Automatic Kinematic Chain Building from Feature Trajectories of Articulated Objects

Jingyu Yan and Marc Pollefeys
Department of Computer Science
The University of North Carolina at Chapel Hill
Chapel Hill, NC 27599
yan,marc@cs.unc.edu

Abstract

We investigate the problem of learning the structure of an articulated object, i.e. its kinematic chain, from feature trajectories under affine projections. We demonstrate this possibility by proposing an algorithm which first segments the trajectories by local sampling and spectral clustering, then builds the kinematic chain as a minimum spanning tree of a graph constructed from the segmented motion subspaces. We test our method in challenging data sets and demonstrate the ability to automatically build the kinematic chain of an articulated object from feature trajectories. The algorithm also works when there are multiple articulated objects in the scene. Furthermore, we take into account non-rigid articulated parts that exist in human motions. We believe this advance will have impact on articulated object tracking and dynamical structure from motion.

1. Introduction

Recently, analysis and reconstruction of dynamical scenes has attracted more and more attention. Structure from motion of independently moving objects under affine projection were the first to attract efforts[2][9][14][23][18][26]. Secondly, non-rigid structure from motion has been studied thoroughly in [4][19][22][15][3]. The trajectory matrix can be written as a projection matrix combined with a linear combination of a number of key shapes. Last but not the least, articulated structure from motion, as another important paradigm in dynamical scenes, have also received a lot of attention[13][11][25][16][24][17].

Besides structure from motion problem, previous works on articulated objects are focusing on other problems like tracking or pose estimation[6][5][10][7]. However, to our knowledge, automatically building the kinematic chain from feature trajectories of an articulated object has never

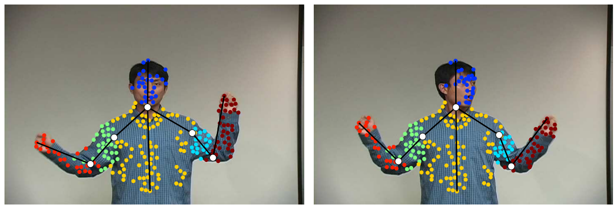

Figure 1. The kinematic chain is automatically computed from feature tracks.

been attempted. The kinematic chain is usually supplied as a prior. We believe the ability to automatically get this information from feature trajectories will impact future work on articulated objects and motions.

We demonstrate the possibility of building the kinematic chain of an articulated object from feature trajectories by proposing an algorithm which first segments the trajectories by local sampling and spectral clustering, then builds the kinematic chain from a graph constructed from the segmented motion subspaces. We test our method in challenging datasets and demonstrate the ability to automatically build the kinematic chain of an articulated object from feature trajectories. The algorithm also works when there are more than one objects in the scene or when some articulated parts are non-rigid.

In the following sections, we discuss motion subspaces in Section 2, which is a prior to understand our algorithm; we describe our algorithm in Section 3; we demonstrate the algorithm in Section 4 and draw conclusion and discuss future work in Section 5.

2. Motion Subspaces

In order to explain our algorithm, we need to introduce the concept of motion subspace, which is the subspace that contains the trajectories of a certain type of object undergoing a certain type of motion. Our paper focuses on articulated objects. But we will start from single rigid objects and multiple independently moving objects.

2.1. Rigid motion subspace for single and multiple independently moving objects

We discuss the motion subspace of a single rigid object and independently moving objects in the following in order for a better understanding of articulated motion subspaces in the next section.

- The trajectories of a rigid object forms a linear subspace of dimensions no more than 4 [8].

$$M_{2F\times p} = [R_{2F\times 3}|T_{2F\times 1}]\begin{bmatrix} S_{3\times P} \\ \mathbf{1}_{1\times P} \end{bmatrix} \tag{1}$$

  F is the number of frames and P, the number of feature trajectories. The subspace spanned by the columns of M is called a rigid motion subspace.

- For multiple independent rigid motions, the trajectory matrix can be written as the following given that the trajectories are properly grouped. R_i, T_i and S_i represents the rotation, translation and shape of the ith object [14]. For a total of N objects we have the following.

$$W = (R_1|T_1|...|R_N|T_N)\begin{pmatrix} S_1 & & \\ 1 & & \\ & ... & \\ & & S_N \\ & & 1 \end{pmatrix} \tag{2}$$

  Each object has its own rotation and translation while the shape matrix consists of columns belonging to orthogonal shape subspaces. Thus the motion subspace of each object is independent and of dimension no more than 4.

2.2. Articulated objects with rigid parts

It has recently been shown that for articulated objects with rigid parts, the motion subspaces are not independent [16][24]).

- If the link is a joint, we can make it the origin of the world coordinate. Then $[R_1|T_1]$ and $[R_2|T_2]$ of two linked parts have $T_1 = T_2$ under that coordinate system. So the trajectory matrices of M_1 and M_2 lie in different linear subspaces of dimension no more than 4 but have a one-dimensional intersection.

- If the link is an axis, we can make it an axis of the world coordinate, e.g the z axis. Then $[R_1|T_1]$ and $[R_2|T_2]$ have $T_1 = T_2$ and the last column of R_1 and R_2 being the same. So M_1 and M_2 lie in different linear subspaces but have a two-dimensional intersection.

To sum up, the trajectories of an articulated object with rigid parts lie in a mixture of linear subspaces, each of which is of dimension no more than 4 and some of which are intersecting in 1 or 2 dimensions depending on whether the two parts are linked by an axis or a joint. The intersecting property of these motion subspaces is what our algorithm relies on to build the kinematic chain.

2.3. Extension to non-rigid parts

In this section, we extend our discussion to articulated objects with non-rigid parts. A case in point is the human motion whose facial motion is non-rigid and whose head and body motions combined can be considered as articulated. We will discuss non-rigid motion subspace first; then we will focus on a typical non-rigid case and build some theorems; lastly, we discuss how this typical case can fit into the articulated motion subspace discussed above.

The trajectories of a non-rigid object can be approximated by different linear combinations of a number of, e.g. K, key shapes [4][22]. For the fth frame, the image coordinates of the features are m^f.

$$m^f_{2\times P} = [R^f_{2\times 3}|T^f_{2\times 1}]\begin{bmatrix} \sum_{i=1}^{K} c^f_i S^i_{3\times P} \\ \mathbf{1}_{1\times P} \end{bmatrix} \tag{3}$$

Putting all frames together, we have the trajectory matrix written as the following, which forms a linear subspace of dimensions no more than $3K+1$.

$$M = \begin{bmatrix} c^1_1 R^1_{2\times 3}|...|c^1_K R^1_{2\times 3}|T^1_{2\times 1} \\ ... \\ c^F_1 R^F_{2\times 3}|...|c^F_K R^F_{2\times 3}|T^F_{2\times 1} \end{bmatrix}\begin{bmatrix} S^1_{3\times P} \\ ... \\ S^K_{3\times P} \\ \mathbf{1}_{1\times P} \end{bmatrix} \tag{4}$$

where c^i_j $(1 \le i \le F, 1 \le j \le K)$.

Let us consider a typical case: the non-rigid shape has rigid components. This includes human facial motion which deforms on top of rigid head motion. More formally, we are considering such a case that a non-rigid shape can be represented by linear combinations of a number of key shapes S^1,...,S^K that satisfy $S^1_i = ... = S^K_i$ as long as its ith component is rigid.

We can prove then the following theorems.

Theorem 1. *If a non-rigid shape can be represented by linear combinations of S^1,...,S^K that satisfy $S^1_i = ... = S^K_i$ for any rigid component i, the sum of the linear coefficients of any frame f is 1, i.e. $\sum_{i=1}^{K} c^f_i = 1$.*

Proof. Let S_i be a rigid component of the non-rigid shape. For any frame f, its 2D coordinates are:

$$M^f_i = [c^f_1 R^f|...|c^f_K R^f|T^f]\begin{bmatrix} S_i \\ ... \\ S_i \\ 1 \end{bmatrix} \tag{5}$$

Because S_i is rigid, M_i^f can also be written as the following.

$$M_i^f = [R^f|T^f] \begin{bmatrix} S_i \\ 1 \end{bmatrix} \quad (6)$$

By comparing Equation 5 and 6, we have $\sum_{i=1}^{K} c_i^f = 1$. □

Theorem 2. *If a non-rigid shape can be represented by linear combinations of S^1,...,S^K that satisfy $S_i^1 = ... = S_i^K$ for any rigid component i, the rigid motion subspace formed by the rigid components is embedded in the non-rigid motion subspace.*

Proof. From Theorem 1, we know $\sum_{i=1}^{K} c_i^f = 1$ for any frame f. Let S_I be the set of all rigid components. We have the following.

$$\begin{bmatrix} c_1^1 R^1|...|c_K^1 R^1|T^1 \\ ... \\ c_1^F R^F|...|c_K^F R^F|T^F \end{bmatrix} \begin{bmatrix} S_I \\ ... \\ S_I \\ 1 \end{bmatrix} = \begin{bmatrix} R^1|T^1 \\ ... \\ R^F|T^F \end{bmatrix} \begin{bmatrix} S_I \\ 1 \end{bmatrix} \quad (7)$$

Notice the left are trajectories in the non-rigid motion subspace; the right are trajectories in the rigid motion subspace formed by all rigid components. So the rigid motion subspace formed by those components must be embedded in the non-rigid motion subspace. □

Now we can deal with articulated objects with non-rigid parts that satisfy the above specification. The result is similar to the rigid case because essentially it is the embedded rigid motion subspace that interacts with its linked part. This result remains valid if both linked parts are non-rigid[1].

- If the link is a joint, two subspaces have in general a one-dimensional intersection.
- If the link is an axis, two subspaces have in general a two-dimensional intersection.

Notice that for either case, we do not need to extract the embedded rigid motion subspace out of the non-rigid one in order to find the intersection. We intersect the motion subspaces directly.

3. Automatic Kinematic Chain Building From Feature Trajectories

In this section, we describe the algorithm of building the kinematic chain from feature trajectories of an articulated object under affine projection. It consists of two stages: at the first stage, trajectories are segmented according to the articulated parts and the motion subspaces are formed after rejecting outliers; at the second stage, the proximities between these motion subspaces is computed to build a proximity graph, then a minimum spanning tree algorithm is performed on the graph to retrieve the kinematic chain information of the articulated object.

3.1. Motion Segmentation

The trajectories of an articulated object are from different rigid or non-rigid parts which form different motion subspaces that may intersect one or the other. The motion segmentation stage is to segment them accordingly. Notice that these motion subspaces are not independent from each other and there are dependencies between those of linked parts. Thus motion segmentation algorithms for independent motions[14][23][18] are not appropriate in this situation. Though [26] may be capable to segment dependent motions theoretically, in practice the sampling size required is too big to satisfy when the number of motion subspace increases. For more details, please refer to [1].

We use the algorithm proposed in [1] which can segment rigid or non-rigid motion subspaces when they are either independent or dependent. The algorithm is described in the following.

- Trajectory Data Transformation
  Transform each trajectory of dimension $2F$ (F is the number of frames) onto a R^K unit sphere ($rank(W_{2F\times P}) = K$, $W_{2F\times P}$ is the trajectory matrix). This can be done by SVD, $W_{2F\times P} = U_{2F\times K} D_{K\times K} V_{P\times K}^T$, and normalizing each row of V. Each unit vector $v_i (i = 1...P)$ becomes the new representation of the corresponding trajectory.
- Local Subspace Estimation
  Without knowing the underlying subspace each v_i belongs to, we estimate it from itself and its $n-1$ closest neighbors, i.e. computing the subspace of $[v_i, v_{i1}, ..., v_{in}]_{K\times(n)}$ using SVD. n is normally chosen to be larger than the dimension of the underlying subspace.
- Spectral Clustering
  An affinity matrix can be built from the distance between every pair of the locally estimated subspaces for each v_i. Then we can perform spectral clustering and segment the trajectories. The distance between two equidimensional subspaces is typically represented by the sine of their largest principle angle[12] (see Section 3.2 for the definition of principle angles). In our case, we define the affinity as below.

  $$\textit{affinity}(\alpha, \beta) = e^{-\sum_{i=1,...,M} sin^2(\theta_i)}$$

  where θ_1,...,θ_M are the principal angles between two locally estimated subspaces α and β.

[1]For cases where the non-rigid deformations between the articulated parts are dependent, it might be possible that higher dimensional intersections are obtained.

After segmenting the trajectories, we perform outlier rejection within each segment. This can be done using a RANSAC approach [21] that robustly fit the data into a subspace and reject outliers. The motion subspaces are formed by the remaining trajectories in each group.

3.2. Kinematic Chain Building

Given the motion subspaces, either rigid or non-rigid, their dependency on each other are measurable by their minimum principle angles between every pair of them. The principal angles [12] between two subspaces are defined recursively as a series of angles $0 \leq \theta_1 \leq, ..., \leq \theta_M \leq \pi/2$ (M is the minimum dimension of both subspaces):

$$\cos(\theta_m) = max_{u \in S^1, v \in S^2} u^T v = u_m^T v_m$$

where

$$\begin{aligned} \|u\| &= \|v\| = 1 \\ u^T u_i &= 0 \quad i = 1, ..., m-1 \\ v^T v_i &= 0 \quad i = 1, ..., m-1 \end{aligned}$$

For two linked parts, either rigid or non-rigid, either for a joint link or an axis link, their motion subspaces are intersecting on at least one dimensional subspace (see Section 2), thus have at least one zero principle angle. In practice, the value will not be exact zero so a threshold is required. For parts that are not linked, the motion subspaces do not have this property and have a larger minimum principle angle. Depending on how independent the motion subspaces are, the minimum principle angles may vary.

Based on the above analysis, we will describe our kinematic chain building algorithm in the following.

- Build the proximity graph
  We use a graph to represent the proximity between every pair of motion subspaces.

  $$G = (V, E)$$

  where $V = \{v_1, ..., v_S\}$ (v_i is the ith motion subspace; S is the number of motion subspaces) and $E(v_i, v_j) = \theta_{ij}$ (θ_{ij} is the minimum principle angle between subspace i and j).

- Find the minimum spanning tree(s)
  Based on the proximity graph we find a minimum spanning tree using Algorithm 1. The spanning tree corresponds to the kinematic chain that we compute.

  With a small modification, our algorithm can handle multiple articulated objects in the scene and find multiple kinematic chains (see Algorithm 2). The key is that whenever the smallest edge connecting T to G-T is over some threshold, we stop spanning the current tree and start building another spanning tree using the same procedure.

Algorithm 1 Finding single minimum spanning tree

```
let T be the graph of the smallest edge of G
while T has fewer than S − 1 edges do
    find the smallest edge in G connecting T to G − T
    add it to T
end while
```

Algorithm 2 Finding multiple minimum spanning trees

```
let P = G
while P has more than one edge < threshold do
    let T be the minimum edge of P
    while the smallest edge connecting T to P − T <
    threshold do
        add it to T
    end while
    save T as a kinematic chain
    let P = P − T
end while
```

4. Experiments

We test our algorithm in two kind of data sets, synthetic and real.

4.1. Synthetic tests

The first synthetic test demonstrates our method in a highly challenging case in which the kinematic chain is automatically built from a human model of 10 parts including the head, the body, two upper arms, two lower arms, two thighs and two legs. Each part has 20 trajectories. And the trajectories are perturbed by 15% noise. The segmentation result is shown in Figure 2 with bigger black dots showing the 10 misclassified points. Notice that the misclassification happens mostly around the joints and axes. After outlier rejection within each segment, the remaining 171 features are shown in Figure 2.

The minimum principle angles (the proximity graph) between 10 motion subspaces are shown in Table 1. The bold font indicates the edges of the minimum spanning tree of the graph. The kinematic chain are built from that. The recovered kinematic chain is correct.

We show the second minimum principal angles between parts in Table 2 to see how the second dimension intersection between motions may be detected. The bold font shows the angles that are much lower than the average, which indicates a second dimensional subspace intersection between motions and thus indicates that there is an axis link (See Section 2.2. The data shown in the table matches our synthetic model in which the upper arms are connected with the lower arms with axis links and so are the thighs with the legs.

In summary, in order to detect links for kinematic chain

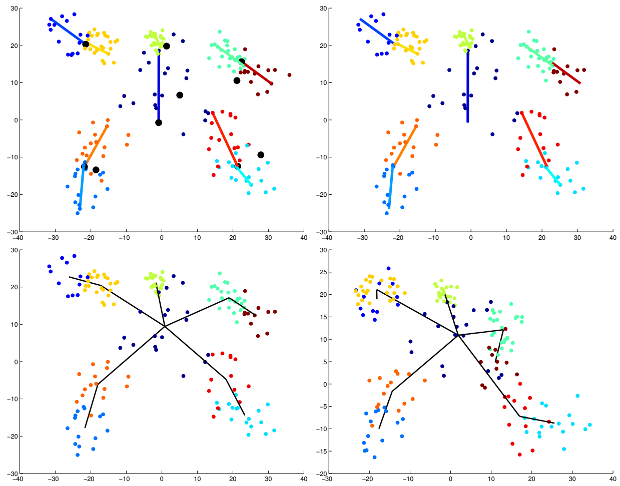

Figure 2. (*top left*) The segmentation of trajectories over 10 articulated parts of a synthetic human model. The sticks are shown for illustration purpose. (*top right*) Trajectories after rejecting outliers. (*bottom left and right*) The kinematic chain (black lines) built from trajectories.

Table 1. Proximity graph and its minimum spanning tree – Synthetic human model

$\times 10^{-4}$	ruarm	rleg	lleg	llarm	head	rlarm	rthigh	lthigh	luarm
body	1643	1519	1524	**254**	**221**	**303**	**340**	**397**	1672
ruarm		3106	3037	7719	3735	**238**	2606	2922	8276
rleg			2080	2344	3117	2935	**261**	1609	2906
lleg				3040	3546	2813	1499	**291**	3075
llarm					2805	6661	2302	2519	**236**
head						2612	3067	3252	3593
rlarm							2382	2623	7237
rthigh								1454	2774
lthigh									2700

building, inspecting the minimum principal angle is enough. The second minimum principal angle can be used for further determining the type of the link.

The second synthetic test will demonstrate our algorithm applied to multiple articulated objects. We generate two synthetic human models in a scene. The motion segmentation and the outlier rejection steps proceed as described before. At the kinematic chain building step, we use a threshold of 0.1000 for determining whether the algorithm should stop adding new links to the current chain.

The kinematic chains built from the trajectories are shown in Figure 3.

The minimum principal angles (proximity graph) between the motions of the same synthetic human model is similar to Table 4. The minimum principal angles between the motions of the two different synthetic human models is shown in Table 3. Obviously, the magnitude of these angles is far above that of the ones between motions of linked parts. The point to be made here is that, though our algorithm needs a threshold value for building multiple kinematic chains, this threshold value is not very sensitive.

Table 2. The second minimum principal angles between parts of the synthetic human model

$\times 10^{-4}$	ruarm	rleg	lleg	llarm	head	rlarm	rthigh	lthigh	luarm
body	8279	3298	3687	6768	4263	6149	3123	3348	8826
ruarm		8634	7899	8583	8504	**1521**	8570	7766	9446
rleg			5131	7887	5219	7622	**1275**	4807	9156
lleg				8644	5084	6879	4956	**1102**	9742
llarm					7912	8402	7267	8582	**1515**
head						7633	4748	4815	9378
rlarm							7579	6745	9440
rthigh								4639	8719
lthigh									9437

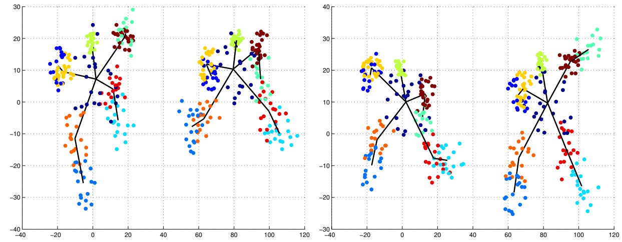

Figure 3. (*left and right*)Two kinematic chains (black lines) built from trajectories of a two-person scene.

Table 3. The minimum principal angles between motions of two synthetic human models

$\times 10^{-4}$	body	ruarm	rleg	lleg	llarm	head	rlarm	rthigh	lthigh	luarm
body	1605	2144	2163	2111	2123	2014	2084	2051	2060	2050
ruarm	2741	5965	3589	3931	6247	4901	5918	3253	3624	6280
rleg	2205	2396	2627	2304	2266	2222	2283	2356	2209	2201
lleg	2854	3077	2874	2851	2957	2858	2995	2824	2862	2919
llarm	2360	4282	4034	4030	4824	3882	4303	3149	3374	4624
head	1646	4844	4160	4558	4861	4636	4833	3208	3683	4823
rlarm	1981	5948	3316	3730	6224	4825	5901	2866	3315	6273
rthigh	2010	2153	2248	2080	2098	2000	2089	2047	2022	2023
lthigh	1995	2174	2238	2074	2092	1976	2109	2027	1976	2020
luram	1658	4324	4060	3740	4860	3726	4344	2845	2983	4642

4.2. Real tests

The first real example is an articulated puppet with 6 rigid parts: the head, the upper body, the hip, 2 arms and 2 legs. A KLT tracker tracks a total of 114 features over 564 frames. The segmentation result is shown in Figure 4. After outlier rejection within each segment, the remaining

Table 4. Proximity graph and its minimum spanning tree – Puppet

	larm	lleg	hip	rarm	body
rleg	0.0111	0.0007	**0.0002**	0.0126	0.0006
larm		0.0110	0.0060	0.0250	**0.0008**
lleg			**0.0002**	0.0170	0.0006
hip				0.0175	**0.0005**
rarm					**0.0003**

Table 5. Proximity graph and its minimum spanning tree – Person

	luarm	ruarm	body	rlarm	llarm
head	0.0015	0.0033	**0.0011**	0.0035	0.0065
luarm		0.0036	**0.0008**	0.0058	**0.0009**
ruarm			**0.0008**	**0.0003**	0.0145
body				0.0018	0.0033
rlarm					0.0103

97 features are shown in Figure 4.

The minimum principle angles (the proximity graph) between 6 motion subspaces are shown in Table 4. The bold font indicates the edges of the minimum spanning tree of the graph. The kinematic chain are built from that and the links are recovered by intersecting linked subspaces[16][24] based on the kinematic chain (Figure 4). The recovered kinematic chain is correct. However, one can notice that the hip-body link is only marginally preferred to the lleg-body link. The reason for this is that the motion of the puppets leg is mostly restricted to a plane orthogonal to the image so that it is hard to differentiate between the legs and the hips.

The second real example is an upper body motion of a person with 6 parts: the head, the upper body, 2 upper arms and 2 lower arms. The head has some non-rigid facial motion. A KLT tracker tracks the total of 268 features over 40 frames. The segmentation result is shown in the top left of Figure 5. After outlier rejection within each segment, the remaining 97 features are shown in the top right of Figure 5. The minimum principle angles (the proximity graph) between 6 motion subspaces are shown in Table 5. The bold font indicates the edges of the minimum spanning tree. The kinematic chain are built from the 6 motion subspaces and the links are recovered by intersecting subspaces[16][24] based on the kinematic chain, shown in the bottom of Figure 5.

The non-rigid part is the head which has a joint link with the upper body. The link can be recovered simply by finding the 1-dimensional intersection between both motion subspaces as discussed in Section 2.3.

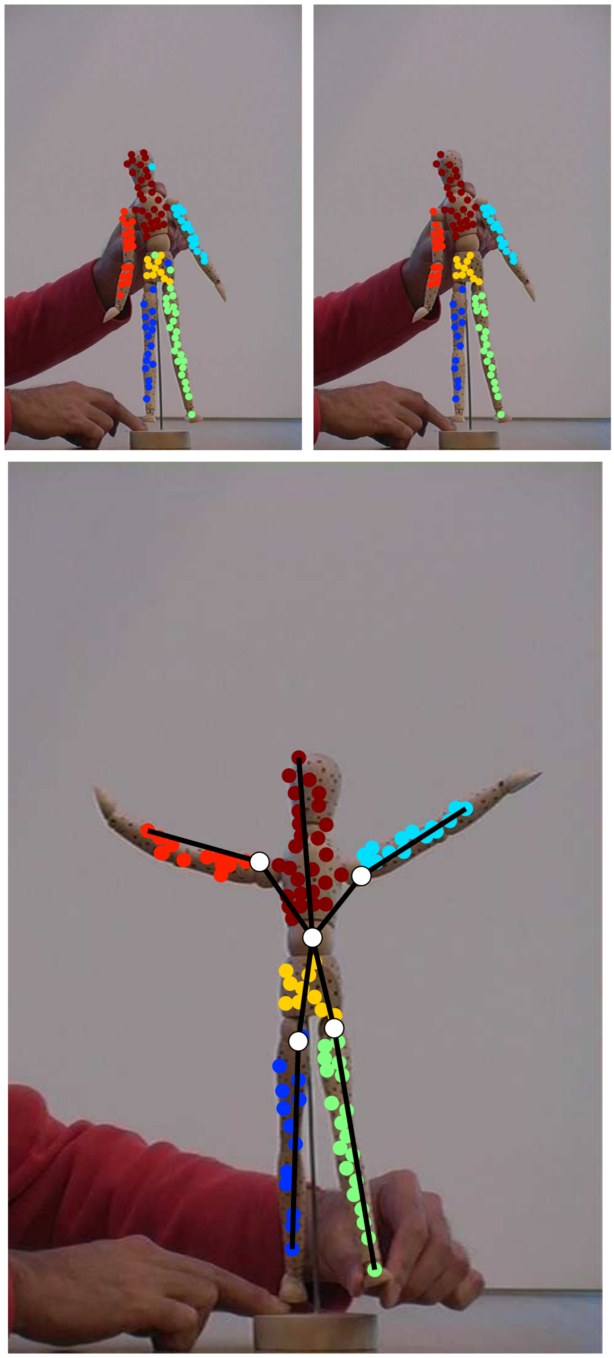

Figure 4. (*top left*) The segmentation of trajectories over 6 articulated parts of a puppet. (*top right*) Trajectories after rejecting outliers. (*bottom*) The kinematic chain built from trajectories and the links (white dots) recovered by intersecting the linked subspaces based on the kinematic chain.

5. Conclusion and Future Work

We propose an algorithm that builds a kinematic chain from feature trajectories of an articulated object under affine projections. The algorithm can be extended to handle scenes of multiple articulated objects and articulated non-rigid parts. It first segments trajectories according to the articulated parts; then it rejects outliers; in the end, it builds

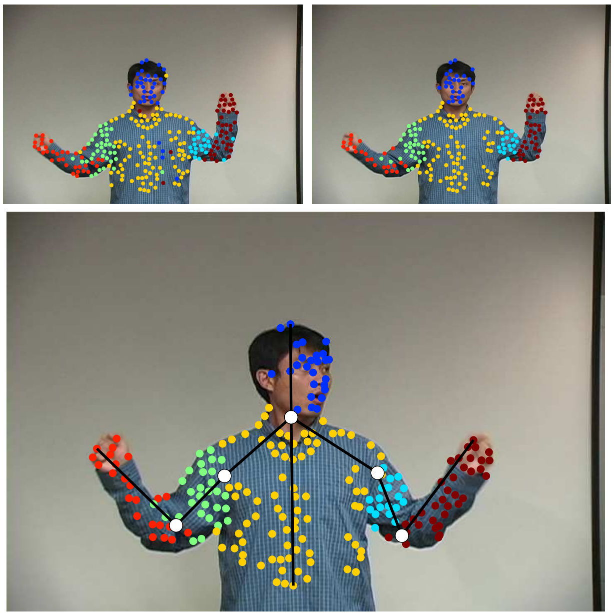

Figure 5. (*top left*) The segmentation of trajectories over 6 articulated parts of an upper-body human motion. (*top right*) Trajectories after rejecting outliers. (*bottom*) The kinematic chain built from trajectories and the links (white dots) recovered by intersecting the linked subspaces based on the kinematic chain.

a kinematic chain from a minimum spanning tree of a proximity graph that is constructed from the minimum principle angles between the motion subspaces of the articulated parts.

We plan to handle occlusions and missing trajectories in the near future which will make our algorithm more practical. Ultimately, we aim to recover articulated human motion with non-rigid parts.

By learning the structure of an articulated object, constraints can be automatically imposed on the motions of the object on the fly which may extend the existing tracking or shape from motion applications of articulated objects.

Acknowledgments

The support of the NSF ITR grant IIS-0313047 is gratefully acknowledged.

References

[1] A General Framework for Motion Segmentation: Independent, Articulated, Rigid, Non-rigid, Degenerate and Non-degenerate, ECCV 2006 Submission ID 795, supplied as additional material eccv06submission.pdf

[2] Boult, T. and Brown, L. 1991. Factorization-based segmentation of motions. In Proceedings of the IEEE Workshop on Visual Motion.

[3] Brand, M., ”A Direct Method for 3D Factorization of Non-rigid Motion Observed in 2D”, CVPR 2005

[4] C. Bregler, A. Hertzmann, H. Biermann, ”Recovering Non-Rigid 3D Shape from Image Streams”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’00), June 2000.

[5] C. Bregler, J. Malik, “Tracking people with twists and exponential maps”, CVPR, pp. 8–15, 1998.

[6] C.J. Taylor, ”Reconstruction of Articulated Objects from Point Correspondences in a Single Image”, CVPR, pp. 677, 2000.

[7] C. Sminchisescu and B. Triggs, ”Covariance Scaled Sampling for Monocular 3D Body Tracking”, Proc. CVPR, vol. I, pp. 447-454, 2001.

[8] C. Tomasi, T. Kanade, ”Shape and motion from image streams under orthography: a factorization method”, *IJCV*, Vol. 9, Issue 2 pp. 137-154, 1992.

[9] C.W. Gear 1994. Feature grouping in moving objects. In Proceedings of the Workshop on Motion of Non-Rigid and Articulated Objects, Austin, Texas

[10] D. Hogg. “Model-based vision: A program to see a walking person”, Image and Vision Computing, 1(1):5–20, 1983.

[11] D. Sinclair, L. Paletta and A. Pinz, ”Euclidean Structure Recovery through Articulated Motion”, Proc. 10th Scandinavian Conference on Image Analysis, Lappeenranta, Finland, 1997, pp. 991-998.

[12] G. Golub and A. van Loan. Matrix Computations. Johns Hopkins U. Press, 1996

[13] H. Zhou, T. Huang, ”Recovering Articulated Motion with a Hierarchical Factorization Method”, Gesture Workshop 2003: 140-151

[14] J.P. Costeira, T. Kanade, ”A Multibody Factorization Method for Independently Moving Objects”, *IJCV*, Vol. 29, Issue 3 pp. 159-179, 1998.

[15] J. Xiao, J. Chai, and T. Kanade. “A closed-form solution to non-rigid shape and motion recovery”, Proceedings of the European Conference on Computer Vision, 2004.

[16] J. Yan, M. Pollefeys, A Factorization-based Approach to Articulated Motion Recovery, IEEE Conf. on Computer Vision and Pattern Recognition, 2005

[17] J. Yan, M. Pollefeys, Articulated Motion Segmentation Using RANSAC With Priors, ICCV Workshop on Dynamical Vision, Beijing, China, 2005

[18] K. Kanatani. Motion segmentation by subspace separation and model selection:model selection and reliability evaluation. Intl. J. of Image and Graphics, 2(2):179-197, 2002.

[19] L. Torresani, D. B. Yang, E. J. Alexander, and C. Bregler. "Tracking and modeling non-rigid objects with rank constraints". CVPR, pages 493–500, 2001.

[20] L. Zelnik-Manor and M. Irani. Degeneracies, dependencies and their implications in multi-body and multi-sequence factorizations. In Proc. IEEE Computer Vision and Pattern Recognition, 2003.

[21] M. A. Fischler, R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, Vol 24, pp 381-395, 1981.

[22] M. Brand, "Morphable 3D models from video", CVPR, pp. II:456-463, 2001.

[23] N. Ichimura. Motion segmentation based on factorization method and discriminant criterion. In Proc. IEEE Int. Conf. Computer Vision, pages 600-605, 1999.

[24] P. Tresadern and I. Reid, Articulated Structure From Motion by Factorization, Proc IEEE Conf on Computer Vision and Pattern Recognition, 2005

[25] R. J. Holt, T. S. Huang, A. N. Netravali, and R. J. Qian, " Determining articulated motion from perspective views: A decomposition approach". Pattern Recognition, 30:1435-1449, 1997.

[26] R. Vidal and R. Hartley. Motion Segmentation with Missing Data using PowerFactorization and GPCA. IEEE Conference on Computer Vision and Pattern Recognition, 2004