Group-valued regularization framework for motion segmentation of dynamic non-rigid shapes

Guy Rosman[1], Michael M. Bronstein[2], Alexander M. Bronstein[3],
Alon Wolf[4,5], and Ron Kimmel[1,5⋆]

[1] Dept. of Computer Science, Technion – Israel Institute of Technology
Haifa, 32000, Israel,
rosman,ron@cs.technion.ac.il
[2] Institute of Computational Science, Faculty of Informatics
Universit della Svizzera italiana, CH - 6904 Lugano, Switzerland
michael.bronstein@usi.ch
[3] School of Electrical Engineering, Faculty of Engineering
Tel Aviv University, Ramat Aviv 69978, Israel
bron@eng.tau.ac.il
[4] Dept. of Mechanical Engineering, Technion – Israel Institute of Technology
Haifa, 32000, Israel
alonw@tx.technion.ac.il
[5] The Goldstein UAV and Satellite Center, Technion – Israel Institute of Technology
Haifa, 32000, Israel,
alonw@tx.technion.ac.il,ron@cs.technion.ac.il

Abstract. Understanding of articulated shape motion plays an important role in many applications in the mechanical engineering, movie industry, graphics, and vision communities. In this paper, we study motion-based segmentation of articulated 3D shapes into rigid parts. We pose the problem as finding a group-valued map between the shapes describing the motion, forcing it to favor piecewise rigid motions. Our computation follows the spirit of the Ambrosio-Tortorelli scheme for Mumford-Shah segmentation, with a diffusion component suited for the group nature of the motion model. Experimental results demonstrate the effectiveness of the proposed method in non-rigid motion segmentation.

Key words: Motion Segmentation, Lie-groups, Surface Diffusion, Ambrosio-Tortorelli

1 Introduction

The analysis of articulated motion in three-dimensional space is a key problem in biomechanics [1], mechanical engineering, computer vision [28, 31, 20, 37], and computer graphics [24, 29, 41, 27, 44, 6, 43]. Specific problems of deformation analysis [4] and motion segmentation [5, 13] try to infer the articulated motion of an object, given several instances of the analyzed object in different poses. The desired outcome is the

⋆ This research was supported in part by The Israel Science Foundation (ISF) grant number 623/08, and by the Goldstein UAV and Satellite Center.

segmentation of the object into rigid parts and motion estimation between the corresponding parts.

Most motion analysis techniques either assume a known prior on the articulated structure of the inspected object (e.g., in the form of a skeleton), or decide on the structure in an *ad hoc* manner, not based on the kinematic model commonly assumed for near-rigid objects [1, 4]. Since in many cases such *a priori* assumptions about the data are only approximate, they can lead to errors in the segmentation and motion estimation.

Another common assumption, especially in graphics applications, is that of known correspondence. In computer graphics, the problem is usually referred to as *dynamic mesh segmentation*.

The above assumptions are often too limiting in real-world applications. Instead, we would like to apply the intuition often used when studying real-life near-rigid objects, about the existence of an average rotational motion existing for each body part, but do so without attempting to detect the articulated parts in advance, and without assuming the existence of a clear partition of the surface. In other words, we would like to obtain a "soft" segmentation of the surface, without knowing the number or location of regions in advance, without analyzing the surface features, or having additional priors on the various object parts. In addition, we expect a complete formulation of motion segmentation to incorporate an implicit handling of the correspondence problem, given a reasonable initialization.

Main contribution. In this paper we try to remedy the shortcoming of existing approaches to articulated motion estimation by combining the two tasks of motion estimation and segmentation into a single functional. Unlike existing methods, we propose a principled variational approach, attempting to find a rigid transformation at each surface point, between the instance surfaces, such that the overall transformation is described by a relatively sparse set of such transformations, each matching a rigid part of the object. The functional we propose regularizes the motion between the surfaces, and is guided by the fact that the parameters of the motion transformations (i) should describe the motion at each point with sufficient accuracy; (ii) should vary smoothly within (unknown) rigid parts; (iii) can vary abruptly between rigid parts.

We see the main contribution of this paper in the following three aspects: First, we propose an axiomatic variational framework for articulated motion segmentation. While focusing on the segmentation problem in this paper, our framework is more general and the proposed functionals can be easily incorporated into other applications such as motion estimation, tracking, and surface denoising. Second, we demonstrate that the articulated motion segmentation problem can be solved within the proposed framework by adapting standard tools from variational segmentation to the geometry of the case, and obtain results competitive with domain-specific state-of-the-art tools. Third, we suggest a spatially-coherent algorithm for spatial visualization of group valued data on manifolds, which draws from the same variational principles.

Relation to prior work. The scheme we propose involves diffusing the transformations along the surface, in the spirit of the Ambrosio-Tortorelli scheme [2] for Mumford-Shah segmentation [33]. The diffusion component of our scheme is a diffusion process of Lie group elements, which has recently attracted significant attention in other applications [16, 39, 18]. In diffusing transformations on the surface, our work

is similar to that of Litke et al. [30]. We do not, however, make any assumption on the surface topology; to that end, the proposed method diffuses transformations along the surface, rather than representing the surface in an evenly sampled 2D parametrization plane. When dealing with real-life deformable objects that seldom admit regular global parametrization, such an assumption could be too restrictive.

The idea of combining soft segmentation and motion estimation has been attempted before in the case of optical flow computation (see, e.g., [3]). In optical flows, however, the motion field is merely expected to be piecewise smooth. For truly articulated objects one would expect piecewise-constant flow fields, when expressed in the correct parametrization.

Finally, our work is related, and complementary, to the topic of geometry-based mesh segmentation. While several works from this field can be combined with motion based segmentation techniques, this is not the focus of this work. We point the reader to [7, 38, 12, 26], and references therein, for additional examples of mesh segmentation algorithms.

2 Problem formulation

Articulation model. Let us be given a three-dimensional shape, which we model as a two-dimensional manifold X. In the following, we will denote by $\mathbf{x} : X \to \mathbb{R}^3$ the embedding of X into $\mathbb{R}^3$, and use synonymously the notation x and $\mathbf{x}$ referring to a point on the manifold and its Euclidean embedding coordinates, respectively.

We further assume that the shape X is *articulated*, i.e., can be decomposed into *rigid parts* $S_1, \ldots, S_p$ and nonrigid *joints* $J_1, \ldots, J_q$, such that $X = \bigcup_{i=1}^{p} S_i \cup \bigcup_{k=1}^{q} J_k$. An *articulation* $Y = \mathbf{A}X$ is obtained by applying rigid motions $\mathbf{T}_i \in \mathrm{Iso}(\mathbb{R}^3)$ (rotations and translations) to the rigid parts, and non-rigid deformations $\mathbf{Q}_k$ to the joints, such that $\mathbf{A}X = \bigcup_{i=1}^{p} \mathbf{T}_i S_i \cup \bigcup_{k=1}^{q} \mathbf{Q}_k J_k$.

Motion segmentation. The problem of *motion-based segmentation*, in its simplest setting can be described as follows: given two articulations of the shape, X and Y, extract its rigid parts. Extension to the case of multiple shape poses is straightforward. We therefore consider in the following only a pair of shapes for the sake of simplicity and without loss of generality.

Assuming that the correspondence between the points on two shapes X and Y is known, given two corresponding points $x \in X$ and $y(x) \in Y$, we can find a motion $g \in \mathcal{G}$ such that $g\mathbf{x} = \mathbf{y}$, where $\mathcal{G}$ is some representation of coordinate transformations in $\mathbb{R}^3$, and with some abuse of notation, $g\mathbf{x} \in \mathbb{R}^3$ denotes the action of g on the coordinates of the point x. We can represent the transformation at each point as a field $g : X \to \mathcal{G}$.

Since the articulated parts of the shape move rigidly, if we choose an appropriate motion representation (as detailed below), two points $x, x' \in S_i$ will undergo the same transformation, from which it follows that $g(x)|_{x \in S_i} = \mathrm{const}$. One possibility is to adopt a constrained minimization approach, forcing $g(X) = Y$, where $g(X)$ is a short notation for the set $g(x)\mathbf{x}(x)$ for all $x \in X$. A more convenient possibility is to take an unconstrained formulation,

$$\min_{g:X\to\mathcal{G}} \lambda E_{\mathrm{DATA}}(g) + \rho(g), \tag{1}$$

where ρ denotes some regularization term which is small if g is piecewise constant. $E_{\mathrm{DATA}}(g)$ is our fitting term which penalizes the discrepancy between the transformed template surface $g(X)$ and Y,

$$E_{\mathrm{DATA}}(g) = \int_X \|g(x)\mathbf{x} - \mathbf{y}(x)\|^2 da, \tag{2}$$

where $\mathbf{y}(x)$ denotes the coordinate of the point on Y corresponding to x, $g(x)$ is the transformation at x, and da is a measure on X.

Such a formulation allows us to recover the articulated parts by clustering g into regions of equal value. This formulation (presented in Section 4.4) bears much resemblance to *total variation* regularization common in signal and image processing [35].

One tacit assumption in this problem is that the correspondence between X and Y is known, which is usually not true. We will mention this issue in Section 4.1. Second, it is crucial to observe that the effectiveness of (1) relies on some correct representation $\mathcal{G}$ of the motion. The simplest representation is the *linear motion*, assuming $\mathcal{G} = \mathbb{R}^3$ such that $g\mathbf{x} = \mathbf{x} + \mathbf{t} = \mathbf{y}$ for some $\mathbf{t} \in \mathbb{R}^3$. However, such a simplistic model fails to capture the piecewise constancy of the motion field. It is thus clear that we need a representation of motion that is *redundant* (i.e., an *over-parametrization* using more than three degrees of freedom to describe a transformation) and in which motions of points that move rigidly are described by the same element of $\mathcal{G}$.

One parametrization often used in computer vision and robotics [42, 32, 27, 18] is the representation of rigid motions by the Lie group $SE(3)$ and the corresponding Lie algebra $se(3)$, respectively. Lie groups are topological groups with a smooth manifold structure such that the group action $\mathcal{G} \times \mathcal{G} \mapsto \mathcal{G}$ and the group inverse are differentiable maps. Each Lie group has a Lie algebra associated with it, which can be mapped via the *exponential map* onto the tangent space at the identity operator. The Lie algebra $se(3)$ allows us to represent rotations and rigid motions locally in a consistent linear space using the logarithm and exponential maps at each point. We refer the reader to standard literature on the subject (e.g., [19]) for more information.

The Lie algebra of $SE(3)$ is the group of 4×4 matrices of the form

$$se(3) = \begin{pmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}, \mathbf{A} \in so(3), \mathbf{t} \in \mathbb{R}^3, \tag{3}$$

where $so(3)$ is the set of 3×3 skew-symmetric matrices.

Under the assumption of $\mathcal{G} = SE(3)$, we have our desired property that the transformation of points undergoing a rigid motion are described by the same group element. Solving problem (1) thus requires a regularization term ρ that favors piecewise constancy of group elements on the shape. We discuss such a regularization in Section 3. We also note that due to the non-Euclidean structure of the group, special care should be taken when parameterizing such a representation [32, 18, 39, 27], as discussed in Section 4.2.

3 Diffusion-based regularization

Thinking of the Lie group $\mathcal{G}$ as a Riemannian manifold, we look for a functional defined on maps between manifolds of the form $g : X \to \mathcal{G}$. Such maps can be regularized by

the well-known *Dirichlet energy* [25],

$$\rho_{\mathrm{DIR}}(g) = \frac{1}{2}\int_X \langle \nabla g, \nabla g \rangle_{g(x)} da, \tag{4}$$

where ∇g denotes the *intrinsic gradient* of g on X, $\langle \cdot, \cdot \rangle_{g(x)}$ is the Riemannian metric on $\mathcal{G}$ at a point $g(x)$, and da is the area element of X. The minimizer of the Dirichlet energy is called a *harmonic map*, and it is the solution of a diffusion equation. In the case where both X and $\mathcal{G}$ are Euclidean, ρ_{DIR} reduces to the standard Tikhonov regularization.

Ambrosio-Tortorelli scheme. Unfortunately, the Dirichlet energy does not favor piecewise-constancy of g, as is desired. We therefore adopt the Ambrosio-Tortorelli scheme [2] for Mumford-Shah regularization [33], in which the Dirichlet energy term is modulated by a diffusivity function $v : X \to [0, 1]$,

$$\rho_{\mathrm{AT}}(g) = \int_X \left(\frac{1}{2} v^2 \langle \nabla g, \nabla g \rangle_g + \epsilon \langle \nabla v, \nabla v \rangle + \frac{(1-v)^2}{4\epsilon} \right) da, \tag{5}$$

where ϵ is a small positive constant. This allows us to extend our outlook in several ways. The Mumford-Shah functional replaces the notion of a set of regions with closed simple boundary curves with general discontinuity sets. It furthermore generalizes our notion of constant value regions with that of favored smoothness inside the areas defined by these discontinuity curves. This is in order to handle objects which deviate from articulated motion, for example in flexible regions or joints.

Furthermore, the generalized Ambrosio-Tortorelli scheme allows us to explicitly reason about the places in the flow where the nonlinear nature of the data manifold manifests itself. Suppose we have a solution (g^*, v^*) satisfying our piecewise-constancy assumptions of g, and a diffusivity function with 0 at region boundaries and 1 elsewhere. At such a solution, we expect two neighboring points which belong to different regions to have a very small diffusivity value v connecting them, effectively nullifying the interaction between far-away group elements which is dependent on the mapping used for the logarithm map at each point, and hence can be inaccurate [22, 32]. While such a solution (g^*, v^*) may not be a minimizer of the functional, it serves well to explain the intuition motivating the choice of the functional.

Diffusion of Lie group elements. In order to efficiently compute the Euler-Lagrange equation corresponding to the generalized Ambrosio-Tortorelli functional (5), we transform the neighborhood of each point into the corresponding Lie algebra elements before applying the diffusion operator. Using Lie algebra representation of differential operators for rigid motion has been used before in computer vision [39], numerical PDE computations [22], path planning and optimal control theory [32, 27].

The Euler-Lagrange equation for the generalized Dirichlet energy measuring the map between two manifolds is given as [25]

$$\Delta_X g^\alpha + \Gamma^\alpha_{\beta\gamma} \left\langle \nabla g^\beta, \nabla g^\gamma \right\rangle_{g(x)} = 0, \tag{6}$$

where $\alpha, \beta\, \gamma$ enumerate the local coordinates of our group manifold, $se(3)$, and we use Einstein's notation according to which corresponding indices are summed over. $\Gamma^\alpha_{\beta\gamma}$

are the *Christoffel symbols* of $SE(3)$, which express the Riemannian metric's local derivatives. We refer the reader to [15] for an introduction to Riemannian geometry. Finally, Δ_X denotes the Laplace-Beltrami operator on the surface X.

In order to avoid the computation of the Christoffel symbols, we transform the point and its neighbors using the logarithm map at that point in $SE(3)$. The diffusion operation is now affected only by the structure of the surface X. After applying the diffusion operator, we use the exponential map in order to return to the usual representation of the transformation.

4 Numerical considerations

We now describe the algorithm for articulated motion estimation based on the minimization of the functional

$$E(g, v) = \lambda E_{\mathrm{DATA}}(g) + \rho_{AT}(g, v), \tag{7}$$

where $E_{\mathrm{DATA}}(g)$ is the matching term defined by Equation 2, and $\rho_{AT}(g, v)$ is defined in Equation 5. The main steps of the algorithm are outlined as Algorithm 1. Throughout the algorithm we parameterize $g(x)$ based on the first surface, given as a triangulated mesh, with vertices $\{x_i\}_{i=1}^{N}$, and an element from $SE(3)$ defined at each vertex. The triangulation is used merely to obtain a more consistent numerical diffusion operator, and is not required otherwise. Special care is made in the choice of coordinates during the optimization as explained in Section 4.2.

4.1 Initial correspondence estimation

As in other motion segmentation algorithms, some initialization of the matching between the surfaces must be used. One approach [6] is to use nonrigid surface matching for initialization. Another possibility, in the case of high framerate sequences [44], is to exploit temporal consistency. While we focus on the functional itself, we present a possible initialization scheme which assumes a known sparse correspondence between the surfaces (simulating motion capture markers). We then interpolate this sparse set in order to initialize an *iterative closest point* (ICP) search [8], matching the patch around each point to the target mesh. In Figure 3, we use 30 matched points for initialization. This number of points is within the scope of current motion capture marker systems, or of algorithms for global nonrigid surface matching such as spectral methods [23, 34, 36], or the *generalized multidimensional scaling* (GMDS) algorithm [9].

We expect better initial registration, possibly using a smoothness assumption, to allow fewer markers to be used.

4.2 Diffusion of Lie group elements

Rewriting the optimization over the functional in Equation 7 in a fractional step approach [45], we update each function in a suitable representation.

Using the transformation described in Section 3, the update step with respect to the regularization now becomes [18]

$$g^{k+1/2} = \exp\left(-dt\frac{\delta\rho_{\mathrm{AT}}}{\delta\tilde{g}}\right)g^k, v^{k+1} = v^k - dt\frac{\delta\rho_{\mathrm{AT}}}{\delta v} \tag{8}$$

where $\exp(A) = I + A + A^2/2! + A^3/3! + \ldots$ denotes the matrix exponential, $\tilde{g}$ denotes the logarithm transform of g, and dt denotes the time step. $\frac{\delta\rho_{\mathrm{AT}}}{\delta\tilde{g}}$ denotes the variation of the regularization term $\rho_{AT}(g)$ w.r.t. the Lie-algebra local representation of the solution, describing the Euler-Lagrange descent direction. $g(x)$ and the neighboring transformations are parameterized by a basis for matrices in $se(3)$, after applying the logarithm map at $g(x)$. The descent directions are given by

$$\begin{aligned}\frac{\delta\rho_{\mathrm{AT}}}{\delta\tilde{g}_i} &= v^2\Delta_X(\tilde{g}_i) + v\left\langle\nabla v, \nabla\tilde{g}_i\right\rangle \\ \frac{\delta\rho_{\mathrm{AT}}}{\delta v} &= \left\langle\nabla g, \nabla g\right\rangle_{g(x)}v + 2\epsilon\Delta_X(v) + \frac{(v-1)}{2\epsilon},\end{aligned} \tag{9}$$

where $\tilde{g}_i$ denote the components of the logarithmic representation of g. The discretization we use for Δ_X is a cotangent one suggested by [14], which has been shown to be convergent for relatively smooth and well-parameterized surfaces. It is expressed as

$$\Delta_X(u) \approx \frac{3}{\mathcal{A}_i}\sum_{j\in\mathcal{N}_1(i)}\frac{\cot\alpha_{ij} + \cot\beta_{ij}}{2}\left[u_j - u_i\right], \tag{10}$$

for a given function u on the surface X, where $\mathcal{N}_1(i)$ denotes the mesh neighbors of point i, and α_{ij}, β_{ij} are the angles opposing the edge ij in its neighboring faces. $\mathcal{A}_i$ denotes the area of the 1-ring around i in the mesh. After a gradient descent step w.r.t. the diffusion term, we take a step w.r.t. the data term.

$$g^{k+1} = P_{SE(3)}\left(g^{k+1/2} - dt\frac{\delta E_{DATA}}{\delta g}\right), \tag{11}$$

where $P_{SE(3)}(\cdot)$ denotes a projection onto the group $SE(3)$ obtained by correcting the singular values of the rotation matrix. We compute the gradient w.r.t. a basis for small rotation and translation matrices comprised of the regular basis for translation and the skew-matrix approximation of small rotations. We then reproject the update onto the manifold. This keeps the inaccuracies associated with the projecting manifold-constrained data [32, 18] at a reasonable level.

Finally, we note that we may not know in advance the points $y(x)$ which match X in Y. The correspondence can be updated based on the current transformations in an efficient manner similarly to the ICP algorithm [8], using a KD-tree.

4.3 A patch-based data term

The data term we use works to fit the 3 output functions of the transformations defined on the surface. As in the case of the aperture problem in optical flow computation, it

is the regularization term that helps us obtain a complete view of the transformations field. However, as in optical flow computation [11], extending the surface matching to a small patch around each point gave us a more robust estimation of the transformations. The revised data term reads

$$E_{\mathrm{DATA}}(g) = \int_X \int_{\mathcal{N}(x)} \|g(x)\mathbf{x} - \mathbf{y}(x)\|^2 da \times da, \tag{12}$$

where $\mathcal{N}(x)$ denotes a small neighborhood around the point x.

Algorithm 1 Articulated Surface Segmentation and Matching

1: Given an initial correspondence.
2: **for** $k = 1, 2, \ldots,$ until convergence **do**
3: Update $g^{k+1/2}, v^{k+1}$ w.r.t. the diffusion term, according to Equation 8.
4: Obtain g^{k+1} according to the data term, using Equation 11.
5: Update $y^{k+1}(x)$, the current estimated correspondence of the deformed surface.
6: **end for**

4.4 Visualizing Lie group clustering on surfaces

Finally, we need to mention the approach taken to visualize the transformations as the latter belong to a six-dimensional non-Euclidean manifold. Motivated by the widespread use of vector quantization in such visualizations, we use a clustering algorithm with spatial regularization. Instead of minimizing the Max-Lloyd cost function, we minimize the function

$$E_{VIS}(g_i, R_i) = \sum_i \int_{R_i} \|g - g_i\|^2 da + \int_{\partial R_i} v^2(s) ds, \tag{13}$$

where ∂R_i denotes the set of boundaries between partition regions $\{R_i\}_{i=1}^N$, g_i are the group representatives for each region, and $v^2(s)$ denotes the diffusivity term along the region boundary. Several (about 50) initializations are performed, as is often customary in clustering, with the lowest cost hypothesis kept.

While this visualization algorithm coupled with a good initialization at each point can be considered as a segmentation algorithm in its own right, it is less general as it assumes a strict separation between the parts. We further note, however, that the diffusion process lowered the score obtained in Equation 13 in the experiments we conducted, indicating a consistency between the two algorithms in objects with well-defined rigid parts.

5 Results

We now demonstrate the results obtained by our method, in terms of the obtained transformation field and the diffusivity function. In Figure 1 we demonstrate matching between two human body poses taken from the TOSCA dataset [10]. As can be seen,

the diffusivity function hints at the location of boundaries between parts. In addition, we visualize the transformations obtained using the clustering algorithm described in subsection 4.4.

Figure 1 also demonstrates the results of comparing four poses of the same surface, this time with the patch-based data term described by (12). In our experiments the patch-based term gave a cleaner estimation of the motion as is observed in the diffusivity function.

We also demonstrate our algorithm on a horse taken from [40] as a set of 6 poses in Figure 2. In this figure we compare our results to those of [43], obtained on a similar set of poses with 10 frames. In Figure 3 we demonstrate our algorithm while initializing it from a set of 30 points where displacement is known. The relatively monotonous motion range available in the dynamic mesh sequence leads to a less complete, but still quite meaningful, segmentation of the horse, using an initialization which can be obtained by a feasible setup. We also note the relatively low number of poses required for segmentation – in both Figure 2 and Figure 3 we obtain good results despite the fact we use only a few poses, six and eight respectively. This contrasts with 10 poses used in [43], the results of which are shown in Figure 2 for comparison.

Finally, in Figure 3 we demonstrate convergence based on a sparse initialization, with 30 known correspondence points, arbitrarily placed using farthest point sampling [17, 21]. This demonstrates a possibility of initializing the algorithm using motion capture markers, coupled with a 3D reconstruction pipeline, for object part analysis. While the examples shown in this paper are synthetic, this experiment demonstrates the algorithm can be incorporated into a real-world system.

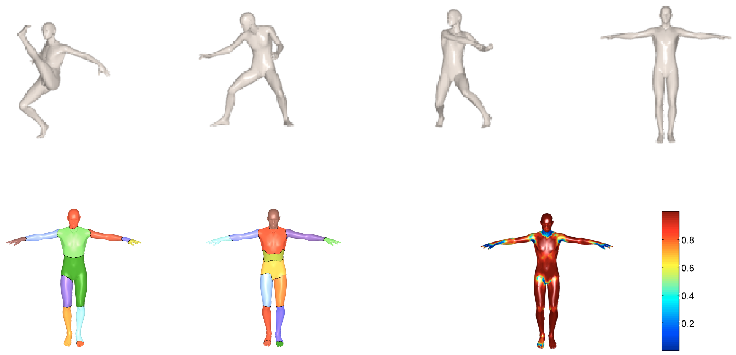

Fig. 1. Segmenting a human figure. Top row: the set of poses used. Bottom row, left-to-right: the transformations obtained from the two left most poses, the transformations obtained from all four poses using Equation 12 as a data term, and the Ambrosio-Tortorelli diffusivity function based on four poses.

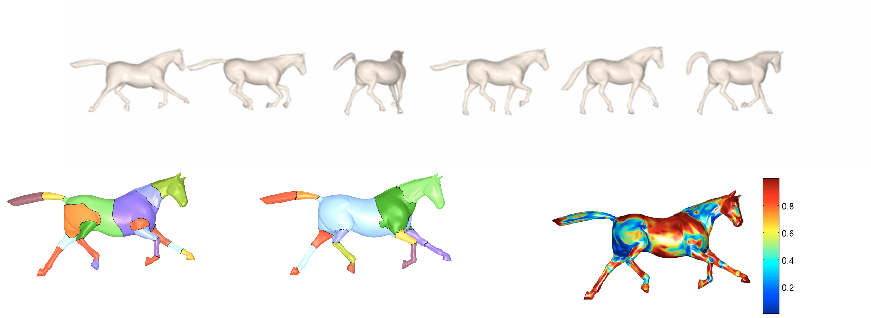

Fig. 2. Segmenting a horse dynamic surface motion based on six different poses. Top row: the poses used. Bottom row, left to right: a visualization of the transformations of the surface obtained by our method, and the segmentation results obtained by [43], and the diffusivity function v.

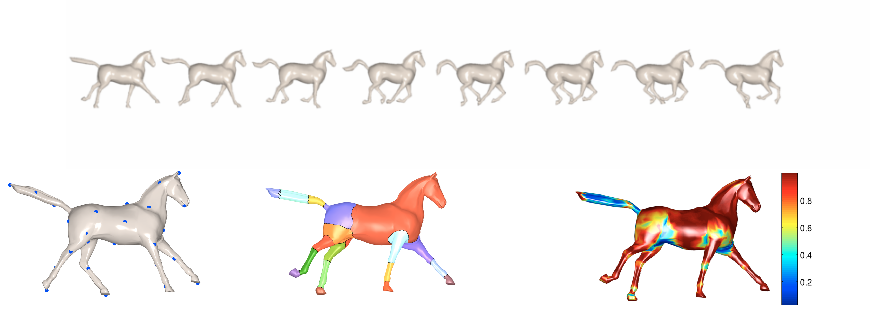

Fig. 3. Segmenting a horse dynamic surface motion with a given sparse initial correspondences. Top row: the eight random poses used. Bottom row, left to right: the set of points used for initializing the convergence, and a visualization of the transformations obtained, and the diffusivity function v.

6 Conclusion

In this paper we have presented a method for simultaneous segmentation and motion estimation in articulated objects, based on a variational formulation. Several results shown demonstrate the method's effectiveness, and merit its application to specific problems where it can be contrasted and combined with domain-specific algorithms for articulated object analysis. In future work we intend to adapt the proposed algorithm to real data from range scanners, and explore initialization methods as well as use the proposed framework in other applications such as articulated surfaces tracking and denoising.

References

1. E. J. Alexander, C. Bregler, and T. P. Andriacchi. Non-rigid modeling of body segments for improved skeletal motion estimation. *Computational Modeling Engineering Science*, 4:351–364, 2003.
2. L. Ambrosio and V. M. Tortorelli. Approximation of functional depending on jumps by elliptic functional via Γ-convergence. *Comm. on Pure and Appl. Math.*, 43(8):999–1036, 1990.
3. T. Amiaz and N. Kiryati. Piecewise-smooth dense optical flow via level sets. *Int. J. of Comp. Vision*, 68(2):111–124, 2006.
4. M. S. Andersen, D. L. Benoit, M. Damsgaarda, D. K. Ramsey, and J. Rasmussen. Do kinematic models reduce the effects of soft tissue artefacts in skin marker-based motion analysis an in vivo study of knee kinematics. *Journal of Biomechanics*, 43:268–273, 2010.
5. D. Anguelov, D. Koller, H.-C. Pang, P. Srinivasan, and S. Thrun. Recovering articulated object models from 3d range data. In *Proc. Conf. on Uncertainty in Artificial Intelligence*, pages 18–26, Arlington, Virginia, United States, 2004. AUAI Press.
6. R. Arcila, S. K. Buddha, F. Hétroy, F. Denis, and F. Dupont. A framework for motion-based mesh sequence segmentation. In *Int. Conf. on Comp. Graphics, Visual. and Comp. Vision*, Plzeň, Czech Republic, 2010.
7. M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, and A. Tal. Mesh segmentation - a comparative study. In *Proc. IEEE Int. Conf. on Shape Modeling and Applications*, pages 7–18, Washington, DC, USA, 2006. IEEE Computer Society.
8. P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *Trans. PAMI*, 14(2):239–256, 1992.
9. A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proc. National Academy of Science (PNAS)*, 103(5):1168–1172, 2006.
10. A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical geometry of non-rigid shapes*. Springer-Verlag New York Inc, 2008.
11. A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. *Int. J. of Comp. Vision*, 61(3):211–231, 2005.
12. X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3D mesh segmentation. *ACM Trans. Graphics*, 28(3), Aug. 2009.
13. D. Cremers and S. Soatto. Motion competition: A variational framework for piecewise parametric motion segmentation. *Int. J. of Comp. Vision*, 62(3):249–265, 2005.
14. M. Desbrun, M. Meyer, P. Schroder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. *Proc. SIGGRAPH*, pages 317–24, 1999.
15. M. P. DoCarmo. *Riemannian Geometry*. Birkhäuser Boston, 1992.
16. R. Duits and B. Burgeth. Scale spaces on lie groups. In *Proc. SSVM*, pages 300–312, Berlin, Heidelberg, 2007. Springer-Verlag.
17. T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38:293–306, 1985.
18. Y. Gur and N. A. Sochen. Regularizing flows over lie groups. *J. of Math. in Imag. and Vis.*, 33(2):195–208, 2009.
19. B. C. Hall. *Lie Groups, Lie Algebras,and Representations, An Elementary Introduction*. Springer, 2004.
20. S. Hauberg, S. Sommer, and K. S. Pedersen. Gaussian-like spatial priors for articulated tracking. In *Proc. CVPR*, pages 425–437. Springer, 2010.
21. D. Hochbaum and D. Shmoys. A best possible approximation for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985.
22. A. Iserles, H. Z. Munthe-kaas, S. P. Nrsett, and A. Zanna. Lie group methods. *Acta Numerica*, pages 215–365, 2000.

23. V. Jain and H. Zhang. Robust 3D shape correspondence in the spectral domain. In *Proc. of Shape Modeling International*, pages 118–129, 2006.
24. D. L. James and C. D. Twigg. Skinning mesh animations. *Proc. SIGGRAPH*, 24(3):399–407, Aug. 2005.
25. J. E. Jr. and J. H. Sampson. Harmonic mappings of Riemannian manifolds. *American J. of Math*, 86(1):106–160, 1964.
26. E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3D Mesh Segmentation and Labeling. *ACM Trans. Graphics*, 29(3), 2010.
27. M. Kobilarov, K. Crane, and M. Desbrun. Lie group integrators for animation and control of vehicles. *ACM Trans. Graphics*, 28(2):1–14, 2009.
28. I. Kompatsiaris, D. Tzovaras, and M. G. Strintzis. Object articulation based on local 3D motion estimation. In *Proc. of ECMAST*, pages 378–391, London, UK, 1999. Springer-Verlag.
29. T.-Y. Lee, Y.-S. Wang, and T.-G. Chen. Segmenting a deforming mesh into near-rigid components. *Vis. Comput.*, 22(9):729–739, 2006.
30. N. Litke, M. Droske, M. Rumpf, and P. Schröder. An image processing approach to surface matching. In *Proc. SGP*, pages 207–216, Aire-la-Ville, Switzerland, 2005. Eurographics Association.
31. D. Mateus, R. Horaud, D. Knossow, F. Cuzzolin, and E. Boyer. Articulated shape matching using Laplacian eigenfunctions and unsupervised point registration. In *Proc. CVPR*, pages 1–8, 2008.
32. A. Müller and Z. Terze. Differential-geometric modelling and dynamic simulation of multibody systems. *Strojarstvo: Journal for Theory and Application in Mechanical Engineering*, 51(6):597–612, 2009.
33. D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.
34. D. Raviv, A. Dubrovina, and R. Kimmel. Hierarchical shape matching. In *Proc. SSVM*, 2011. Accepted.
35. L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D Letters*, 60:259–268, 1992.
36. M. R. Ruggeri, G. Patanè, M. Spagnuolo, and D. Saupe. Spectral-driven isometry-invariant matching of 3D shapes. *Int. J. of Comp. Vision*, 89(2-3):248–265, 2010.
37. B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *Proc. CVPR*, pages 406–420. Springer, 2010.
38. A. Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6):1539–1556, 2008.
39. R. Subbarao and P. Meer. Nonlinear mean shift over riemannian manifolds. *Int. J. of Comp. Vision*, 84(1):1–20, 2009.
40. R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. In *Proc. SIGGRAPH*, pages 399–405, New York, NY, USA, 2004. ACM.
41. J. Tierny, J.-P. Vandeborre, and M. Daoudi. Fast and precise kinematic skeleton extraction of 3D dynamic meshes. In *Proc. Int. Conf. Pattern Recognition*, pages 1–4, 2008.
42. O. Tuzel, F. Porikli, and P. Meer. Learning on lie groups for invariant detection and tracking. In *Proc. CVPR*, pages 1–8, 2008.
43. S. Wuhrer and A. Brunton. Segmenting animated objects into near-rigid components. *The Visual Computer*, 26:147–155, 2010.
44. T. Yamasaki and K. Aizawa. Motion segmentation for time-varying mesh sequences based on spherical registration. *EURASIP Journal on Applied Signal Processing*, 2009.
45. N. N. Yanenko. *The method of fractional steps: solution of problems of mathematical physics in several variables*. Springer, 1971. Translated from Russian.