Computers & Graphics 47 (2015) 16–23

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Low budget and high fidelity relaxed 567-remeshing

Vincent Vidal [a,*], Guillaume Lavoué [b], Florent Dupont [a]

[a] Université de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205, Villeurbanne F-69622, France
[b] Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, Villeurbanne F-69621, France

ARTICLE INFO

Article history:
Received 17 July 2014
Received in revised form
23 October 2014
Accepted 28 October 2014
This article was recommended for publication by M. Botsch.
Available online 15 November 2014

Keywords:
Remeshing
Connectivity regularization
Feature edges
Triangular mesh
2-manifold
Closed mesh

ABSTRACT

567-meshes are a new type of closed and 2-manifold triangular meshes introduced by Aghdaii et al. in 2012 [1]. Vertices with valence less than 5 or greater than 7 are problematic for many mesh processing tasks such as edge collapse or surface subdivision. However, vertices with valence equal to 6 everywhere is most often impossible due to either surface topology or surface feature preservation, that is why 567-meshes are particularly of interest.

This paper proposes a 567-remeshing algorithm that locally retriangulates the mesh considering vertex valence, vertex budget and mesh fidelity as a whole. This algorithm also offers the possibility to preserve a set of feature edges during remeshing. This results in a framework capable of low budget 567-remeshing where remeshed models have a much higher fidelity to the original surface compared to the state of the art.

As applications of this work, we demonstrate that our remeshing improves the performances of mesh regularization and mesh connectivity compression.

1. Introduction

Non-regular triangular meshes are common place and their low and high valence vertices raise several issues. For instance, Aghdaii et al. [1] mention that valence-3 vertices may cause an edge collapse to generate a non-manifold mesh and that high valence vertices can lead to visible artifacts during mesh subdivision (e.g. for the butterfly scheme [2]). In addition, irregular valences, especially high ones, result in irregular sampling when applying either Laplacian or angle-based smoothing [3,4]. In particular, Surazhsky and Gotsman [3] noticed that edges whose two vertices have a valence greater than 7 (resp. smaller than 5) are made longer (resp. shorter) after an angle-based smoothing. The same authors [4] also mention that angle-based smoothing produces less inverted elements when the mesh is close to regular, which effectively happens for 567-*meshes*. 567-meshes are closed and 2-manifold triangular meshes, whose vertex valence is either 5, 6 or 7 [1]. In the remainder of this paper, k^- and l^+ denote a vertex with a valence, respectively, strictly less than k and strictly greater than l.

A regular mesh has faces only of the same degree and vertices only of the same valence. Completely regular (closed) meshes with only valence-6 vertices exist only for genus $g=1$ [5]. It is not possible to get a completely regular closed manifold of genus 0 [6]. In addition, either regular or semi-regular remeshing algorithms usually need a costly 2D (global) parametrization [7] which generally introduces some distortion.

Highly regular meshes are necessary for engineers performing numerical simulations [3]. Many authors [1,3,8,9] have developed (complex) strategies to either drastically reduce the number of non-regular vertices or at least avoid too irregular ones (5^- and 7^+). 567-*remeshing* algorithm [1] falls in the later category and has gain particular attention since it can resolve all issues mentioned in the first paragraph. In particular, a 567-remeshing step applied before a mesh smoothing greatly increases the final quality of the remeshed model. Note that 567-remeshed models could benefit from a specific data structure for representing adjacency relationships with constant-size buffers instead of linked-lists.

However, there is actually a costly counterpart with 567-remeshing: the decrease in the amplitude of valence irregularity is offset by an increase in the number of irregular and regular vertices. That is due by essence to *vertex split* operations that add vertices and to a theorem [1] that states that a valence $d>7$ vertex can be replaced by $\lfloor (d-2)/3 \rfloor - 1$ valence-7 vertices plus a valence 5 or 6 or 7 vertex, while incrementing the valence of $2(\lfloor (d-2)/3 \rfloor - 1)$ one-ring vertices by at most one. The current reference 567-remeshing algorithm [1] does a 1–9 triangle subdivision (see Fig. 1 (b)) before applying vertex splits to avoid creating a new valence-7^+ vertex. The resulting 567-remeshed models have their number of faces multiplied by about 10 before the algorithm' vertex removal step. And because the vertex removal step must preserve the 567 property and the fidelity to

* Corresponding author. Tel.: +33 4 72 43 13 38; fax: +33 4 72 43 15 36.
E-mail address: vincent.vidal@liris.cnrs.fr (V. Vidal).

http://dx.doi.org/10.1016/j.cag.2014.10.004
0097-8493/

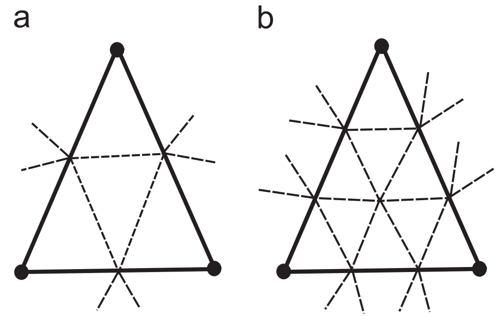

Fig. 1. Local triangle subdivision schemes whose the 1–9 scheme used in [1]: (a) 1–4 subdivision; (b) 1–9 subdivision.

the initial 567-remeshed model, it rarely counterbalances all added vertices, which thus becomes a severe issue when processing large meshes.

This paper addresses the vertex budget issue of the 567-remeshing introduced in [1] and proposes a framework also capable of controlling the geometric error and preserving mesh features (e.g. feature edges) during all the 567-remeshing operations. Our main contributions are the following

- *New local strategies for the removal of valence* 5^- *and* 7^+ *vertices*: We only consider local remeshing strategies (no global triangle subdivision) and we minimize the valence potential increase (see Eq. (1)) when we remove irregular vertices, by greedily selecting the local configuration associated with less valence potential increase.
- *Better control of the mesh quality*: Additionally to the vertex valences, the vertex budget and the fidelity to the original surface are monitored during the removal of valence 5^- or 7^+ vertices. An additional criterion, the triangle equilateralness, is also taken into account during the final mesh enhancement steps.
- *Preservation of lines of feature edges during all 567-remeshing operations*: Our framework offers the possibility to preserve a set of feature edges during 567-remeshing. It considerably improves the fidelity to the original surface, even for non-CAD models.

2. Related work

This section presents some works related to vertex valence optimization and the Aghdaii et al. [1] 567-remeshing algorithm published recently.

2.1. Minimal number of valence 5 and 7 vertices

When dealing with vertex regularization, it is a good start to know what is the theoretically minimal number of irregular vertices for 567-triangular models of closed 2-manifold meshes of arbitrary genus. Using the Euler characteristic for a 567-remeshed model we get $n_5 - n_7 = 12(1-g)$ [1] where g is the genus of the mesh and n_i denotes the number of valence-i vertices. It means that, when considering only topological aspects, the minimal number of valence-5 and valence-7 vertices for a *genus 0* closed 2-manifold mesh is 12 and 0 (the icosahedron), it is 0 and 0 for *genus* $g=1$ (the torus) and 0 and $12(g-1)$ for *genus* $g>1$ ($g=2$: we are able to get a eight mesh with 0 and 12; $g>2$: by cutting an handle along a circle that does not pass through any vertex and then reassembling like in [5]). These values of minimum number of valence-5 and valence-7 vertices may help for identifying the potential decrease of the number of irregular vertices within a 567-remeshed model. However, care should be taken to preserve the geometric fidelity to original surface, since each topological operation may potentially degrade it.

2.2. Connectivity regularization

Regularizing the connectivity of a given 2-manifold mesh M usually means minimizing the following functional [3]:

$$E(M) = \sum_{v \in M} \left(d(v) - d_{opt}(v)\right)^2 \quad (1)$$

where $d(v)$ is the valence of a vertex v and $d_{opt}(v)$ its optimal/ideal valence (6 for closed triangular 2-manifold meshes). This functional owns many local minima and is thus hard to globally minimize. Simulated annealing could be used for minimizing such a function, but its convergence towards a good local minima is too slow for practical purpose. In addition, the more local topological operations are applied on the mesh, the more its intrinsic geometrical features are lost. That is why an approach that uses the less as possible local operations to decrease the energy should be used. Surazhsky and Gotsman [3] have proposed an interesting strategy to further minimize the energy $E(M)$ starting from a local minima, by flipping towards each other edges made of one valence-6^- vertex and one valence-6^+ vertex up to they allow an additional energy decreasing flip. However, their algorithm still performs too many intermediate local modifications and thus degrades significantly the geometry. Moreover, since they forbid too degrading local operations, there is no guaranty that a good minima of the energy $E(M)$ is reached. A solution to better preserve the original geometry can be to severely limit the geometric error introduced during all local modifications such as in [8], but the resulting minimum of energy $E(M)$ is worst.

567-remeshing [1,10,11] has an advantage over general regularization algorithm such as the one presented in the last paragraph: the elimination of valence 3, 4 and 7^+ is a simpler problem than the elimination of all valence $\neq 6$ vertices. In particular, valence 3 and 4 vertices can be treated by a local retriangulation scheme (without geometric error), and valence-7^+ vertices can be tackled by well-chosen vertex splits. It means that a 567 regularization (local) is faster than the 6 regularization (global). In addition, the 567 regularization gives guarantee that no more vertex valence is out of 567, while the 6 regularization cannot provide this kind of guarantee. 567 regularization methods thus appear among the best connectivity regularization techniques, because they offer efficiency (running time and mesh quality) and guarantees.

In the following sections "*567 preserving*" means that it does not introduce neither any new valence-5^- nor any new valence-7^+ vertex.

2.3. Aghdaii et al. 567-remeshing algorithm

Aghdaii et al. [1] have introduced the concept of 567-meshes and proposed the following 567-remeshing algorithm:

1. *Firstly, all valence-*5^- *vertices are eliminated* without introducing geometrical error through the use of local retriangulation schemes (see Fig. 2). This step increases the valence of some vertices within the two-ring neighborhood of a former valence-5^- vertex. It can therefore add new valence-7^+ vertices, and that is why this step is done before the valence-7^+ vertex removal.
2. *Secondly*, *all valence* $d>7$ *vertices are eliminated* through a 1–9 triangular mesh subdivision (see Fig. 1) followed by $\lfloor (d-2)/3 \rfloor - 1$

vertex splits. Note that during vertex splits, added vertices can have their position set to either the former valence-7^+ vertex position (*degenerate case*) or to a slightly different one using the direction towards the centroid of their one-ring vertices. The latter choice will inevitably introduce geometric error in curved areas.

3. *Thirdly, a 567 preserving decimation is done* to attempt to reach the initial vertex budget using edge collapses. The order of edge contractions is based on the QEM error metric [12]. All edge-contractions must preserve the 567 property and thus only a limited number of edges can be collapsed. To increase the number of collapsed edges, the decimation algorithm alternates between 567 edge contractions and 567 edge flips. To monitor the introduced local error, a threshold on the cosine of the dihedral angle is used to authorize edges to be flipped. In practice, this decimation step only removes a small number of the vertices added during the two first steps, except if the selected threshold is permissive, which means more geometric error.

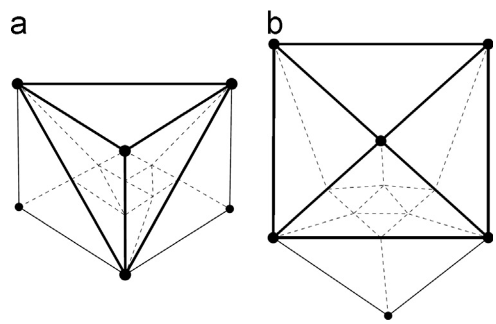

Fig. 2. Local retriangulation schemes to remove either (a) valence-3 or (b) valence-4 vertices. Solid lines represent the edges of the original mesh while dashed lines are the new edges added by the local retriangulation. Images are courtesy of [1]: (a) Valence-3 removal; (b) valence-4 removal.

4. *A mesh quality enhancement* concludes the algorithm. It is based on a Laplacian smoothing followed by a back projection of the points onto the surface of the original mesh.

As can be seen this 567-remeshing algorithm suffers from two problems. Firstly, it introduces too many new triangles during the removal of valence-7^+ vertices, triangles which are difficult to eliminate during the decimation step. Secondly, it lacks of a precise monitoring of the fidelity to the original surface.

The main idea of our approach is to deal with the two aforementioned issues. Firstly, our framework limits the number of added triangles, in particular during the removal of valence-7^+ vertices. Secondly, our framework allows to monitor the mesh fidelity to the initial mesh surface. To get a high fidelity, our framework adds new geometries only when needed during the valence 5^- and 7^+ vertices elimination, and it favors local remeshing operations that do not introduce geometric error. To go a step further in the geometric fidelity, our framework is also capable of dealing with lines of features edges, to preserve them during all remeshing steps.

2.4. Other work on 567-remeshing

Li et al. [10] have proposed some (topological) edition operations to relocate, remove or generate valence 5 and 7 vertices according to user needs. Note that their operators use pair or triple of irregular vertices and do not need these vertices to be adjacent. Their method cannot preserve sharp features and could be used at some point to extend Surazhsky and Gotsman [3] work on mesh connectivity regularization.

Yan and Wonka [11] are able to produce 567 remeshed models using Maximal Poisson-disk Sampling (MPS). They are able to get minimal angle guarantees for uniform sampling only at the cost of a sharp increase of the vertex budget if the input mesh sampling was less dense than required by the density function guiding the

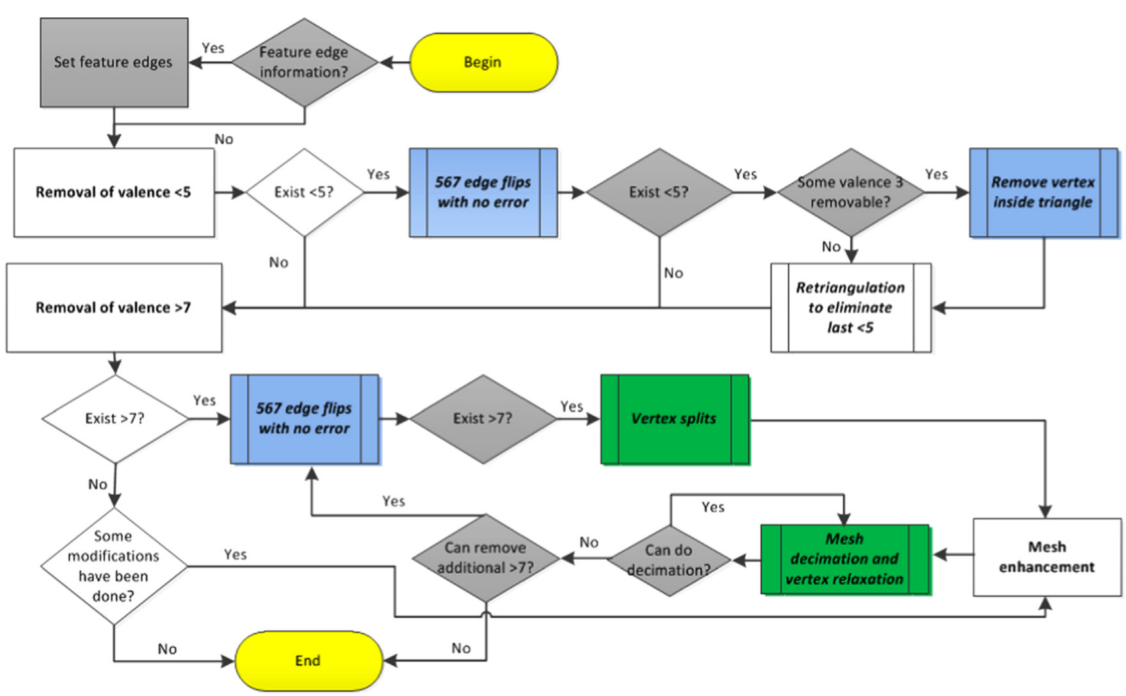

Fig. 3. Diagram of our 567-remeshing algorithm. Our main contributions, compared to Aghdaii et al. [1], have been highlighted in this scheme: new part and tests are painted in dark gray, and new (resp. modified) sub-parts are painted in blue (resp. green). Note that all remeshing (sub-)parts of our algorithm preserve feature edges, which is one of the most important contribution of our work. The mesh enhancement step, in particular the mesh decimation, is guided by the functional $U(M)$ (see Eq. (2)). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

sampling. Their method can preserve sharp features, but is not able to provide a pure 567-remeshed model in that case.

3. Our framework

3.1. Overview of our relaxed 567-remeshing algorithm

Our 567-remeshing framework, presented in Fig. 3, contains the following steps:

1. optional: feature edge detection;
2. elimination of valence-5^- vertices without introducing geometrical error;
3. elimination of valence-7^+ vertices: the choice to introduce either some geometric error or some degenerated triangles is set as a parameter of our framework;
4. final mesh quality improvement: local remeshing operations are applied both to attempt to reach the initial vertex budget and to improve the quality of mesh triangles without giving up the fidelity to the initial surface. Afterwards, if some remaining valence-7^+ vertices can now be eliminated, the algorithm jumps to step 3. Otherwise the algorithm terminates.

3.2. Feature lines extraction (optional)

The feature line extraction step is not a contribution of this paper, therefore we used a simple dihedral angle based feature line extraction, regularized by a graphical model to favor sets of almost aligned feature edges, as proposed in [13]. Without giving details, the regularization model is isotropic for normal edges in order to remove isolated feature edges, while the regularization of feature edges is anisotropic, along the direction of local feature edges to fill in some "holes" in lines of feature edges.

It is worth noting that this step labels all mesh edges either as feature or as non-feature. These labels will then be taken into account in the local remeshing operations:

1. to avoid removing current lines of feature edges;
2. to update feature edge information consequently: a feature edge split will ensure that the two edges replacing the old one are feature edges; a corner vertex is not authorized to move its position during all remeshing operations. A corner has either 3 incident feature edges or 2 which are not aligned.

3.3. Removing low-valence vertices

For all valence 3 and 4 vertices to eliminate, the strategy outlined in Fig. 3 is applied. For each such low-valence vertex, the first step is to apply as many as possible 567 preserving edge flips that do not neither introduce geometrical error, nor break an existing feature edge (see Fig. 4). The order of edge flips is guided by the greedy minimization of valence potential functional presented in Eq. (1). The next step, dedicated to remaining valence-3 vertices, is to remove such a vertex if its 3 neighboring vertices have a valence strictly greater than 5, and if its removal does not introduce geometrical error (see Fig. 5). Finally, if a low-valence vertex is still present, a local retriangulation scheme is applied (see Fig. 2), in such a way that it minimizes the valence potential increase. For that purpose, the functional presented in Eq. (1) is used to select the best central edge (resp. central triangle) for valence-3 (resp. valence-4) retriangulation.

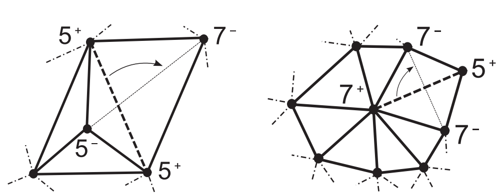

Fig. 4. Contrary to Aghdaii et al. [1], our framework does 567 preserving edge flips before applying any local retriangulation scheme to limit the number of added vertices.

3.4. Removing high-valence vertices

For all valence-7^+ vertices to eliminate, the strategy outlined in Fig. 3 is applied. For each such high-valence vertex, the first step is to apply as many as possible 567 preserving edge flips that do not neither introduce geometrical error, nor break an existing feature edge (see Fig. 4). The order of edge flips associated with a valence-7^+ vertex is guided by the greedy minimization of valence potential functional presented in Eq. (1): the 567 preserving edge flip associated with the largest energy decrease, regarding only the 4 vertices taking part in the flip, is attempted first. The second step, for the remaining valence $d > 7$ vertices, is to apply at most $\lfloor (d-2)/3 \rfloor - 1$ vertex splits. Contrary to Aghdaii et al. [1] algorithm, our method is relaxed, namely it may not be able to apply as many vertex splits as needed to get a completely 567 remeshed model. The reasons are as follows: first of all, the preservation of feature edges may severely restrict the possible vertex split operations, if no degenerated triangles are allowed; secondly, to make our algorithm convergent, the vertex split operation is forbidden for a valence $d > 7$ vertex if this transforms a neighboring vertex with valence $d-1$ to d.

The removal of high-valence vertices via successive vertex splits is thus a highly tailored part of our algorithm. For each vertex split of a valence $d > 7$ vertex, its adjacent feature edges and normal edges are analyzed in order to select the best set of consecutive normal edges such that the valence of the 2 neighboring vertices which increases by 1 is strictly less than d. The vertex splits are tried first in area with more consecutive normal edges (see Fig. 6).

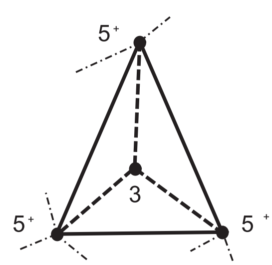

Fig. 5. Our framework try to remove valence-3 vertex when edge flips (see Fig. 4) on front edge cannot be applied. Of course, this removal is authorized only if no geometric error is introduced.

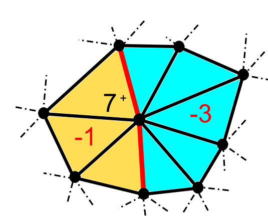

Fig. 6. In the presence of feature edges (here, the 2 almost vertical edges), the first vertex split operation is attempted firstly in the triangle strip where the potential valence loss is higher. Here, in the right triangle strip the potential valence loss is in-between −3 and 0 depending on neighboring vertex valences. In the left triangle strip, only a potential valence loss of -1 can be reached. Therefore the right triangle strip is selected to apply a vertex split. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

4. Mesh enhancement

The mesh enhancement step is needed both to reach the initial vertex budget and to improve the vertex positioning. Indeed, even if we monitor the vertex budget, the removal of irregular vertices usually adds many new vertices and introduces many ill-positioned vertices due to vertex split operations. Therefore, we propose a mesh enhancement step, which alternates vertex decimation and vertex relaxation up to no more decimation is neither needed nor possible.

Up to this section, the used criteria for local remeshing were only related to a combination of *the fidelity to the original mesh*, of *the vertex budget* (compactness), and of *the vertex valence potential*. Indeed, our main concerns were to minimize geometric error and add as less as possible new geometries to the remeshed model, while minimizing the valence potential increase. From now, an additional criterion is taken into account, *the triangle equilateralness*, to better monitor the quality of the remeshed models, especially for the decimation step that tends to introduce many flat triangles without this criteria. All these criteria can be combined within an energy functional $U(M)$ and we use the one presented in [8]:

$$U(M) = \lambda_f \sum_{t \in M} Vol(t, M) + \lambda_p \sum_{v \in M} 1 + \lambda_v \sum_{v \in M} \left(d(v) - d_{opt}(v)\right)^2 + \lambda_s \sum_{t \in M} Equi(t) \quad (2)$$

Here, λ_f, λ_p, λ_v, and λ_s are positive scalar weights which monitor the family of mesh qualities. $Vol(t, M)$ is the volume error between the triangle t and the initial mesh surface, and $Equi(t)$ is a measure of equilateralness which reaches its minimum (resp. maximum) when t is an equilateral triangle (resp. a flat triangle). The respective label subscripts f, p, v, and s denote data fidelity, vertex penalty, valence quality, and shape quality, respectively.

4.1. Mesh decimation and mesh quality improvement

A 567-remeshed model with many more vertices compared with the initial mesh model is not satisfying, especially for large meshes. Therefore, our framework proposes, as Aghdaii et al. [1], to decimate the mesh up to either obtain a 567-mesh with the same vertex budget as the initial mesh model or up to no more *567 preserving* topological operation can be applied. However, contrary to Aghdaii et al. [1], our framework tries to optimize the mesh quality during the decimation. In particular, if the energy functional (see Eq. (2)) increases when a 567 preserving topological operation is applied, we do not apply it. Allowing only energy decrease is a good way for monitoring mesh quality from a global point of view.

Table 1
Some statistics on our and Aghdaii et al.' [1] venus 567-remeshed model: irregular vertex valences, Hausdorff distance computed by the Metro tool [15], and running time (not for the same PC).

Model	Valence (%)			Error (10^{-3})	Time (s)
	$\neq 6$	5^-	7^+		
venus-ours	46.8	0	0.6	9.6	1
venus-[1]	64.4	0	0.0	20.0	4

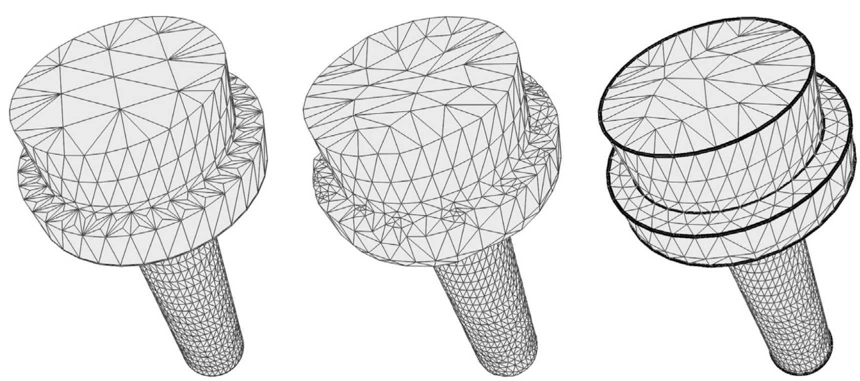

Fig. 7. Results obtained with our 567-remeshing method on the screw model. From left to right: Initial model, 567-remeshed model without feature edge preservation, and 567-remeshed model with feature edge preservation. As can be seen, feature edge preservation is crucial.

Table 2
Various statistics on initial and our 567-remeshed models (**in bold font**) with feature edge preservation except in the presence of *no-f* suffix: number of vertices, irregular vertex valences, mean minimal angle, mean maximal angle, Hausdorff distance computed by the Metro tool [15], running time of our method and result of a compression algorithm (valence driven, Alliez and Desbrun [16]) showing bpv (bit per vertex) for connectivity and geometry. For all these models valence-7^+ vertices are totally eliminated by our method when feature edges are not preserved.

Model	#V (K)	Valence (%)			Amin (deg)	Amax (deg)	Error (10^{-3})	Time (s)	Compression (bpv)	
		$\neq 6$	5^-	7^+					Conn	Geom
venus	0.7–**1.4**	64.4–**46.8**	9.3–**0**	10.4–**0.6**	36.8–**38.8**	86.6–**85.0**	9.65	1	2.39–**1.54**	20.03–**20.55**
genus3	0.8–**1.2**	53.5–**48.8**	3.8–**0**	5.0–**0.6**	26.5–**31.8**	105.6–**95.5**	6.49	1	2.13–**1.70**	20.31–**19.80**
screw	1.2–**1.2**	47.0–**25.3**	6.8–**0**	9.2–**0.6**	25.1–**29.0**	95.4–**92.5**	4.43	1	2.33–**1.22**	3.24–**3.67**
triceratops	2.8–**2.8**	59.3–**39.7**	0.07–**0**	4.8–**0.1**	31.1–**34.5**	92.7–**88.4**	4.09	2	1.89–**1.44**	2.60–**3.54**
cow	2.9–**4.7**	53.3–**48.8**	10.3–**0**	2.7–**0.8**	30.2–**34.8**	93.7–**90.1**	4.12	7	2.05–**1.63**	20.41–**18.40**
joint	3.0–**3.0**	20.2–**27.1**	5.5–**0**	4.6–**0.0**	11.2–**15.6**	98.2–**95.5**	3.20	2	1.82–**1.37**	16.05–**0.72**
egea	8.3–**14.8**	74.9–**38.9**	17.1–**0**	16.4–**0.3**	34.7–**40.8**	93.5–**84.2**	2.68	97	2.72–**1.41**	17.83–**15.57**
egea-*no-f*	8.3–**10.4**	74.9–**45.7**	17.1–**0**	16.4–**0.0**	34.7–**40.8**	93.5–**83.6**	3.74	26	2.72–**1.48**	17.83–**18.87**
carter	20.9–**32.9**	76.6–**47.2**	17.8–**0**	16.0–**0.7**	32.1–**36.7**	98.0–**90.7**	5.28	483	2.59–**1.55**	18.34–**15.60**
carter-*no-f*	20.9–**30.7**	76.6–**50.6**	17.8–**0**	16.0–**0.0**	32.1–**39.2**	98.0–**85.5**	5.10	43	2.59–**1.54**	18.34–**17.81**

The mesh decimation algorithm is quite simple: it alternates *567 preserving edge collapses* and *567 preserving edge flips* up to either reach the initial vertex budget or no more topological operation is allowed. Edge collapses and edge flips ordering is managed via a priority queue whose priority is simply the energy decrease of $U(M)$ (to interpret as a mesh enhancement) due to the local operation. One could imagine to flip a set of edges that will maximize the number of allowed edge collapses. However this combinatoric problem is intractable and the chosen heuristic works well in practice.

4.2. Vertex relaxation and back projection

After the mesh decimation step, a vertex relaxation is needed to improve the overall vertex positioning that may have been degraded during the 567-remeshing steps, especially during the vertex splits. For the vertex relaxation, we use Laplacian smoothing minimizing the local area dispersion [14]. It experimentally both improves the isotropic distribution of vertices and authorizes more decimation steps afterwards, while introducing less geometrical error than the usual Laplacian smoothing used by Aghdaii et al. [1]. Each vertex relaxation is followed by a back projection of vertex positions onto the original mesh surface. This reduces the small shrinkage effect due to vertex relaxation.

Mesh enhancement step ends when no more decimation is possible after a vertex relaxation.

5. Experimental results

In order to demonstrate the efficiency of our 567-remeshing framework, we applied it to several mesh models with irregular vertices, which contain both smooth parts and sharp features. Results are presented in Figs. 7, 9 and 10 and in Table 2. The presented results have been obtained on an Intel Core I7-2760QM (2.4 GHz) with 8 GB RAM .

5.1. Analysis of our results

Table 2 shows that our relaxed 567-remeshing framework is able to eliminate all valence-5^- vertices, and to eliminate most of valence-7^+ vertices, even when feature edges are preserved. Even very high valence (e.g. ≥ 50) can be processed by our method, as can be seen for the sphere model (see Fig. 10). When feature edges do not need to be preserved, all valence-7^+ vertices are removed by our method for presented models in Table 2. Our algorithm performs well even when the model is very hard to regularize. Indeed, the carter model from Table 2 (see Fig. 10) is very irregular and slightly noisy. An additional interesting point when feature edges are preserved is that, experimentally, the few remaining valence-7^+ vertices are located at corners which are the desired locations, since corners usually need to be preserved. We experimentally observe that the 567 preserving edge flips mainly reduce the maximal number of triangles and the introduced geometrical distortion. When this heuristic is combined with 567 preserving edge collapses during the vertex decimation step, it significantly reduces the number of irregular vertices of 10.2% on average of the total vertex count (from 2.9% for genus3 model to 21.3% for screw model).

Our algorithm running times are more or less related to the complexity of regularizing the input mesh. The harder the regularization is, the larger the number of edge collapses and edge flips is, which results in a more time consuming decimation step. The algorithm running times degrade when a lot of feature edges are present. For instance, for the carter model, the running time is 11 times smaller when no feature edge are extracted, and the decimation counts for between 45% and 83% of total running times.

To illustrate the mesh quality control during all 567-remeshing steps, average triangle minimum and maximum angles are presented in Table 2. The closer to 60° the mean minimum and maximum angles are, the better are the results. Our 567-remeshing method always improves averaged minimum and maximum angles when regularizing vertex valences.

To evaluate the surface fidelity of the remeshed models, the Hausdorff distance obtained using the Metro tool [15] is presented in Table 2. According to that distance, the geometric error introduced by our method is small. If the end-user desires either less geometric error or more decimation, our framework can give more or less weight to the surface fidelity via the parameter λ_f of $U(M)$ (see Eq. (2)) and thus limits or favors edge collapses during the decimation step. In our experimental results, we found a good trade-off between the fidelity to the original surface and the

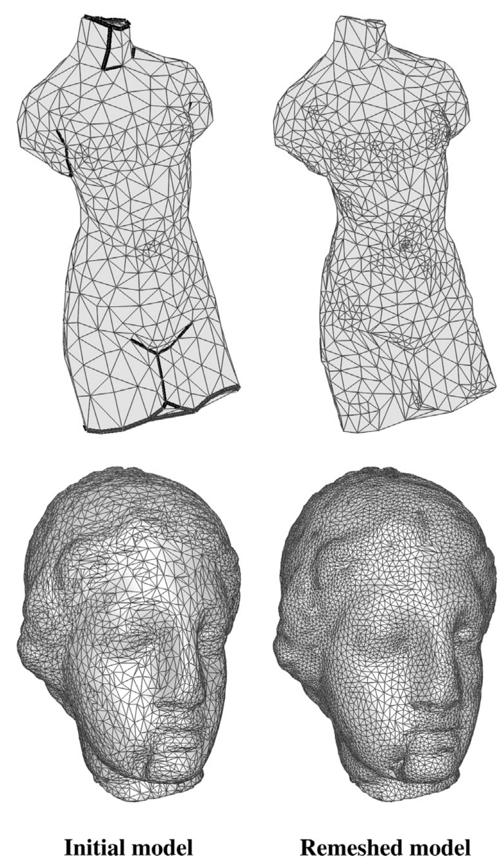

Fig. 9. Results obtained with our 567-remeshing method. From top to bottom: Venus and egea models. Feature edges are outlined.

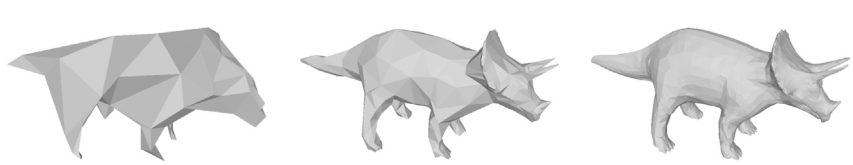

Fig. 8. Results obtained with the progressive compression algorithm [17] on our triceratops 567-remeshed model. From left to right, coarsest, medium and highest mesh resolutions are presented.

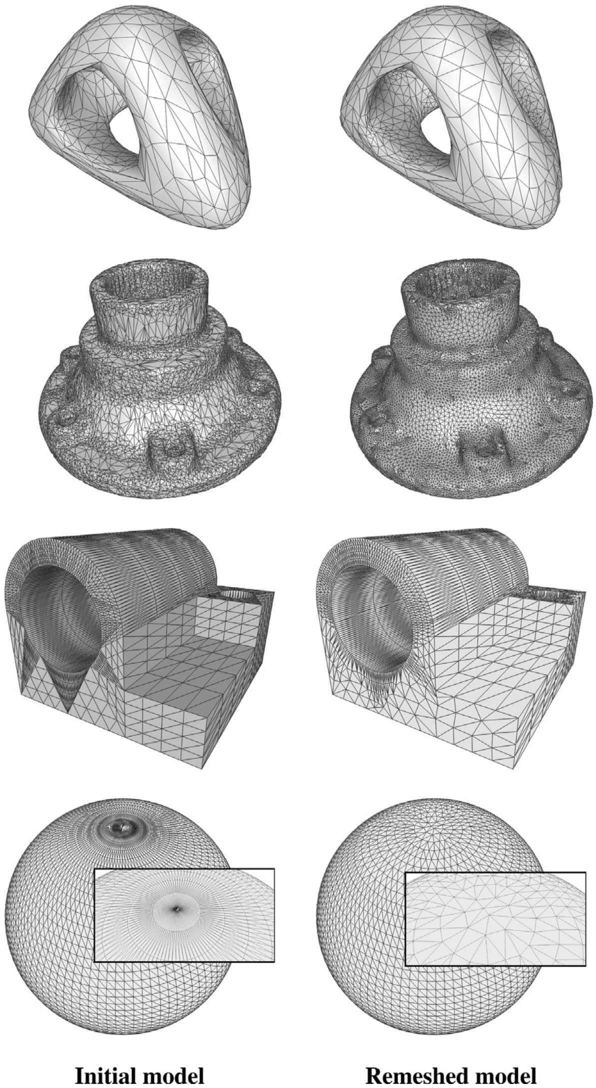

Fig. 10. Results obtained with our 567-remeshing method. From top to bottom: genus3, carter, joint and sphere models.

Table 3
Some statistics on our and Yan and Wonka [11] 567 remeshed models (trice stands for triceratops): number of vertices at the beginning and at the end of the remeshing, Hausdorff distance and the root mean square distance between remeshing and the input mesh (% of the bounding box diagonal), % of bad min angles and % of 567 vertices. Best result is highlighted with bold font.

Model	$\#V_{in}$ (K)	$\#V_{out}$ (K)	Error (10^{-3})	RMS (10^{-4})	$\theta^{min}_{<30^\circ}$ (%)	v_{567} (%)
trice-ours	2.8	**2.8**	**4.1**	**3.1**	34.3	99.9
trice-[11]	2.8	9.0	4.6	6.2	**0**	**100.0**
joint-ours	3.0	**3.0**	**3.2**	**1.0**	86.8	**100.0**
joint-[11]	0.2	3.2	3.7	5.6	**0**	99.4

allowed decimation to reach the original vertex budget with the following parameter values: $\lambda_f = 10^5$, $\lambda_p = 1$, $\lambda_v = 0.5$, and $\lambda_s = 1$.

5.2. Comparison with previous work

Main author from [1] does not provide us his models to date and we thus only present comparison for the venus model, model which is visually the same as the one in [1] except for the small area related to the hole closure (at bottom).

On the venus model, the known feature lines are better preserved by our method. Indeed, our method is able to keep the lines of feature edges such as the one located in the underarm area (see Fig. 9), while [1] method is in essence unable to keep that line. Table 1 shows that the Hausdorff distance computed by the Metro tool [15] is smaller for our method. We can also notice that our method running time is smaller than that from Aghdaii et al. [1], mostly due to the fact that they process a global 1–9 subdivision step, which adds many triangles, whose elimination, during the decimation step, is time consuming. We experimentally notice that the maximal number of triangles during our remeshing is at most 2.84 times the number of triangles in the input mesh, while Aghdaii et al.' method multiplies it by about 10. We can thus conclude that our method adds at least about 3.5 times less new triangles during the elimination of valence 5^- or 7^+ vertices than [1]. This is of great benefice when processing large meshes. Finally, Table 1 indicates that our method produces much less irregular vertices, which means that our local valence potential optimization is able to reduce significantly the number of irregular vertices. Aghdaii et al.' method eliminates all valence 5^- or 7^+ vertices, while our method eliminates all valence-5^- vertices, but a few valence-7^+ vertices may remain at corner vertex when feature edges are preserved and no degenerate triangle is authorized. When either no feature edge is preserved or degenerate triangles are authorized, we experimentally always eliminated all valence 5^- or 7^+ vertices .

We also compare our method with a high quality 567-remeshing technique [11] and results are presented in Table 3. Here, the quality is related to the equilateralness of triangles. Remeshed models obtained via the high quality method exhibit a perfect percentage of very good triangles while our method still keeps a too high percentage to be of practical purposes for numerical simulations. However, our method offers a slightly higher fidelity to the input mesh surface in terms of Hausdorff and RMS distances. Note also that the high quality 567-remeshing technique [11] is unable to preserve the initial vertex budget, because it uses a particular density function to sample the mesh surface.

6. Applications

Our 567-remeshing framework might be a good preprocessing step for triangular mesh regularization and for connectivity-based mesh compression.

Table 4
Some statistics on the regularization [8] of venus model: % of 567 vertices, % of bad min angles, % of bad max angles, mean minimal angle, mean maximal angle, and Hausdorff distance computed by the Metro tool [15] (% of the bounding box diagonal).

Model	v_{567} (%)	$\theta^{min}_{<30^\circ}$ (%)	$\theta^{max}_{>105^\circ}$ (%)	Amin (deg)	Amax (deg)	Error (10^{-3})
venus	92.9	19.6	5.6	38.6	83.6	5.62
567-venus	95.6	12.3	4.1	41.8	81.2	9.65

6.1. Mesh regularization

A mesh regularization algorithm attempts to regularize both vertex valences and triangles (to make them as equilateral as possible) while preserving the fidelity to initial surface. The closed venus model (the same as the one in Table 2) has been regularized by a regularization algorithm [8], one time starting from the initial model, another time starting from the 567-remeshed model. Results are presented in Table 4. We notice that better regularization performances are reached for [8] when a 567-remeshing has been applied as a preprocessing step. That is certainly due to the fact that vertex repositioning is more efficient in [8] starting form a 567-remeshed model, than from a highly irregular one. On the one hand, it seems that the problem of regularizing a 567-triangular mesh is simpler than those of regularizing any kind of triangular mesh. On the other hand, the 567-remeshed models generally introduce slightly more geometric error, because high-valence vertices may be needed to preserve high fidelity without introducing degenerated triangles (e.g. at peaks).

6.2. Mesh compression

Valence driven mesh compression methods are more efficient on meshes whose vertex valence is highly regular. Our 567-remeshing algorithm thus seems to be a good pre-processing step when the original connectivity can be lost. We applied a valence driven mesh compression algorithm [16] to our remeshed models. As can be seen in Table 2, our remeshing significantly improves the connectivity encoding efficiency. However, the geometry encoding performance of Alliez and Desbrun [16] can be degraded by our remeshing. Note that geometric prediction schemes adapted to 567 models may improve these results and we let this as future work. Our 567-remeshing method is thus beneficial in terms of connectivity performance whenever it does not increase the number of vertices significantly, and may not be beneficial otherwise. When the number of vertices in the remeshed model is the same as the one of the original model, we can compare the total compression enhancement due to our framework in terms of file size. For instance, the original joint (resp. triceratops) connectivity is encoded into a file of 683 bytes (resp. 670 bytes), while our 567-remeshed model gives 516 bytes (resp. 512 bytes). Other compression methods can balance the connectivity and geometry coding performances such as a progressive mesh compression [17]. Examples of obtained levels of details are illustrated in Fig. 8.

7. Conclusion and future work

In this paper, we demonstrate that 567-remeshing can be conducted without the need of a 1–9 global subdivision step, which is a severe issue of state-of-the-art for large meshes because the algorithm can be run out of memory. We also showed that the decimation step, needed to balance added vertices during 567-remeshing with edge collapses, could be done by also taking into account triangle equilateralness; by doing so, the mesh quality is improved at the end of the decimation process, and thus, only a few smoothing iterations are needed for the final enhancement. We have also successfully introduced the preservation of lines of feature edges during the 567-remeshing, which is crucial for CAD models and more generally for human perception. When too many feature edges are present and no degenerate triangle is authorized, our framework may fail to remove a few valence-7^+ vertices. The obtained results outperform the state-of-the-art in terms of fidelity to the original mesh. The main drawback of our 567-remeshing is that there is no guarantee to get the same vertex budget as the original mesh model had. This is mainly due to our focus on the fidelity to the original mesh. This constraint can be relaxed by giving less weight to the parameter λ_f of $U(M)$ (see Eq. (2)).

As challenging future work, we see two main directions. The first one will be to find a way to decrease the irregular sampling introduced by 567-retriangulation operations without the smoothing which is responsible for most of the final geometric error. The second one will be to improve the location of irregular vertices for improving geometry compression performance.

References

[1] Aghdaii N, Younesy H, Zhang H. 567 Meshes: remeshing and analysis. Comput Graph 2012;36(8):1072–83.
[2] Zorin D, Schröder P, Sweldens W. Interpolating subdivision for meshes with arbitrary topology. In: SIGGRAPH '96 Proceedings of the 23rd annual conference on Computer graphics and interactive techniques; 1996. p. 189–92.
[3] Surazhsky V, Gotsman, C. Explicit surface remeshing. In: Symposium on computational geometry; 2003. p. 20–30.
[4] Surazhsky V, Gotsman C. High quality compatible triangulations. Eng Comput 2004;20(2):147–56.
[5] Akleman E, Chen, J. Regular meshes. In: Proceedings of the 2005 ACM symposium on solid and physical modeling—SPM '05. New York, USA: ACM Press; 2005. p. 213–19. ISBN 1595930159.
[6] Pugh A. Polyhedra: a visual approach. University of California Press; 1976.
[7] Alliez P, Ucelli G, Gotsman C, Attene M. Recent advances in remeshing of surfaces. Springer; 2008.
[8] Vidal V, Wolf C, Dupont F. Combinatorial mesh optimization. Vis Comput 2012;28(5):511–25.
[9] Clark B, Ray N, Jiao X. Surface mesh optimization, adaption, and untangling with high-order accuracy. In: Proceedings of the 21st international meshing roundtable; 2013. p. 385–402.
[10] Li Y, Zhang E, Wonka P. Editing operations for irregular vertices in triangle meshes. ACM Trans Graph (Proc Siggraph Asia) 2010;29(6):153.
[11] Yan DM, Wonka P. Gap processing for adaptive maximal poisson-disk sampling. ACM Trans Graph 2013;32(5) 148:1–148:15.
[12] Garland M, Heckbert P. Surface simplification using quadric error metrics. In: SIGGRAPH'97 Proceedings of the 24th annual conference on Computer graphics and interactive techniques; 1997. p. 209–16.
[13] Vidal V, Wolf C, Dupont F. Robust feature line extraction on CAD triangular meshes. In: International conference on computer graphics theory and applications. Algarve, Portugal; 2011. p. 106–12.
[14] Alliez P, Desbrun M, Meyer M. Efficient surface remeshing by error diffusion. Technical Report, INRIA Sophia Antipolis; 2002.
[15] Cignoni P, Rocchini C, Scopigno R. Metro: measuring error on simplified surfaces. Comput Graph Forum 1998;17(2):167–74.
[16] Alliez P, Desbrun M. Valence-driven connectivity encoding for 3D Meshes. Comput Graph Forum 2001;20(3):480–9.
[17] Lee H, Lavoué G, Dupont F. Rate-distortion optimization for progressive compression of 3D mesh with color attributes. Vis Comput 2011;28(2):137–53.