Computing 79, 185–196 (2007)
DOI 10.1007/s00607-006-0197-8
Printed in The Netherlands

Computing

Fast Voronoi modeling

H. Hagen, M. Schneider, M. Schlemmer, M. Ruby, and **I. Scheler,** Kaiserslautern

Received January 5, 2006; revised May 16, 2006
Published online: March 7, 2007

Abstract

The common way to construct Voronoi tessellations is to compare the distances between given reference points using a given distance function. To generalize this distance-function concept we expand an existing approach which defines distance functions by their "unit circles". Our new approach allows modeling the "unit circles" by a closed Spline curve. Changing the control polygon directly affects the tessellation's appearance. Typically generalized Voronoi diagrams are represented by Voronoi vertices and curves separating the individual tiles. To obtain interactive modeling we extended an existing hardware accelerated rendering approach computing a bitmap-representation using different colors for individual tiles. With our extension, we are able to use our Spline distance representations as input for a growing process. This growing process easily takes into account weighting approaches like multiplicative, additive, and even free functional weighting.

AMS Subject Classifications: 51H20, 51H99.

Keywords: Scientific visualization, geometric modelling, Voronoi diagrams.

1. Introduction

Voronoi diagrams are used to handle scattered data in many application areas. The mathematical fundamentals of this technique were independently proposed by Voronoi [17], [18] and Dirichlet [9]. In most cases the applied distance function is simply the Euclidean metric. Nevertheless, there are applications where the Euclidean distance is not sufficient. For example, in Environmental Engineering several attributes could be linked to spatial information using this method. Another important goal is to generate Euclidean Voronoi diagrams on arbitrary surfaces.

Generalizations of Voronoi diagrams for example the application of Minkowski metrics have been discussed by several authors, e.g., Okabe et al. [15], Aurenhammer et al. [1]–[4], and Klein [13], [14].

Another way to extend Voronoi diagrams is to apply special weighting functions. Some examples for weighting can be found in the book of Okabe et al. [15]. Our approach combines these weighting methods with a new way of constructing distance functions. These are in our case not required to be real metrics. A distance function is described as a contour line, being represented by a Jordan curve. In our approach, this curve is modeled by using NURBS. For efficient calculation, we

Fig. 1. Interactive Voronoi modeling by manipulation of the distance function

extend hardware accelerated methods, that have been developed by Hoff et al. [12]. We also show a new way to efficiently implement arbitrary weighting functions for the Euclidean metric.

The paper is structured as follows: in Sect. 2, we give an introduction to Voronoi diagrams, weighting, and common metrics. Then, the foundations of the extension to arbitrary distance functions are presented in Sect. 3. Sections 4 and 5 contain our two main contributions. We explain in Sect. 4 our extension of the algorithm of Hoff et al. [12] for efficient calculation of generalized Voronoi diagrams using NURBS to model arbitrary distance functions. Our new efficient method for generating arbitrarily weighted Euclidean Voronoi diagrams is described in Sect. 5. In Sect. 6, we finally present results of the software we developed for fast Voronoi modeling.

2. Voronoi diagrams

2.1. State of the art

Voronoi diagrams can be used for clustering by k-means. Having a set of scattered data points A in $\mathbb{R}^2$, the construction of a Voronoi diagram leads to the so called Delaunay triangulation (see Delaunay et al. [8]) of this data. For a point $p_i \in A$, the Voronoi region V_i of p_i is defined as the set of all points p in $\mathbb{R}^2$ being closer to the corresponding reference point p_i than to any other point $p_{j \neq i}$ in the plane. A Voronoi region V_i of p_i is therefore the set

$$V_i = \{p | d(p, p_i) < d(p, p_j), j \neq i\} \tag{1}$$

containing all points being closer to the Voronoi seed point p_i than to any other one. The collection of those Voronoi regions $V_1, V_2 \cdots V_n$ constitutes the generalized Voronoi diagram (VD) of A.

The plane is partitioned completely with respect to the chosen metric $d(p, p_i)$, usually being the Euclidean distance function. Assuming the Euclidean metric, without

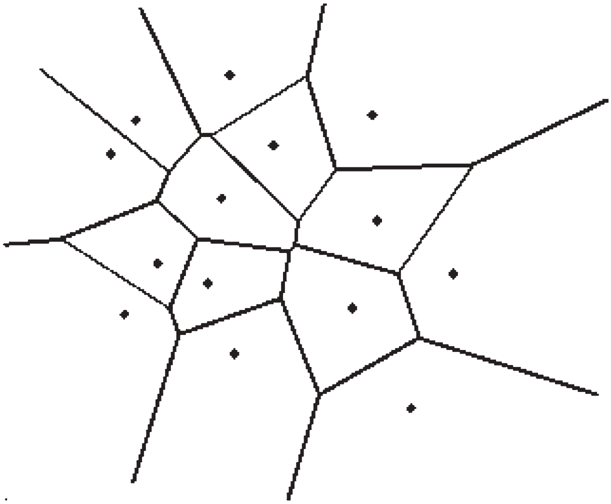

Fig. 2. Voronoi diagram with Euclidean metric, not weighted

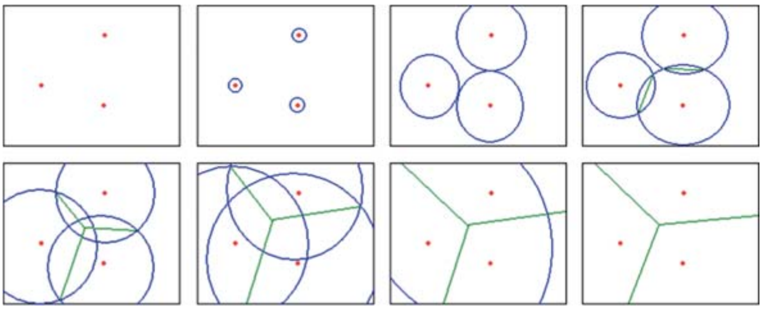

Fig. 3. Growing algorithm as proposed by Hagen et al. [11]

introducing any weighting one yields, for example, a Voronoi diagram with segments, as depicted in Fig. 2.

There are various ways of constructing Voronoi diagrams. A general but slow approach was presented by Hagen et al. [11], implementing a region growing algorithm. Starting from each Voronoi point the regions are growing, finally forming the bi-sectors of the Voronoi diagram. The growing algorithm allows a very general description of the distance function d. Figure 3 illustrates the growing process.

For simple cases, like the unweighted Euclidean metric, K. E. Hoff III et al. [12] propose a method accelerated by making use of graphics hardware. The idea is to draw the growing process of the Voronoi diagram in a higher dimension using OpenGL and project it into the plane. Figure 4 shows how this method works. For a 2D-Voronoi diagram with Euclidean metric as distance function, the growing process can be regarded as cones in 3D. In OpenGL cones are primitives, meaning that they can be drawn and intersected very efficiently. A 2D-projection in growth direction

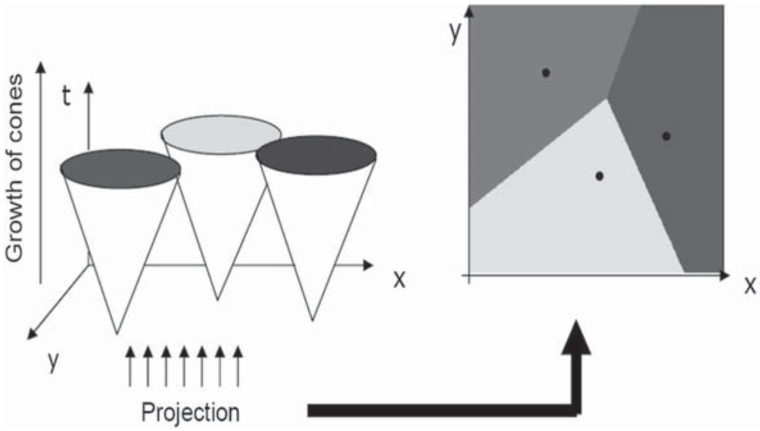

Fig. 4. Illustration of hardware accelerated Voronoi diagram construction as done by Hoff et al.

produces the 2D-Voronoi diagram. A drawback of this method is, that it does not automatically deliver the bi-sectors, but a bitmap representation of the Voronoi tiles.

The authors also show, how Voronoi diagrams of higher order sites can be computed. With this extension, Voronoi diagrams can not only be generated from start points, but from lines or curves. However, the change of the metric, being another degree of freedom in constructing general Voronoi diagrams, has not been discussed. Therefore, we propose another way of generalization. Keeping points as start location, we provide tools to construct and efficiently calculate Voronoi diagrams with selectable, possibly non-convex distance functions.

Voronoi diagrams can also be used for weighted interpolation, regarding for example the areas of the Voronoi regions. Therefore, it makes sense to introduce weighted Voronoi diagrams, as described, e.g., in Okabe et al. [15]. Let w_i be weights at the Voronoi points. An additive weighted Voronoi-diagram can be obtained by changing the distance function d to

$$d_{aw}(p, p_i) = \| p - p_i \| - w_i. \tag{2}$$

A multiplicative Voronoi diagram has according to [13] the distance function

$$d_{mw}(p, p_i) = \frac{1}{w_i} \| p - p_i \|, \qquad w_i > 0. \tag{3}$$

Therefore, a mixed weighting can be obtained by combining these formulas [13]:

$$d_{cw}(p, p_i) = \frac{1}{w_{i1}} \| p - p_i \| - w_{i2}, \qquad w_{i1} > 0. \tag{4}$$

Different forms of weighting functions were also used in the mentioned algorithm by Hagen et al. [11].

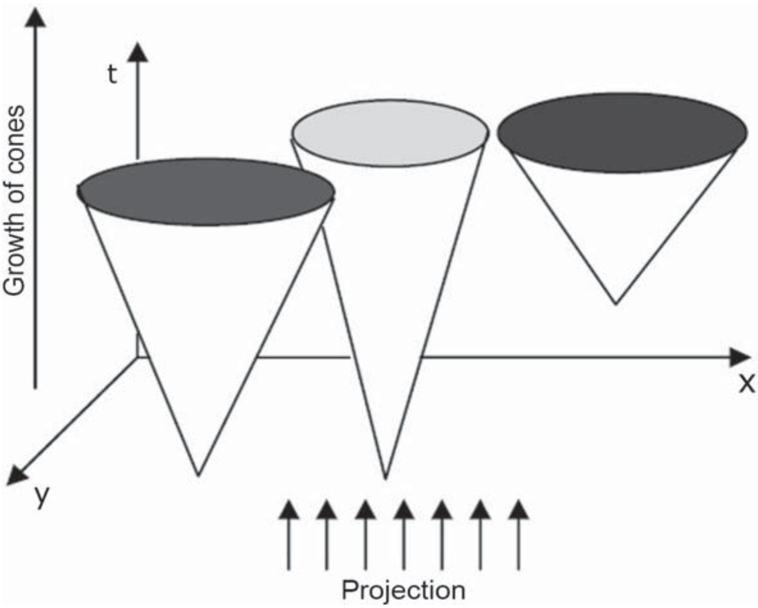

Fig. 5. Hoff et al.'s algorithm modified by weighting

2.2. Extensions

The method of Hoff et al. [12] can also be extended by additive and multiplicative weighting. As Fig. 5 shows, additive weighting can be introduced by arranging the Voronoi start points along the z-plane, whereas multiplicative weighting is controlled by the angle of the cones. A higher (lower) additive weight moves the start point towards (away from) the eye position, a higher (lower) multiplicative weight induces a larger (smaller) growth angle of the cone in 3D.

Another extension of Voronoi diagrams is the choice of an arbitrary distance function. Instead of using a metric, we insert a general distance function being described by a Jordan curve. A Jordan curve is also called a simple closed curve. Our approach allows the user to interactively control his self-defined Jordan curve while watching the resulting Voronoi diagram. B-Splines and Nonuniform Rational B-Splines (NURBS) are used as modeling techniques and for efficient calculation.

3. B-Splines and NURBS

For the modeling of curves, one can use various techniques. Given a set of control points, one can use Lagrangian interpolation, Bezier curves, B-Splines, and other methods to construct a curve. With Lagrangian interpolation and Bezier curves, one has no local control of the curve. In contrast, B-Splines offer local control.

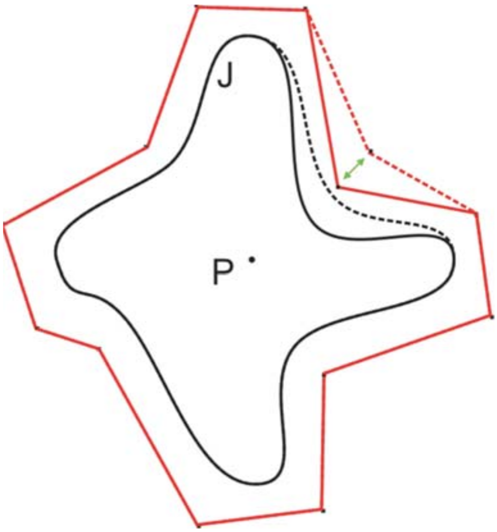

Fig. 6. Interactive modification of an example distance function

Given a knot-vector $(t_j)_{j=0,\ldots,n}$, B-spline basis functions can be defined recursively:

$$N_j^1(t) := \begin{cases} 1 & t_j \leq t \leq t_{j+1} \\ 0 & \text{else} \end{cases}, \tag{5}$$

$$N_j^k(t) := \frac{t - t_j}{t_{j+k-1} - t_j} N_j^{k-1}(t) + \frac{t_{j+k}}{t_{j+k-1} - t_j} N_{j+1}^{k-1}(t) \tag{6}$$

$$\text{for } k > 1, \quad j = 0, \ldots, n - k.$$

In addition to polynomial modeling between knots, one can use rational functions such as nonuniform rational B-splines (NURBS). NURBS extend the B-spline concept by introducing additional weight functions, resulting in a higher degree of freedom in modeling of the curves.

Let $(d_i)_{i=0,\ldots,m}$ be a set of $m + 1$ control points and $(w_i)_{i=0,\ldots,m}$ corresponding weights. Then, a NURBS curve $K(t)$ can be created as [16]

$$K(t) = \frac{\sum_{i=0}^{m} w_i \cdot d_i \cdot N_i^k(t)}{\sum_{i=0}^{m} w_i \cdot N_i^k(t)}. \tag{7}$$

NURBS offer a high degree of freedom in modeling, even if there are only few control points given. Therefore, we will only use the NURBS concept for our method.

4. Arbitrary distance functions algorithm

Our algorithm for modeling Voronoi diagrams requires a Jordan curve as distance function. This means a closed curve having no self-intersections. Another require-

Fig. 7. Resulting Voronoi regions (colored), control points (white), one of the disconnected regions is highlighted in the white circle

ment is that the Voronoi start point should be inside the closed area, bounded by the closed curve, as presented in Fig. 6.

In our implementation, the curve can be modeled using B-Splines or NURBS. Given the control polygon and weights, the polygon is refined by application of the De-Boor algorithm [7], [10] according to a given level of detail. We then apply Hoff et al.'s idea by extending the polygon in 3D to an object, similar to a cone or a pyramid. The object is positioned and sized according to the given weighting as described before. After the application of this procedure to each Voronoi point, the scene is rendered in OpenGL and also projected into 2D. The result is the weighted Voronoi diagram for the arbitrarily chosen distance function and weighting. The new feature is that one can interactively control the distance function and see the changes in the Voronoi diagram immediately. This provides great insight into the evolution of Voronoi diagrams. By interactively changing the distance function (see Fig. 6), we can for example avoid (or on the other hand trigger) unconnected regions in our resulting Voronoi diagram. Figure 7 shows an example of Voronoi tiles with unconnected regions.

5. Arbitrary weighting algorithm

In contrast to the algorithm in the last section dealing with different metrics, the method presented in this section provides an efficient method for the construction of Euclidean Voronoi diagrams having general weighting functions.

The main idea is, to construct our weighting function as a NURBS curve in a higher dimension. For a 2D-Voronoi diagram in the xy-plane, we construct the curve in the xz-plane. Rotating this curve around the z-axis yields a rotational solid, containing the Voronoi seed point. This proceedure is performed for each seed point. Finally, the projection of all of those solids into the xy-plane yields the arbitrarily weighted Voronoi diagram. The specific weights are described by functions, modeled with the discussed NURBS curves.

We now formulate the method more precisely:

As Euclidean Voronoi diagrams can be generated by a projection of cones (Hoff [12]), we know that additive and multiplicative weighting can be controlled by the position (height) and the angle of the cone, respectively. Let n Voronoi points v_i be given as

$$v_i = \begin{pmatrix} v_{i_1} \\ v_{i_2} \\ 0 \end{pmatrix}, \quad i = 1, ..., n$$

with v_{i_1} and v_{i_2} as 2D Voronoi seeds. Then, we can construct k additional control points $p_{i,k}$for each v_i, $k \in \{1, ..., m_i\}$, with $p_{i,1_3} = 0$ and $p_{i,k_3} > p_{i,k-1_3}$ for $k > 1$. W.l.o.g., we choose $p_{i,k_2} = 0$, so that we obtain points lying in the xz-plane. We can also define a specific weight $w_{i,k}$ for each of those control points. This is everything we need for building our desired NURBS curves. They have Voronoi points as start points, are located in the xz-plane, and have the property of being monotonously increasing with respect to the z-axis.

Rotating those curves around their specific rotation axes (always parallel to the z-axis) given by the straight lines

$$a_i(t) = \begin{pmatrix} a_{i_1} \\ a_{i_2} \\ t \end{pmatrix}, \quad i = 1, ..., n \quad t \in \mathbb{R}$$

for each Voronoi seed point, we construct rotational solids. A suitable projection of those solids in 2D (from below onto the in z-direction shifted xy-plane) delivers the Euclidean Voronoi diagram for arbitrary weighting functions, being described by the control points and weights. Figure 8 illustrates this concept as extension of the method of Hoff et al. [12]. By using NURBS curves, we are able to rebuild or integrate additive and multiplicative weighting, as well as providing more general functions.

Of course, with some badly chosen conditions we do not obtain a real Voronoi tessellation. That means, when the growth of the Voronoi regions is strongly limited by our curves, empty regions not associated with any of the Voronoi points can appear. We have found some weak conditions to over-come this issue. For example, we can postulate the last point of one of our curves being located outside a certain radius (in x-direction), so that this Voronoi region is capable of covering the whole projection area if necessary. Choosing this radius infinite, solves the problem for arbitrary settings, in practice it is sufficient to use the viewport size as radius.

The concept of drawing and projecting rotational solids is OpenGL-supported and therefore provides a fast method for generation and controllable observation of general weighting functions for Euclidean Voronoi diagrams.

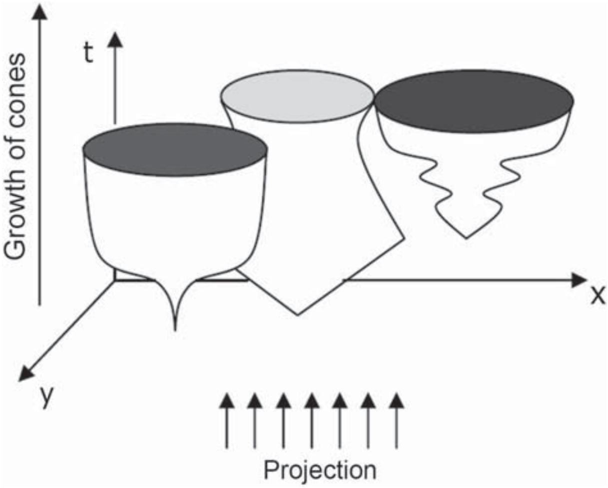

Fig. 8. Fast generation of arbitrarily weighted Voronoi diagrams

6. Results

In this section, we show examples of the developed tool for generating additively and multiplicatively weighted, controllable distance function based Voronoi diagrams.

An example for a modeled distance function is shown in the upper left of Fig. 9. The case when all Voronoi points have this distance function is also shown in Fig. 9. The upper right of the figure shows the unweighted case, in the second row one can see the additively and multiplicatively weighted Voronoi diagrams, respectively. The lower left shows the compoundly weighted version of the diagram. In this image there is no region being topologically separated from its Voronoi seed point. The lower right shows an example of the multiplicative case.

More applications of our methods can also be found in Environmental Engineering. The special application area we are working is the field of re-use of former military land. The last 20 years have brought the closure of several military bases throughout Germany. The actual situation and the forthcoming years will bring about more land which will be available for civilian use because of troop reduction, and these sites have to be redeveloped. Base redevelopment initializes a complex planning process that needs a high structural effort. The need of re-planning the area for different utilizations is expected.

Commonly used systems in this area are only suitable for explaining and illustrating the planning process. There ist a lack of goal oriented tools for interpretation of the mostly unstructured data. The need of re-planning conversion areas for different utilizations is expected. Generalized Voronoi diagrams therefore are interesting tools for environmental engineers. A large number of actors are involved in the process of redevelopment and re-use. The main actors are the county, the investor and the vendor. The different views can be defined by an individual weighting of hard and soft location factors, represented by the weighted Voronoi diagrams.

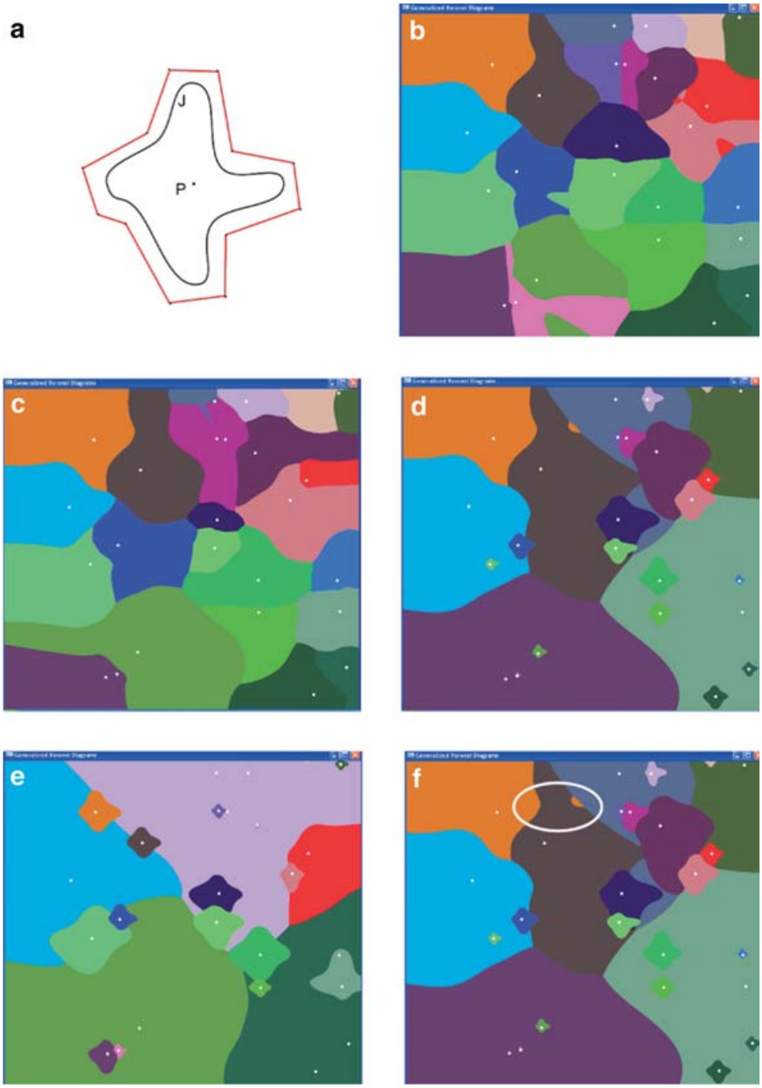

Fig. 9. Fast Voronoi modeling with a controllable distance function (**a**), results for unweighted case (**b**), additively (**c**), multiplicatively (**d+f**) and compoundly (**e**) weighted cases. A topological separation is highlighted in (**f**)

In application of Voronoi diagrams in this field, for example, regions that are topologically separated from their Voronoi origin point are not welcome. In this case, it will be impossible for the actor to match the hard and soft location factors with the region. Our tool enables us to model our Voronoi diagram interactively by changing the distance functions for given control points.

Another possible application can be the construction of Euclidean Voronoi tessellations on arbitrary surfaces. Our second algorithm even provides an efficient way of free weighting for the Euclidean case. It has been implemented separately from the first algorithm and still has to be integrated into our system.

7. Conclusion

We presented new methods for the interactive modeling of Voronoi diagrams. Our two major contributions are the extensions of the efficient algorithm of Hoff et al. [12] regarding arbitrary metrics on the one hand, and free functional weighting on the other hand. Having given a Voronoi seed point distribution, we can control the appearance and properties of the resulting Voronoi diagram by manipulation of the distance function and weighting. The distance function describes the propagation of the growing Voronoi region. We furthermore presented an efficient method for generation of arbitrarily weighted Euclidean Voronoi diagrams. With this high degree of freedom, we are able to observe how changes in distance function or weighting affects the resulting Voronoi diagram immediately.

Applications are for example in the area of Environmental Engineering. The planning process itself is divided into several steps during the redevelopment of conversion areas. Our work is focused on the profiles of different sites and their analysis based on the need of the decision makers. A deciding aspect is that a lot of restrictions during the change of use are given by multiple parameters in the sense of soft and hard location factors. On the other hand, e.g. regions inhering disconnected areas should be avoided, holes within an area are special, undesireable artefacts. The possibility of seeing the effects of changing the metric and weighting enables Environmental Engineers to identify possible heuristics for the distance and weighting functions more easily.

The arbitrary weighting algorithm has been implemented separately and has to be integrated in our main software. We also plan to combine both techniques to construct arbitrarily weighted Voronoi diagrams with controllable distance functions. Future applications of our methods will be the generation of Voronoi diagrams on arbitrary surfaces.

Acknowledgements

The authors would like to thank Ariane Middel, Martin Bertram, and especially Frank Leissing for their support. This work is a project of the International Research Training Group "Visualization of Large and Unstructured Data Sets" funded by the German Science Foundation (DFG).

References

[1] Aurenhammer, F., Klein, R.: Voronoi diagrams. In: Handbook of computational geometry (J.R. Sack and J. Urrutia, eds.). Elsevier 1999.

[2] Aurenhammer, F.: The one-dimensional weighted Voronoi diagram. Information Processing Lett. *22*(3), 119–123 (3 March 1986).

[3] Aurenhammer, F.: Voronoi diagrams-a survey of a fundamental geometric data structure. ACM Comput. Surv. *23*(3), 345–405 (1991).

[4] Aurenhammer, F., Edelsbrunner, H.: An optimal algorithm for constructing the weighted voronoi diagram in the plane. Technical report, University of Graz, 1983.
[5] Chew, A., Drysdale, S.: Voronoi diagrams based on convex distance functions. In: COMPGEOM: Annual ACM Symposium on Computational Geometry, 1985.
[6] de Boor, C.: On calculating with B-spline. J. Approx. Theor. pp. 50–62, 1972.
[7] de Boor, C.: A practical guide to splines. Berlin: Springer 1978.
[8] Delaunay, B. N.: Sur la sphere vide. Bull. Acad. Sci. USSR VII: Class. Sci. Math. pp. 793–800 (1934).
[9] Dirichlet, G. L.: Über die Reduktion der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. J. Reine Angew. Math. *40*, 209–227 (1850).
[10] Farin, G.: Curves and surfaces for computer aided geometric design: A practical guide. Academic Press 1988.
[11] Hagen, H., Ruby, M., Scheler, I., Schneider, M.: Topology-preserving distance functions for tessellations. In: Proc. 5th IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP 2005) (J.J. Villanueva, eds.), pp. 674–679, Anaheim, Calgary, Zürich, 2005. ACTA Press.
[12] Hoff III, K. E., Culver, T., Keyser, J., Lin, M., Manocha, D.: Fast computation of generalized Voronoi diagrams using graphics hardware. In: Proc. Conf. Computer Graphics (Siggraph99) (Alyn Rockwood, ed.), pp. 277–286. N.Y., August 8–13 1999. ACM Press.
[13] Klein, R.: Concrete and abstract Voronoi diagrams. Lecture Notes in Computer Science, Vol. 400. Springer 1989.
[14] Klein, R.: Algorithmische Geometrie. Addison-Wesley Publishing 1997.
[15] Okabe, A., Boots, B., Sugihara, K.: Spatial tesselations – concepts and applications of Voronoi diagrams. Chichester: Wiley 1992.
[16] Piegl, L.: On NURBS: A survey. IEEE Computer Graphics and Applications *11*(1), 55–71, January 1991.
[17] Voronoi, G.: Nouvelles applications des paramètres continus à la théorie des formes quadratiques – premier mémoire: Sur quelques propriétés des formes quadtratiques positives parfaites. J. Reine Angew. Math. *133*, 97–178 (1907).
[18] Voronoi, G.: Nouvelles applications des parametres continus a la theorie des formes quadratiques. J. Reine und Angew. Math. *134*, 198–287 (1908).

H. Hagen, M. Schneider, M. Schlemmer, M. Ruby and I. Scheler
IRTG "Visualization of Large and Unstructured Data"
Department of Computer Science
TU Kaiserslautern
Germany
e-mails: hagen@informatik.uni-kl.de
marco.schneider@informatik.uni-kl.de
schlemmer@informatik.uni-kl.de
ruby@rhrk.uni-kl.de
scheler@rhrk.uni-kl.de