EUROGRAPHICS '96/ J. Rossignac and F. Sillion (Guest Editors), Blackwell Publishers
© Eurographics Association, 1996

Volume 15. (1996), Number 3

Mesh Simplification

Maria-Elena Algorri and Francis Schmitt

Images Department, Ecole Nationale Supérieure de Télécommunications
46 rue Barrault, 75634 Paris France

Abstract

Mesh simplification is an important stage after surface reconstruction since the models produced can contain a large number of polygons making them difficult to manipulate. In this paper we present a mesh simplification algorithm to reduce the number of vertices in a dense mesh of triangles. The algorithm is based on edge operations that are performed in the inside of independent clusters distributed over the entire mesh. The clusters are well-characterized regions that can successfully accept simplification operations. The simplification operations produce only local transformations on the mesh. This region-based, distributed approach permits to easily track and control the changes in the triangulation and avoids the appearance of particular cases that would require a special handling. The algorithm uses two user-specified parameters to guide the operations. These parameters allow various simplification strategies that are illustrated on several dense triangulations.

Keywords: *mesh simplification, mesh decimation, triangulation, geometric modeling*

1. Introduction

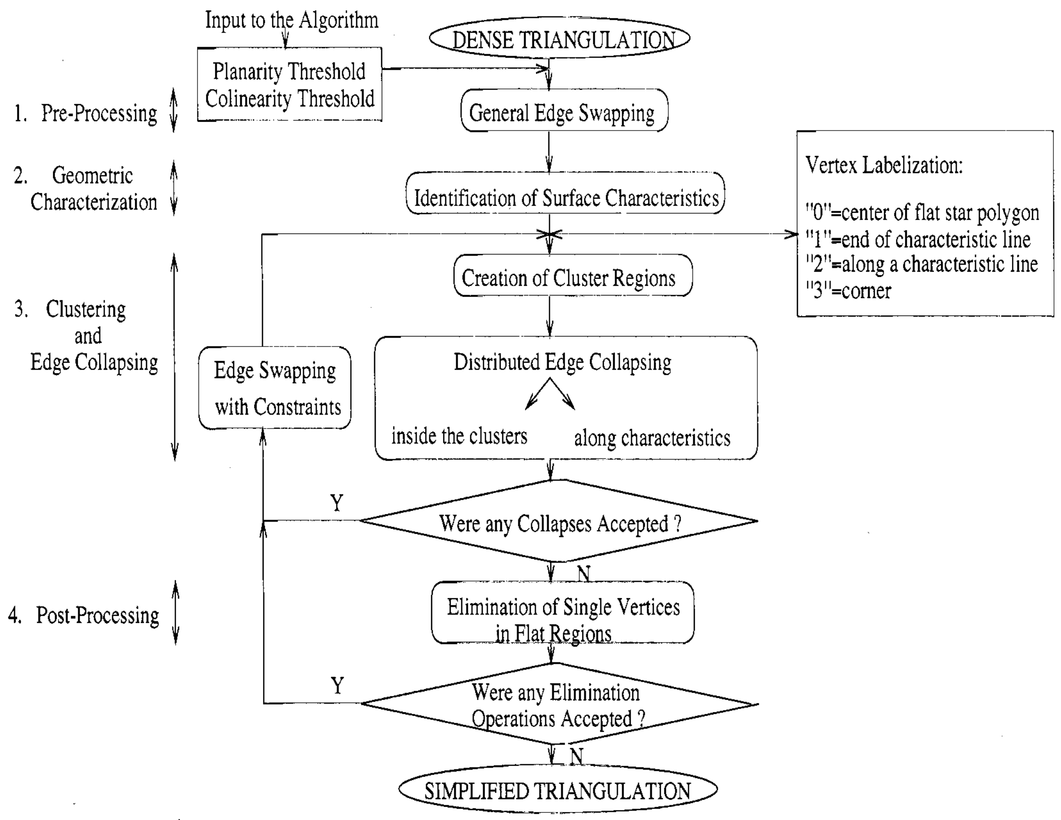
Dense triangulations S' of a surface S can be produced by different techniques of surface reconstruction. Simplifying S' to different levels of detail is a valuable choice when the goal of the reconstruction is easiness of model manipulation or compression of the information, or when the shape to reconstruct is a simple geometrical one which can be represented with a reduced number of triangles. Building coarse but accurate models is also important when reconstructing large databases as those coming from medical or $3D$ laser scanners.

In general, except for extremely simple objects, there is no one ideal solution to the problem of mesh simplification. Rather, it is a user-specified compromise between conciseness of the representation and goodness of fit. Given a user-specified parameter such as an error tolerance in the reconstruction, a planarity-tolerance, a maximum number of triangles in the representation, or a percentage of triangle decimation, a simplification algorithm will then decimate the mesh to attain the specification.

Several authors have put together different methods for simplifying polygonal meshes. Schroeder et al.[7] propose a network filter to reduce the number of triangles in very dense meshes. Their algorithm characterizes the geometry and topology of independent vertices and evaluates a decimation criterion based on the distance of each vertex to the average plane computed from all the in-coming triangles to the vertex. After a vertex and its incoming triangles are deleted, the resulting hole is patched using a local re-triangulation process.

Turk[8] re-tiles meshes in order to create surface models at different levels of detail from an original dense triangulation. The algorithm distributes a new set of vertices over a triangulated surface and creates an intermediate model called mutual tesselation of the surface that contains both the vertices from the original model and the new points that are to become vertices in the re-triangulated surface. The new triangulation is then created by removing each original vertex and locally re-connecting the new vertices in a way that matches the local connectedness of the original surface.

Hoppe et al.[6] propose a complete mesh optimization scheme that improves the fit of S' to the original surface S during the simplification process. The objective is to minimize a compromise function between accuracy and conciseness of the representation. Their approach allows all the edges in the mesh to be candidate for the hierarchical operations of edge collapse, edge split, and edge swap. The algorithm applies the three edge operations (always in that order) to every edge in the mesh, and accepts the first operation that reduces the value of the compromise function. If one of the edge operations is validated for an edge, the edge and its neighboring edges continue to be in an active list of edges to be operated upon. Edges that do not accept any operation are removed from the active list. The algorithm terminates when the active list is empty.



Input to the Algorithm
Planarity Threshold
Colinearity Threshold
DENSE TRIANGULATION
1. Pre-Processing
General Edge Swapping
2. Geometric Characterization
Identification of Surface Characteristics
Vertex Labelization:
"0"=center of flat star polygon
"1"=end of characteristic line
"2"=along a characteristic line
"3"=corner
3. Clustering and Edge Collapsing
Creation of Cluster Regions
Edge Swapping with Constraints
Distributed Edge Collapsing
inside the clusters
along characteristics
Y
Were any Collapses Accepted ?
N
4. Post-Processing
Elimination of Single Vertices in Flat Regions
Y
Were any Elimination Operations Accepted ?
N
SIMPLIFIED TRIANGULATION

Figure 1: *Block diagram of the simplification algorithm.*

Gourdon[3] proposes a simplification algorithm to reduce the number of vertices and facets on irregular meshes which may not be triangulations. He iteratively removes edges in the mesh using a curvature criterion and verifying a set of topological conditions to preserve good properties in the mesh. To avoid shrinking of the mesh or the appearance of degenerated facets, the meshes are regularized by constraining the positions of the new vertices to be parallel to the average plane of their neighbors and towards their barycenter.

We present in this paper a distributed algorithm to simplify a closed, dense triangulation S' to varying levels of detail. The main contributions of our algorithm are:

- It implements a clustering strategy to simplify a dense triangulation as opposed to a global strategy. The re-triangulation operations take place inside independent clusters without propagating through the mesh. As a result, the chain-effects so difficult to track in global implementations are substituted with well-characterized local transformations of the triangulation.
- It implements a region-identification strategy as opposed to an edge selection strategy to perform the simplification operations. That is, rather than identifying independent edges to collapse, the algorithm, identifies suitable regions containing the edges. As a result, we eliminate the appearance of non-foreseen configurations that have to be handled as particular cases.
- It uses two user-specified parameters to guide the simplification. Such parameters allow the algorithm to decimate some regions on the mesh more importantly than others. As a result, the user can decimate triangulations while preserving the regions of high detail, or can uniformly simplify the entire mesh.

2. Overview of the Simplification Algorithm

The simplification algorithm is based on edge operations. and is implemented in four steps shown in Figure 1. It takes on entry a dense triangulation S' and two user specified threshold values, a planarity threshold, and a collinearity threshold (introduced in the next subsections). In a first step we evaluate for swapping the edges of S' to regularize the triangulation. In the second step we characterize the geometry of all the triangle edges and vertices in S' in order to identify important shape features such as regions of high curvature, corners. or characteristic lines that we want to preserve in the simplified

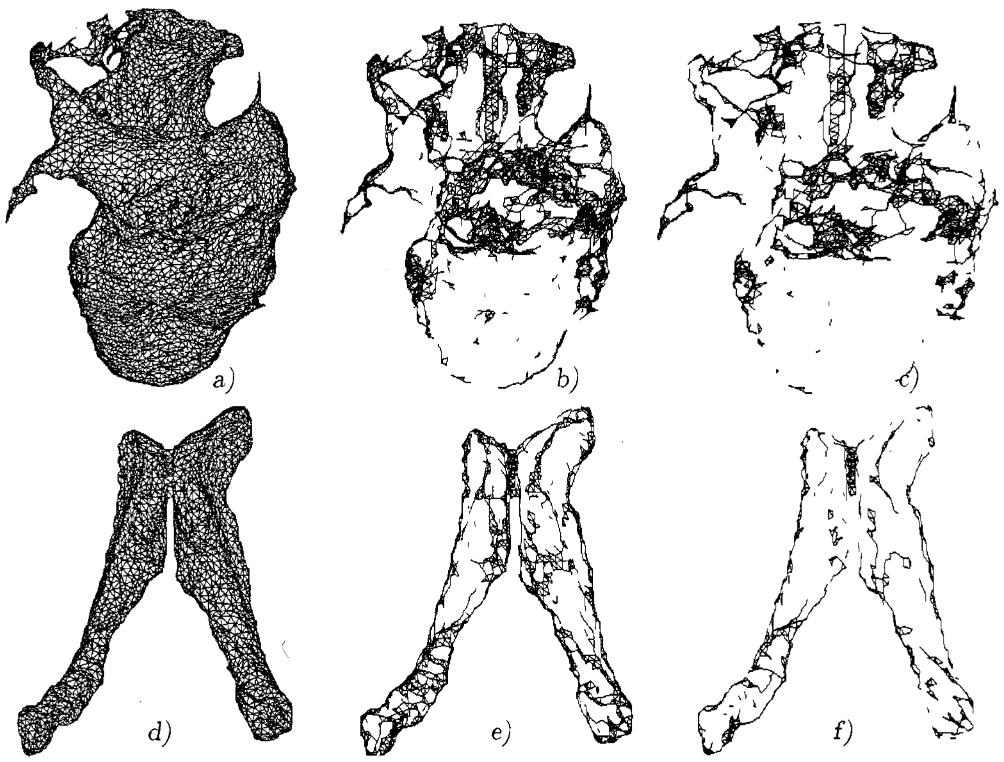


Figure 2: *a) , d) triangulation of a heart (16,404 triangles) and of brain ventricles (7,520 triangles), b), e) (c), f) respectively) feature-edges having dihedral angles larger than 20° (35° respectively).*

mesh. In a third iterative step we identify all the candidate cluster regions containing edges that can be successfully collapsed. We then perform the edge collapse operations over the entire mesh in a distributed way: Every edge collapse takes place strictly inside a cluster region and does not affect the rest of the mesh. The algorithm iterates on the mesh identifying at each pass all the candidate cluster regions and then performing the set of independent edge collapses. Between the iterations we evaluate for swapping all the edges in the triangulation that were not identified as important shape features. In a fourth step we eliminate the isolated vertices that remain in the flat regions of the mesh. The algorithm iterates over the last two steps until no more simplification operations are accepted.

3. General G^1-Continuity Edge Swapping

In general, we have no guarantee that the dense triangulation that is input to the simplification algorithm be smooth, for example that the triangle edges be correctly aligned to well represent the surface characteristics. Therefore, before extracting the characteristics of the surface S' and simplifying the triangulation, we perform a global edge swapping to minimize the small oscillations that might be present in the input mesh. To guide the swapping we use the G^1-continuity criterion defined by Dyn et. al2 which inverts ("swaps") the diagonal of a quadrilateral if the swapping maximizes the continuity of the triangle planes that share an edge of the quadrilateral. This criterion allows an enhancement of the principal directions of S, a more accurate identification of the surface characteristics, and a better identification of planar regions in the mesh[1]. The swapping proceeds as follows: We evaluate the G^1-continuity criterion on every edge by calculating the normal vectors of the two triangles sharing the edge and also the normal vectors of their four neighboring triangles. We then measure the five dihedral angle variations between the pairs of adjacent normal vectors and accept to swap the edge if the mean variation in normal direction is reduced when swapping the edge (we define dihedral angle at an edge as the angle formed by the outward normals of the two triangles sharing the edge).

4. Geometric Characterization

The objective of this step is to identify the important shape characteristics of the surface S and to label the vertices in the triangulation S' that constitute these important characteristics. The identification of shape characteristics is based on a planarity-threshold value specified by the user.

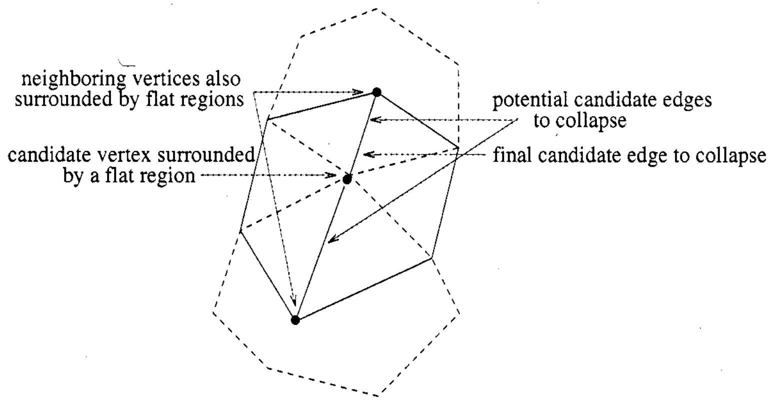

neighboring vertices also surrounded by flat regions
candidate vertex surrounded by a flat region
potential candidate edges to collapse
final candidate edge to collapse

Figure 3: *If several edges around a vertex are candidates to be collapsed, the shortest one is chosen.*

We make one pass over the triangulation where we measure, at each vertex, the dihedral angles between all pairs of adjacent triangles converging at the vertex. If the dihedral angle between any two triangles is larger than the specified planarity-threshold value, their common edge forms part of an interesting shape characteristic to be preserved. We call such an edge a “feature-edge”. We call a set of connected feature-edges, a “characteristic line”. Figure 2a shows a dense triangulation and Figure 2b shows the feature-edges corresponding to a planarity-threshold value of 20’ to be preserved in the triangulation. Figure 2c shows the feature-edges corresponding to a planarity-threshold value of 35°. Figure 2d showsanother dense triangulation and Figures 2e and 2f show the feature-edges corresponding to planarity-threshold values of 20’ and 35°, respectively. As can be seen, the higher the value of the specified planarity-threshold, the more a triangulation will be simplified, since less edges will be considered as representing important features.

We label each vertex according the number of feature-edges adjacent to it. A vertex labeled as “1”, “2”, “3”, or more is a vertex surrounded by exactly one, two, three, or more feature-edges, and corresponds respectively to a termination of a characteristic line (“1”), to a vertex lying along a characteristic line (“2”), or to a corner (“3” or greater). If all pairs of triangles adjacent to a vertex have dihedral angles smaller than the specified planarity-threshold, the vertex is labeled as “0”. We consider a “0” vertex as sorrounded by a planar region.

5. Clustering

After the vertices have been labeled we define “clusters” over the mesh inside of which we will collapse the edges. Clusters consist of the union of two adjacent “0” vertices and the triangles around them.

We first flag all the “0” vertices as candidates for clustering and then identify pairs of adjacent flagged vertices. Their common edge, which is surrounded by a planar region, becomes a candidate for an edge collapse operation. If a flagged vertex has more than one adjacent flagged vertex we choose the nearest one to form the shortest “collapsible” edge, as illustrated in Figure 3. Once an edge is a candidate for a collapse operation, all the vertices forming the polygon around it are no longer valid candidate vertices and are unflagged. The polygons surrounding the candidate edges to be collapsed become “clusters” where the re-triangulation operations take place. The cluster boundaries remain unchanged, rather, it is the edges inside that are operated upon. Figure 4 shows the star polygons that share the edges to collapse and that form the “clusters” for re-triangulation for the two dense surfaces of Figure 2 and for planarity-threshold values of 20’ and 35’. We observe, as expected, that more clusters are selected for simplification when the planarity-threshold increases.

This step generates a list of edges to collapse and terminates when all the candidate clusters have been identified. Our clustering strategy distributes the re-triangulation operations over the triangulation, since no two edges to be collapsed are nor connected, nor shared by the same triangle. This allows for a good control of the re-triangulation operations on the mesh. Since the chain-effects present in global re-triangulation processes are avoided, the local connexity is easy to track and alterations of the mesh topology are easy to check and avoid. But, because of the clustering, not all the collapsible edges whose two vertices have been flagged can be collapsed in a single pass. Furthermore, from the simplification operations result new edges having small dihedral angles that can be further collapsed. Several iterations are then needed to exhaustively re-triangulate all the planar regions. At every new iteration, the clustered polygons become larger and less edges are collapsed. The algorithm is relatively fast, for the two examples illustrated one simplification iteration where 1,000 edges are collapsed (and 2,000 triangles eliminated) takes less than 10s on a Sparc Center with one 80 MHz processor.

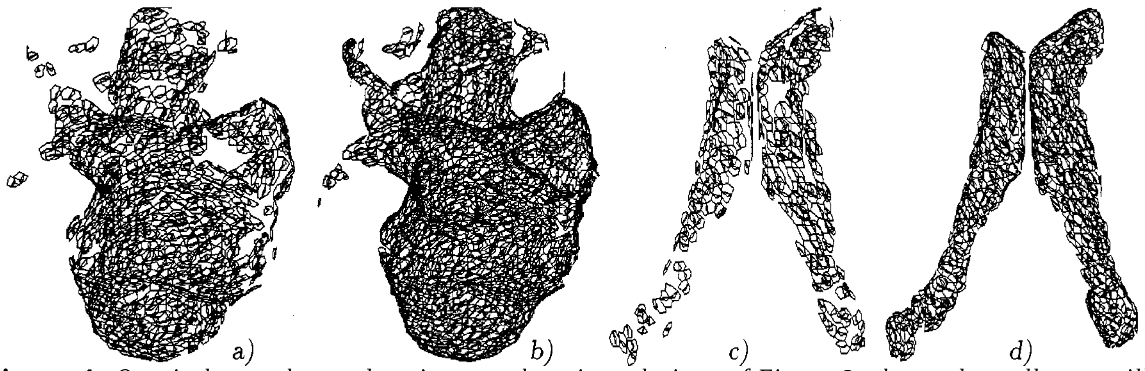


Figure 4: *Quasi-planar clustered regions on the triangulations of Figure 2 where edge collapses will be performed. a), c) with a planarity-threshold value of 20°. b), d) with a planarity-threshold value of* 35°.

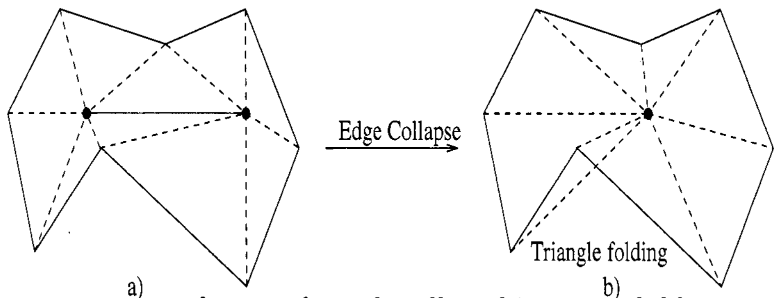


Figure 5: *Foldings may appear when an edge to be collapsed is surrounded by a very concave polygon.*

6. Collapsing the Edges

This step takes the list of the edges to collapse and performs the collapsing operations ensuring that they respect the geometry of the original surface and that they do not create foldings on the mesh.

The re-triangulation step performs three operations as follows: 1) each edge to be collapsed is replaced by a single vertex located at the center of the collapsed edge, 2) the vertices of the star polygon that surrounds the collapsing edge are then connected to the new vertex. A total of one vertex and two triangles are suppressed, 3) the process terminates by up-dating the local neighborhood relations between the remaining triangles.

Before performing the re-triangulation operations we must verify that they will not alter or degenerate the triangulation or that they will not create features that do not exist on the surface S'. One typical alteration of the triangulation occurs when the mesh folds over itself. This happens when the star polygon around the edge to be collapsed is very concave. Then, when the vertices forming the star polygon are reconnected to the central vertex where the edge collapses, two or more triangles inside the polygon might fold one over the other as shown in Figure 5. To avoid this, we first estimate the dihedral angles between adjacent triangles in the star polygon as they would result after the edge collapse has taken place. If these dihedral angles exceed the planarity-threshold value, then the edge collapse operation is rejected. This test avoids the creation of new features in the triangulation that are not present in S' thus being consistent with our strategy of preserving the initially detected features without generating new artificial ones.

An alternative to collapsing the edges inside the clusters is to retriangulate the clusters. Retriangulating the clusters is a valid option, but also a more computationally intensive one. The edge collapse operations require the repositioning of two vertices in a central position and the checking for foldings. Retriangulating the cluster requires carrying out combinatorial tests to find a valid way to re-connect the vertices. In cases where clusters contain concavities or when the planarity-threshold value is very high and the clusters are non-planar, the retriangulation technique can change the topology of the original triangulation more often than the edge collapse operations, necessitating the validation of more sophisticated topological checks.

We have chosen to place the new vertices that are created in the edge collapse operations in the center of the collapsing edge, and to reject the collapse if it generates foldings or new features in the mesh. However, there are other placing alternatives that could result in successful collapses such as placing the new vertex in the kernel region of the collapsing edge from where all the edges of the star polygon are visible. Implementing such alternatives increases the success rate of the collapse operati-

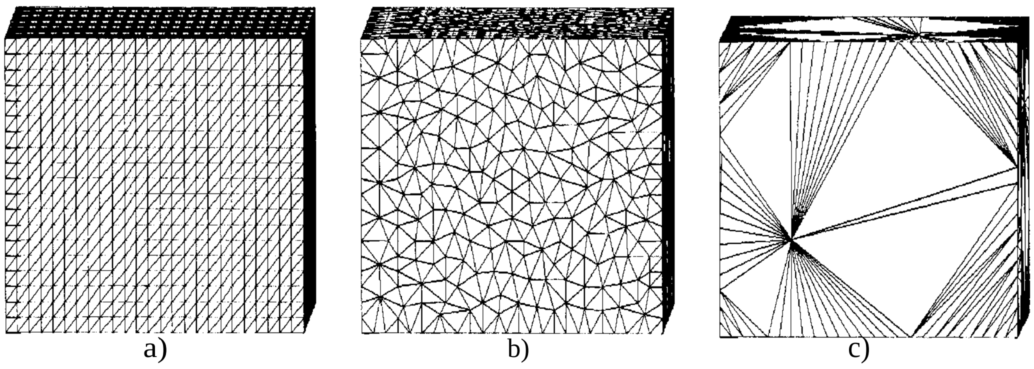


Figure 6: *a) dense triangulation of a cube (11,552 triangles), b) decimation of the triangulation with the basic algorithm by 50% c) decimation until the final step with one remaining vertex per planar region (by 91%).*

ons, but at the cost of increased computation. We preferred to keep a simple collapse strategy since the edge swapping that follows, and the additional iterations, make it likely for the rejected collapse operations to take place in the following iterations.

7. Characteristic Lines and Isolated Vertices

The simplification algorithm as described so far, can simplify almost exhaustively the regions of a dense triangulation that are under the planarity threshold. However, two situations remain which are disregarded by the basic algorithm but that can be further simplified. The first untreated situation is the simplification of characteristic lines. Since the simplification algorithm considers only planar polygons to carry out the edge collapse operations, all the vertices that were originally present on the characteristic lines are still present after the simplification. The second situation comes from the fact that the algorithm needs two planar polygons to form a cluster and perform an edge collapse. After a planar region has been totally simplified there will still be one remaining vertex left over by the last collapse operation that can no longer be removed. Both situations are illustrated by the simplification example of Figure 6. Figure 6 shows the process of simplifying a dense triangulation representing a cube. The original triangulation in Figure 6a contains 11,552 triangles. Figure 6b has been decimated by **50%** and the final decimation result in Figure 6c has been decimated by 91%.

To simplify characteristic lines, we expand the edge collapse operations to consider not only "0" vertices, but also **"2"** vertices surrounded by two feature-edges (vertices lying on a characteristic line). To collapse an edge having a "2" vertex, we compute the cosine of the angle between the two feature-edges reaching the "2" vertex. The cosine value tells how parallel the two feature-edges are, and, if it is smaller than the collinearity-threshold value we simplify the characteristic line by replacing the two feature-edges by a single one.

The collapse operation takes place inside the clustered star polygon around the two chosen adjacent vertices, but, instead of placing the new vertex in the center of the collapsed edge it replaces (as was done when collapsing two "0" vertices), the new vertex is placed directly on the vertex of the collapsed edge that is not the original **"2"** vertex (but that can also be a **"2"** vertex). This means that when we simplify along a characteristic line, we simply slide the original **"2'** vertex to the position of its neighbor vertex along the characteristic line. We only allow **"2"** vertices to slide along characteristic lines in order to preserve unchanged the position of the corners and of the terminations. In Figure 7a the simplification along characteristic lines is illustrated on the cube of Figure 6c with a **99%** decimation of the original triangulation.

The terminal stage of the algorithm eliminates the single vertices left over in planar regions. This stage consists in finding the left over "0" vertices that are surrounded exclusively by vertices labeled "1" or higher and that can no longer be eliminated by our edge collapse technique. We eliminate a left over "0" vertex by identifying its star polygon and re-triangulating it using a recursive loop-splitting procedure described in **7.** This last stage is illustrated in Figure 7b, where the simplification of the cube triangulation is optimal.

Finally, after the isolated vertices have been removed from planar regions, the triangulation is input again into the edge collapse routine to check if any further simplification may occur after a new constrained edge swapping. The algorithm ends when no more simplifications of the mesh can be done under the planarity-threshold and collinearity-threshold constraints.

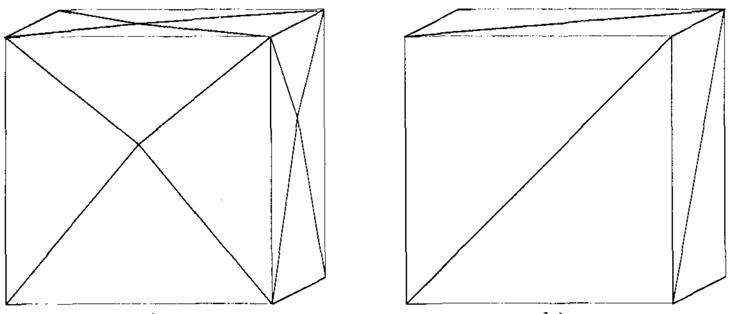


Figure 7: *a) Simplification along characteristic lines on the cube of Figure 6c. b) Elimination of the remaining vertices on planar regions, the decimation is optimal.*

7.1. Constrained Edge Swapping

An important part of the simplification process consists in swapping the edges after every simplification iteration. As in the general edge swapping (section 2.1) we maximize the G^1-continuity criterion to guide the edge swapping operations, but now under the constraint that only non-feature edges are eligible for swapping.

When the region affected by the swapping already includes one or several feature-edges before the swapping, we just perform the swapping tests without considering these feature-edges. It might occur that the swapping operation will diminish the angle of a feature-edge (even under the planarity-threshold value) thus diminishing (even vanishing) the shape feature itself. In this situations we allow the swapping to take place since the swapping operation is used as the main tool for surface regularization. However, we have chosen to continue to consider the feature-edge as so, in order to keep track of the initially extracted characteristic lines.

Edge swapping between the simplification iterations is important to guide the operations to produce a more regular simplification. Regularization of the triangulation is also important to obtain comparable simplification results even if the clusters on the mesh are selected randomly.

8. Simplification Strategy

The type of simplification that can be obtained depends on the values of the two thresholds used during the simplification iterations. Two main strategies can be adopted. One consists in assigning, from the beginning. two very “tolerant’) threshold values. That is, threshold values that are large and will accept a large decimation of triangles. A second strategy is to start the simplification algorithm giving “strict” threshold values to the algorithm such that only very planar regions and very straight edges will be simplified and then to slowly relax the threshold values as the simplification iterations proceed. Other variations of these schemes can be implemented by independently varying only one of both user specified thresholds. Every time we change the planarity-threshold value the geometric characterization of the vertices in the mesh has to be re-calculated. However, changing the collinearity-threshold value only affects which characteristic lines can be simplified, since it only modifies the “straightness” threshold between the feature-edges. We cannot conclude that using one strategy gives better results than the other. The choice depends on the application of the reconstructed surface. When we specify “strict” thresholds in the beginning and slowly increment their values, the surface characteristics are preserved longer and the algorithm makes many more passes over the more planar regions. The simplified triangulation is then a combination of big triangles in very planar regions and smaller triangles in less planar regions. When we specify large “tolerant” angle values from the beginning, the simplification operations take place all over the mesh and the simplified mesh is more regular with triangles of comparable size everywhere. Figure 8 shows the result of simplifying the dense triangulation of the heart using both described strategies. Figure 8a shows the original dense triangulation after the general G^1-continuity swapping. In Figure 8b we show the result of starting the simplification with planarity-threshold and collinearity-threshold values of 40°, and then incrementing them to 55° in Figure 8c, and then to 70’ in Figure 8d. The final decimation of triangles obtained is of 82%. Figures 8e, 8f, 8g show the results of performing the simplification with planarity-threshold and collinearity-threshold values of 70’. The figures show three stages of the simplification corresponding to the same levels of triangle decimation as those in Figures 8b, 8c, 8d. We obtain in this case a final decimation of triangles of 81%. Even though the simplified meshes are different, we observe that both strategies give similar levels of decimation.

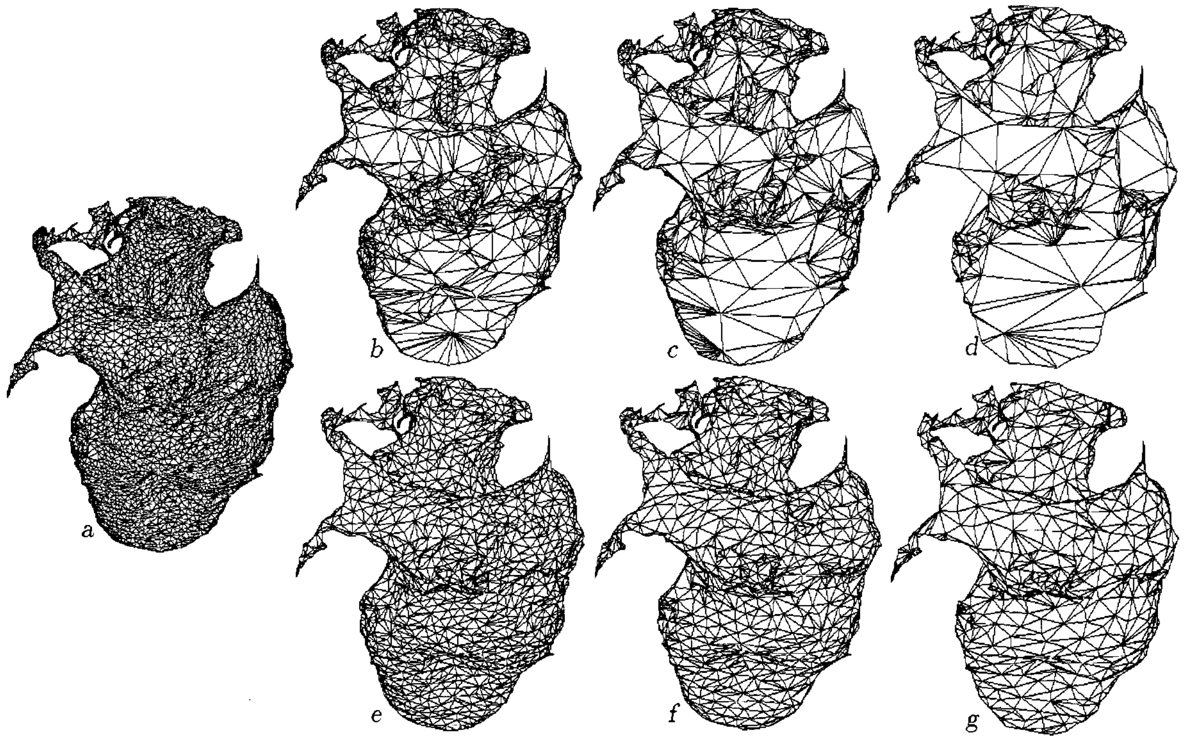


Figure 8: *Two different simplification strategies: a) original dense triangulation, b) simplification with planarity-threshold and collinearity-threshold values of 40° (decimation by 62%), c), d by incrementing the thresholds by 15° to 55° and then to 70° (decimation by 70% and 82%), e), f), g) direct simplification with planarity-threshold and collinearity-threshold values of 70° e) , f) partial decimation results that coincide with the level* of *decimation of b) and c), e) the final result (decimation by 81%).*

9. Simplification Results

Results of the simplification algorithm are shown in Figures 9 and 10 for six dense triangulations. We show the original meshes and the results of simplifying them by progressively incrementing the threshold values. The heart triangulation (16,404 triangles, back side shown) in Figure 9a was simplified using planarity-threshold and collinearity-threshold values of 25" and 80° to obtain decimations of 40% and 85% respectively. In Figure 9b we show two decimations of the triangulation of the brain ventricles (7,520 triangles). They were produced using planarity-threshold and collinearity-threshold values of 35° and *80*°, with a decimation of triangles of 60% and 93% respectively. Figures 9c shows simplifications of a cube with a hole (5,200 triangles). The planarity-threshold and collinearity-threshold values in this case were fixed at 90°, but we show two different stages of the simplification process, with a decimation of triangles of 55% and 99.6% respectively.

In Figure 10a we show two stages of the simplification of a spherical mesh with 2048 triangles using a fixed planarity-threshold and collinearity-threshold values of 35'. The simplified meshes have been decimated by 51% and 92% respectively. In Figure 10b we show the dense triangulation of a pair of brain ventricles with 15,804 triangles. The two simplifications were produced by progressively incrementing the planarity-threshold and collinearity-threshold values from **30°** to 70°, and have been decimated by 42% and 84% respectively. Figure 10c shows a triangulation of the brain's white matter containing 100,981 triangles and two simplifications with planarity-threshold and collinearity-threshold values of 30°, and 75°, decimated by 28% and 66% respectively.

10. Future Work

The simplification algorithm produces satisfactory decimations of arbitrary dense meshes of between 80% and 90% of the triangles. The edge-feature selection strategy succeeds well in identifying the main characteristics of the original meshes and in preserving them in the simplified meshes. In our implementation, these surface characteristics are the "frame" that supports the simplification operations and that preserves the good fit between the dense and simplified meshes even with severe decimation ratios. However, when decimating dense meshes with little or no surface characteristics such as a

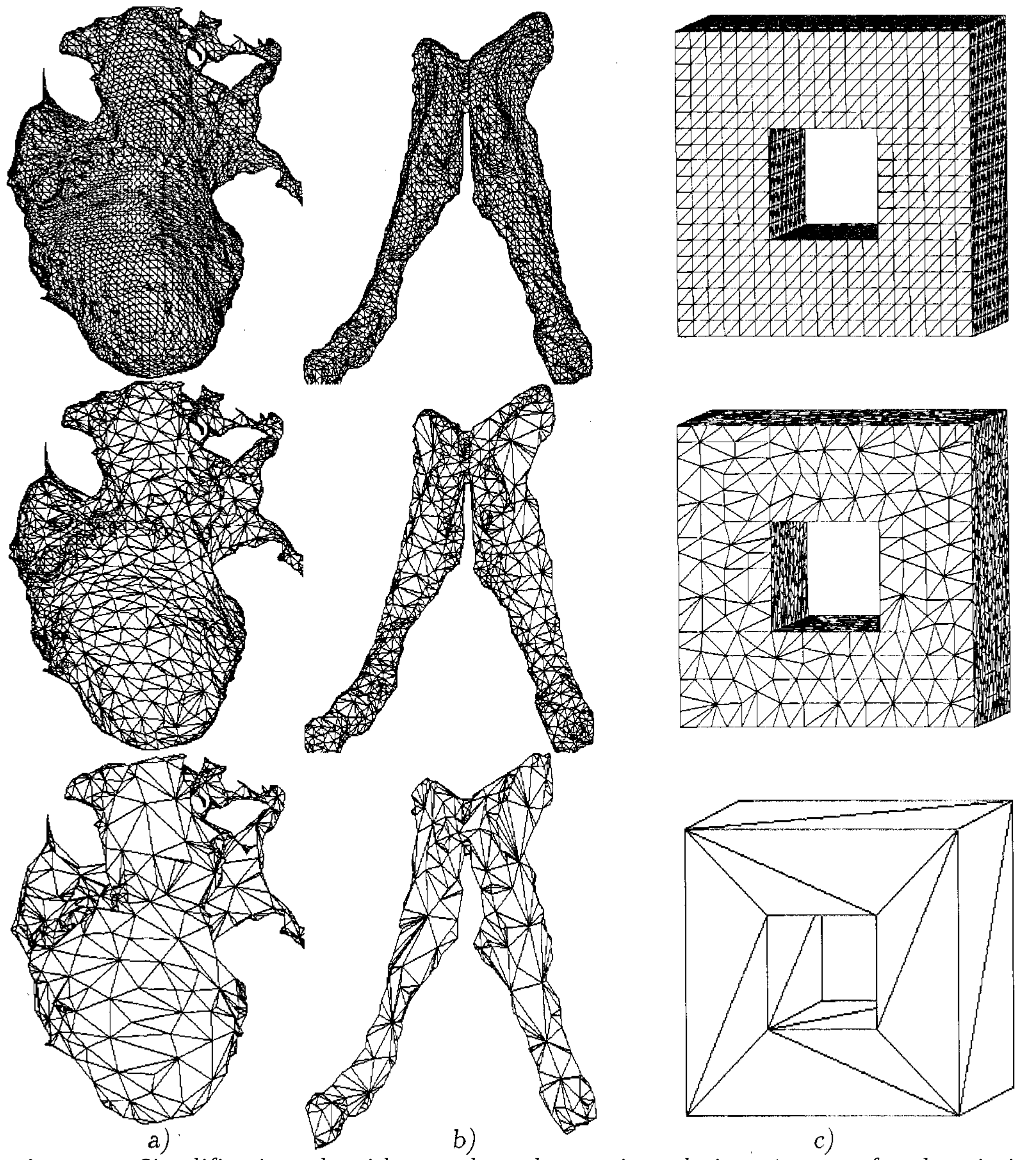


Figure 9: *Simplification algorithm on three dense triangulations (see text for description).*

dense sphere or a dense cylinder, our decimation strategy will lose the global shape characteristics if we push the decimation level beyond the 80%-90% limit. This is the result of our algorithm being a triangulation filter rather than a triangulation optimization. Our current work consists in constraining the position of the vertices as the simplification proceeds so that they remain at a tolerance distance from the original dense triangulation S'. This implies measuring the error in the simplification results with respect to the dense triangulation. We can incorporate techniques as those proposed by Guéziec and Hummel [4] and Kalvin and Ross [5] for measuring the error in simplified triangulations.

11. Conclusion

We propose a cluster algorithm to implement edge-collapse operations on a triangulation in order to produce a simplified mesh. The clusters distribute the edge-collapse operations over the mesh such that they remain independent from one another. This distributed implementation avoids the appearance of difficult to control propagation effects and reduces the global simplification problem to a set of strictly local operations. The clusters are pre-characterized and well-suited regions for re-triangulation. They avoid the appearance of particular cases and allow the performance of fast edge collapses. An advantage of this approach is that it allows a uniform treatement of all the triangles in

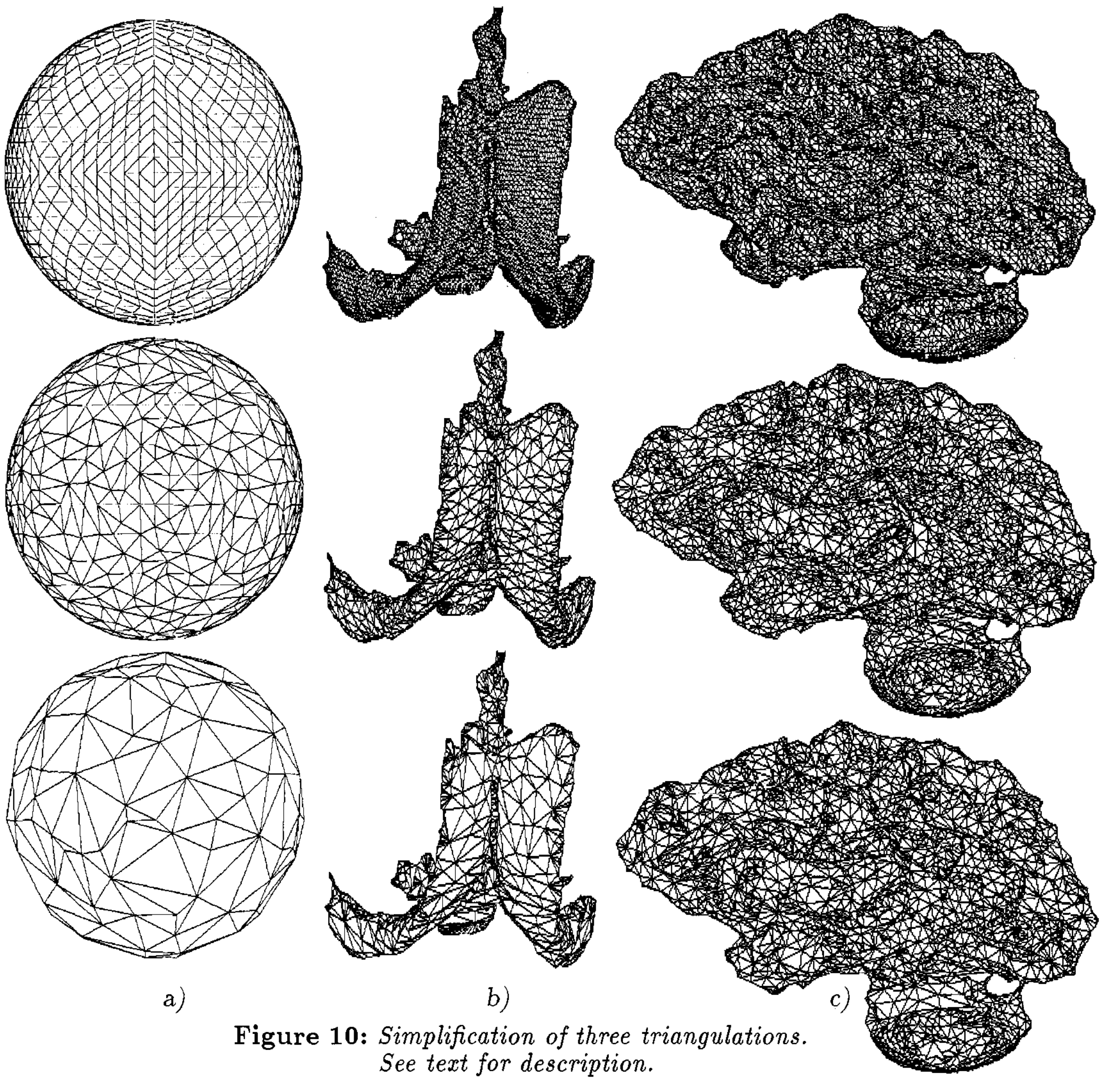


Figure 10: *Simplification of three triangulations. See text for description.*

a closed mesh and fast topological checks to verify that the triangulation is not altered. By carrying out a global, constrained swapping between the simplification iterations we guide the algorithm to a better simplification minimum and we regularize the decimated mesh. The algorithm takes on input two user-specified angle values: a planarity threshold under which the edge-collapse operations will proceed, and a collinearity-threshold under which the edges of characteristic lines can be simplified. By manipulating the values of the thresholds we can make the simplification operations evolve differently over the mesh to produce different types of triangulations for a same level of triangle decimation.

References

1. X.Chen, "Modélisation Géometrique par Vision Artificielle" (in english), Ph.D. dissertation, Ecole Nationale Supérieure de Télécommunications, 923023 (1992).
2. N.Dyn, D.Levin, S.Rippa "Data Dependent Triangulations for Piecewise Linear Interpolation", IMA Journal of Numerical Analysis, 10 pp. 137-154 (1990).
3. A.Gourdon, "Decimation of Orientable Surface Meshes", Proc.9th Scan.Conf.Image Anal., pp.787-794(1995
4. A. Gueziec, R. Hummel, "The Wrapper Algorithm: Surface Extraction and Simplification", in IEEE CVPR - *WBIA*, pp. 204-213, Seattle (1994).
5. A.D.Kalvin, and R.H.Taylor, "Polyhedral Approximation with Bounded Error", *SPIE's* Medical Imaging'94 - Image Display, bf 2164 pp. 247-258 (1994).
6. H.Hoppe, T.deRose, T.Duchamp, J.McDonald, W.Stuetzle, "Mesh Optimization", SIGGRAPH, pp.19-26(1993)
7. W.J.Schroeder, J.Zarge, W.E.Lorensen, "Decimation of Triangle Meshes", SIGGRAPH, 26(2) pp.65-70(1992)
8. G.Turk, "Re-Tiling Polygonal Surfaces", SIGGRAPH, 26(2) pp.55-64(1992)