

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/221126786>

# K-Voronoi diagrams computing in arbitrary domains.

Conference Paper in Proceedings / ICIP ... International Conference on Image Processing · January 2003

DOI: 10.1109/ICIP.2003.1246838 · Source: DBLP

CITATION

1

READS

59

4 authors, including:



**Rubén Cárdenes**

HENSOLDT Sensors

71 PUBLICATIONS 676 CITATIONS

[SEE PROFILE](#)



**Simon K Warfield**

Harvard University and Boston Children's Hospital

614 PUBLICATIONS 21,906 CITATIONS

[SEE PROFILE](#)



**Juan Ruiz-Alzola**

Universidad de Las Palmas de Gran Canaria

94 PUBLICATIONS 1,229 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Motion-compensated DCE-MRI for functional imaging of kidneys [View project](#)



Population studies of the brain microstructure [View project](#)

# K-VORONOI DIAGRAMS COMPUTING IN ARBITRARY DOMAINS

Rubén Cárdenes<sup>1,3</sup>, Simon K. Warfield<sup>2</sup>, Andrea J.U. Mewes<sup>2</sup> and Juan Ruiz-Alzola<sup>3</sup>

<sup>1</sup>Dept. Ingeniería Telemática, Universidad de Las Palmas de GC, SPAIN

<sup>2</sup>Harvard Medical School and Brigham and Women’s Hospital, Dept. Radiology, USA

<sup>3</sup>Medical Technology Center, Univ. Las Palmas GC & Gran Canaria Dr. Negrín Hospital, SPAIN

E-mail:ruben@ctm.ulpgc.es, {warfield,mewes}@bwh.harvard.edu and jruiz@ctm.ulpgc.es

## ABSTRACT

We propose a novel algorithm to compute Voronoi diagrams of order  $k$  in arbitrary 2D and 3D domains. The algorithm is based on a fast ordered propagation distance transformation called *occlusion points propagation geodesic distance transformation* (OPPGDT) which is robust and linear in the domain size, and has higher accuracy than other geodesic distance transformations published before. Our approach has proved to have a computational complexity of order  $O(k.m)$  with  $m$  the domain size and  $k$  the order of the diagram. Voronoi diagrams have been extensively used in many areas and we show here that Voronoi diagrams computed in non convex domains, are extremely useful for the segmentation of medical images. We validated our algorithm with a set of 2D and 3D synthetic non convex domains, and with the segmentation of a medical dataset showing its robustness and performance.

## 1. INTRODUCTION

A Voronoi diagram is a subdivision of the space into regions such that all the points of a region have the same closest object site. Object sites will be the points from which we will compute our diagrams. The Voronoi diagram and its variants (different metrics, higher dimensions, sites which are segments or polygons instead of points, etc.) have been rediscovered many times in literally dozens of fields, including biology, crystallography, geology, metallurgy, meteorology, mathematics, robotics, geography, and marketing. Some good surveys of the types of Voronoi diagrams and their applications in a variety of fields can be found in Okabe et al. [1] and Aurenhammer et al. [2].

One of the many variants, is the Voronoi diagram of order  $k$  or  $k$ -Voronoi diagrams, which is a partition of the space into regions such that all the points in a given region

have the same  $k$  closest sites. Applications include image segmentation [3], interpolation, rank order filtering, clustering, medial axis transform, mesh generation, route planning, and curve and surface reconstruction [4]. One of them, conveying a high relevance for medical imaging, is shown in section 4, with the segmentation of the knee cartilage derived from a medical dataset.

In this paper we present a new algorithm to compute the discretized representation of the  $k$ -Voronoi diagrams, called the  $k$ -nearest-neighbor transform, with the addition that it can be computed in any arbitrary 2D or 3D domain, convex or non-convex. We will call to this variant geodesic  $k$ -Voronoi diagrams, and geodesic  $k$ -nearest-neighbor transform to its discretized representation, which consist of computing a subdivision of the space into regions such that each point of the region has the same  $k$  closest object sites according to a geodesic metric defined in arbitrary domains. This algorithm can be considered the generalization to arbitrary domains of the  $k$ -nearest-neighbor transform proposed by [5] for convex domains.

The Voronoi diagram is computed by first obtaining the distances from the objects sites, using a distance transformation (DT). The distance transformation of a binary image consisting of object and non object pixels, is the operation that computes for every pixel, the distance to the nearest object pixel. There are several implementations for computing efficiently DT, most of them take advantage of the fact that distances vary smoothly in the distance map, so that it must be possible to deduce the value of the map in one pixel from the values of the map around it. Thus many DT algorithms are based on mask propagations like in Rosenfeld [6] and Borgefors [7]. In the paper presented by Verwer [8], a new approach is introduced, that consists of propagating the objects in increasing order, from the objects itself ranked first to the rest of the image. With this idea, several DT algorithms have been developed, for example see Ragnemalm [9] and Cuisenaire [5]. The later proposed a new approach in this context to compute geodesic DT [10]. A geodesic DT is the operation that computes the nearest distances to the objects for every pixel, constrained

---

The first author is funded by a FPU grant from the University of Las Palmas de Gran Canaria. This work has been partially supported by the Spanish Ministry of Science and Technology and European Commission, co-funded grant TIC-2001-38008-C02-01

to an arbitrary domain, i.e. computing the distances as the shortest path length. Several geodesic DT algorithms have been proposed before, as the  $B_d$ -geodesic DT introduced by [5], based on ordered propagation, but it is an approximation of the geodesic version of the Euclidean DT. We propose to use a novel geodesic DT also based on ordered propagation, which is highly efficient and which uses a new geodesic metric in order to obtain the shortest path length more accurately than any other algorithm proposed before. The key point is the detection of occlusion points in a domain from a selected point of view, figuring out the location of obstacles, and starting a new propagation front from there. Our geodesic DT is termed *occlusion points propagation geodesic DT* and its details can be found in [11].

## 2. ORDER K GEODESIC VORONOI ALGORITHM

We will describe here the algorithm to compute geodesic  $k$ -Voronoi diagrams. For simplicity, we will describe the 2D algorithm. The extension to 3D is straightforward. We start from a set of objects sites  $O$ , from which we want to compute the Voronoi diagram, restricted to a generic domain  $\mathcal{M}$ .

The order one Voronoi diagram, is computed generating wavefronts or propagation fronts from every object site  $O_i$  until all the domain has been filled. Every non-connected object site  $O_i$  generates a single non-connected propagation front  $P_i$  that grows at the same rate, until they collide with themselves or with the domain boundaries. Each of the wavefronts will generate a unique Voronoi cell, which is a region whose points have the same closest object site. If two fronts ( $P_i$  and  $P_j$ ) collide at pixel  $p$ , object sites  $O_i$  and  $O_j$  are equidistant, so a Voronoi cell border is generated at  $p$  and propagation fronts  $P_i$  and  $P_j$  do not need to be propagated further at this point. We will refer to these set of propagation fronts as level one propagation fronts  $P_i^1$ .

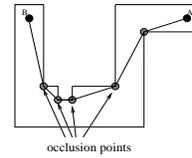
In order to compute  $k$ -Voronoi diagrams, with  $k > 1$ , we propose a novel scheme of multi-level propagation, where the  $i^{th}$  propagation level computes the cells whose points share the same  $i^{th}$  closest object site. In this algorithm we obtain the  $k$ -nearest neighbor transform which is the discretized representation of the  $k$ -Voronoi diagram. This representation allows a direct identification of the  $i^{th}$  closest object site in a given region. This is quite useful in some applications as in [3] and [5], to identify the  $k$  nearest neighbors from a given point in the image. To perform this multi-level technique the propagation fronts at level one should go on propagating after they collide. In this case, when two fronts at level one  $P_i^1$  and  $P_j^1$  collide at pixel  $p$ , a new propagation level appears (level 2), and those propagation fronts becomes  $P_i^2$  and  $P_j^2$ , at  $p$ , and continues propagating further. The second and following propagation levels are then a continuation of the propagation fronts at level one, and they will fill the domain as many times as the order of the

Voronoi diagram.

In order to account for convex and non convex domains, we compute the distances with a novel geodesic metric called occlusion points geodesic metric, which is defined as the shortest path between two points A and B in the domain, such that the path is a chain of segments through the nearest points that hides behind the obstacles or corners, (occlusion points), see figure 1 and [11] for details. For this purpose, for every pixel  $p$  in the domain, we store its coordinates  $\vec{p} = (p_x, p_y)$ , and at propagation level  $l$ , we store the coordinates of the nearest occlusion point from  $p$ :  $\vec{r}_{obj}^l(\vec{p}) = (x_{obj}^l(\vec{p}), y_{obj}^l(\vec{p}))$  and the distance from this occlusion point to the initial object site,  $d_{obj}^l(\vec{p})$ . When a new pixel  $p$  is reached at distance  $d + 1$ , from the propagation of a pixel  $q$  at distance  $d$ , we calculate the new distance as follows:

$$d_{new}(\vec{p}) = dist_{Euclidean}(\vec{p}, \vec{r}_{obj}(\vec{q})) + d_{obj}(\vec{q}) \quad (1)$$

the Euclidean distance from  $p$  to the nearest occlusion point from  $q$  ( $\vec{r}_{obj}(\vec{q})$ ), plus the distance from that occlusion point to the starting object  $d_{obj}(\vec{q})$ . Therefore the algorithm does not propagate absolute distances; instead, it propagates the occlusion point coordinates and the distance from this occlusion point to the original object site from where the propagation front started.



**Fig. 1.** Occlusion points and geodesic path in a non convex domain

The propagation fronts are implemented as two lists ( $list1$ ,  $list2$ ), where every list element represents a pixel in a propagation front, and consists of its coordinates, and the object index from which it originally started.  $list1$  stores pixels belonging to the propagation front at distance  $d$ , and  $list2$  stores pixels at distance  $d + 1$ . The algorithm starts putting the object points  $O$  in  $list1$ , and propagates the elements of a list to the other iteratively until both lists are empty, i.e. when no more pixels remain unreached in the domain.

In 2D we propagate using a neighborhood of size 8,  $N_8$ , and the 3D version is implemented using a neighborhood of size 26,  $N_{26}$ .

## 3. COMPUTATIONAL COMPLEXITY AND MEMORY LOAD

The computational complexity derives from the number of distance computations, assignments, and the number of comparisons made in the propagations. The propagation is only

executed once for every points in the domain  $\mathcal{M}$ , for that reason the number of distance calculations in this algorithm is the domain size:  $m$ . There are a number of additional distance computations corresponding to the collision of different propagation fronts, but they are in general negligible with respect to  $m$ . With respect to the number of comparisons made at every propagation, the algorithm needs a minimum of 5 comparisons for every pixel propagation, and could need a maximum of  $6+k$  comparisons at special points, when two or more propagations fronts collide. Thus, our algorithm has a computational complexity of order  $O(m)$  for every propagation level, and the overall computational complexity is  $O(m.k)$ , for a  $k$  order Voronoi diagram.

On the other hand the memory load is moderately high, because the storage consist of  $3.k.m$  values for the output, the occlusion points, and distance to occlusion points, plus  $4.m$  values for the domain mask and three auxiliar parameters for every pixel. Adding  $N$  values for the prototypes, the overall memory load is  $3.k.m + 4.m + N$ . In practice we use the size of the bounding box of the domain  $M$  instead of  $m$ , that can be well managed by a single workstation.

number of pixels	22214	88856	355424
k=1	0.140	0.559	2.288
k=2	0.234	0.960	3.912
k=3	0.329	1.350	5.644
k=4	0.422	1.767	7.370

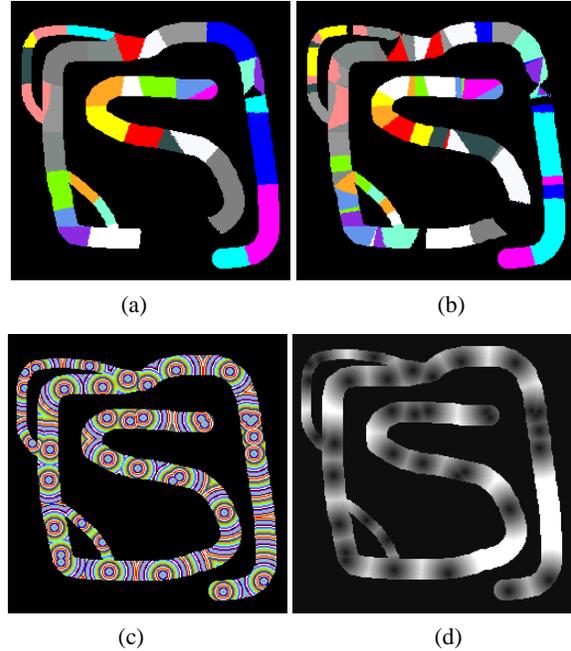
**Table 1.** Execution times in seconds for the 2D geodesic Voronoi algorithm in three experiments carried out for  $k = 1$  to  $k = 4$

number of pixels	120	1035	8291	66040
k=1	0.011	0.065	0.487	3.909
k=2	0.013	0.084	0.645	5.680
k=3	0.018	0.110	0.845	7.138
k=4	0.038	0.146	1.043	8.813

**Table 2.** Execution times in seconds for the 3D geodesic Voronoi algorithm in four experiments carried out for  $k = 1$  to  $k = 4$

## 4. RESULTS

We have tested the algorithm in a 2D synthetic non convex domain, (see figure 2) with 22214 pixels and two up-sampled domains, with domain sizes of 88856 pixels, and 355424 pixels. The 3D experiments have been carried out with four domain sizes of 120, 1035, 8291 and 66040 voxels. Figure 2 shows a Voronoi diagram of order one and order two, and the geodesic distance transformation restricted



**Fig. 2.** Order one Voronoi geodesic map (a), order two geodesic Voronoi map (b), geodesic distance map coded in a cyclic colormap (c), and geodesic distance map represented in a grayscale map (d), all restricted to a synthetic domain

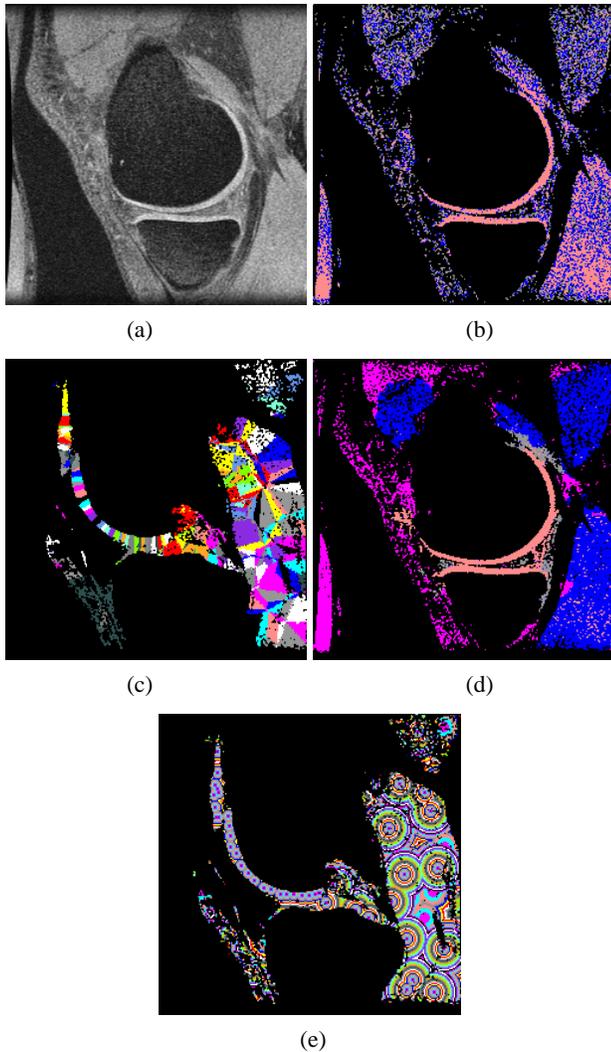
to the synthetic 2D domain from a set of 37 points.

Figure 3 displays two different segmentation results for the cartilage in the knee joint achieved from a 3D MR dataset. Figure 3 (b) has been carried out using a common kNN classifier, classifying every voxel according to the intensity level of a set of selected training prototypes. In comparison to this, figure 3 (d) shows the results after applying an additional channel to the classifier, by means of the geodesic Voronoi diagram obtained from the location of the same prototypes. The improvement of the segmentation using the geodesic Voronoi diagram is clearly shown. We also show in figure 4 a 3D model of the segmented cartilage.

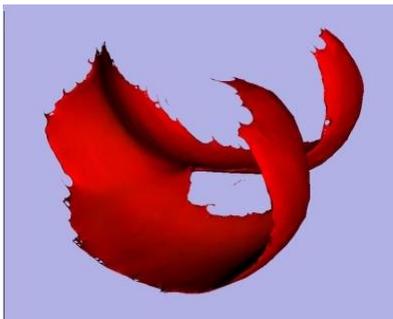
In tables 1 and 2 the execution times for the 2D and 3D experiments are shown respectively. The experiments have been carried out using several domain sizes and different values of  $k$ . The times has been measured in a SUN-Ultra 10 with an Ultra-SPARC II 440 MHz processor and 512 MB RAM. Notice that the times increase linearly with  $m$ , the number of points in the domain, and with  $k$  showing a computational complexity of  $O(m.k)$ .

## 5. CONCLUSIONS

As far as we know, the method introduced here is the first to compute high order geodesic Voronoi diagrams. The method is based on an efficient geodesic DT, that makes



**Fig. 3.** Sagittal slice of a MRI acquisition of a human knee (a), segmentation without Voronoi diagrams (b), geodesic Voronoi diagram of order two (c), segmentation using geodesic Voronoi diagrams (d) and geodesic distance map coded in a cyclic colormap (e)



**Fig. 4.** 3D model of the segmented cartilage

use of the occlusion points geodesic metric and on multiple levels ordered propagation to compute efficiently  $k$  order Voronoi diagrams. This algorithm is highly efficient and has proved to have a computational complexity of order  $O(m.k)$ , as we show in tables 1 and 2. With this scheme we can improve the performance of some applications computing Voronoi diagrams in reduced domains and not in the whole image, reducing the number of points of the diagram and thus, reducing the execution times. We can also take advantage of the geometry of the domain, using our occlusion points geodesic metric instead of the Euclidean metric.

We have also shown an application, such as the segmentation of medical images, which is of high interest in the medical image processing area. We are convinced that our algorithm can also be applied to a wide variety of areas where Voronoi diagrams are of routine application and where the search of  $k$  nearest neighbors in a generic domain is required, improving dramatically the accuracy and the performance.

## 6. REFERENCES

- [1] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tesselations: Concepts and Applications of Voronoi Diagrams*, Wiley, 1992.
- [2] F. Aurenhammer and R. Klein, "Voronoi diagrams," in *Handbook of Computational Geometry*, J. Sack and G. Urrutia, Eds., pp. 201–290. Elsevier Science Publishing, 2000.
- [3] S. Warfield, "Fast k-nn classification for multichannel image data," *Pattern Recogn. Let.*, vol. 17(7), pp. 713–721, 1996.
- [4] N. Amenta and M. Bern, "Surface reconstruction by voronoi filtering," 1999, vol. 22, pp. 481–504.
- [5] O. Cuisenaire and B. Macq, "Fast k-nn classification with an optimal k-distance transformation algorithm," *Proc. 10th European Signal Processing Conf.*, pp. 1365–1368, 2000.
- [6] A. Rosenfeld and J.L. Pfaltz, "Distance functions on digital pictures," *Pattern Recognition*, vol. 1(1), pp. 33–61, 1968.
- [7] G. Borgefors, "Distance transformations in arbitrary dimensions," *Computer Vision, Graphics and Image Processing*, vol. 27, pp. 321–345, 1984.
- [8] B.H. Verwer, P.W. Verbeek, and S.T. Dekker, "An efficient uniform cost algorithm applied to distance transforms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11(4), pp. 425–429, 1989.
- [9] I. Ragnemalm, "Neighborhoods for distance transformations using ordered propagation," *CVGIP, Image Understanding*, vol. 56, no. 3, pp. 399–409, 1992.
- [10] O. Cuisenaire, *Distance Transformations: fast algorithms and applications to medical image processing*, Ph.D. thesis, UCL, July 1999.
- [11] R. Cárdenes, S.K. Warfield, E. Macías, and J. Ruiz-Alzola, "Occlusion points propagation geodesic distance transformation," in *ICIP*, Sept. 2003.