Scanning Large-Scale Articulated Deformations

Jochen Lang[a] Dinesh K. Pai[b] Hans-Peter Seidel[a]

[a] MPI Informatik, Saarbrücken, Germany.

[b] Rutgers, the State University of New Jersey, Piscataway, NJ, USA.

Abstract

Scanning the deformation behavior of real objects is a useful technique to acquire physically realistic behavior. It has been shown previously how to acquire physically realistic object behavior for small deformation and how to render the behavior at interactive rates. This paper introduces a novel method to extend previous work to handle large scale deformation. We model large scale deformation as articulation in combination with local linear deformation. The articulation may either reflect a underlying physical structure or may be purely a modeling technique. In this paper, we show examples of both applications. Our acquisition method is applicable to deformable modeling but it also has implications for motion capture.

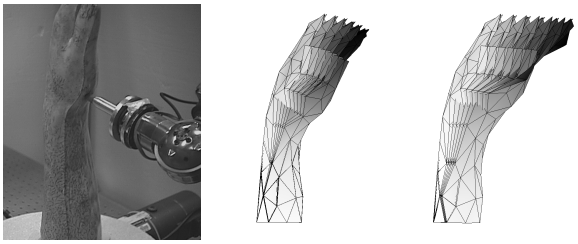
Key words: Reality-Based Modeling, Deformable Solids, Articulated Models, Motion Capture.

1 Introduction

Deformable models are of significant interest in computer graphics; after all, many animated characters and objects deform as they move and interact. There has been a lot of work in modeling deformable objects analytically in computer graphics (see Section 2). Very recently, deformable models have also been acquired from measurements of real objects (e.g., [17, 13]). Just as motion capture has led to more realistic human motion models, capturing the deformation behavior of real objects and modeling them efficiently could increase the realism of deformable models for visual and haptic simulation. It also promises to make deformable object modeling very easy.

A few examples of application areas for deformation modeling include interactive medical simulation (both for training and for support in the operating room), modeling objects for animation and games, and capturing the behavior of actors for animation including human motion capture.

In computer graphics the deformations of interest are visible, by definition. Visible deformations are of a large scale in contrast to typical engineering applications. In engineering applications the quantity of interest is mainly the internal stress due to loads associated with typically very small strain; in such applications the strain tensor is often approximated to be linear. The linear approximation of the Green-Lagrange strain tensor introduces artifacts during large scale deformation (see, for example, [4, 18]). This is true in particular if the displacement of points on the surface of a solid during deformation is not well approximated by a straight line path. Consider Figure 1(c) which shows the simulation of a hand being pushed at its back to bend around the wrist. The simulation is based on a linear deformation model acquired by scanning a medical training phantom. The simulation shows that the linear deformation model exhibits increasing distortions for larger deformation. During deformation, points on the surface of the object describe a straight line path rather than the expected rotational and translational path. Because of the distortion, the volume of the object is increasing which is clearly a non-physical phenomenon [23, 18, 15].



(a) (b) (c)

Figure 1: Single compliance measurement in ACME (1(a)) and simulation with an estimated articulated deformation model (1(b)) and linear deformation model (1(c)).

The distortion of the deformed object can be avoided by utilizing the second order Green-Lagrange tensor [23, 18, 4] instead of the linear approximation. This makes simulation of such models significantly more complicated since the stiffness matrix relating forces and displacements depends on the initial displacement. A promising alternative, first suggested by Terzopoulos et al. [21], and more fully developed recently [9, 15] is to combine articulated and deformable models. These models (called Elastokinematic models in [9]) promise

the speed of linear deformable models while being able to capture non-linear effects in large scale deformation. However, all work in this direction to date has only addressed analytically computed models and not models of existing objects.

In this paper, we show for the first time how such articulated deformable models can be scanned from real objects. Our method acquires models by measuring the response of actual physical objects. No prior segmentation of the body into links is required — we infer the structure directly from the measurements. The method produces a relationship between applied traction over the surface of the measured solid to resulting displacement of the surface. The model is suitable for rendering realistic visual deformations at interactive rates. The method is also applicable to segment and model dense displacement fields in motion capture, based on visually captured scene-flow.

2 Related Work

An overview of approaches to physical and non-physical deformation modeling can be found in the review by Gibson and Mirtich [5]. Elastically deformable models were introduced to computer graphics by Terzopoulos et al. [20]. Since then, numerous physics-based simulation techniques have been developed, many addressing computational efficiency and physical realism of the model. One approach to interactive physical deformation models is to use linear models in quasi-static simulations based on finite elements [2] or based on boundary elements [8]. Pai et al. [17] have also shown how to capture linear quasi-static models by scanning physical deformation behavior of objects.

There have been various attempts to overcome the limitation of quasi-static linear models while maintaining interactivity. James and Pai [9] use multi-zone elastic models to accommodate articulation. Müller et al. [15] use a per-element rotation to overcome the lack of rotational invariance of linear models. Zhuang and Canny [23] use a quadratic strain tensor in a dynamic finite element simulation. They use mass lumping to diagonalize the mass matrix in order to improve performance. Picinbono et al. [18] follow a similar approach but also introduce adaptivity by using a linear model when appropriate. The success of these dynamic finite element approaches is still limited to models discretized at a coarse resolution. Adaptive resolution techniques [22, 4, 6] aim to address this issue. However, the instability of dynamic elastic simulation is still an open issue, while quasi-static models are inherently stable. In this paper, we use a quasi-static linear model for simulation combined with rotational pseudo-articulation. Our model is based on scanned deformation behavior and extends the work of Pai et al. [17].

Scanning deformation behavior of an object entails the estimation of a relationship between a contact force and the displacements of its surface. It is a different task than, e.g., shape estimation in computer vision which may employ deformable models for regularization. Deformable models have been used successfully in tracking and segmenting a human body by Kakadiaris and Metaxas [10]. However, the most common way to relate a skeleton to video data is by tracking an *a priori* model. Bregler and Malik [1] describe an approach to track a human during articulated motion. Plänkers and Fua [19] use detailed shape models in their tracker. In our work, we do not make use of *a priori* articulation structure (i.e. a skeleton).

In the typical motion capture set-up, the kinematic structure is known and motion estimates are linked to individual limbs. O'Brien et al. [16] describe a technique to adapt a skeleton to data. They solve the problem of fitting data to a known structure given only rotational joints. However, they also describe a graph based method to infer the structure of a kinematic tree given measured motion of known links. The recovery of parameters of an articulated structure from measurements is solved in robot calibration. Hollerbach and Wampler [7] review this large body of work.

Finally, for rendering simulations of our results we use simple linear blending which is also called Skeletal-Subspace Deformation [14].

3 Deformation Model

We use a "discrete Green's functions matrix" to model the global deformation of a solid in static equilibrium. This discrete representation can be derived analytically based on continuum mechanics [2, 8] and it can be directly estimated from observation of deformation behavior [13]. In this paper, we assume that a discrete Green's functions matrix is available for the object we would like to model. The matrices in our examples are estimated from measurements with the ACME facility [17] (see Figure 1(a)).

For a given boundary configuration, the discrete Green's function matrix $\mathbf{\Xi}$ relates the block vector of prescribed and complementary boundary values of displacement and traction vectors. The boundary configuration in our examples divides the surface in a support surface where the object is attached to a fixed support and the free remainder of the surface. Discrete Green's function matrices approximate the surface displacement due to a load by straight lines. In the remainder of this paper we write the discrete Green's function model of deformation as a sum as in Equation 1. The $3N \times 3$ block vector $\mathbf{\Xi}_k$ specifies a displacement field at the N vertices of the free sur-

face. It contains a three-dimensional displacement vector for each component of the load 3 vector $\bar{\mathbf{p}}_k$ at each of the N vertices .

$$\mathbf{u} = \sum_{k=1}^{N} \mathbf{\Xi}_k \bar{\mathbf{p}}_k \tag{1}$$

There is a crucial difference between the Green's functions estimated from measurements and those derived analytically by approximating the Green-Lagrange tensor. A displacement field of a discrete Green's functions model derived from measurements is a linear approximation of the *real, non-linear, large scale* deformation of the object from the undeformed to deformed state. The approximation can therefore be considered a secant approximation to the actual path of a point on the surface. Analytic derivations of discrete Green's function matrices will give a tangential approximation from the undeformed state instead.

4 Articulation Model

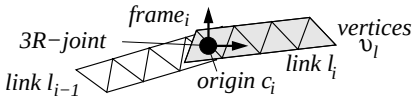


Figure 3: Articulation Model

Suppose, for the moment, that the kinematic structure of the object is a chain of L links; the extension to tree structures is straightforward (see Figure 3 for definitions). The coordinates of the vertices of each link l in the reference (world) frame are given by Υ_l which is a $3N$ vector. Vertices not belonging to the link are simply zero in Υ_l, while vertices at the seams between links are blended linearly. The coordinates of all vertices due to the articulation of the joints is given by

$$\Upsilon = \sum_{l=1}^{L} \Upsilon_l. \tag{2}$$

The articulated model of Equation 2 may be combined with the linear deformation model of Equation 1. Denoting the additional displacement due to deformation at vertex k by $\mathbf{\Delta}_k \bar{\mathbf{p}}_k$, the articulated deformable model is Equation 3.

$$\upsilon_{all} = \sum_{l=1}^{L} \Upsilon_l + \sum_{k=1}^{N} \mathbf{\Delta}_k \bar{\mathbf{p}}_k \tag{3}$$

The linear deformation model is capable of capturing translational joints exactly since translational joints result in straight-line displacements of the surface of a link. Therefore, we model only rotational joints by articulation. Each of the rotational joint is a *3R* joint having three degrees of rotational freedom. The homogeneous transformation matrix ${}^{i-1}_{i}\mathrm{E}$ between the link frame i and its parent frame $i-1$ can then be parameterized by three variable rotational angles $(\alpha_i, \beta_i, \gamma_i)$ and a fixed translation ${}^{i-1}c_i$ corresponding to the joint location. Picking a *X-Y-Z* fixed angle parameterization of the rotational submatrix R yields $R(\alpha_i, \beta_i, \gamma_i) = R_Z(\alpha_i)R_Y(\beta_i)R_X(\gamma_i)$. The rotational axes are fixed with respect to the joint. A simple linear rotational spring model for each joint with a 1×3 resistance κ results in $\alpha_i = \sum_{k=1}^{K} \kappa_{i,k,\alpha}\, \bar{\mathbf{p}}_k$ (with β_i and γ_i correspondingly).

The major challenge in fitting the articulated deformable model of Equation 3 to a displacement field is to identify which vertex belongs to which link. This is the topic of the following section.

5 Segmentation of displacement field

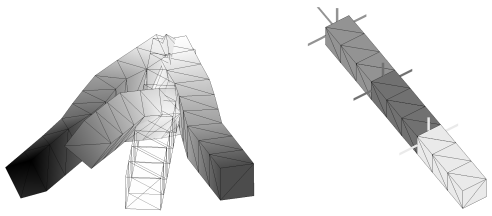
We would like to segment the displacement field of an object during deformation in a rigid motion of an articulated kinematic chain plus a linear displacement. We do not assume that the number, the extent or the relative position of the individual elements of the articulated model are known beforehand. Instead, we would like to infer this structure directly from multiple observations of the deformation behavior of the solid. We segment the displacement field into multiple clusters based on the rigid part of the motion of individual surface elements. The discrete Green's functions matrix described in Section 3 represents a displacement field in each reduced block column $\mathbf{\Xi}_k$. In particular, the x-column, the y-column and the z-column represent the displacement field due to a unit load in the x, y and z direction, respectively. Instead, the displacement field may also be observed directly by tracking the surface of an object from undeformed to deformed state. The tracker would have to produce a dense linear displacement field in order to be compatible with our approach.

Clustering the displacement field of the surface of an articulated deformable object is challenging for numerous reasons. A displacement field of an articulated deformable object defines a non-rigid motion in three dimensions. This non-rigid motion is the sum of a linear displacement field and of a rigid motion in our model. An inherent difficulty in clustering rigid motions is the lack of a consistent error metric in $SE(3)$. Instead, we choose an error metric which does not compare motions directly but compares the effect of motions on different areas of the surface. This is detailed in Section 5.1.

Beside the choice of a suitable error metric, the other major task in clustering is to choose an approach which

can work with the distribution of errors in the observations. The popular approach of using *k-means* for clustering tasks is neither robust in the face of large-scale deviations from the mean (distributions with heavy tails) nor can it easily handle an unknown number of clusters k (see e.g. Kaufman and Rousseeuw [12]). A more versatile set of segmentation algorithms are hierarchical clustering methods which build a hierarchy of segmentations. The hierarchy encodes a clustering result for all possible values of k. We employ an agglomerative hierarchical method because of its robustness [12]. Agglomerative methods start with each data element in a separate cluster, then merging clusters until the whole data set is in one cluster. We discuss the method of our choice in Section 5.2.

In the following sections, we use a simple "blocks world" example to clarify the steps of our method. The example consists of three displacement fields of a three-link articulated rod. Vertices at the seams are linearly blended, i.e., the displacement fields correspond to an articulated body with deformation at the seams. The displacements are shown in Figure 4(a).



(a) Articulated motion (b) Clustering result

Figure 4: Clustering the motion of an 3-Link articulated chain with linear blending between links

5.1 Locally-Rigid Motion Estimation

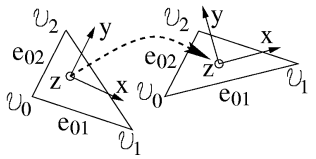


Figure 5: Locally rigid motion

The motion of a triangulated surface is the set of motions of each individual triangle. We model the motion of an individual triangle as a rigid motion plus some additional displacement of its vertices in order to accommodate deformations. We estimate the rigid motion by attaching a coordinate frame to the centroid of the triangle (see Figure 5). The coordinate axes of the frame are then

$$\begin{aligned} \hat{z} &= \frac{e_{01} \times e_{02}}{|e_{01} \times e_{02}|_2} \ , \\ \hat{x} &= \frac{e_{01}}{|e_{01}|_2} \ , \text{ and} \\ \hat{y} &= \hat{z} \times \hat{x} \ . \end{aligned}$$

The spatial velocity of the frame between deformed state $t1$ and the undeformed state $t0$ of the triangle is ${}^0_0\nu = {}^0_{t0}\dot{\mathrm{E}}\,{}^{t0}_0\mathrm{E}$. The transformation derivative from undeformed $t0$ to deformed state $t1$ is ${}^0_0\dot{\mathrm{E}} = {}^0_{t1}\mathrm{E}\,{}^{t0}_0\mathrm{E}$. Therefore, the spatial velocity of a triangle is Equation 4. We like to apply the spatial velocity to vertices of other triangles, i.e., it is advantageous to keep the velocity as a relative transform and not to represent them as a twist.

$${}^0_0\nu = {}^0_0\dot{\mathrm{E}}\,{}^0_{t0}\mathrm{E}\,{}^{t0}_0\mathrm{E} = {}^0_{t1}\mathrm{E}\,{}^{t0}_0\mathrm{E} \tag{4}$$

Having estimated the rigid spatial velocity of each triangle with Equation 4, we need to select a distance metric to compare the velocities of two triangles A and B. The estimated rigid spatial velocity of each triangle accounts for the vertex displacements during deformation up to some non-rigid motion. Non-rigid motion results in a change of shape of the triangle. We measure the amplitude of the non-rigid motion by the 2-norm of the difference between the rigid motion applied to a triangle and the actual displacement of the vertices of the triangle, i.e. $\left|{}^0_0\nu\ \upsilon_{t0} - \upsilon^A_{t1}\right|_2$. This amplitude is a measure of how well a rigid motion accounts for the actual displacements of a triangular element. We can compare the effect of different spatial velocities since we express all velocities in the reference coordinate frame 0. This non-rigid motion residual forms the basis of our error metric. We employ it to calculate the residual in both directions: the effect of the velocity of triangle B on A and vice versa. The resulting commutative distance metric between two velocities ${}^0_0\nu^A$ and ${}^0_0\nu^B$ associated with triangles A and B is Equation 5.

$$\begin{aligned} \varepsilon(A,B) = \ & \left|{}^0_0\nu^B\upsilon^A_{t0} - \upsilon^A_{t1}\right|_2 - \left|{}^0_0\nu^A\upsilon^A_{t0} - \upsilon^A_{t1}\right|_2 \\ & - \left|{}^0_0\nu^B\upsilon^B_{t0} - \upsilon^B_{t1}\right|_2 + \left|{}^0_0\nu^A\upsilon^B_{t0} - \upsilon^B_{t1}\right|_2 \end{aligned} \tag{5}$$

5.2 Robust Clustering

Our choice of clustering algorithm starts by calculating a triangular matrix of distances between elements. Each element corresponding to a triangle forms its own cluster. First, the two closest elements A and B are merged. The task then is to decide the distance of the combined cluster AB from other single element clusters C. We used the average distance which performed

satisfactorily. The average distance is $\varepsilon(AB, C) = \frac{1}{|AB||C|} \sum_{i \in AB} \sum_{j \in C} \varepsilon(i, j)$. The method continues by merging clusters in a greedy fashion until all elements form a single cluster. The agglomerative tree produced by this process has a height according to the distance between clusters as they were merged. Selecting a number of cluster is achieved by traversing down the tree in a breadth first manner until the required number of clusters have been encountered.

The result of the above clustering process is a grouping of triangles for which the spatial velocity is close in a single rigid motion. However, a single motion of an articulated body will typically not reveal all links since not all degrees of freedom are executed. Additionally, if the displacement field contains noise due to the measuring process or deviations from our motion model, a single pose will not identify links reliably. Several displacement fields from different poses need to be combined. We combine clustering results in a split-and-merge approach. First, clustering results for single poses are combined by a set union of their seams preserving all seams of every pose. This combination typically results in many more clusters than each individual pose by itself. The following merging process reduces the clusters again.

We merge clusters across displacement fields by a hierarchical clustering process again but now with the previously split clusters and the motion of their centroid as a starting point. The distance between clusters is the distance between centroids with the error metric of Equation 5. We employ a triangular distance matrix of the *split* cluster which is the weighted sum of the centroid distances in each pose. This distance may be zero if the split clusters are part of the same cluster in an individual pose. The weight is the inverse of the maximum distance of a velocity from the zero velocity in a pose. This weight will prevent small motions in one pose from being swamped by large motions in another pose. The distance metric between clusters CA and CB is now based on centroid A and B, respectively. The distance is calculated by $\varepsilon(CA, CB) = \frac{1}{|CA||CB|} \frac{\varepsilon(A,B)}{|CA|+|CB|}$. Figure 6(a) shows the split clusters with the three displacement fields of one block column of the discrete Green's functions matrix as input. Figure 6(b) shows the result of merging the split clusters.

6 Motion Estimation of an Articulated Chain

Estimating the motion of an articulated chain given known chain element encompasses various sub-tasks. First, the motion of each element in the chain has to be estimated. Given the motion of individual elements the relationship between them needs to be fit to a joint. This step includes the determination of joint positions and joint types. In our situation we also have to determine which chain elements to link with a joint. Once the motion of each link is known, we employ the method of 0'Brien et al. [16] in order to identify each joint location and the tree of links. This method has produced satisfactory results in our examples. If finding the tree is problematic, the neighborhood information between elements could be employed to prune the tree first. Additionally, if an actual kinematic structure is to be identified (see Section7.2), joint locations which are outside of the solid could be pruned since they are not physical.

	${}^0c_l$	Joint 1	Joint 2	Joint 3
Actual	x	0	12	24
Location	y	0	0	0
	z	0	0	0
Estimated	x	-0.0693	11.9739	24.7461
Location:	y	-0.0139	0.0660	0.0602
LSQ	z	-0.1162	0.1641	-1.0921
Estimated	x	0.0000	12.0000	24.0000
Location:	y	0.0000	0.0000	0.0000
Robust	z	0.0000	0.0000	0.0000

Table 1: Rod Example: Comparison of actual and estimated joint Location

Below, we compare two methods for estimating the motion of a link or cluster. One method is a least squares fitting approach of all vertices of a link to a rigid motion, the other method is a robust method which selects the motion centroid of each link with respect to our error metric in Equation 5. The least squares approach registers the vertices relative to each other in a two-step algorithm ignoring deformation by first aligning the geometric center of a link and then fitting a rotation. We fit the rotation with a method (analyzed by Kanatani [11]) which is optimal if the noise has a Gaussian distribution of zero mean and is isotropic, identical and independent. This condition is violated here since the noise in our case includes systematic deformation around the contact point.

The synthetic example of the deforming rod in Figure 4(a) gives a clear indication of the importance of robust fitting. Our distance metric identifies correctly non-deformed triangles as centroids of the motion for each link. This results in perfect estimation of the joint location listed in Table 1 and perfect identification of the rotational angles of each joint in the three poses. For comparison the least squares examples produces significant error due to the blended vertex locations at the seams of the links. Robust fitting is clearly necessary but the case is slightly more complicated in the real world examples which we discuss in Section 7.

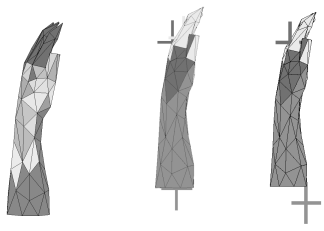
7 Applications

7.1 Pseudo Articulation for Large Scale Deformation

Deformable object modeling is the primary application of our method where the articulated deformation model of Equation 3 is purely a modeling tool. The goal here is to extend the range of applicability of linear deformation models. We find the partioning of the discrete Green's functions matrix defined in Equation 6.

$$\forall k \left(\bar{\mathbf{\Xi}}_k \bar{\hat{\mathbf{p}}}_k = \sum_{l=1}^{L} \Upsilon(k)_l + \bar{\mathbf{\Delta}}_k \bar{\hat{\mathbf{p}}}_k \right) \tag{6}$$

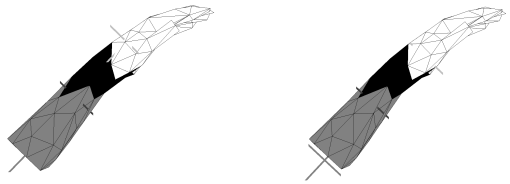
In fitting the articulated deformation model, the choice of an unit traction $\hat{\mathbf{p}}$ to multiply the Green's function matrix is arbitrary and any small traction will work. However, employing tractions matching the ones exerted by the robotic probe during estimation of the Green's function ensures the best match to the observed deformation behavior. We fit a separate articulated model $\sum_{l=1}^{L} \Upsilon(k)_l$ to each individual block column k of the discrete Green's functions matrix. This pseudo articulation approach provides a high number of degrees of freedom for capturing the large scale deformation behavior. It does not represent some kind of internal skeleton of the solid.



(a) Split (b) Robust (c) Least squares

Figure 6: Clustering result and joint estimation. A cross marks a joint location. Notice the poor location of the joint for the light grey cluster in Figure 6(c).

We demonstrate the application of our method to the discrete Green's functions model of the medical training phantom. The clustering result for the block column $k = 8$ is shown in Figure 6. The small blue cluster between the thumb and the back of the hand is around the location of vertex 8. The clustering identifies this region as one with similar motion. The other two clusters are the lower arm and the upper hand corresponding to regions with common global motion during the probing. Again, the least squares solution to motion estimation produces unsatisfactory results with one joint location far from the object (see Figure 6(c)). A simulation based on the solution of Equation 6 is shown in Figure 7(g) and Figure 7(e). Figure 7(g) and Figure 7(e) compare the effect of employing the robust and least squares joint estimation. All the simulations of the articulated deformable models contain a strain hardening model of the form $\tilde{p}_k = p_{max}(1-((p_{max}-1)/p_{max})^{p_k})$, a model which we observed but have not explicitly measured. Strain hardening limits the linear displacements of the deformation model as large forces are applied. It reduces geometric distortions even further than the articulated deformation model by itself. This ability to limit the linear deformation while still providing increased deformation for large forces is another feature of our method.



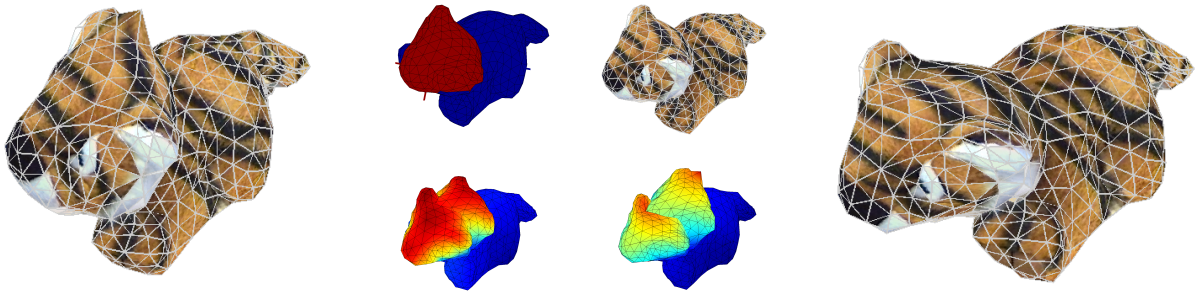
(a) Robust (b) Least squares

Figure 8: Comparison: Robust and LSQ joint estimation

7.2 Recovering Unknown Kinematic Structure

The above application of extending the range of applicability of a deformable model does not aim to find a consistent articulation model for all displacement fields of an object. However, finding the number of links and the location of the joints of an articulated structure is of interest in itself in motion capture applications. Our method solves this task by estimating a segmentation consistent over all displacements. A consistent segmentation is achieved by adding another hierarchical clustering step. If we start with a Green's functions matrix, we add a combined estimation for all the block-columns $k = 1 \ldots M$; if we start with dense motion capture data, this means simply combining a large enough number of poses in order to exercise all degrees of freedom of the articulated structure. The lower number of degrees of freedom compared to the pseudo articulation approach will handle noisy data better. It will also not fit a separate cluster to any local deformation present in just a single pose or a single block column of discrete Green's function matrix due to contact with a probe.

The medical wrist phantom groups into three clusters according to the motion of its rubber skin (see Figure 8). In this example, the least squares method performs similar to the robust estimation method because of the large

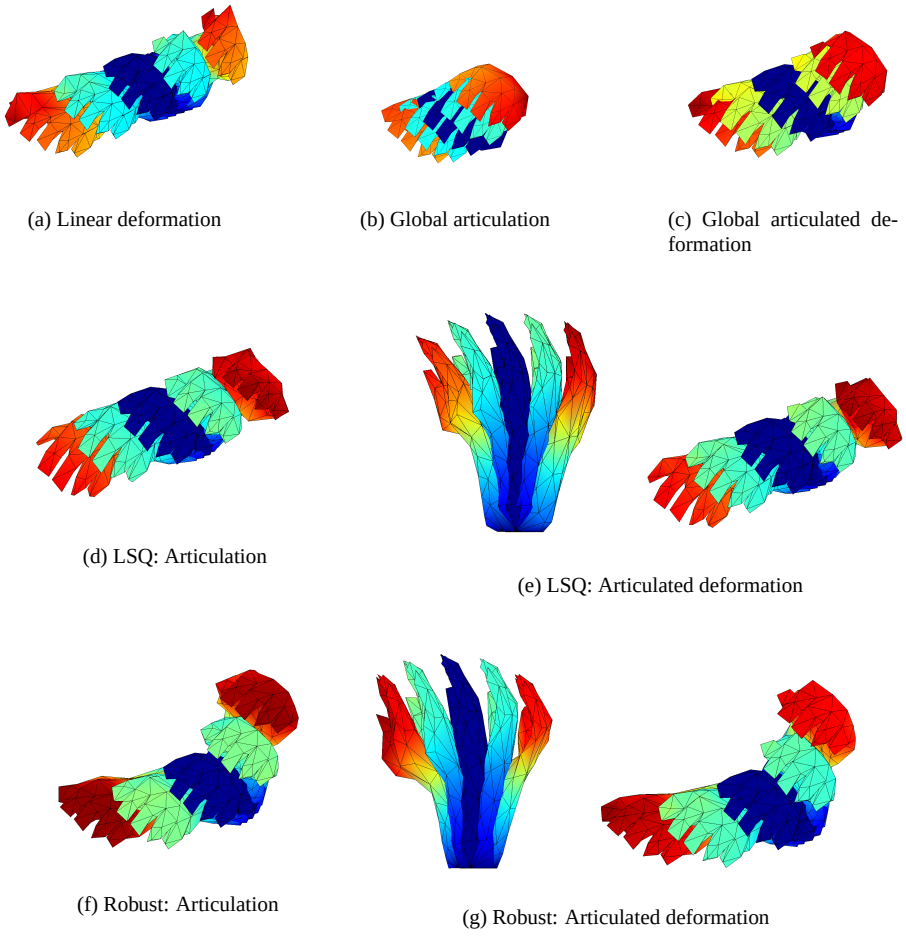


(a) Pulled

(b) Clustering result, undeformed shape, articulation only, articulated deformation (left to right, top to bottom)

(c) Pushed

Figure 2: A plush tiger being pulled and pushed behind its ear



(a) Linear deformation

(b) Global articulation

(c) Global articulated deformation

(d) LSQ: Articulation

(e) LSQ: Articulated deformation

(f) Robust: Articulation

(g) Robust: Articulated deformation

Figure 7: Model comparison: Linear deformation model vs. global and local articulated deformation model. The robust joint estimation method captures also the twisting motion of the arm due to the off-center load. The load is applied at the back of the hand on the side of the lower thumb (see Figure 7(a)).

number of displacement fields employed when estimating the joint locations. Figure 7 compares between simulation with pseudo articulation and with the global articulation structure. The simulation of the articulated model indicates how much of the displacement field is accounted for by the articulation. Larger articulation is observed with the pseudo-articulation approach since it fits the observed behavior closer. This is to be expected because of the larger number of degrees of freedom in the model compared to the global articulation. Figure 2 shows an example of applying the global articulation method to a plush toy without any internal structure. Due to seams in the textile fur the head of the the tiger apparently hinges around its neck. The global articulation model captures this behavior nicely.

8 Conclusion

In this paper, we present a method which is able to segment displacement fields into areas with similar motion. We show how to apply this clustering method to fit an articulated deformable model to real-world objects. Our method has applications in the acquisition and simulation of deformable objects and in motion capture. Pseudo-articulation extends the range and versatility of linear deformation models maintaining their low computational burden but providing realistic large deformation. In motion capture, the method allows the identification of the links of the kinematic chain based only on displacement data. This means that the motion of a subject or object can be captured without specifying any kind of skeleton. The only tuning parameter required in our method is the number of expected links.

Acknowledgements

We are grateful for access to the University of British Columbia Active Measurement Facility and like to thank John E. Lloyd.

References

[1] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Computer Vision and Pattern Recognition*. IEEE, 1998.

[2] M. Bro-Nielsen and S. Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. In J. Rossignac and F.X. Sillion, editors, *Eurographics*, Computer Graphics Forum, 15(3), pages 57–66, Poitiers, France, 1996.

[3] J.J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Pub., Reading, Ma., USA, 2^{nd} edition, 1989.

[4] G. Debunne, M. Desbrun, M.-P. Cani, and A.H. Barr. Dynamic real-time deformations using space & time adaptive sampling. In *Computer Graphics, Annual Conference Series*, pages 31–36, Los Angles, USA, Aug 2001. ACM SIGGRAPH.

[5] S.F.F. Gibson and B. Mirtich. A survey of deformable modeling in computer graphics. Technical Report TR-97-19, Mitsubishi Electric Research Laboratory, Cambridge, MA, USA, Nov. 1997.

[6] E. Grinspun, P. Krysl, and P. Schröder. CHARMS: A simple framework for adaptive simulation. In *ACM Transactions on Graphics*, volume 21, pages 281–290, San Antonio, USA, Jul 2002. ACM SIGGRAPH.

[7] J.M. Hollerbach and C.W. Wampler. The calibration index and taxonomy for robot kinematic calibration methods. *IJRR*, 15(6):573–591, 1996.

[8] D.L. James and D.K. Pai. ArtDefo accurate real time deformable objects. In *Computer Graphics, Annual Conference Series*, pages 65–72, Los Angles, USA, Aug 1999. ACM SIGGRAPH.

[9] D.L. James and D.K. Pai. Real time simulation of multizone elastokinematic models. In *International Conference on Robotics and Automation*, pages 927–932, Washington, D.C., USA, May 2002.

[10] I.A. Kakadiaris and D. Metaxas. Three-dimensional human body model acquisition from multiple views. *International Journal of Computer Vision*, 30(3):191–218, 1998.

[11] K. Kanatani. Analysis of 3-d rotation fitting. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 16(5):543–549, 1994.

[12] L. Kaufman and P.J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, New York, 1990.

[13] J. Lang, D.K. Pai, and R.J. Woodham. Acquisition of elastic models for interactive simulation. *IJRR*, 21(8), 2002.

[14] N. Magnenat-Thalmann, R. Laperrière, and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Graphics Interface*, pages 26–33, Edmonton, Canada, June 1988.

[15] M. Müller, L. McMillan, J. Dorsey, R. Jagnow, and B. Cutler. Stable real-time deformations. In *Symposium on Computer Animation*, San Antonio, Texas, USA, July 2002. ACM SIGGRAPH.

[16] J.F. O'Brien, B.E. Bodenheimer, G.J. Brostow, and J.K. Hodgins. Automatic joint parameter estimation from magnetic motion capture data. In *Proceedings of Graphics Interface 2000*, pages 53–60, Montreal, Quebec, Canada, May 2000.

[17] D.K. Pai, K. van den Doel, D.L. James, J. Lang, J.E. Lloyd, J.L. Richmond, and S.H. Yau. Scanning physical interaction behavior of 3D objects. In *Computer Graphics, Annual Conference Series*, pages 87–96, Los Angles, USA, Aug 2001. ACM SIGGRAPH.

[18] G. Picinbono, H. Delingette, and N. Ayache. Non-linear and anisotropic elastic soft tissue models for medical simulation. In *International Conference on Robotics and Automation*, pages 1371–1376, Seoul, South Korea, May 2001.

[19] R. Plänkers and P. Fua. Articulated soft objects for video-based body modeling. In *International Conference on Computer Vision*, volume 1, pages 394–401, Vancouver, Canada, Jul 2001. IEEE.

[20] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. In *Computer Graphics, Annual Conference Series*, pages 205–214, Anaheim, Ca., USA, Jul 1987. ACM SIGGRAPH.

[21] D. Terzopoulos and A. Witkin. Physically-based models with rigid and deformable components. In *Graphics Interface*, pages 146–154, Edmonton, Canada, June 1988.

[22] X. Wu, M.S. Downes, T. Gotekin, and F. Tendick. Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. In A. Chalmers and T.-M. Rhyne, editors, *Eurographics*, Computer Graphics Forum, 20(3), pages C349–C358, Manchester, Uk, 2001.

[23] Y. Zhuang and J. Canny. Haptic interactions with global deformations. In *International Conference on Robotics and Automation*, pages 2428–2433, San Francisco, USA, April 2000.