Contents

CHAPTER 1

Image Pyramid

" We think too small, like the frog at the bottom of the well. He thinks the sky is only as big as the top of the well. If he surfaced, he would have an entirely different view. "

by Mao Tse-Tung.

Summary Pyramids are hierarchical structures that are able to transform local information into global one. Two processing paradigms are used in pyramids: fine-to-coarse and coarse-to-fine information processing. The pyramid is a trade off between parallel architecture and the need for a hierarchical representation of an image at several resolutions. In this chapter an overview of basic concepts of image pyramid are presented. Their structure, content and information processing are discussed in more detail.

Keywords: Image pyramids, regular pyramid, irregular pyramid, structure of pyramid.

1.1 Introduction

The visual data are characterized with a large amount of data and high redundancy, relevant information are clustered in space and time, all this indicates for a need of organization and aggregation principles, in order to cope with computational complexity and to bridge the gap between raw data and symbolic description. Local processing is important in early vision, since operation like convolution, thresholding, mathematical morphology etc. belong to this class of operations. However, this approach is not efficient for high or intermediate level vision, such as symbolic manipulation, feature extraction etc., because these processes need both local and global information. Therefore a data structure must allow the transformation of **local** information (based on sub-images) into **global** information (based on the whole image), and be able to handle both local distributed and global centralized information. This data structure is known as *hierarchical architecture* [Jolion and Rosenfeld, 1994], and allows distribution of the global information to be used by local processes.

The earliest uses of hierarchical methods in image analysis are the work of [Rosenfeld and Thurston, 1971], finding edges by using differences of average gray levels in neighborhoods of many sizes, and selecting the best size at each pixel. The work of [Kelly, 1970] uses edgel detection in reduced resolution to guide the search for edges in a full-resolution image. This approach was one of the first to use a two level $(5 \times 5/25)$ multi-resolution hierarchy for edge detection. First, edges and lines are found in the reduced resolution and then used as a plan to constrain the search in the higher resolution. The name *pyramids* were first coined in [Tanimoto and Pavlidis, 1975] as solution of contour detection and delineation in digital images, and in conjunction with the merge and split segmentation algorithm of [Horowitz and Pavlidis, 1976] are used extensively in image segmentation.

The hierarchy is a stack of levels of reduced abstraction. There are two types of hierarchies [Jolion and Rosenfeld, 1994]:

- visual hierarchy, and
- conceptual hierarchy.

The conceptual hierarchy represent object relationship, like class inclusion or neighborhood relations. An example of a possible hierarchy of abstraction is given in Figure 1.1.

Usually, the bottom layer of the pyramid, is the image and the top layer or *apex* is related to more abstraction levels. Information flows up, down and laterally in the hierarchy and is transformed between layers. In hierarchies there are two kind of processes:

- bottom-up, and
- top-down.

In bottom-up or *fine to coarse* processes, the information is transported from the bottom to the top of the hierarchy. Information of the local data set is transformed into global one recursively. Bottom-up or data driven analysis process data in the uniform matter and aggregate data into more abstract higher level of representation. This allows extraction and detection of important features in image. During this bottom-up processes the data volume is reduced, meaning that levels toward the top will contain less data.

In the top-down process or *fine to coarse* feature values are propagated from the top to the bottom of the hierarchy. The top-down processed use knowledge (in form of a model) to search for the image data in an nonuniform manner to find support or against the presence of particular structure in the image. This process allows to delineate features extracted by bottom-up processes by propagating the global information to lower levels or in the coarse to fine strategy the higher (coarse) levels propose hypothesis that are used by lower (finer) levels to verify these hypothesis.

The pyramid is a trade off between parallel architecture and the need for a hierarchical representation of an image at several resolutions [Jolion and Rosenfeld, 1994]. There are other multi-resolution approaches, just to mention some without trying to be complete. [Witkin, 1986] introduced the continuous scale-space theory and [Lindeberg, 1990] proposed a discrete version of this theory. [Mallat, 1989] introduced the wavelet theory, another hierarchical processing

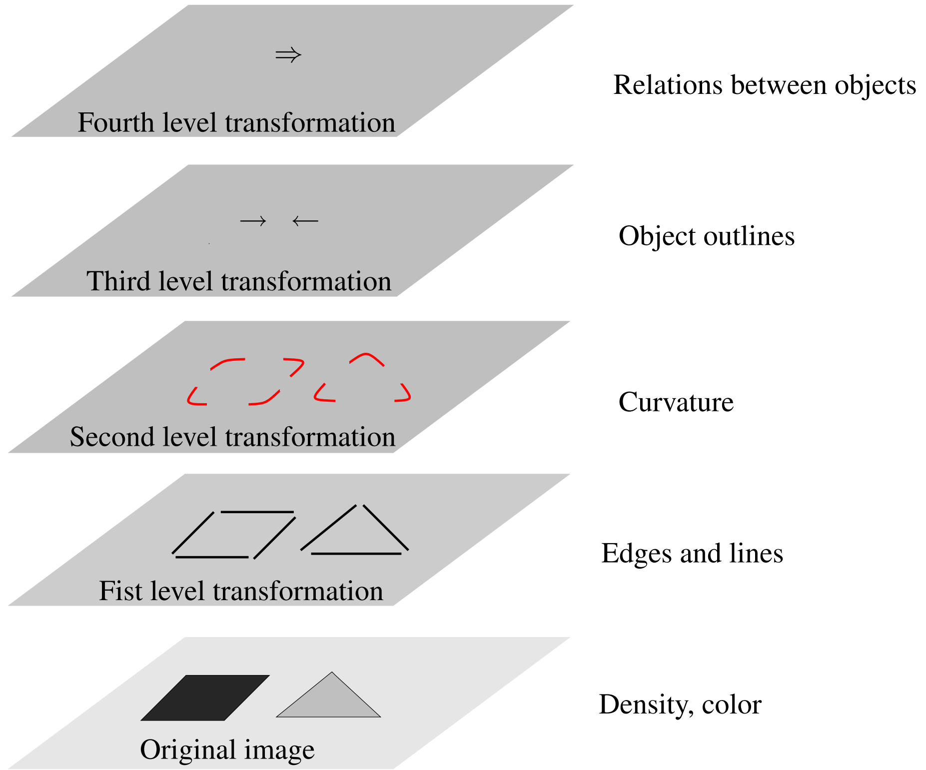


Figure 1.1: Conceptual representation of an image as an abstract hierarchy [Granlund, 1999].

paradigm. For an overview and the relation between different hierarchical approaches consult [Kropatsch, 1991].

In this chapter we summarize the concepts and definition developed for building and using multi-resolution pyramids [Rosenfeld, 1984], [Jolion and Rosenfeld, 1994], [Kropatsch, 1999]. First a definition of the image is presented in Section 1.2. Section 1.3 presents the architecture consideration of image pyramids. This chapter closes with a short summary in Section 1.4.

1.2 Discrete 2D Images

Almost every image processing technology processes the discrete spatial data by a computer. The function of mapping continuous space $\mathbb{R}^2$ into a discrete space $\mathbb{Z}^2$ is called *spatial quantization* or *digitisation* (in general one can think of a mapping from $\mathbb{R}^n$ into $\mathbb{Z}^n$). Seldom the sampling scheme is adapted to the local variability of the signal. Usually a regular sampling grid is considered, either a triangular or a square. In practice almost all capturing devices have a square regular sampling scheme, since the image sensors are arranged in 2D arrays located in the nodes of the regular grid. So the digital image is the result of sampling the continuous signal at location of the *sampling points*. A discrete representation of an image associates a numerical countable value with each point (x, y) in the digitization grid. Since it is practically impossible to measure the signal in the infinite small surface area and in an infinite small time step, each value is in fact an average over a sampling window and over a time. Thus sampling points can be considered as the centers of convex polygons [Soille, 1994], which tessellate the signal space. For non-regular grids these polygons are referred as Voronoi polygons, whereas for triangular grid this polygons have all hexagonal shape. In the case of square grids these polygons are called meshes. Usually these polygons are called in the digital image processing *pixels* (picture elements). The set of all pixels cover the entire image [Sonka et al., 1999].

The most primitive discrete representation assigns to each pixel an average measurement, which is in general a continuous value, but in image processing this value is *quantized*. The number of quantization levels should be high enough to allow humans a satisfactory perception of the image. Based on how many quantization levels are used to express the values of the pixel brightness (**g**), images are said to be binary or gray. In (multi-spectral) color images, vectors of scalar values are associated for each pixel. Hence a digital image is a finite set of triples:

$$(x, y, \mathbf{g}) \in \mathbb{Z}^3, \tag{1.1}$$

where (x, y) is the location of the pixel in the digitization grid and **g** is the quantized brightness function. x, y are usually called coordinated and are represented by using integer values. If **g** is a scalar of two values (0 and 1) the image is called a binary images; if **g** is an integer from the interval $[0, 255]$, the image is called a gray value image; if **g** is a vector of scalars from the interval $[0, 255]$ representing red, green and blue color, the image is called a color image and so on. Detailed definition of the image types are found in [Soille, 1994, Sonka et al., 1999].

To summarize, discretization process maps any object of the continuous image into a discrete version if it is sufficiently large to be captured by the sensors at the sampling points. Resolution relates the unit distance of the sampling grid with a distance in reality. The properties of the continuous object, i.e. color, texture, shape, as well as its relations to other (nearby) objects are mapped into the discrete space, too.

In this document only binary, gray and color images are analyzed, it is supposed that images are quantized with enough brightness levels such that problem with false contour do not occur. It is also assumed that during the spatial digitization the proper topology of the image object is also captured. For techniques of spatial digitization that preserves the topology consult [Stelldinger and Ullrich, 2005]. Note that the framework presented in this document is general enough and is not limited in using only images as inputs.

1.3 Pyramid Architecture

Image pyramid have been defined as a stack of images of decreasing resolutions [Burt et al., 1981], [Rosenfeld, 1982]. This framework is used for efficient data processing in may areas of computer vision [Rosenfeld, 1984, Jolion and Rosenfeld, 1994]. Usually, higher levels of the pyramid are computed successively by a filtering operation followed by a re-sampling operator [Rosenfeld, 1984].

Image pyramid have these advantages [Bister et al., 1990, Kropatsch et al., 1999]:

- details are removed in lower resolutions, thus reducing the influence of noise,
- resolution independent processing of regions of interest,
- transformation of local information to global one,

Table 1.1: Image qualities at different resolutions [Kropatsch et al., 1999]

Characteristics	Resolution	
	high	low
data amount	huge	smaller
computing times	long	short
details	rich and many	very few
overview	bad	good
precision	hight	low

- divide-and-conquer principle applied to reduce the computational complexity, and
- finding models at lower resolution, thus ignoring details, and employing these models in a low cost top-down model based analysis.

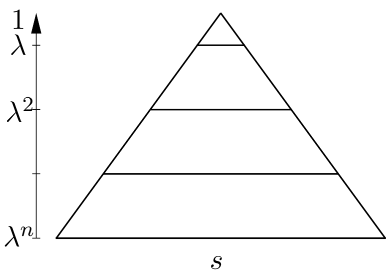
In Table 1.3 some qualities of different resolutions are shown (adapted from [Kropatsch et al., 1999]).

Image pyramids are characterized by these important properties [Jolion and Rosenfeld, 1994, Bischof, 1995]:

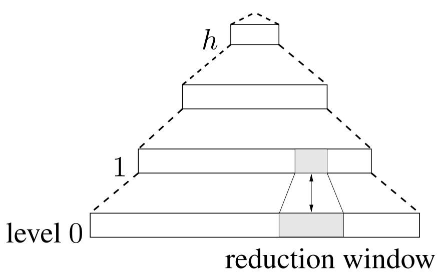
- structure[1], e.g. vertical and horizontal neighborhood relations
- content of the cell, e.g. pixel, region, edge, curve or more, and
- processing of the cells, e.g. filtering, symbolic processing.

A pyramid (Fig. 1.2a,b) describes the contents of an image at multiple levels of resolution. High resolution input image is at the base level. Successive levels reduce the size of the data by *reduction factor* $\lambda > 1.0$. *Reduction windows* relate one cell at the reduced level with a set of cells in the level directly below. Thus local independent (and parallel) processes propagate information up and down and laterally in the pyramid. The contents of a lower resolution cell is computed by means of a *reduction function* the input of which are the descriptions of the

[1] also called communication network.



(a) Pyramid concept



(b) Discrete levels

Figure 1.2: Multiresolution pyramid.

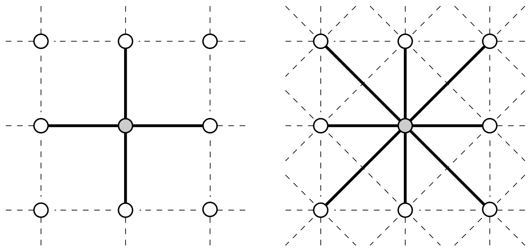


Figure 1.3: Neighborhood in a square grid graph.

cells in the reduction window. Two successive level of a pyramid are related by the reduction window and the reduction factor. Higher level description should be related to the original input data in the base of the pyramid. This is done by the *receptive field* (RF) of a given pyramidal cell c_i. The $RF(c_i)$ aggregates all cells (pixels) in the base level of which c_i is the ancestor. In the sections below these notions are explained in more detail following the work of [Jolion and Rosenfeld, 1994], [Bischof, 1995] and [Kropatsch et al., 1999].

1.3.1 Structure

The structure of the image pyramid is made of two types of neighborhood: the *horizontal*, i.e. intra-level neighborhood, and the *vertical*, i.e. inter-level neighborhood are defined in the image pyramid. Each cell of the pyramid is related with its neighbors in the same level and with other cells in the level above and below, except cells on the base and cell(s) on top of the pyramid. Cells on the base have only relation with the level above, and the cell(s) on the top has(ve) only relations with the level below. Let each cell (e.g. pixel, edge, region etc.) of the pyramid be represented by a vertex of the graph, and each level k of the pyramid by represented by a graph. Thus the horizontal and vertical network can be defined using graphs.

Horizontal neighborhood

Let level k of the image pyramid be represented by $G_k = (V_k, E_k)$, where V_k represent the set of cells, and E_k the set of edges joining cells Every vertex $v \in V_k$ on level, say k, is related to its neighbors on the same level by edge(s) $e \in E_k$. Two vertices, $v, w \in V_k$ are neighbors if they are joint by an edge $e = (v, w) \in E_k$ (Definition **??**). This defines the horizontal neighborhood. If the image plane is regularly tessellated, say by a regular grid mesh, than on the base level one can define 4 or 8 connectivity of cells, as shown in Figure 1.3, the cell (vertex) in gray in (a) has 4-neighborhood and under (b) 8-neighborhood. Note that 8-neighborhood grid graph is not planar. Other neighborhoods are possible, and are discussed in more detail in Chapter 2.

Vertical neighborhood

Every vertex in level k is linked with vertices on level directly below $k - 1$, its *children(s)*, and vertices on level directly above $k + 1$, its *parent(s)*. Vertices on the base level have no children(s), and those on the top have no parent(s). The number and form of these links define the vertical neighborhood, and this neighborhood be defined by an undirected (vertical) graph $G_k^V = (R_k, L_k)$, where $R_k = (V_k \cup V_{k+1})$, and $L_k \subseteq (V_k \times V_{k+1})$. The children relation of vertex $v \in V_{k+1}$ is defined as:

$$Ch(v, w) = \{\exists w | e = (v, w) \in L_k\}; \tag{1.2}$$

we say vertex w is a children of v. Analogously, the parent relation of a vertex $w \in V_k$ as:

$$P(w, v) = \{\exists v | e = (w, v) \in L_k\}; \tag{1.3}$$

we say vertex v is a parent of w. Note that these relations, are non-reflexive, anti-symmetric, and non-transitive. A parent has one or more children and in general a child can have many parents. If a child has many parent, the pyramid is called an *overlapping pyramid*. Based on these definition, through the transitive closure of these graphs, the *ancestor(s)* and *descendant(s)* of a given vertex can be defined. Let h be the height of the pyramid. The ancestor of a vertex $v \in V_k$ is vertex $w \in V_l$, $h \geq l > k$ if and only if:

$$A(v, w) = \{\exists z_n \in V_n \,|\, P(z_n, z_{n+1}) \forall n = k, ..., l-1\}, \text{ where } v = z_k \text{ and } w = z_l \tag{1.4}$$

Analogously, the descendant of a vertex $w \in V_l$ is vertex $v \in V_k$, $l > k \geq 0$ if and only if:

$$D(w, v) = \{\exists z_n \in V_n \,|\, Ch(z_n, z_{n-1}) \forall n = l, ..., k-1\}, \text{ where } w = z_l \text{ and } v = z_k \tag{1.5}$$

These definition can be used to define the *receptive field* of a vertex, as the set of all its descendants on the base level of the pyramid. Formally, the receptive field is the set of vertices on the base level $G_0 = (V_0, E_0)$ which influence the cell $v \in V_k$:

$$RF(v) = \{ \text{ all vertices } v' \in V_0 \,|\, D(v, v')\} \tag{1.6}$$

One can define also the *projective field* [Bischof, 1995] of vertex $v \in V_m$ on a given level V_n, $(n > m)$ as:

$$PF(v) = \{ \text{ all vertices } v' \in V_n \,|\, A(v, v')\} \tag{1.7}$$

In Figure 1.4, pictorially the concepts of parent-children, and ancestor-descendant relation are given with solid line. If we assume that $G_{k-1} = G_0$ is the bottom of the pyramid, then in this figure all the cells w' at the bottom enclosed in the gray area are the receptive field of $v : RF(v)$.

Two types of pyramid exist, based on how the vertical neighborhood is defined:

- regular, and
- irregular pyramids.

These concepts are strongly related to ability of the pyramid to represent the regular and irregular tessellation of the image plane.

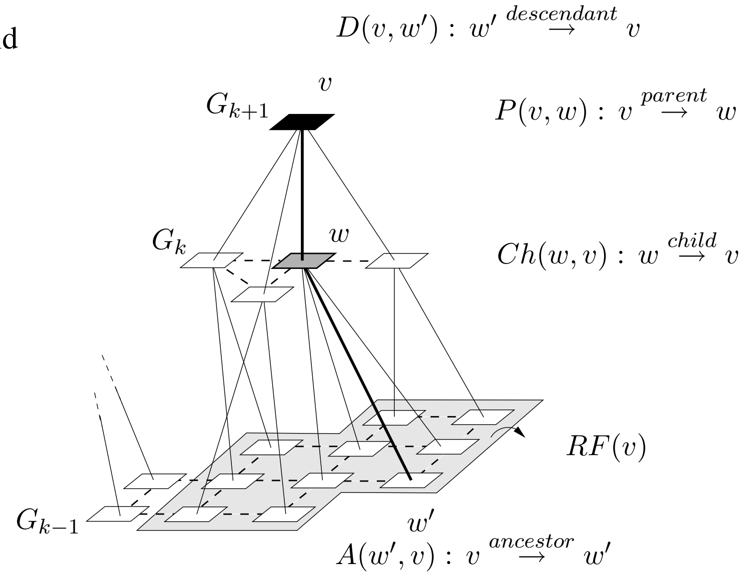
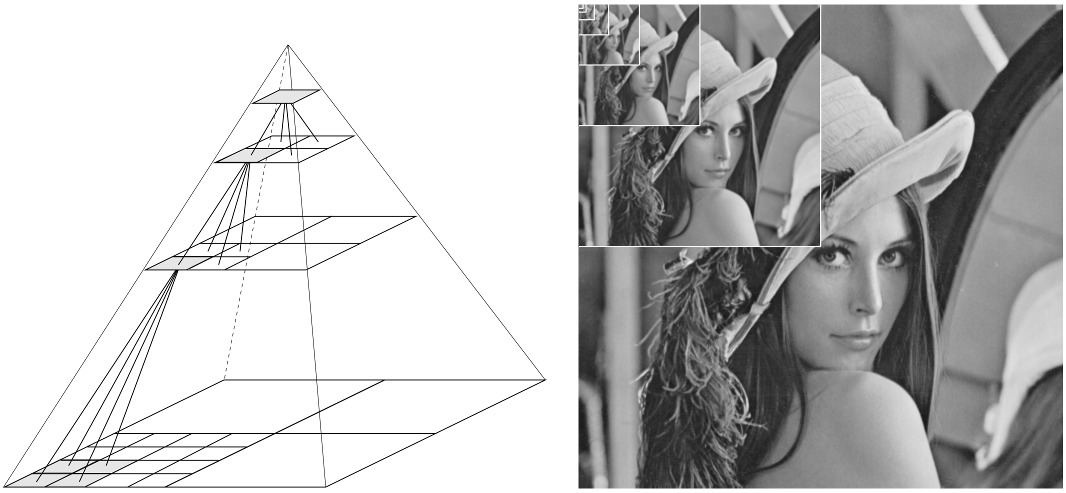


Figure 1.4: Vertical neighborhood.

Regular Pyramids

The *constant reduction factor* and *constant size reduction window* completely define the structure of the regular pyramid. The decrease rate of cells from level to level is determined by the reduction factor. The number of levels h is limited by the reduction factor $\lambda > 1$: $h \leq \log(image\_size)/\log(\lambda)$. The *main computational advantage* of regular image pyramids is due to this *logarithmic complexity*. Usually regular pyramids are employed in a regular grid tessellated image plane, therefore the reduction window is usually a square of $n \times n$, i.e. the $n \times n$ cells are associated by a cell on a higher level directly above. Regular pyramid are denoted using notation $n \times n/\lambda$. The vertical structure of a classical $2 \times 2/4$ is given in Figure 1.5a. In this regular pyramid $2 \times 2 = 4$ cells are related to only one cell in the higher level directly above. Since the children have only one father this class of pyramids is also called non-overlapping regular pyramids. Therefore the reduction factor is $\lambda = 4$. An example of $2 \times 2/4$ regular image pyramid is given in Figure 1.5b. The image size is $512 \times 512 = 2^9 \times 2^9$ therefore the image pyramid consist of $1 + 2 \cdot 2 + 4 \cdot 4 + ... + 2^8 \times 2^8 + 2^9 \times 2^9$ cells, and the height of this pyramid is 9. The pyramid levels are shown by a white border on the left upper corner of image. The $2 \times 2/4$ regular pyramid is called also the quad pyramid, because of the similarity with the quad tree representation [Samet, 1990]. See [Kropatsch, 1991] for extensive overview of other pyramid structures with overlapping reduction windows, e.g. $3 \times 3/2$, $5 \times 5/4$. It is possible to define pyramids on other plane tessellation, e.g. triangular tessellation [Jolion and Rosenfeld, 1994]

Thus, the regular image pyramid are efficient structure for fast grouping and access to image objects across the input image, because of the rigid vertical structure. globally defined sampling grids and lack shift invariance [Bister et al., 1990]. The regular pyramid representation of a shifted, rotated and/or scaled image is not unique, and moreover it does not preserve the connectivity. Thus, [Bister et al., 1990] concludes that regular image pyramids have to be rejected as general-purpose segmentation algorithms. This major drawback of the regular pyramid motivated a search for a structure that is able to adapt on the image data. It means, that the regularity of the structure is to be abandon.



(a) Vertical structure (b) Image pyramid

Figure 1.5: $2 \times 2/4$ regular pyramid.

Irregular Pyramids

Abandoning the regularity of the structure means that the horizontal and vertical neighborhood have to be explicitly represented, usually by using graph formalism. These not-regular structures are usually called *irregular pyramids*. One of the main goals of irregular pyramids is to achieve the shift invariance, and to overcome this major drawback of regular counterparts. [Kropatsch et al., 1999] list other motivations why one has to use irregular structures:

- arrangement of biological vision sensors is not completely regular,
- the CCD cameras cannot be produced without failure, resulting into an irregular sensor geometry,
- perturbation may destroy the regularity of regular pyramid, and
- image processing to arbitrary pixels arrangement (e.g. log-polar geometries [Bederson, 1992])

Two main processing characteristics of the regular pyramids should be preserved by building irregular ones [Bischof, 1995]:

- operation are local, i.e. the result is computed independently of the order, this allows parallelization, and
- bottom-up building of the irregular pyramid, with an exponentially decimation of the number of cells.

The structure of the regular pyramid as well as the reduction process is determined by the type of the pyramid (e.g. $2 \times 2/4$). Removing this regularity constraint one has to define a procedure to derive the structure of the reduced graph G_{k+1} from G_k, i.e. a graph contraction method has to be defined. Irregular pyramid can be build by parallel graph contraction [Rosenfeld, 1985], or graph decimation [Meer, 1989]. Parallel graph contraction has been developed only for special graph structures, like trees, and will not be analyzed in this thesis.

[Meer, 1989] introduced an efficient random decimation algorithm for building regular pyramid, called *stochastic pyramid*. A detailed discussion of this and similar methods is postponed until Chapter **??**. It is shown that stochastic pyramid in some cases is not logarithmically tapered, i.e. the decimation process does not exponentially reduce the number of cell successively. The main reason for this behavior is that the cell's neighborhood is not bounded, for some cases the degree of the cell increases exponentially. In Chapter **??** we discuss methods that overcome this drawback.

An overview of properties of regular and irregular pyramid are found in [Kropatsch and Montanvert, 1991]. In irregular pyramids the flexibility is paid by less efficient data access.

1.3.2 Contents

The type of information stored into cells defines also the the content of the pyramid. Numeric and/or symbolic information can be stored into the cells. A cell store the information that it is gathered (e.g. by a bottom-up process) from its receptive field. [Kropatsch, 1991] defines two conditions to be fulfilled by the representation of the cell:

- the receptive field of the cell covers the pictorial entity (e.g. primitive object parts, objects) completely, and
- no smaller cells (in lower levels) fulfill the above property

A unique cell in the pyramid is allocated for each pictorial entity. Usually, a cell stores only one numeric value, e.g. a gray value. This pyramid is called *numeric pyramid*. More complex numeric values can be stored as well, e.g. parameters of a model. Symbolic information can be stored into cells as well, e.g. curve information [Kropatsch, 1986, Kropatsch and Burge, 1998]. This class of pyramid is called *symbolic pyramid*. It is also possible to mix numeric and symbolic information into a cell as well.

1.3.3 Processing

The type of the information stored into cells of the pyramid conditioned also the processing that can be carried out by the pyramid. Therefore, two types of information processing are performed:

- numeric, and
- symbolic processing

Note that one of the major properties of the pyramid is the capability of local processing, i.e. the information in the cells is computed using only other cells in their horizontal and vertical neighborhood, thus these processing can be done in parallel. For an overview of pyramid hardware implementations see [Jolion and Rosenfeld, 1994].

Numeric Processing

Filtering is usually used as a reduction function in numerical pyramids. Different types of filters are used: 1) linear filters, 2) non-linear filters, and 3) morphological filters. Most commonly used filters are the low-pass ones. Gaussian filter is one of the most used low-pass linear filters in regular pyramid architecture, since it is the only filter that preserves the zero crossings across the scales [Yuille and Poggio, 1986]. An example of the Guassian pyramid is given in Figure 1.5b. The resulted pyramid is called *Gaussian pyramid* [Burt and Adelson, 1983]. Minimum and the maximum filter are most commonly used non-liner filters.These filters compute the minimum, respectively maximum, of the receptive field of the cell. One can also use other non-linear filters, e.g. median. The morphological operators [Serra, 1982] are introduced into pyramids as well. A large body of the literature exist of using different filters integrated into a pyramid.

Symbolic Processing

In symbolic pyramid one has to define also symbolic reduction functions. A finite state machine can be used to compute the symbolic reduction. E.g. [Kropatsch, 1986, Kropatsch and Burge, 1998] introduces a set of rules as reduction function.

1.4 Summary

All pyramids are characterized by three properties: its structure (horizontal and vertical neighborhood), its cell content and the way it processes the information content of the cells. We distinguish two main types of pyramids based on the structure: if the structure is beforehand defined the pyramid is called regular; if the structure is adapted on the image data it is called an irregular pyramid. Numeric or/and symbolic information can be stored into the pyramid, therefore one can differentiate between numeric pyramids and symbolic pyramids. Different filters can be used in the pyramid framework, which allows also symbolic computations. In the rest of the chapters we will intensify the discussion on irregular graph pyramids. We will discuss how to optimize the structure of the irregular pyramid, trying to archive a logarithmic height, and how to apply efficiently these structures in image segmentation.

CHAPTER 2

Irregular Dual Graph Pyramids

" Beauty is the first test: there is no permanent place in the world of mathematics for ugly mathematics. "

by G.H. Hardy.

Summary The duality concept in planar graphs is presented. It is shown how the dual graphs can encode a topology properly. The transformation of image plane into a dual graph is described. The dual graph contraction, as a topology preserving graph contraction is presented in depth. This graph contraction method contains basic operations to build a stack of hierarchical graphs, called dual graph pyramid. The construction of the dual graph pyramid is shown by an example.

Keywords: Topology, dual graph, planar graph, dual graph contraction, dual graph pyramids, topology preserving contraction.

2.1 Introduction

Most information in vision today is in the form of array representation. This is advantageous and easily manageable for situations having the same resolution, size, and other typical properties equivalent. Various demands are appearing upon more flexibility and performance, which makes the use of array representation less attractive [Granlund, 1999]. The increasing use of actively controlled and multiple sensors requires a more flexible processing and representation structure [Kropatsch et al., 1999, Kropatsch, 2002]. Cheaper CCD sensor could be produced if defective pixels would be allowed, which yields in the resulting irregular sensor geometry [Bederson, 1992], [Wallace et al., 1994]. Image processing functions should be generalized to arbitrary pixel geometries [Rojer and Schwartz, 1990], [Bederson, 1992]. The conventional array form of image is impractical as it has to be searched and processed every time if some action is to be performed and that

- features of interest my be very sparse over parts of an array, leaving a large number of unused positions in the array;
- a description of additional detail can not be easily added to a particular part of an array.

It is desirable to have information in some partly interpreted form to fulfill its purpose to rapidly evoke actions. Information in interpreted form, implies that it should be represented in terms of content or *semantic* information, rather than in terms of array values. Content and semantics implies *relations* between units of information or symbols. [Granlund, 1999] calls the relations between object as *linked objects*. These object could be represented as a *graph* [Haralick and Shapiro, 1993].

In order to express the connectivity or other geometric or topological properties the image representation must be enhanced by a neighborhood relation. In the regular square grid arrangement of sampling points it is implicitly encoded as 4- or 8-neighborhood with the well known paradox in conjunction with Jordan's curve theorem. The neighborhood of sampling points can be represented explicitly, too: in this case the sampling grid is represented by a *graph* consisting of vertices corresponding to the sampling points and of edges connecting neighboring vertices. Although this data structure consumes more memory space it has several advantages, as follows [Kropatsch et al., 1999]:

- the sampling points need not be arranged in a regular grid,
- the edges can receive additional attributes too, and
- the edges may be determined either automatically or depending on the data.

Planarity and duality of graph are two closely related concepts. Planar graph separates the plane into regions called faces. This idea of separating the plane into regions is helpful in defining the dual graphs (Section 2.2). Duality of a graph brings together two important concepts in graph theory: cycles and cut-sets. Kirkoff's laws of voltage and current in electrical engineering are the real wold problem of this duality concept. The law of voltage is in term of cycles and the law of current is in term of cut-sets. This concept of duality is also encountered in graph-theoretical approach of image region and edge extraction. The definition of dual graphs representing the partitioning of the plane, allows one to apply transformation on these graphs, like edge contraction and/or removal to simplify graphs in the sense of less vertices and edges. Edge contraction and removal introduces naturally a hierarchy of dual graphs, the so called *dual graph pyramid*.

Hence the dual graph representation presented in this chapter addresses primarily the structure, on which a dual graph hierarchy is built, by reducing the number of descriptive elements by applying the dual graph contraction successively. In this chapter, the graph-based image representation and the operation on these graphs are given. In Section 2.3 the transformation of image plane into a dual graph is shown. The analogy of dual graphs with abstract cellular complexes, and the equivalence with combinatorial maps is given in Section 2.3.1, and Section 2.3.2 respectively. The dual edge contraction algorithm is described in Section 2.4 and the hierarchy of graphs build by this algorithm in Section 2.5.

2.2 Planar and Dual Graphs

A graph $\widetilde{G}$ of finite sets of vertices V and edges E is called a *plane graph* if it is drawn in a plane in $\mathbb{R}^2$ such that [Diestel, 1997]:

- all $V \subset \mathbb{R}^2$
- every edge is an arc[1] between two vertices,
- no two edges are crossed.

Note that $\mathbb{R} \setminus \widetilde{G}$ is an open set and its connected regions are faces f of $\widetilde{G}$. It is said that the plane graph divides the plane into regions. Since $\widetilde{G}$ is finite, one of its faces in an unbounded one (infinite area). This face is called the *background face*[2]. Other faces enclose finite areas, and are called interior faces. Edges and vertices incident with the face are called the boundary elements.

Definition 2.1 *An embedding of a graph G on a plane is an isomorphism between G and a plane graph $\widetilde{G}$.*

$\widetilde{G}$ is called a drawing of G. When G is drawn, we do not care whether its edges intersect on vertices only or each other. This is in contrast to drawing $\widetilde{G}$ where its edges intersect only on vertices. The graph G can be considered as an abstraction of $\widetilde{G}$. An example of a planar graph and its planar embedding is shown in Figure 2.1. The edge e_5 in Figure 2.1b is drawn such that it does not intersect with other edges, whereas in Figure 2.1b we do not ask strictly for edges not to intersect. In this figure f represent faces, and a background face is denoted by b. For example the edges e_1, e_2 and e_5 form the face f_1 and they make the cycle $C_{f_1} = \{e_1, e_2, e_5\}$, which enclose the region f_1. The cycle that enclose the background face b in this figure is $C_b = \{e_5, e_8, e_6\}$.

Definition 2.2 (Planar graphs) *A graph G is planar if it can be embedded on the plane.*

The concept of embeddings can be extended to any surface. A graph G is embeddable in surface S if it can be drawn in S so that its edges intersect only on their end vertices. A graph embeddable on the plane is embeddable on the sphere too. It can be shown by using the stereoscopic projection of the sphere onto a plane [Thulasiraman and Swamy, 1992]. Note that the concept of faces is also applicable to spherical embeddings.

A planar graph $G = (V, E)$ with m vertices, n edges and l faces satisfies the condition:

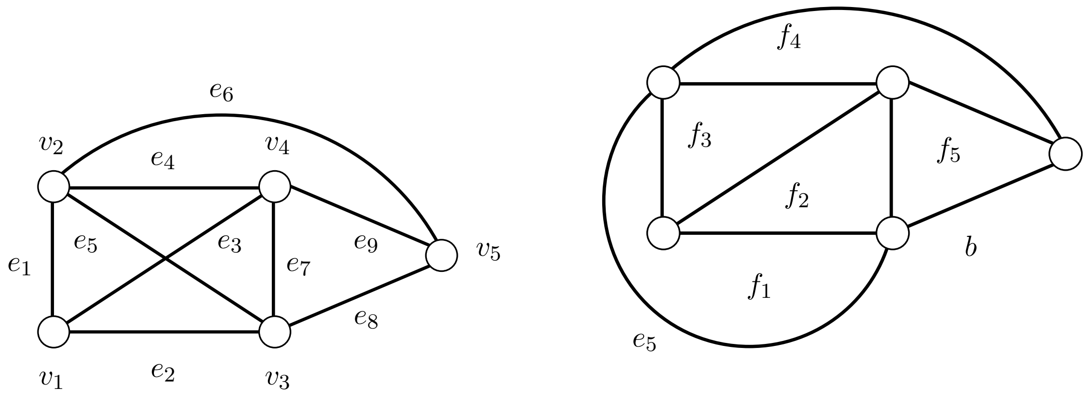
$$m - n + l = 2. \tag{2.1}$$

This is called Euler's formula[3]. Another nice property of simple planar graph is that if G has $m \geq 3$ vertices and n edges than $n \leq 3m - 6$ edges. The so called Kuratowski graphs K_5

[1] An arc is a finite union of straight line segments, and a straight line segment in the Euclidean plane is a subset of $\mathbb{R}^2$ of the form $\{x + \lambda(y - x) | 0 \leq \lambda \leq 1\}, \forall x \neq y \in \mathbb{R}^2$.

[2] Called also exterior face.

[3] Called also Euler characteristic.



(a) $G = (V, E)$ (b) $\widetilde{G}$

Figure 2.1: A planar graph G and its embedding in a plane, the plane graph $\widetilde{G}$.

(complete graph on 5 vertices) and $K_{3,3}$ are not planar. The planar graph can be characterized also using the Kuratowski graphs [Harary, 1969].

Let G in Figure 2.1a represent a plane graph, in general with parallel edges and self-loops. Since the graph is embedded onto a plane, it divides the plane into faces. Let each of these faces be denoted by a new vertex say fs, and let these vertices be put inside the faces, as shown in Figure 2.1b. From this point on the notion of face vertices and face are synonymous. Let the faces that are neighbors, i.e. that share the same edge e_2 (incident on the same edge), be connected by the edge, say $\overline{e}_2$, so that edge e and $\overline{e}$ are crossed. At the end, for each edge $e_2 \in G$ there is an edge $\overline{e}_2$ of the newly created graph $\overline{G}$, which is called the dual graph of G. If the e is incident only with one face a self-loop edge $\overline{e}_2$ is attached to the vertex on the face in which the edge e_2 lays, of course e_2 and the self-loop edge $\overline{e}_2$ have to cross each other (see Appendix **??** for a procedure for building dual graph out of plane ones). The adjacency of faces is expressed by the graph $\overline{G}$.

More formally one can define dual graphs for a given plane graph $G = (V, E)$ in this form [Thulasiraman and Swamy, 1992]:

Definition 2.3 (Dual graphs) *A graph $\overline{G} = (\overline{V}, \overline{E})$ is a dual of $G = (V, E)$ if there is bijection between the edges of G and $\overline{G}$, such that a set of edges in $\overline{G}$ is a cycle vector of $\overline{G}$ if and only if the corresponding set of edges in G is a cut vector.*

There is a one-to-one correspondence between the vertex set $\overline{V}$ of $\overline{G}$ and the face set F of G, therefore sometimes graph $\overline{G} = (\overline{V}, \overline{E})$ is written as $\overline{G} = (F, \overline{E})$ instead, without fear of confusion. In order to show that $\overline{G}$ is a dual of G, one has to prove that vectors forming a basis of the cycle subspace of $\overline{G}$ correspond to the vectors forming a basis of the cut subspace of G. The edges e_i of graph G in Figure 2.2 correspond to edges $\overline{e}_i$ in graph $\overline{G}$. The cycles $\{e_1, e_3, e_4\}$, $\{e_2, e_3, e_6\}$, $\{e_4, e_5, e_8\}$, and $\{e_6, e_7, e_8\}$ form a basis of the cycle subspace of G (see Chapter **??**, Section **??**). These cycles correspond to the set of edges $\{\overline{e}_1, \overline{e}_3, \overline{e}_4\}$, $\{\overline{e}_2, \overline{e}_3, \overline{e}_6\}$, $\{\overline{e}_4, \overline{e}_5, \overline{e}_8\}$, and $\{\overline{e}_6, \overline{e}_7, \overline{e}_8\}$, which form a basis of cut subspace of $\overline{G}$. It follows according to the definition of the duality, that graph $\overline{G}$ is a dual of G. By convention, the graph G is called the *primal graph* and $\overline{G}$ *dual graph*. If a planar graph G' is the dual of G,

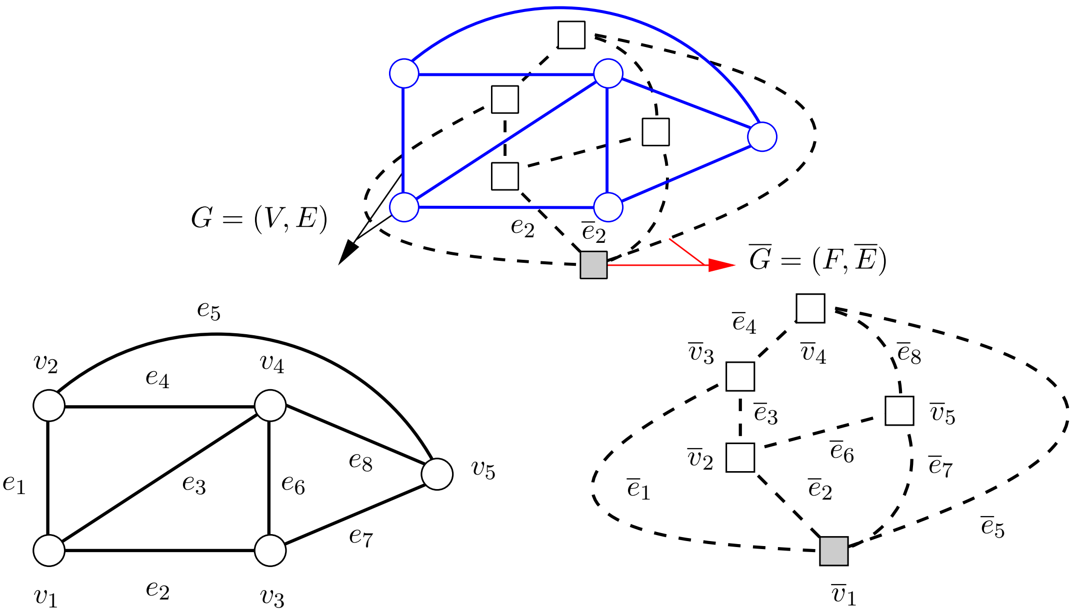


Figure 2.2: A plane graph G and it dual $\overline{G}$.

then a planar G is a dual of G' as well, and every planar graph has a dual [Diestel, 1997, Harary, 1969]. Therefore the dual of a dual graph is the primal graph. Dual graphs are denoted by a bar above the capital letter.

In the following two important properties of dual graphs with respect to the edge contraction and removal operation are given, the proofs are due to [Thulasiraman and Swamy, 1992]. These properties are used in Section 2.4 to prove that graphs during the process of dual graph contraction stay planar and are duals. Let G and $\overline{G}$ be two graphs. Let edge $\overline{e} \in \overline{G}$ correspond to edge $e \in G$. Note that a cycle in G corresponds to a cut in $\overline{G}$ and vice versa [Thulasiraman and Swamy, 1992]. Let $\overline{G'}$ denote the graph $\overline{G}$ after the *contraction* of the edge $\overline{e}$, and G' the graph after the *removal* of the corresponding edge e from G.

Theorem 2.1 *A graph and its dual are duals also after the removal of an edge e in the primal graph G and the contraction of the corresponding edge $\overline{e}$ in the dual graph $\overline{G}$.*

Proof: Let C and $\overline{C}$ be the corresponding set of edges in G and $\overline{G}$, respectively. Assume that C is a cycle in G'. Since it does not contain e, it is also a cycle in G. Hence $\overline{C}$ is a cut, say $\{\overline{V_1}, \overline{V_2}\}$ in $\overline{G}$. Since cut $\overline{C}$ does not contain $\overline{e} = (\overline{v_1}, \overline{v_2})$, the vertices $\overline{v_1}$ and $\overline{v_2}$ are both either in $\overline{V_1}$ or in $\overline{V_2}$, implying $\overline{C}$ is a cut in $\overline{G'}$. Therefore every cycle in G' is a cut in $\overline{G'}$.

Suppose that $\overline{C}$ is a cut in $\overline{G'}$. Since $\overline{C}$ does not contain $\overline{e}$, it is also a cut in $\overline{G}$. Hence C is a cycle in G'. Since it does not contain e, it is also a cycle in G. Thus every cut in $\overline{G'}$ is a cycle in G'. □

Corollary 2.1 *If a graph G has a dual, then every edge-induced subgraph of G has also a dual.*

Proof: Every edge-induced subgraph G' of G can be obtained by removing from G the edges not in G' and using the Theorem 2.1, the proof follows. □

This section is concluded by the a topological characterization of graphs that have duals. The duality property of a graph is as important as planarity, and these properties are symbiotics.

Theorem 2.2 (Whitney 1933) *A graph is planar if and only if it has a dual.*

Proof: The proof can be found in [Diestel, 1997]. □

A detailed discussion of the data structures for the dual graph are given in Appendix **??**.

2.3 Dual Image Graphs

An image is transformed into a graph such that, for each pixel a vertex is associated, and pixels that are neighbors in the sampling grid are joint by an edge . Note that no restriction in the sampling grid is made, therefore an image of regular as well as non-regular sampling grid can be transformed into a graph. The gray value or any other feature is simply considered as an attribute of a vertex (and/or an edge). Since the image is finite and connected, the graph is finite and connected as well. The graph which represent the pixels is denoted by $G = (V, E)$ and is called *primal graph*[4]. Note that pixels represent finite regions, and the graph G is representing in fact a graph with faces as vertices. The dual of a face graph (see Section 2.2) is the graph representing borders of the faces, which in fact are inter-pixel edges and inter-pixel vertices [Braquelaire and Brun, 1998, Kropatsch, 1994]. This graph is denoted by $\overline{G}$ and is called simply *dual graph*. This discussion, which is done on purpose here to show the duality concept, is not in contradiction with the presentation of dual graphs given in Section 2.2, because of the property of dual graphs to be dual of each other. The above discussion could have been started by defining the primary graph as the graph denoting borders of faces, and its dual would have been the graph representing the faces. An example of image with square grid sampling transformed into a graph is given in Figure 2.3, where square vertices represent faces, the bold square vertex represents the background face, circle vertices represent meeting points of at least three boundary segments.

Based on the Theorem 2.2 dual graphs are planar, therefore images with square grid are transformed into $4-$ connected square grid graphs, since $8-$ connected square grid graphs are in general not planar [5].

The same formalism as is done for the pixels can be used at intermediate levels in image analysis i.e. for region adjacency graphs (RAGs). RAGs are the results of image segmentation processes. Region are connected sets of pixels, and are separated by region borders. Its geometric dual causes problems [Kropatsch, 1995a]. Let a simple example, adapted from [Kropatsch, 1994], clarifies this claim and motivate the usage of pair of graph (duals of each other) to represent an image or in general adjacency of regions. Note that in this presentation the graph and its duals are depicted as in Section 2.2, and is different with the above presentation of dual graphs. In the example of the house in Figure 2.4a the graph $\overline{G'}$ representing regions of the house, like door, windows etc., is depicted by square vertices and dashed edges. To reconstruct the boundary graph G' out of the face graph $\overline{G'}$, circle vertices are put where at least three boundary segments intersect. The edges of G are drawn between these vertices following the boundary

[4] Called also neighborhood graph.

[5] This hold for square grid graphs of grid size $\geq 4 \times 4$.

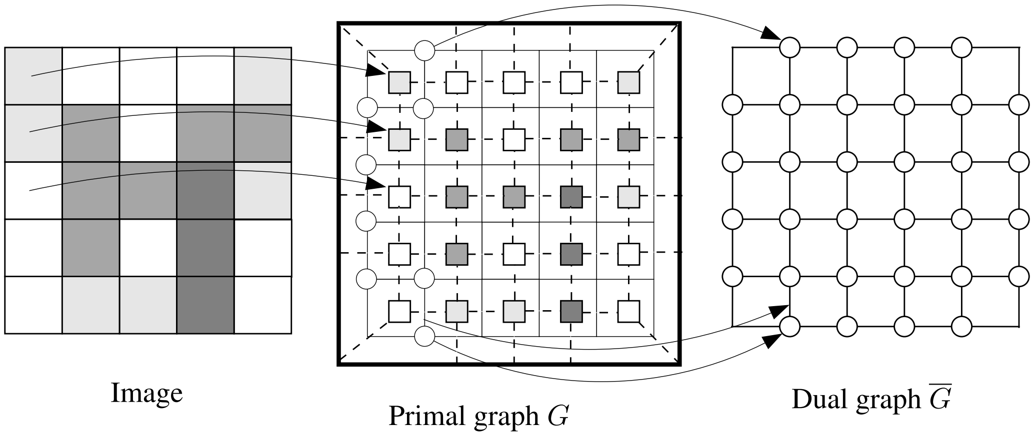


Figure 2.3: Image to dual graphs.

of the house regions such that these edges cross the edges of the graph $\overline{G'}$ (as shown with bold lines in Figure 2.4b. As can be seen from the Figure 2.4b there are two problems:

- one boundary of the front side of the house is not crossed by any edge in $\overline{G'}$, and
- the above described procedure will not produce an edge for the window since there is no crossing of the border of the window with the surrounding region of the front side, and therefore no clear answer where to put a vertex of G'.

The problems are encountered due to the fact that the border of the front side consists of two parts not connected to each other: the inner part of the window, and the outer part which is also fragmented into pieces separating the front from the roof, from the background and from the door. A resolution to these problems is given in Figure 2.4c: a self loop is put in $\overline{G'}$ surrounding the window, a circle vertex is put in G' arbitrarily in the window border and this vertex is connected by a fictive edge [6] such that it connects the boundary of the window with the front

[6] Called a bridge in [Kropatsch, 1994].

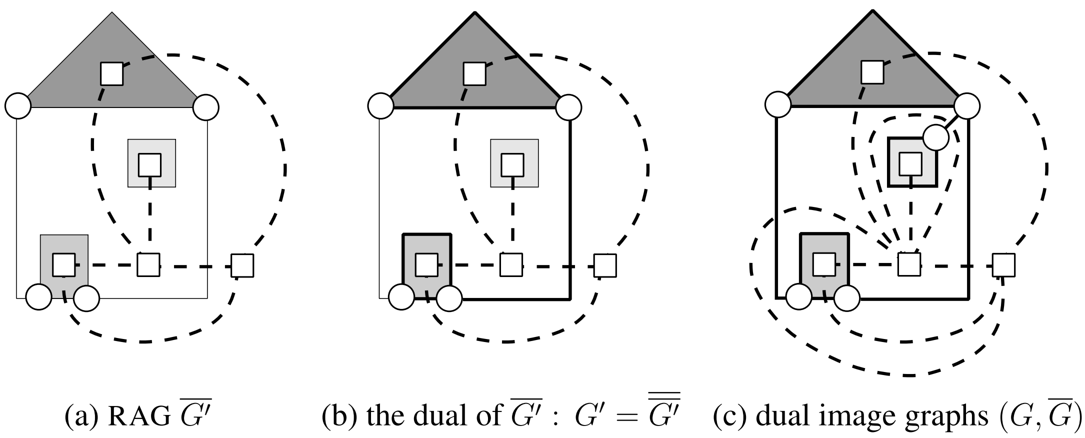


Figure 2.4: A house example and the dual image graph [Kropatsch, 1994].

side of the house, and front side and background are connected by parallel edges in $\overline{G}$. The resulting pair of graph $(G, \overline{G})$ are plane, dual of each other, and in general not simple, it contains self-loops and parallel edges. This section is concluded by a formal definition of the dual image graphs:

Definition 2.4 (Dual image graphs [Kropatsch, 1994]) *The pair of graphs $(G, \overline{G})$, where $G = (V, E)$ and $\overline{G} = (\overline{V}, \overline{E})$ are called dual image graphs if both of the graphs are finite, planar, connected, not simple in general and duals of each other.*

Dual graphs can be seen as an extension of the well know region adjacency graphs (RAG). Note that this representation is capable to encode multiple boundaries between neighboring regions. See the outside border of the house in Figure 2.4c connected multiple times (roof, left wall, part of the door and right wall) with the background.

2.3.1 Dual Image Graph and Cellular Complexes

In this section a relation between dual graphs and 2D-abstract cellular cells (ACC) is shown. [Kovalevsky, 1989] showed that on the abstract cellular cells one can define a topological space. Topology is formally defined as:

Definition 2.5 (Topology) *Let $\mathcal{X}$ be a non-empty set, the universe. A topology on $\mathcal{X}$ is a family $\mathcal{T}$ of subsets of $\mathcal{X}$ such that:*

(T1) $\mathcal{X}$ and $\emptyset$ belong to $\mathcal{T}$,

(T2) the union of any number of sets of $\mathcal{T}$ belongs to $\mathcal{T}$,

(T3) the intersection of any two sets of $\mathcal{T}$ belongs to $\mathcal{T}$.

A set $\mathcal{X}$ for which a topology $\mathcal{T}$ has been specified is called a topological space. The members of $\mathcal{T}$ are called open sets. A subset C of $\mathcal{X}$ is called closed set if its compliment C^c is in $\mathcal{T}$. Analogously to the previous definition one can define a topological space using only closed sets. If the set $\mathcal{X}$ contains finite number of elements then it is called finite topology. A set $\mathcal{X} = \{1, 2, 3\}$ with $\mathcal{T} = \{\{\emptyset\}, \{1, 2, 3\}\}$ is a trivial topology and $\mathcal{X}$ a topological space. The trivial topology is the smallest topology on the set $\mathcal{X}$. The largest topology on $\mathcal{X}$ is called *discrete topology*. For the set example above the discrete topology is $\mathcal{T} = \{\{\emptyset\}, \{1, 2, 3\}, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}\}$. A detailed treatment of topological spaces can be found in [Munkres, 1993].

Definition 2.6 (Abstract cell complex [Kovalevsky, 1989]) *An abstract cell complex $\mathcal{C} = (O, B, dim)$ is a set O of abstract elements, with a binary relation $B \subset O \times O$ called bounding relation, which is antisymmetric, irreflexive and transitive; and with dimension mapping $dim : O \to \mathbb{I}$, from O into a set of non-negative integers $\mathbb{I}$ such that $\dim(o) < dim(o')$ for all $(o, o') \in B$.*

If the dimension of cell o is d then o is called d-dimensional cell or simply d-cell, and ACC is n dimensional if one of the cells is n-cell. An example of the 2-dimensional ACC is shown in Figure 2.5, where 0-cells are vertices, 1-cells are edges, and 2-cells are faces. Formally,

$$O = V \cup E \cup F, \tag{2.2}$$

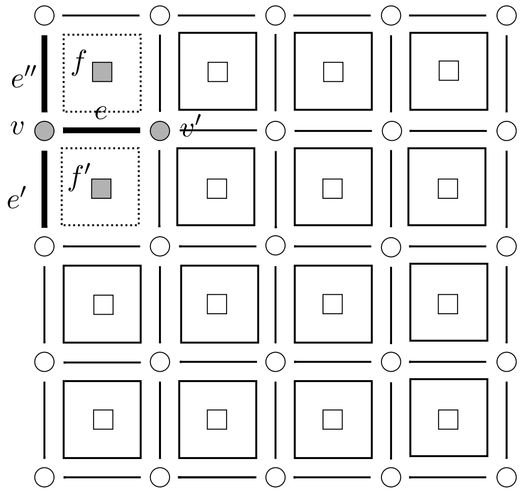


Figure 2.5: A 2-dimensional cellular model.

and

$$\dim(o) = \begin{cases} 0 & \text{if } o \in V, \\ 1 & \text{if } o \in E, \\ 2 & \text{if } o \in F. \end{cases} \quad (2.3)$$

A simple example of a 2D finite topological space is given in Figure 2.5; it consists of three types of elements: faces (□), edges (−) and vertices (❍). As can be seen from the figure, an edge bounds two faces, say f and f'. An edge e is bounded by two vertices, say v and v'; two faces f and f' are bounded by these two vertices as well. Let any subset O of faces, edges and vertices be called *open set*, such that every element $o \in O$ of the surface bounded by o is also in O. From this definition a face is not bounded by any element, therefore it is an open subset; an edge e with the two faces f and f' that it bounds is an open subset; a vertex v with all the edges (e, e', e'') that it bounds and all the couple of faces (f, f') bounded by these edges is an open set. A set of cells containing o and all higher dimensional cells bounded by o is called the open star of cell o [Ahronovitz et al., 1995]. The open sets defined like this satisfy the axioms (T1), (T2) and (T3) and define a topological space [Kovalevsky, 1993, Ahronovitz et al., 1995].

Note that abstract cells should not be regarded only as Euclidean point set. For a square grid image, the pixels could be considered as square faces, inter-pixels boundaries as edges, and the intersection of two or more inter-pixel boundaries could be represented with vertices. Note that this is exactly the same representation for the 2D space as the dual image graphs [Kropatsch, 1994]. The cell model is more general and can be used to represent space element of higher dimension, whereas the dual graph representation for higher dimension is not defined yet. In 2D space a dual graphs representation is able to encode any subdivision of the 2D topological space. Encoding higher dimension with (dual) graphs is a difficult problem.

2.3.2 Dual Image Graph and Combinatorial Map

The usage of dual graph framework for 3 or higher dimension is cumbersome and not well defined. These problems are alleviated by using the combinatorial maps or generalized maps. N-dimensional combinatorial maps [Lienhardt, 1989] may be seen as a graph with an embedding in an N-dimensional space i.e in the case of 2D [Brun and Kropatsch, 2001b], combinatorial maps are planar graphs encoding the orientation of edges around vertices. The base elements of a N-dimensional combinatorial map are the darts, also called half edges, which are connected together (sewed) by the orbits of 1 permutation and $N-1$ involutions. In the case of 2D [Brun and Kropatsch, 2001b] the permutation is called σ and forms vertices, and the involution is called α and specifies edges. One of the advantages of combinatorial maps is that in the 2D case, unlike dual-graphs, they explicitly encode the orientation of the plane, correctly handling all the complicated cases with self-loops and parallel edges.

Like combinatorial maps, n-dimensional Generalized maps [Lienhardt, 1991] are defined in any dimension and correctly represent all topological configurations of the n-dimensional space (including 2D). Their base elements are darts and use only involutions to represent the connections between them, that describe cells in any dimension.

Combinatorial maps and generalized combinatorial maps define a general framework which allows to encode any subdivision on nD topological spaces orientable or non-orientable with or without boundaries. Using 2D images, combinatorial maps my be understood as a particular encoding of a planar graph, where each edge is split into two half-edges called darts. Since each edge connects two vertices, each dart belongs to only one vertex. A 2D combinatorial map is formally defined by the triplet $G = (\mathcal{D}, \sigma, \alpha)$ [Brun and Kropatsch, 2001b] where $\mathcal{D}$ represent the set of darts and $\sigma(d)$ is a permutation on $\mathcal{D}$ encountered when turning clockwise around each vertex. Finally $\alpha(d)$ is an involution on $\mathcal{D}$ which maps each of the two darts of one edge to the other one. Given a combinatorial map $G = (\mathcal{D}, \sigma, \alpha)$, its dual is defined by $\overline{G} = (\mathcal{D}, \varphi, \alpha)$, with $\varphi = \sigma \circ \alpha$. The cycles of permutation φ encode the faces of the combinatorial map. In what it follows, the cycles of α, $\sigma(d)$ and φ containing a dart d will be respectively denoted by $\alpha^*(d)$, $\sigma^*(d)$ and $\varphi^*(d)$. An example of the combinatorial map is given in Figure 2.6a and (b).

Thus all graph definitions used in irregular graph pyramids [Kropatsch, 1994] are analogously defined with combinatorial maps [Brun and Kropatsch, 2003].

2.3.3 Dual Graphs versus Combinatorial Maps

Advantages of combinatorial maps over dual graphs come form the embedding, that is inherently present at the former ones. Let us analyze the *'flower'* example given in Figure 2.6b, and (c) with respect to uniqueness of topological representation. The combinatorial map of this *'flower'* is shown and defined in Figure 2.6b by $G = (\mathcal{D}, \sigma, \alpha)$[7]. If the leafs of the 'flower' exchange position for e.g. leafs 1 and 3, a different $\sigma = (3, -3, 2, -2, 1, -1, 4, -4)$ will be defined, hence uniquely encoding the topology. The dual graphs are encoded by a pair of graphs, the (planar) primal graph vertices) and its dual. For each edge in the primal graph there is a corresponding one in the dual, that crosses it (Figure 2.6c). Since there is no ordering of the edges around the vertices, the dual graph representation does not uniquely encode the topology of the *'flower'*, as can be easily seen if we exchange the position, for e.g. of leafs 1 and 3, the dual graph describing this configuration is identical to the previous one (the one without exchanging the position of leafs.) The presence of similar cases happens very rare in 2D, thus the dual graph representation is used as the representation in the rest of discussion.

[7] σ is encoded clockwise, shown with the arrow in Figure 2.6b.

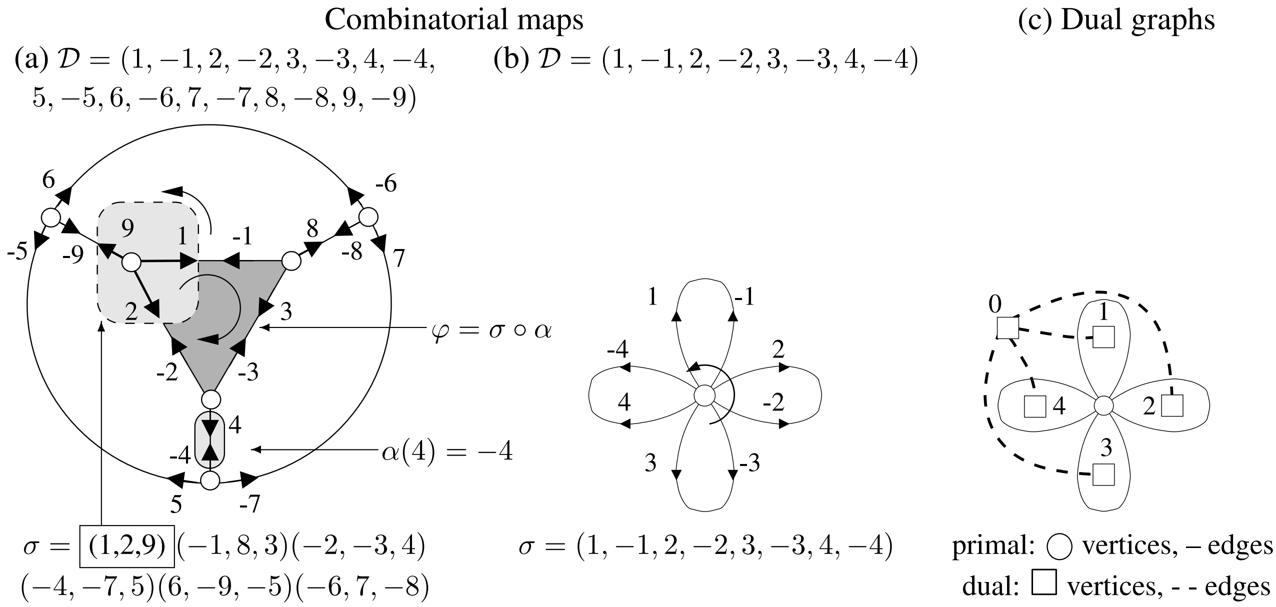


Figure 2.6: Combinatorial map and correctly handling topology.

2.4 Dual Graph Contraction

The irregular (dual graph) pyramids are constructed in bottom-up way such that subsequent level (say $k + 1$) results by (dually) contracting precedent level (say k). In this section a short exposition of dual graph contraction algorithm is given, following the work of [Kropatsch, 1995a], and building the dual graph pyramid using this algorithm is presented in the next section. Dual graph contraction (DGC) [Kropatsch, 1994], [Kropatsch, 1995a] proceeds in two steps:

I. primal-edge contraction and removal of its dual, and
II. dual-edge contraction and removal of its primal.

In Figure 2.7 examples of these two steps are shown in three possible cases. Note that these two steps correspond in [Kropatsch, 1995a] to the steps (I) dual edge contraction, and (II) dual face contraction.

The base of the pyramid consists of the pair of dual image graphs $(G_0, \overline{G_0})$. In order to proceed with the dual graph contraction a set of so called contraction kernels (decimation parameters) must be defined. The formal definition is postponed until the Section 2.4.1. Let the set of contraction kernels be $\langle S_k, N_{k,k+1} \rangle$. This set consists of a subset of surviving vertices $S_k = V_{k+1} \subset V_k$, and a subset of non-surviving primal-edges $N_{k,k+1} \subset E_k$ (where index $k, k + 1$ refer to contraction from level k to $k + 1$). Surviving vertices in $v \in S_k$ are vertices not to be touched by the contraction, i.e. after contraction these vertices make up the set V_{k+1} of the graph G_{k+1}; and every non-surviving vertex $v \in V_k \backslash S_k$ must be paired to one surviving vertex in a unique way, by non-surviving primal-edges (Figure 2.8). In this figure, the shadowed vertex s is the survivor and this vertex is connected with arrow edges (ns) with non-surviving

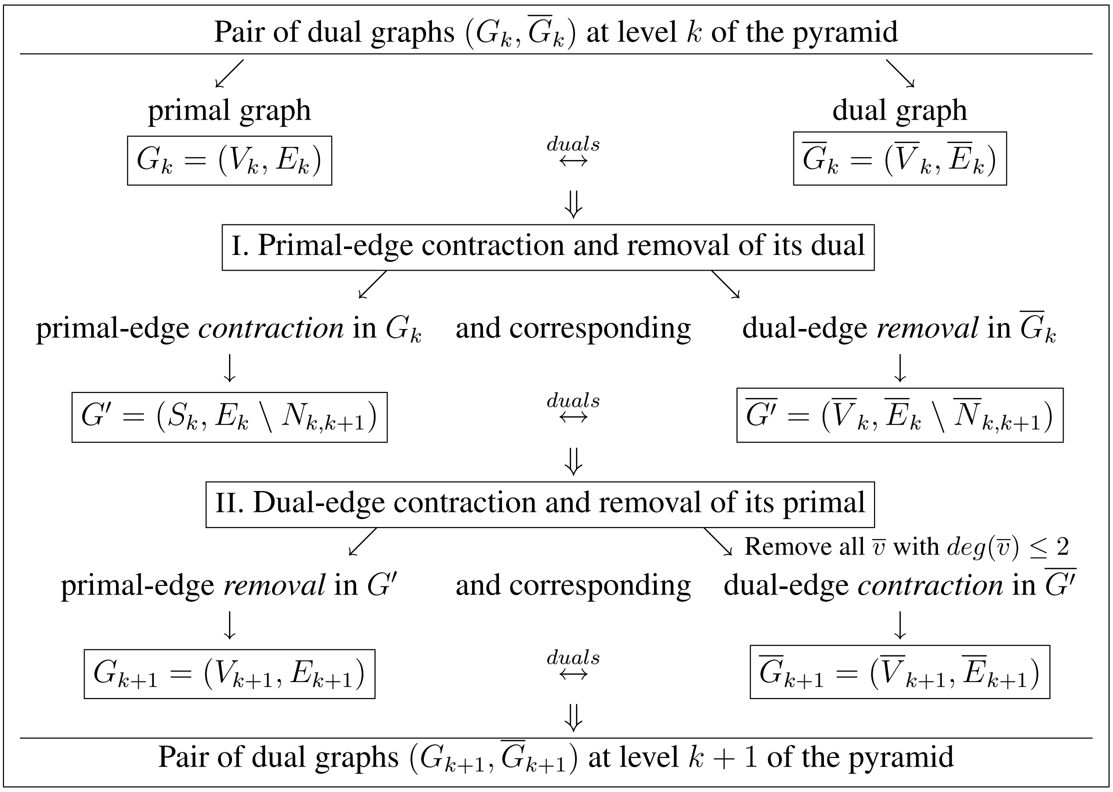


Figure 2.7: Dual graph contraction procedure (DGC).

vertices. Note that a contraction kernel is a tree of depth one, i.e. there is only one edge between a survivor and a non-survivor, or analogously one can say that the diameter of this tree is two.

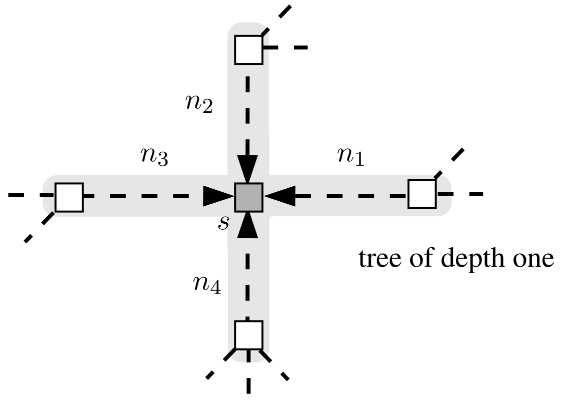
The contraction of a non-surviving primal-edge consists in the identification of its endpoints (vertices) and the removal of both the contracted primal-edge and its dual edge (see Chapter **??**, Section **??** for more details on these operations). Figure 2.9a shows the normal situation, Figure 2.9b the situation where the primal-edge contraction creates multiple edges, Figure 2.9c and self-loops. Redundancies (lower part) in case Figure 2.9c are decided through the corresponding dual graphs and removed by dual graph contraction. In Figure 2.9, the primal graph is shown with square (☐) vertices and broken lines (- -) and its dual with circle vertices (❍) and full lines (-).

[Kropatsch, 1995a] shows that $\langle S_k, N_{k,k+1} \rangle$ determine the structure of an irregular pyramid. The relation between two pairs of dual graphs,$(G_k, \overline{G}_k)$ and $(G_{k+1}, \overline{G}_{k+1})$, is established by dual graph contraction with the set of contraction kernels $\langle S_k, N_{k,k+1} \rangle$ as:

$$(G_{k+1}, \overline{G}_{k+1}) = C[(G_k, \overline{G}_k), \langle S_k, N_{k,k+1} \rangle]. \tag{2.4}$$

Dual-edge contraction and removal of its primal (second step) has a role of cleaning the primal graph by simplifying most of the multiple edges and self-loops[8], but not those enclosing

[8]Called also redundant edges.



$s \in S_k$ and $n_1, n_2, n_3, n_4 \in N_{k,k+1}$

Figure 2.8: Contraction kernel with s as a survivor and arrow edges (n-s) to be contracted.

any surviving parts of the graph. They are necessary to preserve correct structure [Kropatsch, 1995a]. So, the dual graph contraction reduces the number of vertices and edges of a pair of dual graphs, while preserving the topological relations among surviving parts of the graph. In [Kropatsch, 1994, Willersinn, 1995, Kropatsch, 1995b], a detailed presentation of dual graph contraction with some computational aspects (time and space complexity) are given. Moreover, in [Willersinn, 1995] different implementation details are presented.

2.4.1 Contraction Kernels

Let S be the set of surviving vertices, and N the set of non-surviving primal-edges. The connected components [9] $CC(s), s \in S$, of subgraph (S, N) form a set of tree structures $T(s)$ that if contracted would collapse into vertex s of the contracted graph. The number of this trees is $|S|$. The union of trees $T(s)$ contain the non-surviving primal-edges N. $T(s)$ is a spanning tree of the connected component $CC(s)$, or equivalently, (V, N) is a spanning forest of graph $G = (V, E)$.

In order to decimate the graph $G = (V, E)$ a set of *surviving* vertices $S \subset V$ and a set of *non-surviving primal-edges* $N \subset E$ must be selected, such that the following conditions are satisfied:

- graph (V, N) is a spanning forest of graph $G = (V, E)$, and
- the surviving vertices $s \in S \subset V$ are the roots of the forest (V, N).

Definition 2.7 (Contraction kernels) *The set of disjoint rooted trees with length two of path going through the root is called a set of contraction kernels.*

Analogously, the trees $T(v)$ of the forest (V, N) with root $v \in V$ are *contraction kernels*.

[9] Neglected level index refer to contraction from level k to level $k + 1$.

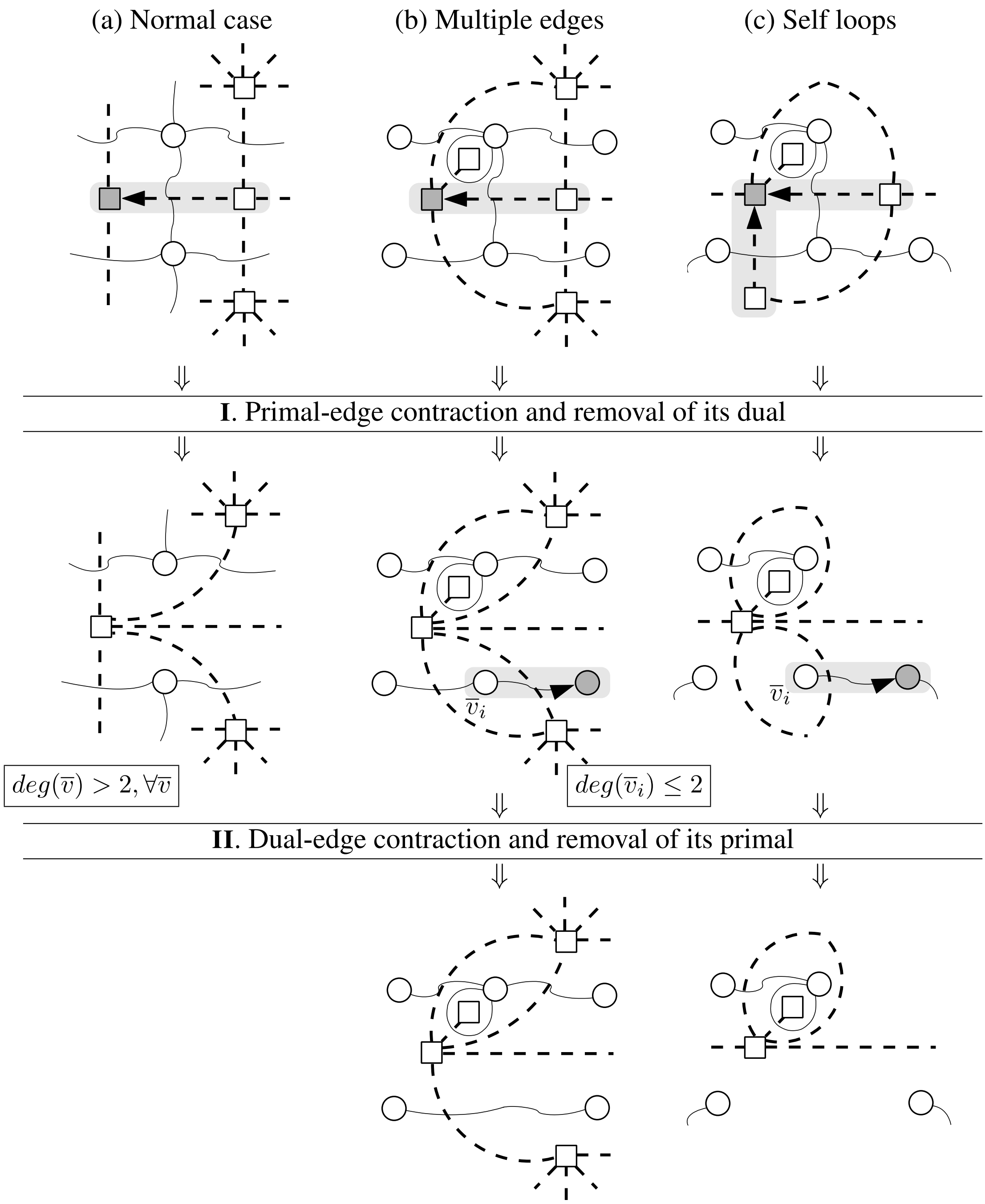


Figure 2.9: Dual graph contraction of a part of a graph.

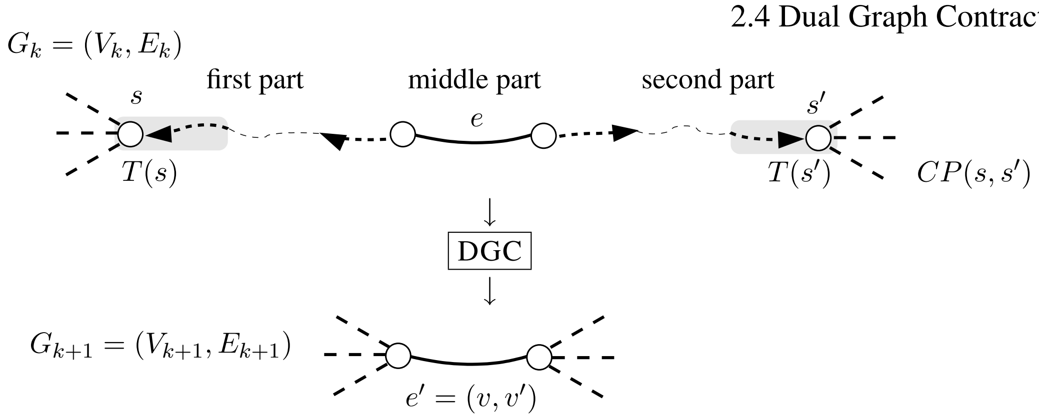


Figure 2.10: Connecting path $CP(v, v')$, e is the bridge of this path.

After deploying the dual graph contraction algorithm on a graph one has to establish a path connecting between two surviving vertices on the resulted new graph. Let $G = (V, E)$ be a graph with decimation parameters (S, N).

Definition 2.8 (Connecting path [Kropatsch, 1994]) *A path in $G = (V, E)$ is called a connecting path between two surviving vertices $s, s' \in S$ if it consists of three subsets of edges:*

- *the first part is a possibly empty branch of contraction kernel $T(s)$.*
- *the middle part is an edge $e \in E \setminus N$ that bridges the gap between the two contraction kernels $T(s)$ and $T(s')$.*
- *the third part is a possibly empty branch of contraction kernel $T(s')$.*

See Figure 2.10 for explanation. Connecting path is denoted by $CP(s, s')$. Edge e is called the *bridge* of the connecting path $CP(s, s')$. Each edge $e' = (v, v') \in E_{k+1}$ has a corresponding connecting path $CP_k(s, s')$, where $s, s' \in S \subset V_k$ are survivors in graph $G_k = (V_k, E_k)$. This means that two surviving vertices s, and s', $s \neq s'$ that can be connected by a path[10] $CP_k(s, s')$ in G_k are connected by an edge in E_{k+1}. If the graph G_k is connected, than after the dual graph contraction the connectivity of the graph G_{k+1} is preserved [Kropatsch, 1994].

Dual edge contraction can be implemented by (1) simply renaming all the non-surviving vertices to their surviving parent vertex (e.g. by using a find union set algorithm [Cormen et al., 2001]), (2) deleting all non-surviving edges N and (3) their duals $\overline{N}$. The question on how to build contraction kernels is given in Chapter **??**, where different methods are presented and their properties are analyzed.

2.4.2 Equivalent contraction kernels

[Burt and Adelson, 1983] combines two or more successive reductions in one equivalent weighting function in order to compute any level of any regular pyramid directly from the base level.

[10] By definition of connectivity of graph, there exist always a path between any two vertices of graph, see Chapter **??**.

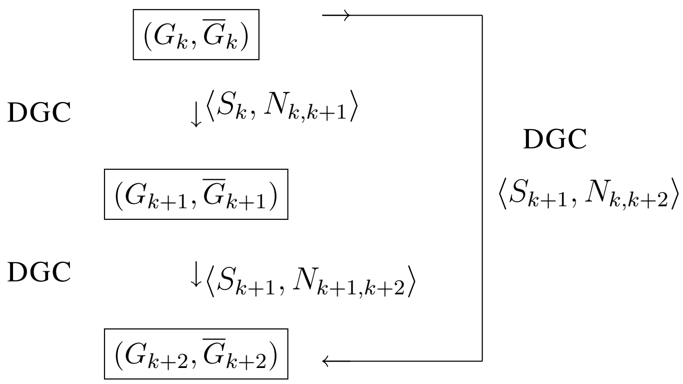


Figure 2.11: Equivalent contraction kernel.

Similarly, [Kropatsch, 1995b] combines two (or more) dual graph contractions (as shown in Figure 2.11) of graph $G_k = (V_k, E_k)$ with decimation parameters $\langle S_k, N_{k,k+1} \rangle$ and $\langle S_{k+1}, N_{k+1,k+2} \rangle$ into one single equivalent contraction kernel (ECK) $N_{k,k+2} = N_{k,k+1} \circ N_{k+1,k+2}$[11]:

$$
\begin{aligned}
C[C[G_k, \langle S_k, N_{k,k+1} \rangle], \langle S_{k+1}, N_{k+1,k+2} \rangle] &= C[G_k, \langle S_{k+1}, N_{k,k+2} \rangle] \\
&= G_{k+2}
\end{aligned}
\tag{2.5}
$$

The structure of G_{k+1} is determined by G_k and the decimation parameters $\langle S_k, N_{k,k+1} \rangle$. Simple overlaying the two sets of contraction kernels, $\langle S_k, N_{k,k+1} \rangle$ (the one from level k to $k+1$) and $\langle S_{k+1}, N_{k+1,k+2} \rangle$ (the one from level $k+1$ to $k+2$) will not yield a proper equivalent contraction kernel $\langle S_{k+1}, N_{k,k+2} \rangle$. The surviving vertices from G_k to G_{k+2} are $S_{k+1} = V_{k+2}$. The edges of the searched contraction kernels must be formed by edges $N_{k,k+2} \subset E_k$. An edge $e_{k+1} = (v_{k+1}, v'_{k+1}) \in N_{k+1,k+2}$ corresponds to a connecting path $CP_k(v_{k+1}, v'_{k+1})$ in G_k[12]. By Definition 2.8, $CP_k(v_{k+1}, v'_{k+1})$ consists of one branch of $T_k(v_{k+1})$, one branch of $T_k(v'_{k+1})$, and one surviving edge $e_k \in E_k$ connecting the two contraction kernels $T_k(v_{k+1})$, and $T_k(v'_{k+1})$.

Definition 2.9 (Bridge [Kropatsch, 1994]) *Function bridge*: $E_{k+1} \mapsto E_k$ *assigns to each edge* $e_{k+1} = (v_{k+1}, w_{k+1}) \in E_{k+1}$ *one of the bridges* $e_k \in E_k$ *of the connecting paths* $CP_k(v_{k+1}, w_{k+1})$*:*

$$
bridge(e_{k+1}) = e_k. \tag{2.6}
$$

Connecting two disjoint tree structures by a single edge results in a new tree structure. Now, $N_{k,k+2}$ can be defined as the result of connecting all contraction kernels T_k by bridges as:

$$
N_{k,k+2} = N_{k,k+1} \quad \cup \bigcup_{e_{k+1} \in N_{k+1,k+2}} \text{bridge}(e_{k+1}) \tag{2.7}
$$

This definition satisfies the requirements of a contraction kernel [Kropatsch, 1994]. Analogously, the above process can be repeated for any pair of levels k and k' such that $k < k'$. If

[11] Only for G_k is shown instead of $(G_k, \overline{G_k})$ for simplicity.

[12] If there are more than one connecting paths, one is selected.

$k = 0$ and $k' = h$, where h is the level index of the top of the pyramid, than with the resulted equivalent contraction kernel ($N_{0,h}$), the base level (0) is contracted in one step into an apex $V_h = \{v_h\}$. ECKs are able to compute any level of the pyramid directly from the base.

2.4.3 Homotopy Preserving Transformations

As already discussed in Section 2.2, the plane graph partitions the plane into faces. Faces can have holes which are represented by connected components, and have to be connected by the fictive edges with the connected component which surrounds it (see the Figure 2.4c where the window is connected by a fictive edge with the front of the house). The notion of homotopy on (dual) graphs, i.e. for the 2D case, is derived from [Serra, 1982, page 187]. Let $\mathcal{G}$ be the set of all graphs.

Definition 2.10 (Homotopy graphs) *A mapping Ψ from $\mathcal{G}$ into itself is said to be homotopic (or preserves the homotopy) if it transforms a graph G into a graph $\Psi(G)$ such that there is a one-to-one and onto correspondence between connected components and the holes of G and $\Psi(G)$.*

Two finite graphs G and G' are said to be homotopic when there exist a homotopic transformation Ψ such that $G' = \Psi(G)$. [Marchadier et al., 2003] proved that the set of transformation that preserves the homotopy of (dual) graphs are:

- contraction of an edge, which is not a self loop,
- removal of a pendant edge, and
- contraction of redundant edges.

All these transformation do not change the number of connected components and holes of the dual graphs, therefore they preserve the graph homotopy. In the case of the 2D, the homotopy can be defined also using the homotopy tree (or adjacency tree) [Soille, 1994, page 56]. Two set are homotopic if their homotopy trees are identical.

2.4.4 Graph Minors with Dual Graph Contraction

Two containment relations between graphs: the subgraph relation and the induced subgraph relation are exposed in Chapter **??**, Section **??**. The graph contraction operation is described in details in this section and in Chapter **??**, Section **??**. In this section the minor relation is presented. Minor relation is closely related with the graph contraction operation, as will be shown.

Let $G' = (V', E')$ be a graph and $\{V_v \mid v \in V'\}$ a set partition of V' into connected subset such that for any two vertices $v, v' \in V'$ there is an edge between two connected subsets V_v and $V_{v'}$ in G if and only if $(v, v') \in E'$. Then G is called a $\mu G'$ and it is written[13] as $G = \mu G'$. The sets V_v are the contraction kernels (called also branch sets [Diestel, 1997]) of $\mu G'$. One can think of obtaining G' from G by contracting every contraction kernel to a single vertex.

[13] $\mu G'$ denotes a whole class of graphs and $G = \mu G'$ means that G belongs to this class.

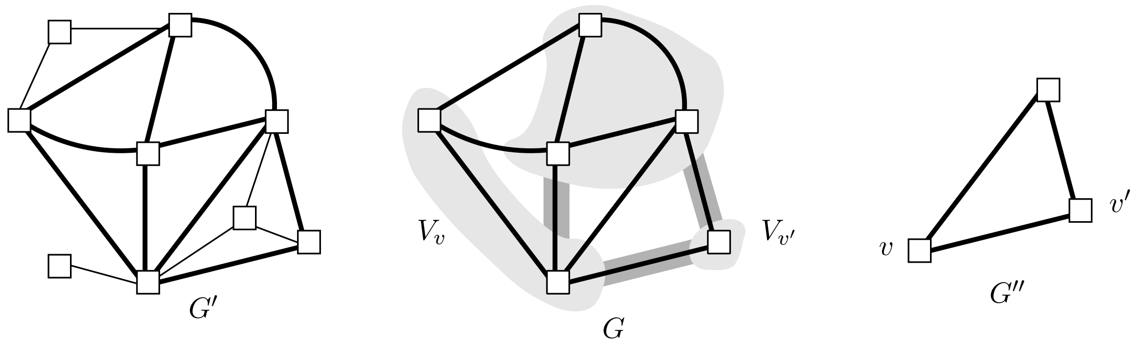


Figure 2.12: $G = \mu G'$ and $G \subseteq G'$, then G'' is minor of G'.

Proposition 2.1 ([Diestel, 1997]) *G is an $\mu G'$ if and only if G is contractible to G', i.e. if there exist graphs $G_0, G_1, ..., G_h$ and edges $e_i \in G_i$ such that $G_0 = G$, $G_h \simeq G'$ and $G_{i+1} = G_i/e_i$ for all $i < h$.*

Intuitively, G is created by repeated contraction and deletion (or vice versa) of edges in the graph G'. If $G = \mu G'$ is a subgraph of another graph G'', then G'' is a *minor* of G' and it is written as $G'' \preccurlyeq G'$. Every subgraph of a graph is its minor, moreover every graph is its own minor. In general any graph derived from another by repeated deletion and contraction (in any order) is its minor [Diestel, 1997]. The Figure 2.12 clarifies the minor relation.

Proposition 2.2 (Partial orderings of minors [Diestel, 1997]) *The minor relation $\preccurlyeq$ is a partial orderings on the class of finite graphs, i.e. they are reflexive, antisymmetric and transitive.*

Proof: Can be found in [Diestel, 1997] □

Note that the minor relation in Proposition 2.1 is defined only for an edge contraction. Moreover this relation is defined for many edge contractions as well, because of the transitivity of the minor relation (Proposition 2.2).

In the previous section it is shown that dual graph contraction consists in repeated contraction and removal operations. The graph, say G_{k+1} at level $k + 1$ is obtained by contracting and removing edges in G_k at level k, thus based on Proposition 2.1 and 2.2 graph G_{k+1} is a minor of G_k, i.e. $G_{k+1} \preccurlyeq G_k$ for all $0 \leq k \leq h$, all higher levels G_k are minors of the base graph G_0.

2.5 Dual Graph Pyramid

A graph pyramid is a pyramid where each level is a graph $G = (V, E)$ consisting of vertices V and of edges E relating two vertices. In order to correctly represent the embedding of the graph in the image plane [Glantz and Kropatsch, 2000] additionally store the dual graph $\overline{G} = (\overline{V}, \overline{E})$ at each level.

In irregular pyramids, each level represents a partition of the pixel set into cells, i.e. connected subsets of pixels. The construction of an irregular image pyramid is iteratively local [Meer, 1989], [Bischof and Kropatsch, 1993], [Jolion, 2003], [Haxhimusa et al., 2002]:

- the cells have no information about their global position,
- the cells are connected only to (direct) neighbors, and
- the cells cannot distinguish the spatial positions of the neighbors.

This means that we use only local properties to build the hierarchy of the pyramid. Usually, on the base level (level 0) of an irregular image pyramid the cells represent single pixels and the neighborhood of the cells is defined by the 4 -connectivity of the pixels. A cell on level $k+1$ (parent) is a union of neighboring cells on level k (children). As shown in Section 2.4 this union is controlled by contraction kernels (decimation parameters). Every parent computes its values independently of other cells on the same level. This implies that an image pyramid is built in $O[log(image\_diameter)]$ parallel steps. Neighborhoods on level $k+1$ are derived from neighborhoods on level k. Two cells c_1 and c_2 are neighbors if there exist pixels p_1 in c_1 and p_2 in c_2 such that p_1 and p_2 are 4-neighbors.

The levels are represented as *dual pairs* $(G_k, \overline{G_k})$ of plane graphs G_k and $\overline{G_k}$. See Section 2.3 for more details on this representation. The sequence $(G_k, \overline{G_k})$, $0 \leq k \leq h$ is called dual *graph pyramid*, where 0 is the base level index and h is the top level index, also called the height of the pyramid. Moreover the graph is attributed, $G = (V, E, attr_v, attr_e)$, where $attr_v : V \rightarrow \mathbb{R}^+$ and $attr_e : E \rightarrow \mathbb{R}^+$ are functions, i.e. content of the graph is stored in attributes attached to both vertices and edges. In general a graph pyramid can be generated bottom-up as follows:

Algorithm 1 – Constructing Dual Graph Pyramid

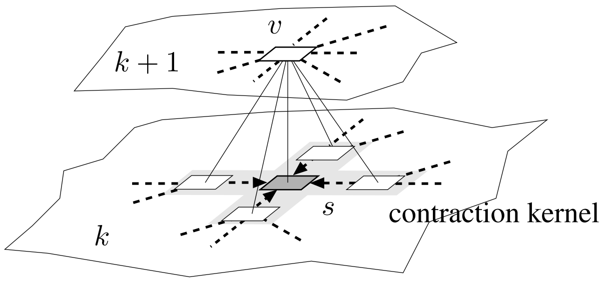
Input: Graphs $(G_0, \overline{G_0})$

1: **while** further abstraction is possible **do**
2: select contraction kernels
3: perform dual graph contraction and simplification of dual graph
4: apply reduction functions to compute content of new reduced level
5: **end while**

Output: Graph pyramid – $(G_k, \overline{G_k})$, $0 \leq k \leq h$.

In the previous section it is shown that G_{k+1} is minor of G_k, i.e. $G_{k+1} \preccurlyeq G_k$ for all $0 \leq k \leq h$, therefore graphs in a pyramid belong to a class of graphs related by a minor relation. One can say that a dual graph pyramid is a set of partial order graphs.

Let the building of the dual graph pyramid be explained by using the simple 5×5 gray value image in Figure 2.3. For the sake of simplicity of the presentation in the figures afterward the dual graphs as well as inter-level relations are not shown explicitly. An example of this inter-level relation is shown in Figure 2.13 with solid lines. In this figure a contraction kernel (shadowed) on level k is shown, with its surviving vertex s (dark shadowed) and its non-surviving vertices (white). After the dual graph contraction a new vertex v on the level k is created as well as the child-parent relations. In this example initially the attributes of the vertices receive the gray values of the pixels, and the cell in the new level (the parents) becomes the gray value of its children.



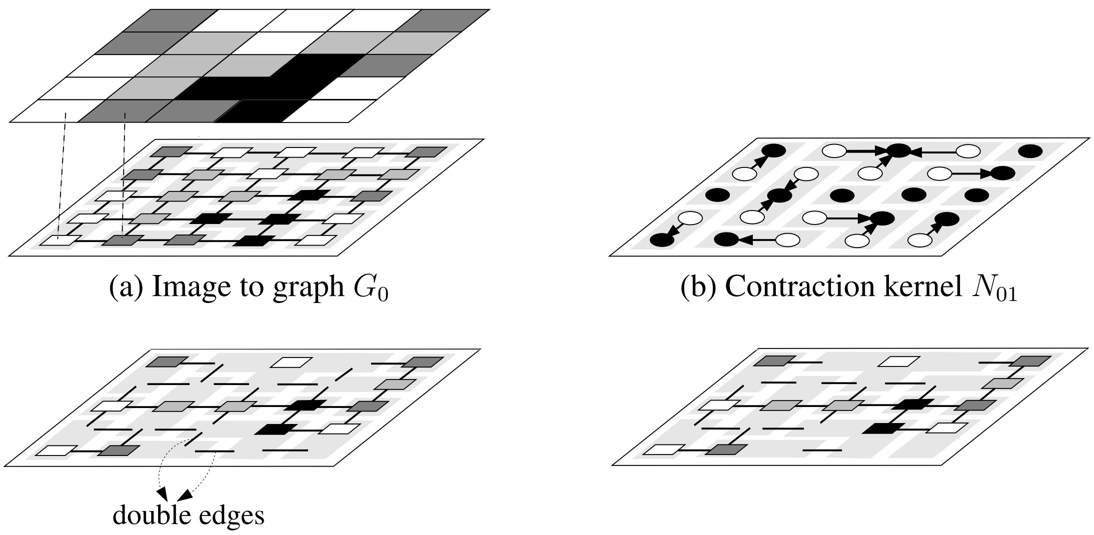
$s \in V_k$ and $v \in V_{k+1}$
Creation of a new vertex v in the level $k+1$ of the pyramid.

Figure 2.13: Parent-child relation.

The first step determines what information in the current top level is important and what can be dropped. A contraction kernel is a (small) sub-tree of the top level the root of which is chosen to survive (black circles in Figure 2.14b. Figure 2.14a shows the window and the selected contraction kernels each shown with gray. Selection criteria in this case contract only edges inside connected components having the same gray value.

All the edges of the contraction trees are dually contracted during step 3. Dual contraction of an edge e (formally denoted by $G/\{e\}$) consists of contracting e and removing the corresponding dual edge $\overline{e}$ from the dual graph (formally denoted by $\overline{G} \setminus \{\overline{e}\}$). This preserves duality and the dual graph needs not be constructed from the contracted primal graph G' at the next level.

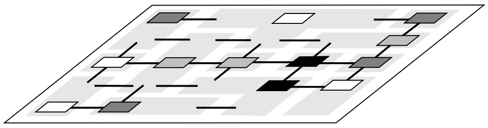
Since the contraction of an edge may yield multi-edges (an example shown with arrow in



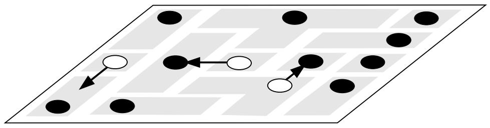
(a) Image to graph G_0 (b) Contraction kernel N_{01}

(c) G' after dual-edge contraction of G_0 (d) G_1 after the removal of redundant edges in G'

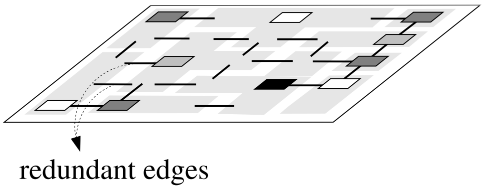
Figure 2.14: Dual graph contraction in G_0 and the creation of the G_1 of the pyramid.



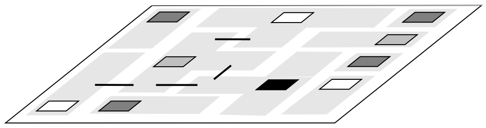
(a) Dually contracted graph G_1



(b) Contraction kernel N_{12}



(c) Edge contraction of graph G_1



(d) Removal of the shown redundant edges

Figure 2.15: Dual graph contraction in G_1 and the resulting G_2.

Figure 2.14c and self-loops there is a second simplification phase of step 3 which removes all redundant multi-edges and self-loops. Note that not all such edges can be removed without destroying the topology of the graph: if the cycle formed by the multi-edge or the self-loop surrounds another part of the data its removal would corrupt the connectivity! Fortunately this can be decided locally by the dual graph since *faces of degree two* (having the double-edge as boundary) and *faces of degree one* (boundary = self-loop) cannot contain any connected elements in its interior. Since removal and contraction are dual operations, the removal of a self-loop or of one of the double edges can be done by contracting the corresponding dual edges in the dual graph, which are not depicted in our example for the sake of the simplicity. The dual contraction of our example remains a simple graph G_1 without self-loops and multi-edges (Figure 2.15a).

Step 3 generates a reduced pair of dual graphs. Their contents is derived in step 4 from the level below using the reduction function. In our example reduction is very simple: the surviving vertex inherits the color of its son.

The resulted graph G_2 of another dual contraction step is shown in Figure 2.16. The selection rules and the reduction function are the same as in the first iteration. The result shows that the regions with the same color are brought together. This fact could be used in a top-down verification step which checks the reliability of merging criterion in the more general context. Figure 2.17 depicts an overview of the results produced by the algorithm, by using a simple merging criterion, in which vertices having the same attributes (gray value) are merged. Moreover in this figure, contraction kernels and the equivalent contraction kernel are shown.

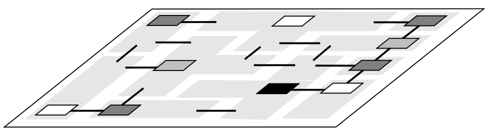


Figure 2.16: Graph G_2 after two steps of dual graph contraction.

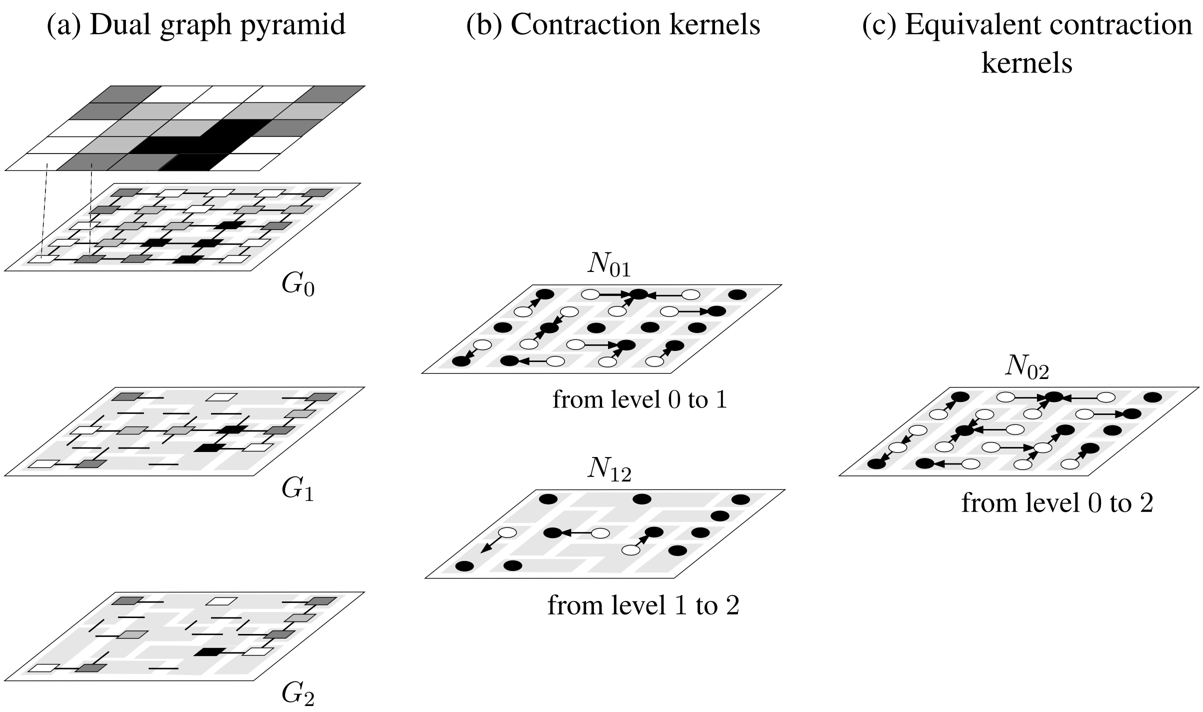


Figure 2.17: Dual graph pyramid with three levels: G_0, G_1, and G_2.

Table 2.1: Graph contraction parameters [Kropatsch, 1994]

Level	representation	contract / remove	conditions
0	$(G_0, \overline{G}_0)$		
	$\downarrow$	contraction kernel $N_{0,1}$	forest, depth 1
	$(G_0/N_{0,1}, \overline{G}_0 \setminus \overline{N}_{0,1})$		
	$\downarrow$	redundant multi-edges, self-loops	$\deg \overline{v} \leq 2$
1	$(G_1, \overline{G}_1)$		
	$\downarrow$	contraction kernel $N_{1,2}$	forest, depth 1
	...	...	...

By contracting the edges of the equivalent contraction kernel $N_{0,2}$ one can reach the top of the pyramid G_2 directly from the base G_0. A more complex merge criterion is shown in Chapter **??**. The following Table 2.1 summarizes dual graph contraction in terms of the control parameters used for abstraction and the conditions to preserve topology (from [Kropatsch, 1994]).

In this section every level of the dual irregular pyramid is presented explicitly. An implicit representation can be done if vertices and edges of ECK of the apex $(V_0, N_{0,h})$ are labeled, such that vertices and edges are labeled with the highest level in which they survive. This labeled spanning tree uniquely defines the structure of the dual irregular pyramid [Kropatsch, 1995b]. If the plane graph is transformed into a combinatorial map the transcribed operations form the combinatorial pyramid [Brun and Kropatsch, 2001a, Brun and Kropatsch, 2001b]. This framework allows to link dual graph pyramids with topological maps (especially combinatorial maps) which enable to extend the scope to three (or more) dimensions.

2.6 Summary

In order to express the connectivity or other geometric or topological properties the image representation must be enhanced by a neighborhood relation. The neighborhood relation can be represented explicitly by a *graph* consisting of vertices corresponding to the sampling points and of edges connecting neighboring vertices. To represent not only neighborhood relations but also the inclusion relations, the planarity and the duality concepts come very good in hand. Since every planar graph has a dual, one can use dual graphs to represent the partitioning of the (image) plane, encoding the neighborhood and inclusion relations. One can define transformation of these graphs, like the dual graph contraction to simplify graphs in the sense of less vertices and edges. Edge contraction and removal introduces naturally a hierarchy of dual graphs, the dual graph pyramid.

Bibliography

[Ahronovitz et al., 1995] Ahronovitz, E., Aubert, J.-P., and Fiorio., C. (1995). The star-topology: A topology for image analysis. In Ahronovitz, E. and Fiorio, C., editors, 5^{th} *Discrete Geometry for Computer Imagery*, pages 107–116. 21

[Bederson, 1992] Bederson, B. B. (1992). *A Miniature Space-variant Active Vision System.* PhD thesis, New York University, Courant Insitute, New York. 9, 13

[Bischof, 1995] Bischof, H. (1995). *Pyramidal Neural Networks.* Lawrence Erlbaum Associates. 5, 6, 7, 9

[Bischof and Kropatsch, 1993] Bischof, H. and Kropatsch, W. G. (1993). Hopfield networks for irregular decimation. In Pölzleitner, W. and Wenger, E., editors, *Image Analysis and Synthesis*, volume 68, pages 317–327. OCG-Schriftenreihe, Österr. Arbeitsgemeinschaft für Mustererkennung, R. Oldenburg. 30

[Bister et al., 1990] Bister, M., Cornelis, J., and Rosenfeld, A. (1990). A critical view of pyramid segmentation algorithms. *Pattern Recognition Letters*, 11(9):605–617. 4, 8

[Braquelaire and Brun, 1998] Braquelaire, J.-P. and Brun, L. (1998). Image segmentation with topological maps and interpixel representation. *Journal of Visual Communication and Image Representation*, 9(1):62–79. 18

[Brun and Kropatsch, 2001a] Brun, L. and Kropatsch, W. G. (2001a). Contraction kernels and combinatorial maps. In Jolion, J.-M., Kropatsch, W. G., and Vento, M., editors, *Proceedings of IAPR Workshop on Graph-based Representations in Pattern Recognition*, pages 12–21. CUEN. 35

[Brun and Kropatsch, 2001b] Brun, L. and Kropatsch, W. G. (2001b). Introduction to combinatorial pyramids. In Bertrand, G., Imiya, A., and Klette, R., editors, *Digital and Image Geometry*, pages 108–128. Springer, Berlin, Heidelberg. 22, 35

[Brun and Kropatsch, 2003] Brun, L. and Kropatsch, W. G. (2003). Contraction kernels and combinatorial maps. *Pattern Recognition Letters*, 24(8):1051–1057. 22

[Burt and Adelson, 1983] Burt, P. J. and Adelson, E. H. (1983). The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540. 11, 27

[Burt et al., 1981] Burt, P. J., Hong, T.-H., and Rosenfeld, A. (1981). Segmentation and estimation of image region properties through cooperative hierarchical computation. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(12):802–809. 4

[Cormen et al., 2001] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*. MIT Press, Massachusetts Institute of Technology. 27

[Diestel, 1997] Diestel, R. (1997). *Graph Theory*. Springer, New York. 15, 17, 18, 29, 30

[Glantz and Kropatsch, 2000] Glantz, R. and Kropatsch, W. G. (2000). Plane embedding of dually contracted graphs. In Borgefors, G., Nyström, I., and Sanniti di Baja, G., editors, *Proceedings of Discrete Geometry for Computer Imagery*, volume 1953 of *Lecture Notes in Computer Science*, pages 348–357, Uppsala, Sweden. Springer, Berlin Heidelberg, New York. 30

[Granlund, 1999] Granlund, G. H. (1999). The complexity of vision. *Signal Processing*, 74(1):101–126. 3, 13, 14

[Haralick and Shapiro, 1993] Haralick, R. M. and Shapiro, L. G. (1993). *Computer and Robot Vision.*, volume II. Addison Wesley. 14

[Harary, 1969] Harary, F. (1969). *Graph Theory*. Addison Wesley. 16, 17

[Haxhimusa et al., 2002] Haxhimusa, Y., Glantz, R., Saib, M., Langs, G., and Kropatsch, W. G. (2002). Logarithmic tapering graph pyramid. In van Gool, L., editor, *Proceedings of German Pattern Recognition Symposium*, volume 2449 of *Lecture Notes in Computer Science*, pages 117–124, Switzerland. Springer. 30

[Horowitz and Pavlidis, 1976] Horowitz, S. and Pavlidis, T. (1976). Picture segmentation by a tree traversal algorithm. *Journal of the Association for Computer and Machinery.*, 2(23):368–388. 2

[Jolion, 2003] Jolion, J.-M. (2003). Stochastic pyramid revisited. *Pattern Recognition Letters*, 24(8):1035–1042. 30

[Jolion and Rosenfeld, 1994] Jolion, J.-M. and Rosenfeld, A. (1994). *A Pyramid Framework for Early Vision*. Kluwer. 2, 3, 4, 5, 6, 8, 10

[Kelly, 1970] Kelly, M. D. (1970). Edge detection in pictures by computer using planning. In Meltzer, B. and Michie, D., editors, *Machine Intelligence*, volume 6, pages 397–409, Edinburgh Scotland. Edinburgh University Press. 2

[Kovalevsky, 1989] Kovalevsky, V. A. (1989). Finite topology as applied to image analysis. *Computer Vision, Graphics, and Image Processing*, 46:141–161. 20

[Kovalevsky, 1993] Kovalevsky, V. A. (1993). Digital Geometry Based on the Topology of Abstract Cellular Complexes. In Chassery, J.-M., Francon, J., Montanvert, A., and Réveillès, J.-P., editors, *Géometrie Discrète en Imagery, Fondements et Applications*, pages 259–284, Strasbourg, France. 21

[Kropatsch, 1986] Kropatsch, W. G. (1986). Curve representations in multiple resolutions. In *Proceedings 8th International Conference on Pattern Recognition*, pages 1283–1285. IEEE Comp.Soc. 10, 11

[Kropatsch, 1991] Kropatsch, W. G. (1991). Image Pyramids and Curves - An Overview. Technical Report PRIP-TR-2, Vienna University of Technology, Faculty of Informatics, Institute of Computer Aided Automation, Pattern Recognition and Image Processing Group, http://www.prip.tuwien.ac.at/ftp/pub/publications/trs/. 3, 8, 10

[Kropatsch, 1994] Kropatsch, W. G. (1994). Building irregular pyramids by dual graph contraction. Technical Report PRIP-TR-35, Vienna University of Technology, Faculty of Informatics, Institute of Computer Aided Automation, Pattern Recognition and Image Processing Group, http://www.prip.tuwien.ac.at/ftp/pub/publications/trs/. 18, 19, 20, 21, 22, 23, 25, 27, 28, 34

[Kropatsch, 1995a] Kropatsch, W. G. (1995a). Building irregular pyramids by dual graph contraction. *IEE-Proc. Vision, Image and Signal Processing*, 142(6):366–374. 18, 23, 24, 25

[Kropatsch, 1995b] Kropatsch, W. G. (1995b). Equivalent contraction kernels and the domain of dual irregular pyramids. Technical Report PRIP-TR-42, Vienna University of Technology, Faculty of Informatics, Institute of Computer Aided Automation, Pattern Recognition and Image Processing Group, http://www.prip.tuwien.ac.at/ftp/pub/publications/trs/. 25, 28, 34

[Kropatsch, 1999] Kropatsch, W. G. (1999). How useful is structure in motion? In D. Chetverikov and T. Szirányi, editor, *Fundamental Structural Properties in Image and Pattern Analysis*. OCG-Schriftenreihe, Österr. Arbeitsgemeinschaft für Mustererkennung, R. Oldenburg. 3

[Kropatsch, 2002] Kropatsch, W. G. (2002). Abstraction pyramids on discrete representations. In Braquelaire, A. J.-P., Lachaud, J.-O., and Vialard, A., editors, *Proceedings of Discrete Geometry for Computer Imagery*, pages 1–21, Bordeaux, France, April 3-5, 2002. Springer. 13

[Kropatsch and Burge, 1998] Kropatsch, W. G. and Burge, M. (1998). Minimizing the topological structure of line images. In Amin, A., Dori, D., Pudil, P., and Freeman, H., editors, *Proceedings of Joint IAPR International Workshops on Syntactic and Structural Pattern Recognition and Statistical Patern Recognition*, volume 1451 of *Lecture Notes in Computer Science*, pages 149–158, Sydney, NSW, Australia, August 11-13, 1998. Springer. 10, 11

[Kropatsch et al., 1999] Kropatsch, W. G., Leonardis, A., and Bischof, H. (1999). Hierarchical, adaptive and robust methods for image understanding. *Surveys on Mathematics for Industry*, 9:1–47. 4, 5, 6, 9, 13, 14

[Kropatsch and Montanvert, 1991] Kropatsch, W. G. and Montanvert, A. (1991). Irregular versus regular pyramid structures. In Eckhardt, U., Hbler, A., Nagel, W., and Werner, G., editors, *Geometrical Problems of Image Processing*, pages 11–22. Springer. 10

[Lienhardt, 1989] Lienhardt, P. (1989). Subdivisions of n-dimensional spaces and n-dimensional generalized maps. In Mehlhorn, K., editor, *Proceedings of the 5th Annual Symposium on Computational Geometry*, pages 228–236, Saarbrücken. ACM Press. 22

[Lienhardt, 1991] Lienhardt, P. (1991). Topological models for boundary representation: a comparison with n-dimensional generalized maps. *Computer-Aided Design*, 23(1):59–82. 22

[Lindeberg, 1990] Lindeberg, T. (1990). Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):234–254. 2

[Mallat, 1989] Mallat, S. G. (1989). A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693. 2

[Marchadier et al., 2003] Marchadier, J., Kropatsch, W. G., and Hanbury, A. (2003). Homotopic transformations of combinatorial maps. In Nystrm, I., di Baja, G. S., and Svensson, S., editors, *Proceedings of Discrete Geometry for Computer Imagery*, volume 2886 of *Lecture Notes in Computer Science*, pages 134–143, Naples, Italy. Springer. 29

[Meer, 1989] Meer, P. (1989). Stochastic image pyramids. *Computer Vision, Graphics, and Image Processing*, 45(3):269–294. Also as UM CS TR-1871, June, 1987. 10, 30

[Munkres, 1993] Munkres, J. R. (1993). *Elements of Algebraic Topology*. Addison-Wesley. 20

[Rojer and Schwartz, 1990] Rojer, A. and Schwartz, E. L. (1990). Design considerations for a space variant visual sensor complex-logarithmic geometry. In *Proceeding of Internation Conference in Pattern Recognition*, volume 2, pages 278–285, Los Alamitos, California. IEEE Computer Society Press. 13

[Rosenfeld, 1982] Rosenfeld, A. (1982). Quadtrees and pyramids: Hierarchical representation of images. Technical Report TR-1171, University of Maryland, Computer Science Center. 4

[Rosenfeld, 1984] Rosenfeld, A., editor (1984). *Multiresolution Image Processing and Analysis*. Springer Verlag. 3, 4

[Rosenfeld, 1985] Rosenfeld, A. (1985). Arc colorings, partial path groups, and parallel graph contractions. Technical Report TR-1524, University of Maryland, Computer Science Center. 10

[Rosenfeld and Thurston, 1971] Rosenfeld, A. and Thurston, M. (1971). Edge and curve detection for visual scene analysis. *IEEE Transactions on Computers*, 20(12):562–569. 2

[Samet, 1990] Samet, H. (1990). *Applications of Spatial Data Structures*. Addison Wesley Publishers. 8

[Serra, 1982] Serra, J. (1982). *Image Analysis and Mathematical Morphology*, volume I. Academic Press. 11, 29

[Soille, 1994] Soille, P. (1994). *Morphological Image Analysis*. Springer Verlag. 4, 29

[Sonka et al., 1999] Sonka, M., Hlavac, V., and Boyle, R. (1999). *Image Processing, Analysis and Machine Vision*. Brooks/Cole Publishing Company. 4

[Stelldinger and Ullrich, 2005] Stelldinger, P. and Ullrich, K. (2005). Towards a general sampling theory for shape preservation. *Image and Vision Computing*, 23(10):237–248. 4

[Tanimoto and Pavlidis, 1975] Tanimoto, S. and Pavlidis, T. (1975). A hierarchical data structure for picture processing. *Computer Graphics and Image Processing*, (4):104–119. 2

[Thulasiraman and Swamy, 1992] Thulasiraman, K. and Swamy, M. N. S. (1992). *Graphs: Theory and Algorithms.* Wiley-Interscience. 15, 16, 17

[Wallace et al., 1994] Wallace, R. S., Ong, P.-W., Bederson, B. B., and Schwatz, E. L. (1994). Space variant image processing. *International Journal of Computer Vision*, 13(1):71–90. 13

[Willersinn, 1995] Willersinn, D. (1995). *Dual Irregular Pyramid.* PhD thesis, Vienna University of Technology, Faculty of Informatics, Institute of Computer Aided Automation, Pattern Recognition and Image Processing Group, Vienna. 25

[Witkin, 1986] Witkin, A. P. (1986). Scale space filtering. In Pentland, A. P., editor, *From Pixels to Predicates: Recent Advances in Computational and Robot Vision*, pages 5–19. Ablex, Notwood, NJ. 2

[Yuille and Poggio, 1986] Yuille, A. L. and Poggio, T. A. (1986). Scaling theorems for zero crossings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):15–25. 11

List of Symbols and Abbreviations

If not stated otherwise in the text the symbol and abbreviations applies throughout the document.

Abbreviations

DGC	Dual Graph Contraction
ECK	Equivalent contraction kernel

Symbols

$N_{k,k+1}$	contraction kernel from k to $k+1$
$\overline{G}$	dual graph
$\overline{E}$	dual edge set
$\overline{e}$	dual edge
F	face set
f	face
$(G_k, \overline{G}_k)$	pair of dual graphs
G_k	graph at level k
h	height of pyramid
k	level k of the pyramid
$\mathbb{R}$, $\mathbb{Z}$	number sets
$\widetilde{G}$	plane graph
λ	reduction factor
$\overline{V}$	dual vertex set
$\overline{v}$	dual vertex

Index

The first page number is usually, but not always, the primary reference to the indexed topic.