Simplification of implicit skeletal models

L. Lucas† and S. Prévost

LERI - Laboratoire d'Etudes et de Recherches en Informatique, Université de Reims Champagne-Ardenne, Reims, France

Abstract

In this paper, we describe a hierarchical representation of unions of balls (UoB) applied to volume graphics. We present an algorithm that generates stable implicit volumes at different levels of resolution in the form of primitives of overlapping spheres from various data sources such as volumetric datasets and other existing models. This is achieved as follows. First, an unstructured set of valued points called "UoB skeleton" is extracted from an exact Euclidean Distance Transform (implicits are centered at the skeletal voxels). Next, the skeletal points are connected and arranged in a "structural graph" called spanning graph, which can be used to obtain simplified multi-scale models. This simplification process consists in gradually removing nodes in this graph while respecting topological and geometrical constraints. The goal is to build an interactive system of visualization for the analysis of volumetric data. The speed of treatment associated with a good visualization should enable to achieve a 3D survey of a natural object in an interactive manner. The method has been successfully applied to both synthetic and real data (medical imaging).

Keywords: *Volume graphics, skeleton shape, multiresolution, graph decimation, optimization.*

1. Introduction

Advances in both computing and scanning device technologies are making simulation and representation of large-scale volumetric datasets at different levels of resolution an important part of research in scientific visualization today. Due to the huge size of data, which affects both storage requirements and visualization times, multiresolution tools provide a good way of reconstructing an object from a regular set of points sampled from its boundary and allow the user to view a dataset at a coarse level-of-detail (*LOD*). Several methods can compute and display surface and volumetric models. In order to display, transform and compare objects, it is often convenient to use different representations. If fundamental desired properties are efficiency of computation, storage and display, other properties like the ability to compute simplified models could be used to obtain other more attractive volume visualization systems.

Generally speaking, volumetric models are given by a function $\mu(p)\ p \in E^n$, whose domain is an n-dimensional region of an n-space. Usually μ is known only at discrete points on a regular grid and is represented by this set of sampled values. Implicit models provide a powerful framework for this kind of description primarily because they unify surface and volume representations which are often antagonistic one towards the other. Volume data can be re-interpreted as samples of an implicit function which allows us to classify points in relation to the surface enclosed in the volume. The shape is then characterized not only by points on the surface, but also by points in the vicinity of the surface. This explicit dependency of the geometric object X on the space E in which it is embedded, in addition to the well-known relation existing between continuous and discrete representations of X always in E, can be used in most computational tasks related to modeling and rendering. If we consider the previous remarks as relevant, implicit shape modeling represents a good 3D reconstruction and visualization alternative to the traditional surface models of an object. Among its advantages, we can mention: the capacity to represent shape in a compact manner, the flexibility to manage shapes efficiently and the appropriateness to represent "natural" objects. The surface S of such an object can then be expressed as a function of the object skeleton $Sk(X)$ by the following relation (see

† Correspondence to: laurent.lucas@univ-reims.fr, Rue des Crayères, BP 1035, 51687 Reims Cedex 2.

Figure 1):

$$\begin{aligned} S_{Sk(X)} &= \{p \in \mathbb{R}^3 \mid f(p) - \varepsilon = 0\} \text{ where} \\ f(p) &= \Sigma_i f_i(p) \end{aligned} \quad (1)$$

where f is a potential function dependant of the distance separating the point p to the nearest skeleton points $s_i \in Sk(X)$, f_i the potential functions associated to the s_i implicit skeletal primitive, and ε a constant called iso-value.

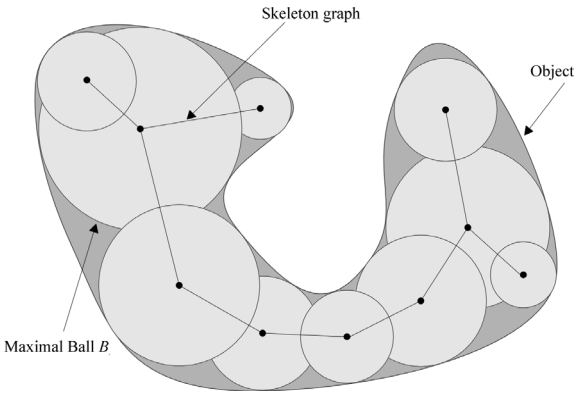

Figure 1: *Principle of shape description by skeletal representation.*

Muraki [1] was the first to use this kind of process to rebuild shapes. The principle consists in minimizing an energy function coding the distance between the raw data and an implicit surface defined as a blobby model. More recently, this approach was extended [2, 3, 4] to provide robust solutions to the reconstruction problem.

Our method introduced in [5] can be presented as a natural extension of these works to multiresolution. It consists essentially of three parts and is organized as follows. First, a skeleton is determined as an unstructured set of valued points by an exact skeletonization method based on a euclidean metric. The shape descriptor obtained at this stage satisfies three essential criteria:

- reversibility: an object must be perfectly reconstructed from its skeleton,
- homotopy: the topology of an object must be preserved,
- minimality: in the sense that no primitive B_j is completly included in a subset E' of an $UoB(E)$ where $E' = E \setminus \{B_j\}$ [6]. This constraint is defined on a set of balls by the relation $\forall B_j \in UoB(X) \mid B_j \not\subset (\cup_{i=0}^{k} B_i - B_j)$

The resulting set of voxels (UoB) are then connected into a graph of primitive-nodes and edges. This graph, called spanning graph (SG), initially containing all the primitives of UoB can be simplified to obtain multi-scale models. Figure 2 illustrates this process. The input is a regular segmented data volume. The result is a simplified geometric object, which may either be used and visualized as implicit model or revoxelized or still triangulated. The combination of skeleton and distance information provides an excellent way to both reconstruct and directly visualize [4] the shape of the original object. The method is computationally efficient and can handle data at different resolutions while taking advantage of data organization to speed up operations during rendering.

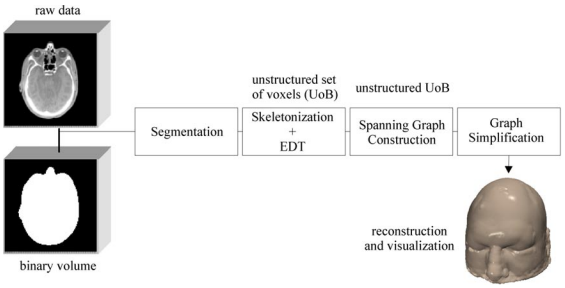

Figure 2: *General presentation of our visualization pipeline.*

The paper is organized as follows. Sections 2 through 3 describe each step in the process of shape descriptor generation and in particular the spanning graph generation. Next, in section 4, the decimation process itself is presented. Experimental results on the construction of the multiresolution model, like multiresolution visualization are reported in section 5. Finally, some conclusions and topics for future research are given in section 6.

2. Shape descriptor extraction

The first step of this pipeline (see Figure 2) is the skeletonization of the segmented binary volume. A consise definition of the skeleton in the continuum was given by [7], who postulated the well-known grassfire analogy. Many skeletonization algorithms have been published, especially in the image processing domain (see [8] for a survey). In $\mathbb{Z}^3$, a standard method to perform skeletonization is to use a distance transformation (DT) where values for each voxel within the object are defined by the minimum distance to the shape boundary. The local maxima of this transformation is the skeleton. It is also related to the set of the centers of maximal balls (CMB) which are the locus of points centered within the object.

The algorithm we use, called reversible euclidean distance transformation ($REDT$) [9, 10], is based on this CMB principle. The EDT algorithm consists of n one-dimensional local operations performed serially, each of which corresponds to the direction of each coordinate axis. It does not use vector propagation, nor fixed template strategies and still always gives exact euclidean distance stored as an array of squared distance values. The $REDT$ algorithm consists in extracting the skeleton from the EDT. The result of this transformation is a set of points p which satisfie the following relation:

$$\forall p \in Sk(X), \exists m \in X | (p-m)^2 < EDT(m) \quad (2)$$

This set of points is then filtered to respect the minimality property [6] already defined in the introduction section. The pseudo-code algorithm is given below:

EDT($B = \{b_{ijk}\}$) // input binary volume of size L×M×N
Begin
 derive from B a volume $G = \{g_{ijk}\}$
  with $g_{ijk} = \min_x\{(i-x)^2; b_{xjk} = 0, 1 \leq x \leq L\}$
 derive from G a volume $H = \{h_{ijk}\}$
  with $h_{ijk} = \min_y\{g_{iyk} + (j-y)^2, 1 \leq y \leq M\}$
 derive from H a volume $S = \{s_{ijk}\}$
  with $s_{ijk} = \min_z\{h_{ijz} + (h-z)^2, 1 \leq z \leq N\}$
 return S
End

REDT($B = \{b_{ijk}\}$) // input binary volume
Begin
 let DT = EDT(B)
 let $X = \{x_{ijk}\}$, $Y = \{y_{ijk}\}$, $Z = \{z_{ijk}\}$, $R = \{r_{ijk}\}$
 for all voxels $p_{ijk} \in DT \mid DT[p] > 0$
  for all voxels $q_{ijk} \in DT \mid (q-p)^2 < EDT[p]$
   if $R[q] < Q[p]$ then
    $X[q] = i, Y[q] = j, Z[q] = k, R[q] = DT[p]$
   endif
 for all voxels $p_{ijk} \in DT \mid R[p] > 0$
  $Sk = \{sk_{x_{ijk}y_{ijk}z_{ijk}}\}$
 return Sk
End

The ouput of this stage is a unstructured set of valued points p defining a UoB of balls B centered out of p and with the radius $EDT[p]$.

3. Spanning graph construction

Because the UoB skeleton $Sk(X)$ of an object X previously extracted is not in a format easily amenable to manipulation, a structuration stage is then performed to create a weighted undirected graph where all skeleton balls are nodes and where adjacent balls are connected by edges represented in Figure 3 by line segments. This spanning graph - noted SG - is obtained from $Sk(X)$ through the construction of the minimum spanning tree (MST) of $Sk(X)$ where two balls B_1 and B_2 are adjacent in SG if $B_1 \cap B_2 \neq \emptyset$. SG is then defined as an acyclic planar graph and corresponds to a connected tree. If as a first approximation, this structure is sufficient to preserve topological properties of X in so far as X is represented as a genus-0 surface, it is insufficient for higher genus surfaces. In order to preserve the initial topology of the studied object, the knowledge of holes in this object is significant. The purpose is then to modify SG so that it authorizes as many loops as existing holes. At this stage, SG is not an acyclic graph but verifies the Euler formula defined as $H = 1 - V + E$ with V the number of nodes, E the number of edges and H the number of holes. Practically, that consists in adding edges in SG. These edges are deduced for each hole of an object from an homotopic kernel introduced by [11, 12]. An homotopic kernel is a set which contains no α_n-simple point, an α_n-simple point being able to be seen as an "inessential" element for the topology. In our case, this principle was extented to balls to be able to characterize the homotopic kernel of a UoB. The algorithm we use consists in identifying each hole [13] first and then shrinking the UoB in the vicinity of the hole defined as partitions of a digital Voronoi diagram directly obtain from the EDT. The process terminates when no deletions occur (see Figure 3 for the homotopic kernel of the vertebra). When the problem of holes is solved, the graph can be simplified by removing nodes. The reduction operation is a pyramidal deterministic process since all LODs are coded in SG. and stored as a stack of successively reduced graph $(G_0, G_1, G_2, G_3, ...)$. Each graph is built from the graph below by selecting a set of vertices named surviving vertices and mapping each non-surviving vertex to a surviving one. The initial graph $G_0 = (V_0, E_0)$ is defined from SG where each vertex V_0 are associated to each ball of UoB and where edges E_0 represented the adjacency relationship defined on V_0. The graph $G_{l+1} = (V_{l+1}, E_{l+1})$ defined at level $l+1$ is deduced from the graph defined at level l by the following steps:

- the selection of the vertices of G_{l+1} among V_l (these vertices are the surviving vertices of the decimation process),
- a link of each non-surviving vertex to a surviving one. This step defines a partition of V_l. To ensure that each non-surviving vertex is adjacent to at least a surviving one the following constraint $\forall v \in V_l - V_{l+1} \exists v' \in V_{l+1} : (v, v') \in E_l$ is checked.

4. Application to multiresolution

The concept of multiresolution modeling is not particularly new, but most of the research in this domain has been conducted recently. A multiresolution model is defined as a model capable of keeping a wide range of LODs of an object and of reconstructing any one of these levels on demand. Many approaches based on different strategies, such as vertices decimation, edges contraction, envelope simplification or wavelet surface representation have been recently proposed in literature for the multiresolution management of surfaces, while multiresolution volume data management is still in an unsufficiently developed stage [14, 15]. In our case, the creation of a succession of representations increasingly simplified, such as the differences between two successive LODs which are the least perceptible, rests on some rules defined as follows:

- structure of LODs: they are stored in a unique structure coding the initial model as well as its different LODs simultaneously.
- preservation of the meaningful shapes of an object: in order to minimize differences from a LOD to another, our decimation algorithm must take into account the meaningful parts of the object to preserve them in the sim-

plified representations. The preservation of shapes essentially governs the choice of simplification parameters.
- measurement and controll of the simplification process: in order to better preserve the forms, the measure of the approximation error must be local and take into account the configurations of a restricted number of primitives. However, in order to facilitate the evaluation by the user, it is necessary to control globally the cost of the simplification. Two measurements are given to the user to control the degree of simplification that he wishes to get. The first one gives us a rate of recovering compared to the reference object, the second one characterizes the greatest detail missing. It is computed from an Hausdorff distance between two sets of balls (see Figure 3).
- topological modification: the initial topology must be preserved as well as holes in models to produce exploitable results.

The algorithm we use is based on the simplification of *SG*. It operates directly on the shape descriptor while removing a subset of the geometric skeleton step after step. To achieve this, two operators were developed. The first one, which mainly operates as a topological operator, corresponds to the principle of erosion used in mathematical morphology. The erosion of size B of a weighted graph G is defined by the structuring function ε_B equivalent to:

$$\forall y \in G \;\; \varepsilon_B(G) = \{\inf(G(x)), x \in B_y\} \tag{3}$$

Associated with this first operator (known as structuring decimation), the second operator allows to remove implicit low potential primitives by modifying the scheduling of nodes candidate to the suppression in *SG*. This contribution characterization of each implicit primitive used to represent the surface of the object can be considered primarily in two manners. A first analytical solution will locally bind us to define the parametric characteristics of the surface in order to evaluate its area. The second approach will consist in a discrete approximation of this same calculation. This is carried out with a simple spatial partitioning method which amounts to discretizing the workspace. The algorithm then consists in sampling, at each position on a regular grid, discrete potential values which are then stored. The information contained in each node of this grid allows us to deduce which implicit primitive produces the lower potential and so to sort them. This discrete formulation offers several advantages. First, the evaluation cost of the potential of a point in space can be performed in constant time, independently of the surface complexity. Secondly, even if a continuous approach provides a more precise surface characterization, the approximation that we make does not modify the order relationship which we seek to establish with, in addition, a better calculation time. This second operator is initialized from a parameter describing the number of primitives which must be removed at each decimation step. The whole (the simplification process) operates according to a pyramidal strategy decimation. In image analysis [16, 17] such a representation is defined as a succession of images with decreasing resolution. Usually, the initialization of such a process is established on the basis of the pyramid, namely the original image. In the simplest case, the various levels of the pyramid are obtained starting from the previous ones by a simple filtering operation followed by an operation of decimation.

The algorithm that we propose consists in gradually removing nodes in *SG*. The structure of the pyramid is determined here by relations of vicinities on a given level and by relationships between two consecutive levels. Each level i of the pyramid can be described by the graph of vicinity $G_i =< V_i, A_i >$ where the sets of nodes V_i correspond to the points of level i, and where $A_i \subseteq V_i \times V_i$ expresses the relationships of vicinity between the points. Two nodes p and $q \in V_i$ are adjacent in G_i if they are close in the structure. In practice, the top level graph will correspond to *SG*.

5. Results and Discussion

The performance of our system was evaluated on three datasets, representative of two classes of topological shapes. In order to facilitate comparisons with other proposals, datasets were chosen as they are commonly used in the volume rendering field:

- Bunny, a $256 \times 256 \times 256$ dataset which represents a rabit, extracted and re-voxelized from a traingular mesh of 69451 faces (file: 4.41 Mb)
- Mush, a $91 \times 105 \times 61$ dataset which represents a mushroom;
- Vertebra, a $359 \times 359 \times 49$ dataset which represents a human vertebra provided by the TIMC-IMAG lab (Grenoble, France);

All results presented here have been calculated on an Intel Pentium III 450 MHz PC/Linux platform with 256 Mb of memory or on a cluster constituted of 6 Pentium III (450 and 800 MHz) inter-connected on a local ethernet 100M network.

Visually, the quality of results obtained from our algorithm seems to be relevant. Figure 3 shows that the shape legibility remains effective (without optimization) for this algorithm down to 2800 iterations where only 32% of initial primitives are used to reconstruct the object's surface from the rabbit and 19% after 60 iterations for the mushroom. From a quantitative point of view, the algorithm seems to be efficient. However, the quality of LODs is not necessary usable. Gaps as the safeguarding of the shape of the object sometimes make the results difficult to appreciate in particular when the level of decimation increases. This observation made, we tried to correct this drift which is characterized by a volume variation on the fall between two consecutive LODs. Several solutions were considered to solve this problem such as for example, re-examine the decimation process by integrating other heuristics than those used (exclusively

geometrical), or by modifying the blending functions associated with the implicit primitives, or even while optimizing the positions of each primitive along the morphological graph. If the first solution seems theorically more direct, in practice, it induces technical problems of implementation. Indeed, that comes to add some constraints in the system like shape memory to drive the decimation process in a better way. In the same way, if the second solution seems efficient when the transformation is done with a constant number of primitives [18], that becomes more delicate when this number changes over the time. So, we chose the third solution primarily to allow us to control locally the optimization process while focusing itself on search of global extrema. This stage, called relaxation, is based on a genetic algorithm regularization [19]. Regarding the structure of LODs, our algorithm has the main advantage of using a single structure coding the set of the various levels of details, contrary to other techniques which generate as many different data bases as levels. We can observe that the whole of the simplification processes are continuous, which leads us indeed to obtain geomorph transformations. The more numerous the stages of decimation are, the more the models obtained are faded compared to the initial data. We can however notice that a rate of decimation equivalent to 70% on average still preserves the legibility of the forms, which becomes more problematic after this limit. Beyond, the resulting 3D shapes can be used for an iconic visualization of the features [20]. Primarily relying on an intuitive specification, the control of the simplification process is guided by two error measurements (global and local) in addition to a visual control. The first one gives us a rate of recovering compared to the reference object, the second one characterizes the greatest detail missing and is computed from an Hausdorff distance. Finally, on the topological aspect, we can say that our algorithm preserves the initial topology of the studied object. In this case, the spanning graph will have more loops than holes.

Concerning now the rendering, it is performed by a special volume rendering engine based on an implicit accelerated distance-based ray tracing algorithm. The system developped is designed as a portable message passing programming system called K10, used to link separates host machines according to a principle equivalent to PVM. K10 is an object-oriented client-server library based on an asynchronous communication protocol by which the processes exchange data, developed for high performance volume graphics. The choice to develop a specific interprocess communication system rest on the following considerations: first, the data are immutable (even in a multiresolution framework since only one single structure is handled) so, write protection is not required and communication requirments are greatly simplified. Secondly, the rendering algorithm is an image order algorithm. This engine is decomposed into 3 components: a display driver, an image server and n raytracer clients. The display driver handles user interaction as a "virtual trackball" and paints the pixels on the screen. The image server decompose the image into components and distribute them to the distributed ray-tracers clients.

6. Conclusion and Future Work

In this paper, we have described a general framework to define hierarchically, optimize and visualize three-dimensional voxel object representations. Based on the characterization and the simplification of a skeletal graph, our multiresolution implicit modeling for volume visualization toolkit seems able to provide satisfactory representations. However, we think that it can still be improved significantly. One of the main limitations is that it remains difficult to anticipate the quality of results out of these simplifications. This is due, for a large part, to the fact that the control parameters involved are not intuitive. Moreover, this algorithm relies mainly on qualitative criteria based on the human vision system, and not on a quantitative evaluation at least concerning the decimation process. As a result, the estimation of the amount of simplification compatible with high quality results still requires human intervention. Therefore, plans should integrate in the system some constraints like shape memory to drive the user of our platform in a better way.

Acknowledgements

This research is supported by a grant from the ARERS in collaboration with the Prof. Dominique Ploton (MeDIAN CNRS FRE 2141).

References

1. S. Muraki. Volumetric shape description of range data using " blobby model ". *In ACM Computer Graphics (SIGGRAPH'91 Proc.)*, 25:227–235, 1991.
2. E. Bittar, N. Tsingos, and M.P. Gascuel. Automatic reconstruction of unstructured 3d data : Combining a medial axis and implicit surfaces. *Computer Graphics Forum (Eurographics'95 Proc.)*, 14:457–468, 1995.
3. E. Ferley, M.P. Cani-Gascuel, and D. Attali. Skeletal reconstruction of branching shapes. *Computer Graphics Forum*, 16(5):283–293, 1997.
4. S. Prévost and L. Lucas. An efficient voxel-based visualization system from an implicit skeletal surface characterization. *J. of Visua. and Comp. Animation*, 11:39–49, 2000.
5. L. Lucas, E. Bittar, and S. Prevost. Multiresolution modeling and implicit skeletal surfaces. In *IS'99, joined Eurographics ACM workshop*, pages 123–130, 1999.
6. G. Borgefors and I. Nystrom. Efficient shape representation by minimizing the set of centres of maximal discs/spheres. *Pattern Recognition Letters 18*, pages 465–472, 1997.

7. H. Blum. A transformation for extracting new decriptors of shape. *Models for the perception of speech and visual form*, pages 362–380, 1967.

8. S.F. Vidal and G. Malandain. Squelettes euclidiens d'objets discrets n-dimensionnels. Technical Report 2771 programme 4, INRIA, 1996.

9. T. Saito and J. Torikawi. New algorithms for euclidean distance transformation of an n-dimensional digitalized picture with applications. *Pattern Recognition*, 27(11):1551–1565, 1994.

10. T. Saito and J. Torikawi. Reverse distance transformation and skeletons based upon the euclidean metric for n-dimensional digital binary pictures. *IEICE Trans. Inf & Syst.*, E77-D(9):1005–1016, 1994.

11. C. Lohou and G. Bertrand. Poset approach to 3d parallel thinning. *SPIE Vision Geometry VIII*, 3811:45–56, 1999.

12. G. Bertrand. New notions for discrete topology. In *DGCI'99, Lecture Notes in Computer Science*, volume 1568, pages 218–228, 1999.

13. Z. Aktouf, G. Bertrand, and L. Perroton. A three-dimensional holes closing algorithm. *DGCI'96, Lecture Notes in Computer Science*, 1176:36–48, 1996.

14. T. He, L. Hong, A. Kaufman, A. Varshney, and S. Wang. Voxel-based object simplification. *In IEEE Visualization'95 Proc.*, pages 296–303, 1995.

15. P. Cignoni, C. Montani, E. Puppo, and R. Scopigno. Multiresolution modeling and visualization of volume data. *IEEE Trans. on Visual. and Comp. Graph.*, 3(4):352–369, 1997.

16. P. Meer. Stochastic image pyramids. *CVGIP*, 45:269–294, 1989.

17. J.M. Jolion and A. Montanvert. True adaptative pyramid : a framework for two images analysis. *CVGIP*, 55(3):339–348, 1992.

18. M.P. Cani-Gascuel and M. Desbrun. Animation of deformable models using implicit surfaces. *IEEE Trans. on Visualization and Computer Graphics*, 3(1):339–348, 1997.

19. S. Prevost, L. Lucas, and E. Bittar. Multiresolution and shape optimization implicit skeletal model. In *WSCG'01, Plzen (République Tchèque*, volume SC, pages 8–15, 2001.

20. F. Reinders, M.E.D. Jacobson, and F.H. Post. Skeleton graph generation for feature shape description. *in Data Visualization 2000, Springer Verlag*, pages 73–82, 2000.

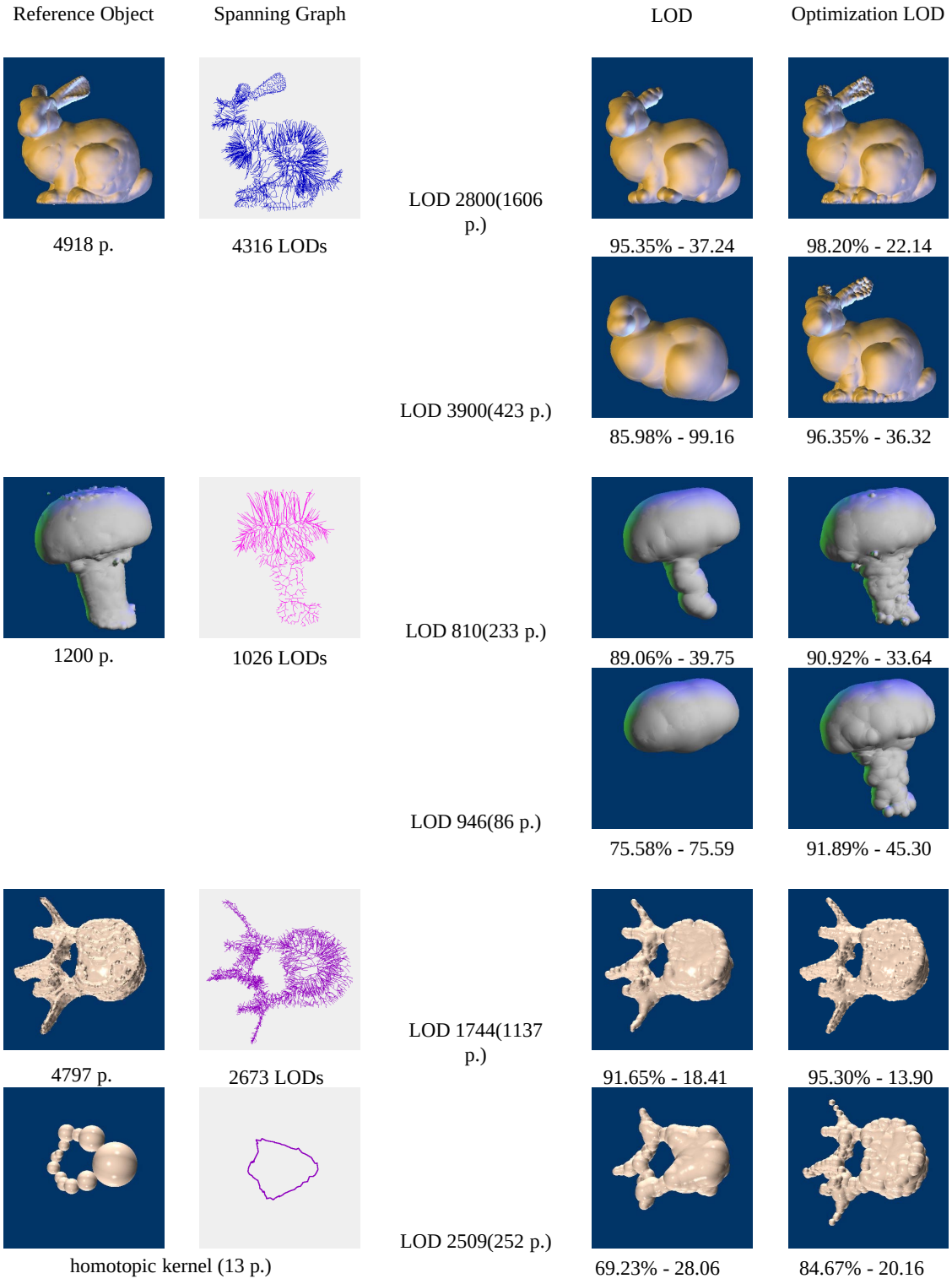

Figure 3: *Some results on three different dataset.*
the first column corresponds to the reference objects with their numbers of primitives. The second column represents the spanning graph with the maximal number of LOD. The two other columns shows us different LOD with their global and local error measurements before and after optimization.