Kropatsch2024a

© 2025 World Scientific Publishing Company
https://doi.org/10.1142/9789811289125_0002

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Chapter 2

Controlling Topology Preserving Graph Pyramids

Walter G. Kropatsch*, Majid Banaeyan*, and Rocio Gonzalez-Diaz†

**TU Wien, Vienna, Austria*
† University of Seville, Seville, Spain

Abstract

Since the beginning of the use of pyramidal structures for processing images some 40 years ago, several different operations have been applied for a large variety of different applications. The basic advantage of the pyramids is the progressive reduction of the data, level by level by a reduction factor that limits the pyramid's height to the logarithm of the diameter of the base level. Differently from the classical (Gaussian) pyramids, we focus on pyramids where the basic data structure is not an array but a graph structure embedded in the image space. In this chapter, we target (1) topological issues of objects in images like holes in a region, (2) what operations can be used to propagate image information from the input to the high levels as well as in the opposite direction, (3) what specific properties can be generalized by what operations, (4) how to achieve the logarithmic computational complexity, and last but not least, (5) how to coordinate the different processes. In the second part, we focus on a new type of pyramid, the LBP pyramid, that uses a variant of the local binary patterns to recognize critical points and contracts lowest contrast edges during the bottom-up phase. Not only the topology among the relevant parts of the image is preserved, but, as experiments have shown, it also allows the reconstruction of images with only a few colors that are often hard to distinguish from the original.

Kropatsch2024a

2.1 Introduction

In this chapter, we describe a hierarchical structure that has its origins in the classical image pyramids like Gaussian or Laplacian pyramids but with the big advantage that the data structure for the individual levels of the pyramid are no more rigid grids or arrays but are based on planar embedded graphs. This enables the pyramid to adapt its structure to the needs of the data: Parts of the data that are considered important for the processing can survive to higher levels while redundancies in the data like homogeneous regions can be reduced during the bottom-up construction phase. Graphs are used here because they are widely known as versatile data structures although the proper representation of topological relations needs the dual graphs. However, there are other less known data structures like combinatorial maps [1] or generalized maps [2] or cell complexes [3,4] that can replace the graphs in the pyramid.

All these data structures do not only describe the topological arrangement of the data but can also describe the complex arrangements of semantic objects of different sizes and shapes that should appear as results of the processing. An important issue in dealing with the huge amounts of data is the possibility to process them in a massive parallel way to reach reasonable processing efficiency. A requirement for parallelism is the independence of the operations such that the result does not depend on the order of the applied operations.

Controlling the big variety of possible choices in the general concept of irregular pyramids is one of the main issues of this chapter. There are several choices in the bottom-up construction of the pyramids but also parameters that have an influence on the abstract concepts surviving to the higher levels need to be chosen or even optimized and adapted to the data. But not only the bottom-up processes are important, irregular pyramids allow also a top-down expansion process that provides an insight into the visual information at the higher levels by, i.e., visualization but also enable to better tune the repeated bottom-up processes with a better overview of what objects with known properties are where in the input. In such cases, attention could be put on particular object details in a

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

repeated bottom-up phase. These up and down phases also provide explanations of what the pyramid has found in its higher levels.

At some places, we also relate the presented concepts to the very popular methods of machine learning (ML) and artificial intelligence (AI). There are differences but also similarities, advantages as well as drawbacks. We see the irregular pyramids not as a competitor of the ML approaches but see possibilities for fruitful combinations.

Section 2.2 starts with five subsections giving motivations and some background of the presented concepts. It follows a recall on irregular pyramids in Section 2.3, the processes for propagating data from the base level to the top as well as in the opposite direction, the expansion from top down to the base. Section 2.4 discusses the many options for controlling the processes, the tasks, and the properties that have been explored in different applications (Section 2.4.5).

In Section 2.5, we adapt the local binary patterns (LBP) to the basic data structure: the graphs, and study their relation to the critical points of curves and surfaces. Monotonic paths, curves, and profiles through an image show that these 1D manifolds have invariant LBPs. On this basis, we construct the LBP pyramid (Section 2.6) and show reconstructions with only a small percentage of the original input. These reconstructions are visually difficult to distinguish from the original data. We draw the conclusion that the structure of the critical points and their adjacencies extracted by the LBP pyramid is extremely important, while the actual gray levels or colors of the image are visually less relevant. This raises the question about the space between the critical points (Section 2.6). We give a simple definition of the concept of a 'slope' with several interesting properties leading to future directions of research addressed in the conclusion.

2.2 Motivations and Background

In this section, we mention five different motivations for the use of irregular pyramids. We start with some requirements described by Leonard Uhr, 1986. We continue with some facts about biological plausibility often used as arguments to justify approaches in

Kropatsch2024a

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

recent AI. We then shortly mention a recent project: There we study biological images with extremely high resolutions. The next motivation addresses the problem that not all problems can be solved by the same architecture. Psychology has identified so-called insight problems that cannot be solved by simply optimizing a universal architecture. Finally, we shortly summarize some crucial insights of a seminal paper by Jan Koenderink [6]. They gave us the strong motivation for the research presented in this chapter.

2.2.1 *The problem of biological perception*

Leonard Uhr [5] summarized the problem of human visual perception in 1986 with a few facts and some conclusions: Each human eye has about 10^7 cones and 10^8 rods sensing the light entering the human eye. The measured intensities and frequencies are processed by a large number of synapses where each one takes about 1.5 μs allowing about 1000 serial operations in one second. In order to "see" and to accurately react on the visual stimuli, no more than 600 serial steps are available. This can be achieved by the human brain only by massive parallel processes that converge in logarithmic complexity toward the location where decisions are taken.

Leonard Uhr proposed pyramidal data structures as the only chance to solve the vision problem. But he also clarified that "*pyramids are not (only) multiresolution, parallel bottlenecks, low level, array processors, or trees.*" He further stated that "*a pyramid needs augmentation*" and "*... any connected (data-flow) graph could be used.*" Furthermore, they need to "*combine bottom-up and top-down*" processes to solve the complex vision problems.

2.2.2 *The human retina is irregular*

Most neural network architectures claim biological plausibility. This is partly true for the general functionality of the signal processing (weighted averages and activation functions), but it certainly does not apply to the underlying architecture: Both the sensors for the visual input as well as the many other sensors providing valuable input for the information processing of the human brain are not regular grids in contrast to most of the artificial neural networks that are currently popular. Figure 2.1 shows a small segment of the

Kropatsch2024a

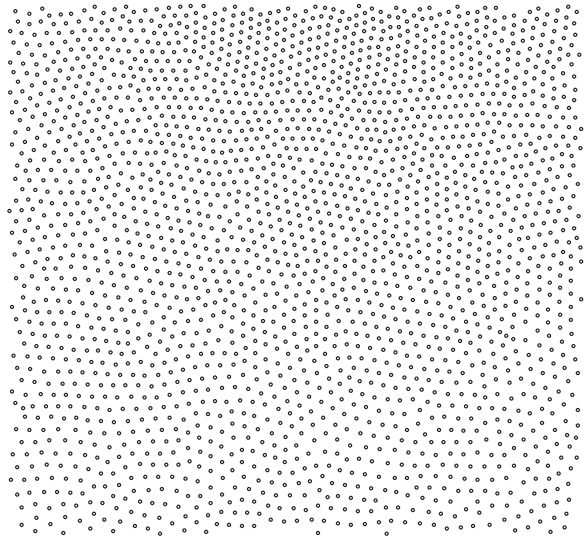


Fig. 2.1. A section of rods and cones in the retina.

retina of a monkey's eye.[a] It is very similar to the human retina and it is clearly not an array! The natural arrangement of sensors in the human eye needs data structures such as graphs to properly represent the irregular embeddings and to learn more about the benefits of these irregular sensor arrangements, in particular the relationship to saccadic eye movements that certainly are not just an accident of nature but may have a considerable importance for the reliable processing of noisy visual data.

2.2.3 *Project: Water's gateway to heaven*

This research project[b] that our group started in 2020 together with two groups in biology raised some very essential problems typical for the trend to use extremely high resolutions and also temporal changes in three dimensions (Figure 2.2).

The project studies 3D imaging and modeling of transient stomatal responses in plant leaves. Input to these studies are

[a]Data of the monkey's retina have been gratefully provided by Peter Ahnelt.
[b]https://waters-gateway.boku.ac.at/.

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

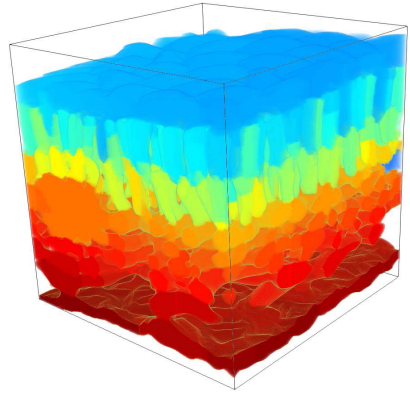


Fig. 2.2. 3D μCT image with color labels.

high-resolution X-ray micro-tomography (μCT) and fluorescence microscopy images. μCT images have the challenging dimensions ranging up to $2000 \times 2000 \times 2000 \approx 2^{33}$ voxels and are taken at 2–4 instances of time. Visible objects are different cells, water ways, and the airspace in between. Leaves are not rigid but to a certain extent deformable. Consequently, rigid matching may not work so well when comparing different images of the same specimen, in particular if the concentration of water is different in the two acquisitions.

The main goal in this project is to **understand** the causality of opening and closing of the stomata. These are cells that can open to allow gases to enter (e.g., CO_2) for **photosynthesis** and water to leave.

The huge amount of data and the complexity of the models describing the processes require a very efficient processing of the data. We are confident that pyramids provide the requested performance.

2.2.4 *Critical/stationary points are relevant*

Jan Koenderink [6] draws some important conclusions in his seminal contribution "The Structure of Images" (1984). He considers intensity images as a function in three-dimensional space $\Phi(x, y, t)$, where (x, y) are the spatial coordinates and t is the scale dimension. He considers the scale as generated by convolution with a Gaussian kernel $\Phi(x, y, 0) * G(t)$. The **Diffusion** $\Delta\Phi = \Phi_t$ is the basis for his

Downloaded from www.worldscientific.com by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

scale space theory. He requests that "*Any feature at a coarse resolution is required to possess a 'cause' at finer resolution.*" He considers stationary (critical) points by setting the spatial derivatives to zero: $\Phi_x = \Phi_y = 0$. Among those satisfying these constraints, the Hessian distinguishes between the different critical points:

$$\Phi_{xx}\Phi_{yy} - \Phi_{xy}^2 \geq 0 \text{ for extrema and} \tag{2.1}$$

$$\Phi_{xx}\Phi_{yy} - \Phi_{xy}^2 < 0 \text{ for a saddle point.} \tag{2.2}$$

We shall find a solution in Section 2.5.4 for both decisions **without the noise-sensitive partial derivatives**. A particular observation of Jan Koenderink could be verified after the new identification of critical points: *Extrema and saddle points disappear pairwise when t increases*. It turns out to be useful to eliminate pairs of critical points that are not persistent (i.e., very close peaks with similar height separated by a saddle not much below the peaks).

2.2.5 *An insight problem*

In this fifth motivation, we discuss the limitations of solutions found by optimization processes. In his book, Pizlo [7] demonstrates impressively that there exist problems that cannot be solved simply by optimization (the most frequent strategy for most machine learning approaches). He gives a very simple example:

Create n equilateral triangles ($\triangle$) with m matchsticks:

(1) Create one triangle with three matchsticks (Figure 2.3). This has the obvious solution in Figure 2.3.
(2) Make two triangles with two more matchsticks (Figure 2.4).
(3) Can you produce four triangles with one more matchstick?

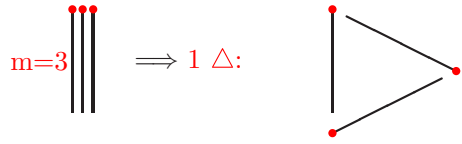


Fig. 2.3. Three match sticks form one triangle.

Kropatsch2024a

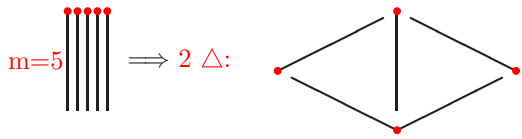


Fig. 2.4. Five matchsticks form two triangles.

For the solution, consider the Euler–Poincaré characteristic to balance the number of points (•), the number of matchsticks (m), and the number of triangles (△):

Euler–Poincaré characteristic

	#P	-	#E	+	#F	= 1
Case	•	-	m	+	△	= 1
1.	3	-	3	+	1	= 1
2.	4	-	5	+	2	= 1
3.	?	-	6	+	4	= 1

The last case would suggest that the solution has three points that seems impossible. This is the characteristic of an 'insight problem': An 'insight problem' is typically difficult to solve.

Reference [7] shows an elegant solution with a change in representation: "*If you exclaim 'aha!' at the moment the solution suddenly occurs to you, you had an insight.*" Once the solution strategy is understood, it is easy to explain. However, the above Euler–Poincaré characteristic shows that the optimization would not find a proper solution. A similar reasoning could be applied to several machine learning solutions.

2.3 Recall on Irregular Pyramids

The irregular pyramid consists of a stack of graphs with decreasing size. Each graph of this stack is called a level of the pyramid and the lowest level is the base graph corresponding to the input image where pixels correspond to the graph's vertices and two vertices are joined by an edge if the corresponding pixels are 4-connected. This base graph is also called the neighborhood graph $G(V, E)$ of the image. 4-neighborhood is preferred since edges between diagonal neighbors

Downloaded from www.worldscientific.com by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

of 2×2 pixels would intersect, with the consequence that the 8-connected graph is not planar. The pixel value is an attribute to the corresponding vertex and it can range from a single gray value to a vector of either spectral channels or additional information like filter responses, lengths, and distances. In order to properly describe the embedding in the image plane, we use the dual graph $\overline{G} = (\overline{V}, \overline{E})$ that is implicitly given by the embedding of the image. The dual vertices $\overline{V}$ identify the face formed by any 2×2 block of pixels and the dual edges $\overline{E}$ correspond to the boundary segment between any two adjacent pixels.

2.3.1 *Extended region adjacency graph*

Image segmentation typically assigns each pixel a label identifying the set of pixels having the same or a similar property. The adjacencies of these regions are typically described by the region adjacency graph (RAG) where each vertex represents a connected set of pixels with the same label and two vertices are connected in the RAG if two regions with two different labels share a common boundary.

Most approaches consider the RAG as a simple graph without multiple edges and without self-loops. But the simple graph cannot describe all the topological configurations that these regions can be related to in practice: The left and right riverbanks of a river may be connected by more than one bridge. The simple RAG just states that the two riverbanks are connected but not by how many bridges. This can be resolved by simple RAGs by sub-dividing each riverbank into as many segments as there are bridges. This is not only increasing the size of the graph, but it is also difficult to handle since the characteristic features of the segments may be similar if not identical such that they cannot be easily classified.

A second example where the simple RAG has problems describes the relationship between a lake and its islands. The outer boundary of the lake is a closed curve and each island is also bounded by a closed curve: In a typical inclusion relationship, the islands are completely surrounded by the lake. Let us describe the mainland with a vertex of the RAG, the lake with a vertex, and each island also with a vertex. Clearly, the mainland is connected to the lake and the lake is connected to each of its islands. But what expresses the fact that the lake surrounds all islands? One solution is to introduce a separate

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

data structure, an inclusion tree. It works in 2D, but what about a tunnel in 3D?

We found the extension of the simple RAG, a good solution to solve both problems: The multiple bridges can be represented by multiple edges without the need to arbitrarily sub-divide the homogeneous riverbanks and self-loops that surround the islands can represent the inclusion relation. To distinguish the more frequently used RAG from the non-simple RAG, we denote the extended version by E-RAG.

2.3.2 *Overview of the bottom-up construction*

Figure 2.5 gives an overview of irregular pyramids. The base level is the 4-neighborhood graph of the image and each level above the base represents an E-RAG $G = (V, E)$.

The next higher level is reached by contracting selected edges while preserving certain relevant points. They form the **contraction kernel**. The smaller graph contains less vertices and less edges, but some edges have become multiple and some even self-loops. Therefore, the next step is to **simplify** the graph from unnecessary multi-edges and self-loops. Before repeating the contraction, the attributes of the newly generated, smaller graph need to be derived by **reduction functions** taking as input the receptive field of each surviving vertex and edge and computing the attributes of the elements of the higher-level graph. Then, this process can be repeated until a **termination criterium** is satisfied and the apex of the pyramid is reached. The overall process is controlled by the following steps:

- the selected contraction kernel,
- the simplification process,
- the reduction function, and
- the termination criterium.

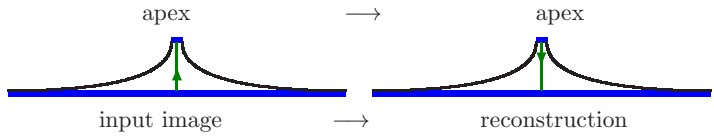


Fig. 2.5. Bottom-up and top-down processes in an irregular pyramid.

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

If the reduction process reduces the graph from level to level by a constant **reduction factor** ≥ 2, then the height of the pyramid is bound by the logarithm of the diameter of the base graph. This contributes to the efficiency of the pyramid when the level by level processing can be massively parallel (compare with Uhr [5] and Section 2.2.1).

2.3.3 *Overview of the top-down reconstruction*

The apex graph of the pyramid is a very abstract representation of the visual entities of the image and their spatial and topological relations. For the purpose of explaining what has been derived from the given input image, the high levels can be successively down-projected to the lower levels and to the base in order to show the entities that have been derived above. For this purpose, we keep some information about the bottom-up process that enables then to reverse the construction and to propagate downwards the insights gained at the higher levels hopefully explaining what and why certain entities have been found.

The basis for the reconstruction is the canonical representation of Torres and Kropatsch [8]. It stores the contraction kernels and simplification parameters in chronological order together with links that enable to undo edge contraction by edge decontraction, edge removal by edge reinsertion, and the attributes at input for the reduction function.

The following subsections introduce more details about these processes with the purpose of showing some interesting properties.

2.3.4 *Contracting an edge*

Definition 2.1 (Edge contraction). The operation of contracting an edge $e = (v_1, v_2) \in E, v_1 \neq v_2 \in V$, of a graph $G = (V, E)$ consists in first identifying the two end points $v_1 \mapsto v_s, v_2 \mapsto v_s$ of the edge e into a new 'surviving' vertex $v_s \in \{v_1, v_2\}$ and replacing v_1, v_2 in all edges by v_s. Finally, the edge e is removed.

The graph after the contraction of edge e has one less edge and one less vertex: $G' = G/e = (V \setminus \{v_1, v_2\} \cup \{v_s\}, E \setminus \{e\})$. Note that the condition $v_1 \neq v_2$ excludes self-loops (v, v) from being contracted.

Kropatsch2024a

Contraction preserves the connectivity of G in G'. As the dual operation of contracting an edge in G/e, the corresponding dual edge is removed from $\overline{G}$: $\overline{G'} = \overline{G} \setminus \overline{e}$. Consequently, the dual graph of G' needs only the dual operation applied to $\overline{G}$ and the duality is preserved.

As a result of contraction, G' may contain parallel edges and even self-loops. Most of them can be removed in the successive simplification step in Section 2.3.6. The remaining parallel edges and self-loops identify special topological properties like the inclusion of holes.

Independent edges can be contracted simultaneously in parallel. All edges that are simultaneously contracted form a **contraction kernel**.

2.3.5 *Contraction kernel*

In order to be able to execute many contractions in parallel (with many processors), they must be independent of each other. In other words, the order in which the set of edges is contracted should not affect the result. Several methods have been used to create contraction kernels (CK) with independent edges: maximal independent vertex set (MIS), Meer [9], maximal independent edge set (MIES), and maximal independent, directed edge set (MIDES), Kropatsch *et al.* [10] with different properties and advantages.

Definition 2.2 (Contraction kernel). Let $G(V, E)$ be the input graph to be contracted. A contraction kernel $K \subset E$ is a subset of edges that forms a spanning forest of G. Each tree of the forest contains one surviving vertex; in some extreme cases, the tree can even be a single (surviving) vertex.

There may be different criteria (examples are given in Section 2.4) for selecting concrete edges to contract and for selecting vertices to survive. The surviving vertices are the vertices of the next pyramid level; the edges are the result of the contraction processes. If each connected component of the contraction kernel covers at least two vertices of V_n, the number of vertices V_{n+1} will be less than $|V_n|/2$. We call this the reduction factor of 2. Isolated vertices can be compensated by larger trees in different parts of the graph. Both selection methods, MIES and MIDES, have this property. If the trees of the

Downloaded from www.worldscientific.com by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

contraction kernel are independent of each other, all can be contracted in parallel, while the edges of each tree may need sequential processing. The most efficient contraction kernels are many small trees with more than one vertex.

Consequently, if the height of the pyramid has h levels, then the base graph has $|V_0| \geq 2^h$ vertices. If all the trees of the forest are independent and a sufficient number of processors are available, the next pyramid level can be computed in $\mathcal{O}(\max\{\deg(v)|v \in V_n\})$ parallel steps.

2.3.6 *Simplifying multiple edges and self-loops*

The contraction of one edge of a triangle leads to the creation of a double edge or even multiple edges. The contraction of one of the multiple edges creates self-loops (Figure 2.6). Note that the dual faces f_1, f_2 are preserved. Before the first two contractions, the degree of the faces $\deg(f_1) = \deg(f_2) = 3$. Inside the triple edges, the degrees shrink to 2 and the self-loops surround faces with degree 1. This example also shows that a simplification after the first contraction would simplify the further processing.

Multiple edges and self-loops are not topology relevant if they don't surround any further (sub-)structure. This can be decided by looking at the dual graph $\overline{G}(\overline{V}, \overline{E})$ where the degree of a face $\in \overline{V}$ provides such a decision:

Definition 2.3 (Topology-relevant). A face of the dual graph $\overline{G}$ is topology relevant for G if its degree is higher than 2: $\deg(\overline{v}) > 2$ for $\overline{v} \in \overline{V}$.

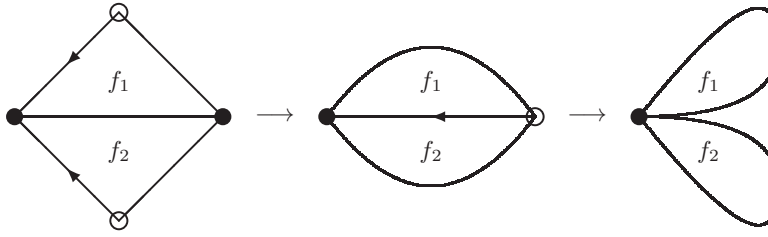


Fig. 2.6. Creation of multiple edges and self-loops.

Kropatsch2024a

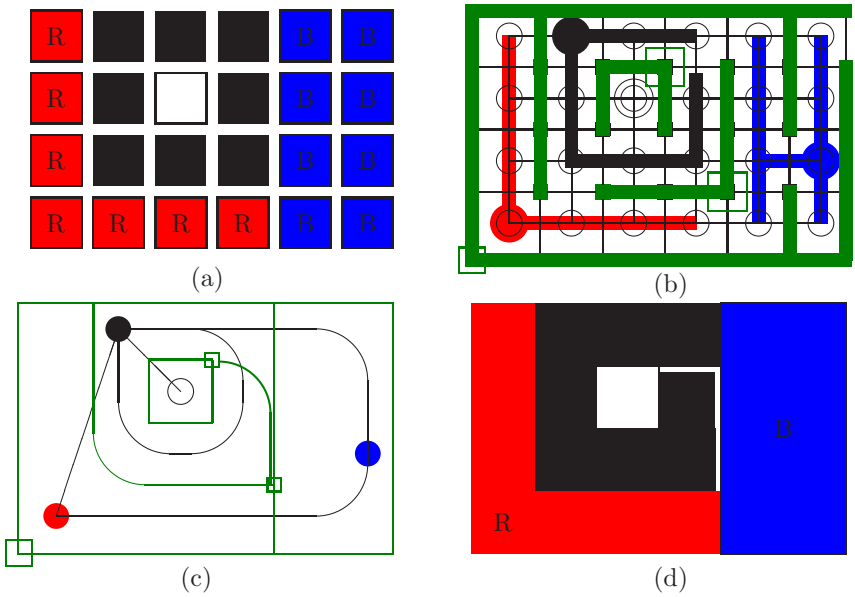


Fig. 2.7. A pseudo-edge connects the white island. (a) 4×6 colored pixels; (b) equivalent CK and RK; (c) resulting E-RAG and $\overline{E-RAG}$; (d) 4 colored regions with pseudo-edge.

Multiple edges and self-loops surrounding topology-irrelevant faces are not relevant for topology; self-loops can be removed without disconnecting either a hole or any sub-structure in the dual graph. Multiple edges can be removed as well, as long as the last remaining edge is preserved to keep the connectivity. The remaining edges are called pseudo-edges since they have the same face on both sides (see the example in Figure 2.7). Let us denote all the edges that can be removed as the removal kernel (RK).

Definition 2.4 (Removal kernel). Let $G'(V', E') = G(V, E)/K$ be the graph after contracting all edges of the contraction kernel K and let $\overline{G'}$ be its dual:

$$R'(G') = \{(v, v) \in E' | f \in \overline{(v, v)} \subset \overline{V'}, \deg(f) = 1\}, \tag{2.3}$$

$$\cup \{e_1 = e_2 = (v, w) \in E' | f = \overline{e_1} \cap \overline{e_2} \subset \overline{V}, \deg(f) = 2\}. \tag{2.4}$$

Downloaded from www.worldscientific.com by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

The set of edges in R' can be removed without modifying the topology relevance of the graph. Note that removing the edges of the removal kernel may create further redundant edges. For the complete simplification, a few more iterations of simplification may be needed, since the removal of an edge may decrease the degree of adjacent faces and may create further edges that can be removed. The complexity of this process has been shown to be the inverse of the Ackermann function [11]. A faster version has been proposed by Banaeyan and Kropatsch [12] by anticipating the contractions and removing the redundant edges in parallel before actually executing the contractions.

2.3.7 *Example with a hole and a pseudo-edge*

Edges that are not relevant for topology are often called **redundant**. There is one exception: if the removal of an edge would disconnect the graph or its dual graph. Consider the example in Figure 2.7(a). It shows the 24 pixels with the colors red, black, blue, and white. The white pixel is completely surrounded by the black connected component. Figure 2.7(b) shows the contraction kernels for the three colors red, black, and blue together with the selected surviving vertices. The white pixel survives and is indicated in Figure 2.7(b) by two concentric circles. The removal kernels are shown in green with surviving dual vertices (these are the intersections of the boundaries) marked by green squares. The background is the larger square in the left bottom corner.

Figure 2.7(c) shows the pair of dual graphs after contracting and simplifying the CK and RK. The fact that the black region completely surrounds the white pixel is expressed by the self-loop attached to the black vertex in $G(V, E)$. The edge dual to this self-loop is the pseudo-edge which connects the boundary of the white pixel with the intersection of the three connected components of the red, black, and blue regions. We call it "pseudo"-edge since both sides have the same color black while all other dual edges have different colors on both sides. The geometric placement of the pseudo-edge can be any connection of the boundary of the white with any intersection of the black region with other colors. It is illustrated by the white line in Figure 2.7(d).

Kropatsch2024a

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

The main role of the pseudo-edge is to keep the graph $\overline{G(V,E)}$ connected. The pseudo-edge is a bridge in $\overline{G(V,E)}$, the dual of which expresses the fact that black surrounds white.

If there are multiple holes in a region, each hole creates one pseudo-edge. Since their placements just need to cross the surrounding region, both can connect to the outer boundary of the surrounding region or, equivalently, only one connects to the outer boundary and the other connects the two holes. Together with the pseudo-edges the surrounding region remains homeomorphic to a topological ball. And reversely, each pseudo-edge indicates the presence of a hole in 2D. Extensions to higher dimensions exist but are not treated here.

2.3.8 *The bottom-up construction of the irregular pyramid*

The bottom-up construction of an irregular pyramid is an iterative parallel process that can be repeated until all the properties to be transferred bottom-up are application-relevant and any further shrinking would destroy relevant properties or relations. This process generates an abstraction of the base-level graph.

Given graph $G_0(V_0, E_0)$ and its dual graph $\overline{G_0}(\overline{V_0}, \overline{E_0})$, iteration count $n = 0$.

While further abstraction is possible do

(1) select contraction kernels $K_n \subset E_n$ as in Definition 2.2;
(2) perform contraction $G' = G/K_n$, $n = n + 1$;
(3) select removal kernel $R'(G')$ as in Definition 2.4;
(4) and simplify $G_n = G' \setminus R'$;
(5) apply reduction functions $RF(\cdot) : G(K_{n-1}) \rightarrow$ new reduced content

$\text{attr}(v_n) = RF(\mathcal{N}_V(v_{n-1})), v_n \in V_n$ and
$\text{attr}(e_n) = RF(\mathcal{N}_E(e_{n-1})), e_n \in E_n$.

Each iteration creates a new level $G_i(V_i, E_i), i = 0, \ldots n$, of the pyramid.

Kropatsch2024a

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

2.3.9 *Preserving topology*

Already in Section 2.2.5 we used the Euler–Poincaré characteristic referring to the relationship between the number of points P, of edges E, and of faces F in a 2D plane graph. Let us now consider the changes Δ created by the primitive operations, edge contraction, and edge removal:

Change of Euler–Poincaré characteristic

Operation	Δ**#P**	$-$	Δ**#E**	$+$	Δ**#F**	$= \mathbf{0}$
Contraction	**1**	$-$	**1**	$+$	**0**	$= \mathbf{0}$
Removal	**0**	$-$	**1**	$+$	**1**	$= \mathbf{0}$

That means that the characteristic does not change after the application of our primitive operations. More generally, any number of **contractions and removals do NOT change the characteristic**!

We have seen that regions surround their holes by a self-loop, the dual of which is a pseudo-edge. By keeping these pseudo-edges, the characteristic of the region is not changed since together with the pseudo-edges the region remains homeomorphic to a topological ball.

2.3.10 *Equivalent contraction kernels*

Similar to the equivalent weighting functions in Burt's regular pyramid, Ref. [13] introduces equivalent contraction kernels. Contraction kernels cover the receptive field of the surviving vertex. For every edge e_n in a higher pyramid level n, there exists one edge $e_i, 0 \leq i < n$, in the levels below that survives to e_n in the sense that if e_n is contracted at level $n+1$, then edge e_0 can be added to the contraction kernel $K_0(v_{n+1})$ at the base level such that the receptive field is covered by $K_0(v_{n+1})$ for vertex v_{n+1}. With the same argument, the equivalent contraction kernel of the top vertex is a spanning tree of the receptive field in the base level of the pyramid.

2.3.11 *The top-down expansion process*

Top-down expansion has been used effectively in classical Laplacian pyramids by Burt and Adelson [14]. In regular pyramids, the structure of the pyramid depends only on the size of the base image and

Kropatsch2024a

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

hence the size of the different levels above the base does NOT vary for images with the same input size. For irregular pyramids, the structure of the graphs of the different levels depends strongly on the content of the data. The selection, the contracted edges, as well as the other control parameters may depend on the content of the image. Hence, irregular pyramids on different images may have a completely different graph structure. However, we built them bottom-up, level by level, and with only two different operations: contraction and removal of edges.

In Ref. [8], we have shown (1) that there are inverse operations to the two basic operations and (2) that we need to remember only a few parameters of the bottom-up process to reconstruct the higher-resolution graph. We call the inverse operations **decontraction** of a contracted edge and **reinsertion** of a removed edge (Figure 2.8). In this canonical encoding of the irregular pyramid, we store the parameters of the contracted and removed edges in the order they have been applied. These recycled garbage parameters allow us in the top-down reverse process to recover the graphs at the lower levels.

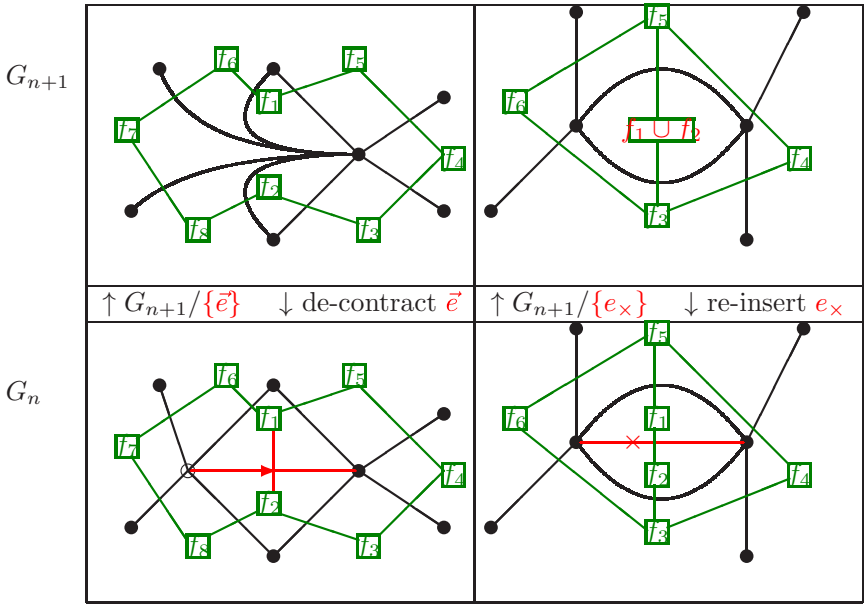


Fig. 2.8. Inverse operations: contract and de-contract, remove, and reinsert.

Kropatsch2024a

The canonical encoding enables first of all to reconstruct the levels below the top level. However, more importantly, we can down-propagate the abstract attributes collected in the higher levels. Different options are discussed in Section 2.4.4.

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

2.4 Control by the Content

The previous section covered the main components for constructing an irregular pyramid and for expanding the abstract information from the top level down to the pixels of the original image. We can identify four categories of control over the general process, influencing either the constructed structure or the architecture of the hierarchy or preserving certain real-world properties of objects to be represented in the base-level image: The selection of contraction kernels identifies the surviving vertices and some of the incident edges that are not relevant for the main properties of the objects. The simplification strategies 'clean' the graph after each contraction phase. Also here there are possible choices, e.g., what parallel edges should survive. Reduction functions use the attributes of the survivor's children to compute a more abstract description of the content of the receptive field. Once a certain number of levels of the pyramid has been generated, the extracted high-level description can be expanded to the lower levels in order to (1) display the abstract content of any higher level in the form of an image and (2) probably revise some decisions taken at the bottom-up process in order to make the content of the complete pyramid consistent with the abstract findings at the higher levels.

2.4.1 *Select contraction kernels*

The simplest choice of contraction kernels is a random choice for constructing the stochastic pyramid of Meer [9]. In the adaptive pyramid of Jolion and Montanvert [15], the random choice is replaced by choosing the irregular sampling from the content of the data. Such adaptation could be convolution filters of which a local maximum identifies the surviving vertex. Note the high similarity to common *'max-pooling'* layer in deep learning architectures. But also rules can be used to select edges to be contracted. In connected component

Kropatsch2024a

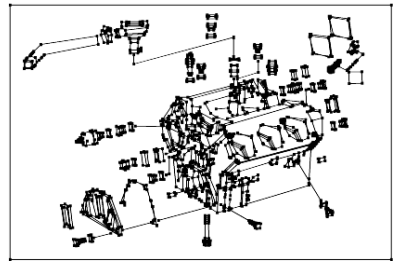


Fig. 2.9. Technical drawing of a motor engine.

labeling (CCL), a simple rule is to contract only edges connecting vertices with the same label. A more complicated rule for selecting contraction kernels has been used for closing gaps in scanned line drawings [16, 17] (see the example in Figure 2.9). Even parametric models could be used for determining the important vertices to preserve and the edges to contract. This could be as simple as correlating the data with the model or finding the best *'goodness of match'*. Finally, the matching of graphs that is in general NP hard could be done using the fact that using the same selection rules for two images is likely to generate much simpler and similar graphs at higher levels of the two pyramids in this case [18].

2.4.2 *Simplification strategies*

There are two different criteria for selecting the removal kernels: either the content-based choice in choosing the surviving edge of multiple edges according to the attributes of the edge (e.g., shortest accumulated arc length) or the attributes of the two adjacent faces of the dual graph (e.g., distance to the outer most parallel edge) or the computational choice of how many iterations of simplification should be done. Complete simplification after each contraction needs $\mathcal{O}(a^{-1}(n,n))$ steps in the worst case where $a(n,n)$ is the Ackermann function. Alternatively, only one simplification pass is executed after contraction, leaving the remaining multiple edges and self-loops for simplification at higher levels. This may of course indirectly slow down the construction since neighborhoods with non-simplified redundant edges are larger. The last alternative is to do all simplifications after all contractions.

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

Under certain conditions (having a total order of all vertices), simplification can be anticipated before contraction.

A noteworthy alternative has recently been proposed by Banaeyan [19]. This approach can achieve simplification prior to contraction under certain conditions, specifically when a total order of all the vertices exists. In the case of a binary image, independent edges (i.e., edges not sharing an endpoint, [20]) are encoded to allow for the removal of redundant edges originally at an upper level, at the current level with parallel constant complexity [12]. This method accelerates the construction of the pyramid and transforms it into an efficient tool for computing the distance transform of a binary image with parallel logarithmic complexity [21], provided that there is sufficient number of processing elements for parallel computations.

2.4.3 *Reduction functions*

The role of reduction functions is to propagate the image content to a lower resolution while at the same time increasing the degree of abstraction. While a pixel in the base may have the color, i.e., red, it may be aggregated at a higher level into a red ball.

The simplest reduction function is used in CCL: All the vertices in the contraction kernel have the same color hence the surviving vertex will inherit the same color. The second most frequently used choice is a (weighted) average or, more generally, a convolution filter (as frequently used in DCNN[c]). A more sophisticated reduction uses the transitive closure of a set of relations (i.e., describing the layout of curves in line drawings, such as Figure 2.9 [16]). Both in the processing of line drawings as in the closing of gaps [22], i.e., between the dashes of a dashed line, the introduction of an **isolated blob** ⊙ allows establishing neighbor relations between the dashes rather than connecting all the dashes to the common (white) background. Figure 2.10 shows an example: the input image, the resulting graph, and the receptive fields of the different isolated blobs. The survivor received an additional symbol ⊙ for dashed/dotted lines if the black dash ● appears completely surrounded by the white background ○ . The rule for contraction is then extended by the isolated blob ⊙ : in addition to the fact that the same categories ● , ○ can be merged as

[c]Deep Convolutional Neural Network.

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

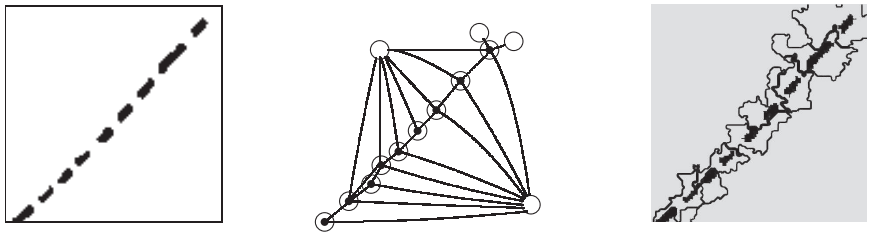


Fig. 2.10. Recognizing a dashed line.

in CCL, we allow ⊙ to merge with ○ but not with ⊙ . The growth of the isolated blobs happens concurrently to the growth of the (empty) background ○ such that close-by ⊙ neighbors are detected before all the background merges into a large region where the individual blobs are all surrounded by individual self-loops.

But also parametric models may determine the parameters best describing the receptive field of the surviving vertex. Of course, models can become more complex and parameters that best match the data [23] can be used to describe the vertex by the name of the model and its parameters. All these models offer opportunities for optimization and learning.

There is no need to use the same reduction function when reducing one level to the next. Of course, it is the simplest choice if no other source of information is available. But if you consider the dynamic processing of visual data or have a target segmentation available, there may be previous labels and features available such that the reduction function can be adapted for the general model from the previous image frame. And not only concerning the parameters of the reduction function but also the principle type of function, e.g., switching from a filter to inheritance or the transitive closure of the boundary segments.

2.4.4 *Controlling the top-down expansion*

As with the classical Laplacian pyramid [14], a first motivation is to show that the original image can be reconstructed from the higher levels. But even more, a simple inheritance expansion where children inherit their attributes from its parents, without trying to reconstruct

Downloaded from www.worldscientific.com by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

the original attributes, provides some insight into what has been aggregated in the higher levels. In the Laplacian expansion, the high frequencies of the lower levels have not been added.

In the irregular pyramid, the expansion has become feasible by the inverse operations [8], decontraction for contraction and reinsertion for removed edges. Originally, these inverse operations were applied in reverse order to be able to re-establish properly the links to the already expanded graphs. However, this strict order, which would prevent parallel application, can be relaxed since also the bottom-up operations were independent and create layers of contracted edges (by one contraction kernel) alternating with removed edges through simplification. Similar to the concept of wavelets, this process can be memory neutral in the sense that the active level where the current top level graph is stored complements the passive part where links of the contracted and removed edges are kept. Together they occupy the same memory as the base level.

The recovery of structure of the lower levels of the pyramid offers a wide variety of possibilities to propagate high-level information (referred to as the parents, the surviving vertex together with its neighbors) to the lower levels (referred to as the children). Options that have been used are as follows:

- interpolating the attributes of the children from the attributes of the parents,
- or using convolution filters applied on the parent's level,
- or inheriting the parent's attributes (as for CCL),
- or refining the high-level model and potentially updating the bottom-up model by properties like straightness of a dotted line that cannot be done locally during the bottom-up process,
- or reinsertion of curve segments to re-establish connectivity,
- or generative models like fractals.

We give examples for some of the operations in the following section.

2.4.5 *Preserving relevant properties*

In Table 2.1, we give examples of the different choices of the control decisions used by specific applications together with citations to papers with the details and results. In nearly all cases, empty faces,

Kropatsch2024a

Table 2.1. Overview of control for specific applications.

Application	Important elements survive	Negligible elements are merged
CCL	1 repr/CC(lab)	(L, L)
segmentation	1 repr/ region	similar, end points
2x on curve	X, ends	empty space, connections
line images	ends, junctions	empty space, connections
matching	discrim.template, object boundary	simil.inside object
motion	foreground, static background, articulations	occluded backgr. moving foregr.
gap closing	1 repr/lab incl. background	(L, L)
E-RAG Hierarchy	max.ext.Contrast, MST	min.int.Contrast

with deg < 3, are considered redundant in the simplification and merged with one adjacent face (corresponding to the removal of the separating edge).

Connected component labeling (**CCL**) [17, 24–26] has as input a labeled image. It could be a hand-labeled ground truth or the result of a segmentation, and the task is to find the connected components of the different labels together with their adjacencies. One vertex of every connected component should survive to the top (1 repr/CC(lab) in Table 2.1) while edges connecting vertices with the same label can be contracted (denoted by (L, L) in Table 2.1).

There are numerous studies of **segmentation**, i.e., [27–30]. In this case, every connected region will be represented by one surviving vertex in the top level and edges connecting similar vertices are contracted to the edges of the RAG. For thin regions, it may be useful to keep the end points to some higher levels.

The psychological test of "**2X on a curve**" consists in finding out whether two X placed on two complicated but non-overlapping curves are on the same curve or on different curves. It was argued that humans need a time proportional to the length of the curves. In our paper [31], we showed that the pyramid can solve it in logarithmic time by (1) preserving the "X" vertices and (2) contracting

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

the empty space without curve segments and contracting connected curve segments.

In processing **line images** (e.g., technical drawings, Figure 2.9), the preservation of line ends and of junctions is important [32–34]. Similar to the previous application, contraction applies to the empty space and to connected curve segments. For line images, there was an additional constraint that the face should not touch any curve because it would establish a wrong connectivity. Here the adaptivity of the irregular pyramids is a great advantage.

In the application of finding **matching**s between two images [18, 35, 36], as in stereo or in image mosaics, the most discriminative template should survive together with the object boundaries while edges connecting similar vertices inside an object can be contracted.

The problem of detecting **motion** in image sequences involves more than a single or a pair of images [37–39]. In this application, the task is to identify a moving object in front of a static background and to identify the moving parts of an articulated movement (walking or hand gestures). It is important to keep one vertex of each connected foreground object and the static parts of the background. In addition, the articulation points need to be preserved in order to derive, e.g., a proper walking pattern. Contraction can be applied to edges inside the background or inside a moving foreground object. Expansion can be used to build a more complete background model by inserting parts that have been temporarily occluded by a moving foreground as well as tracking the moving foreground objects over time.

The **gap closing** application [40, 41] has been discussed together with the drawing of line images in Section 2.4.3. We have shown that the introduction of a new label $\odot$ for isolated blobs can be determined locally by the given graph structure (the self-loop surrounding a blob) and can be efficiently used as new entity to control the growth of the different categories of labels.

The last example in Table 2.1 is entitled "**E-RAG hierarchy**" [42–44]. The preservation of topology enables the classical region adjacency graph to allow also self-loops and multiple edges. These are necessary to properly represent the inclusion of holes in a large region and the fact that two regions may be connected by more than one connected boundary. The criteria used in this application were that vertices with the highest external contrast (according to [45])

Kropatsch2024a

survive and the edges of a minimal spanning tree of the internal contrast are contracted.

The last application, the LBP-pyramid, is discussed in Section 2.6.

2.4.6 *Properties of topological pyramids*

Let us call topological the pyramids that preserve the topological properties of the data/images. In particular, it concerns holes of regions and the related inclusion relationships. Topological properties are to a large extent invariant to geometric deformations like different view points, perspective projection, articulated movements, etc. But topological properties are also sensitive to noise and care must be taken when removing noise.

Concerning the data structure for storing the topological pyramid, the matrix structure of a regular pyramid is definitely not able to properly represent all relevant topological features explicitly. For example, a small hole may quickly be too small to be represented at lower resolutions. But more importantly, thin structures like roads or rivers in a remotely sensed image are likely to disappear when their width drops below the sampling distance. That is why we have focussed on embedded graphs as a primary data structure, although there are less known representations like combinatorial maps, generalized maps, or cellular complexes that suit the purpose for preserving topology as well.

We already addressed an important aspect of graphs: Simple graphs without multiple edges and self-loops cannot capture holes and multiple connected boundaries. We showed that graph pyramids can be constructed with only two operations: edge contraction and edge removal.

We have seen also the particular importance of preserving key vertices to higher levels; they allow keeping the overview of the main components of an image and often relax particular details that may not be necessary once the object has been identified.

In two dimensions, plane graphs represent the graph embedded in the plane as shown in Figure 2.11. The base of the primal graph corresponds to the 4-neighborhood of the image while the dual graph has an important role in deciding the removal kernels without long search processes.

The bottom-up construction is controlled by application-specific properties. It preserves the connectivity and the relevant inclusions

Downloaded from www.worldscientific.com by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

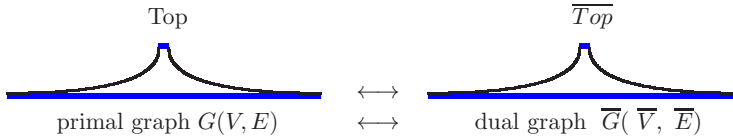


Fig. 2.11. Dual graph pyramid.

during the bottom-up process. Pseudo-edges are bridges in the dual graph that connect a hole with the remaining graph. Their deletion would disconnect the graph and remove the information about what end point is included in the receptive field of the other. Its dual edge is a self-loop indicating the inclusion. Each hole can be associated with one pseudo-edge. This remains true also in higher dimensions, i.e., a pseudo-face characterizes a tunnel through a volume, a typical example is a torus.

The concept of equivalent contraction kernel (ECK) relates the higher levels to the lower levels directly without the need to propagate across several levels. The ECK(v) of any vertex of the pyramid covers the complete receptive field of vertex v. This becomes of particular interest for color images where each color channel creates a separate pyramid structure. In this case, it is very difficult to compare the higher levels of the three pyramids directly. However, through the ECK, each vertex can be down-projected to the common image structure where the comparison could be done.

Another important aspect of pyramids is that many operations can be executed in parallel on different processors. We have shown ways to identify independent operations, making the computational complexity even for large images as those mentioned in Section 2.2.3 feasible [19].

2.5 Local Binary Patterns (LBPs)

Local binary patterns have been introduced by Ojala and Pietikainen [46] in 1996 as an efficient descriptor for textures in images. The eight neighbors of the center of a 3×3 window compare their gray value with the center and set a 1 if the neighbor is higher in value and a 0 otherwise. The resulting eight bits are concatenated in a

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

pre-defined (clockwise) order and form a value in [0,255]. This works very well for the eight neighbors of an 8-connected grid of an image.

It fails if the number of neighbors varies like in a graph with vertices of different degrees. However, a graph has also edges in addition to the vertices. We therefore store the result of the comparison not with the (center) vertex but with the edge connecting the center vertex with the neighbor by simply orienting the edge such that it always points to the lower valued vertex. In this case, the characteristic bit switches of LBPs translates into an orientation switch of the edges surrounding a vertex. This way not only relaxes the degree of the vertices but also saves more than 50% of the memory. In addition, all the characteristics of LBPs like the differentiation of critical points (minima, maxima, and saddle points) translate 1-1 to the new representation.

LBPs identify the class of uniform codes: These are codes that contain maximally two bit-switches when turning around the center. In our new representation, these are local configurations that are either extrema (0 bit switches) or their neighborhood splits into a higher connected part and a lower connected part separated by a level curve across the center. The two bit-switch configurations roughly form a slope (precise definition is given in the following). Non-uniform LBPs correspond to saddle points.

2.5.1 *Critical points of a height profile*

Let us first consider an LBP along a one-dimensional (1D) curve (Figure 2.12) or a profile across a two-dimensional (2D) surface. Basic mathematics tells us that critical points are characterized by horizontal tangents. In 1D, critical points are local extrema: local maxima $\oplus$ and local minima $\ominus$. From Figure 2.12, we see that the curves

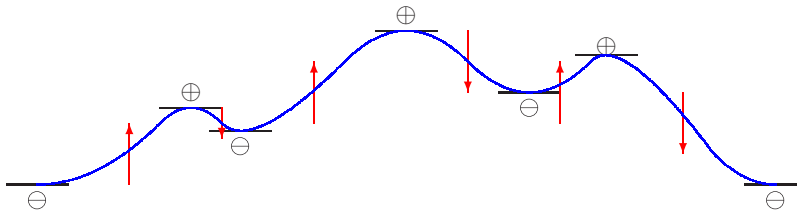


Fig. 2.12. A smooth curve with seven critical points.

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

between the critical points are **monotonically** increasing from the minima toward the maxima while they **monotonically** decrease from the maxima toward the minima.

Definition 2.5 (Monotonic). A function $f(x) : D \mapsto \mathbb{R}$ is called **monotonically increasing** in the domain $D \subset \mathbb{R}^n$ if $f(y) \leq f(x)$ for all $x \leq y$ in D, and it is called **monotonically decreasing** in the domain $D \subset \mathbb{R}^n$ if $f(x) \leq f(y)$ for all $x \leq y$ in D. It is called **strictly monotonic** for strict inequalities.

2.5.2 *LBPs along a monotonic curve*

LBPs compare a central point $f(x)$ with its neighbors $\mathcal{N}(x) = \{n|\delta(n,x) \leq \Delta\}$. In 1D, we use $\delta(x,y) = |x-y|$ and $\Delta = 1$, and in 2D, the Euclidean distance

$$\delta\left(\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}\right) = \sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}. \tag{2.5}$$

LBP stores a binary value of 0 if the neighbor is smaller or equal and a value of 1 if the neighbor is greater than the central point:

$$LBP(x) = b_0, b_1 \text{ with} \tag{2.6}$$

$$b_i(x) = \begin{cases} 0\ldots \text{ iff } f(n_i) \leq f(x) \\ 1\ldots \text{ iff } f(n_i) > f(x) \end{cases} \quad \text{and} \quad i \in \{0,1\}. \tag{2.7}$$

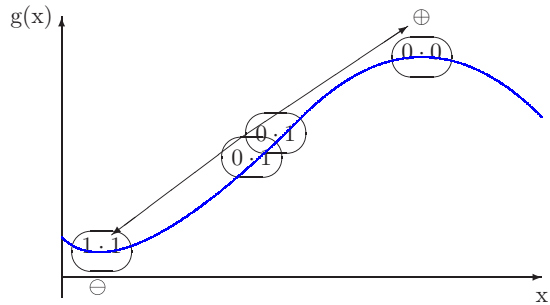


Fig. 2.13. A 1D curve between a local minimum and a local maximum.

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

In 1D, every point has two neighbors: one (n_0) with lower x and one (n_1) with higher x (Figure 2.13). Consequently, there are four different LBP codes:

code	meaning
00	local maximum ($\oplus$)
01	monotonically increasing curves
10	monotonically decreasing curves
11	local minimum ($\ominus$)

2.5.3 *Monotonic curves/paths π*

Figure 2.14 shows a curve with two sharp peaks (local maxima) and a flat minimum. In the following, it shows the corresponding graph $G(V, E)$ where V represents the critical points/segments ($\oplus, \ominus$) of the curve and the edges between are oriented such that the end point is lower than the begin of the corresponding curve segment. Along the monotonic curves, the orientation of all the edges remains the same, hence they can be collapsed into a single edge if a varying steepness does not matter. The critical points in a 1D continuous curve can be determined by a 1D LBP without computing derivatives and even at non-smooth locations (like the two sharp peaks in Figure 2.14). At sharp corners, there are multiple orientations of tangents and the derivative cannot be computed.

The curve segments between the critical points correspond to the edges and they are all oriented toward the minimum: $\oplus \longrightarrow \ominus$ is monotonically decreasing and $\ominus \longleftarrow \oplus$ is monotonically increasing. Note that the curve need not be smooth with the only requirement that the sampling satisfies the Nyquist–Shannon theorem (at least for the critical points [47]).

A discrete monotonic path $\pi(p_1, p_n) = (p_1, \ldots, p_n), \quad p_i \in \mathbb{R}^n$, is a polygon in $\mathbb{R}^n$ without self-intersection. Formally, we can state

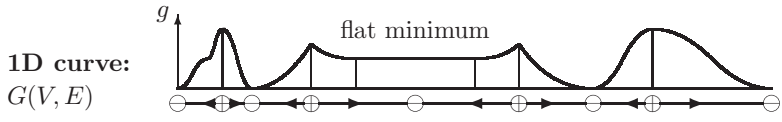


Fig. 2.14. A 1D curve with its oriented graph $G(V, E)$.

Downloaded from www.worldscientific.com by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

that the orientation of a monotonic sequence of edges $(p_i, p_{i+1}) \in E$ can be derived from the sign of σ in (2.8):

$$(f(p_{i+1}) - f(p_i))\sigma \leq 0 \quad \forall i \in [1, n-1], \sigma \in \{-1, +1\}. \tag{2.8}$$

The sequence is increasing with $\sigma = +1$ and decreasing with $\sigma = -1$.

The original LBP bits associated with the vertices in V are transferred to the orientation of the respective edges. Turning around any vertex $v \in V$ in a plane graph,[d] we can derive the corresponding LBP: bit 0 if the edge is $e = (v, w) \in E$ and 1 for $e = (w, v) \in E$. The change in orientation corresponds to a bit switch in the LBP code and enables vertices with different degrees.

Flat regions introduce an asymmetric LBP code in the original definition of Ojala *et al.* [46] giving raise to a local ternary pattern (LTP) in the work of Tan and Triggs [48]. The drawback is that 2 bits are necessary instead of 1 bit doubling the size of the code and there is no obvious translation into orientation. Since flat edges do not contribute to the detection of critical points, we propose to contract flat edges in the graph. This preserves the connectivity and converts monotonicity into strict monotonicity. There is one exception: self-loops are crucial for describing holes in a region. But self-loops are easy to detect by the fact that both end points are the same vertex. As we explore later, a small extension to edges with the smallest contrast allows removing most flat edges. As a side effect, it also shortens the paths π and preserves the critical points.

2.5.4 *Critical points in 2D*

Critical points in 2D can be recognized by **LBP** [46]. As in 1D, bit switches translate into a change in orientation of the edges between adjacent neighbors. We keep the downwards orientation as in 1D. Hence, there are no changes in the orientation of an extremum and two changes for saddle points (see examples in Figure 2.15). Even a third category of local configuration with two bit switches can be described by 'uniform' LBP codes: slopes.

[d] A plane graph is an embedded planar graph such that the order of edges around every vertex in the embedding is given.

Downloaded from www.worldscientific.com by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

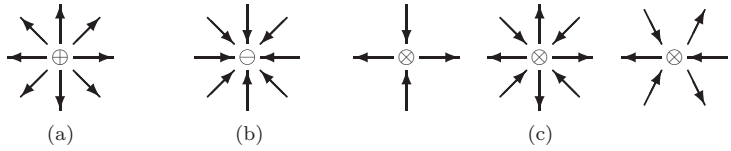


Fig. 2.15. Examples of critical points in 2D. (a) local max. $\oplus$; (b) local min. $\ominus$; (c) local saddle $\otimes$ with degrees 4, 8, 6.

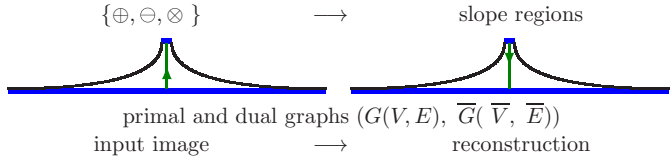
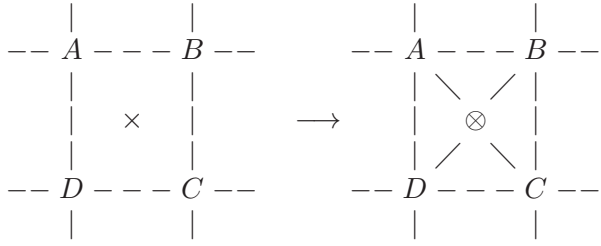


Fig. 2.16. Bottom-up and top-down processes in the LBP pyramid.

2.6 The LBP Pyramid

This section follows the general principles laid out in Section 2.3. References [49, 50] focus (see overview in Figure 2.16) on the particular choices of the LBP pyramid. We have seen in the previous section that the binary coding of local binary patterns (LBP) can be transferred to an oriented graph and that the LBP along a monotonic path shows the same pattern, e.g., the orientation of edges along a monotonic path is the same. Critical points can be determined from the number of orientation changes around a vertex, in most cases by a local process. In case of flat regions, the detection of critical points is postponed until all the flat regions are represented by a single vertex. Then the detection is local.

There is one special case for a 'hidden' saddle point. This is a saddle point that falls between the sampling points of the data and is characterized by a condition related to the non-well-composed 2×2 configurations of Latecki [51]:



(2.9)

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

The block of pixels A, B, C, D is not well composed if either

$$L = \max(A, C) < H = \min(B, D) \text{ or} \tag{2.10}$$

$$L = \max(B, D) < H = \min(A, C). \tag{2.11}$$

In all such cases, an extra saddle vertex $\otimes$ is inserted in the center of the 2×2 block of the neighborhood graph, connected to all four vertices A, B, C, and D of the 2×2 pixels by edges and with a gray value $f(\otimes)$ in the interval (L, H).

2.6.1 *Bottom-up construction and top-down expansion*

As contraction kernels, we select edges with the locally lowest contrast and choose the critical points as survivors. If an edge with lowest contrast is not incident to a critical point, any incident vertex can be chosen. There are two main arguments for this choice:

(1) Since low-contrast edges are visually nearly indistinguishable, we use the following selection criterium for edges to be contracted: Contracting edges of contrast zero shrinks successively flat areas until reaching a single vertex for each connected flat area.[e] After zero-contrast edges have been contracted, the remaining edges with contrast $> 0, e = (v, w) \in E$ are downwards-oriented, i.e., $f(v) > f(w)$, and the contrast of e is the difference between the end points: $contrast(e) = f(v) - f(w)$. In order to satisfy the independence condition, one can first determine a maximal independent vertex set starting with the critical points and then choose trees of incident edges with locally lowest contrast.

(2) Each connected component of the contraction kernel contains one critical point, if possible. It remains critical even after contraction, and, consequently, the critical point survives the contraction process. Preserving the critical points $\oplus, \ominus, \otimes$ of the base graph

[e]Care must be taken if these flat areas connect non-connected parts of the boundary of the graph since the remaining vertex becomes an articulation point the removal of which would disconnect the graph. This can be avoided by first contracting and simplifying the boundary and then the inner flat areas while preserving the boundary. This enables treating sub-graphs separately and stitching them together after contraction.

Downloaded from www.worldscientific.com by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

does not shrink the range of values (in contrast to smoothing or interpolation). Together with a strong contrast they contribute to the high visual quality of the reconstructed image. In case of dense clusters of critical points or in case of critical points generated by noise, the rule of preservation may be relaxed for generalization by allowing lowest contrast pairs $(\oplus, \ominus), (\oplus, \otimes), (\otimes, \ominus)$ to be contracted (as in [6]).

The attributes of the base level are the gray values of the pixels. An alternative would be to use the contrast as an attribute of an edge where the orientation encodes the sign of the contrast. With this encoding, only a few gray values need to be kept with some vertices since the other gray values can be recomputed by propagation along the edges of the graph.

In the simplification of multiple edges, the longest equivalent path in the base can be chosen. This length can be easily integrated in the bottom-up process by first initializing a length attribute of each edge by the value 1, and after each edge contraction, the length of the edge becomes the sum of the lengths of the two involved edges.

As reduction function, surviving vertices inherit the value of the level below. Since critical points are primarily chosen for survival, the range of gray values is preserved.

The top-down expansion can be done using the canonical representation [8] by edge decontraction and (removed) edge reinsertion. During the expansion process, the children of the lower level either retrieve their value from the status of the bottom-up process or they inherit the value of their parent from the level above. In the experiments, we chose the option of inheritance to judge the quality of the reconstruction with only a few values of the top level.

2.6.2 *Main properties of the LBP pyramid*

The bottom-up construction preserves relevant critical points (see Table 2.2) that are determined by LBP [50, 52, 53]. Hidden saddle points are inserted in the original neighborhood graph.[f] The selection of edges with the lowest contrast also preserves the original contrast

[f] This is easy in a graph but would require an increase of the resolution of an array.

Downloaded from www.worldscientific.com by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

Table 2.2. Control of LBP pyramid.

Application	Important elements survive	Negligible elements are merged
LBP pyramid	critical points, texture, high freq.	lowest contrast

of the image as well as the quality of fine details like the grass in Figure 2.18. Monotonic paths remain monotonic if no critical point is removed. LBP codes are known for their texture representation and this property is clearly visible in the reconstructions.

The reconstruction quality of the LBP pyramid has been tested in Refs. [50,52,53] with a variety of images from the Berkeley database. The reconstructions use much less colors and preserve very well the structure and topology of the image. Selected pictures in Figure 2.17 are from the Berkeley image database [54]. More examples can be found in PRIP TR-133, the Master Thesis of Martin Cerman.[g]

2.6.3 *Image = Structure + Few Colors*

In Ref. [55], we investigated the reasons why the LBP pyramid reconstructs images with surprisingly high visual quality. We could confirm the main observation of Koenderink [6], although he used Gaussian-type smoothing for the construction of the lower scales that cannot preserve so well thin structures as the LBP pyramid does. This can be visually verified in Figure 2.18 where three reconstructions with the LBP pyramid are compared with a classical Gaussian pyramid with a reduction window of 5×5 pixels and a reduction factor of 4, corresponding to a stride of 2 in x and y directions. GE stands for one Gaussian reduction and one expansion, and GGEE for two Gaussian reductions and two expansions. Figure 2.18 summarizes the number of vertices or pixels at the apex of the pyramid and the corresponding reduction from the base to the top. Most of the critical points survive to the low resolutions at high levels and contracting the lowest contrast first preserves the high contrast in the image. And, in contrast to ALL smoothing reductions involving convolutions, it preserves high

[g]https://www.prip.tuwien.ac.at/publications/technical_reports.php.

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

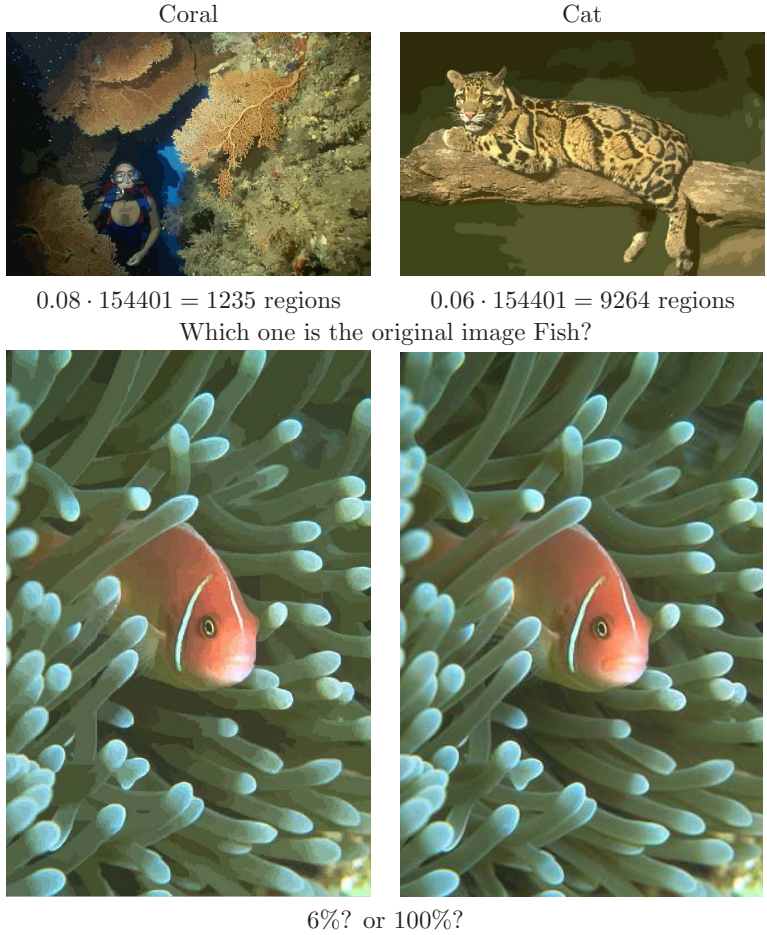


Fig. 2.17. Three reconstructions with the LBP pyramid.

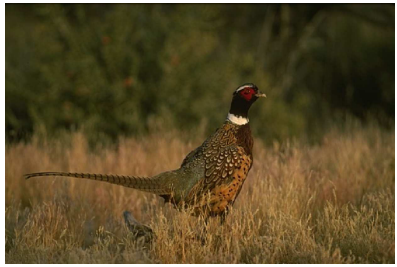
frequencies in the image (small, thin details). Reconstructions with only a few highest levels give good results. Reconstructions with only 30% down to 3% of the regions of the original number of pixels are shown in Figure 2.18.

In Ref. [55], we qualitatively and quantitatively compared the results on 100 images of the Berkeley database with 4 other methods. For the assessment of the quality, we used the Structural Similarity Index Measure (SSIM) [56], the Feature Similarity Index Measure (FSIM) [57], and the Peak Signal-to-Noise Ratio (PSNR).

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.



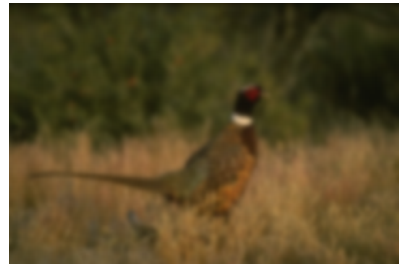
Original

with 30% = 51 467 Regions

with 10% = 15 440 Regions

with 3% = 5 147 Regions

1 level Gaussian 25% = 38 801 Regions



2 levels with 6% = 9 801 Regions

Pyr(Picture)	pixels	$\|V_n\|$	reduction by
SCIS(Pheasant)	154401	46320	70%
GE(Pheasant)	154401	38801	75%
SCIS(Pheasant)	154401	15440	90%
GGEE(Pheasant)	154401	9801	94%
SCIS(Pheasant)	154401	4632	97%

$|V_n|$ is the number of vertices/pixels at top
'reduction by' is the reduction of the vertices from the base to the top level.

Fig. 2.18. Pheasant, Berkeley# 43074, three reconstructions, two $5 \times 5/4$ Gaussian.

Kropatsch2024a

2.7 The Space between Critical Points

After constructing an irregular LBP pyramid, most of the vertices correspond to critical points of the base level. Hence Ref. [58] asks the following question: What are the spaces between the critical points? In this section, we give an overview about the concept of **slopes** and their interesting properties with some outlook for future research directions. This concept relates the different levels of the pyramid by covering the regions at different levels with such slopes. Since the critical points determine a partition into slopes, it enables the interpretation of the information at different levels of the pyramid.

We have seen in Figure 2.13 that the curves between extrema in 1D are monotonic curves or profiles. Now, we extend this concept to two dimensions. We define the 2D counterpart of a monotonic curve as a slope:

Definition 2.6 (Slope). A connected region R of a continuous surface is a **slope region** iff all pairs of points $\in R$ are connected by a continuous monotonic curve $\in R$.

The smallest slope in an image is a single pixel. Locally, a slope can be characterized by a uniform LBP.[h] Definition 2.6 defines the slope in continuous Euclidean space, but it is also valid in discrete spaces like images or graphs.

We recall some of its properties and refer to our previous publications for proofs and further examples. The domain of the image function f can be partitioned by slopes. Critical points determine the structure of a slope $\mathcal{S}$: Every slope can contain one local maximum ($\oplus$) and one local minimum ($\ominus$). Saddles ($\otimes$) appear exclusively on the boundaries between slopes [59]. All **level curves**[i] in a slope are connected. Level curves of f may be open when intersecting the boundary of the domain of f or closed. Level curves can intersect exclusively at saddle points, never inside R (more in [60]).

We distinguish between two types of slopes:

- slopes bounded by level curves and
- slopes bounded by monotonic curves.

[h] A uniform LBP has maximally two bit switches or, equivalently, maximally two changes of orientation when turning around the center.

[i] Also called contour lines or isolines.

Downloaded from www.worldscientific.com by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

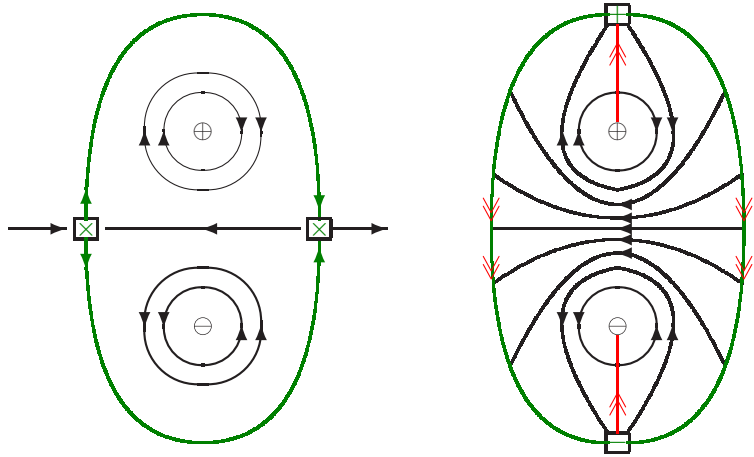
2.7.1 *Slopes bounded by level curves*

The complete boundary is a level curve at the level $f(\boxtimes)$ of the saddle point[j] with following constraints if a local maximum $\oplus$ and/or a local minimum $\ominus$ exists inside the slope (Figure 2.19(a)):

$$f(\oplus) > f(\boxtimes) > f(\ominus). \tag{2.12}$$

The region including the $\oplus$ will be higher than the boundary, and the region with $\ominus$ will be below the boundary if both minimum and maximum are inside the slope. This implies that there exists a curve inside the slope at the level $f(\boxtimes)$ that separates the higher from the lower parts of the slope. This level curve connects the two saddle points $\otimes$ having the same level. These two saddle points of identical level are necessary to prevent the slope from having an articulation point.

Both the boundary and the separation curve meet at the two saddle points. All other level curves inside the slope are closed. Orienting level curves such that the right side is higher than the left side, the level curves opposite $\boxtimes$ have opposite directions (see Figure 2.20).



(a) (b)

Fig. 2.19. Two basic types of slopes. (a) A level-bounded slope; (b) A slope bounded by two monotonic curves.

[j] $\boxtimes$ denotes a saddle point along the boundary, $\boxplus$ and $\boxminus$ a local maximum and a local minimum along the boundary, while $\oplus, \ominus$ denote local extrema in 2D.

Downloaded from www.worldscientific.com by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

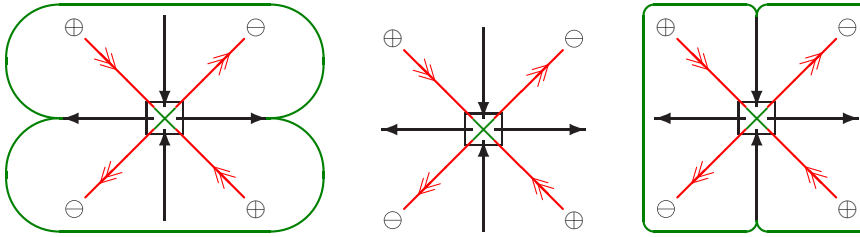


Fig. 2.20. Orientations around a saddle point.

Each extremum inside $\mathcal{S}$ is surrounded by closed level curves. Regions around ⊠ alternate higher and lower parts of f (Figure 2.20; ⟸ indicates that the edge increases from left to right). As a consequence, there are two ways to group the higher and the lower parts into two slopes with maximal extension: (a) using horizontal neighbors in Figure 2.20 or (b) using vertical neighbors. Such grouping can be related with certain semantic properties like size, shape, or texture. It can be established at the top level of the pyramid and then successively refined top-down, level by level, to the base level.

2.7.2 *Slope with a monotonic boundary*

This is most likely the more frequent type of slope since the bounding saddle points have different levels and are connected by monotonic curves. Figure 2.19(b) shows a prototype of such a slope. We draw boundary curves in green, and monotonic curves with ⟸ pointing downwards the levels.

Level λ curves inside a slope are connected and can be open or closed with the following constraints:

Level curves around the two extrema are closed for levels λ satisfying:

$$f(\boxplus) \leq f(\lambda) \leq f(\oplus), \tag{2.13}$$

$$f(\ominus) \leq f(\lambda) \leq f(\boxminus). \tag{2.14}$$

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

All level curves $f^{-1}(\lambda)$ with $f(\boxminus) < f(\lambda) < f(\boxplus)$ are open and connect the two monotonic branches of the boundary (except $\boxminus$, $\boxplus$).

Slopes can partition any continuous surface [59]. However, partitioning into slope regions is not unique. Monotonic boundaries have some degree of flexibility to grow or shrink within the limits of the level curves through the saddle points. This opens also the possibility to introduce a limited overlap between two slopes where the boundary is not clearly determined.

2.7.3 *Outlook on slopes*

Besides the receptive fields of high level vertices and faces, slopes provide a further tool to explain the derived structure and attributes of the higher levels at the higher resolutions in the levels below. One target could be to derive a covering of the base level with a minimal number of slopes. Together with features derived for each slope it could be used to recognize similar image regions or objects in the image.

Since saddle points appear exclusively along the boundary of slopes and the level curves inside have characteristic patterns, we could consider the level curves that pass through saddle points. The connected components between these level curves form a hierarchical structure that has great similarity with the topological tree of shapes [61]. These similarities would be interesting to study not only due to the continuous and the discrete concept but also where the two concepts match and whether there are differences.

Finally, any hill-climbing inside a slope region reaches the peak, and any steepest descent inside a slope region reaches the minimum in all cases. There is definitely a potential for optimization processes to avoid being trapped in intermediate local extrema.

2.8 Conclusion

In this chapter, we first give an overview of the main components of irregular pyramids. They differ from the classical (regular) pyramids in that they are based on irregular data structures like graphs. This enables them to adapt their internal structure to the input data in the base level or to the target structures important for the application.

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Irregular pyramids preserve the intrinsic (cell) structure[k] at higher levels. In this chapter, we have used plane graphs as basic topological data structures because graphs can be assumed to be widely known. But several other data structures can be used with varying advantages, in particular for dimensions higher than two: combinatorial maps [1], generalized maps [2], or CW complexes [3,4].

Irregular pyramids have a strong biological motivation: They satisfy the biologically motivated architectural and functional requirements of Uhr [5]. The basis of an irregular pyramid need not be an array but any irregular graph as the Delaunay-triangulated [62] human retina (or any subgraph of it). Also, the neural connections in the brain differ strongly from the artificial counter parts that are currently very popular; they are neither fully connected bipartite graphs nor regular local connections. However, there is some similarity in the functionality of the connections with the significant difference that natural neurons are much slower than massively parallel architectures or modern GPUs. This efficiency of the parallel architecture has become very important in the project 'Water's gateway to heaven' where the 2^{33} data can be processed in parallel up and down the only 33 levels of the irregular pyramid.

The irregular pyramids accept as input an image, data from a retina, any plane graph, combinatorial map, or generalized map. The main goal is

to reduce the huge amount of data while preserving certain properties.

The construction and expansion in irregular pyramids is controlled by

- selected contraction and removal kernels,
- massive parallel graph contraction,
- different reduction functions (decoupled from the flexible architecture!),

[k]The term "cell" is just a coincidence with the biological cells in Section 2.2.3, here an abstract cell is meant.

Kropatsch2024a

- inverse operations of contraction and removal: decontraction and reinsertion, and
- termination criteria.

Let us repeat here that the choice of reduction functions can vary not only between different levels of the pyramid but also within the same level in the case that there is a strong hypothesis that a certain object with specific properties is located at a particular location. In this case, a concept similar to the object-oriented programming paradigma can be applied.

For the LBP pyramid, we first translate the binary LBP code into the orientation of edges. This enables the recognition of critical points $\oplus, \ominus, \otimes$ and slopes without the need of derivatives. This follows the principle of changing the representation to solve an insight problem (Section 2.2.5). The observation that continuous curves between extrema are monotonic inspired the choice of the lowest contrast for contracting edges. Experimental results showed that in images the arrangement of critical points plays a dominant role and images can be reconstructed with only a few colors if the structure is preserved. Finally, monotonicity also led in 2D to the novel concept of a slope, opening possibilities to explain the achieved results through integrating bottom-up and top-down processes and the relations between the receptive fields at different levels of the pyramid.

In contrast to many machine learning approaches, like deep convolutional neural networks, which have some architectural similarities with the pyramid but have a strong association between the layer and the applied functionality (i.e., convolution layers and pooling layers), irregular pyramids separate their architecture and functionality. In addition, the construction of the hierarchy enables the architecture to adapt the representation to the structure of the data.

The algorithms operating in the irregular pyramid are designed to work (also) on massively parallel architectures with a parallel complexity of $\mathcal{O}(\log(\text{diameter}))$ following parent–child links.

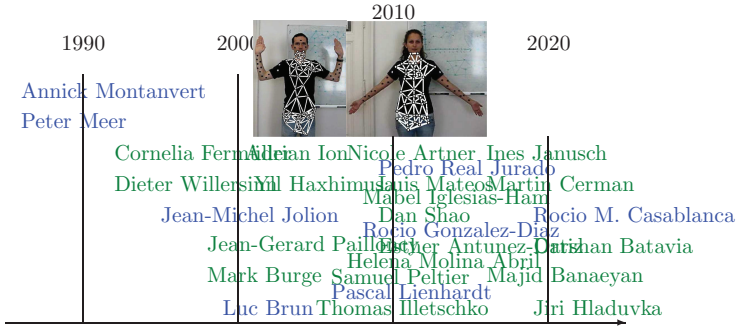
Acknowledgement

Part of this chapter was supported by the Vienna Science and Technology Fund (WWTF), project LS19-013.

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

Contributions by . . .



. . . are cordially acknowledged.
New, future collaborations are very welcome!

References

[1] L. Brun and W. G. Kropatsch, Introduction to combinatorial pyramids, in G. Bertrand, A. Imiya and R. Klette (eds.), *Digital and Image Geometry*. Lecture Notes in Computer Science, Vol. 2243. Springer, Berlin, pp. 108–128 (2001).

[2] G. Damiand and P. Lienhardt, Removal and contraction for n-dimensional generalized combinatorial maps, in H. Wildenauer and W. G. Kropatsch (eds.), *Computer Vision — CVWW'02, Computer Vision Winter Workshop*. PRIP, TU Wien, Wien, Austria, pp. 208–221 (2002).

[3] R. Forman, Morse theory for cell complexes, *Advances in Mathematics* **134**, pp. 90–145 (1998).

[4] R. Forman, Combinatorial differential topology and geometry, *New Perspectives in Geometric Combinatorics* **38**, pp. 177–206 (1999).

[5] L. Uhr, Parallel, hierarchical software/hardware pyramid architectures, in V. Cantoni and S. Levialdi (eds.), *Pyramidal Systems for Image Processing and Computer Vision*. NATO ASI Series, Vol. F25. Springer-Verlag, Berlin, pp. 1–20 (1986).

[6] J. J. Koenderink, The structure of images, *Biological Cybernetics* **50**, pp. 363–370 (1984).

[7] Z. Pizlo, *Solving Solving: Cognitive Mechanisms and Formal Models*. Cambridge University Press, Cambridge (2022).

[8] F. Torres and W. G. Kropatsch, Canonical encoding of the combinatorial pyramid, in Z. Kúkelová and J. Heller (eds.), *Proceedings of the 19th Computer Vision Winter Workshop 2014*. Křtiny, CZ, pp. 118–125 (2014).

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

[9] P. Meer, Stochastic image pyramids, *Computer Vision, Graphics, and Image Processing* **45**(3), pp. 269–294 (1989).
[10] W. G. Kropatsch, Y. Haxhimusa, Z. Pizlo and G. Langs, Vision pyramids that do not grow too high, *Pattern Recognition Letters* **26**(3), pp. 319–337 (2005).
[11] D. Willersinn, *Irreguläre Kurvenpyramiden: ein Schema für perzeptuelle Organisation*, Ph.D. thesis, Vienna University of Technology (1995).
[12] M. Banaeyan and W. G. Kropatsch, Parallel $\mathcal{O}(\log(n))$ computation of the adjacency of connected components, in *3rd International Conference on Pattern Recognition and Artificial Intelligence*. LNCS, Vol. 13364. Springer (2022).
[13] W. G. Kropatsch, Equivalent contraction kernels to build dual irregular pyramids, *Advances in Computer Vision*. Advances in Computer Science, pp. 99–107 (1997).
[14] P. J. Burt and E. H. Adelson, The Laplacian pyramid as a compact image code, *IEEE Transactions on Communications* **31**(4), pp. 532–540 (1983).
[15] J.-M. Jolion and A. Montanvert, The adaptive pyramid, a framework for 2D image analysis, *Computer Vision, Graphics, and Image Processing: Image Understanding* **55**(3), pp. 339–348 (1992).
[16] M. Burge and W. G. Kropatsch, A minimal line property preserving representation of line images, *Computing, Devoted Issue on Image Processing* **62**, pp. 355–368 (1999).
[17] M. Banaeyan and W. G. Kropatsch, Pyramidal connnected component labeling by irregular pyramid, in *5th International Conference on Pattern Recognition and Image Analysis*. IEEE (2021).
[18] J.-G. Pailloncy, W. G. Kropatsch and J.-M. Jolion, Object matching on irregular pyramid, in A. K. Jain, S. Venkatesh and B. C. Lovell (eds.), *14th International Conference on Pattern Recognition*, Vol. II. IEEE Computer Society, pp. 1721–1723 (1998).
[19] M. Banaeyan, D. Batavia and W. G. Kropatsch, Removing redundancies in binary images, in *2nd International Conference on Intelligent Systems & Pattern Recognition*. LNCS. Springer (2022), doi: 10.1109/IPRIA53572.2021.9483533.
[20] M. Banaeyan and W. G. Kropatsch, Fast labaled spanning tree in binary irregular graph pyramids, *Journal of Engineering Research and Sciences* **1**(10), pp. 69–78 (2022).
[21] M. Banaeyan, W. G. Kropatsch and J. Hladůvka, Fast distance transforms in graphs and in gmaps, in A. Krzyzak, C. Suen and A. Torsello (eds.), *S+SSPR 2022*. Lecture Notes in Computer Science, Vol. 13813. Springer Nature, pp. 193–202 (2022).

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

[22] W. G. Kropatsch, Building irregular pyramids by dual graph contraction, *IEE-Proceedings Vision, Image and Signal Processing* **142**(6), pp. 366–374 (1995).
[23] T.-H. Hong and A. Rosenfeld, Compact region extraction using weighted pixel linking in a pyramids, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**(2), pp. 222–229 (1984).
[24] W. G. Kropatsch and H. Macho, Finding the structure of connected components using dual irregular pyramids, in *Cinquième Colloque DGCI*. LLAIC1, Université d'Auvergne, pp. 147–158 (1995).
[25] H. Macho and W. G. Kropatsch, Finding connected components with dual irregular pyramids, in F. Solina and W. G. Kropatsch (eds.), *Visual Modules*, OCG-Schriftenreihe, Österr. Arbeitsgemeinschaft für Mustererkennung, band 81. R. Oldenburg, pp. 313–321 (1995).
[26] W. G. Kropatsch, Properties of pyramidal representations, *Theoretical Foundations of Computer Vision*. Computing Supplement, Vol. 11, pp. 99–111 (1996).
[27] W. G. Kropatsch and S. BenYacoub, Universal segmentation with p*IRR*amids, in A. Pinz (ed.), *Pattern Recognition 1996, Proceedings of 20th ÖAGM Workshop*, OCG-Schriftenreihe, Österr. Arbeitsgemeinschaft für Mustererkennung, band 90. R. Oldenburg, pp. 171–182 (1996).
[28] W. G. Kropatsch and S. BenYacoub, A general pyramid segmentation algorithm, in R. Melter, A. Y. Wu and L. Latecki (eds.), *Vision Geometry V, International Symposium on Optical Sciences, Engineering, and Instrumentation*, Vol. 2826. SPIE, pp. 216–224 (1996).
[29] W. G. Kropatsch and Y. Haxhimusa, Grouping and segmentation in a hierarchy of graphs, in C. A. Bouman and E. L. Miller (eds.), *Computational Imaging II*, Vol. 5299. SPIE, Bellingham, pp. 193–204 (2004).
[30] A. Hanbury, J. Marchadier and W. G. Kropatsch, The redundancy pyramid and its application to image segmentation, in W. Burger and J. Scharinger (eds.), *Digital Imaging in Media and Education, 28th ÖAGM Workshop*, OCG-Schriftenreihe, Österr. Arbeitsgemeinschaft für Mustererkennung, band 179. R. Oldenburg, pp. 157–164 (2004).
[31] W. G. Kropatsch, Property preserving hierarchical graph transformations, in C. Arcelli, L. P. Cordella and G. Sanniti di Baja (eds.), *Advances in Visual Form Analysis*. World Scientific Publishing Company, pp. 340–349 (1998).
[32] W. G. Kropatsch, M. Burge and H. L. Idl, Dual graph contraction for run graphs, in A. Leonardis and F. E. Solina (eds.), *Computer Vision — CVWW'98, Proceedings of the Computer Vision Winter Workshop*. IEEE Slovenia Section, Ljubljana, pp. 75–86 (1998).

Downloaded from www.worldscientific.com by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

[33] M. Burge and W. G. Kropatsch, Contracting line images using run graphs, in M. Gengler, M. Prinz and E. Schuster (eds.), *22nd ÖAGM Workshop on Pattern Recognition and Medical Computer Vision 1998*, OCG-Schriftenreihe, Österr. Arbeitsgemeinschaft für Mustererkennung, band 106. R. Oldenburg, pp. 235–244 (1998).

[34] W. G. Kropatsch and M. Burge, Minimizing the topological structure of line images, in A. Amin, D. Dori, P. Pudil and H. Freeman (eds.), *Advances in Pattern Recognition, Joint IAPR International Workshops SSPR'98 and SPR'98*, Sydney, Australia. Lecture Notes in Computer Science, Vol. 1451. Springer, Berlin, pp. 149–158 (1998).

[35] R. Glantz, M. Pelillo and W. G. Kropatsch, Matching hierarchies of segmentations, in H. Wildenauer and W. G. Kropatsch (eds.), *Computer Vision — CVWW'02, Computer Vision Winter Workshop*. PRIP, TU Wien, Wien, Austria, pp. 149–158 (2002).

[36] R. Glantz, M. Pellilo and W. G. Kropatsch, Matching segmentation hierarchies, *International Journal for Pattern Recognition and Artificial Intelligence* **18**(3), pp. 397–424 (2004).

[37] R. Glantz, R. Englert and W. G. Kropatsch, Contracting distance maps of pores to pore networks, in N. E. Brändle (ed.), *Computer Vision — CVWW'99, Proceedings of the Computer Vision Winter Workshop*. PRIP TU Wien, Wien, Austria, pp. 112–121 (1999).

[38] J. Marchadier, W. G. Kropatsch and A. Hanbury, The redundancy pyramid and its application to segmentation on an image sequence, in C. E. Rasmussen, H. H. Bülthoff, M. A. Giese and B. Schölkopf (eds.), *DAGM Symposium 2004*, Tübingen, Germany. Lecture Notes in Computer Science, Vol. 3175. Springer, Berlin, pp. 432–439 (2004).

[39] N. M. Artner, A. Ion and W. G. Kropatsch, Rigid part decomposition in a graph pyramid, in J. O. E. Eduardo Bayro-Corrochano (ed.), *The 14th International Congress on Pattern Recognition, CIARP 2009*. Lecture Notes in Computer Science, Vol. 5856. Springer-Verlag, Berlin, pp. 758–765 (2009).

[40] W. G. Kropatsch, Abstraction pyramids on discrete representations, in A. Braquelaire, J.-O. Lachaud and A. Vialard (eds.), *Discrete Geometry for Computer Imagery, 10th DGCI*, Bordeaux, France. Lecture Notes in Computer Science, Vol. 2301. Springer, Berlin, pp. 1–21 (2002).

[41] W. G. Kropatsch and Y. Haxhimusa, Hierarchical grouping of non-connected structures, in W. Burger and J. Scharinger (eds.), *Digital Imaging in Media and Education, 28th ÖAGM Workshop*, OCG-Schriftenreihe, Österr. Arbeitsgemeinschaft für Mustererkennung, band 179. R. Oldenburg, pp. 165–172 (2004).

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

[42] Y. Haxhimusa and W. G. Kropatsch, Hierarchy of partitions with dual graph contraction, in E. Michaelis and G. Krell (eds.), *DAGM 2003, 25th DAGM Symposium*, Magdeburg, Germany. Lecture Notes in Computer Science, Vol. 2781. Springer, Berlin, pp. 338–345 (2003).

[43] Y. Haxhimusa and W. G. Kropatsch, Segmentation graph hierarchies, in A. Fred, T. Caelli, R. P. Duin, A. Campilho and D. de Ridder (eds.), *Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshops on SSPR 2004 and SPR 2004*, Lisbon, Portugal. Lecture Notes in Computer Science, Vol. 3138. Springer, Berlin, pp. 343–351 (2004).

[44] Y. Haxhimusa, A. Ion and W. G. Kropatsch, Comparing hierarchies of segmentations: Humans, nomalized cut, and minimum spanning tree, in F. Lenzen, O. Scherzer and M. Vincze (eds.), *Proceedings of 30th OEAGM Workshop*, Obergurgl, Austria. OCG-Schriftenreihe books@ocg.at, band 209. Österreichische Computer Gesellschaft, pp. 95–103 (2006).

[45] P. F. Felzenzwalb and D. Huttenlocher, Image segmentation using local variation, in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 98–104 (1998).

[46] T. Ojala, M. Pietikäinen and D. Harwood, A comparative study of texture measures with classification based on featured distributions, *Pattern Recognition* **29**(1), pp. 51–59 (1996).

[47] C. E. Shannon, Communication in the presence of noise, in *Proceedings of the Institute of Radio Engineers* (IEEE) **37**(1), pp. 10–21 (1949).

[48] X. Tan and B. Triggs, Enhanced local texture feature sets for face recognition under difficult lighting conditions, *IEEE Transactions on Image Processing* **19**(6), pp. 1635–1650 (2010), doi:10.1109/TIP.2010.2042645.

[49] R. Gonzalez-Diaz, W. G. Kropatsch, M. Cerman and J. Lamar, Characterizing configurations of critical points through lbp, in D. M. Onchis and P. Real Jurado (eds.), *Proceedings of the 5th International Workshop on Computational Topology in Image Context, CTIC2014*, Timisoara, Romania (2014).

[50] M. Cerman, I. Janusch, R. Gonzalez-Diaz and W. G. Kropatsch, Topology-based image segmentation using LBP pyramids, *Machine Vision and Applications* **27**(8), pp. 1161–1174 (2016).

[51] L. Latecki, U. Eckhardt and A. Rosenfeld, Well-composed sets, *Computer Vision and Image Understanding* **61**(1), pp. 70–83 (1995).

[52] M. Cerman, R. Gonzalez-Diaz and W. G. Kropatsch, LBP and irregular graph pyramids, in N. Petkov and G. Azzopardi (eds.), *Computer Analysis of Images and Patterns*, Malta. Lecture Notes in Computer Science, Vol. 9256. Springer, Cham (2015).

Downloaded from www.worldscientific.com by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

[53] W. G. Kropatsch, R. M. Casablanca, D. Batavia and R. Gonzalez-Diaz, Computing and reducing slope complexes, in R. Marfil, M. Calderon, F. D. del Rio, P. Real and A. Banderas (eds.), *Proceedings 7th International Workshop on Computational Topology in Image Context*, Malaga, Spain. Lecture Notes in Computer Science, Vol. 11382. Springer, Berlin, pp. 12–25 (2019).

[54] D. R. Martin, C. Fowlkes, D. Tal and J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in *8th International Conference on Computer Vision, ICCV 2001*, Vol. 2, pp. 416–423 (2001).

[55] D. Batavia, R. Gonzalez-Diaz and W. G. Kropatsch, Image = structure + few colors, in A. Torsello, L. Rossi, M. Pelillo, B. Biggio and A. Robles-Kelly (eds.), *S+SSPR 2020*. Lecture Notes in Computer Science, Vol. 12644. Springer Nature, pp. 365–376 (2021).

[56] Z. Wang, A. Bovik, H. Sheikh and E.P. Simoncelli, Image quality assessment: From error visibility to structural similarity, *IEEE Transactions on Image Processing* **13**(4), pp. 600–612 (2004).

[57] L. Zhang, L. Zhang, X. Mou and D. Zhang, FSIM: A feature similarity index for image quality assessment, *IEEE Transactions on Image Processing* **20**(8), pp. 2378–2386 (2011), doi:10.1109/TIP.2011.2109730.

[58] W. G. Kropatsch, R. M. Casablanca, D. Batavia and R. Gonzalez-Diaz, On the space between critical points, in M. Couprie, J. Cousty, Y. Kenmochi and N. Mustafa (eds.), *Discrete Geometry for Computer Imagery*, Marne-la-Vallée, France. Lecture Notes in Computer Science, Vol. 11414. Springer, Berlin, pp. 115–126 (2019).

[59] R. Gonzalez-Diaz, D. Batavia, R. M. Casablanca and W. G. Kropatsch, Characterizing slope regions, *Journal of Combinatorial Optimization* **44**, pp. 1–20 (2021).

[60] D. Batavia, J. Hladůvka and W. G. Kropatsch, Partitioning 2D images into prototypes of slope region, in M. Vento and G. Percannella (eds.), *Proceedings of 18th International Conference on Computer Analysis of Images and Patterns*, Salerno, Italy. Lecture Notes in Computer Science, Vol. 11678. Springer Nature Switzerland AG, pp. 363–374 (2019).

[61] N. Passat and Y. Kenmochi, A topological tree of shapes, in B. E. (ed.), *DGMM 2022*, Vol. 13493. Springer Nature, Switzerland, pp. 221–235 (2022).

[62] B. N. Delaunay, Sur la sphère vide, *Bulletin of Academy of Sciences of the USSR* **7**(6), pp. 793–800 (1934).

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Kropatsch2024a

This page intentionally left blank

Downloaded from www.worldscientific.com
by Walter Kropatsch on 10/07/24. Re-use and distribution is strictly not permitted, except for Open Access articles.