COMPUTING DIGITIZED VORONOI DIAGRAMS ON A SYSTOLIC SCREEN AND APPLICATIONS TO CLUSTERING *

FRANK DEHNE

Center for Parallel and Distributed Computing
School of Computer Science, Carleton University, Ottawa, Canada K1S 5B6

Abstract. A *systolic screen* of size M is a $\sqrt{M} \times \sqrt{M}$ mesh-of-processors where each processing element P_{ij} represents the pixel (i,j) of a *digitized plane* Π of $\sqrt{M} \times \sqrt{M}$ pixels. In this paper we study the computation of the Voronoi diagram of a set of n planar objects represented by disjoint images contained in Π. We present $O(\sqrt{M})$ time algorithms to compute the Voronoi diagram for a large class of object types (e.g., points, line segments, circles, ellipses, and polygons of constant size) and distance functions (e.g., all L_p metrices).
Since the Voronoi diagram is used in many geometric applications, the above result has numerous consequences for the design of efficient image processing algorithms on a systolic screen. We obtain, e.g., an $O(\sqrt{M})$ time systolic screen algorithm for "optical clustering"; i.e., identifying those groups of objects in a digitized picture that are "close" in the sense of human perception.

1 INTRODUCTION

Consider a *digitized plane* Π of size M, i.e. a rectangular array of M lattice points, or *pixels*, with integer coordinates $(i,j) \in \{1,..., \sqrt{M}\}^2$, and a set $I_1, ... , I_n$ of n disjoint *images* in Π where an image (or digitized picture) I_i is defined as an arbitrary subset $I_i \subseteq \Pi$.

In this paper we study efficient parallel algorithms for processing such images. We consider the *mesh-of-processors* architecture; i.e., a set of m processors P_{ij} $(i,j \in \{1,..., \sqrt{M}\})$ arranged on a $\sqrt{M} \times \sqrt{M}$ grid where each processor is connected to its four direct neighbors, if exist. This architecture is particularly useful for image processing, since n disjoint images $I_1, ... , I_n$ in Π can be naturally represented on a mesh-of-processors of size M: Every processor P_{ij} has a *color-register* C-Reg(i,j) with value

$$\text{C-Reg}(i,j) = \begin{cases} k & \text{if } (i,j) \in I_k \ (1 \le k \le n) \\ 0 & \text{otherwise} \end{cases} .$$

* Research partially supported by the Natural Sciences and Engineering Research Council of Canada under Grant A9173.

For the remainder we will refer to a mesh-of-processors that represents a set of images as described above as a *systolic screen* (see Figure 1).

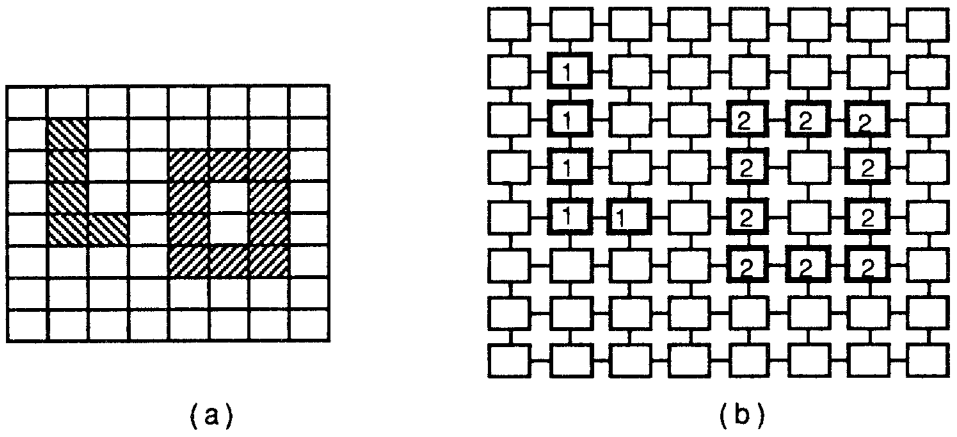

Figure 1: (a) Two Images in Π. (b) Systolic Screen Representation of these Two Images.

Systolic screen architectures have already been extensively used to manipulate images. A well known existing system is the MPP designed by NASA for analysing LANDSAT satellite data [Re84]. The MPP consists of 16,384 processing units organized in a 128x128 matrix where each processing unit, which has a local memory between 1K and 16K bits, represents a subsquare of pixels.

While most of the early applications of systolic screens considered "low level" image processing operations such as contour extraction or connected component labeling, recent research has also focussed on computing "high level" geometric operations on images. Miller and Stout [SM84], [MS85] have proposed $O(\sqrt{M})$ time algorithms for computing, e.g., the distance between two images, the convex hull, diameter, and smallest enclosing circle of an image. Dehne, Sack, and Santoro [DSS87] and Dehne, Hasenklover, Sack, and Sanotoro [DHSS87] have introduced $O(\sqrt{M})$ time algorithms for computing all nested rectilinear convex hulls of an image and for solving visibility problems on a systolic screen, respectively.

In this paper, we continue the study of algorithm design on a systolic screen, and consider the problem of computing the digitized Voronoi Diagram. We present an $O(\sqrt{M})$ time solution for computing the (digitized) Voronoi diagram of a set of n disjoint objects for a large class of object types (e.g., points, line segments, circles, ellipses, and polygons of constant size). The algorithm can compute the (digitized) Voronoi diagram for a number of distance functions which include, e.g., all L_p metrices. [1]

Since the Voronoi diagram is used in many geometric applications, the above result has numerous consequences for the design of efficient image processing algorithms on a systolic screen. In this paper we will present an $O(\sqrt{M})$ time systolic screen algorithm for "optical clustering"; i.e., identifying those groups of objects in a digitized picture that are "close" in the sense of human perception.

2 DIGITIZED VORONOI DIAGRAMS

Consider a set $S=\{s_1,...,s_n\}$ of n geometric objects in $\mathbf{R}^2$ (e.g., points, line segments, polygons, cicles, ellipses) and let $d : \mathbf{R}^2 \times \mathbf{R}^2 \rightarrow \mathbf{R}^+$ be a distance function.
The well known *Voronoi diagram* V(S) (see, e.g., [SH75]) partitions $\mathbf{R}^2$ into n *Voronoi regions*
$V(s_i) := \{x \in \mathbf{R}^2 \mid d(x,s_i) \leq d(x,s_j) \text{ for all } j \neq i\}$.
Every Voronoi region $V(s_i)$ consists of two disjoint parts, the *interior*
$IV(s_i) := \{x \in \mathbf{R}^2 \mid d(x,s_i) < d(x,s_j) \text{ for all } j \neq i\}$,
and the *border*

$$BV(s_i) := V(s_i) - IV(s_i).$$

$BV(S) := \bigcup_{1 \leq i \leq n} BV(s_i)$, the union of all borders, is usually referred to as the set of *Voronoi points* of V(S).

1 The problem of computing the digitized Voronoi diagram for point sets (for Euclidean and L_1 metric) on a mesh-of-trees architecture has recently been studied by Schwarzkopf [S88].

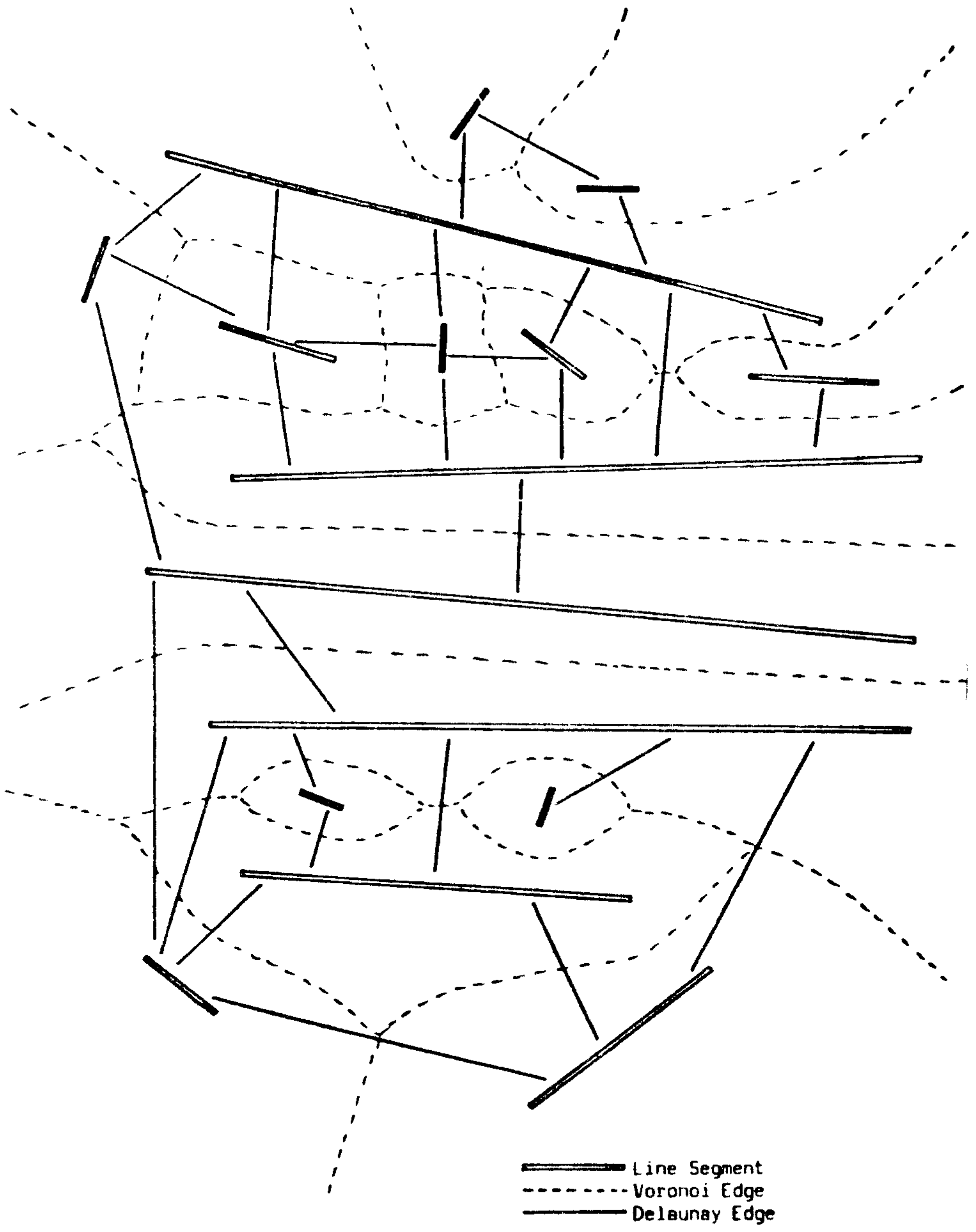

Figure 2: Voronoi Diagram for a Set of Line Segments.

In the remainder of this section, we will present how to translate the Voronoi diagram definiton to the digitized environment (see also [S88]).

We first introduce some definitions (cf., e.g., [Ro79] and [Ki82]):

- The *direct neighbors* of a pixel $(x,y) \in \Pi$ are the eight pixels $(x \pm 1, y)$, $(x, y \pm 1)$, $(x+1, y \pm 1)$, and $(x-1, y \pm 1)$. The *border* I^{o} of an image I is the set of all pixel of I which have a direct neighbor in $\Pi - I$. The *interior* of I, $I - I^{o}$, is denoted by $I^{\bullet}$.

- A *path* from $p \in \Pi$ to $q \in \Pi$ is a sequence of points $p=p_0,\dots,p_r=q$ such that p_i is a neighbor of p_{i-1}, $1 \le i \le r$. An image I is *connected* if for every $p,q \in I$ there exists a path from p to q consisting entirely of pixels of I. An image I which is connected is referred to as an *object.*
- With each pixel $p=(i,j) \in \Pi$ we associate its *cell* $<p>:= [i-0.5, i+0.5] \times [j-0.5, j+0.5] \subseteq \mathbf{R}^2$ and with each image $I \subseteq \Pi$ its *region*

$$<I>:= \bigcup_{p \in I} <p>.$$

- Conversely, we define for a set $R \subseteq \mathbf{R}^2$ its *image* $\mathrm{Im}(R) := \{ p \in \Pi \mid <p> \cap R \neq \emptyset \}$.

The *digitized Voronoi diagram* $V_d(S)$ can now be defined as follows:

Consider a set $S=\{s_1,\dots,s_n\}$ of n geometric objects $s_i \in <\Pi>$ such that $<s_i> \cap <s_j> = \emptyset$ for $i \neq j$; that is, consider a set S of n objects from "real geometry" such that their image representations in Π do not intersect. Let $d : \mathbf{R}^2 \times \mathbf{R}^2 \rightarrow \mathbf{R}^+$ be a distance function.

As described above, the standard Voronoi diagram V(S) induces a Voronoi region $V(s_i)$ for each object which consists of an interior $IV(s_i)$ and a border $BV(s_i)$.

The *digitized Voronoi diagram* $V_d(S)$ again consists of n *digitized Voronoi regions* $V_d(s_i)$, one for each object s_i. Each digitized Voronoi region consists of an *interior* $IV_d(s_i)$ and a *border* $BV_d(s_i)$ defined as follows:

- $BV_d(s_i) := \mathrm{Im}(BV(s_i))$
- $IV_d(s_i) := \mathrm{Im}(V(s_i)) - BV_d(s_i)$.

That is, the border of a digitized Voronoi region is the image of the border of the respective standard Voronoi region; the interior of a digitized Voronoi region consists of the remaining pixels in the image of the respective standard Voronoi region.

Consequently, the set $BV_d(S)$ of all *Voronoi pixels* of the digitized Voronoi diagram is defined as follows:

$$BV_d(S) := \bigcup_{1 \le i \le n} B_d(s_i) ;$$

i.e., the Voronoi pixels of $V_d(S)$ are obtained by computing the image of the Voronoi points of V(S).

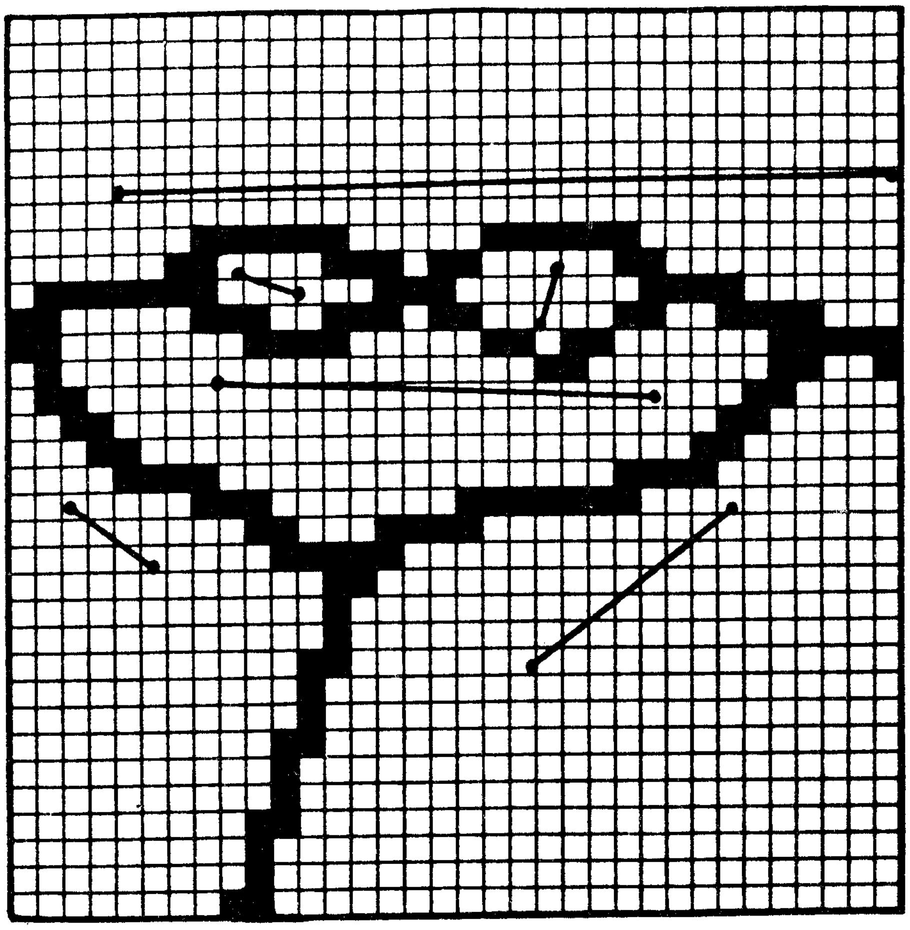

Figure 3: Digitized Voronoi Diagram for the Set of Line Segments of Figure 2. (The Black Pixels Represent the Voronoi Pixels.)

Note, that Voronoi points which do not intersect $<\Pi>$ are not represented in the digitized Voronoi diagram V(S) and that all Voronoi points which are contained in a cell $<p>$, $p \in \Pi$, are represented by one Voronoi pixel only.

In the following Sections 3 and 4 we will describe how to compute digitized Voronoi diagrams on a systolic screen. To simplify exposition, we will first consider the basic case of a set of points and Euclidean metric, and will then generalize our result to more general sets of objects and distance functions.

3 COMPUTING DIGITIZED VORONOI DIAGRAMS FOR POINT SETS AND EUCLIDEAN METRIC

Let $S=\{s_1,...,s_n\}$ be a set of n points in $<\Pi>$, and consider the Euclidean metric. (We assume that $Im(s_i) \cap Im(s_j) = \emptyset$ for $i \neq j$.)

We will now present an $O(\sqrt{M})$ time algorithm for computing the digitized Voronoi diagram $V_d(S)$ on a systolic screen of size M. The algorithm assumes as input that $Im(s_1),\ldots,Im(s_n)$ are represented on a systolic screen of size M as described in Section 1. The digitized Voronoi diagram of S will be reported by the systolic screen as follows:

Every processing element P_{ij} has a *Voronoi register*, V-Reg(i,j), and upon termination of the algorithm their values are

$$\text{V-Reg}(i,j) = \begin{cases} k & \text{if } (i,j) \in IV_d(s_k) \ (1 \le k \le n) \\ * & \text{if } (i,j) \in BV_d(S) \end{cases}$$

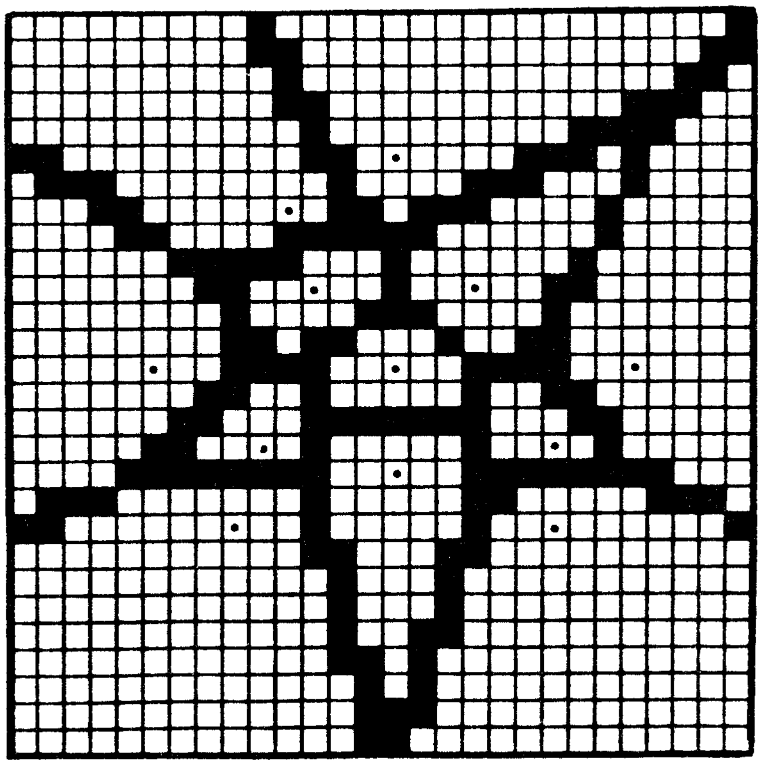

Figure 4: Digitized Voronoi Diagram for a Set of Points.

Before describing the algorithm, we need to introduce the following notations:

- Given a point $s \in \mathbf{R}^2$ and radius $r \in \mathbf{R}$, then $B(s,r) := \{x \in \mathbf{R}^2 / \ d(s,x) \le r\}$ denotes the *ball* with center s and radius r.
- The *processor distance* between two processors P_{ij} and $P_{i'j'}$ is their manhatten distance $|i-i'| + |j-j'|$.

Algorithm DIG-VOR:

```
(1) All Pij initialize their V-Register:  V-Reg(i,j) ← C-Reg(i,j)
(2) For t := 1 to $\sqrt{M}$ do
    (a) All $P_{ij}$ with V-Reg(i,j) = k > 0 send a message "k" to all $P_{i'j'}$ within processor distance
        $\lambda \in O(1)$ with V-Reg(i',j')=0 and $<(i',j')> \cap B(s_k,t) \neq \emptyset$.
    (b) All $P_{ij}$ with V-Reg(i,j) = 0 which receive only messages "k"  set V-Reg(i,j)← k.
    (c) All $P_{ij}$ with V-Reg(i,j) = 0 which receive at least two different messages "$k_1$" and "$k_2$"
        set V-Reg(i,j) ← *.
    (d) All $P_{ij}$ with V-Reg(i,j) = $k_1$ which receive a message "$k_2$" such that $B(s_{k_1},t) \cap$
        $<(i,j)> \neq \emptyset$ and $B(s_{k_2},t) \cap <(i,j)> \neq \emptyset$ set V-Reg(i,j)←*.
```

Theorem 1. *Algorithm DIG-VOR computes, on a systolic screen of size M, the digitized Voronoi diagram of a set S of n points in <Π>, for Euclidean metric, in time O($\sqrt{M}$).*

Proof. The minimum distance of a pixel $(i',j') \in B(s_k,t+1)$ from some $(i,j) \in B(s_k,t)$ is at most $\lambda \in O(1)$. Thus, in oder to send a message "k" from all pixels in $B(s_k,t)$ to the pixels in $B(s_k,t+1)$-$B(s_k,t)$, it suffices that each P_{ij} with $<(i,j)> \cap B(s_k,t) \neq \emptyset$ sends a message "k" to all processors within distance λ. Hence, at time t, a processor P_{ij} with $<(i,j)> \cap B(s_k,t) \neq \emptyset$ has either already received some message or does now receive a message "k".
Consider a processor P_{ij} with all points in <(i,j)> closer to s_k than to any other $s_{k'}$. There exists some minimum $t \in \{1, ..., \sqrt{M}\}$ such that $<(i,j)> \subseteq B(s_k,t)$ but $<(i,j)> \cap B(s_{k'},t') = \emptyset$ for all $k' \neq k$. Hence, before time t, processors P_{ij} has not received any message yet and, at time t, gets only "k" messages. Thus, at time t, P_{ij} sets its Voronoi register V-Reg(i,j) to k (Step 2b).
On the other hand, consider a processor P_{ij} with points $x \in <(i,j)>$ that have the same distance to two objects s_k and $s_{k'}$. For such a P_{ij}, there exists some time $t \in \{1, ..., \sqrt{M}\}$ such that at that time it receives a message "k_1" and either at the same time or later receives a message "k_2". In both cases, P_{ij} sets its Voronoi register V-Reg(i,j) to * (Step 2c and 2d, respectively).
Thus, the correctness of algorithm DIG-VOR follows.
Since the execution of Step 1 and Parts a, b, c and d of Step 2 take time O(1), each, the running time of algorithm DIG-VOR is O($\sqrt{M}$). ♦

4 Computing Digitized Voronoi Diagrams for Sets of Objects and Convex Distance Functions

After having solved the basic case of point sets and Euclidean metric, we will now generalize our result to other classes of objects and convex distance functions. It turns out that algorithm DIG-VOR does not need many modifications to handle more general cases, too.

Theorem 2. The digitized Voronoi diagram of a set $S=\{w_1,...,w_n\}$ of n objects $w_i \subseteq <\Pi>$ for any convex distance function can be computed on a systolic screen of size M in time $O(\sqrt{M})$ provided that the following conditions hold:

(i) For any two objects $w,w' \in S$, $Im(w) \cap Im(w') = \emptyset$.

(i i) For any object $w \in S$ there exists an O(1) space description such the from this deecription it can be decided for every $p \in \Pi$ and $t \in \{1,...,\sqrt{M})\}$ in O(1) time whether $<p> \cap B(w,t) = \emptyset$.

(i i i) There exists a constant $\lambda \in O(1)$ such that for every $w \in S$, $t \in \{1,...,\sqrt{M}\}$, and $p \in \Pi$ with $<p> \cap B(w,t) \neq \emptyset$:

$\min \{ d_1(p,p') | \; p' \in \Pi, <p'> \cap B(w,t-1) \neq \emptyset \} < \lambda$,

where d_1 refers to the L_1-metric (processor distance).

Proof: Algorithm DIG-VOR needs only two minor modifications to handle the generalized case: B(s,r) needs to be generalized to the given type of objects and the given distance function, and the value of λ needs to be adjusted to the particular case. While Condition i ensures that, again, the images of two objects do not intersect, we need however two more conditions to show that algorithm DIG-VOR performs corretcly and terminates after $O(\sqrt{M})$ steps.

In Steps 2a and 2d of the algorithm, intersection tests between a ball B(s,r) and a rectangle <p> are performed. While such a test can clearly be executed in O(1) time for point objects and Euclidean metric, this may no longer be the case for arbitrary objects and distance functions. In fact, the processor performing this test does not only need the number of the oject but also the necessary information about the object to compute the intersection test. Therefore, an O(1) space decription of this information must be available, and the test must be executable in O(1) time; i.e., Condition ii must hold.

For Step 2a of algorithm DIG-VOR, the processor distance λ within that each P_{ij} has to scan all neighbors and send a message k to all $P_{i'j'}$ with V-Reg(i',j')=0 and $<(i',j')> \cap B(s_k,t) \neq \emptyset$ has to be modified according to the type of objects and the given metric. Condition iii ensures that λ is still O(1), which may not be the case in general.

However, with the above conditions, the correctness of this modified algorithm DIG-VOR follows in the same way as in the proof of Theorem 1, and its asymptotic running time does not change. Thus, Theorem 2 follows. ♦

The number of classes and object types for which the conditions in Theorem 2 apply is faily large. It contains all "simple" geometric objects that have an O(1) description and most of the standard distance function; in particular, all L_p metrices.

Corollary 3. *On a systolic screen of size M, the digitized Voronoi diagram of a set of points, line segments, circles, ellipses, and polygons of constant size can be computed, for any L_p-metric, in time $O(\sqrt{M})$ provided that their images do not intersect.*

5 Optical Clustering on a Systolic Screen

In [De86] we have presented a sequential technique for "optical clustering", i.e., identifying those groups of objects in a digitized picture that are "close" in the sense of human perception. Given a set of n line segments in the Euclidean plane and a separation parameter $r \in \mathbf{R}$, two line segments s and s' are called *r-connected* if and only if there exists a ball with radius less than or equal to r intersecting s and s'. The optical clustering with respect to separation parameter r is then defined as the partitioning of the set of lines segment into equivalence classes with respect to the relation *r-connected* . In [De86] it is shown that the transitive closure of the relation *r-connected* is equivalent to the transitive closure of the relation *Delaunay connected with respect to r* obtained as follows:

Compute the Voronoi diagram of the set of line segments. Two line segments s and s' are Delaunay connected if the Voronoi polygons of s and s' share a point that has distance of at most r from either line segment.

Therefore, Theorem 2 provides a way of computing, on a systolic screen, the optical clustering with respect to separation parameter r for any set of objects and distance function that have the properties listed in Theorem 2. We compute the digitized Voronoi diagram as described in Sections 3 and 4 with the following minor modification:

For every message originating at an object and travelling through the systolic screen, as described in algorithm DIG-VOR, its current distance from the object (i.e., minimum distance, with respect to the given distance funtion, of the current processor to the object) is constantly updated. When a message changes the register V-Reg(i,j) of a processor P_{ij}, the message's current distance from its object is also stored in an additional register D-Reg(i,j). Every processor P_{ij} with C-Reg(i,j)$\neq$0 sets D-Reg(i,j)=0.

Consider for a given separation parameter $r \in \mathbf{R}$ the image I(r) consisting of all those pixels (i,j) with D-Reg(i,j)$\leq$r, then the optical clustering of the object set with respect to separation parameter r corresponds exactly to the set of connected components of the image I(r) (see [NS80] for a definition of connected components of an image). In [NS80] it is shown that on a systolic screen of size M, the connected components of an image can be computed in time $O(\sqrt{M})$; hence, we obtain

Corollary 4. *For any set $S=\{w_1, \ldots, w_n\}$ of n objects $w_i \subseteq <\Pi>$ and any convex distance function that have the properties listed in Theorem 2, the optical clustering with respect to separation parameter r can be computed on a systolic screen of size M in time $O(\sqrt{M})$.*

REFERENCES

[AH86] M.J.Atallah, S.E.Hambrusch, "Solving tree problems on a mesh-connected processor array", Information and Control, Vol.69, Nos.1-3, 1986, pp.168-186.

[AK84] M.J.Atallah, S.R.Kosaraju, "Graph problems on a mesh-connected processor array", J. of the ACM, Vol.31:3, 1984, pp.649-667.

[De86] F. Dehne, "Optical clustering", The Visual Computer 2:1, 1986, pp. 39-43.

[DHSS87] F. Dehne, A. Hassenklover, J.-R. Sack, and N. Santoro, "Parallel visibility on a mesh-connected parallel computer", in Proc. International Conference on Parallel Processing and Applications, L'Aquila (Italy), 1987, North Holland 1988, pp. 203-210.

[DSS87] F. Dehne, J.-R. Sack and N. Santoro, "Computing on a systolic screen: hulls, contours and applications", in Proc. Conference on Parallel Architectures and Languages Europe, Eindhoven (The Netherland), 1987, Vol. 1, Lecture Notes in Computer Science 258, Springer Verlag, pp. 121-133.

[Ki82] C.E. Kim, "Digital disks", Report CS-82-104, Computer Science Dept., Washington State University, Dec. 1982.

[Mi84] P.L.Mills, "The systolic pixel: A visible surface algorithm for VLSI", Computer Graphics Forum 3, 1984, pp.47-60.

[Mo70] G.U. Montanari, "On limit properties of digitization schemes", J. ACM 17, 1970, pp 348-360.

[MS85] R.Miller and Q.F.Stout, " Geometric algorithms for digitised pictures on a mesh-connected computer", IEEE Trans. on PAMI 7:2, 1985, pp.216-228.

[NS80] D.Nassimi and S.Sahni, "Finding connected components and connected ones on a mesh-connected parallel computer", SIAM J. Computing 9:4, 1980, pp.744-757.

[Re84] A.P.Reeves, "Survey parallel computer architectures for image processing", Computer Vision, Graphics, and Image Processing 25, 1984, pp.68-88.

[Ro79] A. Rosenfeld, "Digital topology", Amer. Math. Monthly 86, 1979, pp 621-630.

[S88] O.Schwarzkopf, "Parallel computation of discrete Voronoi diagrams", Tech. Rep., Fachbereich Mathematik, Freie Universität Berlin (W.-Germany), 1988.

[SH75] M.I.Shamos, D.Hoey, "Closest Point Problems", Proc. 7th Ann. IEEE Symp. on Found. of Comp. Sci., 1975.

[SM84] Q.F.Stout, R.Miller, "Mesh-connected computer algorithms for determining geometric properties of figures", in Proc. 7th Int. Conf. on Pattern Recognition, Montreal, 1984, pp.475-477.

[Un58] S.H.Unger, "A computer oriented towards spatial problems", Proc. IRE 46, 1958, pp.1744-1750.