Hierarchical Discrete Medial Axis for Sphere-Tree Construction

No Author Given

No Institute Given

Abstract. In discrete geometry, the Distance Transformation and the Medial Axis Extraction are classical tools for shape analysis. In this paper, we present a Hierarchical Discrete Medial Axis, based on a pyramidal representation of the object, in order to efficiently create a sphere-tree, which has many applications in collision detection or image synthesis.

1 Introduction

In interactive environments, hierarchical representations are classical tools in many applications, *e.g.* in image synthesis, multi-resolution representations and interference detection, as allowing a fast interaction location. The Bounding Volume Hierarchies (BVH) consist in coverage with an increasing number of simple volumes (spheres [19, 16, 10], axis-aligned bounding boxes [20], oriented bounding boxes [12], ...) on different levels, ensuring that each part of the object covered by a children node must be covered by the parent node. Hence the collision detection between two hierarchical models is computed by a recursive process in which overlaps between pairs of primitive volumes are tested. The choice of the primitive volume is important for the BVH properties and sphere-trees are very interesting for interference detection [12]. Nevertheless, as they are bad estimators of the object geometry, the hierarchy needs an effective construction of the tree, reducing the error between the object and the associated set of spheres.

In computational geometry, first algorithms for sphere-tree construction were based on Octree data structures which consist in recursive spatial subdivisions of the object [16, 10, 13]. Later, efficient algorithms were proposed, based on the object Medial Axis (MA) [4], which corresponds to a skeletal representation of the object. MA extraction is based in a Voronoi Diagram [14, 6, 7], and the sphere-tree is produced by complex optimization heuristics to reduce the number of spheres.The error is controled by an approximated Haussdorf distance.

In discrete geometry, the study of multiresolution representation of digital objects has been carried out using an homotopic thinning [18]. However this approach should be considered as a medial axis filtering instead of a hierarchical reduction of the set of balls. Our goal is to develop a hierarchical structure which is flexible with respect to the reversibility of the construction. Indeed the Discrete Medial Axis (DMA) is a convenient tool to represent objects in digital space, thanks to its reversibility : from the DMA balls, we can exactly reconstruct the original shape. The exact euclidian DMA can be efficiently computed from

Distance Transformation (DT) [9]. Furthermore, a discrete approach benefits from the exact computation of the error between the object and the hierarchy (at each level) with a Hamming distance.

In this paper, we present an original method for sphere-tree construction (sketched in Figure 1): the sets of spheres at each level are obtained by a DMA extraction at different object resolution levels in a regular pyramid. Preliminaries are presented in Section 2. The sphere-tree construction is achieved by linking the spheres on consecutive levels. This method is first defined for a reversible model for volume synthesis (Sections 3.1), however it can be easily adapted for interactive environments (Section 3.2). Experiments in Section 4 also show that this last modification reduces the error. Finally we obtain a d-dimensional generic sphere-tree computation in linear time for any discrete distance and including pyramidal model.

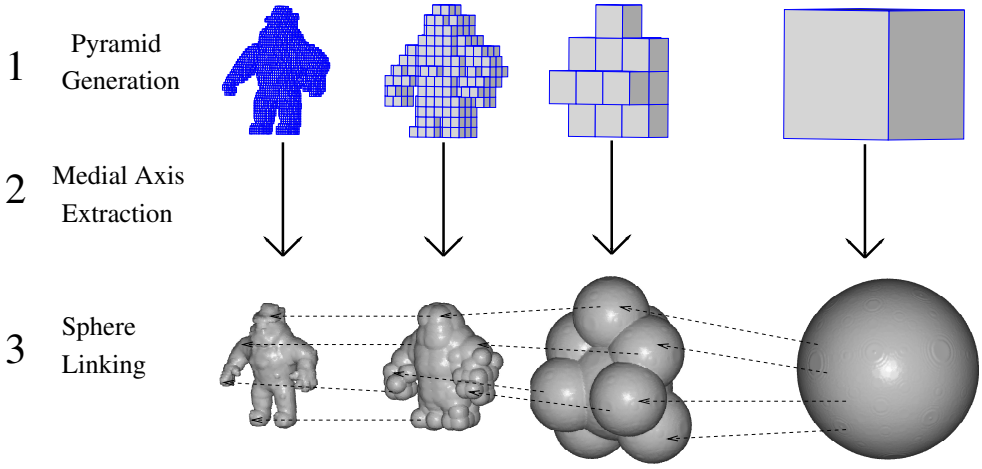

Fig. 1. The main stages of the sphere-tree construction.

2 Preliminaries

As sketched in the introduction, the methods is based on a regular pyramid, where set of spheres are produced by a DMA extraction on each level. This section presents these preliminary stages.

2.1 Pyramidal Model

In image analysis, the pyramidal structure is a convenient tool [8, 15]. In dimension 2, a pyramid $\mathcal{P}$ of depth $N+1$ can be defined by a set of 2D images $\{\mathcal{F}_0, ..., \mathcal{F}_N\}$. The regular pyramid construction is a bottom-up process: $\mathcal{F}_N$ is

the original image, the upper levels are its representations at lower resolutions in a quad-tree approach; the pixel size at each level is 4 times bigger than at next level. The pixel color is based on the 4 pixels and computed by a transfer function (cf Figure 2).

We can generalize this process to any dimension d with an integer factor f for the voxel size expansion. In other words, a voxel v' at level $L-1$ contains f^d voxels $\{v_1, ..., v_{f^d}\}$ at level L. So, the transfer function, denoted $\mathcal{M}$ (standing for Model), is a mapping between these voxels:

Definition 1. *$\mathcal{M}$ is a model between $\mathcal{F}_L$ and $\mathcal{F}_{L-1}(\mathcal{F}_{L-1} = \mathcal{M}(\mathcal{F}_L))$ if* $\forall v',\ \exists\{v_1, ..., v_{f^d}\} \setminus \mathcal{F}_{L-1}(v') = \mathcal{M}(\mathcal{F}_L(v_1), ...\mathcal{F}_L(v_{f^d}))$

In the proposed bounding volume hierarchy approach, the original object should be completely covered by each level of the hierarchy. So, we used the OR-Model: each voxel at level $L-1$ belongs to $\mathcal{F}_{L-1}$ when at least one of the f^d voxels that it contains (at level L) belongs to $\mathcal{F}_L$. Figure 2 shows an application of this model on 2D images.

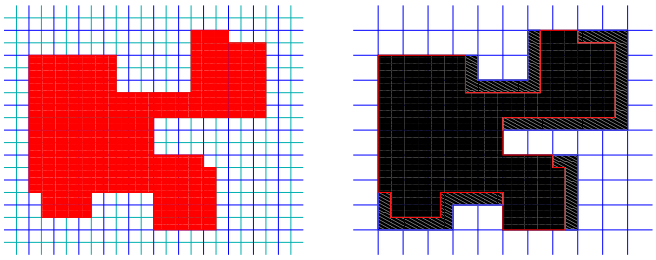

Fig. 2. Consecutive levels on a regular pyramid with the OR-Model.

2.2 Discrete Medial Axis and Discrete Power Diagram

Discrete Medial Axis. The Medial Axis is a shape descriptor first presented by Blum in 1967 [4] in order to simulate the wavefront propagation from its boundary (*prairie fire model*). The medial axis is defined by (1) the locus of of points equidistant from two sides of the object, or (2) the locus of centers of maximals balls included in the object (a ball is *maximal* if it is not included in any other ball included in the object) [17].

In binary images, the Discrete Medial Axis (DMA) can be efficiently extracted from a Distance Transformation (DT). It consists in labeling each voxel of an object with distance of the closest voxel of the complement (background). In other words, the DT value at a voxel v corresponds to the radius of the largest discrete ball centered in s included in the object. From the Euclidean DT, linear time algorithms exist to extract the set of maximal balls [9].

Discrete Power Diagram. In computational geometry, the power diagram (also known as the *Laguerre diagram*) is a generalization of the Voronoi Diagram [3]. This tool is widely used for ball interaction computation and surface reconstruction [5, 2, 1]. We consider a set of sites $\mathcal{S} = \{s_i\}$, such that each point s_i is associated with a radius r_i. The power $\sigma_i(p)$ of a point p according to the site s_i is given by:

$$\sigma_i(p) = d(p, s_i) - r_i \tag{1}$$

If $\sigma_i(p) < 0$, p belongs to the ball of center s_i and radius r_i. The power diagram is based on the metric induced by σ and is a decomposition of the object into cells $\mathcal{C} = \{c_i\}$ associated with each site s_i such that:

$$c_i = \{p \in \mathbb{R}^d : \sigma_i(p) < \sigma_j(p), i \neq j\} \tag{2}$$

In discrete geometry, the power labeling is defined as the power diagram labeling of grid points. More precisely, we assign to each grid point the index of the cell it belongs to. In [9], authors have illustrated the links between MA balls and power diagram cells.

3 Sphere-tree construction

As the set of maximal spheres can be extracted at each level of the pyramid, it remains to link the spheres of consecutive levels to complete the sphere-tree construction. The following section first presents a graph construction by a simple linking process, and its reduction in different sphere-trees.

3.1 Power diagram and Sphere-DAG

In order to link a sphere s at level N and a sphere t at level $N-1$, a simple intersection test can be used to express the covering of a part of s by t. We note $t \rightarrow s$ an edge between t and s:

$$t \rightarrow s \Leftrightarrow t \cap s \neq \emptyset \tag{3}$$

As these edges are always oriented from a sphere t to a sphere s at next level, the graph defined by the equation is a Direct Acyclic Graph (DAG), where nodes are DMA spheres at different levels. The DAG construction needs to test the intersection between two spheres of consecutive levels, so computation time is in $O(n^2)$ for n spheres at the lower level. However, in a correct hierarchy, each part of the object does not need to be covered by a large set of spheres. Hence, this Sphere-DAG is too exhaustive for an efficient sphere-tree simplification. In order to reduce the number of spheres of the complete Sphere-DAG, we use the discrete power diagram, defined in Section 2.2. Indeed it is a voxel labeling of discrete points by the sphere which best covers it. Let v be a voxel of $\mathcal{F}_N$ at level N. As the upper representation $\mathcal{F}_{N-1}$ at level $N-1$ is built with a model $\mathcal{M}$, v is included in a voxel v'. We denote $v' = \mathcal{R}(v)$. Including models, like

the "OR" model defined in Section 2.1, ensure that $\mathcal{F}_{N-1}(v') = 1$. So we can compare the power diagrams $\mathcal{C}_N$ (from $\mathcal{F}_N$) and $\mathcal{C}_{N-1}$ (from $\mathcal{F}_{N-1}$). If v belongs to the cell $\mathcal{C}_N(s_i)$ and v' to $\mathcal{C}_{N-1}(t_j)$, the sphere s_i (at level N) covers a part of $\mathcal{F}_N$ which includes the voxel v, and the representation of this part in $\mathcal{F}_{N-1}$ is covered by the sphere t_j. In other words, t_j covers a part of s_i, and we represent it by linking t_j and s_i.

Definition 2. *t is a parent sphere for s ($t \rightarrow s$) in the Sphere-DAG if* $\exists v \in \mathcal{F}_N \setminus (v \in \mathcal{C}_N(s) \wedge \mathcal{R}(v) \in \mathcal{C}_{N-1}(t))$

By an overlapping of $\mathcal{C}_N$ and $\mathcal{C}_{N-1}$ we can detect all relations by only one scan on each voxels at level N (cf Figure 3). The algorithm 1 is generic for objects in dimension d and for any including pyramidal model.

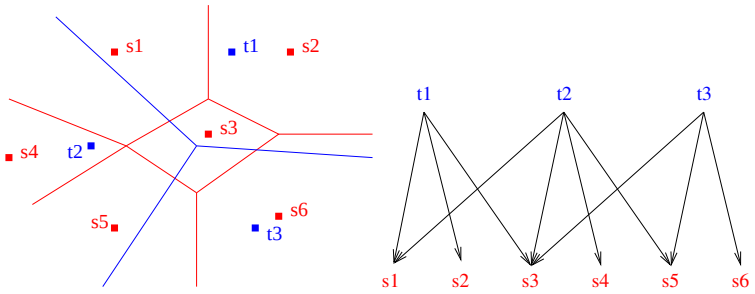

Fig. 3. Overlapping of power diagrams $\mathcal{C}_N$ and $\mathcal{C}_{N-1}$ and the associated sphere DAG between $\mathcal{F}_{N-1}$ and $\mathcal{F}_N$. For example, the cell s_1 in $\mathcal{C}_N$ is covered by both t_1 and t_2 in $\mathcal{C}_{N-1}$, so we have $t_1 \rightarrow s_1, t_2 \rightarrow s_1$.

Algorithm 1 Generic Algorithm for Sphere-DAG Computation

Input: $\mathcal{F}_N$ the original object,
  $\mathcal{AM}_N$ its Discrete Medial Axis and $\mathcal{C}_N$ its Discrete Power Diagram
Output: $\mathcal{T} = \{\mathcal{AM}_0, \mathcal{AM}_1, \ldots, \mathcal{AM}_N\}$ the Sphere-tree
while $|\mathcal{AM}_N| > 1$ **do**
  $\mathcal{F}_{N-1} \leftarrow \mathcal{M}(\mathcal{F}_N)$ {with $\mathcal{M}$ the pyramidal Model function}
  Extraction of $\mathcal{AM}_{N-1}$ and $\mathcal{C}_{N-1}$
  for all v such as v is a voxel $\in \mathcal{F}_N$ **do**
    $v' \leftarrow \mathcal{S}(v)$
    if $(v \in \mathcal{C}_N(s))$ **and** $(v' \in \mathcal{C}_{N-1}(t))$ **then**
      Linking the sphere $t \in AM_{N-1}$ with $s \in AM_N$
    end if
  end for
  $N \leftarrow N - 1$
end while

Theorem 1. *The generic Sphere-DAG construction process is linear in the number of voxels in $\mathcal{F}_N$.*

Proof. We consider that the original object $\mathcal{F}_N$ is composed of m voxels (or c^d voxels if the object is bounded by a cube of side c in dimension d). The Medial Axis extraction and power diagram computation are in $O(c^d)$ [9]. Furthermore, the construction of $\mathcal{F}_{N-1}$ and the linking between the two sets of spheres can be also computed in one scan of voxels in $\mathcal{F}_N$. So the computation of one iteration of the while loop ($\mathcal{F}_{N-1}$ construction and sphere linking) is in $O(c^d)$. Next, we work with the object $\mathcal{F}_{N-1}$, whose size is f^d times smaller, and the computation of $\mathcal{F}_{N-2}$ is in $O(\frac{c^d}{f^d})$, and so on. Thus the overall computation time for the whole hierarchy is given by the geometric series $c^d + \frac{c^d}{f^d} + \frac{c^d}{(f^d)^2} + ... = c^d \sum_{i=0}^{N} \frac{1}{(f^d)^i}$ bounded by $\frac{c^d}{1-\frac{1}{f^d}} = c^d \frac{f^d}{f^d-1}$. Hence the generic Sphere-DAG computation process is linear in the number of voxels in $\mathcal{F}_N$. □

3.2 Reversible and Extended Sphere-Trees

Usually, in a bounding volume hierarchy each node has to cover the union of parts of the object covered by its children nodes, and not the whole volume of each child. Here, our spheres result from DMA extraction, ensuring the fact that they cover a part of the object without error. So, each sphere has to be completely covered by (at least) one parent sphere at upper level. However, this condition implies an expansion of parent spheres and a modification of the representation. Two methods can be distinguished, respecting either the pyramidal reversibility or the BVH properties.

Reversible Sphere-Tree To compute a sphere-tree from the sphere-DAG, we extract a spanning tree, keeping the parent sphere for each node which best covers it. We can easily determine if a sphere $s_1(c_{s_1}, r_{s_1})$ is covered by another sphere $s_2(c_{s_2}, r_{s_2})$ comparing the radius r_{s_2} with the distance between centers $d(c_{s_1}, c_{s_2})$ added to the radius r_{s_1}:

$$s_1 \subseteq s_2 \;\Leftrightarrow\; d(c_{s_1}, c_{s_2}) + r_{s_1} - r_{s_2} \leq 0 \tag{4}$$

However it is not sufficient if we want to know if the sphere is covered by more than one sphere.We reformulate the previous covering relation considering the σ function defined in the power diagram description (in Section 2.2):

Definition 3. *let $s(c_s, r_s)$ a sphere at level N and $t(c_t, r_t)$ at level $N-1$, the covering power $\sigma'_t(s)$ is given by*

$$\sigma'_t(s) = d(\mathcal{S}(c_s), c_t) - r_t + r_s$$

If $\sigma'_t(s) \leq 0$ then s is entirely covered by t. More precisely, the intersection between s and t increases when $\sigma'_t(s)$ is small. So, for a children sphere s with p

parents $t_1, ..., t_p$, we choose the parent t_j where $\sigma'_{t_j}(s)$ is minimal, in order to get the best covering of s. In fact, we can immediately determine the best parent: for a sphere s_i with radius r_{s_i} we can only search the minimum of the quantity $d(\mathcal{S}(c_{s_i}), c_{t_j}) - r_{t_j}$ for each parent t_j. With the computation of the power diagram $\mathcal{C}_{N-1}$, it corresponds to the function $\sigma_{t_j}(\mathcal{S}(c_{s_i}))$ of the point $\mathcal{S}(c_{s_i})$ for the site t_j. The minimum of σ is reached at t_j if the point $\mathcal{S}(c_{s_i})$ is included in the cell associated to t_j. Hence, for each sphere s_i at level N, we just need to detect the position of $\mathcal{S}(c_{s_i})$, which represents the center c_{s_i} at level $N-1$. If $\mathcal{S}(c_{s_i})$ belongs to the cell associated to the sphere t_j then t_j is the parent sphere of s_i. Figure 4 shows this computation in the overlapped power diagrams of Figure 3.

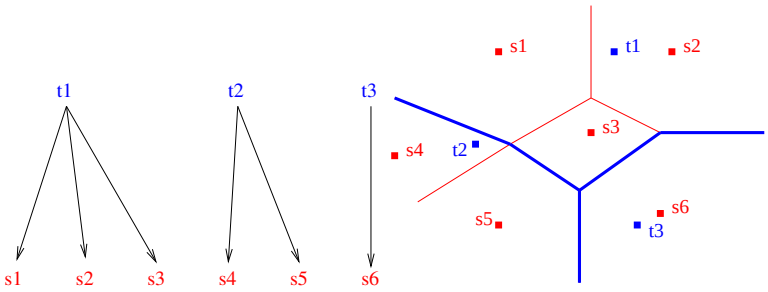

Fig. 4. Sphere-tree reduction of the previous DAG. We can also create the recovering diagram, extending each cell in $\mathcal{C}_{N-1}$ (associated with a sphere t_j) as the union of children cells.

Extended Sphere-Tree In order to respect the inclusion property, we could replace each parent sphere by the minimal bounding sphere of its children. However, the minimal covering sphere computation is not very efficient since the problem is related to the minimal enclosing ball of a set of points in dimension d [11]. In order to maintain the reversibility of multiresolution representations of the original object, we propose an original approach, which consists in extending the radius of parent spheres.

Theorem 2. *Let a sphere t and its set of children spheres $\{s_i\}$. The sphere t is a minimal bounding sphere centered at c_t for its children spheres if its radius r_t is extended by $r'_t = \sigma'_t(max)$, where max is the child sphere with maximal σ'_t.*

Proof. For two spheres s, t, from Definition 3 we know that t entirely covers s if $\sigma'_t(s) \leq 0$. Moreover, if $\sigma'_t(s) = 0$ we have $d(\mathcal{S}(c_s), c_t) + r_s = r_t$, so the sphere t is the minimal bounding sphere for s centered at c_t. Let r'_t the quantity to add to r_t in order to have a covering of s by t. So, we have:

$$d(\mathcal{S}(c_s), c_t) - (r_t + r'_t) + r_s = 0 \Leftrightarrow r'_t = d(\mathcal{S}(c_s)), c_t - r_t + r_s \Leftrightarrow r'_t = \sigma'_t(s)$$

So $\sigma'_t(s)$ defines the extension quantity. Now, for each parent sphere t_j, we search among its children s_i the sphere s_{max}, where $\sigma'_t(s_{max})$ is maximal. Extending the radius with this value, we substract $\sigma'_t(s_{max})$ at all $\sigma'_t(i)$, so we have $\sigma'_t(i) \leq 0$ for all children spheres s_i and $\sigma'_t(max) = 0$. Hence t becomes the minimal bounding sphere of its children centered at c_t.□

The Algorithm 2 adds this extending process to the reversible sphere-tree computation.

Algorithm 2 Generic Algorithm for Exact Sphere-tree Computation

Input: $\mathcal{F}_N$ the original object,
    $\mathcal{AM}_N$ its Discrete Medial Axis and $\mathcal{C}_N$ its Discrete Power Diagram
Output: $\{\mathcal{AM}_0, \mathcal{AM}_1, \ldots, \mathcal{AM}_N\}$ the Sphere-tree
while $|\mathcal{AM}_N| > 1$ **do**
  $\mathcal{F}_{N-1} \leftarrow \mathcal{M}(\mathcal{F}_N)$ {with $\mathcal{M}$ a bounding model}
  Extraction of $\mathcal{AM}_{N-1}$ and $\mathcal{C}_{N-1}$
  for each sphere $s : (c(s), r(s)) \in \mathcal{AM}_N$ **do**
    if $\mathcal{S}(c(s)) \in \mathcal{C}_{N-1}(t)$ **then**
      t is the parent sphere for s
    end if
  end for
  for each sphere $t : (c(t), r(t)) \in \mathcal{AM}_{N-1}$ **do**
    if t has no child **then**
      $\mathcal{AM}_{N-1} \leftarrow \mathcal{AM}_{N-1} - \{t\}$
    else
      $r' \leftarrow max(\sigma'_t(s))$ {for all s child of t}
      $r(t) \leftarrow r(t) + r'$
    end if
  end for
  $\mathcal{F}_{N-1} \leftarrow \cup_t \{\ t \in \mathcal{AM}_{N-1}\ \}$
  $N \leftarrow N - 1$
end while

As radii have been extended, we have to modify $\mathcal{F}_{N-1}$ by a reverse reconstruction of the object with the new set of spheres. For n spheres in $\mathcal{AM}_N$, the linking computation and the extension of the spheres are in $O(n)$, and the reconstruction process in $O(c^d)$. As the number of spheres n is lower than the size of the object (c^d), an iteration is still performed in $O(c^d)$. So this process is also in $O(c^d)$ like the Sphere-DAG Computation (cf Theorem 1).

From the Algorithm 2, we have a sphere-tree which respects the covering condition. Nevertheless, a modification of spheres radii is performed, although the spheres were first defined to ensure the distribution of error along the object. So the extension of radii disturbs the reversibility property on each sphere-tree level. On the other hand, when the value $\sigma'_t(s_{max})$ is negative, the parent spheres radii decreases, generating an improvement of representation tightness. Moreover, as we simplify the sphere-tree by deleting the spheres without child,

this algorithm may reduce the depth of the tree. The following section illustrates these observations on real images. In order to reduce the error with the original object, the radii extension is evaluated for each node with its set of leaves (*i.e.* spheres of the original object).

4 Experiments

This section presents a comparison between the Reversible and the Extended Algorithms, with experiments on several 3D discrete objects in `.vol` or `.longvol` formats, defined in the `simplevol` library. Distance Transformation, Reduce Discrete Medial Axis Extraction and Discrete Power Diagram are computed thanks to the `MAEVA` Toolkit[1].

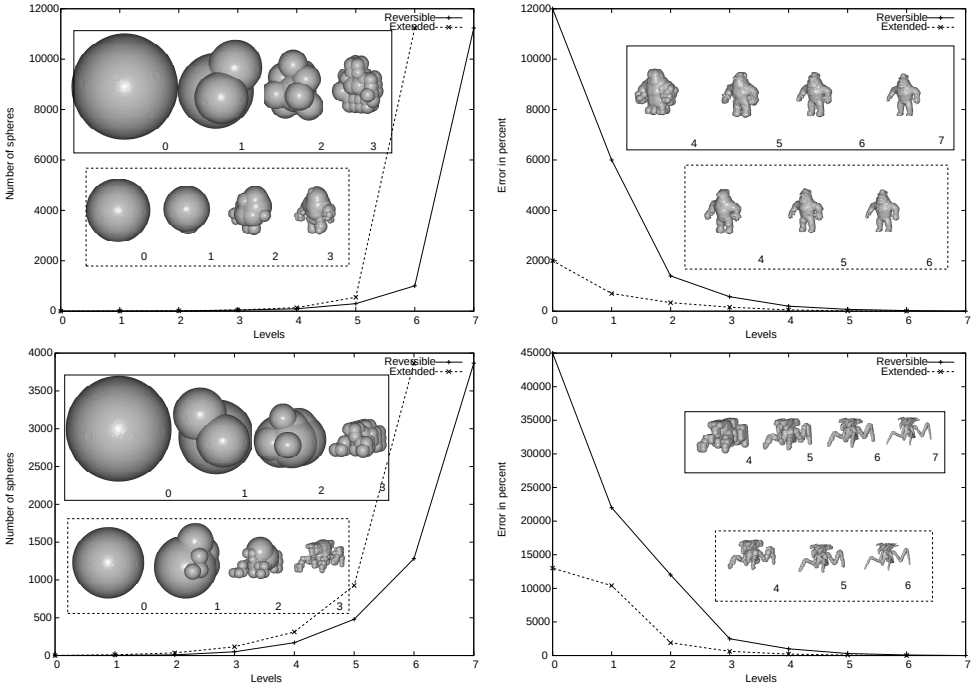

Fig. 5. Experiments for `Al.100.vol` and `ArachnidWarrior.100.vol`.
The left graph shows that the decreasing of the sphere number is faster on the Extended case, and the depht of the hierarchy is smaller. The dynamic extension process also produces an error reduction as we can see on the right graph.

[1] Simplevol and MAEVA Toolkit are available on `http://gforge.liris.cnrs.fr/`

Figure 5 shows differences between both algorithms on objects `Al.100.vol` and `ArachnidWarrior.100.vol`[2]. The left graph presents the number of spheres at each level. The right one details the percentage of error for each representation over the original object. This error is computed with the Hamming Distance which can be efficiently computed in our discrete model. As the experimental objects have different number of voxels, we prefer to represent the Hamming distance by a percentage representing the error volume added to the original object.

4.1 Reversible Algorithm

As said in Section 2.1, the representation $\mathcal{F}_{L-1}$ is f^d times smaller than $\mathcal{F}_L$ in number of voxels, as we can see in Figure 6. Thus, the associated Medial Axis contains fewer spheres. However, the distance is computed between two voxel centers in the discrete approach (here we use the square-euclidean distance), but this distance also depends on the voxel sizes at each resolution level. That's the reason why we scale the spheres to the finer resolution level by a reconstruction process. Moreover, at upper levels, the variability of radii is lower as the interval of radii becomes smaller. This range remains low during the reconstruction because all sphere radii are increased by the same value (at one level). So the sphere-tree looks like an octree, as many spheres have the same radius. Nevertheless, the reconstruction ratio at an upper level is very high, and the spheres overestimate the object geometry on this level.

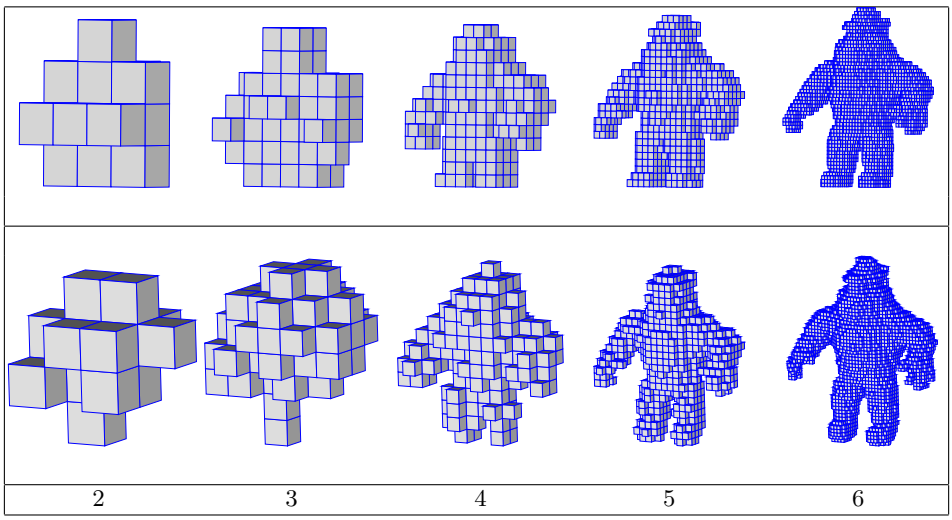

Fig. 6. Comparison between the reversible pyramid and the dynamic reconstruction of the extended version.

[2] These objects are available on `http://www.tc18.org/`

4.2 Extended Algorithm

The experiments for the extended sphere-tree construction show that the dynamic reconstruction before the extension of radius simplifies the upper levels (see Figure 6). As we delete spheres without child, the reduction of the sphere number is faster than in the reversible algorithm. Moreover, in Algorithm 2 we proposed a reconstruction of the upper level just after the extension of the radius of its spheres. The experiments show that this process controls the error increasing which is higher in the reversible case. In order to reduce the incidence of the final reconstruction process, we propose to fit one more time the spheres by $\sigma'_t(s_{max})$ after the increasing. As the reconstruction has produced spheres which overestimate the object, this final extension reduces the error.

5 Conclusion and Future Works

In this paper, we have presented an original method for a sphere-tree construction in discrete geometry. Its construction is based on the Discrete Medial Axis of the object, as best algorithms in computational geometry [14, 6, 7], but we benefit from the fact that we can efficiently extract a reversible skeleton. Hence we extract reversible sets of spheres from levels of a regular pyramid. The sphere linking at different levels is solved using properties of power diagrams. In order to ensure the covering conditions, we also propose a fast method to obtain bounding spheres of nodes, with radii extensions.

Moreover in discrete geometry, we can exactly measure the error with a Hamming distance instead of a Haussdorf estimation. Experiments show that in the reversible algorithm the error is bigger when we return at the original resolution. However, the extended algorithm solves this problem and reduces the error increasing, as we build a dynamic hierarchy.

The methods we have presented here are generic, they can be used in dimension d, and for any including model of the pyramid. We may extend these methods for generic models. We can also imagine other heuristics in order to optimize the sphere-tree. For example, we could replace the extending treatment by others minimal bounding sphere computations. However, the future works could be oriented to the topology maintenance of the object, using adaptative pyramids or morphological models, in order to reduce the error of the reversible method.

References

1. N. Amenta, S. Choi, and R. Krishna Kolluri. The power crust, unions of balls, and the medial axis transform, July 22 2000.
2. D. Attali and H. Edelsbrunner. Inclusion-exclusion formulas from independent complexes. 2007.
3. F. Aurenhammer. Power diagrams: Properties, algorithms, and applications. *SIAM Journal on Computing*, 16(1):78–96, February 1987.

4. H. Blum. A transformation for extracting new descriptors of shape. In W. Whaten-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, MA, 1967.
5. Boissonnat, Cerezo, Devillers, Duquesne, and Yvinec. An algorithm for constructing the convex hull of a set of spheres in dimension d. *CGTA: Computational Geometry: Theory and Applications*, 6, 1996.
6. G. Bradshaw and C. O'Sullivan. Sphere-tree construction using dynamic medial-axis approximation. In Stephen N. Spencer, editor, *Proceedings of the 2002 ACM SIGGRAPH Symposium on Computer Animation (SCA-02)*, pages 33–40, New York, July 21–22 2002. ACM Press.
7. G. Bradshaw and C. O'Sullivan. Adaptive medial-axis approximation for sphere-tree construction. *ACM Transactions on Graphics*, 23(1):1–26, January 2004.
8. Paolo Cignoni, Enrico Puppo, and Roberto Scopigno. Multiresolution representation and visualization of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 3(4):352–369, October 1997.
9. D. Coeurjolly and A. Montanvert. Optimal separable algorithms to compute the reverse euclidean distance transformation and discrete medial axis in arbitrary dimension. *IEEE transactions on pattern analysis and machine intelligence, VOL. 29, NO. 3, MARCH 2007*, aug 22 2007.
10. C. Dingliana. Real-time collision detection and response using sphere-trees. Technical report, March 02 1999.
11. Jacob E. Goodman and Joseph O'Rourke, editors. *Handbook of Discrete and Computational Geometry*. CRC Press, 1997.
12. S. Gottschalk, M. C. Lin, and D. Manocha. OBBTree: A hierarchical structure for rapid interference detection. *Computer Graphics*, 30(Annual Conference Series):171–180, 1996.
13. T. He and A. Kaufman. Collision detection for volumetric objects. In *IEEE Visualization '97*, October 1997.
14. P. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics*, 15(3):179–210, July 1996.
15. J. M. Jolion and A. Montanvert. The adaptive pyramid: a framework for 2D image analysis. *Computer Vision, Graphics, and Image Processing. Image Understanding*, 55(3):339–349, May 1992.
16. I. J. Palmer and R. L. Grimsdale. Collision detection for animation using sphere-trees. *Computer Graphics Forum*, 14(2):105–116, June 1995.
17. J. Pfaltz and A. Rosenfeld. Computer representation of planar regions by their skeletons. *Communications of the ACM*, 10(2):119–122, February 1967.
18. S. Prevost, L. Lucas, and E. Bittar. Multiresolution and shape optimization of implicit skeletal model. In V. Skala, editor, *WSCG 2001 Conference Proceedings*, 2001.
19. S. Quinlan. Efficient distance computation between non-convex objects. In Edna Straub and Regina Spencer Sipple, editors, *Proceedings of the International Conference on Robotics and Automation. Volume 4*, pages 3324–3330, Los Alamitos, CA, USA, May 1994. IEEE Computer Society Press.
20. G. van den Bergen. Efficient collision detection of complex deformable models using AABB trees. *Journal of Graphics Tools: JGT*, 2(4):1–14, 1997.