

# Fast and precise kinematic skeleton extraction of 3D dynamic meshes

Julien Tierny<sup>1</sup>, Jean-Philippe Vandeborre<sup>1,2</sup> and Mohamed Daoudi<sup>1,2</sup>

<sup>1</sup> LIFL (UMR USTL/CNRS 8022), Lille, France

<sup>2</sup> Institut TELECOM; TELECOM Lille 1, Lille, France

{julien.tierny, jean-philippe.vandeborre, mohamed.daoudi}@lifl.fr

## Abstract

*Shape skeleton extraction is a fundamental pre-processing task in shape-based pattern recognition. This paper presents a new algorithm for fast and precise extraction of kinematic skeletons of 3D dynamic surface meshes. Unlike previous approaches, surface motions are characterized by the mesh edge-length deviation induced by its transformation through time. Then a static skeleton extraction algorithm based on Reeb graphs exploits this latter information to extract the kinematic skeleton. This hybrid static and dynamic shape analysis enables the precise detection of objects' articulations as well as shape topological transitions corresponding to possibly-articulated immobile objects' features.*

*Experiments show that the proposed algorithm is faster than previous techniques and still achieves better accuracy.*

## 1. Introduction

Shape skeleton extraction and segmentation are fundamental shape pre-processing tasks in computer vision and shape-based pattern recognition. They provide an intrinsic structural shape description, which is useful for further applications dealing with shape understanding, like shape recognition or retrieval. With the ongoing development of 3D technologies, 3D dynamic shapes (time-varying 3D data) are becoming a media of increasing importance. Such data can be provided by scientific simulations, animation softwares, video games or security systems and are mostly modeled by constant connectivity surface meshes with time-varying geometry. Like 3D static shapes, 3D dynamic shapes also need to be pre-processed for their understanding and have recently drawn the shape analysis community's interest [6, 3]. However, only few papers have addressed the 3D dynamic surface mesh segmentation [5, 4] or skeleton extraction [1, 8, 2] specific problems so far.

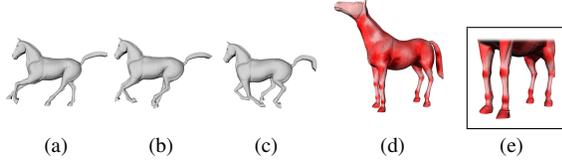
While static 3D shape segmentation methods aim at extracting a meaningful structural representation of the shape by decomposing it into parts of uniform geome-

try, 3D dynamic shape segmentation methods propose to exploit the temporal information to decompose the shape into parts of uniform motion along the sequence. Indeed, this approach is a faithful hypothesis for extracting the functional and thus meaningful structure of an object, revealing its articulation points for example. Consequently, existing techniques first try to characterize the motion uniformity over the surface mesh and then use clustering techniques to decompose it.

As most of real life objects' motions can be defined in terms of local rigid transformations (translations and local rotations), Mamou et al. [5], in the context of dynamic shape compression, propose to locally compute the optimal rigid transformations along the sequence frames for small surface neighborhoods using the least square method. Then, surface neighborhoods affected by the same rigid transformations are gathered using the k-means algorithm to produce the final segmentation. Lee et al. [4] adopt a similar strategy by first computing the so-called *deformation gradient* to characterize the rigid transformation prediction error over the surface and then employ a clustering-based static mesh segmentation algorithm on this error field. Skeleton-extraction techniques for 3D dynamic surface meshes [1, 8, 2] also characterize surface motion with rigid transformation predictions combined with clustering algorithms.

However, predicting local rigid transformations turns out to be computationally expensive and from our experience the computation (especially using least square methods) can suffer from numerical instabilities. Moreover, most of the existing techniques only exploit the temporal information and thus cannot extract immobile shape features which still potentially correspond to functional parts of the object.

In this paper, we present a novel hybrid technique for the extraction of 3D dynamic surface mesh *kinematic skeletons*, a shape representation that reveals the shape kinematic characteristics over the time. This work brings the following contributions. First, we propose a simple, fast and efficient technique for surface motion characterization which is not based on rigid transformation prediction. Then, we present a combined dy-



**Figure 1. Example frames of the *horse* dynamic surface mesh (1(a), 1(b), 1(c)) and its edge-length deviation (1(d), 1(e)).**

dynamic and static approach that precisely identifies the articulations of the object through time and which also distinguishes specifically immobile shape features that contain possibly articulated parts. The rest of the paper is structured as follows: we present motion characterization in section 2, skeleton extraction in section 3, experiments and results (with comparisons to other techniques) in section 4 and finally demonstrate the precision and the applicative interest of our technique in the context of animation reverse engineering (section 5).

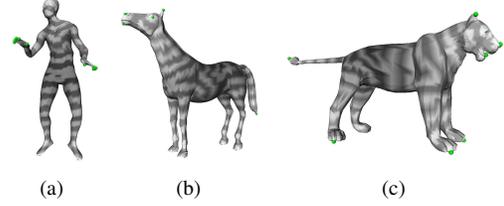
## 2. Edge-length deviation analysis

Let  $M$  be a constant connectivity closed surface mesh, whose vertices' positions vary through time  $p_t(v_i) = (x_i, y_i, z_i)_t, \forall v_i \in M$ . Surface portions that exhibit high rigid transformation prediction error through time correspond to the articulations of the object (affected by *elastic* transformations) and thus should match the boundaries of the final segmentation.

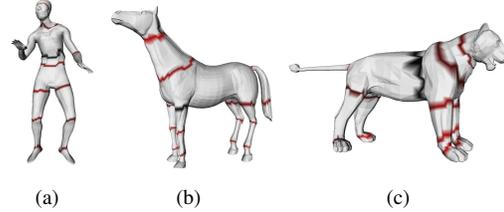
Considered transformations (translations and local rotations) belong to the group of isometric maps (length preserving isomorphisms). Thus, length preservation is also an invariant of rigid transformations. Consequently, we propose to detect surface portions that are not motioned through rigid transformations by computing the surface repartition of the edge-length deviation along the frames. In particular, we introduce the quadratic edge-length deviation  $L_d \forall v_i \in M$ , where  $T$  is the number of frames in the sequence,  $N(v_i)$  is the set of vertices sharing an edge with  $v_i$  and  $d$  is the euclidean distance:

$$L_d(v_i) = \sum_{t=0}^{T-1} \sum_{v_j \in N(v_i)} \frac{1}{|N(v_i)|} \times [d(p_t(v_i), p_t(v_j)) - d(p_{t+1}(v_i), p_{t+1}(v_j))]^2 \quad (1)$$

Figure 1 shows a dynamic shape and its corresponding edge-length deviation (reported on the first frame). Notice in figures 1(d) and 1(e) that light pink surface portions (low edge-length deviation) have been affected by local rotations in the sequence and correspond to rigid portions of the object. Moreover, the articulations of the legs exhibit a high edge-length deviation (dark red) as they have been affected by elastic transformations.



**Figure 2. Feature points (in green) and  $f$  level lines for several meshes.**



**Figure 3. Motion boundaries: connected components of  $f$  level lines locally maximizing the edge-length deviation  $L_d$ .**

## 3. Surface mesh skeleton extraction

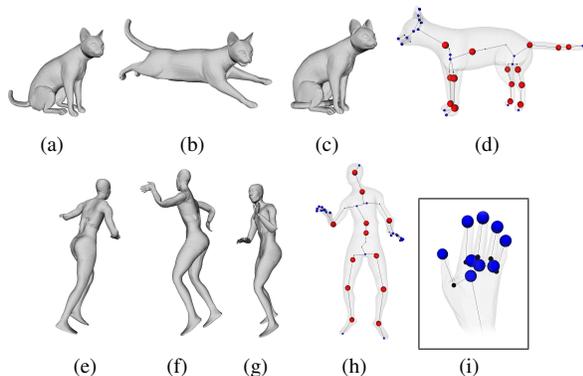
### 3.1. Segmentation boundary computation

The edge-length deviation provides an information similar to the deformation gradient [4] or the rigid error function [8]. Thus, a clustering based segmentation technique can be applied at this stage. Instead, we propose to adapt a static skeleton extraction technique based on Reeb graphs [9] in order to also distinguish shape topological transitions corresponding to possibly-articulated parts. This technique first computes automatically the shape feature points (vertices located at the extremity of prominent components, in green in figure 2) by intersecting geodesic maps extrema [9]. Then, in order to have a shape representation robust to pose change, it computes the following function, where  $g$  is the geodesic distance and  $v_f$  is the geodesic-closest feature point from  $v_i$ :  $f(v_i) = g(v_i, v_f) \quad \forall v_i \in M_{(t=0)}$ .

Then, for each vertex  $v_i \in M$ , the algorithm extracts an upper-value *discrete* approximation of  $f^{-1}(f(v_i))$ , which follows the edges of  $M$  [9]. In particular, let  $c_f(v_i)$  be the "contour" of  $v_i$  (the connected component of  $f^{-1}(f(v_i))$  approximation containing  $v_i$ ).

As  $f$  level lines follow the protrusions of the object (see figure 2), their connected components are good candidates for the definition of the segmentation boundaries. To select these boundaries among the collection of contours, for each contour  $c_f(v_i)$ , we compute its average edge-length deviation (averaging  $L_d(v_j) / v_j \in c_f(v_i)$ ). Then, we identify the contours  $c_f$  which locally maximize this value (with regard to their adjacent contours) as *motion boundaries* (see figure 3).

Finally, in order to extract immobile shape features



**Figure 4. 3D dynamic surface meshes and their extracted kinematic skeletons.**

corresponding to potentially functional parts,  $f$  critical points are extracted at the vertices where contours merge, split or terminate as  $f$  evolves [9]. The contours related to these vertices are denoted as *topological boundaries*, which are the boundaries of the Reeb graph [7] decomposition of  $M$ .

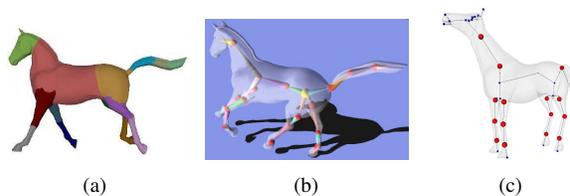
### 3.2. Skeleton embedding

To embed the kinematic skeleton in 3D, we first place the nodes of the Reeb graph ( $f$  critical points, in blue in figure 4) at the barycenter of the *topological boundaries*. Then, the edges of the Reeb graph (linking  $f$  critical points) are subdivided at *motion boundaries* to produce the final kinematic skeleton. In particular, *motion nodes* (in red in figure 4) have been placed at the barycenters of the *motion boundaries*.

## 4. Experiments and results

Figure 4 shows 3D dynamic meshes and their extracted kinematic skeletons. Notice that the motion nodes (in red) are precisely placed at the articulations of the objects (see the next section for precision evaluation). Moreover, the edges of the kinematic skeleton (black lines) correspond to rigid parts of the objects. Thus, motion nodes provides an information about the object motion, revealing the reference points of rigid part rotations. Moreover, the shape topology is efficiently captured by the Reeb graph [7]. Thus, the kinematic skeletons preserves the object’s topology.

Figure 5 presents a visual comparison of our algorithm with existing techniques. Notice that in figure 5(a) the hooves of the horse dynamic shape are not extracted as individual rigid parts, while their motion is different from the rest of the leg. On the contrary, our method precisely identifies articulations and is compatible with de Aguiar’s algorithm’s decomposition (fig.



**Figure 5. Visual comparison between Lee’s segmentation [4] (5(a)), de Aguiar’s skeleton [2] (5(b)) and our method (5(c)).**

Dynamic mesh	Faces	Frames	Extraction (s.)
Cat	14410	10	9.6
Dance	14118	201	21.2
Horse	16843	49	12.7
Lion	9996	10	6.0
Snake	18354	134	20.2

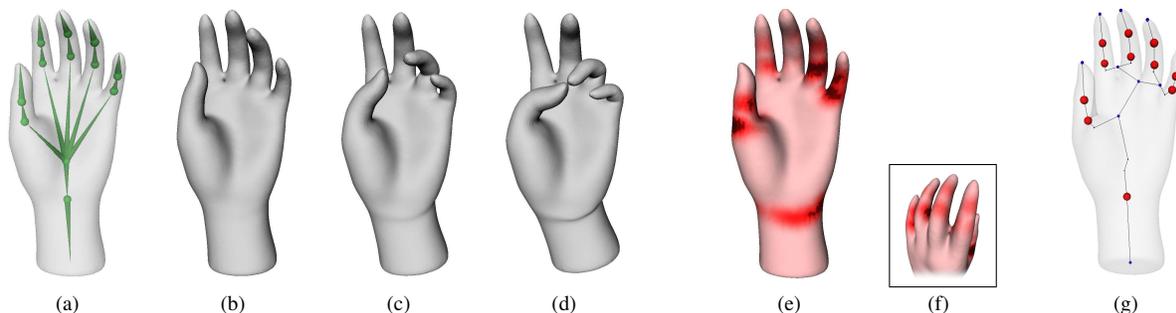
**Table 1. Computation times of kinematic skeleton extraction for several dynamic surface meshes (P4 3GHz CPU).**

5(b)). However, thanks to the Reeb graph and the feature point extractions, our algorithm can also extract and decompose specifically immobile shape features (in blue) like the hands of the dancer (fig. 4(i)), or the mouth and the ears of the horse, which are actually articulated parts of these objects in reality.

As motion analysis is not based on rigid transformation prediction (which requires matrix inversion), our algorithm runs faster than existing techniques (tables 1 and 2). Edge-length deviation computation runs in linear time:  $O(n \times T)$  steps, with  $n$  the number of vertices in  $M$  and  $T$  the number of frames in the dynamic shape. Contour and Reeb graph extractions (boundary computation) only requires  $O(n^2)$  steps [9]: as  $M$  has constant connectivity, its topology does not evolve; thus, the Reeb graph is computed only once (here, for  $M_{t=0}$ ).

## 5. Animation reverse engineering

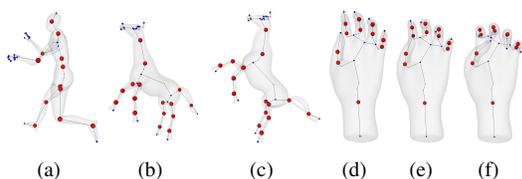
To demonstrate the precision of our algorithm and its applicative interest, in the following experiment, given a skeleton-driven animation of a static shape (realized with an animation software) we propose to retrieve the kinematic model of the animation by kinematic skeleton extraction. Figures 6(a), 6(b), 6(c) and 6(d) show the original designed skeleton and the resulting animation. Figure 6(e) shows the edge-length deviation of this dynamic shape. Notice that the index and the middle fingers have been slightly articulated but still exhibit length deviation (fig. 6(f)). Finally, figure 6(g) shows the extracted kinematic skeleton. Notice that each articulation of the original skeleton has been extracted as a motion node of the kinematic skeleton. Moreover,



**Figure 6. Animation reverse engineering: a dynamic surface mesh (6(b), 6(c), 6(d)) is generated from a predefined skeleton of a static mesh (6(a)). Edge-length deviation (6(e), 6(f)) and extracted kinematic skeleton (6(g)). Motion node location error: 0.44%.**

Algorithms	Skeleton extraction (s.)
Our algorithm	12.7
de Aguiar's [2]	21
Lee's [4]	806

**Table 2. Computation times for the horse dynamic surface mesh.**



**Figure 7. New dynamic surface meshes generated using the extracted kinematic skeletons.**

the average height ( $y$ ) difference between the original and the extracted skeletons' articulations is only **0.44%** of the height of the object bounding box, which is a satisfactory precision (**against 1.7%** in the best case for [2]). Finally, as skeleton extraction is achieved by segmenting the mesh at motion and topological boundaries, there is a full correspondence between the skeleton edges and the related surface segments. Thus, input dynamic shapes can be re-edited from their kinematic skeletons by applying local rotations on the edges of the skeleton, as shown in figure 7.

## 6. Conclusion and future works

In this paper, we presented a new hybrid algorithm for fast and precise kinematic skeleton extraction of 3D dynamic surface meshes based on edge-length deviation computation and Reeb graph extraction.

Experiments showed the rapidity of this method and demonstrated its precision. Thanks to this representation which concisely encodes the dynamic shape motion characteristics, dynamic shapes can be efficiently modeled and edited. As the edges of the skeleton represent

rigid parts of the object, their motion can be modeled by simple rotations.

In the future, we would like to use this representation for 3D dynamic shape comparison (for gesture recognition). Moreover, we would like to adapt the framework to take into account unregistered meshes (which is a crucial issue) and also to efficiently model surface intrinsically elastic transformations.

**Acknowledgements** This work is partially supported by the ANR through the MADRAS project (ANR-07-MDCO-015).

## References

- [1] D. Anguelov, D. Koller, H. Pang, P. Srinivasan, and S. Thrun. Recovering articulated object models from 3D range data. In *Uncertainty in Artificial Intelligence*, pages 18–26, 2004.
- [2] E. de Aguiar, C. Theobalt, S. Thrun, and H.-P. Seidel. Automatic conversion of mesh animations into skeleton-based animations. *Computer Graphics Forum*, 27:In press, 2008.
- [3] A. G. Kirk, J. F. O'Brien, and D. A. Forsyth. Skeletal parameter estimation from optical motion capture data. In *IEEE CVPR*, pages 782–788, 2005.
- [4] T.-Y. Lee, Y.-S. Wang, and T.-G. Chen. Segmenting a deforming mesh into near-rigid components. *The Visual Computer*, 22:729–739, 2006.
- [5] K. Mamou, T. Zaharia, and F. Prêteux. Multi-chart geometry video: A compact representation for 3D animations. In *IEEE 3DPVT*, pages 711–718, 2006.
- [6] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81:231–268, 2001.
- [7] G. Reeb. Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *Comptes-rendus des Séances de l'Académie des Sciences*, 222:847–849, 1946.
- [8] S. Schaefer and C. Yuksel. Example-based skeleton extraction. In *Symposium on Geometry Processing*, pages 153–162, 2007.
- [9] J. Tierny, J.-P. Vandeboorde, and M. Daoudi. Invariant high level Reeb graphs of 3D polygonal meshes. In *IEEE 3DPVT*, pages 105–112, 2006.