Technical Report

Dept. for Pattern Recognition and Image Processing
Institute for Automation
Technical University of Vienna
Treitlstr. 3/1832
A-1040 Vienna AUSTRIA
Phone: +43 (222) 58801-8161
Fax: +43 (222) 569697
E-mail: Etienne.Bertin@imag.fr

PRIP-TR-24 August 16, 1993

# Voronoi Pyramids controlled by Hopfield Neural Networks

*Etienne Bertin*[1], *Horst Bischof*

**Abstract**

We present an algorithm for image segmentation with irregular pyramids. Instead of starting with the original pixel grid, we first apply some adaptive Voronoi tesselation to the image. This provides the advantage that the number of cells in the bottom level of the pyramid is already reduced as compared to the number of pixels of the original image. Furthermore the Voronoi diagram is a powerful tool for shape description and image compression. For the construction of the irregular pyramid we present a Hopfield neural network which controls the decimation process. In this paper we extend our previous results by proving a more general theorem than in [4]. The contributions of this paper are the initialisation of the pyramid by a Delaunay graph and the extension of the results for Hopfield neural networks for decimation. The validity of our approach is demonstrated by several examples.

[1]E. Bertin is with TIMC-IMAG laboratory, P.O.Box 53, 38041 Grenoble Cedex 09, France

# 1 Introduction

Image segmentation is an important step in the analysis of digital images. Usually it is employed after image enhancement (e.g. noise removal) and before object recognition. The goal of segmentation is to completely partition the image into non overlapping regions, where each region satisfies some property (e.g. homogeneous grey value). Ideally these regions found by segmentation correspond to objects (or parts of objects) in the real world. A lot of different segmentation algorithms have been proposed (e.g. [33, 16, 28, 17, 24, 9]). The two main approaches are region based segmentation and contour based segmentation. In the first case we try to find regions, whereas in the second approach we try to find the borders of regions. The approach of this paper belongs to the class of region based methods.

Pyramidal techniques have been shown to be efficient methods for image segmentation. Among the most commonly used methods are Quadtrees [28] and pyramid linking [9]. However it has been shown [6] that segmentation algorithms based on regular pyramids are not shift invariant (i.e. the segmentation of an image and a slighly shifted version of the image may differ considerably). This was the main reason for the introduction of irregular pyramids [23]. Irregular pyramids operate on general graph structures instead of the regular neighborhoods as is the case of regular pyramids. For this reason they offer more freedom in the pyramid construction process, and they can adapt to the content of the image.

Within the framework of irregular pyramids we are not constrained to start form a regular image, this advantage we will use in this paper. We first apply an adaptive Voronoi tesselation to the image. And then we construct on this already irregular structure (we use the Delaunay triagulation which is the dual of the Voronoi tesselation) an irregular pyramid. The advantage of this approach is that the Delaunay graph offers a reduced description of the image, which is already adapted to the image content. This results in pyramids with less levels, as compared to starting with the original image. The Delaunay graph has the further advantage that it is planar (this is not the case for the 8 connected graph), and that the planarity is preserved by the pyramid construction (as was proved in [21]).

A second improvement we propose in this paper is in the pyramid construction process. We show that certain types of Hopfield neural networks [13] can be used to determine the survivors of the irregular pyramid. These networks offer the advantage that we can directly influence the decimation process, which is valuable when dealing with grey level image segmentation. We will prove a theorem which shows how these networks are related to maximum independent vertex set construction. We will also show how the parameters of the network have to be set in order to yield the desired segmentations.

The structure of this paper is as follows: In section 2 we briefly review pyramids, and describe irregular structures. In section 3 some definitions and algorithms for the Voronoi tesselation and the Delaunay graph are given. In particular we present a method for adaptive partitioning of grey level images into Voronoi polygons. Section 4 discusses the decimation by Hopfield neural networks. We present a theorem which shows the decimation by Hopfield networks is equivalent to maximum independent vertex set extraction. In section 5 we present some experimental results on image segmentation with our algorithm. Finally we discuss some further improvements of our method.

# 2 Pyramids

Image pyramids have shown to be efficient data and processing structures for digital images in a variety of vision applications. An image pyramid is a stack of images with exponentially decreasing resolutions [31]. The bottom level of the pyramid is the original image. In the simplest case each successive level of the pyramid is obtained from the previous level by a filtering operation followed by a sampling operator [11]. More general functions can be used to yield the desired reduction. We therefore call them *reduction functions.*

There are three important properties that characterize a pyramid:

1. Structure: e.g. neighbors, father–son relations between levels;
2. Contents of a cell: e.g. pixel, edge, or more;
3. Processing performed by the cells: e.g. filtering.

In this paper we discuss only the structure of pyramids. For a discussion on the contents of the cells and the processing performed by them see [4, 5].

## 2.1 Structure

The structure of a pyramid is determined by the neighbor relations within the levels of the pyramid and by the father–son relations between adjacent levels. The structure of any pyramid can be represented by horizontal and vertical graphs. Each level $i$ of a pyramid can be described by a neighborhood graph $G_i = \langle V_i, A_i \rangle$. Where the set of vertices $V_i$ corresponds to the pixels of level $i$, and $A_i \subseteq V_i \times V_i$ are the neighborhood relations of the pixels. Two vertices $p, q \in V_i$ are connected in $G_i$ if they are neighbors in the structure.

**Definition 1 (Neighborhood)** *The neighborhood of vertex* $p \in V_i$ *is defined by* $\Gamma(p) := \{p\} \cup \{q \in V_i | (p,q) \in A_i\}$.

The structure is **regular** if a well defined neighborhood relation holds for all vertices (except for the boundary).

The vertical structure (i.e. the connectivity between the levels) can also be described by a (bipartite) graph: $R_i = \langle (V_i \cup V_{i+1}), L_i \subseteq (V_i \times V_{i+1}) \rangle$. The receptive field (i.e. the set of all sons) of a cell $q \in V_{i+1}$ is defined as: $RF(q) := \{p | \langle p, q \rangle \in L_i\}$.

Any pyramid with $n$ levels can be described by $n$ neighborhood graphs and $n-1$ vertical graphs. We distinguish between

- regular structures and
- irregular structures

depending on whether the structural relations are the same for all pyramid cells (except on the boundary) or whether they may vary from cell to cell. In this report we will focus on irregular pyramids.

## 2.2 Irregular pyramids

In irregular pyramids the regularity constraint of regular pyramids is relaxed. These pyramids operate on a general graph structure instead of the regular neighborhood graph as in the case of regular pyramids. There are two ways to construct an irregular pyramid:

1. Parallel graph contraction [27]
2. Decimation of the neighborhood graph [23]

The main purpose for the introduction of irregular pyramids was the rigid behavior (e.g. shift variance) of regular structures [6]. Irregular pyramids offer greater flexiblility [24] for the price of less efficient access. Since parallel graph contraction is only possible for certain kinds of graphs [27] we discuss the decimation of the neighborhood graph.

Decimation divides the cells in a pyramid level into two categories: cells that survive form the cells of the next level and cells that do not appear at reduced levels (non survivor). Peter Meer [23] has given two rules which should be fullfilled by the decimation process. His rules are:

**Definition 2 (Decimation rules)**

1. *Two neighbors at level i cannot survive both;*
2. *a non survivor must be a neighbor of a survivor.*

We call a decimation which satisfies these rules a **valid decimation**. In [23] it was shown how a decimation can be computed in parallel by a stochastic algorithm. It is also worth noting that the rules 1 and 2 are equivalent to saying that the vertices $V_{i+1}$ of the Graph $G_{i+1}$ on level $i+1$ define a maximum independet (vertex) set (MIS) of the graph $G_i = \langle V_i, A_i \rangle$.

The algorithm for stochastic decimation proceeds in following major steps (for more details see [21]):

**Algorithm Stochastic Decimation:**

1. Assign uniformly distributed random numbers to the cells.
2. Select local maxima as surviving cells.

3. Fill holes, i.e. repeat step 2 as long as there are non surviving cells which have no surviving neighbor.

4. Every non surviving cell selects a father. This construction also defines the receptive fields.

5. Construct the neighborhood graph of the new level; Two surviving vertices become neighbors if they have vertices in their receptive fields which are neighbors in the level below.

6. Repeat steps 1–6 with new level until only a single vertex is in the receptive field.

This basic algorithm can be modified in order to take into account the contents of a cell ending up with an adaptive pyramid [18]. This has been used for image segmentation or connected component analysis in logarithmic time complexity.

# 3 Voronoi Diagram and Delaunay Graph

In this section we give some definitions of the Voronoi tesselation, and the Delaunay graph. Two main algorithms to compute the Voronoi diagram and its dual are presented. We give also some theoretical results about the complexity of these algorithms. Finally, a description of a split and merge algorithm to compress images in a Voronoi environment is given.

## 3.1 Voronoi Diagram and Delaunay Graph

**Definition 3** *Let $S$ be a finite set of points in $\mathbb{R}^2$. Let $p$ be a point of $S$. The Voronoi region $VOR(p)$ associeted to $p$ is the set of points nearer to $p$ than to any of the other points of $S$: $VOR(p) = \{x \in \mathbb{R}^2, d(x,p) \leq d(x,q), \forall q \in S - p\}$ ($d$ is the Euclidean distance).*

From this definition it is easy to show the following property.

**Property 1** *All Voronoi regions are polygonal and convex.*

**Definition 4** *The Voronoi diagram (or Voronoi tesselation) of $S$, is the set of the Voronoi polygons: $VOR(S) = \bigcup_{p \in S} VOR(p)$.*

We can note from the definition 4 that the Voronoi diagram is a partition of the plane. Let us now introduce the Delaunay graph:

**Definition 5** *The dual of the Voronoi diagram is the Delaunay graph. Two points of $S$, $p_i$, and $p_j$, create an edge of the Delaunay graph if and only if $VOR(p_i)$ and $VOR(p_j)$ are adjacent in the Voronoi diagram: $DEL(S) = \langle S, E = \{(p_i, p_j) \in S^2, VOR(p_i) \cap VOR(p_j) \neq \emptyset\} \rangle$.*

**Property 2** *The Delaunay diagram of a set $S$ is the unique triangulation in which the circle circumscribed by every triangle $(p_i, p_j, p_k) \in S^3$ does not contain in its interior any point of $S$: $Circle(p_i, p_j, p_k) \cap S - p_i - p_j - p_k) = \emptyset$ [25] (see Fig. 1).*

According the Def. 3 and 4, the Voronoi tesselation appears to be a powerful tool to represent the morphology and the shape of objects. For example, the cells of a tissue look like Voronoi polygons [12]. We observe the same phenomenon with soap bubbles, the alveous bee, and the quasi crystal arrangement [30]. According the Prop. 2 the Delaunay graph contains all the information about the neighborhood. Thus, the Delaunay graph is well adapted for modeling in many fields such as biology [22, 10], physics, etc. Furthermore, the duality between the Delaunay and the Voronoi graphs is fundamental for our algorithms. This is the reason, we use the Voronoi polygonisation and the Delaunay graphs for the pyramidal approach.

## 3.2 Algorithms and Complexity

Many algorithms exist to compute the Voronoi tesselation and the Delaunay graph. The famous ones are the "divide and conquer" algorithm and the "incremental" algorithm. The first one is due to Preparata and Shamos [25]. The main idea is to divide the problem into subproblems recursively and then, to merge the results obtained. The main advantage of this algorithm is its $O(n \log n)$ optimal complexity. But due to problems of computational geometry this algorithm is difficult to implement. The second one is due to Green and Sibson [8]. The main idea is to compute the Voronoi diagram interactively. The incremental algorithm works by local modification of the Voronoi diagram after insertion of a new point in the structure. We start with only one point in the structure and then we add one point after the other. Since the modifications are local, the worst case complexity is $O(n^2)$. However, in most cases the incremental method is optimal [1].

With an incremental method, we obtain 10000 polygons with about 20000 triangles in 5 sec and 300000 polygons with 600000 triangles in 2 min on a Silicon Graphics Indigo

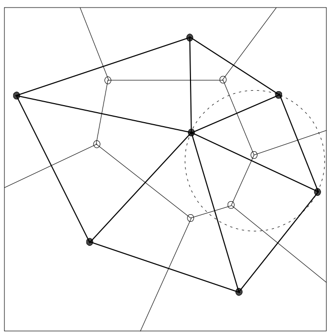


Figure 1: Voronoi polygons and Delaunay triangulation. The circle does not contain in its interior any point of S (Black dots)

.

workstation. Thus, the incremental method has two advantages: the run time optimality in many cases and the dynamic management of the Voronoi structure. These two points are crucial for the performance and the conception of the segmentation and compression applications discussed bellow.

### 3.3 Application in Image Processing

We present an algorithm for convex partitioning of a grey level image. Our algorithm is a generalization of the split and merge algorithm on quatree regular (rigid) structures, extended to Voronoi irregular (non rigid) structures.

This algorithm proceeds in two steps. The first one is the split step: Polygons are added in the support of the image until convergence. This step involves a dynamic management of the Voronoi diagrams. The second is the merge step: Some polygons are deleted. A polygon is added or deleted according to the following definitions:

**Definition 6** *A region enclosed by a polygon $P$ is said to be homogeneous if and only if the variance in the region is less than a given threshold.*

**Definition 7** *A polygon $P$ is said to be useless if and only if all the neighbors of $P$ have almost equal grey level means.*

If a polygon is non homogeneous (Def.6), we add a point in its interior, else we do nothing. According the Def.7 a useless polygon is deleted in the structure. More precisely, the algorithm proceeds in following major steps (for more details see [7, 2]):

1. Assign five points in the image (one point on the center of the image and four on the corners of the image).

2. Compute the Voronoi diagram and the Delaunay graph.

3. Compute mean grey value, standard deviation, and surface of each polygon.

4. For all polygons, split if the polygon is not homogeneous.

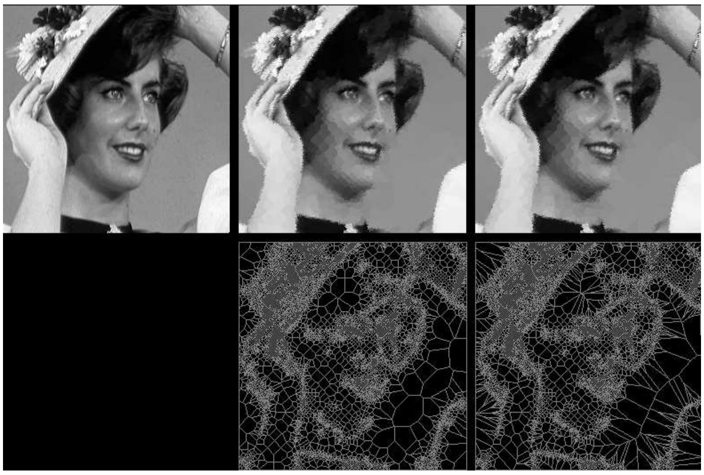
5. Repeat 2–4 until convergence (all the polygons are homogeneous).

6. Merge: supression of the useless polygons.

The proof of convergence is straightforward (in the worst case all the polygons have one pixel in their interior which give a variance equal to zero. We can prove that the complexity of the split and merge algorithm is $O(n)$, in average, where $n$ is the final number of points. This result is due to the fact that we have some information we can use to compute the Voronoi diagram [7, 2]). The information is that we know approximately in which polygon a new point is added.

An illustration of this algorithm is given in figure 2, where the size of the image is $256 \times 256$ with 256 grey values.

(a) Split Step: From left to right and top to bottom: Original image, 5 polygons, 13 polygons, 41 polygons, 126 polygons, 410 polygons, 1316 polygons, 3906 polygons, 9977 polygons, 10350 polygons.



(b) Result: Original image, Split result (10585 polygons) and the graph, Merge result (9033 polygons) and the graph.

Figure 2: Split and Merge algorithm based on Voronoi Diagram.

# 4 Hopfield Neural Network

Neural networks are massive parallel processing structures, which consists of a set of simple yet highly interconnected processing elements called units. Each connection between two units has a weight attached to it. Units perform only simple local computations (e.g. weighted sum). Usually each neural network model has a specific learning algorithm (i.e. an algorithm which specifies how to change the weights according to external stimuli). There exist a large variety of different neural network models, a popular one is the Hopfied model.

In 1982 Hopfield has introduced a network model [13] based on fully connected binary units (i.e. a unit has two states 0 or 1, and every unit is connected to every other unit (except to itself) in the network). The discrete binary states of the units constitute a pattern that can be considered to be the state vector of the system. Hopfield proved that this system will always converge to a stable state, if the weights of the connections are symmetric (i.e. $w_{ij} = w_{ji}$ where $w_{ij}$ is the weight of the connection between unit $i$ and unit $j$) and the units are updated in an asynchronous manner. Hopfield has shown that this system is governed by an energy function $E$ which is decreased when the units are updated. Since $E$ is bounded the system must converge to a stable state.

Hopfield has also given a learning algorithm which can store a set of patterns in the network. In this case the Hopfield network acts as an associative memory. Unfortunately the storage capacity of the network is rather limited. In [26] it was proved that for a network with $n$ binary units, the number of stored patterns that can be recovered exactly is bounded asymptotically by $\frac{n}{4 \log n}$ as $n$ approaches infinity.

In 1984 Hopfield has introduced a continous model [14] (i.e. the state of a unit can change continously). This model has the same characteristics as the binary model, though it is better suited for analog VLSI implementation.

Besides using a Hopfield network as an associative memory there are several attempts to use the network for optimization problems like the Traveling Salesman Problem [15, 20] or clustering [19]. The basic idea is to formulate the optimization problem as an energy function which can be minimized by a Hopfield network. Since the update algorithm of a Hopfield network is a stochastic gradient descent search, finding a global minimum cannot be guranteed. However very good solutions have been reported for the continous Hopfield model [15, 20].

The approach of this paper is similar to using the Hopfield model as an optimization procedure, however we do not need fully connected networks and we are not interested in the global minimum of the energy function. As it is shown in Theorem 1 it is sufficient to find a local minimum.

## 4.1 Decimation by Hopfield networks

In the following we will show that we can replace the steps 1–3 of the algorithm for stochastic decimation by a modified Hopfield network which works on the neighborhood graph $G = \langle V, A \rangle$[1]. Moreover we show that the formulation as a Hopfield network is more general than the stochastic decimation, and it naturally includes the concept of the adaptive pyramid.

Let us introduce the notion of a survival state of a cell:

**Definition 8** *The survival state of a cell $p \in V$ is a function*
$s : V \mapsto \{0,1\}$ *with* $s(p) = \begin{cases} 1 & \textit{if cell } p \textit{ survives} \\ 0 & \textit{otherwise} \end{cases}$

Let us further introduce the following energy function:

$$E = -\frac{1}{2} \sum_{\langle i,j \rangle \in A} w_{ij} s(i) s(j) - \sum_{k \in V} I_k s(k) \tag{1}$$

Now we can proof the following theorem which is a generalization of previous results in [4, 3]:

**Theorem 1** *The energy function $E$ from eq. (1) obtains a local minimum, $E_{min}$, with $I_k > 0 \; \forall k \in V, w_{ij} = w_{ji} < 0$ and $I_k < |w_{ij}| \; \forall k \in V, \langle i,j \rangle \in A$ if and only if the assignment of surviving and non–surviving cells, $s(p)$, is a valid decimation (i.e. satisfying the rules 1 and 2 of definition 2) or, equivalently, forms a maximum independent vertex set of $G$.*

*Proof.*
(a) $E_{min}$ is a local minimum of $E \Rightarrow s(p)$ is a valid Decimation.
Assume $E = E_{min}$ is a local minimum but $\{p \in V | s(p) = 1\}$ is not a valid decimation, then at least one of the rules 1 or 2 must be violated.

Case a.1: rule 1 does not hold:

$\Rightarrow \exists \; p, q \in V$ such that$\langle p, q \rangle \in A$ and (by Def. 8) $s(p) = s(q) = 1$. Changing $s(p)$ to 0 can affect only those terms in equation (1) where $s(p)$ occurs. We can write $E$ as

$$E = -\frac{1}{2} \sum_{\langle p,n \rangle \in A} w_{pn} s(p) s(n) - \frac{1}{2} \sum_{\langle n,p \rangle \in A} w_{np} s(n) s(p) - \frac{1}{2} \sum_{\substack{\langle i,j \rangle \in A \\ i,j \neq p}} w_{ij} s(i) s(j) - I_p s(p) - \sum_{k \neq p \in V} I_k s(k)$$

because E is symmetric. If we change $s(p)$ from 1 to 0 we get following energy difference $\Delta E$: $\Delta E \leq I_p + w_{pn}$ Since $I_p > 0$ and $|w_{pn}| > I_p$ with $w_{pn} < 0$ $\Delta E < 0$ and this is a contradiction that $E_{min}$ is a local minimum.

Case a.2: rule 2 does not hold

$\Rightarrow \exists p \in V$ such that $s(p) = 0$ and $\forall q \in \Gamma(p) : \; s(q) = 0$. Again changing $s(p)$ to one yields $\Delta E = -I_p$ and this is a contradiction to the fact that $E_{min}$ is a local minimum.
(b) $\{p | s(p) = 1\}$ defines valid Decimation $\Rightarrow E = E_{min}$ is a local minimum

We have to show that by changing only one state of a cell $r \in V$ we get a higher energy value.

b.1: $s(r) = 0 \Rightarrow \exists q \in \Gamma(r) : \; s(q) = 1$ changing $s(r) = 1 \Rightarrow \Delta E = -I_r - w_{rq} > 0$

[1] We skip the subindices for level i because the algorithm works only on one level

b.2: $s(r) = 1 \Rightarrow \forall q \in \Gamma(p) : \ s(q) = 0$ changing $s(r) = 0 \Rightarrow \Delta E = I_r > 0$

From b.1 and b.2 we conclude that $E_{min}$ is a local minimum of $E$. qed.

Given the energy function in eq.(1) we can now define a Hopfield network operating on the neighborhood graph which minimizes this energy function. In [13, 14] Hopfield has described a network of fully connected units operating asynchroniously which is governed by the following energy function:

$$E = -\frac{1}{2}\sum_i \sum_j w_{ij} s(i) s(j) - \sum_i I_i s(i) + \sum_i U_i s(i) \tag{2}$$

where $w_{ij} \in \mathbb{R}$ is the weight between unit $i$ and $j$, $I_i$ is the external Input and $U_i$ is the threshold of unit $i$.

Hopfield proved that if the weights are symmetric (i.e. $w_{ij} = w_{ji}$) and the units update asynchroniously, the network will settle in a local minimum of $E$. If we now set in equation 2 $U_i = 0$ we get equation 1.

The resulting Hopfield network operates on the neighborhood graph $G$ and computes valid decimations (according to definition 2). The update procedure of the cells is as follows:

$$s(p) = \begin{cases} 1 & \text{if } I_p + \sum\limits_{\langle q,p\rangle \in A} w_{qp} s(q) > 0 \\ 0 & \text{otherwise} \end{cases}$$

The initial state of the network can be choosen at random. Having established this relationship we can now apply all the theory available for Hopfield networks. For example the convergence time is of interest because in the algorithm of Meer [23] the convergence takes $O(|V|)$ steps in the worst case (when the random number generator produces a ramp function on every iteration). In [29] it was proven that a Hopfield network with negative weights (if weights are 0 the connection is not present) and a sequential update algorithm (i.e. at a time only one unit is updated) converges in the worst case in $2|V|$ steps.

But one should note that by a parallel update scheme and a connectivity graph which is far away from full connectivity one could design a parallel algorithm which converges much faster, because all units which are not connected can be updated in parallel without altering the convergence properties.

Figure 3 is an illustration of the decimation of a Delauney graph by a Hopfield network. One can note that in this example, the triangulation at level 4 is not planar. This is a problem for an implementation of the decimation algorithm in the dual pyramid [32].

## 4.2 Adaptive Decimation

The energy function in eq. (1) has the weights $w_{ij}$ and the external inputs $I_i$ as free parameters. These parameters can be used to influence the decimation process. The weights $w_{ij}$ between the cells express the constraints on the states of the cells. If $w_{ij}$ is negative these two cells should not be both on, on the other hand a positive weight increases the likelihood that both cells are on. One should note as long as the weights are

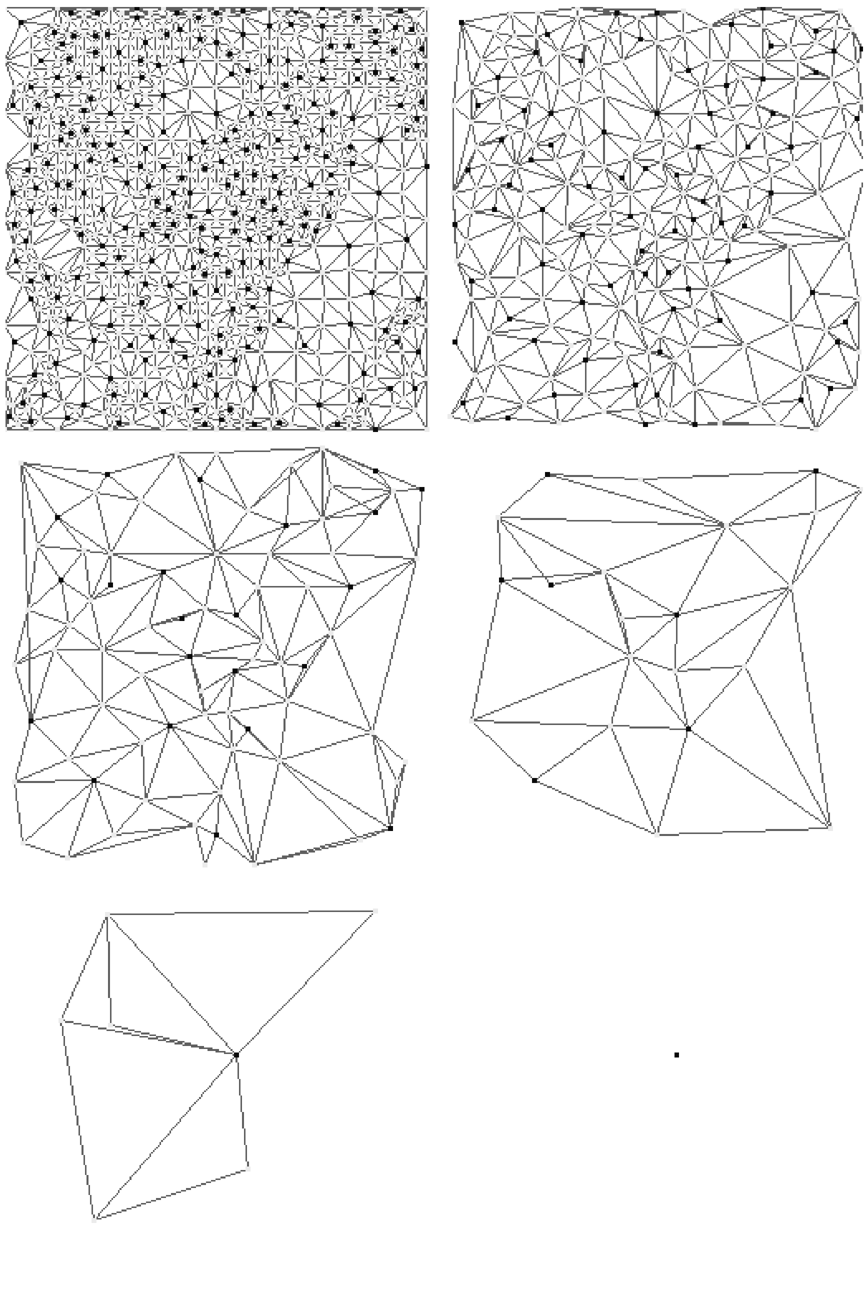


Figure 3: Decimation of a Delaunay Graph, Black dots indicate survivors

identical and $|w_{ij}| > I_k$ the behavior of the network is not changed. This can be easily seen from the proof of thereom 1. The external input $I_k$ can force a single cell to survive or not (e.g. if $I_i > \sum_{j \in \Gamma(i)} |w_{ij}|$ the cell $i$ survives).

In [4] we have shown how to set the weights in order to perform connected component analysis. There we have also listed several possibilities how to set the weights for image segmentation. In section 5 we will demonstrate how the set the weights properly in order to achieve good segmentation results.

# 5 Application to Segmentation

We present here some results obtained with the combinations of the pyramids, Voronoi diagram and Hopfield neural nework. We choose here the following weigths for the edges of the neighborood graph: $w_{ij} = -f(d_{ij})$, where $d_{ij} = |g_i - g_j|$, $g_i$ and $g_j$ are the greyvalues of node $i$ and $j$, and $f$ is a suitable function. We choose here the following function:

$$f(d_{ij}) = \begin{cases} 2 & \text{if } i \text{ and } j \text{ are neighbors and } d_{ij} \leq \Theta \\ 0 & \text{otherwise} \end{cases}$$

Certainly, other types of functions can be chosen to yield better adaptive decimations. A slight generalization of the function above would be:

$$f(d_{ij}) = \begin{cases} 2 & \text{if } i \text{ and } j \text{ are neighbors and } d_{ij} \leq \Theta_1 \\ \frac{-2(\Theta_2 + d_{ij})}{\Theta_1 - \Theta_2} & \text{if } i \text{ and } j \text{ are neighbors and } \Theta_1 < d_{ij} < \Theta_2 \\ 0 & \text{otherwise} \end{cases}$$

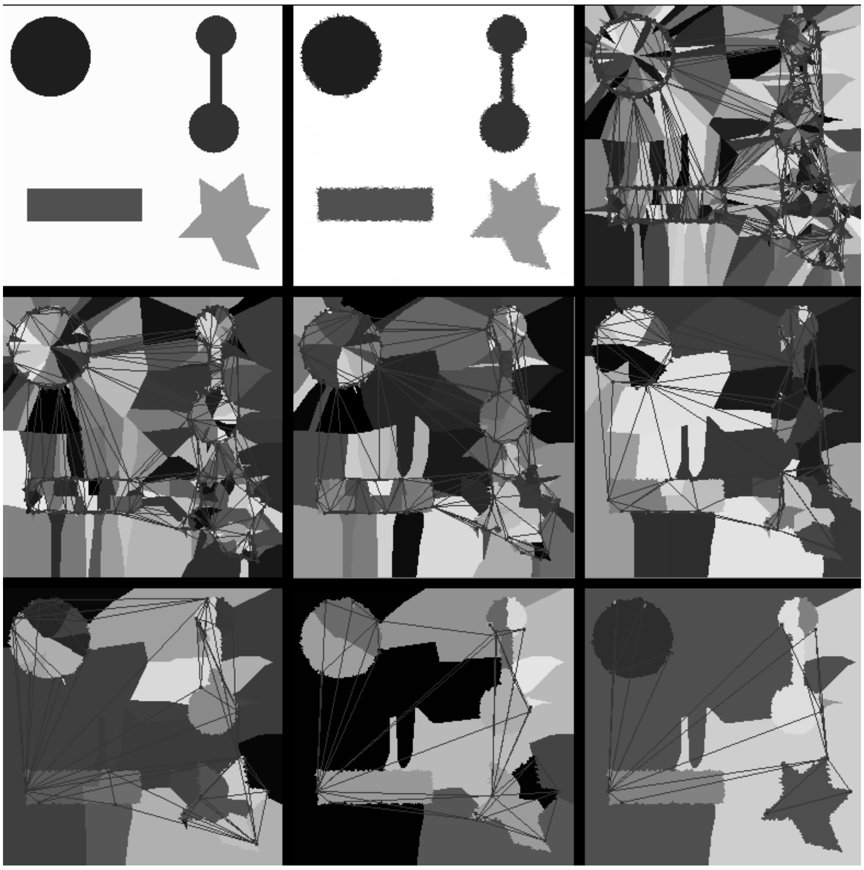
The algorithm proceeds in following major steps.

1. Compute an adaptive Voronoi partition of the image with the split and merge algorithm.

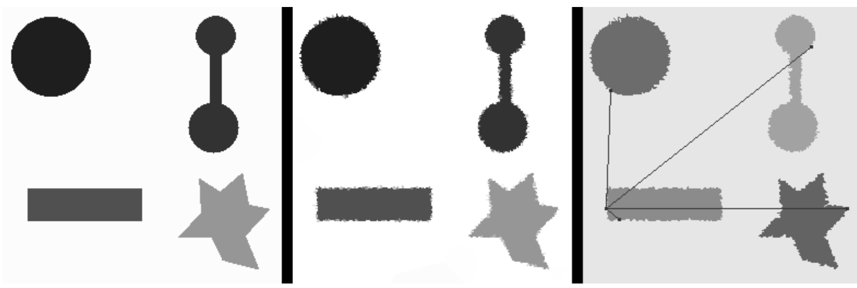
2. Compute the pyramidal process with the adaptive decimation.

Figure 4 is an illustration of the pyramidal process in a Voronoi environment controled by Hopfield neural network ($\Theta = 7$) ($256 \times 256$).

Figure 5 gives the result for a grey level image segmentation ($\Theta = 9$)($256 \times 256$). The total time to process such an image is arround 2 minutes on a Sun 4 workstation. Some improvements can be made: for example local tresholding [24]. We can also take into account the size (surface) of the polygons. Due to our split and merge algorithm, there are many polygons with a small area (1-10 pixels) where there is a large gradient in the image. Conversly there are few polygons with large size (10-50 pixels) where the image is uniform.



(a) From left to right and top to bottom: Original image, Voronoi tesselation: 1500 polygons, Level 1: 521 regions, Level 2: 225 regions, Level 3: 110 regions, Level 4: 57 regions, Level 5: 33 regions, Level 6: 20 regions, Level 7: 14 regions.



(b) Original image, Voronoi tesselation, Final Result (Level 11): 5 regions remain.

Figure 4: Adaptive Pyramidal Process Based on Voronoi Diagram Controled by Hopfield Neural Network.

# 6 Conclusions and Outlook

In this paper we have presented a pyramidal algorithm using Voronoi polygons, Delaunay triangulation, and Hopfield neural networks. The combination of all these notions give an algorithm for the segmentation of the grey level images. We have demonstrated that the algorithms presented provide a valuable tool for image segmentation. The initialisation of the pyramid by the Delaunay graph reduces the computations to build the pyramid, because the number of cells in the base level is already reduced. The control of the decimation process by the Hopfield network allows us to adaptively influence the pyramid construction process in a natural manner. The formulation as an energy minimization procedure might also provide further theoretical insights in the construction process of irregular pyramids.

An interesting point is to compare results obtained here with others methods.

The work presented here might be extended in several ways. For example we can use the algorithms for textural segmentation were we use the Kolgomorov-Smirnov distance as distance measure for initialisation of the weights in a Hopfield network. Such algorithm may be an alternative to the Markov Random Field process. In this case we could control the different steps of the algorithm which is not the case for Markov Random fields. Another extension where our algorithm would prove its advantages is 3D image segmentation (e.g. data from the confocal laser microscope). Since the discrete volume representation is a large mass of data, therefore it is essentially to reduce the amount of data in early stage.

Further improvements can be made by an in depth analysis of the assignment of weights in the Hopfield network. It remains still to be shown which type of assignment presents

Figure 5: Image "femme", final result of the segmentation. Starting with 9033 Voronoi polygons 50 regions remain.

the best segmentation results. With the proof of the Theorem we have made one step in that direction.

Another important extension is the concept of the dual graph. In the dual pyramid we can control the number of neighbors (essentially it can be shown that it will not increase during the decimation process [21]), which is not the case for the original pyramid, where the number of neighbors of a node may grow from level to level. Therefore a formulation of our algorithm for the dual graph would be advantageous. This is possible because the Delaunay triangulation is a planar graph.

All these possible extensions would further improve the results and show the advantages of the Voronoi irregular pyramidal framework controlled by Hopfield neural networks.

## References


[1] Etienne Bertin and Jean-Marc Chassery. Diagramme de Voronoï généralisé 3D: premiers résultats. Technical Report 879, TIM3-IMAG, University Joseph Fourier, Grenoble, 1992.

[2] Etienne Bertin, Franck Parazza, and Jean-Marc Chassery. Segmentation and measurement based on Voronoi diagram: Application to confocal microscopy. *Computerized Medical Imaging and Graphics*, 17(3):001–008, 1993.

[3] Horst Bischof and Walter G. Kropatsch. Hopfield networks for irregular decimation. accepted at 17th ÖAGM, Graz, 1993.

[4] Horst Bischof and Walter G. Kropatsch. Neural Networks versus Image Pyramids. Technical Report PRIP-TR-7, Dept. for Pattern Recognition and Image Processing, TU Wien, 1993.

[5] Horst Bischof and Walter G. Kropatsch. Neural networks versus image pyramids. In R.F. Albrecht, C.R. Reeves, and N.C. Steele, editors, *Artifical Neural Networks and Genetic Algorithms*, pages 137–144. Springer Verlag, 1993.

[6] M. Bister, J. Cornelis, and Azriel Rosenfeld. A critical view of pyramid segmentation algorithms. *Pattern Recognition Letters*, Vol. 11(No. 9):pp. 605–617, September 1990.

[7] Jean-Marc Chassery and Mahmoud Melkemi. Diagramme de Voronoï appliqué la segmentation d'images et à la détection d'évènements en imagerie multi-source. *Traitement du Signal*, 8:155–164, 1991.

[8] P. J. Green and R. Sibson. Computing Dirichlet tessellation in the plane. *Computer Journal*, 21:168–173, 1978.

[9] W.I Grosky and R. Jain. A pyramid based approach to segmentation applied to region matching. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 8(5):639 –650, 1986.

[10] Martial Guillaud, Raphael Marcelpoil, Olivier Baujard, and Sylvie Pesty. Modèle de simulation de l'émergence et de la croissance des tumeurs cancéreuses dans un tissu sain. In *Proc. of The Journées Francophones IAD and SMA, Toulouse*, pages 414–431, April 1993.

[11] Robert M. Haralick and Shapiro Linda G. Glossary of computer vision terms. *Pattern Recognition*, 24:pp.69–93, 1991.

[12] H. Honda. Description of cellular patterns by Dirichlet domains: the two dimensionnal case. *J. Theor. Biol.*, 75:523–543, 1978.

[13] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *PNAS*, 79:2554–2558, 1982.

[14] J.J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *PNAS*, 82:3088–3092, 1984.

[15] J.J. Hopfield and D.W. Tank. Neural computation of decisions in optimization problems. *Biological Cybernetics*, 52:141–152, 1985.

[16] S.L. Horowitz and Pavlidis T. Picture segmentation by directed split-and-merge procedure. In *Proc. 2. ICPR*, pages 424 – 433, 1974.

[17] Tilton J.C. Image segmentation by iterative parallel region growing with applications to data compression and image analysis. In *Frontiers of Massively Parallel Computation*, pages 357 – 361, 1988.

[18] J.M. Jolion and Montanvert A. The adaptive pyramid, a framework for 2D image analysis. *Computer Vision, Graphics, and Image Processing: Image Processing*, Vol. 55(No. 3):pp.339–348, May 1992.

[19] B. Kamgar-Parsi, J.A. Gualtieri, J.E. Devaney, and B. Kamgar-Parsi. Clustering with neural networks. *Biological Cybernetics*, 63:201–208, 1990.

[20] B. Kamgar-Parsi and B. Kamgar-Parsi. Hopfield model and optimization problems. In Wechsler H., editor, *Neural Networks for Perception*, volume 2, pages 94 – 110. Academic Press, Inc., 1992.

[21] Walter G. Kropatsch and Annick Montanvert. Irregular pyramids. Technical Report PRIP-TR-5, Dept. f. Pattern Recognition and Image processing, TU Wien, 1992.

[22] Raphael Marcelpoil and Yves Usson. Methods for the study of cellular sociology: Voronoi diagrams and parametrization of the spatial relationships. *J. Theor. Biol.*, 154:359–369, 1992.

[23] Peter Meer. Stochastic image pyramids. *Computer Vision, Graphics, and Image Processing*, Vol. 45(No. 3):pp.269–294, March 1989.

[24] A. Montanvert and Bertolino P. Irregular pyramids for parallel image segmentation. In H. Bischof and W.G. Kropatsch, editors, *Pattern Recognition 1992*, volume OCG-Schriftenreihe 62, pages 13–34. Oldenbourg, 1992.

[25] J.P. Preparata and M.I.S. Shamos. *Computational Geometry, an Introduction*. Springer Verlag, NewYork, 1988.

[26] McEliece R.J., Posner E.C., and Rodemich E.R. The capacity of the hopfield associative memory. *IEEE Trans. on Information Theory*, 33:461–482, 1987.

[27] Azriel Rosenfeld. Arc colorings, partial path groups, and parallel graph contractions. Technical Report TR-1524, University of Maryland, Computer Science Center, July 1985.

[28] H. Samet. *The design and analysis of spatial data structures*. Addison Wesley, 1990.

[29] Y. Shrivastava, S. Dasgupta, and S.M. Reddy. Guaranteed convergence in a class of hopfield networks. *IEEE Trans. on Neural Networks*, 3(6):951–961, 1992.

[30] D.F. Stevens. *Patterns in Nature*. Little Brown, Boston, 1978.

[31] Steven L. Tanimoto. Paradigms for pyramid machine algorithms. In S. Levialdi and V. Cantoni, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 173–194. Springer-Verlag Berlin, Heidelberg, 1986.

[32] Dieter Willersinn and Walter G. Kropatsch. Preserving topology in the irregular curve pyramid. In Poeppl, editor, *Mustererkennung 1993*, Lübeck, September 1993. Springer.

[33] S.W. Zucker. Region growing: childhood and adolescence. *Computer Graphics and Image Processing*, 5:382 – 399, 1976.