Finding Connected Components with Dual Irregular Pyramids

Herwig Macho*, Walter G. Kropatsch †
TU Vienna, Institute for Automation 183/2
Department for Pattern Recognition und Image Processing
Treitlstrasse 3, A-1040 Vienna, AUSTRIA
Tel.: +43 (1) 58801-8161, Fax: +43 (1) 586 96 97
email: mah@prip.tuwien.ac.at, krw@prip.tuwien.ac.at

Abstract: *Irregular pyramids are sequences of graphs with decreasing numbers of vertices, edges and faces from level to level. The advantage of describing images with graphs helps to overcome problems which can occur in regular pyramid structures, e.g. shift-variance and rotation-variance. Here we present some ideas and results how a recently developed parallel algorithm to reduce the number of elements (vertices, edges and faces) in a graph can be used for finding connected components in an image. The advantage of using this method to build irregular pyramids is that the degree of vertices in one of two parallel constructed graphs stays bounded, which cannot be provided with other techniques. To achieve this, the conventional region adjacency graph is extended, i.e. additional edges between and around vertices are allowed.*

1 Introduction

In image processing, pyramids are used for a variety of applications. They provide a multiresolution representation of the image, which can be calculated in parallel and in a logarithmic (*O(log(imagesize))*) number of steps.

Regular pyramids are a stack of images, they are often characterized by three parameters: the window(e.g. 2×2 pixels), the factor (e.g. 4) and the function (e.g. averaging) of reduction. Each cell in the pyramid has a set of children, a set of neighbours and one or sometimes, even more parents.

Irregular pyramids are a stack of graphs which describe an image. They can be seen as a generalization of the principles of the regular pyramid, bringing more flexibility in the choice of the neighbourhood relations. Instead of the rigid structure of children in regular pyramids, the **receptive field** represents the parent–child connections in a more flexible way:

*A part of this work was done at the laboratory LIP at the ENS Lyon/France

†This work was supported by the Austrian Science Foundation under grant S 7002-MAT.

at each level of the pyramid surviving elements (vertices) cover a variable set of non-surviving elements in their neighbourhood.

As the construction of the regular pyramid, building an irregular pyramid is a parallel process: each vertex can decide independently if it survives to the next level or not. The assignment of a non-surviving vertex to a receptive field and the construction of the next level can be done in parallel.

Some existing methods to select surviving vertices are, e.g. the stochastic decimation, where the selection process is determined by the outcome of random variables [7]. The selection of surviving vertices with a Hopfield network can be found in [1]. Adaptation of this process to image data was done in [8], improved results with region and contour cooperation in irregular pyramids were achieved in [2]. In the adaptive pyramid, the selection of surviving vertices is based on a variable which represents a value of interest. This is used to obtain better adaptation to image content [3]. Texture segmentation was achieved by storing additional texture information about the region one vertex describes at higher levels. Criteria were e.g. area, perimeter, roundness, orientation and major to minor axis ratio [6].

In the next section we give an overview about a new algorithm to build hierarchies of graphs which was developed by Kropatsch and Willersinn [10, 11, 4]. We compare it with "standard versions" of irregular pyramids, particularly we focus on the different graph structures obtained by this method. A small example, the process to find connected components of an artificially generated image of a house, helps to understand our ideas. An outlook to future work concludes this paper.

2 Basic graph operations

Before we start with an overview about this process we recall some basic vocabulary concerning graphs.

A *graph* G(V,E) consists of a set of vertices and edges. Edges which connect one and the same vertex are called *self-loops*, if more than one edge connect two specific vertices we call these edges *double* or *parallel edges*. The *degree* of a vertex is the number of edges incident on it.

For a planar graph (a graph in which edges can be drawn in the plane without crossings) the *dual graph* exists. The mechanism to construct it is very simple: each vertex of the dual graph describes a face (or area) in the original (neighbourhood) graph, the edges connecting the vertices in the dual graph correspond 1:1 to the edges in the neighbourhood graph. Both graphs are dual to each other, to distinguish between them we refer to the graph which carries the vertices describing the images as the "original" or "neighbourhood" graph, the second graph will be called the "dual" or "face graph".

There are two basic operations for graphs concerning elimination of edges and vertices:

- **Contracting** an edge: An edge is removed from a graph and the end vertices are joint together.

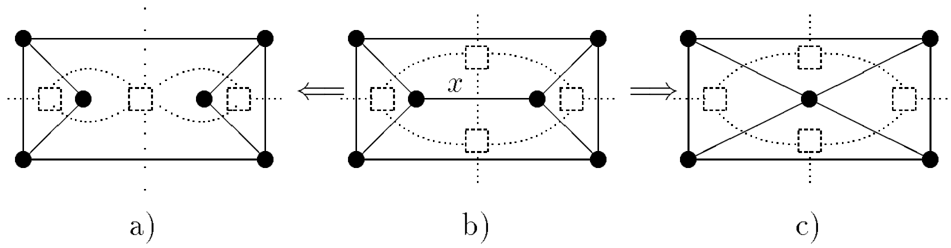

Figure 1: Basic operations to remove edges and/or vertices from a graph.

- **Removing** an edge: An edge is just removed from a graph.

The duality of these basic operations can be seen in Figure 1. Figure 1b) shows a graph (vertices drawn as solid circles, edges drawn as solid lines) and its dual (dual vertices are drawn as dashed boxes, dual edges are drawn as dotted curves). In Figure 1a) edge x has been removed from the neighbourhood graph, the corresponding dual edge has been contracted in the face graph. Figure 1c) shows the resulting graphs after contraction of edge x in the original graph and removing it from the dual.

The degree of a vertex cannot be bounded during the contraction step, see the vertices on the ends of x for example: two vertices of degree 3 are replaced by one vertex of degree 4.

3 The significance of the dual graph

Usually a planar graph cannot be interpreted (or drawn) in a unique way. We show this with a graph $G = (V, E), V = \{v_1, v_2, v_3, v_4, v_5\}$ and $E = \{(v_1, v_2), (v_1, v_3), (v_2, v_4), (v_3, v_4), (v_3, v_5), (v_4, v_5)\}$. Two different ways to draw it are shown in Figure 2.

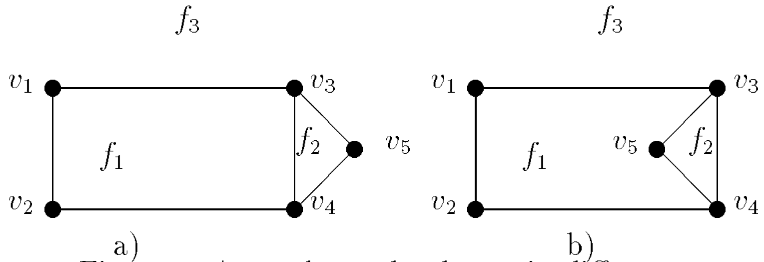

Figure 2: A graph can be drawn in different ways

The dual graph distinguishes between the two configurations: It consists of three dual vertices $F = \{f_1, f_2, f_3\}$. In Figure 2a) we see only one edge connecting f_1 and f_2 (v_3,v_4), in 2b) there are two edges connecting these faces ((v_3, v_5) and (v_4, v_5)).

The edges in the face graph correspond to borders of the regions which are represented by the vertices in the neighbourhood graph (chapter 6.3.1. in [9]). Using this information it is possible to reconstruct this border between two regions in the pyramid.

4 Building the pyramids

The process to build dual pyramids consists of two independent operations [4]:

Selecting the "decimation parameters" (surviving vertices and their receptive fields) by an arbitrary method. Some possible ways of doing this were mentioned in the introduction.

The algorithm to connect the vertices in the next level of the graph is **Dual Graph Contraction**, which can be viewed as a combination of two basic operations:

- **Edge Contraction:** The edges which describe the receptive field of a survivor are contracted in the neighbourhood graph and removed from the dual graph. Non surviving vertices deliver their data (e.g. incoming edges) to their parent (which is the surviving vertex of the receptive field to which they belong) during the merging process of edge contraction.
- **Face Contraction:** Edge contraction produces "degenerate" faces, which consist of edges which are self-loops or double edges. Some of these edges are contracted in the face graph and thus removed from the neighbourhood graph. This elimination can be performed in parallel in a logarithmic number of steps (accesses to edges) in the face graph [10]. The additional number of steps are necessary because contracting a face of degree one can decrease the degree of another face. See Chapter 4 of [12] for details.

To be able to deal with dual pyramids it is necessary to use planar graphs at the base level. The process described above creates a pair of planar graphs in all the higher levels, with a bounded degree for the vertices in the face graph. As a consequence of planarity at higher levels, a vertex cannot be connected to all the 8 neighbours surrounding it in the base level. The planarity is a reason to restrict the connection to 4 neighbours. Other possibilities to obtain planar graphs are triangular or hexagonal meshes, for example.

5 Visualisation of results

Edge and face contraction are shown in a small example in Fig. 3. In (a) we give a graph with surviving vertices ($s_1 \ldots s_5$, drawn as solid circles) and non surviving vertices ($n_1 \ldots n_5$, drawn as circles) and an assignment of the receptive field of the survivors indicated by arrows. In (b) we see the result after edge contraction. The "degenerated" faces are eliminated after face contraction (c).

We can use the information in the decimation parameters directly to draw the planar graph. The edge which connects the vertices s_2 and n_2 is stretched to connect s_2 and s_3 in the next level. This observation can be used to draw the planar (neighbourhood) graph, we do not draw an edge as a direct line between two vertices. Following the whole path which an edge was constructed, it is possible to draw the graph without any crossing of edges. Each edge at a higher level was constructed by the corresponding edge in the lower level and the concatenation of 0, 1 or 2 edges describing a receptive field in the lower level [5]. Recursive computing of those edges from the top to the base of the pyramid yields the desired path of an edge.

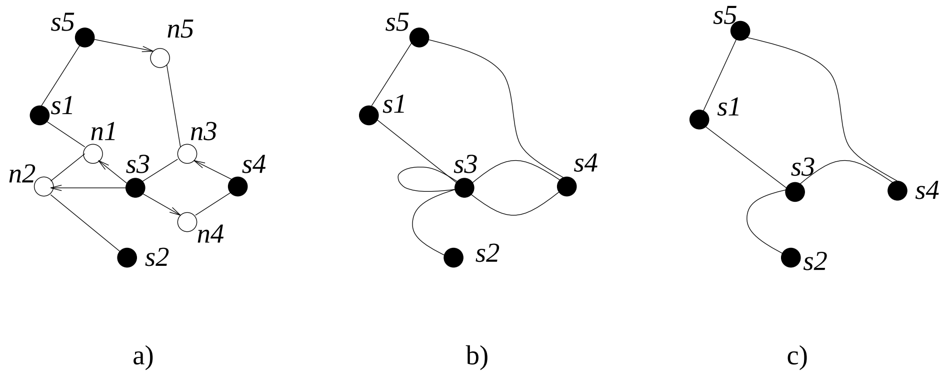

Figure 3: Visualisation of the two processes in the dual graph contraction

6 Searching for connected components

6.1 Our image "House"

Figure 4 shows an example of an image of size 16×16 pixels, similar to one used in [9], page 100. In the upper left corner we see the original image, from left to right and top to bottom the receptive fields of the vertices of the neighbourhood graph in 5 levels of searching for connected components are visualized in false colors.

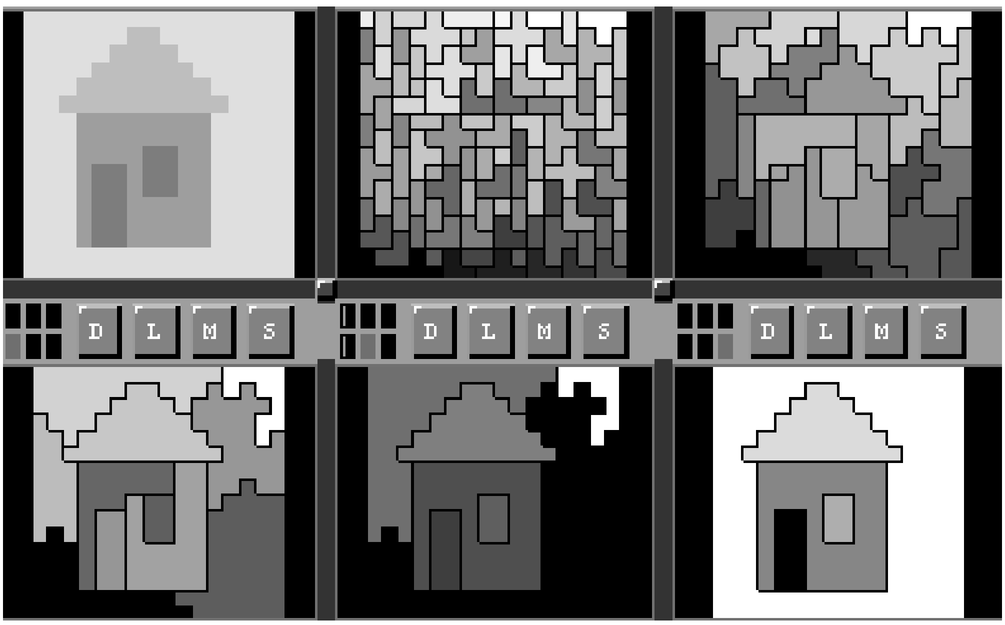

Figure 4: Steps to find connected components in the image "House"

The results were obtained with the principle of stochastic decimation for finding decimation parameters. Two modifications allow us to obtain adaptation to image data in the process of reduction:

- Two adjacent vertices in the neighbourhood graph are both allowed to survive if the difference in the graylevel is larger than a threshold.
- For the assignment of the receptive field, a non-surviving vertex chooses the surviving vertex with the smallest difference in gray level as its parent.

Using these supplementary rules the connected components can be found easily. Similar ideas with additional options were used by Bertolino [2], for example.

6.2 The resulting graph

An adjacency graph is the result of a conventional segmentation process [8]: a vertex represents a connected set of pixels in the image, edges describe neighbourhood relations between vertices. Since a vertex describes a region in the image, the term region adjacency graph (RAG) is often used as well.

The graph obtained by dual graph contraction shows more edges than the classical RAG, the motivation to introduce them is explained in [4]: ideally each dual edge corresponds to an intersection between two regions. Adjacent regions can have more than one common boundary, the consequence is that multiple edges between vertices in the neighbourhood graph occur as soon as the face graph is used additionally. The self-loop around the object *window* connects its border to the other borders of the *wall* artificially. This is shown in fig. 5(a).

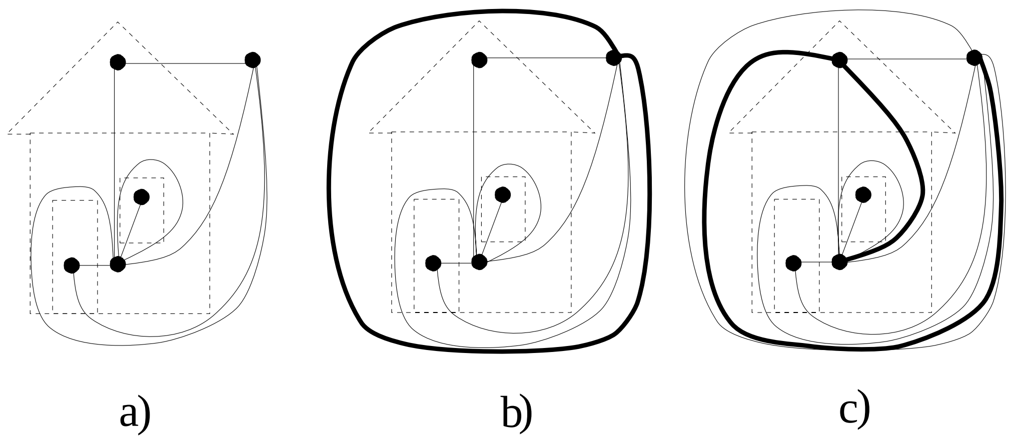

Figure 5: Explanation of additional edges in our graph

Following the algorithm like it is described in [4], one sees other edges survive, which were not described there: one is a second self-loop surrounding the whole house (see 5b), which can be interpreted as a description of the relation between the object *background* and the boundary of the image! An advantage of the dual graph is that we have the possibility to find all the objects which are on the boundary of the image. They have an incoming edge in which one of the dual vertices is a (not explicitly) represented vertex of an imaginary region outside the image. In our example only the self-loop carries this information.

Figure 5c) shows two additional edges between the *roof* and *background*. Each creates two faces of degree 3 instead of one face of degree 4.

In our implementation exactly 4 faces survive on a position that can be determined exactly. They are located on the intersection between 3 regions in the image. The position of other faces varies because the process which creates them depends on the “decimation parameters” which are generated by a random process.

Figure 6 shows two different runs of our experiments. The edges are drawn as described in section 5, the vertices of the RAG (describing regions) are indicated as black dots. Faces are drawn as small white squares, the beginning of the dual edges can be seen as short lines. The same position of the dual vertex on the border of the window in both experiments is just a random result.

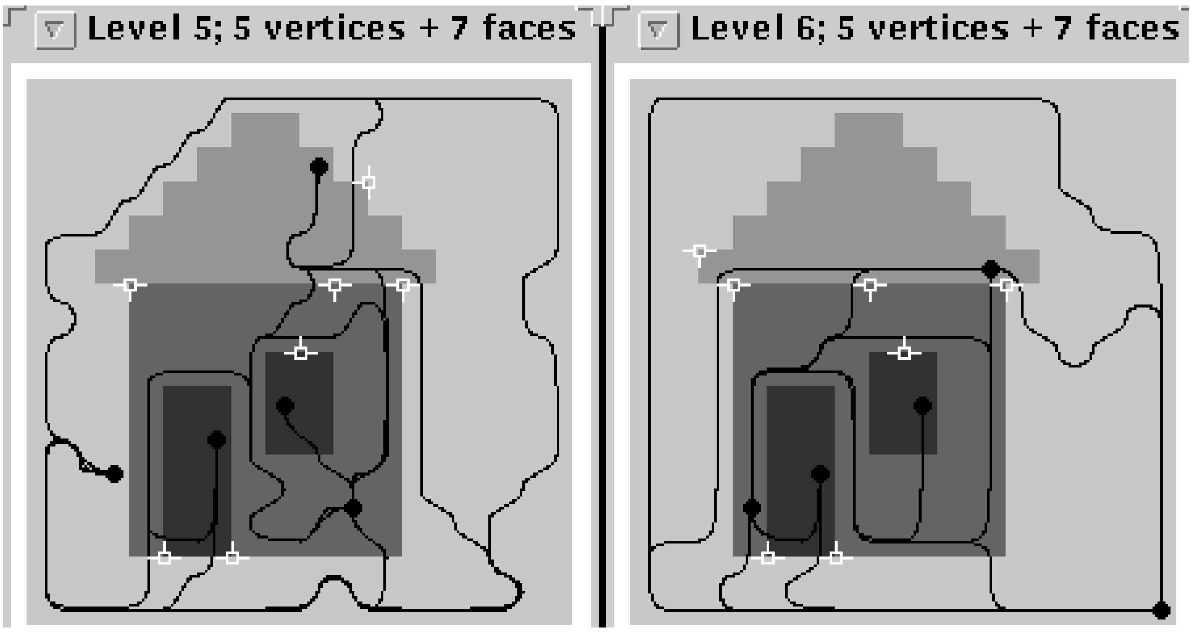

Figure 6: Location of vertices and dual vertices

The face graph, which has a bounded degree, describes borders between different regions in the image. Usually edges connect vertices which belong to different regions having a common boundary. In contrast, the self loops around the window and the house describe objects surrounded by other objects. Redundant parallel edges are necessary to provide the bounded degree in the face graph. With dual graphs it is possible to interpret the neighbourhood graph and the topological position of the objects.

7 Conclusion

Dual graph contraction has been used for finding the connected components of a labeled graph. Extensions to image segmentation are possible but not treated in this paper. The results differ mainly in two aspects from classical RAGs:

- The planar topology of the receptive field is captured correctly.
- Self-loops and double edges may occur in some cases.

Islands like the window in the wall are artificially connected to the remaining graph by an edge traversing a homogeneous region (the self-loop). Parallel edges provide the bounded degree in the face graph.

We are currently working on experiments to achieve segmentation of gray level images. Therefore it is necessary to use a vertex which describes “the region outside the image” (see Chapter 3 of [12]) to have a correct (dual) representation of regions on the border.

References

[1] Horst Bischof and Etienne Bertin and Pascal Bertolino. Voronoi Pyramids and Hopfield Networks. IEEE-Computer Society Press, *Proceedings of 12th ICPR (Jerusalem)*, Vol. III:pp 330–333, October 1994

[2] Pascal Bertolino. Structure Pyramidale Irregulière pour la Segmentation d'Images en niveaux de gris. *Université UJF Grenoble, Labratoire TIMC*, Master thesis, September 1992.

[3] Jean-Michel Jolion and Annick Montanvert. The adaptive pyramid, a framework for 2D image analysis. *Computer Vision, Graphics, and Image Processing: Image Understanding,* 55(3):pp 339-348, May 1992

[4] Walter G. Kropatsch. Building Irregular Pyramids by Dual Graph Contraction, *Technical Report PRIP-TR-35,* TU Vienna, July 26,1994

[5] Walter G. Kropatsch. Towards Higher Decimation Ratios. Submitted to CAIP 1995 in Prague.

[6] Stephen Lam, Horace Ip. Structural texture segmentation using irregular pyramid. *Pattern Recognition Letters*, pp691-698, July 1994

[7] Peter Meer. Stochastic Image Pyramids. *Computer Vision, Graphics and Image Processing*, Vol. 45 (No. 3): pp269-294, March 1989

[8] Annick Montanvert, Peter Meer and Azriel Rosenfeld. Hierarchical Image Analysis Using Irregular Tessellations. *IEEE transactions on Pattern Analysis and Machine Intelligence* **PAMI** -13(No.4):pp307-316, April 1991

[9] Peter Nacken. Image Analysis Methods Based on Hierarchies of Graphs and Multi-Scale Mathematical Morphology. *PhD-thesis, University of Amsterdam*, 1994

[10] Dieter Willersinn and W. G. Kropatsch. Dual Graph Contraction for Irregular Pyramids. IEEE-Computer Society Press, *Proceedings of 12th ICPR (Jerusalem)*, Vol. III:pp 251–256, October 1994

[11] Dieter Willersinn. Parallel Adaptive Contraction for Hierarchical Cellular Complexes. *Mustererkennung 1994, Erkennen und Lernen, 16. DAGM Symposium und 18. Workshop der ÖAGM*, Wien, Springer-Verlag, pp 401-408, Sept.1994

[12] Dieter Willersinn. Dual Irregular Pyramids. *PhD-thesis, Technical University of Vienna, Austria*, Feb. 1995