

# QUANTIFYING DIFFERENCES IN SKELETONIZATION ALGORITHMS: A CASE STUDY

K.J. Kruszyński and Robert van Liere  
Center for Mathematics and Computer Science  
Kruislaan 413  
Amsterdam, The Netherlands  
{K.J.Kruszynski,Robert.van.Liere}@cwi.nl

J. Kaandorp  
Section Computational Science  
University of Amsterdam  
Kruislaan 403  
Amsterdam, The Netherlands  
jaapk@science.uva.nl

## ABSTRACT

In this case study we present a framework for quantifying differences in skeleton algorithms when applied to coral-like branching structures. The output of three skeleton algorithms is applied to a set of well defined morphological metrics, and the results are compared with an a-priori known geometric structure. The results of this study help coral biologists to determine under which circumstances a skeleton algorithm performs adequately.

## KEY WORDS

Quantitative Visualization Techniques, Skeletonization, Metrics.

## 1 Introduction

Branching growth processes are ubiquitous in nature. Examples are bacterial colonies, seaweeds, higher plants and various marine sessile organisms as well as the growth of tissues and organ systems, ranging from the venous system to the pattern of alveoli within lungs. For several of these processes simulation models have been developed. A major challenge is the quantitative comparison between actual and simulated branching objects. For this, there is a need for methods that can quantify branching patterns.

Our work has been inspired by collaboration with computational and coral biologists. The growth process of a stony coral can be computed using advanced modeling and simulation techniques, [6]. Precise 3D skeleton based representations of corals are of significant interest to the biologists. Skeletons provide a compact representation while preserving the coral's topology, and retain sufficient local information to reconstruct a close approximation of the coral. This facilitates a number of important tasks including the quantification of the local width of the coral, the analysis of its topology, and its branching pattern.

Despite their popularity, the robust and stable numerical computation of skeletons remains a challenging problem. Consider Figure 1, which shows a surface rendering of a coral visualized as a semi-transparent iso-surface, and three skeletons drawn as colored lines. The data set is from a CT scan of a *Madracis mirabilis* coral. Although the skeletons were computed using published algorithms,

it is clear that each skeleton is very different. For the coral biologist this can be misleading and may lead to biased conclusions. For example, biologists use the skeleton to perform morphological measurements (such as thickness of branches, branching rates, branching angles). When the skeleton does not approximate the medial axis of the coral, these measurements become ill-defined.

The goal of this case study is to quantify differences in skeleton algorithms when applied to coral-like branching structures. The approach we have taken is to apply published morphological metrics to three skeletons of a synthetic coral-like object, [3, 4]. The algorithms that compute the skeletons are well known and have been published. For this comparison, we create a synthetic object from an a-priori known geometric structure. We apply the morphological metrics to the output of the three skeleton algorithms and compare these results with the a-priori known geometric structure.

The contribution of this case study is that it provides a framework to quantitatively compare skeletonization algorithms. It should be noted that although the framework is generic, the morphological metrics used for this case study are not. The metrics we use are derived from coral biology, and have specific meaning for analyzing the branching growth processes of coral. We must stress that when this framework is used to compare performance in application areas other than coral analysis (e.g. blood vessels, gray matter in the brain, etc.), other morphological metrics will be required.

## 2 Related work

Many books and papers describing skeleton algorithms have been published. The approaches to computing skeletons can be broadly classified into four categories; topological thinning, distance transform algorithms, wave propagation based algorithms, and Voronoi diagram based algorithms. Although each algorithm has its merits, it is beyond the scope of this paper to provide the details of each approach:

- Topological thinning methods reveal the skeleton by removing the outer layer of the object, and repeat-



Figure 1. Two views of the *Madracis mirabilis* coral. On the left, a surface rendering with three different skeleton graphs, one shown as solid lines, the other two using different stipple patterns. On the right, a close-up look of the same coral and skeletons.

ing this until only the skeleton remains. They consider the topology of surrounding voxels when deciding whether a particular voxel should be removed, [5]. Topological thinning algorithms are simple, and they preserve the topology of the object. They can, however, be rather sensitive to noise and object orientation.

- Distance transform based methods compute the distance to the image background for each object voxel and use this information to determine which voxels are part of the skeleton, [8]. Although skeleton voxels found by these methods are always exactly in the center of the object, the detection of these voxels is non-trivial, and they are usually not connected, requiring additional methods to obtain a connected skeleton.
- Wavefront propagation methods propagate a wavefront from the root of the object to the outer reaches, and use the path of this front to create a skeleton, [13]. They can be made highly insensitive to noise, and they can produce smooth continuous skeletons, but they will not always produce a skeleton which is centered inside the object. They can also require manual selection of skeleton endpoints.
- Voronoi diagrams are created by choosing a set of voxels and partitioning the image into regions closer to one of these voxels than to any of the other voxels. When these voxels are chosen on the boundary of the object, the skeleton can be constructed from the boundaries dividing the regions, [7]. While Voronoi diagram based methods can produce good skeletons, the step from diagram to skeleton is mostly based on heuristics, and can be computationally complex.

Comparisons between skeletonization algorithms are most often visual. Sometimes a single skeleton of a real-world object is assessed by an expert [12]. Often two

or more different skeletons, typically computed from very simple objects, are put side-by-side, and compared visually, [9]. As these comparisons are mostly made by researchers presenting their new algorithm, the goal of the comparison is often only to demonstrate one algorithm's superiority over another. Such comparisons are not satisfactory for a number of reasons. First, if the goal is to demonstrate the superiority of a new algorithm, there is a lack of objectivity. If algorithms are only compared with each other, it cannot be determined whether the best of the skeletons is actually close to what the skeleton of the object truly is. Performing an algorithm on a simple object might not be representative for the performance of the algorithm in general; it does not clarify whether an algorithm would be suitable for an other type of object than the one which was used. Finally, if the resulting skeletons do not differ much, it is not possible to decide which is the better one; some quantification needs to be made in order to draw a conclusion.

From the four categories of skeleton algorithms, we have chosen to compare two topological thinning algorithms and one wavefront propagation algorithm. The reason for this is that the thinning algorithms are relatively simple and easy to use. The Palágyi algorithm was chosen because it is an example of thinning based on template matching, a very popular technique for thinning algorithms. The Xie algorithm was chosen because it is an example of a thinning algorithm which does not use template matching, and because it uses a noise reduction technique as a pre-processing step. Also, given the visual results in Xie's paper, it seemed to perform better than the Palágyi algorithm for coral-like structures. The wave propagation method was chosen because wave propagation is a continuous method and not based on voxels. Deschamps' algorithm was chosen since it seemed to provide good visual results on medical data sets.

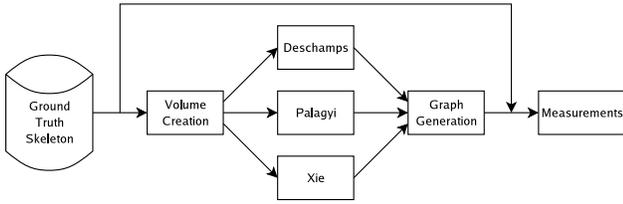


Figure 2. The skeleton comparison framework.

### 3 Method

The framework can be seen as a pipeline, consisting of several steps (see Figure 2). The first step is to use a "Ground Truth" skeleton to create a series of binary 3D images, each with a different amount of noise. Next, skeletons are extracted from each image, using three different skeletonization algorithms. Each skeleton is then converted to a graph representation, if necessary. Finally, the calculated skeletons, as well as the "Ground Truth" skeleton, are measured using several metrics.

#### 3.1 Skeletonization algorithms

Of the three algorithms used, two are based on iterative thinning, while the third uses wavefront propagation.

Thinning is a method of obtaining the skeleton of an object by removing each successive outer layer of an object's voxels, until the object is just one voxel thick. Each iteration of a thinning algorithm consists of one or more sub-iterations. Each sub-iteration considers one or more voxels in the image, and decides whether they can be set to the background value. A parallel sub-iteration considers each voxel separately, and then removes all suitable voxels at once; a sequential sub-iteration removes voxels one at a time, thus the removal of each voxel is influenced by voxels removed previously in the same sub-iteration.

The *neighborhood* of a voxel is an important notion in thinning algorithms. If the voxels of an image are considered to be adjacent cubes, centered at the voxel locations, then the 6-neighborhood of a voxel consists of all surrounding cubes with which the voxel shares a face; similarly, the 18-neighborhood consists of all face and edge neighbors, and the 26-neighborhood consists of all face, edge, and vertex neighbors. Two voxels are  $n$ -adjacent if they are in each others'  $n$ -neighborhood.

In wavefront propagation, a wavefront is propagated inside the object. A skeleton is then obtained by tracking the path of the wavefront back to the starting point.

##### 3.1.1 Palágyi

Palágyi and Kuba [10] describe a border sequential thinning algorithm. It repeatedly compares border voxels and their neighborhood with a set of templates, and removes

them if there is a match. When no more matches can be made, the algorithm stops.

Each iteration of this algorithm consists of six parallel sub-iterations. Each sub-iteration only considers object voxels, which have a neighboring background voxel in one particular direction. The templates are appropriately rotated or mirrored for each direction. Each voxel matching one of the templates is removed. The order of the directions is chosen so that the object is thinned evenly on all sides.

##### 3.1.2 Xie

In the algorithm of Xie, Thompson and Perucchio [14], each iteration consists of six directional parallel sub-iterations, followed by a repeating sequential sub-iteration. As in Palágyi's algorithm, the directions are chosen to ensure the object is thinned evenly.

The parallel sub-iterations remove candidate border voxels by classifying neighboring voxels, and counting the number of connected components formed by the voxels of each class. If the numbers of components satisfy certain requirements, the voxel can be removed.

The sequential sub-iteration attempts to remove voxels which were considered but not removed in the parallel sub-iterations. It uses a slightly different set of conditions, and checks the voxels in a specific order, to avoid removing too many voxels.

This algorithm also includes a pre-processing stage for reducing noise in the input image. Single-voxel protrusions are removed, and single-voxel indentations are filled.

##### 3.1.3 Deschamps

The algorithm of Deschamps [2] uses a modified Fast Marching Method [13] to propagate a wavefront from a starting point throughout the object. The Fast Marching Method calculates the front arrival time for all voxels. This wavefront is then traced back to the origin using the Gradient Descent method, which results in a set of lines representing the skeleton.

The speed of the wavefront at each voxel is determined by a speed function. We used the square of the distance transform [1] of the data set for this purpose; this ensures that the skeleton is centered within the object.

One of Deschamps' modifications to the Fast Marching Method is the so-called 'freezing' of the front in locations that are relatively close to the starting point of the propagation. This results in improved detection of elongated shapes by stopping propagation in directions where the front travels slowly.

The points from which the front should be traced back to the origin are determined with the aid of another modification to the Fast Marching Method. In addition to the arrival time, the length of a path, which the front followed to each voxel, is calculated. The object is then divided into connected regions where the length of this path is within

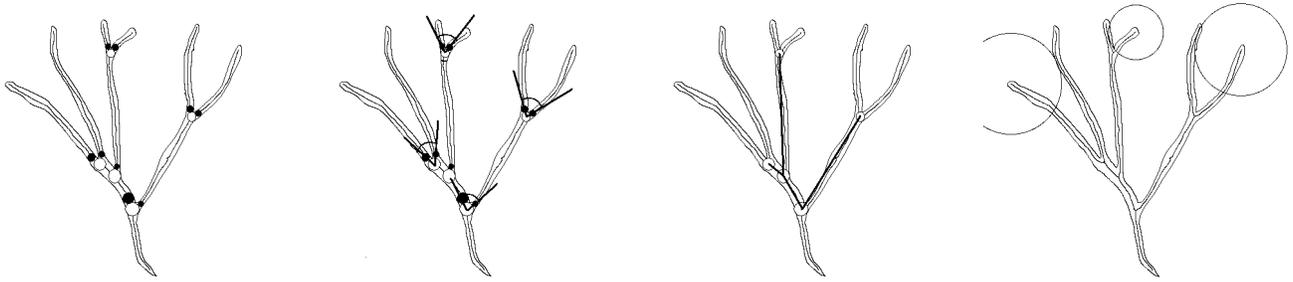


Figure 3. The four measurements performed on coral branch structures. From left to right: minimum and maximum branch thickness (a-spheres and b-spheres), angles between branches, branching rate and branch spacing.

intervals of a given size. For each region it is determined which other region preceded it, and which region follows it; the regions which do not have any region following it are then considered end regions. From each end region, the point with the highest path length is selected as the starting point for a trace back to the origin.

The backtracking does not follow the voxels; instead the object voxels are considered to be grid points where the wavefront arrival time and its gradient are known. At each step, the gradient vector of the arrival time at the current location is interpolated, and the current location is moved a pre-determined distance in the opposite direction. This is repeated until the starting point is reached.

Because the tracing would produce a set of unconnected lines, each trace is halted and connected to another trace if the distance to this other trace is smaller than a specified threshold value.

### 3.2 Skeleton graph

Voxel skeletons must be converted to skeleton graphs, before any measurements can be performed. The voxels of the data set are assumed to form a three-dimensional grid, with the far bottom left corner at location  $(0, 0, 0)$ . The distance between two 6-adjacent voxels in the grid is defined as 1. A graph can be constructed from the voxels by connecting the grid locations of adjacent skeleton voxels with lines. These form the edges of the graph, while the grid locations are the vertices. The distance between vertices, and thus between voxels, is the Euclidean distance between the corresponding grid locations.

To prevent certain irregularities and loops in the graph, each voxel is connected to the graph only once, and voxels are added with a preference for skeleton voxels closer to ones which are already part of the graph [11]. The initial voxel is chosen manually.

### 3.3 Measurements on stony corals

Several measurements were performed on the skeletons. These measurements are commonly used by coral biolo-

gists.

- The maximum thickness of a branch is defined as the thickness at a junction of the skeleton. It is the radius of a sphere, centered at a the junction, touching the background in the original voxel image. This radius is thus equal to the Euclidean distance transform at the junction point. These spheres will be referred to as a-spheres.

The minimum thickness of a branch is the radius of a sphere on the part of the skeleton following the junction, with a radius such that it touches both the image background and the sphere at the junction. These spheres will be referred to as b-spheres. The first image in Figure 3 shows the equivalent minimum and maximum discs in a 2D image.

- The angle between two branches is defined as the angle between two lines, starting at the center of a maximum sphere, and each ending at the center of a different minimum sphere. The second image in Figure 3 shows these angles in a 2D image.
- The branching rate is defined as the distance between two successive branching points in the skeleton. It is a measure of how often new branches are formed; a high value indicates relatively slow formation of new branches during the growth process. The third image in Figure 3 shows the branching rate.
- The branch spacing is defined as the distance between the endpoint of a branch, and the closest point on the skeleton which does not belong to the current branch. The endpoint of a branch is not the last skeleton point belonging to the branch. Instead it is a point found by searching for a maximum distance transform sphere along the skeleton, starting at the end. A sphere is considered a maximum sphere if it is not completely contained inside another sphere on the skeleton. The last image in Figure 3 shows the branch spacing.
- In addition to the measurements used by biologists, the ratio of the length of the skeleton between every

two successive branching points, and the Euclidean distance between these two points, was calculated. For a segment of the skeleton consisting of the points  $p_0, \dots, p_n$ , with branching points  $p_0$  and  $p_n$ , the formula is:

$$\frac{\sum_{i=0}^{n-1} \|p_{i+1} - p_i\|}{\|p_n - p_0\|}$$

The original skeleton consists of straight lines, with ratio 1.0, therefore this ratio indicates how straight the calculated skeleton is.

### 3.4 Test data

The data used for testing is a set of 3D binary images of a fractal-like tree, to which increasing amounts of noise have been added. A synthetic object was chosen in order to have a reference skeleton, or "Ground Truth", to which the computed skeletons could be compared.

#### 3.4.1 Data creation

To create the data sets, first a polygonal line model of a tree was generated. The model starts with a vertical line. From the top of this line, two new lines continue, one at  $55^\circ$  to the left, the other at  $55^\circ$  to the right from the initial line, in the XY plane. The length of the new lines is 0.8 times the length of the initial line. This process of adding new, shorter lines is repeated seven more times, with each two new lines constructed in a different plane, rotated  $65^\circ$  to the right around the, relative to the previous plane.

Next, around these lines a set of round polygonal tubes was constructed, with a linearly decreasing diameter, such that the diameter at the base of the tree was 10 times the diameter at the end; the diameter at the base was set to 0.2 times the length of the first segment. Spheres were added at the root and the endpoints of the tree, with the same diameter as the tubes, to create round endpoints.

The polygonal model was then voxelized as a binary volume, with the inside filled with value 1.

#### 3.4.2 Noise

Additional data sets were produced by adding increasing amounts of noise to the initial data, around the boundary between object and background. To add the noise, first a volume of the same size as the data set was filled with random floating point values between 0.0 and 100.0. To generate a particular noise level  $l$ , first all points from this volume with a value less than  $l$  were selected. All corresponding points in the binary image, which were found to be 6-adjacent to at least one voxel with value 1, but themselves had value 0, were then set to 1, producing an intermediate image.

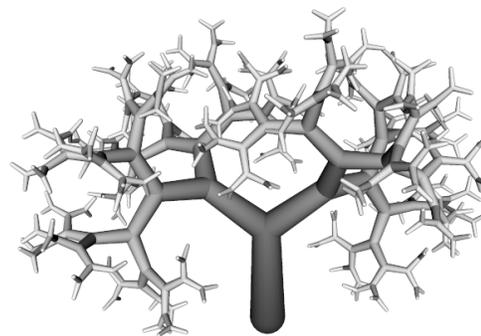


Figure 4. The polygonal model of the test object. Shade of gray indicates the diameter of the object.

Next, from the random data all points with a value greater than  $100.0 - l$  were selected. Each corresponding 1-valued voxel from the intermediate image, which was 6-adjacent to at least one 0-valued voxel, was then set to 0 in a final image. For our analysis we have chosen noise levels 0, 1, 2, 4 and 8.

This method produces noise which is only present on the surface of an object, as could be the case with a segmented image of an object, obtained from a CT-scanner or similar device.

## 4 Results

The synthetic object was voxelized to create a 3D binary image with a resolution of  $360 \times 300 \times 240$  voxels. The algorithms of Palágyi, Xie, and Deschamps were then used to obtain skeletons of the images.

While the algorithms of Palágyi and Xie are turn-key solutions, the wavefront propagation algorithm requires many parameters. It was used with the following settings: points were frozen when the distance to the starting point was less than 0.5 times the maximum distance; the distance interval for endpoint detection was 6.0; the minimum region size was 5 voxels; the Gradient Descent step size was 0.5; and the minimum distance between traces was 3.5.

### 4.1 Quantitative results

The graphs in Figure 5 show the average values of each measurement, for each algorithm at each noise setting. In addition, the results of the same measurements for the original skeleton, which was used to construct the test object, are shown in each graph. These measurements were made by subdividing the lines of the original skeleton until each of these smaller lines had a length of less than 0.5, and using these lines as a skeleton graph for measurements.

The graph of the average radius of the a-spheres shows that the Xie results are the best for this particular measurement, but the difference with wave propagation is

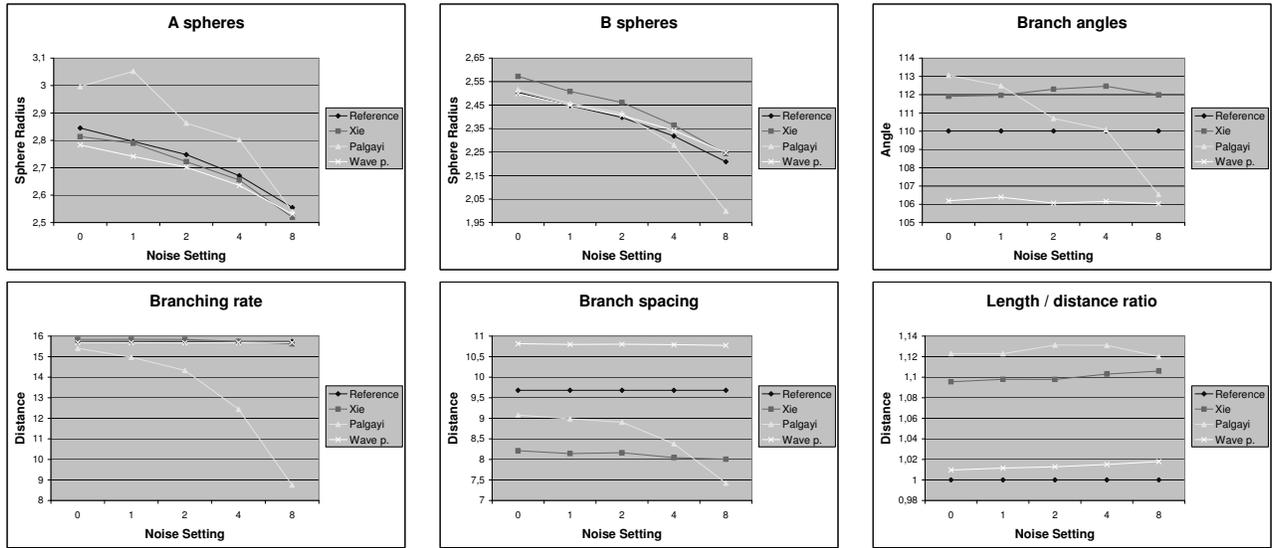


Figure 5. Graphs of each measurement for the reference skeleton, the Palágyi skeleton, the Xie skeleton and the Deschamps skeleton. The x-axis denote noise settings 0, 1, 2, 4, 8. The y-axis denote average distances or angles.

minimal. Palágyi shows erratic results, and only performs well for the highest noise level.

For the average radius of the b-spheres, Xie performs slightly worse than the other algorithms at lower noise levels, but at highest noise level Palágyi produces a quite different average, while wave propagation performs the same as Xie.

The angles between branches are fairly constant across noise levels for Xie and wave propagation. The angles with Xie are a little too high, while they are a bit more too low with wave propagation. Palágyi gives higher angles than Xie at lower noise levels, but gradually decreases to almost the same angle as wave propagation at the highest noise level.

For the branching rates, both Xie and wave propagation give results very close to the reference skeleton. The results with Palágyi are a little worse without noise, but deteriorate rapidly when noise is added.

The branch spacing is consistently too high with wave propagation, and much too low with Xie. Palágyi with little noise produces the best results, but is still too low. With more noise, the average value becomes much too low.

The length/distance ratio between branching points is reasonable consistent between noise levels. The thinning algorithms have a much higher ratio than the wave propagation algorithm. This is mostly due to the voxel-based nature of these algorithms, which increases the length of a diagonal path between more distant voxels due to aliasing, as opposed to the smooth lines produced by the wave propagation algorithm.

Table 1 shows the number of a-spheres found by each algorithm for each noise level. Recall that an a-sphere indicates a junction point in the skeleton. The table shows that the Palágyi skeleton is very sensitive to noise, while the Xie

algorithm is less sensitive. The number of a-spheres found by Deschamps algorithm remains the same as that of the reference skeleton.

	0	1	2	4	8
Reference	255	255	255	255	255
Xie	255	255	255	257	259
Palágyi	267	275	287	341	517
Wave propagation	255	255	255	255	255

Table 1. Number of a-spheres for each noise level.

## 4.2 Visual results

Figure 6 gives a visual comparisons of the test results. The top row shows a surface rendering of the test object for noise levels 0, 2, and 8. The bottom row shows the corresponding skeletons in yellow (Palágyi), cyan (Xie), and magenta (Deschamps). As stated in the introduction, with only visual inspection it is very difficult to determine which algorithm performs best.

Figure 7 shows the Palágyi skeleton with a-spheres, at three noise levels. These images show where the Palágyi algorithm introduces additional junctions. From the image it can be observed that junctions are generated at the lower branching levels of the tree (where the branch thickness is large), and not at the higher branching levels (where the branch thickness is small).

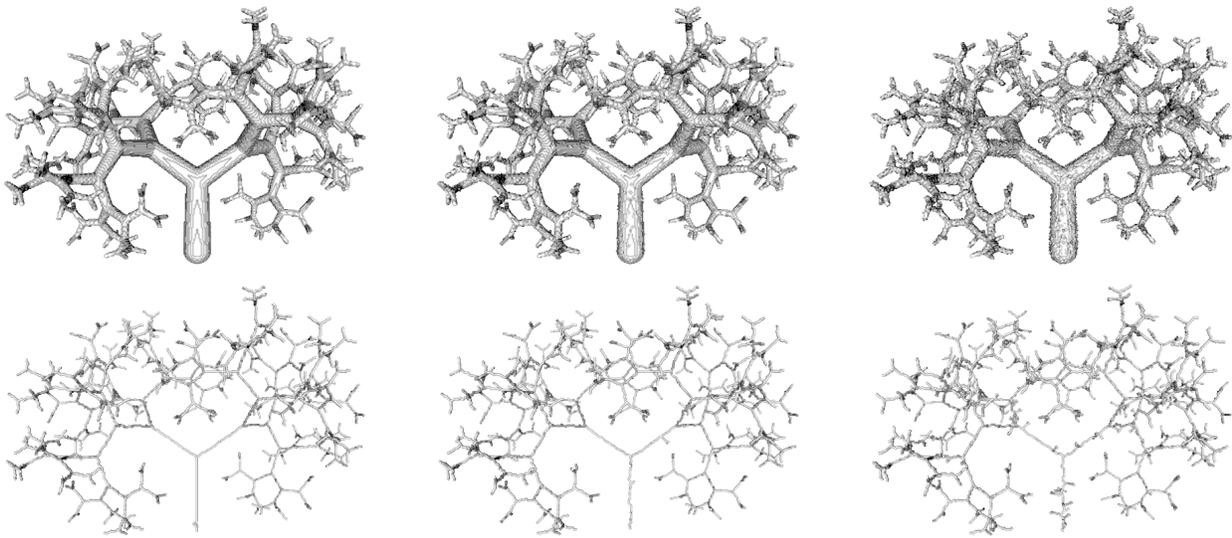


Figure 6. Surface renderings of the test object for three noise levels (top), and the corresponding Palágyi skeletons (bottom).

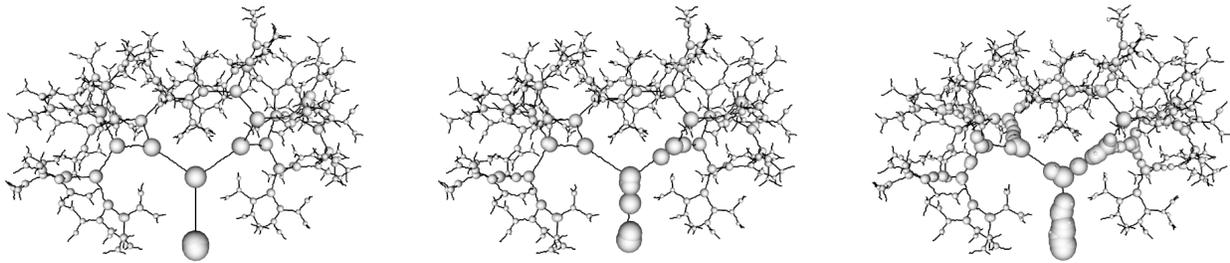


Figure 7. The Palágyi skeleton with a-spheres, at noise level 0 (left), 2 (middle), and 8 (right).

## 5 Discussion

In the previous sections we have compared three skeleton algorithms by applying morphological metrics to the output of the algorithms with the results of a-priori known geometric structure. Two observations can be made when studying the results:

- No algorithm preforms clearly best for all metrics. Even for one type of metric, results can vary depending on the amount of noise present in the data set.

From this we can conclude that it cannot be determined in advance which algorithm will give the best performance. The choice will depend on which metric is most important for the biologist.

- Wave propagation is less sensitive to noise than the thinning algorithms, although the pre-processing stage of the Xie algorithm appears to eliminate most noise-related artifacts. On the other hand, wave propagation requires that all parameters be set to appropriate values. The user must fine-tune the algorithm for each new object; the thinning algorithms, having no parameters, require no user interaction.

From this we can say that the wave propagation method is the safest to use if the data set of the coral has much noise.

It must be noted that various assumptions were made for this study:

- The skeletons are assumed to contain no loops. While the test object has been created without loops, many real branching objects can and do contain loops. While the thinning algorithms can properly handle loops, the wave propagation algorithm ignores loops altogether.
- The resolution of the synthetic test object is fairly arbitrary chosen. The skeleton algorithms which preform well for certain resolutions, do not necessarily produce the same results for data sets with significantly higher or lower resolutions.
- The values used in the results were computed by taking the average of all measurements over the complete skeleton. An alternative, but also valid approach, would be to compute the average of all measurements at each branch level. This may lead to other results, as

the algorithms may perform differently on thick and thin branches.

- In the study, we have applied noise to the surface of the synthetic test object. An alternative to adding noise would be to perturb the geometry of the test object, and to generate test objects from the perturbed geometry. Although this approach leads to different skeletons for each noise level, it may be a better approach to simulate the irregularities of stony coral branching processes.

The morphological metrics we use are derived from coral biology and have specific meaning for analyzing coral growth processes. However, it could be argued that many of the metrics are quite generic for studying branching objects. For example, branching angles, branching rates and branch spacings are quite generic. These metrics may be of interest for users that study branching objects in other application domains.

## 6 Conclusion

The contribution of this case study is that it provides a framework to quantitatively compare skeleton algorithms. Morphological metrics, that have specific meaning for analyzing coral branching growth processes, have been used as basis for these comparisons. The algorithms of Palágyi, Xie and Deschamps have been used in this study. An a-priori known geometric structure has been used a ground truth, for which we relate the output of the three algorithms.

We conclude that it cannot be determined in advance which algorithm will give the best performance. The choice will depend on which metric is most important for the biologist.

In the future we will use the skeletons for other measurements; e.g. Horton statistics, Tokunaga statistics, fractal dimensions.

## Acknowledgments

This work was carried out in the context of the Virtual Laboratory for e-Science project ([www.vl-e.nl](http://www.vl-e.nl)). This project is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OC&W) and is part of the ICT innovation program of the Ministry of Economic Affairs (EZ).

## References

- [1] G. Borgefors, Distance transformations in digital images, *Computer Vision, Graphics, and Image Processing*, 34, 1986, 344–371.
- [2] T. Deschamps, *Curve and shape extraction with minimal path and level-sets techniques - applications to 3D medical imaging*, Ph.D. thesis, Université Paris-IX Dauphine, Place du maréchal de Lattre de Tassigny, 75775 Paris Cedex, December 2001.
- [3] J. Kaandorp, Morphological analysis of growth forms of branching marine sessile organisms along environmental gradients, *Mar. Biol.*, 134, 1999, 295–306.
- [4] J. Kaandorp and R. G. Leiva, Morphological analysis of two- and three-dimensional images of branching sponges and corals, *Morphometrics and their applications in Paleontology and Biology* (Berlin) (A. Elewa, ed.), Springer-Verlag, 2004, pp. 83–94.
- [5] L. Lam, S. Lee, and C. Suen, Thinning methodologies—a comprehensive survey., *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9), 1992, 869–885.
- [6] R. Merks, A. Hoekstra, J. Kaandorp, and P. Sloot, Branching and morphologic plasticity in corals: the polyp oriented approach, *J. Theor. Biol.*, 228, 2004, 559–576.
- [7] M. Näf, G. Székely, R. Kikinis, M. E. Shenton, and O. Kübler, 3D Voronoi skeletons and their usage for the characterization and recognition of 3D organ shape, *Comput. Vis. Image Underst.*, 66(2), 1997, 147–161.
- [8] C. Niblack, P. Gibbons, and D. Capson, Generating skeletons and centerlines from the distance transform, *Graphical Models and Image Processing*, 54, 1992, 420–437.
- [9] K. Palágyi and A. Kuba, A parallel 3D 12-subiteration thinning algorithm., *Graphical Models and Image Processing*, 61, 1999, 199–221.
- [10] K. Palágyi and A. Kuba, A 3D 6-subiteration thinning algorithm for extracting medial lines, *Pattern Recognition Letters*, 19(7), 1998, 613–627.
- [11] F. Reinders, M. E. D. Jacobson, and F. H. Post, Skeleton graph generation for feature shape description, *Data Visualization 2000* (W. d. Leeuw and R. v. Liere, eds.), Springer Verlag, 2000, pp. 73–82.
- [12] J. A. Schaap, P. J. H. de Koning, J. P. Janssen, J. J. M. Westenberg, R. J. van der Geest, and J. H. C. Reiber, 3D quantification visualization of vascular structures in magnetic resonance angiographic images., *International Conference on Computational Science* (3), 2002, pp. 242–254.
- [13] J. A. Sethian, *Level set methods and fast marching methods*, Cambridge University Press, 1999.
- [14] W. Xie, R. P. Thompson, and R. Perucchio, A topology-preserving parallel 3D thinning algorithm for extracting the curve skeleton, *Pattern Recognition*, 36(7), 2003, 1529–1544.