

View-Dependent Extraction of Contours with Distance Transforms for Adaptive Polygonal Mesh Simplification

Susana Mata*, Luis Pastor†, Angel Rodríguez‡

*Dpt. de Arquitectura y Tecnología de Computadores, Ciencia de la Computación e Inteligencia Artificial
U. Rey Juan Carlos (URJC)
C. Tulipán, s/n., 28933 Móstoles, Madrid, Spain
+34914888111, email: susana.mata@urjc.es
www: <http://www.dac.escet.urjc.es/personal/smata/>

†Dpt. de Arquitectura y Tecnología de Computadores, Ciencia de la Computación e Inteligencia Artificial
U. Rey Juan Carlos (URJC)
C. Tulipán, s/n., 28933 Móstoles, Madrid, Spain
+34916647442 email: luis.pastor@urjc.es
www: <http://www.dac.escet.urjc.es/personal/lpastor/>

‡Dpt. de Tecnología Fotónica
U. Politécnica de Madrid (UPM)
Campus de Montegancedo s/n., 28660 Boadilla del Monte, Madrid, Spain
+34913367371, email: arodri@dtf.fi.upm.es
www: www.dtf.fi.upm.es/~arodri

Abstract

During decades Distance Transforms have proven to be useful for many image processing applications, and more recently, they have started to be used in computer graphics environments. The goal of this paper is to propose a new technique based on Distance Transforms for detecting mesh elements which are close to the objects' external contour (from a given point of

view), and using this information for weighting the approximation error which will be tolerated during the mesh simplification process. The obtained results are evaluated in two ways: visually and using an objective metric that measures the geometrical difference between two polygonal meshes.

Keywords: Computational Geometry and Object Modelling, Three Dimensional Graphics and Realism, Picture and Image Generation

Digital Peer Publishing Licence

Any party may pass on this Work by electronic means and make it available for download under the terms and conditions of the current version of the Digital Peer Publishing Licence (DPPL). The text of the licence may be accessed and retrieved via Internet at <http://www.dipp.nrw.de/>.

First presented at the International Conference on Computer Graphics Theory and Applications (GRAPP) 2006, extended and revised for JVRB

1 Introduction

The number of fields using three dimensional graphics is constantly increasing. Different areas like medicine, education, training, leisure, etc. are more and more taking advantage of the benefits that 3D visualization can bring to them.

Representing objects is one of the key stages of any computer graphics system. Among the mainstream models in this field, the most extended technique for

representing the surface of objects is based on polygonal meshes, being the simplicity of its basic elements one of the underlying reasons [Wat00].

However, polygonal meshes need a large number of polygons for approximating real objects with a reasonable accuracy. Whenever computation time is a concern, techniques which decrease the model's polygon count while keeping an acceptable visual appearance are mandatory. An approach that has been widely investigated is multiresolution modelling, which consists in representing objects at different resolution levels and choosing the proper approximation according to the visualization conditions [XV96, Hop97]. Basic principles of this approach were set by James Clark [Cla76]; comprehensive surveys can be found at [PS97, Gar99, Lue01, LRC⁺03, FKP05].

The key point is to obtain a simplified model using less computational resources but looking similar to the original one when rendered under certain conditions. No attention needs to be paid to features not observable by the end user. Moreover, some details in a model become more obvious than others to a human viewer. For example, silhouettes are known to be critical for the final visual quality appreciated by our visual system [LE97].

The goal of this paper is to propose a new technique that allows taking into account the proximity of a mesh element to the mesh's external contour for weighting the approximation error which will be tolerated during the simplification process. More specifically, the contributions of this work can be briefly summarized as follows:

- Analyzing the applicability of Distance Transforms for detecting the proximity of mesh elements to the external silhouette.
- Proposing mesh simplification criteria based on a precomputed silhouette proximity measure, obtained by means of a Distance Transform.
- Presenting a simplification technique using a Distance Transform for selecting the allowed approximation error in regions close to the external silhouette.

The rest of the paper is organized as follows: Section 2 presents a short overview of some previous work related to mesh simplification algorithms and the different approaches to identify and preserve the model's silhouette. A brief introduction to basic concepts of digital Distance Transforms and Multi-Tessellation is

also included. Section 3 describes the proposed approach, while Section 4 shows some experimental results. Finally the conclusions and future work are presented in Section 5.

2 Previous Work

2.1 Mesh Simplification

Many mesh simplification techniques have been proposed during the last years. Among the methods based on objective metrics, work has been done in order to incorporate other attributes besides geometry like color, texture or normals [GH98, COM98]. Perceptual metrics have also been developed [OHM⁺04, CB05]; Lindstrom and Turk use an image metric to guide the simplification process [LT00]. Reddy introduced a perceptive model to guide the selection of the appropriate level of detail [Red97]. [Lue98] defined a contrast sensitivity function that predicts the perception of visual stimuli. Some of the perceptually driven simplification methods explicitly pursue a good silhouette preservation, defining normal cones [WLC⁺03]. Good silhouette approximation through contour computation in image space has also been researched [RC99, SGG⁺00].

The approach presented here not only identifies the objects' silhouette. It performs also a mesh elements' explicit classification in object space, attending to its proximity to the external contour from a given point of view.

2.2 Digital Distance Transforms

Measuring the distance between image elements may be of interest for further processing in many image analysis applications. Basics concepts regarding digital distances can be found in [RP66, RP68].

The application of a Distance Transform to a binary image produces as output a distance image, where each element of this distance-image is assigned a distance label. For any element its label stores a value indicating its closest distance to the background. Therefore, the computed distance image can be seen as a grey-level image where the intensity level identifies the minimum distance to the complement of the object.

A distance transform can be computed in two steps by propagating local distances over the image; this is true for 2D, 3D and higher dimensions [RP66]. Initially, the elements belonging to the object are set to

Algorithm 1 2D Digital distance transform.

```

1: {INPUTS: 2D binary image  $I$ , 2D digital mask  $w$ }
2: {OUTPUT: 2D digital distance transform computed  $I$ }
3: {First step: forward propagation}
4: for  $y = 1$  to  $n$  do
5:   for  $x = 1$  to  $n$  do
6:      $I(x, y) = \min(i, j) \in fwd(I(x + i, y + j) + w(i, j))$ 
7:   end for
8: end for
9: {Second step: backwards propagation}
10: for  $y = n$  downto 1 do
11:   for  $x = n$  to 1 do
12:      $I(x, y) = \min(i, j) \in bwd(I(x + i, y + j) + w(i, j))$ 
13:   end for
14: end for
15: being  $fwd$  the forward mask (bolded coefficients) and  $bwd$  the backwards mask (italic weights) defined in Table 1.

```

| | | | | |
|-----------|-----------|----------|-----------|-----------|
| | 11 | | 11 | |
| 11 | 7 | 5 | 7 | 11 |
| | 5 | ■ | 5 | |
| <i>11</i> | <i>7</i> | <i>5</i> | <i>7</i> | <i>11</i> |
| | <i>11</i> | | <i>11</i> | |

Table 1: Mask applied for computing the 2D digital transform: 5, 7 and 11 weights are assigned for edge, vertex and knight neighbour relationships.

infinity and the elements belonging to the background are set to 0. In the case of a 2D image, during the first step the image is analyzed from top to bottom and from left to right. During the second step, the image elements are visited from right to left and from bottom to top. Each element is assigned the minimum value between itself and the already visited neighbours incremented by their connectivity weight. Algorithm 1 details the pseudo-code for computing the distance transform over a $n \times n$ image.

Distance transforms and some variations of them in combination with other image processing techniques can be applied for representing and analyzing 3D objects in multiple applications [Nys97, Sve01, Sin05, JBv06]. Distance fields have also been applied in computer graphics environments, such as in collision detection [TKZ⁺04], and have been implemented with graphics hardware [SGGM06].

However, digital distance transforms can be used in other fields that have not been explored so far. The work presented here aims to open a way for new

applications of Distance Transforms within computer graphics environments.

2.3 Multi-Tessellation

The Multi-Tessellation method, originally called Multi-Triangulation, was introduced by De Floriani *et al.* [FMP97]. It provides a general multiresolution framework for polygonal meshes offering several attractive features, like selective refinement, locality or dynamic update [FMP97].

Multi-Tessellation, MT for short, is a hierarchical model defined as a directed acyclic graph, where the nodes represent mesh updates and arcs represent dependencies between updates labelled with sets of triangles. Each triangle t may have associated a global approximation error, describing the difference between t and the surface patch it approximates. Simplification applications may be adapted in order to build an MT while performing the simplification process. Once the MT has been built, it can be queried at run time for extracting a simplified mesh fulfilling some defined restrictions.

Two kinds of criteria can be defined for selectively refining the mesh (see Alg. 2):

- A threshold function, used to bound the approximation error.
- A focus condition, used to set the region of interest under which the query will be evaluated.

The distributed package [MG] implements a hierarchy of classes that permit the construction of an MT

Algorithm 2 Criteria used for refining the mesh.

```

if trianglei ∈ Focus_region then {Focus condition}
    Evaluate threshold_function(trianglei)
end if
boolean threshold_function(trianglei){Bound approximation error} (trianglei.error ≤
Allowed_error(trianglei))
    
```

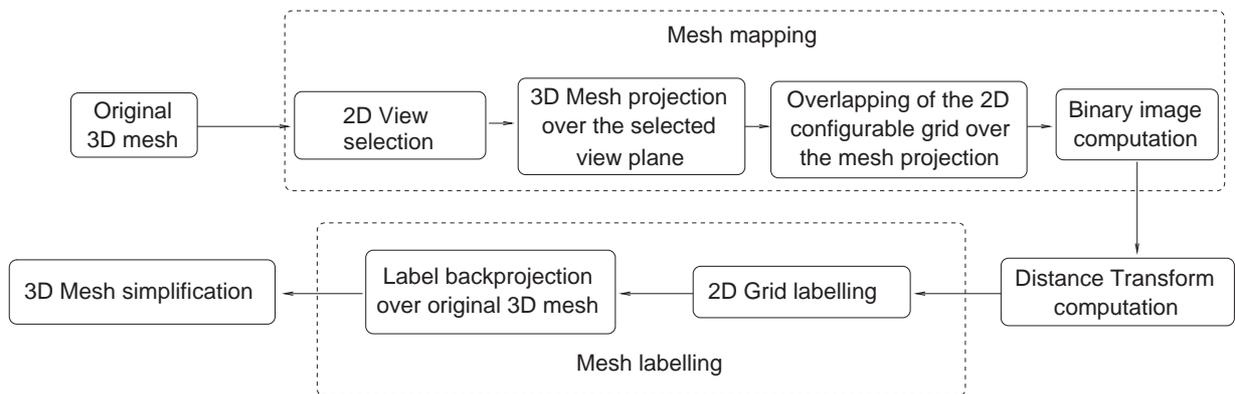


Figure 1: Mesh simplification stages.

by adapting a simplification method in order to perform some basic MT operations. Some useful threshold and focus conditions are already implemented, including point location, region intersection, etc. Additionally, an executable implementing the Jade simplification technique is provided [CCMS97]. In this case, the simplification algorithm is based on vertex decimation operators.

Defining new focus and threshold conditions can be easily done by adding new classes to the library. This sets a general and useful framework for dynamically managing the level of detail of a particular representation.

The MT package has been used in this work to present the results obtained by integrating the precomputed distance to silhouette into the threshold function.

3 Method Description

The approach followed here classifies the mesh faces or vertices attending to their proximity to the external silhouette from a certain point of view. The classification process uses a Distance Transform, computed over the mesh elements' projection on the visualization plane. This transform provides for each element its distance to the projected contour, being useful for extracting the mesh elements which compose or are located near the mesh silhouette for any particular point

of view.

Previous to the projection stage, a 2D grid is created on the visualization plane. This grid needs a resolution level that has to be selected according to the input mesh level of detail, the desired simplification degree and the available time to perform the labeling process. Every cell in the 2D grid has a list associated containing the indexes of the faces that project into it. The occupancy of the grid cells can be represented as a binary image; it is over this binary image where Distance Transforms can be computed. Last, a classification of the model faces can be obtained backprojecting the distance labels to the mesh faces that map into each grid cell.

The same process can be applied for backprojecting the distance values to the vertices instead of to the faces. The tags assigned to the polygonal mesh elements can then be used in different ways to guide the simplification process, providing a criterion for modifying locally the approximation error allowed in areas close to the external contour.

It must be highlighted that the first three steps are performed in a pre-processing stage, producing a labelled polygonal mesh which will be used later on during the simplification stage. Figure 1 depicts a scheme of the whole process and Alg. 3 collects its pseudo-code description. The following Sections describe each of the method's stages.

Algorithm 3 Pseudo-code of the overall contour simplification process.

```
1: {INPUTS: 3D mesh, visualization parameters}
2: {OUTPUT: 3D simplified mesh}
3: Create a 2D grid over the visualization plane of the 3D input mesh
4: for all grid-cells do {It is computed in a pre-processing stage}
5:   Label each grid-cell with the 3D mesh vertices that project onto it
6: end for
7: Extract a binary image representing the grid-cells occupation {Each pixel represents a grid cell}
8: Compute a Distance Transform over the binary image
9: Assign to each grid-cell the distance value of its associated pixel
10: for all grid-cells do {Assign labels to 3D vertices}
11:   Backproject its distance value to all the 3D mesh vertices that project onto it
12: end for
13: Simplify the 3D mesh according to backprojected distance labels
```

3.1 Mesh Mapping

Silhouettes are view-dependent features. For that reason, their extraction must be done for a certain view point. Given a visualization plane, the 3D mesh is projected on it by applying the proper projection matrix to each vertex coordinates. In order to extract the object's silhouette, it is necessary to create a binary image where distance measurements can be carried out. For that purpose the visualization plane is partitioned into cells forming a grid which can be seen as a 2D digital image. The number of cells making up the grid is analogous to the image resolution; consequently the parameterization of this value allows the analysis at different resolutions.

Every face belonging to the projected polygonal mesh is tested to find the cells of the 2D grid with which it intersects.

A data structure is updated where every grid element keeps track of the faces intersecting with it. This way, the posterior backprojection of distance values is straightforward. This procedure is computationally expensive, but affordable as pre-processing.

The binary image is extracted from the grid occupancy information, setting as object every cell with any face mapping over it.

3.2 Distance Transform Computation

Once the 2D image is obtained, the next stage consists in obtaining a distance image by applying a distance transform to the binary image. The result is a new image where the assigned intensity values increase as the pixel gets further away from the background.

3.3 Mesh Labelling

At this point, the distance of an object pixel to the background has already been computed. Previously, the correspondences between pixels and the facets mapping into them have also been calculated. Therefore, the labelling of every face with a value representing its distance from the background is a simple process. The distance label of a pixel, which is equivalent to a grid cell, is assigned to all the faces that intersect with the cell.

As a result, a labelled mesh is obtained where every face stores a value representing its proximity to the external contour.

The same approach may be followed when the distance label is assigned to vertices or edges instead of faces. Figure 2 shows the results of backprojecting the distance values onto the mesh. Fig. 2(a) shows a rendered view of the original mesh. Fig. 2(b) and 2(c) represent the same mesh under different points of view. The grey levels in the images represent distance to the silhouette (lighter intensities represent higher distances to the contour).

3.4 Mesh Simplification

The method's last stage is also the final goal of the whole process, where the extracted distance values are used for mesh simplification purposes.

The use of the distance labels depends on the selected simplification technique. The work presented here has been based on the Jade approach, a vertex decimation technique based on the global error [CCMS97]. The distance information is computed for the vertices of the original mesh. Since the vertices be-

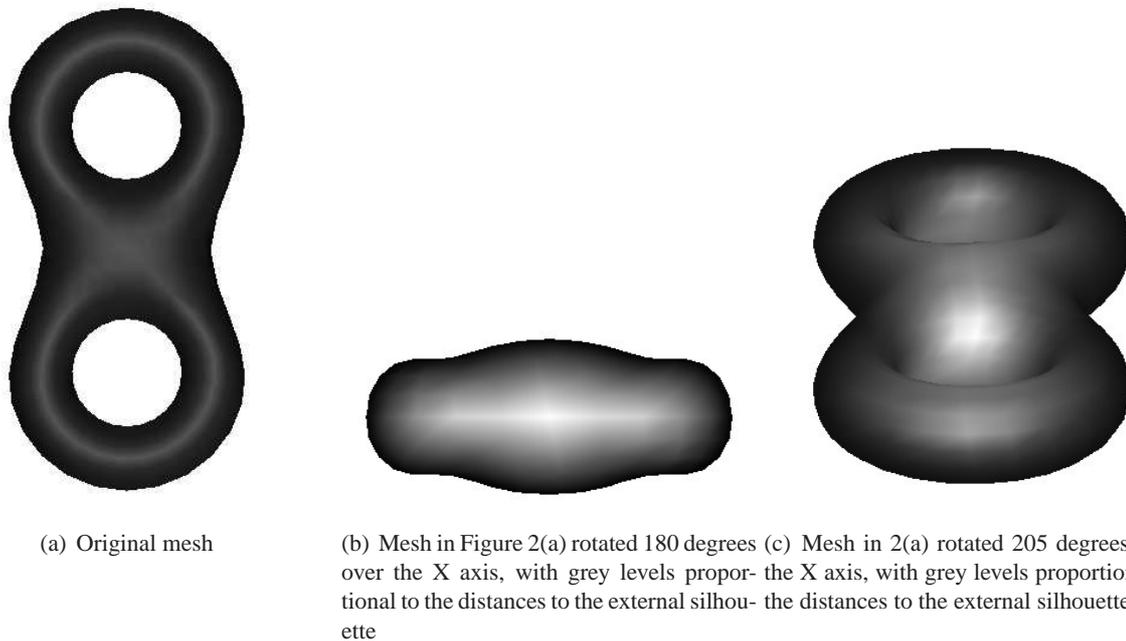


Figure 2: Backprojection of distance values in image 2(a) over the 3D model. Mesh vertices color represent the backprojected distance label

longing to a simplified model are a subset of the original mesh, the precomputed distance labels are valid for any level of detail. Multi-Tessellations obtained through the application of the Jade method are freely distributed with the MT-Package.

The proximity of every facet to the external contour is taken into account in the extraction stage. This means that for a given error threshold, the error allowed in regions close to the external silhouette is reduced according to a predefined law.

The implemented solution, requires the definition of two parameters:

- Distance interval: range of distance labels which identify the region where a more accurate approximation is desired.
- Restriction factor f : the purpose of this parameter is to define a lower error threshold for the portion of the mesh within the region of interest.

The width of the contour area can be simply modified by changing the range of distance labels that define the region of interest. In our case, the range is defined by setting a threshold over the minimum distance of the vertices belonging to a face. Given a distance threshold, and a vertex v_j of triangle_i :

if $v_j.\text{label} \leq \text{distance_threshold}$ **then**

$\text{triangle}_i \in \text{contour}$
else
 $\text{triangle}_i \notin \text{contour}$
end if

Other solutions can be easily devised.

The error factor allows to refine the quality of the approximation in the contour region taking into account the threshold error fixed for the rest of the model. Given a global allowed error e we can define a more restrictive error that will be tolerated in the contour region. Given the restriction factor $f < 1$, the allowed error will be computed in the following manner:

if $\text{triangle}_i \notin \text{contour}$ **then**
 Allowed_error = e
else
 Allowed_error = $e \cdot f$
end if

If the allowed error is uniform along the model, then

$$\text{Allowed_error}(\text{triangle}_i) = e \forall i$$

Again, other error functions are also feasible.

4 Results

The experimental results presented in this section were obtained by applying the technique previously de-

scribed to the Multi-Tessellation models distributed together with the MT-Package.

Figure 3(a) shows the simplification of the mesh obtained by imposing a restrictive error threshold over the silhouette. In this case, the error factor was set to 0, meaning that no error is allowed on the external contour. The region of interest (the mesh portion considered to be near the silhouette) is made up of faces whose vertices have a minimum distance label less than or equal to 2. It can be seen that the rest of the mesh is coarser (it has suffered a strong simplification process), while the density of triangles over the silhouette is extremely high.

The algorithm presented here allows an easy parameterization of the width of the contour area, as it can be appreciated in Figure 3(b). It can be seen how the region of high resolution extends itself towards the interior of the object.

Figures 3(c) and 3(f) show two simplified meshes with approximately the same number of polygons, 1074 and 1077 respectively, obtained with two methods: the Jade and the distance-based approach proposed here. Figures 3(d) and 3(g) show the simplification from a different point of view. Figure 3(d) (1226 triangles) has been obtained from Jade simplification, while Figure 3(g) has been obtained using the distance information. A perceptible improvement in the smoothness of the silhouette can be appreciated in both Fig. 3(f) and 3(g). A wireframe rendering of the model reveals a higher density of facets close to the external contour in Fig. 3(h) with respect to 3(e).

Figure 4 shows some results of simplifications applied to the model shown in Fig. 2(a). Figure 4(a) (612 triangles) is the simplified model extracted from Jade simplification method, while Fig. 4(d) (600 triangles) represents the simplification using distance information. Under a different rotation, the simplification is carried out using distance labels in 4(e) (510 triangles) while in Fig. 4(b) (508 triangles) no distance information is taken into account. Again, the smoothness of the silhouette is higher with our method and the mesh density increases in the proximity of the external contour, as shown in Fig. 4(c) (without distance information) and 4(f) (with distance information).

In addition to a visual inspection, an objective measurement of the approximation error has also been performed. The difference between the two polygonal meshes to be compared is computed following an approach similar to [ASCE02]. Given a mesh M1 and a coarser approximation M2, for every vertex of M1

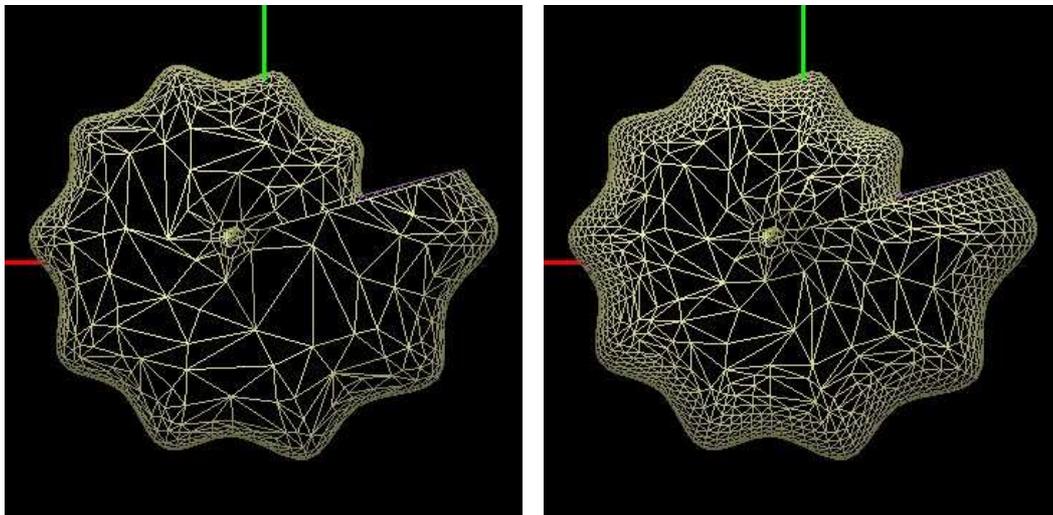
the minimum distance to the faces belonging to M2 is computed. A visual representation of the deviation is shown by colouring M1 with a predefined colour palette. Figure 5 presents the result of comparing the simplifications obtained with and without distance information. It can be observed that the approximation in the silhouette is quantitatively better with our method.

Regarding computational issues, cost in terms of memory requirements is only of one extra value per vertex. In case of tagging the mesh faces or mesh edges an additional value per tagged element would be required. With respect to computational cost, it has to be noted that all the heavy computation is performed at pre-processing time. The most expensive step is the mesh mapping over the 2D grid, in order to collect the information needed for backprojecting the distance values. Efficient implementations for these operations using spatial data partitioning could be considered. As it was explained above, the computation of the Distance Transform can be performed involving only two passes over the 2D image.

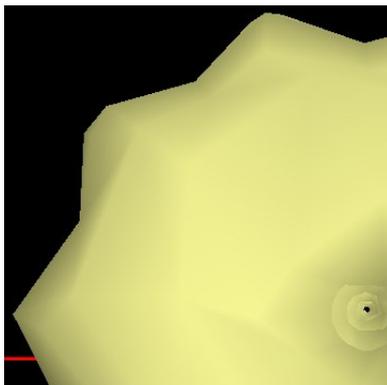
5 Conclusions and Future Work

Simplification algorithms are usually guided by some criteria in order to select which elements of the mesh shall be removed or replaced. Introducing precomputed distance labels as part of the guiding metrics is a straightforward process, opening a new way to design a range of techniques which are useful for including perceptually motivated criteria in mesh simplification algorithms. This approach can be applied in order to achieve higher resolution in other relevant regions besides the silhouette, such as visually outstanding areas, or semantically important parts. The results presented here suggest that the use of distance information is a promising approach for mesh simplification techniques, since adding distance labels to mesh elements provides more information than the conventional methods based on the extraction of the silhouette edges.

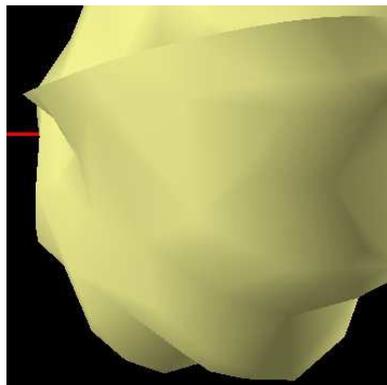
The fact that distance information can be assigned to any element of the mesh (vertices, edges or faces) facilitates adapting these techniques to a wide range of simplification methods. The nature of the basic underlying operator (vertex removal, edge collapse, etc) does not impose additional limitations. Furthermore, the applicability of distance labels goes from off-line simplification processing to run-time selective refine-



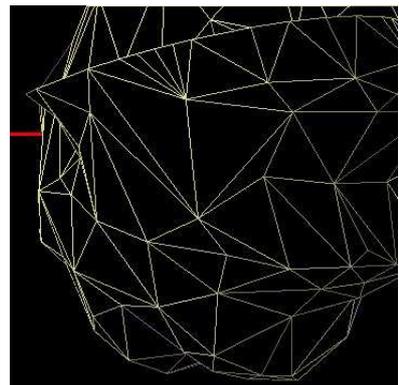
(a) Result of simplifying mesh elements with distance values larger than 2 (b) Result of simplifying mesh elements with distance values larger than 15



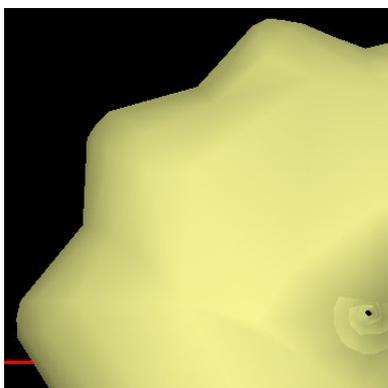
(c) Close-up view after the Jade simplification



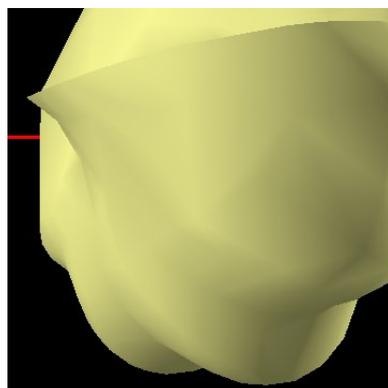
(d) The same simplified mesh from another point of view



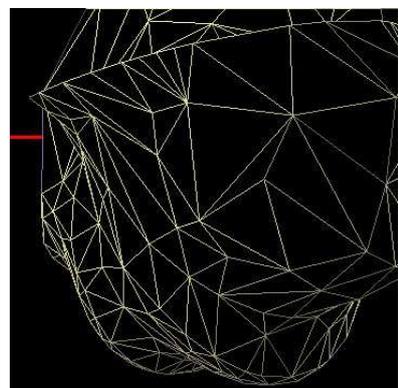
(e) Wire mesh of 3(d)



(f) Close-up view after the Distance Transform based simplification



(g) A Distance Transform based simplification for a different point of view



(h) Wire mesh of 3(g)

Figure 3: Shell example.

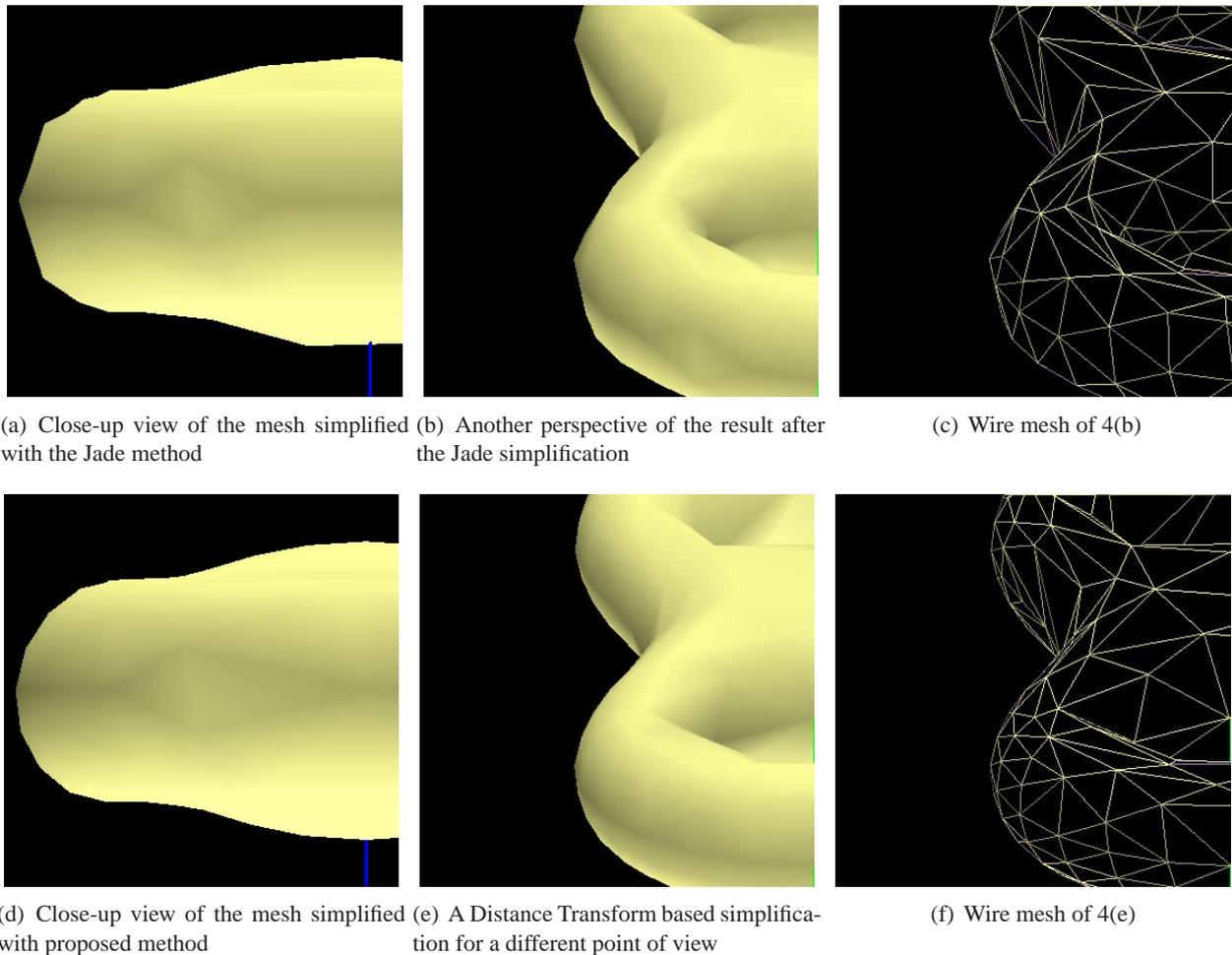


Figure 4: Abstract geometric example.

ment. Simplification techniques have a wide range of applications in leisure, science, industry, arts, etc. All of them can benefit from the improvements in the quality of the simplified models.

The work presented here computes the mesh elements' distance to the extended contour given a predefined point of view. Future work includes:

- Extending the method for covering all possible points of view in a way which is both performant and computationally efficient.
- Integrating distance to the silhouette into other mesh simplification methods besides the Jade method.
- Extending the method in order to consider also internal silhouettes.

6 Acknowledgments

This work has been partially funded by the Spanish Ministry of Education and Science (grant TIC2003-08933-C02) and Government of the Community of Madrid (grants GR/SAL/0940/2004 and S-0505/DPI/0235).

The authors also thank to the Geometric Modelling and Computer Graphics Research Group for distributing the MT-Package.

References

- [ASCE02] Nicolas Aspert, Diego Santa-Cruz, and Touradj Ebrahimi, *Mesh: Measuring error between surfaces using the hausdorff distance*, Proceedings of the IEEE International Conference in Multimedia and

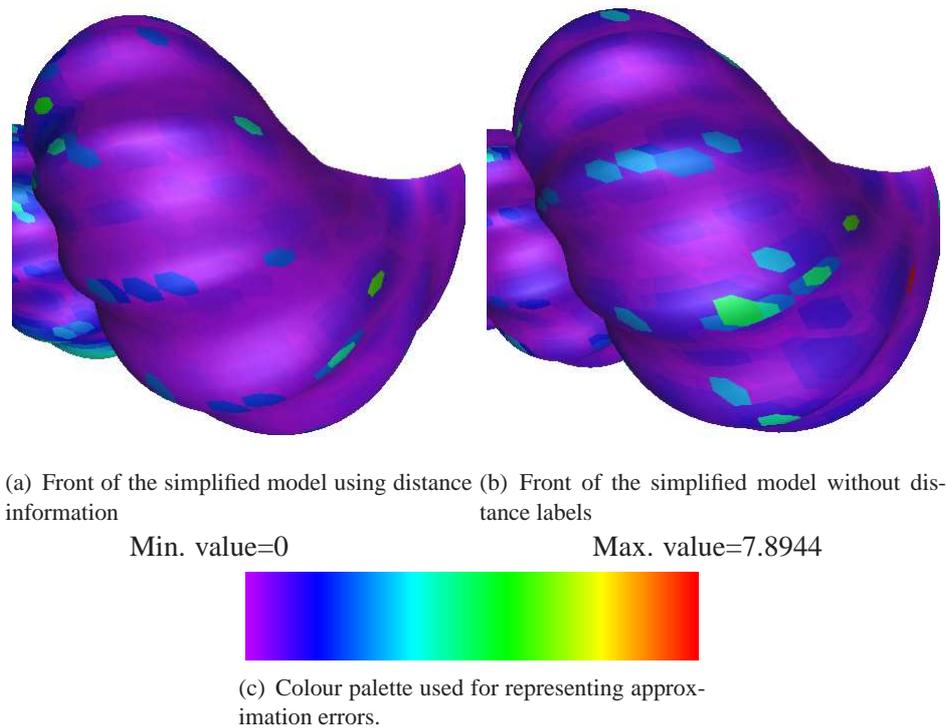


Figure 5: Visual representation of the approximation errors of the simplified models shown in Figures 3(c) and 3(f) rotated for a better perception of some quantitative values.

Expo (ICME), vol. 1, 2002, ISBN 0-7803-7304-9, pp. 705–708.

[CB05] Irene Cheng and Pierre Boulanger, *A 3d perceptual metric using just-noticeable difference*, Proceedings of Eurographics 2005, 2005, ISSN 1017-4656, pp. 97–100.

[CCMS97] Andrea Ciampalini, Paolo Cignoni, Claudio Montani, and Roberto Scopigno, *Multiresolution decimation based on global error*, The Visual Computer **13** (1997), 228–246, ISSN 1432-2315.

[Cla76] James H. Clark, *Hierarchical geometric models for visible surface algorithms*, Communications of the ACM **19** (1976), 547–554, ISSN 0001-0782.

[COM98] Jonathan Cohen, Marc Olano, and Dinesh Manocha, *Appearance-preserving simplification*, Proceedings of SIGGRAPH'98, Annual Conference Series, 1998, ISBN 0-89791-999-8, pp. 115–122.

[FKP05] Leila De Floriani, Leif Kobbelt, and Enrico Puppo, *A survey on data structures for level-of-detail models*, Series in Mathematics and Visualization, pp. 49–74, Springer Verlag, Berlin, 2005, ISBN 3-540-21462-3.

[FMP97] Leila De Floriani, Paola Magillo, and Enrico Puppo, *Building and traversing a surface at variable resolution*, Proceedings of IEEE Visualization 97, IEEE Computer Society Press, 1997, ISBN 0-8186-8262-0.

[Gar99] Michael Garland, *Multiresolution modeling: Survey and future opportunities*, STAR Proceedings of Eurographics'99 (Geneva, Switzerland), Eurographics technical report series, vol. 1, Eurographics Association, 1999, ISSN 1017-4656.

[GH98] Michael Garland and Paul Heckbert, *Simplifying surfaces with color and texture using quadric error metrics*, Proceedings of IEEE Visualization VIS'98, IEEE Computer Society, 1998, ISBN 0-8186-9176-X, pp. 263–270.

- [Hop97] Hughes Hoppe, *View-dependent refinement of progressive meshes*, Proceedings of the ACM Conference SIGGRAPH'97 (New York), Computer Graphics Annual Conference Series, ACM, August 1997, ISBN 0-89791-921-1, pp. 189–198.
- [JBv06] Mark W. Jones, Andreas Bærentzen, , and Miloš Šrámek, *Discrete 3d distance fields: Techniques and applications*, IEEE Transactions on Visualization and Computer Graphics **12** (2006), 581–599.
- [LE97] David Luebke and Carl Erikson, *Viewdependent simplification of arbitrary polygonal environments*, Proceedings of 24th annual conference on Computer graphics and interactive techniques SIGGRAPH' 97 (New York), ACM, 1997, ISBN 0-89791-896-7, pp. 199–208.
- [LRC⁺03] David Luebke, Martin Reddy, Jonathan D. Cohen, Amitabh Varshney, Benjamin Watson, , and Robert Huebner, *Level of detail for 3d graphics*, The Morgan Kaufmann series in computer graphics and geometric modeling, Morgan Kauffmann Publishers, San Francisco, 2003, ISBN 1-55860-838-9.
- [LT00] Peter Lindstrom and Greg Turk, *Imagedriven simplification*, ACM Transactions on Graphics (ToG) **19** (2000), 204–241, ISSN 0730-0301.
- [Lue98] David P. Luebke, *View-dependent simplification of arbitrary polygonal environments*, Ph.D. thesis, University of North Carolina, 1998.
- [Lue01] David P. Luebke, *A developer's survey of polygonal simplification algorithms*, IEEE Computer Graphics and Applications **21** (2001), 24–35, ISSN 0272-1716.
- [MG] Geometric Modeling and Computer Graphics Research Group, *The mt (multi-tesselation) package*.
- [Nys97] Ingela Nyström, *On quantitative shape analysis of digital volume images*, Ph.D. thesis, Uppsala University, 1997.
- [OHM⁺04] Carol O'Sullivan, Sarah Howlett, Yann Morvan, Rachel McDonnell, and Keith O'Connor, *Perceptually adaptive graphics*, Eurographics State-of-the-Art Report (EG-STAR), vol. 6, Eurographics Association, 2004, pp. 141–164.
- [PS97] E. Puppo and R. Scopigno, *Simplification, lod and multiresolution principles and applications*, EUROGRAPHICS'97, vol. 16, 1997, Tutorial Notes PS97 TN4.
- [RC99] Ramesh Raskar and Michael F. Cohen, *Image precision silhouette edges*, Proceedings of the 1999 Symposium on Interactive 3D Graphics SI3D'99, 1999, ISBN 1-58113-082-1, pp. 135–140.
- [Red97] M. Reddy, *Perceptually modulated level of detail for virtual environments*, Ph.D. thesis, University of Edinburgh, 1997.
- [RP66] A. Rosenfeld and J.L. Pfaltz, *Sequential operations in digital picture processing*, Journal of the Association for Computing Machinery **13** (1966), 471–491, ISSN 0004-5411.
- [RP68] A. Rosenfeld and J.L. Pfaltz, *Distance functions on digital pictures*, Pattern Recognition **1** (1968), 33–61.
- [SGG⁺00] Pedro V. Sander, Xianfeng Gu, Steven J. Gortler, Hughes Hoppe, and John Snyder, *Silhouette clipping*, Proceedings of the 27th annual conference on Computer graphics and interactive techniques SIGGRAPH 2000, 2000, ISBN 1-58113-208-5, pp. 327–334.
- [SGGM06] Avneesh Sud, Naga Govindaraju, Russell Gayle, and Dinesh Manocha, *Interactive 3d distance field computation using linear factorization*, SI3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games (New York, NY, USA), ACM Press, 2006, ISBN 1-59593-295-X, pp. 117–124.
- [Sin05] Ida-Maria Sintorn, *Segmentation methods and shape descriptions in digital images*, Ph.D. thesis, Swedish University of Agricultural Sciences, 2005.

- [Sve01] Stina Svensson, *Representing and analyzing 3d digital shape using distance information*, Ph.D. thesis, Swedish University of Agricultural Sciences, 2001.
- [TKZ⁺04] M. Teschner, S. Kimmerle, G. Zachmann, B. Heidelberger, L. Raghupathi, A. Fuhrmann, Marie-Paule Cani, François Faure, N. Magnetat-Thalmann, and W. Strasser, *Collision detection for deformable objects*, Eurographics State-of-the-Art Report (EG-STAR), Eurographics Association, 2004, pp. 119–139.
- [Wat00] Alan Watt, *3d computer graphics*, Addison-Wesley, Harlow, GB, 2000, ISBN 0-201-39855-9.
- [WLC⁺03] Nathaniel Williams, David Luebke, Jonathan D. Cohen, Michael Kelley, and Brenden Schubert, *Perceptually guided simplification of lit, textured meshes*, Proceedings of the 2003 Symposium on Interactive 3D Graphics SI3D'03 (New York, USA), ACM Press, 2003, Session 5: simplification and meshes, pp. 113–121.
- [XV96] Julie C. Xia and Amitabh Varshney, *Dynamic view-dependent simplification for polygonal models*, IEEE Visualization'96, 1996, ISBN 0-89791-864-9, pp. 335–344.

| Citation |
|---|
| Susana Mata, Luis Pastor, and Angel Rodriguez <i>View-Dependent Extraction of Contours with Distance Transforms for Adaptive Polygonal Mesh Simplification</i> , Journal of Virtual Reality and Broadcasting, 3(2006), no. 4, December 2006, urn:nbn:de:0009-6-7643, ISSN 1860-2037. |