Deforming surface simplification based on dynamic geometry sampling

Frédéric Payan
Laboratoire J.Kuntzmann
Grenoble, France
Frederic.Payan@imag.fr

Stefanie Hahmann
Laboratoire J.Kuntzmann
Grenoble, France
Stefanie.hahmann@imag.fr

Georges-Pierre Bonneau
Laboratoire J.Kuntzmann
Grenoble, France
Georges-Pierre.Bonneau@imag.fr

Abstract

Although deforming surfaces are frequently used in numerous domains (scientific applications, games...), only few works have been proposed until now for simplifying such data. However, these time-varying surfaces are generally represented as oversampled triangular meshes with a static connectivity, involving a large number of unnecessary details for some frames. Among the related works, some methods provide globally good results, but fine details appearing during the animation are not always well-preserved, because of a static geometry sampling. We propose a new simplification method for deforming surfaces based on a dynamic geometry sampling. The idea is to compute one coarse version at the first frame, and then to progressively update the coarse sampling for the subsequent frames. In order to optimally approximate each frame, vertices are added or removed following the appearance or disappearance of fine details in the frames. Our approach is fast, easy to implement, and produces good quality time-varying approximations with well-preserved fine details, at any given frame.

1. Introduction

The temporal dimension of the deforming surfaces is particularly attractive in numerous domains. From scientific applications (simulation or physical processes...) to animation (movie, games, morphing...), these time-varying surfaces are frequently used to represent and visualize models or objects in motion.

Deforming surfaces are most of times represented by dense triangular meshes sharing the same connectivity. The shape deformations are simply obtained by spatially displacing the vertices across the time. This representation has a lot of advantages (fast visualization processing, simple data structure, storage), but only for specific surfaces like surfaces undergoing few non-rigid deformations.

For surfaces with strong non-rigid deformations, the static

connectivity involves that the meshes are particularly oversampled at some frames. But, this surplus of vertices is necessary to represent the fine details existing at other frames of the animation. Figure 1 illustrates this problem. It shows one hoof of the morphing sequence HORSE-TO-MAN at the first frame (Figure 2). During the morphing, this hoof is progressively transformed in one hand. The amount of unnecessary vertices at the first frames is nothing but the fingers of the man appearing at the next frames.
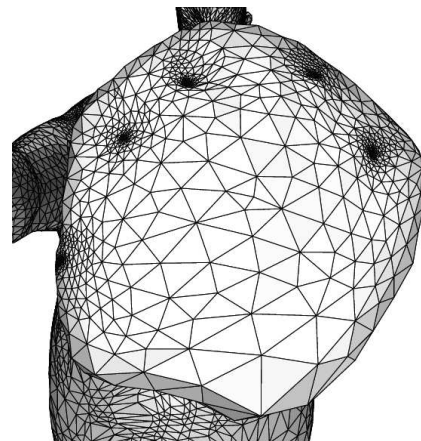
Figure 1. Example of an oversampled region of HORSE-TO-MAN**. The unnecessary vertices will be used to represent the fingers appearing during the morphing.**

In case of static surface meshes, this problem could be easily solved by applying one of the numerous existing simplification algorithms [9, 10, 8]. For a complete overview, see [7, 15].

Unfortunately, for time-varying meshes, the problem of simplification has been rarely addressed [19, 13, 16, 4, 11], and adapting static mesh simplification algorithms to surfaces with strong non-rigid deformations is inefficient (see fig. 3(b)). Recently, a relevant method based on
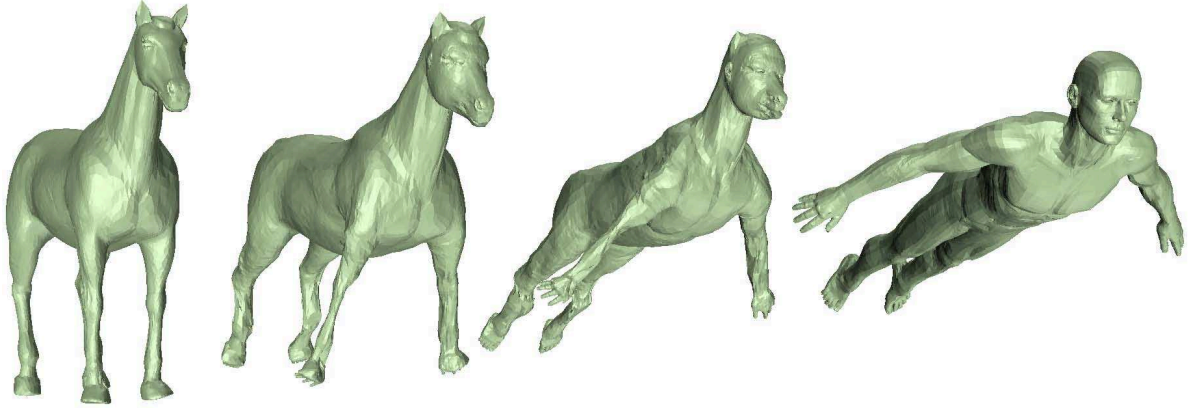
Figure 2. The morphing sequence HORSE-TO-MAN **is represented by a time-varying mesh with a static connectivity (17489 vertices, 34974 triangles)..**

connectivity transformations has been proposed for such surfaces [11]. Despite of this, some fine details appearing during the animation are not always well-preserved even with this method (see fig. 3(c)). The main reason is that a connectivity transformation is not sufficient, when constrained by a fixed geometry sampling.

We therefore propose a new simplification method based on a dynamic geometry sampling, for complex deforming surfaces. The principle is to compute one coarse version of the mesh at the first frame, and then to progressively update the coarse sampling during the animation (by adding or removing vertices), in order to well approximate the mesh at any frame, while preserving appearing details.

The rest of the paper is organized as follows. Section 2 presents the previous works related to simplification of deforming surfaces. Section 3 introduces the problem statement and the approach proposed in this paper. Section 4 deals with the simplification technique used to compute the coarse version of the mesh at the first frame. Section 5 presents the updating scheme based on dynamic geometry sampling providing time-varying approximated meshes. Section 6 gives some experimental results. We finally conclude and present some future works in Section 7.

2. Related works

Shamir *et al.* [19] are the first to adress the problem of simplifying efficiently deforming surfaces. They propose a global multiresolution structure named the TDAG (Time-dependent Directed Acyclic Graph) which allows to represent any time-varying mesh, at any level of details. This structure has the advantage to accept any arbitrary topology and any shape deformation. Unfortunately this scheme is complex, and cannot be easily handled, even if the time-varying mesh has a fixed connectivity [18].

Mohr and Gleicher [16] directly adapt the simplification algorithm for static meshes $Qslim$ [8]. They sum for each vertex the quadrics relative to its different positions during the animation, before applying the decimation sequence. This approach produces a single multiresolution hierarchy for all the frames. Consequently this technique provides a good approximation of the original time-varying mesh only when the original surfaces do not present strong deformations or fine details.

Decoro and Rusinkiewicz [4] propose a view- and pose-independent method for simplifying the skeletally articulated meshes. This technique is efficient but limited to very specific kinds of shape deformations.

In parallel, Kobbelt *et al.* [13] modify the classical concept of multiresolution representation by no longer requiring a global hierarchical structure linking the levels of details. Contrary to the previous works, this approach modifies the connectivity during the animation, and produces a better approximation of the original surface at any frame. The idea was to implicitly represent the details by using the geometric difference between independent approximated meshes. The details are computed by shooting rays in normal direction from one surface to another one. Unfortunately, considering only the normals to compute the details limits the efficiency of the method. very specific kinds of shape deformations.

More recently, Kircher and Garland [11] propose a multiresolution representation with a dynamic connectivity for deforming surfaces. Their idea is to produce a coarse ver-
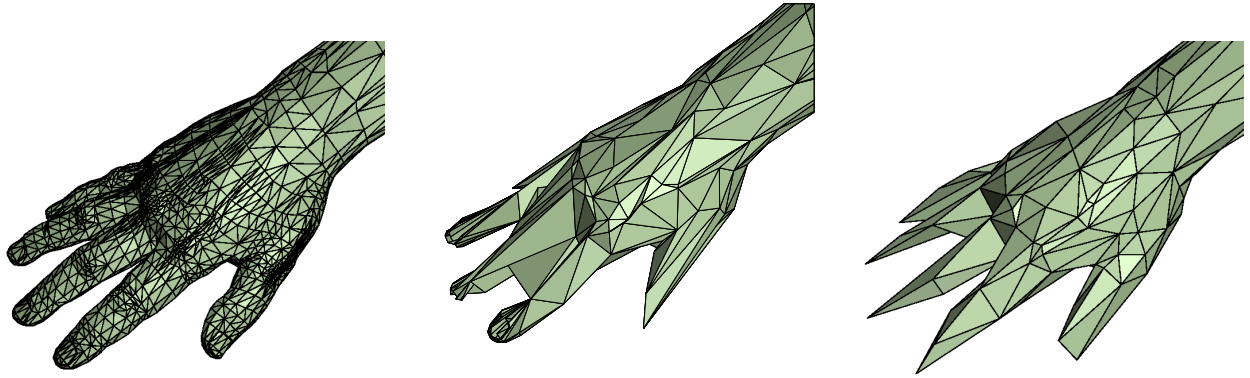
(a) Original last frame: 17489 vertices. (b) Static connectivity approximation (3200 vertices). (c) fixed geometry sampling (3200 vertices).

Figure 3. Approximating a time-varying surface with methods based on single static connectivity (fig. 3(b)) or fixed geometry sampling [11] (fig. 3(c)) does not preserve the appearing fine details.

sion of the first frame and then to adapt this base mesh with a sequence of update transformations to produce a good approximation for the other frames. The update operation consists in edge swaps, due to a reclustering [7] of the decimated vertices. This method seems to be particularly efficient because of its connectivity transformation. Finally, this method parallels the techniques of connectivity transformations for mesh morphing [14, 1, 2].

3. Problem statement and proposed approach

Among the techniques introduced in the previous section, the method of Kircher and Garland is particularly interesting [11]. Its main advantage is the potential connectivity modification (edge swaps) of a coarse mesh at any frame adapted to the shape deformations during the animation. The limitation of this approach is due to the fact that the connectivity changes are restricted to edge swaps, and that the geometry sampling is fixed[1]. This results in two drawbacks:

- some fine details appearing with the shape deformations are not well preserved since no vertex is added during the animation. We observe this drawback on fingers on the last frames of the sequence HORSE-TO-MAN (Fig. 3);

- on the other hand, some vertices relative to details disappearing in one frame are unnecessarily kept in numerous frames.

So, even if the edge swaps due to the reclustering are clearly efficient, the results are not optimal for shapes with strong local deformations when a fixed number of vertices is used during the simplification.

One solution to overcome this problem is to allow a non fixed sampling of the geometry. In other words, additional vertices could be added in regions where fine details appear [14]. In parallel, some vertices should be removed where details disappear during the animation. Hence, a more accurate approximation for any frame is obtained.
By considering a non fixed geometry sampling, the extreme and naive solution is to compute an entirely new simplified mesh at each frame. But this solution is irrelevant, since it does not exploit the temporal coherence of the data and involves some unpleasant visual artefacts (vibrating and twitching surfaces).

We therefore propose a new simplification algorithm for deforming surfaces, based on a dynamic geometry sampling. This method preserves fine details appearing during the animation, without computing an entirely new approximation hierarchy at each frame. The idea is to create a coarse version of the first frame, and then to progressively update the base simplification hierarchy for the subsequent frames by changing the geometry sampling. In other words, we propose to automatically add or remove some specific vertices according to their relevance at any frame. This can be done by a fine-to-coarse updating scheme, contrary to [11], which uses a coarse-to-fine one.

The overview of the proposed algorithm is the following:

---

[1]In parallel to our works, the authors have introduced in [12] a vertex teleportation method that is also meant to solve this problem

1. simplification of the first frame (until a user-given number of vertices is reached). The associated decimation sequence will be considered as the base simplification at the next frame;

2. for any given frame, we apply the simplification of the previous frame on the given mesh while testing if decimations would eliminate some appearing fine details. Such decimations are not applied at this frame.

3. once the previous simplification partially applied at this frame, the resulting simplified mesh obviously have more vertices than the previous one. So, we propose to remove additional vertices until the fixed number of vertices is reached.

4. Creation of the base hierarchy

In this paper we only focus on manifold time-varying meshes. Let us define the deforming surface S represented by a sequence of meshes sharing the same connectivity: $S = \{M_0, M_1, ...\}$. The first stage of our algorithm is to simplify the mesh M_0 by applying a sequence of n decimations $\{dec_0^i\}_{i \in \{0,..,n-1\}}$:

$$(M_0 = M_0^n) \xrightarrow{dec_0^{n-1}} M_0^{n-1} \xrightarrow{dec_0^{n-2}} ... \xrightarrow{dec_0^0} M_0^0. \quad (1)$$

M_0^0 represents the simplified surface at the first frame. As simplification technique, we choose the *Qslim* algorithm of Garland and Heckbert [7]. To evaluate the approximation error relative an edge collapse (*i.e.* the *cost* of removing this edge), we use the simple unweighted *Quadric Error Metric* (QEM) [8]. Moreover, to ensure that our algorithm produces time-varying approximated meshes preserving the topological type at any frame, we add the topology test proposed by Dey *et al.* [5] to the *Qslim* algorithm. This test determines if an edge collapse produces non-manifold meshes. This stage provides the simplified version M_0^0 of the first mesh M_0, but also determines the base hierarchy that will be progressively updated to simplify the following meshes relative to the deforming surface[2]. This reference hierarchy H_0 is defined as the set of edge collapses $\{ec_0^i\}_{i \in \{0,..,n-1\}}$ applied on the mesh M_0, and the associated contraction costs $\{cost_0^i\}_{i \in \{0,..,n-1\}}$:

$$H_0 = \{(ec_0^{n-1}, cost_0^{n-1}), ...(ec_0^0, cost_0^0)\}.$$

5. Updating scheme

5.1 Principle

Once the coarse mesh M_0^0 obtained, we have to deal with the other meshes M_1, M_2... Applying directly the

---

sequence of edge contractions defined at the first frame to all the subsequent frames produces bad time-varying approximations, as pointed out in [11]. Since this sequence depends on the positions of the vertices at the first frame, this is not adapted any more for the other frames.

We therefore propose a progressive updating scheme of the base hierarchy to adapt the sequence of contractions for each mesh, in order to obtain an accurate time-varying approximation at any frame. The originality of our approach is to detect the fine details appearing at some frame, and then to preserve them thanks to geometry sampling transformations across the time. Consequently, the undesired oversampling when details appear/disappear (as illustrated in Figure 1) is automatically removed in all other remaining frames.

Let us consider now the mesh M_f and its simplified version M_f^0 ($1 \leq f$). The principle of our method is to apply the reference contractions $\{ec_{f-1}^i\}$ on M_f only if the costs $\{cost_f^i\}$ at this frame do not differ too much from the costs $\{cost_{f-1}^i\}$ at the previous frame $f-1$. If an edge contraction would introduce a too important additional cost, the contraction is not applied on M_f. Hence, we aim to detect and preserve appearing details, and we obtain a simplified mesh M_f^{n-m}, with m the number of still valid contractions ($m \leq n$).

To estimate the cost evolution for a given edge contraction, the relative errors er_f^i between the reference cost and the new one is computed: $er_f^i = \frac{cost_f^i - cost_{f-1}^i}{cost_{f-1}^i}$. At present, to evaluate the cost evolution, we simply use a user-given threshold ϵ:

$$er_f^i = \frac{cost_f^i - cost_{f-1}^i}{cost_{f-1}^i} < \epsilon. \quad (2)$$

The mesh M_f^{n-m} obviously has more vertices than the approximated M_{f-1}^0, since potentially less contractions have been applied at this frame. If we want time-varying approximated surfaces with a fixed number of vertices, $(n-m)$ contractions are missing to obtain the vertex count of the coarse mesh M_{f-1}^0. We therefore make some additional contractions on M_f^{n-m} by applying the simplification algorithm presented in Section 4, until the objective of n contractions is reached.

We finally obtain for each frame a sequence of n edge collapses depending of the reference hierarchy, allowing a multiscale representation of the input deforming surface. Thus, time-varying surfaces can be displayed, at any level of details and at any frame. the pseudo-code of our approach is given below.

```
 1:  $(M_0^0, H_0) \longleftarrow simplify(M_0,n)$;
 2:  for $f = 0$ to $T - 1$ do
 3:      $m \longleftarrow 0$
 4:      for $i = n - 1$ to $0$ do
 5:          $cost_f^i \longleftarrow computeCost(M_f, ec_{f-1}^i)$;
 6:          if $\frac{cost_f^i - cost_{f-1}^i}{cost_{f-1}^i} < \epsilon$ then
 7:              $M_f^{n-m} \overset{ec_{f-1}^i}{\longrightarrow} M_f^{n-m-1}$
 8:              $ec_{f-1}^m \longleftarrow ec_{f-1}^i$
 9:              m++;
10:          end if
11:          if $n \neq m$ then
12:              $(M_f^0, H_f) \longleftarrow simplify(M_f^{n-m}, n - m)$;
13:          end if
14:      end for
15:  end for
```

5.2   Choice of the threshold value

This algorithm involves that a set of edge collapses is common during the simplification of subsequent meshes. The number of common collapsed edges mainly depends on the user-given threshold ϵ. On one hand, a too high value for ϵ provides that most of collapsed edges of the reference will be good at any frame. In that case, we would obtain only one single hierarchy for all frames, leading to the well known bad approximations at some frames. On the other hand, a too small value for ϵ would tend to create an entirely new hierarchy at any frame, causing unpleasant visual artefacts (vibrations and twitching surfaces), and increasing significantly the time computation. Thus the choice of ϵ is essential.

At present, the threshold value is user-given, but some other solutions are possible in order to compute it automatically. Unfortunately all these solutions would increase the computation time. At least they would need some pre-computation time. For example, one could estimate a threshold value empirically by applying a fixed number of edge contractions for some arbitrary frames individually and then taking a mean value of the resulting thresholds. Many variants of this procedure cam be imagined. Also more sophisticated statistical models can be used.

Another possibility would be to replace the current static threshold by a variable one, in order to take into account more accurately the local shape deformations. To do this, it would be necessary to look at the entire sequence, or at least a large number of frames. Such a threshold would be an advantage for surfaces with few deformations, since this avoids some potential flickering due to collapses close to the threshold value. However, a relevant argument for using a static threshold is the possibility of on-line processing, *i.e.* treating the meshes as they come in (such a scan-based approach enables a memory cost limited to the one needed to store only one frame). The same argument holds for some more sophisticated error measures, since the on-line processing would not be possible anymore, and the time efficiency would globally suffer.

5.3   Time complexity

In this section, we discuss the time complexity of our approach. The first step corresponds to the function *simplify* producing the coarse mesh M_0^0 and the reference hierarchy. This function is the *QSlim* algorithm improved by the topological type preservation technique of [5]. Thus the complexity of this step is the complexity of the *QSlim* algorithm [6], *i.e.*, $O(n \log n)$ (by assuming that $O(n)$ candidate edges are selected). This complexity is due to the handling of the priority queue.

For any subsequent frame, the estimation of the cost evolution given by (2) requires $O(n)$ time. Collapsing the m edges verifying (2) takes only $O(m)$ time, since a priority queue is unnecessary at this step. Then, the complexity of applying the $(n - m)$ additional contractions with the function *simplify* used for the first frame is $O((n - m) \log(n - m))$.
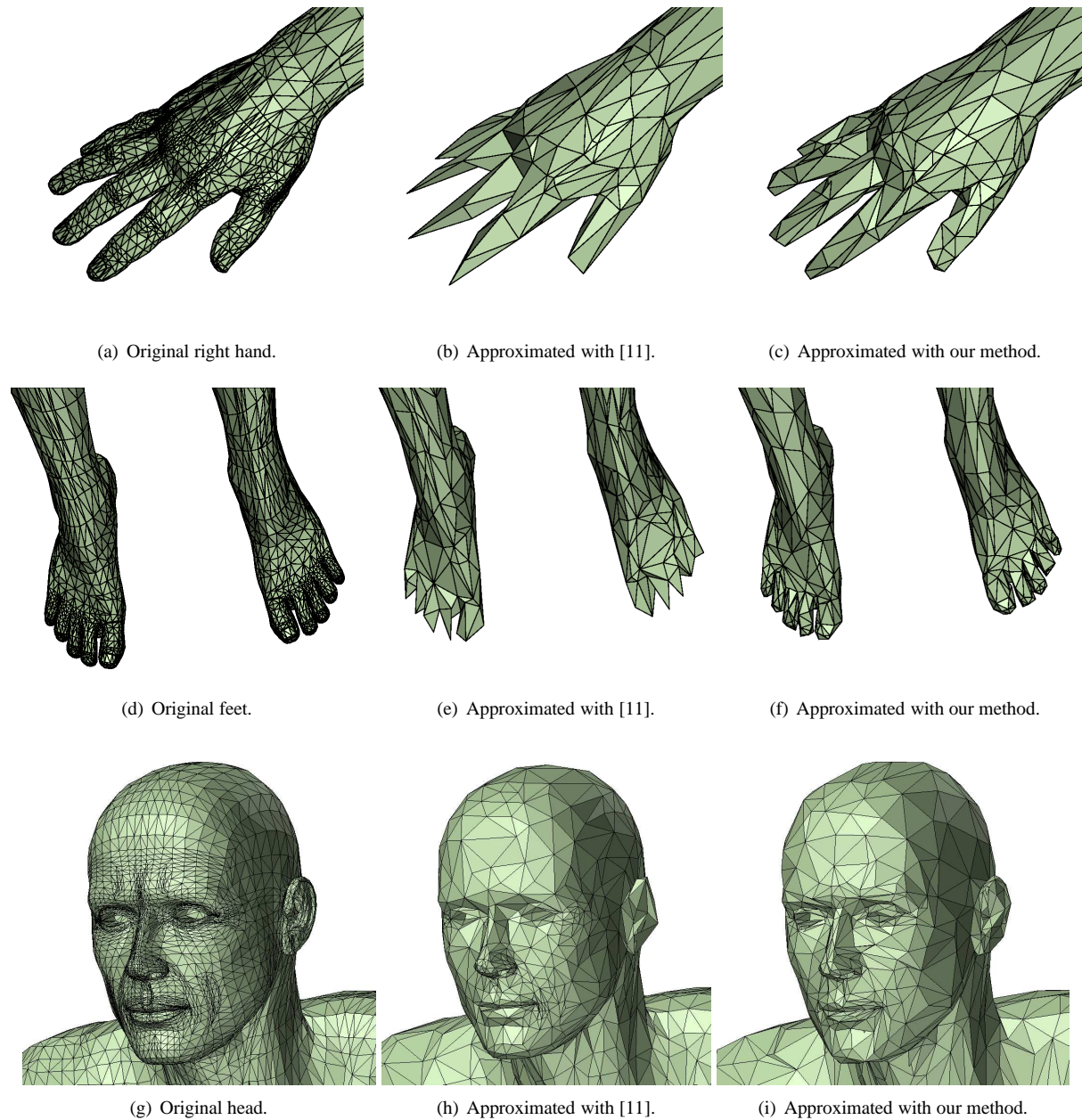In the best case, *i.e*, $(m = n)$, the updating scheme only requires $O(n)$ time. In the worst case, *i.e*, $(m = 0)$, the complexity of the updating scheme for one frame is approximatively equivalent to the complexity of producing the first mesh: $O(n \log n)$.
Finally, by assuming that the deforming surface is represented by T meshes, the overall complexity of our algorithm in the best case is $O(n \log n + (T - 1) \cdot n)$, and in the worst case, is $O(T n \log n)$.

In section 5.2, we discussed the efficiency of our algorithm, which depends on the threshold value of (2). Note that the time complexity depends on this value as well, because of the output parameter $(n - m)$. The best case of the time complexity $(m = n)$ corresponds to the highest threshold value, whereas the worst case $(m = 0)$ corresponds to the lowest threshold value. In the examples given in section 6 for instance, the mean ratio $\frac{n-m}{m}$ along the sequence is always less than 0.5%. For these examples, we tend to be in the best case, meaning that the complexity of this example is $O(n \log n + T \cdot n)$.

6. Results

In this section we present some experimental results relative to our algorithm. First, Figure 4 compares the original last frame of the time-varying surface HORSE-TO-MAN (200 frames, 17489 vertices and 34974 triangles) with

(a) Original right hand.

(b) Approximated with [11].

(c) Approximated with our method.

(d) Original feet.

(e) Approximated with [11].

(f) Approximated with our method.

(g) Original head.

(h) Approximated with [11].

(i) Approximated with our method.

Figure 4. Comparison of the original last frame of the time-varying surface HORSE-TO-MAN **(200 frames,** 17489 **vertices and** 34974 **triangles) with approximated versions (**3200 **vertices and** 6396 **triangles) obtained with [11] and with our method (**$\epsilon = 0.2$**). We observe that the fine details are well-preserved with our method based on dynamic geometry sampling.**
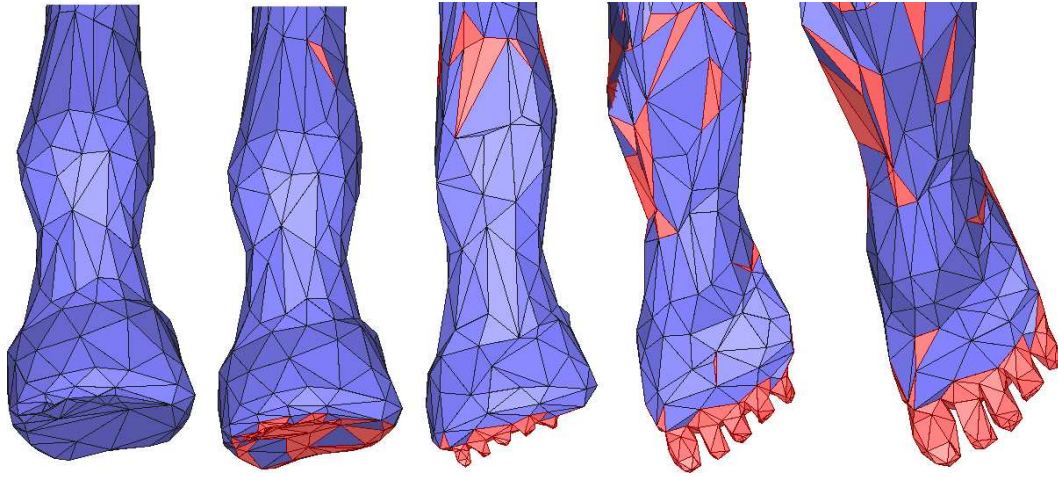
Figure 5. Time-varying approximation of one "hoof". The red triangles correspond to the triangles appearing during the animation, thanks to our dynamic scheme. At the first frame, the oversampling of the original hoof has been strongly removed. On the other hand, observe at the other frames the additional vertices in this region, allowing to preserve the fingers appearing during the animation.

approximated versions (3200 vertices and 6396 triangles) obtained either with [11] or with our method ($\epsilon = 0.2$). We observe that the regions with fine details appearing during the animation (head, hands, feet) are better-preserved by our algorithm. This is due to the fact that our updating scheme based on dynamic geometry sampling takes into account the local shape deformations.

Figure 5 shows the evolution of one of the hoofs simplified with our method. The red triangles correspond to the triangles appearing during the animation. At the first frame, we observe that the oversampling of the original hoof has been strongly removed. On the other frames, some additional vertices only appear in order to allow good approximations of the fingers appearing during the animation.

In addition, Figure 6 shows several frames of the HORSE-TO-MAN sequence approximated with our method, at different levels of details : 6000 vertices (first row), 3200 vertices (second row), and 800 vertices (third row). We can observe that the appearing fine details are well-preserved whatever the desired level of details.

To evaluate the efficiency of our approach, Figure 8 gives the evolution of the RMS error (normalized by the bounding box diagonal of the original meshes [3]) for two different sequences, HORSE-TO-MAN and COW (2904 vertices, 5804 triangles, see Figure 7), according to different simplification methods:

- "static hierarchy" is using the decimation sequence ob-

tained at the first frame for all the subsequent frames;

- "independent hierarchy" is applying an independent simplification at each frame, *i.e.*, the "optimal results";

- "Kircher-Garland" corresponds to the method based on connectivity transformations and static geometry sampling of [11].

- the 3 other curves are using our method with different values for ϵ.

The resulting approximations have respectively 3200 and 704 vertices for the HORSE-TO-MAN and the COW. We observe that our method provides significantly better results than the "static hierarchy" method, for the two data. Moreover, as expected, the lower the threshold value, the higher is the quality of the approximated surface. Furthermore, this is not surprising that our method provides slightly worse results than the method of [11], when the same number of vertices is fixed. This was expected, since the main objective of our method is to preserve fine details, and not to minimize the global error. Preserving regions with appearing details implies that less vertices are located in the other approximated regions. The global error is thus higher. Nevertheless, from a distortion point-of-view, our method provides good results.

Moreover, to underline that a dynamic geometry sampling do not cause unacceptable visual artefacts (vibrations or twitching surfaces, see Section 1), Figure 9 shows the number of triangles appearing at a given frame (in comparison with the previous one) for the two sequences.
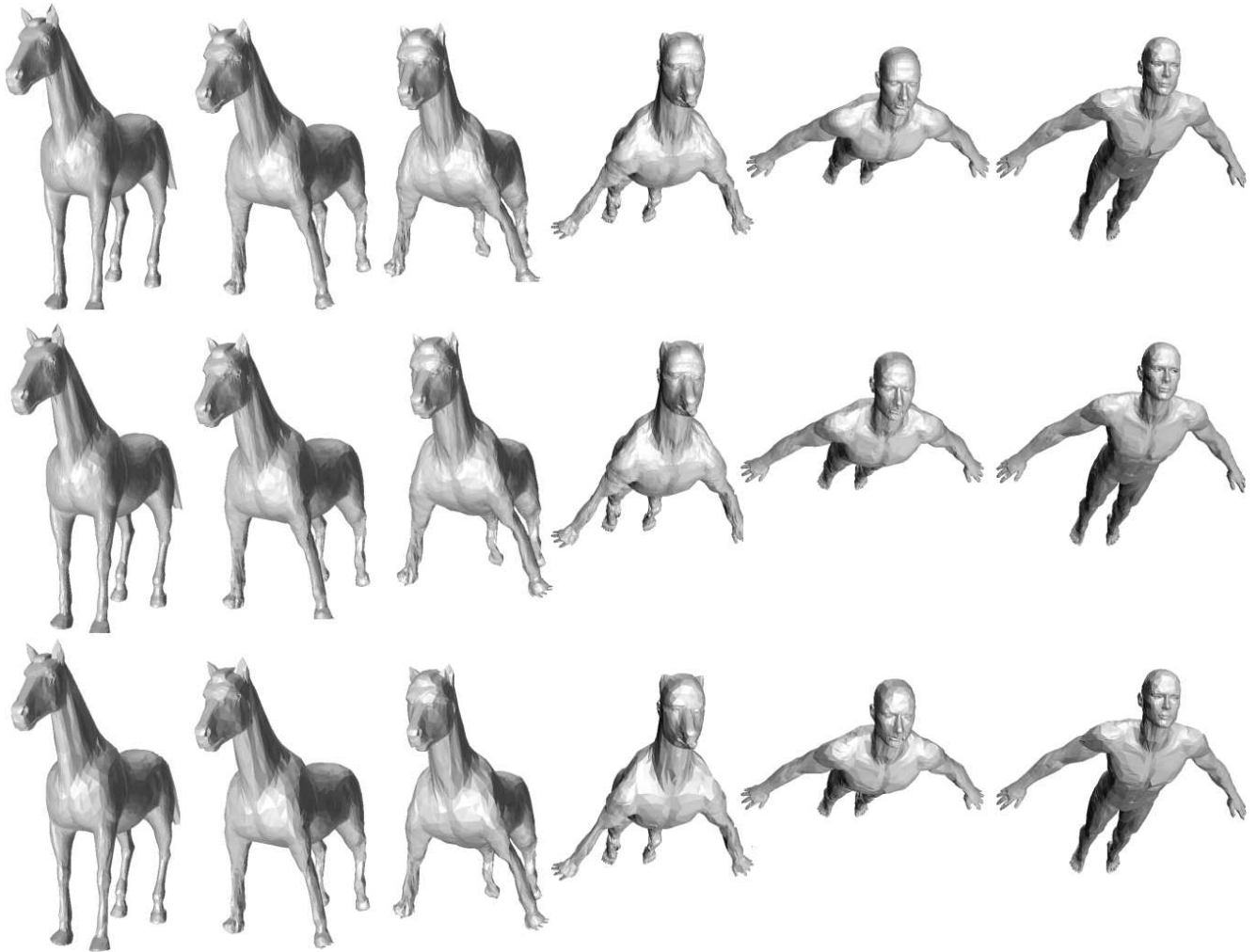
Figure 6. Time-varying Approximation of the HORSE-TO-MAN **sequence, at different levels of details :** 6000 **vertices (first row),** 3200 **vertices (second row), and** 800 **vertices (third row).**

As expected, the number of appearing triangles is inversely proportional to the threshold value. We can also observe that the number of triangles appearing between two successive frames is obviously lower than when using an independent simplification at each frame (dark blue curves). As discussed in section 5.2, there is a risk of creating undesired vibrations and twitching triangles due to the dynamic geometry sampling (in particular when the threshold value is small), but in practice our method leads to relatively smooth connectivity changes, and finally the resulting visual artefacts are not important. This is due to the fact that the appearing triangles are generally smaller than the other ones at this frame. The visual artefacts are therefore minimized. As a proof, the accompanying material contains two videos, the approximated sequence HORSE-TO-MAN resulting from our method (3200 vertices,

threshold $\epsilon = 0.2$), and the approximated sequence when using an independent hierarchy at each frame.

7. Conclusions and future works

In this paper, we have presented a simplification method for deforming surfaces based on dynamic geometry sampling. Given a sequence of meshes representing time varying 3D data, our method produces a sequence of simplified meshes that are good approximations of the original deformed surface for a given frame. A sequence of edge collapses is computed for the first frame only and is then adapted by a fine-to-coarse updating scheme which fully takes advantage of the temporal coherence of the data. This method is fast, easy to implement and produces good ap-
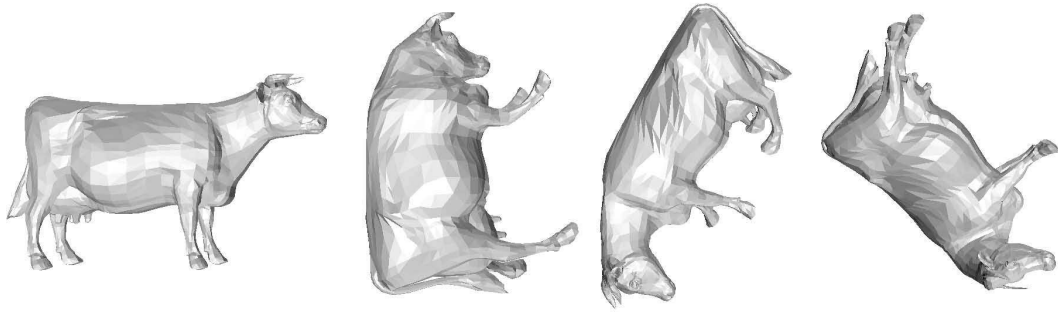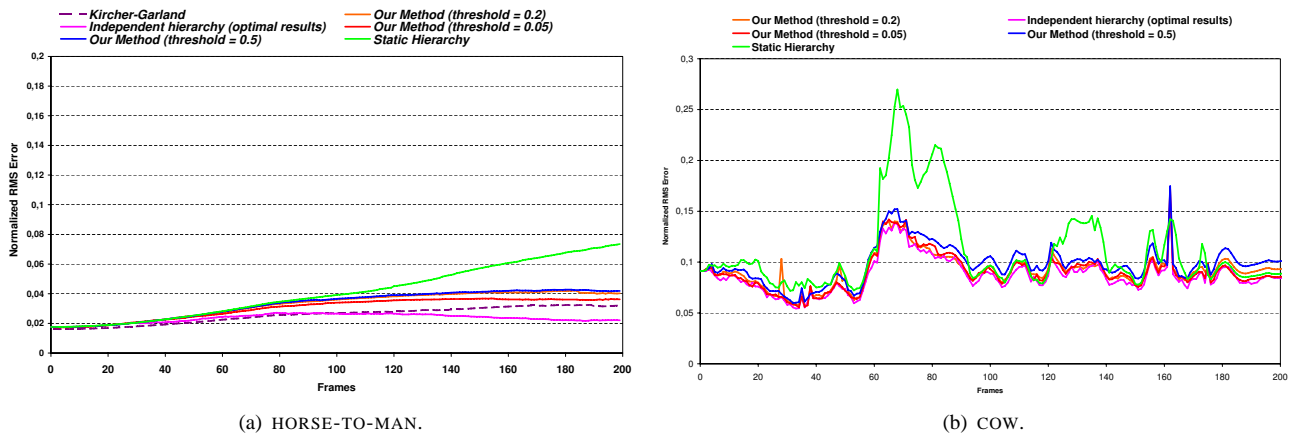
Figure 7. Several frames of the original sequence COW**.**

(a) HORSE-TO-MAN.

(b) COW.

Figure 8. Evolution of the normalized RMS error per frame. Comparison of our method (for three different threshold values ϵ**) with some alternative methods for the sequences** HORSE-TO-MAN **and** COW**.**
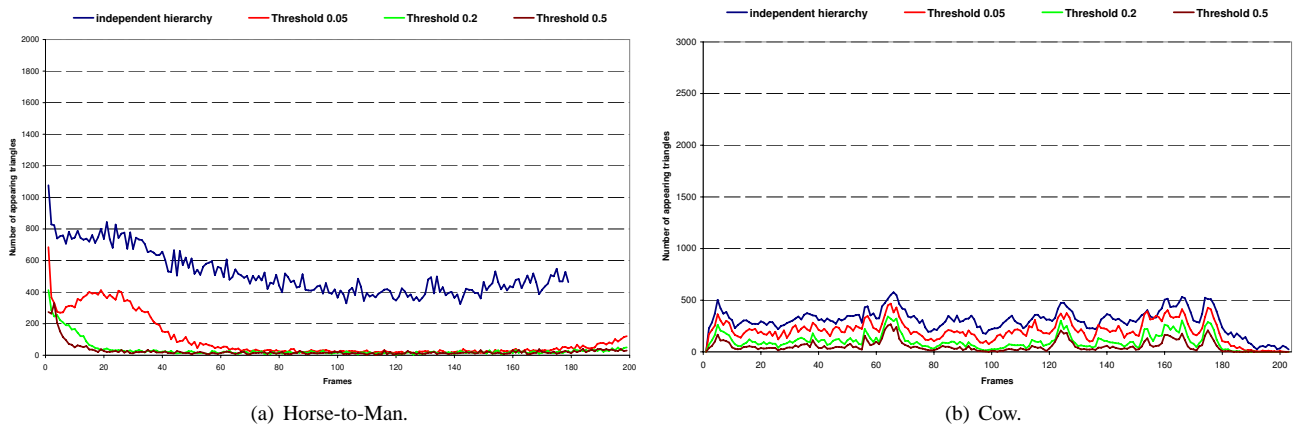

(a) Horse-to-Man.

(b) Cow.

Figure 9. Number of varying triangles between two successive frames for our method with three different threshold values ϵ **in comparison with the independent hierarchy method.**

proximations at any frame. The dynamic geometry sampling enables in particular to well preserve the fine details appearing during the animation. The problem of undesired oversampling of some mesh frames in order to represent appearing/disappearing fine details with a fixed mesh connectivity doesn't occur with our method. Moreover, the progressive representation of each mesh allows to display the deforming surfaces at any level of details, at any frame. Several examples and video have been produced in order to illustrate and to analyze the present algorithm. The technique also enables to create time-varying approximated surfaces with regular quality. All we have to do is to apply our updating scheme until we obtain a user-given target quality, instead of the fixed number of vertices.

Future research would include the application of the present method to view-dependent simplification of deforming surfaces. In fact, the use of edge collapses and the simplicity of the multiscale data may also permit to implement an efficient view-dependent simplification algorithm [17] for deforming surfaces.

Acknowledgments

References

[1] M. Ahn and S. Lee. Mesh metamorphosis with topology transformations. In *10th Pacific Conference on Computer Graphics and Applications*, 2004.

[2] M. Ahn, S. Lee, and H.-P. Seidel. Connectivity transformations for mesh metamorphosis. In *Eurographics Symposium on Geometry Processing 2004*, 2004.

[3] N. Aspert, D. Santa-Cruz, and T. Ebrahimi. Mesh: Measuring errors between surfaces using the hausdorff distance. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, volume I, pages 705 – 708, 2002.

[4] C. DeCoro and S. Rusinkiewicz. Pose-independent simplification of articulated meshes. In *Symposium on Interactive 3D Graphics*, Apr. 2005.

[5] T. Dey, H. Edelsbrunner, S. Guha, and D. Nekhayev. Topology preserving edge contraction, 1998.

[6] M. Garland. *Quadric-Based Polygonal Surface Simplification*. PhD thesis, Carnegie Mellon University, 1998.

[7] M. Garland. Multiresolution modeling: Survey & future opportunities. *Eurographics '99 – State of the Art Reports*, pages 111–131, 1999.

[8] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. *Proceedings ACM SIGGRAPH'97*, 1997.

[9] H. Hoppe. Progressive meshes. *Proceedings ACM SIGGRAPH'96*, pages 99–108, August 1996.

[10] H. Hoppe. Efficient implementation of progressive meshes. *Computer and Graphics*, 22(1), 1998.

[11] S. Kircher and M. Garland. Progressive multiresolution meshes for deforming surfaces. In *ACM-Eurographics Symposium on Computer Animation*, pages 191–200, 2005.

[12] S. Kircher and M. Garland. Editing arbitrarily deforming surface animations. *ACM Transactions on Graphics, Proceedings of SIGGRAPH 2006*, july 2006.

[13] L. P. Kobbelt, T. Bareuther, and H.-P. Seidel. Multiresolution shape deformations for meshes with dynamic vertex connectivity. *Computer Graphics Forum*, 19(3), 2000.

[14] C. Lin and T. Lee. Metamorphosis of 3d polyhedral models using progressive connectivity transformations. *IEEE Transactions on Visualization and Computer Graphics*, 2005.

[15] D. Luebke, M. Reddy, and J. Cohen. *Level of Detail for 3-D Graphics*. Morgan Kaufmann, 2002.

[16] A. Mohr and M. Gleicher. Deformation sensitive decimation. Technical report, University of Wisconsin Graphics Group, 2003. http://www.cs.wisc.edu/graphics/Gallery/DSD/.

[17] R. Pajarola and C. DeCoro. Efficient implementation of real-time view-dependent multiresolution meshing, 2004.

[18] A. Shamir and V. Pascucci. Temporal and spatial level of details for dynamic meshes. In *Proceedings of Virtual Reality Systems and Techniques*, pages 423–430, 2001.

[19] A. Shamir, V. Pascucci, and C. Bajaj. Multiresolution dynamic meshes with arbitrary deformations. In *Proc. IEEE Visualization'00*, pages 423–430, 2000.