

3D-SKELETONIZATION OF URETERIC TREES IN DEVELOPING KIDNEYS

G. FRICOUT, D. JEULIN

Center of Mathematical Morphology, Ecole des Mines, Fontainebleau, France

L.A. CULLEN-MCEWEN, I.S. HARPER, J.F. BERTRAM

Department of Anatomy and Cell Biology, Monash University, Melbourne, Australia

Abstract In this paper, a method is described for producing a skeleton of a 3D branching object, the ureteric tree of the developing mouse kidney. The algorithm has to deal with scale differences between the 3 directions of space, and therefore, a Scaled Distance Function and a Weighted Geodesic Distance are defined. The extremities of the branches are defined as maxima of the Weighted Geodesic Distance and they are linked together using centred paths in order to obtain a skeleton. The results of the algorithm are presented as well as their value in studying the developing kidney.

Keywords: 3D skeletonization, distance function, geodesic distance, kidney, branching.

1. Introduction

This study is the result of a partnership between the Center of Mathematical Morphology of the Paris School of Mines and the Department of Anatomy and Cell Biology of Monash University, Australia. The main topic was the study of the morphogenesis of the ureteric tree of the growing kidney. The ureteric tree develops branches in the 3 directions of space. Understanding of this phenomenon requires quantification of the growth of the kidney, and in particular, accurate measurements of the length of the branches in 3D. To be able to obtain this data, it was decided to produce a thin 3D branching object centred in the kidney tree observed under a confocal microscope which enables us to describe the volume of the kidney by a set of planar frames. For simplification, the resulting 3D object will be called a skeleton in the following.

2. Existing algorithms

The literature provides many skeletonization algorithms [1], even in 3D, but the specificity of the object under study in our topic make them difficult to adapt.

Firstly, some of these algorithms are based on iterative pruning and usually produce some surfaces when working in 3D. This may be very interesting for some purposes, but for the study of the kidney, biologists are mainly interested in the length of the branches: the skeleton should therefore be made of thin lines instead of surfaces.

Some other algorithms enable us to calculate centred lines between two points inside a 3D object: this is much closer to our requirement, but there are still some difficulties specific to our 3D object. In the present case, the problems to solve are:

- The difference of scale in the 3D object obtained by confocal microscopy between the X and Y directions on the one hand, and the Z direction on the other hand. This has to be taken into account during the skeletonisation process.
- We need to build a thin 3D tree which has to be centred in the initial object. This involves a specific order relationship on the extremities of branches and nodes.

To our knowledge, no existing algorithm fulfill all these conditions

3. Distance Function and Scaled Distance Function

The algorithm presented here will be largely based on what is called the distance function of an object and also on the geodesic distance [11] [9] inside an object.

3.1 DISTANCE FUNCTION

The classical Distance Function [3] of a binary object can be represented by an image in which every pixel P of the initial binary object has a gray level corresponding to the length of the shortest path between P and the outside of the object. The length of the path is usually simply represented by its number of pixels.

3.2 SCALED DISTANCE FUNCTION

As mentioned in section 2, one of the main problems of our data was the difference of scale between the different directions of space. The aim of the Scaled Distance Function described in the following is to take into account this difference of scale in a quantitative way.

The only difference between the Scaled Distance Function and the simple Distance Function described in the previous section will be the way to calculate the length of a path. A scale parameter will be associated to each direction of space (usually, s_z can be different from s_x and s_y which are the same). If we assume for simplicity that a 6-connectivity is used, starting from one extremity of a path, its length will be calculated step by step by adding the scale factor corresponding to the direction of the next pixel until the other extremity of the path is reached.

One can notice that this is still a very basic approximation of a “real Euclidean” distance between two points as the distance between $M_1(x_1, y_1, z_1)$ and $M_2(x_2, y_2, z_2)$ is:

$$s_x \cdot |x_1 - x_2| + s_y \cdot |y_1 - y_2| + s_z \cdot |z_1 - z_2|$$

Instead of:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

But this is enough for our purpose and is much easier to implement as compared to more involved algorithms such as Danielsson’s.

4. Weighted Geodesic Distance

In this section, the Weighted Geodesic Distance will be introduced. This is mainly derived from the classical Geodesic Distance inside a binary object. The principle of the Geodesic Distance is to calculate the shortest path between two points if all the possible paths are constrained inside a binary mask. The Weighted Geodesic Distance would be exactly the same, except that the lightest path is researched instead of simply the shortest. A weight is attributed to each pixel of an image, and the weight of a path is simply the sum of the weights of all the pixels in this path. The definition of the Weighted Geodesic Distance can easily be enlarged to the Geodesic Distance between a point and a set. This notion was used in earlier studies to characterize the 3D tortuosity of porous media [5].

5. Implementation

The implementation of the Scaled Distance Function and the Weighted Geodesic Distance are very close and both are based on the hierarchical queue [2] structure. The hierarchical queue is a list of FIFO (“first in first out”) organized with a priority order corresponding to the distance to the origin. This structure enables us to implement in a simple and efficient way the order of processing pixels which depends on the weights.

For the Weighted Geodesic Distance, one starts from the root (defined below), which is queued with a priority 0. It is then possible to determine the priority of its neighbors depending on their weights. These points are queued in the queue corresponding to their distance. The neighbors of points of priority 1 can then be processed and queued at the right priority and so on.

If the Scaled Distance Function is considered, the points to be placed at priority 0 for initializing the hierarchical queue are all the points of the border of the object, and the priority is not calculated according to a weighted image, but according to different scale parameters.

6. Detection of the Extremities of the Skeleton

6.1 EXTREMITIES AND GEODESIC DISTANCE MAXIMA

The way to produce our skeleton is based on centred path calculations between points, and we first need to calculate the extremities of the branches. The idea

is to find the extremities as maxima of a particular Geodesic Distance: starting from one point R ¹ (the root of the tree), the Weighted Geodesic Distance to R is calculated for each point of the tree. The final branches appear as non-way paths in the tree, and therefore, they will correspond to some maxima of the Geodesic Distance to R . The difference of scale will be taken into account by using a Weighted Geodesic Distance with adapted weights. There are two main ways to produce this weights image using the Scaled Distance Function of the binary image: the first one would be to produce the scaled distance function with the right parameters of scale, and the second one to produce this same distance function, but invert it to get the weights image so that the heavy weights are on the border instead of being in the centre.

High centre weights. The weights calculated by the Scaled Distance Function as described in section 3.2 will make the border easy to go through whereas the centred points would be very hard to cross when considering the propagation of the distance in the Weighted Geodesic Distance algorithm (section 4), and that is why the last points to be reached would be the points in the middle of the tree which will become maxima of the distance function: this is an interesting way to get the extremity of the branch centered in the structure instead of being on the border.

Note that some detected points appear as way path in figure (1) on the left-hand side: this can result either from the 2D projection of the 3D extremities, or from branching points generated by the algorithm. They do not generate any artifact since they are recovered later during the backpropagation process.

High border weights. The second way to process the weights will be based on the inverse Scaled Distance Function (section 3.2) and therefore, the “distance flow” will go first through the middle of the branches and will start reaching the border subsequently: the maxima would all be part of the border of the tree. This method is usually more noisy (figure 1 on the right-hand side) than the previous one for detecting the extremities and the high center weights will be used in this purpose. However, the high border weights will enable us to produce minimal path centred in the middle of the structure and this way of calculating the weights will be kept for this purpose (see section 7).

7. Maxima Linking

For producing the skeleton starting from the extremities of the branches, it is necessary to link these maxima using a path centred in the binary object. This will be done by calculating the Weighted Geodesic Distance with light weights in the centre of the binary object. Then, the lightest path between two points will be centered in the object and will not “cut the edge” as the normal shortest geodesic path between two points would do. The problem is then to be able to

¹Notice that the extremities do not depend on the choice of point R since we are looking for local maxima

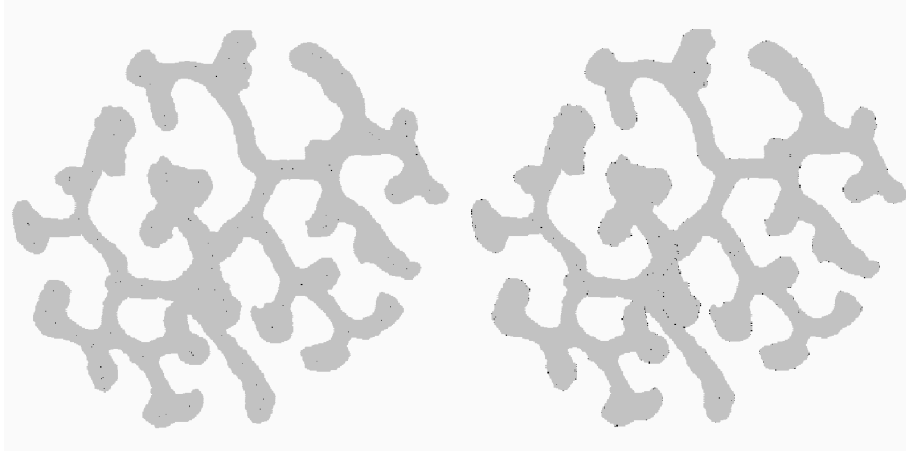


Figure 1. Difference of results in the maxima detection when using the high centre weights (on the left) or the high border weights (on the right)

find the shortest path between one point and the origin using the map of the Weighted Geodesic Distance [12] [6] [10].

7.1 BACKPROPAGATION

In this part, two points R (the root of the tree) and M (one of the extrema previously selected) would be considered. We will use in this paragraph the notation $w(x)$ and $d(x)$ to make reference to the weights function and $dg_O(W, R, \cdot)$, respectively.

When speaking of the Geodesic Distance d , one has to remember the initial definition of $d(x)$ which is the length (or the weight) of the shortest path between x and R , and this information is known for all the points as well as the weight function w .

The research of the minimal path will be done recursively. Starting from M , one selects its neighbor whose distance to R is the smallest (this is done by the use of the image representing $d(x)$). Then, one iterates the process with the neighbors of the point previously selected. Finally, the path made of all the pixels selected until the root is reached is a minimal path between R and M . This method is due to Dijkstra [7] and can be easily adapted to find one of the shortest paths between a set A and a point M (that will be needed in the following).

The linking process will be done as follows: the furthest extremity from the root is linked to the root, then the second furthest is linked to the previously obtained branch and all the maxima are iteratively added to finally obtain the skeleton.

8. Implementation

The implementation of the calculation of minimal paths is easy, considering the method described in the previous section, and it is now possible to implement the full skeletonisation algorithm:

The variables defined by the user are:

binTree: The input binary image representing the tree.

output: The output image which will contain the skeleton.

s_x, s_y, s_z : The scale parameters.

λ : A size parameter for the reconstruction filtering.

```

void  skeletonisation (Image binTree, Image output,  $s_x, s_y, s_z, \lambda$ )
{
    Image W=ScaledDistFunction(binTree,  $s_x, s_y, s_z$ );           (1)
    P=FirstWhitePixel(binTree);                                 (2)
    dist=WeightedGeodesicDistance(binTree, W, P);
    dist2=ReconstructionFiltering(dist, $\lambda$ );
    Max=SelectMaxima(dist,dist2);                               (3)
    [m1,m2...,mn]=sortMaxima(Max);                             (4)
    W2=Inverse(ScaleDistanceFunction(binTree, $s_x, s_y, s_z$ ));   (5)
    R=m1;                                                       (6)
    dist3=WeightedGeodesicDistance(m1, W2, R);
    Skel=R                                                       (7)
    for  (i=n ; i>=n ;i - -) {
        Si=Backpropagation(dist3,mi, W2);                       (8)
        Skel=Skel+Si;                                           (9)
        dist3=WeightedGeodesicDistance(Skel, W2);              (10)
    } endfor
    output=Skel;
} endfunction

```

(1) W is the weight image.

(2) P is the root and its choice doesn't affect the maxima calculation.

(3) The maxima are calculated on a filtered image for limiting noise.

(4) The maxima are sorted by increasing order of distance to the root.

(5) $W2$ is a weight image with light centred weights (section 7).

(6) The first maxima of the sorted list is taken has the root of the final tree.

(7) Initialization of the skeleton with only the root.

- (8) S_i is the branch separating the maximum m_i and the skeleton at the previous step.
- (9) The skeleton $skel$ is updated by adding the branch S_i .
- (10) The Weighted Geodesic Distance is recalculated for the calculation of the next branch with the new skeleton as starting set.

9. Results

The results based on processing more than 60 kidneys are good in 3D (figure 2). The obtained trees are fully satisfactory as compared to the standard manual process implemented by biologists. Very few artifacts like false branches are generated by the roughness of surface after the segmentation process. Some additional functions allow the user to add or remove some branches manually, if there are any problems in the final skeleton. The program can also detect the branching points and generate automatically the corresponding tree structure, as well as compute the length of all the branches.

10. Conclusion

The result of this study is a software for quantitatively analysing the 3D shape of the developing renal ureteric tree in an efficient way, despite the problem of scale coming from the confocal scanning. The data obtained with this method have been used for statistical studies in order to simulate the growth of the kidney by random models with appropriate characteristics [8].

The development of several other organs also involves branching, and some work has been done to adapt the algorithm to study branching in the developing prostate gland. Some biological studies are also in process concerning this organ using this skeletonization algorithm.

References

- [1] Arcelli C., Sanniti di Baja G. (1993) *Euclidean skeleton via centre-of-maximal-disc extraction*, Image and Vision Computing, 11(3):p. 163-173.
- [2] Beucher S., Meyer F. (1993) *The Morphological Approach to Segmentation: The Watershed Transformation*, Mathematical Morphology in Image Processing, E.R. Dougherty, ed.: Marcel Dekker, Inc. p. 433-481.
- [3] Borgefors G. (1984) *Distance transforms in arbitrary dimensions*, Computer Vision, Graphics, and Image Processing, 27: p. 321-345.
- [4] Bouchet S. (1999) *Segmentation et quantification d'images tridimensionnelles*, Internship report, Paris School of Mines Publication.
- [5] Decker L., Jeulin D., Tavena I. (1998) *3D morphological analysis of the connectivity of a porous medium*, Acta Stereologica, Vol. 17, n°1, p. 107-112.
- [6] Deschamps T., Cohen L.D. (june 2000) *Fast extraction of minimal paths in 3D images and application to virtual endoscopy*, Rapport 0026, Les cahiers du Ceremade, Université Paris Dauphine, URL: www.ceremade.dauphine.fr/cadrepub.html
- [7] Dijkstra E. (1959) *A note on two problems in connexion with graph*, Numerische Mathematik, Vol. 1, p. 269-271.

- [8] Fricout G., Jeulin D. (October 2001) *A model of random tree to simulate the growth of kidneys*, N-51/01/MM, Paris School of Mines Publication.
- [9] Lantuejoul C., Maisonneuve F. (1984) *Geodesic methods in quantitative image analysis*, Pattern recognition. 17(2): p. 177-187.
- [10] Soille P. (1994) *Generalized geodesic distances applied to interpolation and shape description*, Mathematical Morphology and Its Application to Image Processing, Serra J., Soille P., Kluwer Academic Publishers, p. 193-200.
- [11] Serra J. (1982) *Image Analysis & Mathematical Morphology*, Acad. Press.
- [12] Zhou Y., Kaufman A., Toga A.W. (1998) *Three-dimensional skeleton and centerline generation based on an approximate minimum distance field*. The Visual Computer 14: p. 303-14.

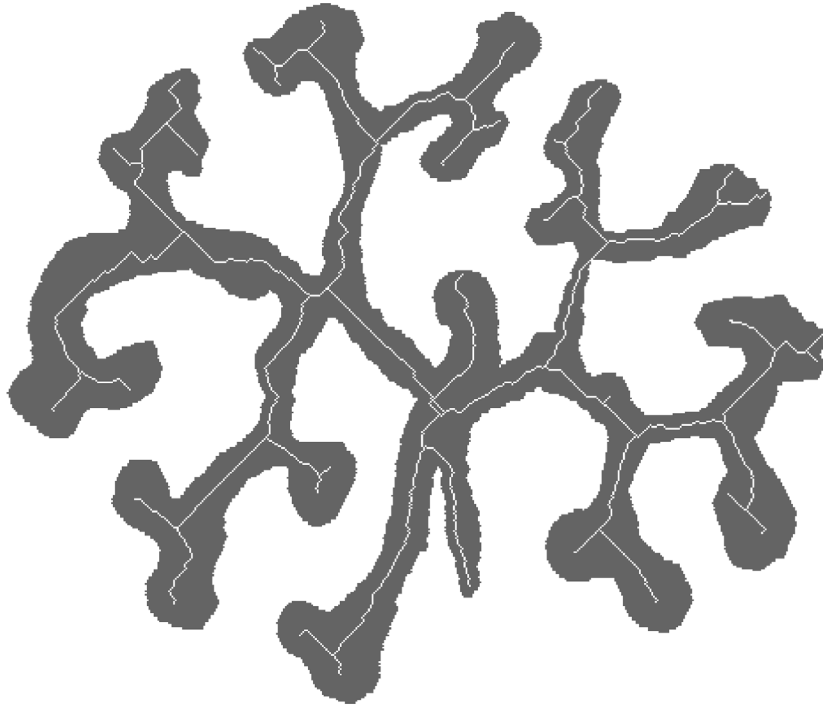


Figure 2. Super-position of the skeleton with the original 3D binary object (2D projection)