Learning Articulated Skeletons from Motion

David A. Ross, Daniel Tarlow, and Richard S. Zemel

Department of Computer Science, University of Toronto, Canada
{dross,dtarlow,zemel}@cs.toronto.edu

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 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. We formulate the problem in 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 2D and 3D datasets including optical human motion capture and video of walking giraffes.

1 Introduction

An important aspect of analyzing dynamical 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 paper, we investigate this 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. Our same approach applies when the observations are the three-dimensional world coordinates of the features or their two-dimensional image coordinates. An example of a stick figure learned by applying our model to two-dimensional feature observations from a video of giraffe motion is shown in Figure 1.

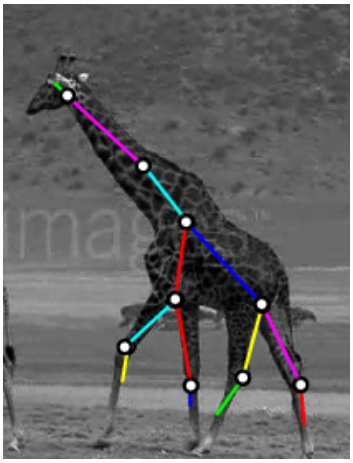

Fig. 1. A frame from a video of a walking giraffe, augmented with a learned skeleton. Each colored line represents a separate stick, and the white circles are joints.

Learned models of skeletal structure have many possible uses. For example, 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, permitting disambiguation of marker correspondence and occlusion [1]. Additionally, 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 that those constructed manually. Finally, a learned skeleton might be used as a rough prior on shape to help guide image segmentation [2].

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 describe the optimization of this model, which proceeds in a stage-wise fashion, building up the structure incrementally to maximize the joint probability of the model variables. We generate several hypothesized structures by sampling from this probabilistic model, and then use validation data in which multiple features are occluded in order to determine an optimal structure. In Section 4 we test the algorithm on a range of datasets: data of human motion from optical motion capture devices; motion capture data of multiple subjects; 2D human data; and features extracted from video of walking giraffes. We show that the algorithm can also work when there are multiple objects in the scene, and when articulated parts are non-rigid. In the final section we describe assumptions and limitations of the approach, and discuss future work.

2 Related Work

The task of learning stick figures from a set of feature point trajectories can be thought of as a variant of the *structure from motion* (SFM) problem.[1] When the trajectories all arise from the motion of one rigid object, Tomasi and Kanade [3] have shown that the matrix of point locations, $\mathbf{W}$, is a linear product of a time-invariant structure matrix, $\mathbf{S}$, and a time-varying matrix of motion parameters, $\mathbf{M}$. $\mathbf{M}$ and $\mathbf{S}$ can be recovered by singular value decomposition.[2] SFM can also be extended to handle multiple rigid objects moving independently. Costeira and Kanade [5] have shown that this problem, known as multibody SFM, can be solved by grouping the point trajectories according to the object they arise from, then solving SFM independently for each object. Grouping is accomplished by forming a shape-shape interaction or *affinity* matrix, indicating the potential for each pair of points of belonging to the same object, and using this matrix to cluster the trajectories.

Several authors have demonstrated that SFM can be interpreted as a probabilistic generative model, *e.g.* [6–8]. This view permits the inclusion of priors on the motion sequence, thereby leveraging temporal coherence. Furthermore, in the multibody case, Gruber and Weiss have presented a single probabilistic model that describes both the grouping problem and the per-object SFM problems [8]. This produces a single objective function that can be jointly optimized, leading to more robust solutions.

Unfortunately, multibody SFM cannot reliably be used to obtain the structure and motion of an articulated figure's parts since, as shown by Yan and Pollefeys [9], the motions of connected parts are linearly dependent. However, this dependence can be used to form an alternative affinity matrix for clustering the trajectories. Yan and Pollefeys use this as the basis for a stage-wise procedure for recovering articulated SFM [10]: (1) cluster point trajectories into body parts; (2) independently run SFM on each part; (3) determine connectivity between parts by running (a variant of) minimum spanning tree, where edge weights are the minimum principle angle between two parts' motion matrices (for connected, dependent parts, this should be zero); (4) finally, solve for the joint locations between connected parts. A disadvantage of this method is its lack of an overall objective function that can be optimized globally, and used to compare the quality of alternative models.

Given three-dimensional observations, if two points are attached to the same rigid body part, the distance between them is constant. Kirk *et al.* [11] use this simple fact as the basis for a stage-wise algorithm to automatically recover articulated structure from motion capture data. First, trajectories are clustered using, as an affinity measure, the (negative) standard deviation of the distance

[1] 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.

[2] This solution assumes an affine camera. Solutions based on the projective camera, perhaps using the above method as an initialization, can be obtained via *bundle adjustment* [4].

between each pair of points. Next, for each pair of body parts they compute a *joint cost*, indicating the likelihood that the parts are connected by a rotational joint. Noting that a joint can be interpreted as an unobserved point belonging to both of the parts it connects, joint cost is the variance in the distance from the putative joint location to each of the points in the two parts, and is computed using nonlinear optimization. Finally, the articulated structure is obtained by running a minimum spanning tree algorithm, using the joint costs as edge weights. Although simple to apply in 3D, this method will not work given observations projected to 2D. Another drawback to this method is that, beyond computing the positions of joints in each frame, it does not produce a time-invariant model of structure or a set of motion parameters. This means that filling in missing observations or computing joint angles would require further processing, and that the learned model cannot be applied to novel motions of the same object (test data).

Learning articulated figures can also be interpreted as structure learning in probabilistic graphical models, with nodes representing the positions of parts and edges their connectivity. Learning structure is a hard problem that is usually solved approximately, using greedy methods or by restricting the class of possible structures. Song *et al.* [12] note that the optimal structure (in terms of maximum likelihood) of a graphical model is the one that minimizes the entropy of each node given its parents. Restricting their attention to graphs in which nodes each have two parents, they propose to learn the structure greedily, iteratively connecting to the graph the node with the smallest conditional entropy given its parents. When the space of graphical models is restricted to trees, the exact maximum likelihood (minimum entropy) structure can be found efficiently by solving for the minimum spanning tree [13, 12], using conditional entropy as edge weight. In their application, Ramanan *et al.* [14] find that the minimum-entropy spanning tree often produces poor solutions, joining together parts which are not near each other. Instead, they obtain good results by replacing entropy with the average distance between parts. Krahnstoever *et al.* [15] use combination of "joint cost" and spatial proximity for edge weight. With the exception of [13] (which is concerned only with the final stage of processing, after the motions of individual parts have been obtained), all of these methods build two-dimensional models directly in image coordinates. Thus, unlike SFM approaches, they are unable to deal with out-of-plane motion; a model trained on side views of a person walking would be inapplicable to a sequence of frontal views.

3 Model Formulation

Here we formulate a probabilistic graphical model for sequences generated from articulated skeletons. By fitting it 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 can be 3D (from motion capture) or 2D (from video), but in either case the goal is always to learn skele-

tons 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 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 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 (see Figure 2, Left).

If all of the sticks are unconnected and move independently, then this model essentially describes multibody SFM [8]. However, when sticks are connected at joints, their motion variables are no longer independent [9]. We will use latent variable $\mathbf{G}$ to denote this connectivity of sticks. Specifically, $\mathbf{g}_{s,e} = \mathbf{g}_{s',e'}$ indicates that endpoint e of stick s ($e \in \{1, 2\}$) connects to endpoint e' of stick s'. The value of $\mathbf{G}$ defines a Bayesian network over the motion variables.

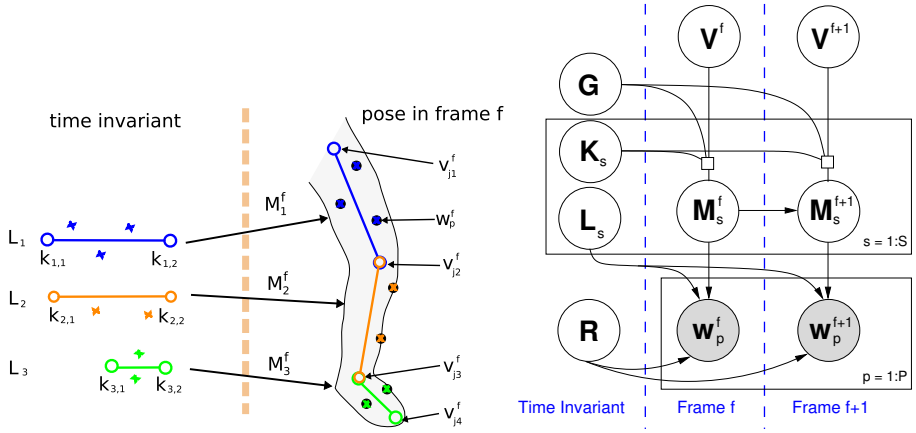

Fig. 2. (Left) The generative process for the observed feature positions, and the imputed positions of the joints. Auxiliary variables specify the local and world locations of the joints. (Right) The graphical model. The motion and geometry variables $(\mathbf{M}_s^f, \mathbf{v}_j^f, \mathbf{L}_s, \mathbf{K}_s)$ are optimized during parameter learning (see Section 3.1); the structure variables ($\mathcal{S} = (\mathbf{R}, \mathbf{G})$) are optimized during structure learning (see Section 3.2)

In allowing connectivity between sticks (possibly even including cycles), the problems of describing the constraints between motions and inferring motions from the observations are made considerably more difficult. To deal with this complexity, we propose the introduction of auxiliary variables called *vertices*, pseudo-observations indicating the locations of the joints and free endpoints 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 $\mathbf{v}_j^f$ represent the world coordinate location of vertex j in frame f, and $\mathbf{K}_s$ specify the coordinates of stick s's endpoints relative to its local coordinate system $\mathbf{L}_s$. Vertices are used to enforce a key constraint: for all the sticks that share a given vertex, the vertex world coordinate locations obtained by applying the respective motion matrices to the local endpoint locations must be consistent. This restricts the range of possible motions to only those resulting in appropriately connected figures. Intuitively, in Figure 2(Left) endpoint 2 of stick 1, $\mathbf{k}_{1,2}$, is connected to endpoint 2 of stick 2, $\mathbf{k}_{2,2}$. Thus in frame f both endpoints must map to exactly the same world coordinate, the location of the knee joint given by vertex $\mathbf{v}_{j2}^f$, *i.e.* $\mathbf{M}_1^f \mathbf{k}_{1,2} = \mathbf{M}_2^f \mathbf{k}_{2,2} = \mathbf{v}_{j2}^f$.

The utility of introducing these auxiliary variables is that, given the vertices $\mathbf{V}$ and endpoints $\mathbf{K}$, 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. The assumed generative process for the feature observations and the vertex pseudo-observations is shown in Figure 2 (Left), and the corresponding probabilistic model is shown in Figure 2 (Right).

The components of the model that comprise the model structure—the feature-stick assignments $\mathbf{R}$ and the articulated graph structure $\mathbf{G}$—are summarized in a single variable $\mathcal{S} = \{\mathbf{R}, \mathbf{G}\}$. The complete joint probability of the model can be decomposed into two factors.

$$\mathrm{P}(\mathbf{W}, \mathbf{M}, \mathbf{L}, \mathcal{S}) = \mathrm{P}(\mathbf{W}, \mathbf{M}, \mathbf{L}|\mathcal{S})\,\mathrm{P}(\mathcal{S}) \tag{1}$$

The first factor, containing the continuous motion and geometry variables, is optimized during parameter learning, while the second, containing discrete assignments and connectivities, is optimized during structure learning. We first consider the parameter learning problem.

3.1 Learning geometry and motion parameters

The aim of parameter learning in the model is to determine a set of motion parameters $\mathbf{M}$ and local point coordinates $\mathbf{L}$ that optimally fit the observed data $\mathbf{W}$, given the model structure $\mathcal{S}$. This distribution can be factored into a likelihood, a motion prior, and a locality prior: $\mathrm{P}(\mathbf{W}, \mathbf{M}, \mathbf{L}|\mathcal{S}) = \mathrm{P}(\mathbf{W}|\mathbf{M}, \mathbf{L}, \mathcal{S})$ $\mathrm{P}(\mathbf{M}|\mathcal{S})\,\mathrm{P}(\mathbf{L}|\mathcal{S})$. For this work, it is convenient to assume a non-informative uniform prior for $\mathrm{P}(\mathbf{L}|\mathcal{S})$. The details of the other factors follow.

Likelihood of observed feature positions Assuming isotropic Gaussian noise with variance σ_w^2, the likelihood function is

$$\mathrm{P}(\mathbf{W}|\mathbf{M}, \mathbf{L}, \mathcal{S}) = \prod_{f,p,s} \mathcal{N}(\mathbf{w}_p^f | \mathbf{M}_s^f \mathbf{l}_{s,p}, \sigma_w^2 \mathbf{I})^{r_{p,s}} \tag{2}$$

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

The distribution over motion matrices, upon introduction of the auxiliary variables, can be written as $\mathrm{P}(\mathbf{M}|\mathcal{S}) = \int \mathrm{P}(\mathbf{M}, \mathbf{V}, \mathbf{K}|\mathcal{S}) d(\mathbf{V}, \mathbf{K})$. The joint distribution inside the integral is composed of three factors capturing constraints on the imputed vertex locations and motion matrices in each frame, and the local vertex locations:

$$\mathrm{P}(\mathbf{M}, \mathbf{V}, \mathbf{K}|\mathcal{S}) = \mathrm{P}(\mathbf{V}|\mathbf{M}, \mathbf{K}, \mathcal{S})\, \mathrm{P}(\mathbf{M}|\mathcal{S})\, \mathrm{P}(\mathbf{K}|\mathcal{S}) \tag{3}$$

Consistent vertex location predictions Since the articulated graph structure $\mathbf{G}$ induces dependencies between the motions of connected parts, the distribution $\mathrm{P}(\mathbf{M}^f|\mathbf{G})$ of the motion matrix components in frame f does not factorize across sticks s. However, given the auxiliary variables $\mathbf{V}^f$ and $\mathbf{K}$ that specify the world and local positions of the vertices respectively, the prior probability over the per-frame motions, $\mathrm{P}(\mathbf{M}^f|\mathbf{V}^f, \mathbf{K}, \mathbf{G})$, *does* factorize across s. In essence, given the auxiliary variables, the articulated SFM problem factorizes into S independent SFM problems.

The important constraint on $(\mathbf{M}^f, \mathbf{V}^f)$ is that the endpoints should coincide, as closely as possible, with the vertices (joints) to which they belong. This can be expressed in the following Gaussian potential function:

$$\mathrm{P}(\mathbf{V}|\mathbf{M}, \mathbf{K}, \mathcal{S}) \propto \prod_{f,s,e} \mathcal{N}(\mathbf{v}_j^f | \mathbf{M}_s^f \mathbf{k}_{s,e}, \sigma_v^2 \mathbf{I})^{g_{s,e,j}} \tag{4}$$

Note that because this constraint is probabilistic, it only needs to be approximately satisfied. This allows the model to capture a degree of non-rigidity in the skeletal structure (*c.f.* [16]), as is illustrated in Figure 3(d).

Motion coherence A final constraint is that the motions be smooth through time:

$$\mathrm{P}(\mathbf{M}|\mathcal{S}) = \prod_{f,s} \mathcal{N}(\mathbf{M}_s^f | \mathbf{M}_s^{f-1}, \sigma_m^2 \mathbf{1}) \tag{5}$$

As with $\mathrm{P}(\mathbf{L}|\mathcal{S})$, the locality prior $\mathrm{P}(\mathbf{K}|\mathcal{S})$ will also be assumed uniform.

This phase of learning in our model solves for the per-frame motion matrix parameters and vertex locations, and the local coordinates of the points and vertices, by optimizing $\mathrm{P}(\mathbf{W}, \mathbf{M}, \mathbf{L}, \mathbf{V}, \mathbf{K}|\mathcal{S})$, which is the product of the three terms specified in Equations (2), (4), and (5). The optimization alternates two steps for a fixed number of iterations or until convergence: vertex locations $\mathbf{V}^f$ are imputed for each frame based on Equation (4), and then these locations are treated as additional observations and a standard EM algorithm for SFM with temporal smoothing [8] is applied to solve for $\mathbf{M}, \mathbf{K}, \mathbf{L}$:

Algorithm 1 Algorithm for optimizing the model parameters given a particular structure.

(1). impute $\mathbf{V}_j$

$V_j^f = \left\langle V_j^f | V_s^f, \mathbf{G} \right\rangle = \sum_{s,e} \mathbf{M}_s^f \mathbf{k}_{s,e}^{g_{s,e,j}} / ||g_j||$

(2). optimize $(\mathbf{M}, \mathbf{K}, \mathbf{L})$

SFM with temporal smoothness $\rightarrow \{M_s^f, \mathbf{K}_s, \mathbf{L}_s\}$

3.2 Learning and validating the model 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 approach in which we use a training set of feature point trajectories to construct a number of hypothesized structures and optimize each structure's parameters, then employ a validation set to evaluate the hypothesized structures. In this section we describe how we use the same probabilistic model to construct hypothesized skeletal structures.

Stick Assignments The first step of structure learning involves hypothesizing an assignment of each observed feature point to a stick. This segmentation step also entails determining the number of sticks. We obtain a segmentation by first creating an empirical prior P($\mathbf{R}$) based on the observations $\mathbf{W}$, then sampling from this distribution. The prior is constructed by computing a feature-point by feature-point affinity matrix, where each pairwise affinity is based on a combination of the consistency in the relationship between the points over frames, and their spatial proximity. We use different affinities for 2D and 3D observations. As in earlier methods [8, 9], we could then obtain a segmentation by applying a clustering algorithm to construct the feature-stick assignments from the affinity matrix.

In our model, we instead compute a distribution over feature-stick assignments, and use this to hypothesize several alternative segmentations. Details on the exact affinity terms and the construction of this distribution over segmentations are discussed in Section 4.1 below.

Articulated Graph Structure The second part of model structure learning involves determining which stick endpoints are joined together. We formulate this in our model as a stick-endpoint by vertex matrix $\mathbf{G}$. Each stick-endpoint is assigned to exactly one vertex. The total number of vertices in the model is at most equal to twice the number of sticks. Valid configurations of this matrix only include cases in which endpoints of a given stick are assigned to different vertices.

We employ an incremental, greedy scheme for hypothesizing a series of graph structures $\mathbf{G}$ given a particular segmentation $\mathbf{R}$. We begin with a fully disconnected graph, so that each vertex corresponds to a single stick endpoint. This

is the first hypothesized graph structure, which corresponds to the multibody factorized SFM problem [5]. At each step, we consider each merge that is valid. A given merge specifies a particular graph structure, a simple modification on an existing structure. For example, two stick endpoints that were previously assigned to their own vertices can be assigned to a common vertex. Or a stick endpoint assigned to its own vertex can be added to a multi-endpoint vertex (a joint). Lastly, two multi-endpoint vertices may be merged into a single, larger multi-endpoint vertex.

The cost of each merge is evaluated by running parameter learning (as described in the previous section) given this structure. The merge cost is the negative log-probability of the full joint probability of the model given $\mathcal{S}$. Since at each step the potential merges all share the same segmentation, the merge cost reduces to $-\log \mathrm{P}(\mathbf{W}, \mathbf{M}, \mathbf{V}, \mathbf{K}|\mathcal{S})$; all three of the required terms (Equations (2), (4), (5)) are computed during the optimization in Algorithm 1. We greedily select the least costly merge, and then evaluate merge costs beginning with this new structure.

Each merge produces a new graph connectivity matrix $\mathbf{G}$. Considered in conjunction with the proposed segmentations, this learning procedure constructs a set of hypotheses about the skeletal structure and geometry of the articulated objects. These various hypotheses are then evaluated against a novel set of validation data. The inference algorithm outlined in Algorithm 1 is run for each structure on the validation data, holding stick and vertex locations fixed. The score of a model is the log probability: $\log \mathrm{P}(\mathbf{W}, \mathbf{M}, \mathbf{V}, \mathbf{K}|\mathcal{S})$.

4 Experimental Results

We evaluate our approach on real data sets of both 2D and 3D point trajectories. Qualitatively, we show that our approach is able to find articulated skeletons that match our intuitive knowledge of a moving object's skeleton. Quantitatively, we are able to show that when used to predict held-out feature locations from previously unseen data, our approach is able to provide a significant improvement in accuracy over single and multibody SFM.

4.1 Experiment details

Stick assignments For 3D trajectory segmentation, pairwise affinities were determined by the variance in the feature locations across frames, combined with a weak spatial prior representing the general belief that points near each other will move together:

$$d(i,j) = Var(\|\mathbf{w}_i - \mathbf{w}_j\|) - \frac{\gamma}{F}\sum_{f=1}^{F} \|\mathbf{w}_i^f - \mathbf{w}_j^f\|^2 \tag{6}$$

where parameter γ weights the relative contribution of the two terms. For 2D trajectory segmentation, rather than using the pairwise variance between point distances, we use the principal angles between locally estimated subspaces [9].

Given this matrix, we used the affinity propagation (AP) algorithm [17] to generate the feature-stick assignments, and to determine the number of sticks. AP contains a self-affinity parameter n, a percentile of the observed affinity values, that determines the degree of over-segmentation. Each run sampled n uniformly from the range $[.5, .95]$. To generate alternative segmentations of the trajectories, we used the Best Max-Marginal First algorithm [18] to pick from amongst the top clusterings determined by AP.

Structure learning When working in 2D, it becomes important to ensure that parts connected by a joint are nearby in 3D (particularly in depth: the axis perpendicular to the camera plane). This can be achieved by the addition of a regularizing constraint on the local coordinates of each stick endpoint, specifying that it should be in the vicinity of the local coordinates of the feature points on that stick. To this end we use the following empirical prior on the endpoint locations $\mathbf{k}_{s,e}$:

$$\mathrm{P}(\mathbf{K}) = \prod_{s,e} \mathcal{N}(\mathbf{k}_{s,e} | \bar{\mathbf{L}_s}, \sigma_k^2 \mathbf{1}) \tag{7}$$

where $\bar{\mathbf{L}_s} = \sum_p L_{s,p}^{r_{s,p}} / ||r_s||$.

Datasets and evaluation metrics We split each dataset into three blocks: the first 60% of the frames used for learning the structure and parameters; the next 20% for validation, to select the single optimal structure; and the last 20% used as a test set to evaluate the selected structure, and compare its performance to other structures. In the validation and test frames, we remove all features from one randomly chosen stick and 10% of the rest of the features, also randomly chosen. This procedure aims to simulate occlusion and feature drop-out. This random selection of missing features is repeated 20 times, and we average the prediction results across the repetitions for both validation and test. We can then compare the predictive performance of the chosen and some other not-selected structures on the held-out feature locations.

Ideally we would compare our algorithm against others on the prediction of held-out data. However, the other algorithms, *e.g.* [11], do not produce detailed enough models to make such predictions on novel frames.

Instead, we use two alternatives as baseline predictors: single-body structure from motion, and multibody SFM (which is equivalent to a fully-disconnected skeleton). In both cases the visible points are used to estimate the motions, then the motions, together with the structures, are used to estimate the locations of the held-out points. This comparison elucidates the quantitative utility of the joints.

Finally, it is important to note that while there are several parameters in the algorithm, only a very small number of them are adjusted for the different datasets. The number of EM iterations in SFM is generally set to 10, but for efficiency is reduced to 1 on the much larger 3D datasets. The segmentation method contains parameters such as γ in (6), which was set to 1/100 in 3D

and 1/1500 in 2D. When constructing the 2D affinity matrix of [9], we used a neighborhood size of 4 and an effective rank parameter of 10^{-6}. The other parameters are not varied for the different data sets; α, the ratio of the feature point variance σ_w^2 to the vertex variance σ_v^2, is set to 4; β, the ratio of the observation variance to the temporal smoothness variance σ_m^2, was set to 0.25. Finally, the marker coordinates for each dataset were rescaled to lie roughly in $[-10, 10]$.

4.2 3D Data: Human Motion Capture

We use 3D point trajectories from the CMU Motion Capture database. The first sequence contains a single person interacting with a box on the floor and has 174 trajectories across 2195 frames. The second sequence contains two people playing catch with a football. There are 327 trajectories across 1285 frames. We selected these sequences due to the marker densities and extensive articulation of the figures, which allows our algorithm to form a detailed, complicated skeleton.

Three optimal skeletons, as selected by validation, are shown in Figure 3(a). These were obtained by running our greedy structure learning algorithm on three different segmentations, and for each selecting the structure with the lowest validation error. From left to right respectively, these structures contain 12, 11 and 23 sticks, and were constructed using 15, 15, and 38 steps of greedy structure learning. These structures obtained test error 39%, 43%, and 59% less error than single-body structure from motion. In comparison, due to its inherent difficulty with fully-occluded sticks, the test error of multibody factorization, over a range of different segmentations, was 7.6 times larger than the largest test error of the three skeletons shown. Although the structure on the left appears to more closely match an intuitive abstraction of a human skeleton, its validation error is slightly higher than that of the other two. This can be attributed to the difficulty of comparing structures with different numbers of sticks.

As an additional comparison, we ran the algorithm of Kirk *et al.* [11] on the same training sequence, and include a representative structure learned by it in Figure 3(b). As can be seen, the segmentation performance is similar to our own, but it has difficulty determining the connectivity of parts and the locations of the joints. These problems may be attributed to the fact that this data does not include a "calibration phase", in which the human fully exercises each joint through its full range of motion.

The optimal skeleton for the two-person video is shown in Figure 3(c). Note that, in contrast with methods based on spanning trees, the algorithm has no trouble finding separate skeletons for each of the two players.

4.3 2D Data: Human and Giraffe

Next, we explored our algorithm's ability to model a 3D scene when given only a 2D view of the feature points. We constructed a set of 2D feature trajectories by taking the single human motion capture sequence and projecting it to a 2D image plane that retained a reasonable amount of information about the motion.

To ensure that quantitative evaluation was fair, we chose the training, validation and test sequences to ensure that they captured roughly the same distribution over the person's range of motions and orientation. Otherwise, we followed the same methodology as in the 3D case.

Optimal skeletons for two different segmentations are shown in Figure 3(e). For the segmentation shown on the left, the validation error was the lowest on the shown structure, with a score of 67.4. Test error was also lowest with a score of 68.8, whereas the test error on the fully disconnected skeleton (multi-body SFM) was 233. For the segmentation on the right, the validation error was the lowest on the shown structure, with a score of 77.3. Test error was also lowest with a score of 80.2, while the fully disconnected skeleton gave a test error of 158. For both of these, the single body SFM test error was much higher at 902.

As an additional experiment, we apply our model to a video of giraffe walking across a plain. We obtain 2D point trajectories by tracking feature points in the video, producing a sequence of 60 trajectories across 130 frames. (Unlike the 2D human motion, in this video the motion of the giraffe is mostly planar. As a result, when solving for SFM there is an inherent degeneracy that, if not handled carefully, will break standard SFM algorithms.) Figure 3(f) shows two representative skeletons learned by our algorithm for the giraffe video, and one of the best skeletons learned appears in Figure 1.

5 Discussion

We have demonstrated a single coherent model that can describe the structures and motion of articulated skeletons. This model can be applied to a variety of structures (including 2D and 3D), requiring no input beyond the observed feature trajectories, and a minimum of manually-adjusted parameters.

To obtain good results, our model requires a certain density of features, in particular because the 2D affinity matrix [9] requires at least 4 points per stick. However in 3D it can work if a few sticks have only a single marker. In addition, the flexibility of learned models are 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. It is important to note that the noise 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.

An important extension to our algorithm would involve iterating between updates of the stick assignments and the connectivity structure, allowing information obtained from one stage to assist learning in the other. Currently we consider multiple hypothesized segmentations, and several structures for each, but there is no provision for reestimating the stick assignments based on an estimated connectivity structure. We also plan to study the ability of learned

models to generalize: applying them to new motions not seen during training, and to related sequences, such as using a model trained on one football player to parse the motion of another.

References

1. Herda, L., Fua, P., Plankers, R., Boulic, R., Thalmann, D.: Using skeleton-based tracking to increase the reliability of optical motion capture. Human Movement Science Journal **20**(3) (2001) 313–341
2. Bray, M., Kohli, P., Torr, P.: Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In: ECCV (2). (2006) 642–655
3. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision **9** (1992) 137–154
4. Hartley, R., Zisserman, A.: Multiple View Geometry. Cambridge University Press (2003)
5. Costeira, J.P., Kanade, T.: A multibody factorization method for independently moving-objects. International Journal of Computer Vision **29**(3) (September 1998) 159–179
6. Dellaert, F., Seitz, S.M., Thorpe, C.E., Thrun, S.: EM, MCMC, and chain flipping for structure from motion with unknown correspondence. Machine Learning **50**(1-2) (2003) 45–71
7. Torresani, L., Hertzmann, A., Bregler, C.: Learning non-rigid 3d shape from 2d motion. In: NIPS. (2003)
8. Gruber, A., Weiss, Y.: Multibody factorization with uncertainty and missing data using the EM algorithm. In: CVPR. (2004) 707–714
9. Yan, J., Pollefeys, M.: A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. (2006)
10. Yan, J., Pollefeys, M.: Automatic kinematic chain building from feature trajectories of articulated objects. In: CVPR. (2006)
11. Kirk, A.G., O'Brien, J.F., Forsyth, D.A.: Skeletal parameter estimation from optical motion capture data. In: CVPR, IEEE Computer Society (2005)
12. Song, Y., Goncalves, L., Perona, P.: Unsupervised learning of human motion. IEEE PAMI **25**(7) (2003) 814–827
13. Taycher, L., III, J.W.F., Darrell, T.: Recovering articulated model topology from observed rigid motion. In Becker, S., Thrun, S., Obermayer, K., eds.: NIPS, MIT Press (2002) 1311–1318
14. Ramanan, D., Forsyth, D.A., Barnard, K.: Building models of animals from video. IEEE PAMI **28**(8) (2006) 1319–1334
15. Krahnstoever, N., Yeasin, M., Sharma, R.: Automatic acquisition and initialization of articulated models. Machine Vision and Applications **14**(4) (September 2003) 218–228
16. Sigal, L., Isard, M., Sigelman, B., Black, M.: Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In: NIPS, MIT Press (2003)
17. Frey, B., Dueck, D.: Clustering by passing messages between data points. Science **315** (February 2007) 972–976
18. Yanover, C., Weiss, Y.: Finding the M most probable configurations in arbitrary graphical models. In Thrun, S., Saul, L.K., Schölkopf, B., eds.: NIPS, MIT Press (2003)

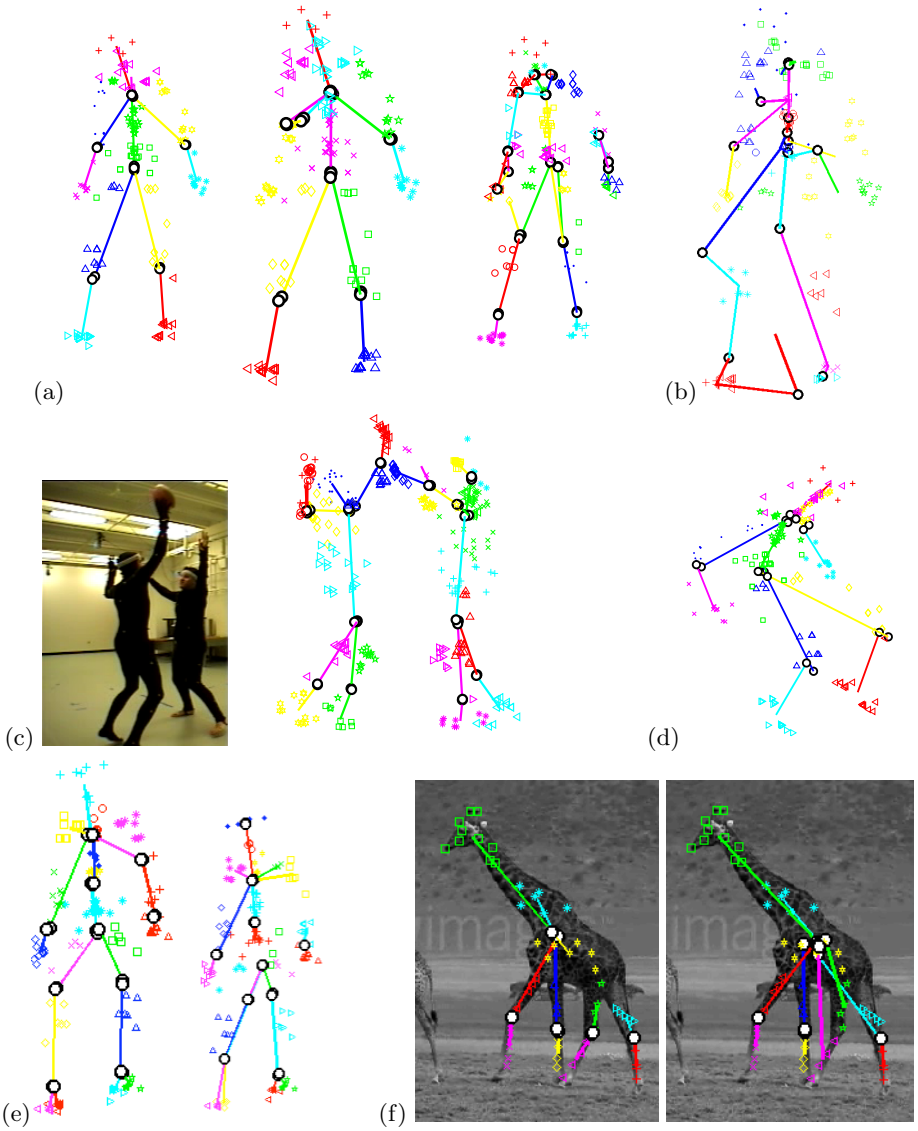

Fig. 3. Experimental Results: (a) The best three skeletons learned for a single human, given trajectories from 3D motion capture. (b) For comparison, the results of the Kirk *et al.* algorithm [11] on the same data. (c) Skeletons learned on data of two humans playing football. (d) Soft joint constraints allow more flexibility in modeling non-rigid deformations, as illustrated here in the knees. (e) The two best validating skeletons learned by our algorithm when given 2D inputs of the single human data. (f) Two skeletons learned by our algorithm on feature trajectories from a video of a walking giraffe. Further illustration of the experiments can be seen in the accompanying videos, available at `http://www.cs.toronto.edu/~dross/articulated/`.