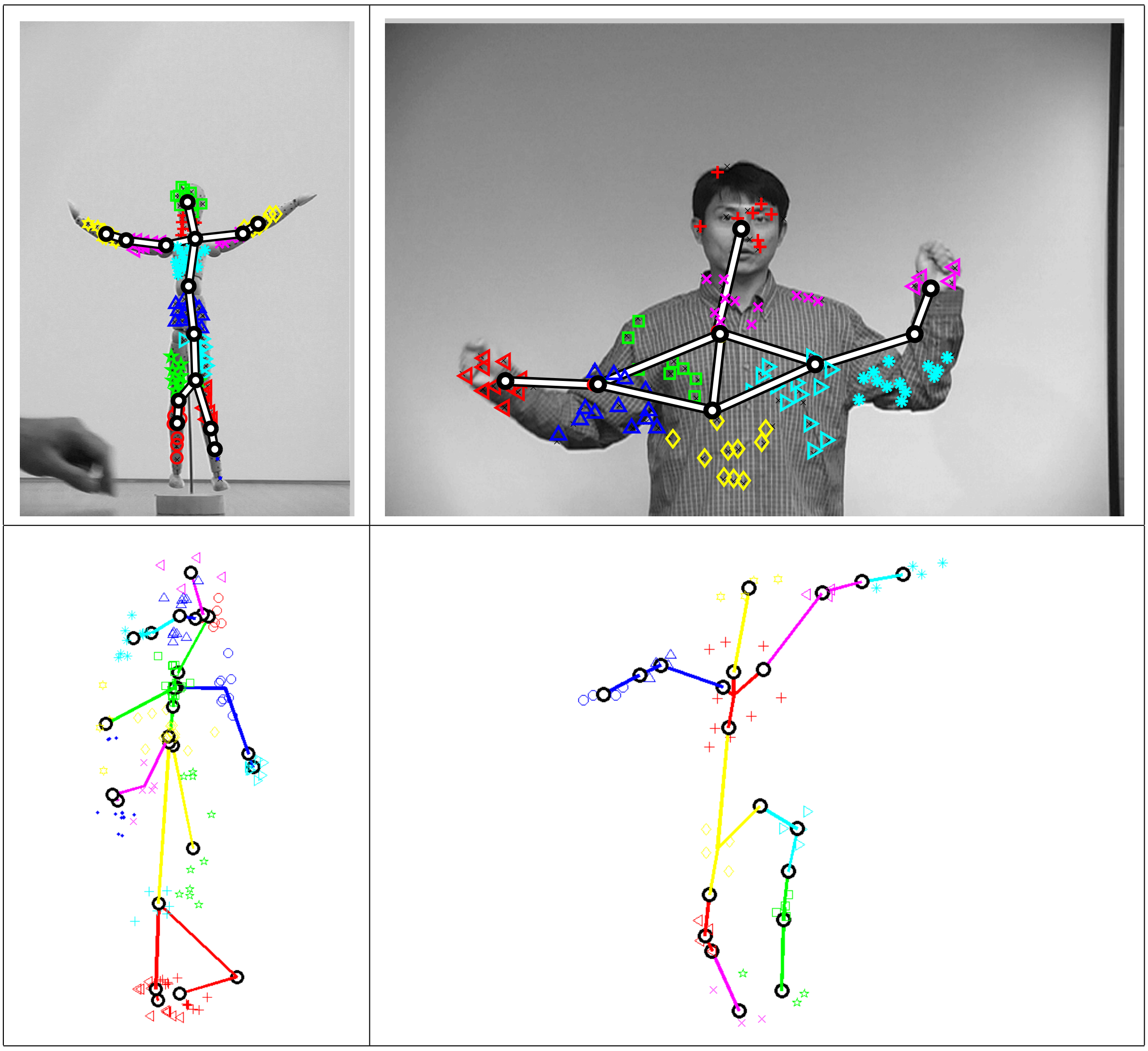

Fig. 17 A comparison of results by related methods. Our method was trained on the "puppet" and "dancing" sequences from Yan and Pollefeys (2008) (top row). The Kirk et al. (2005) method was trained on 3D feature locations from the two datasets of human motion capture (bottom row) .

motion capture data. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2005. ISBN 0-7695-2372-2.

R. Neal and G. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*. Kluwer, 1998.

A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.

David A. Ross. *Learning Probabilistic Models for Visual Motion*. PhD thesis, University of Toronto, Ontario, Canada, 2008a.

David A. Ross. *Learning Probabilistic Models for Visual Motion*. PhD thesis, University of Toronto, Toronto, Ontario, Canada, 2008b.

David A. Ross and Richard S. Zemel. Learning parts-based representations of data. *Journal of Machine Learning Research*, 7:2369–2397, Nov 2006.

David A. Ross, Daniel Tarlow, and Richard S. Zemel. Learning articulated skeletons from motion. In *Workshop on Dynamical Vision at ICCV*, 2007.

David A. Ross, Daniel Tarlow, and Richard S. Zemel. Unsupervised learning of skeletons from motion. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Proceedings of the 10th European Conference on Computer Vision (ECCV 2008)*. Springer, 2008.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.

Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, 1994.

M. C. Silaghi, R.Plankers, R.Boulic, P.Fua, and D.Thalmann. Local and global skeleton fitting techniques for optical motion capture , modeling and motion capture techniques for virtual environments. In *Lecture Notes in Artificial Intelligence*, volume 1537, pages 26–40. Springer, 1998.

C. Sminchisescu and B. Triggs. Estimating Articulated Human Motion with Covariance Scaled Sampling. *International Journal of Robotics Research*, 22(6):371–393, 2003.

Y. Song, L. Goncalves, and P. Perona. Unsupervised learning of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7): 814–827, July 2003.

Yang Song, Luis Goncalves, and Pietro Perona. Learning probabilistic structure for human motion detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 771–777. IEEE Computer Society, 2001. ISBN 0-7695-1272-0.

Leonid Taycher, John W. Fisher III, and Trevor Darrell. Recovering articulated model topology from observed rigid motion. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 1311–1318. MIT Press, 2002.

C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.

P. Tresadern and I. Reid. Articulated structure from motion by factorization. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 1110–1115, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: http://dx.doi.org/10.1109/CVPR.2005.75.

Thomas Viklands. *Algorithms for the Weighted Orthogonal Procrustes Problem and Other Least Squares Problems.* PhD thesis, Ume University, Ume, Sweden, 2006.

Y. Weiss. Segmentation using eigenvectors: a unifying view. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 1999.

Jingyu Yan and Marc Pollefeys. Factorization-based approach to articulated motion recovery. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005a.

Jingyu Yan and Marc Pollefeys. Articulated motion segmentation using ransac with priors. In *(ICCV) Workshop on Dynamical Vision (ICCV)*, 2005b.

Jingyu Yan and Marc Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part III*, 2006a.

Jingyu Yan and Marc Pollefeys. Automatic kinematic chain building from feature trajectories of articulated objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006b.

Jingyu Yan and Marc Pollefeys. A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):865–877, 2008. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2007.70739.