Technical Report

Pattern Recognition and Image Processing Group
Institute of Computer Aided Automation
Vienna University of Technology
Treitlstr. 3/1832
A-1040 Vienna AUSTRIA
Phone: +43 (1) 58801-18351
Fax: +43 (1) 5054668
E-mail: krw@prip.tuwien.ac.at
URL: http://www.prip.tuwien.ac.at/

PRIP-TR-2 October 29, 2006

Image Pyramids and Curves

An Overview

Walter G. Kropatsch

Abstract

This technical report represents the first part of my habilitation (University of Innsbruck, Austria, 1990). It contains the quintessence of my scientific work in the years 1984-1989. The second part of the habilitation consists of a series of my own publications to the subject. It was not reproduced here since the papers already appeared separately.

The present report offers a condensed survey over the international literature in this branch, arranged and commented according to a personal weighting in order to relate it to my works on the one hand and on the other hand, to put the significance of the single publications in the greater context of the current state of the art in computer vision.

Pyramids are important structures in the processing of digital images. Invented as an ordered collection of images at multiple resolutions they have developed into efficient data and processing structures. They give us the hope to achieve acceptable performance in vision tasks that have to process millions of bytes in extremely short time.

Curves represent shapes at an intermediate level. They describe either the boundary of a region or the central axis of an elongated region in the image. Hierarchical curve representations aim at a stepwise reduction of the data to 'significant parts' like corners or curvature extrema while preserving the major property of a curve: its connectivity.

Contents

Zusammenfassung

Dieser technische Bericht stellt den ersten Teil meiner Habilitationsschrift (Universität Innsbruck, 1990) dar. Sie enthält die Quintessenz meiner wissenschaftlichen Arbeit in den Jahren von 1984-1989. Der zweite Teil der Habilitation besteht aus einer Reihe eigener Publikationen zum Thema. Er wurde hier weggelassen, da diese Beiträge separat erschienen sind.

Der vorliegende Bericht bietet einen zusammenfassenden Überblick über die internationale Literatur zu diesem Fachbereich, allerdings nach persönlicher Gewichtung geordnet und mit Kommentaren versehen, um sie einerseits in Beziehung zu meinen Arbeiten zu setzen, andererseits aber die Bedeutung der Einzelpublikationen in einen größeren Zusammenhang zu stellen, und zwar den des Fachgebietes "Computer Vision".

Pyramiden stellen wichtige Strukturen bei der Bearbeitung digitaler Bilder dar. Erfunden als geordnete Sammlung von Bildern in verschiedenen Auflösungen haben sie sich zu effizienten Daten- und Verarbeitungsstrukturen entwickelt. Sie geben uns die Hoffnung auf akzeptable Leistungen bei Aufgaben des Computersehens, bei denen Millionen von Bytes in extrem kurzer Zeit zu verarbeiten sind.

Kurven stellen Formen in Bildern auf einer mittleren Ebene dar. Sie beschreiben entweder den Randverlauf einer Region oder die Hauptachse einer länglichen Region im Bild. Hierarchische Kurvenrepräsentationen haben das Ziel, die Datenmenge schrittweise auf 'signifikante Teile' wie Ecken oder Krümmungsextrema zu reduzieren, während gleichzeitig die Haupteigenschaft einer Kurve, ihr Zusammenhang, erhalten bleibt.

1 Introduction

This report gives an overview of the state of the art in the field of computer vision embedding the new pyramid structures and curve representation schemes. It is combined with a guide to the related literature. It represents a methodical preparation and combines branches of computer vision that have been treated separately in the past. In particular following research areas are covered and aggregated to show several underlying general principles:

- The role of abstraction and of different levels of processing in image analysis.
- New pyramidal structures and their properties.
- Various numerical and symbolic computations in a pyramid.
- Efficient multi-scale representations.
- The role and representation of curves
- in continuous scale-space and
- in discrete curve pyramids.
- Cooperation between both dual pyramids and complementary representation schemes.

The intention of this document is to summarize the own research contributions in the contextof the important works in the related fields and to explain some of the underlying key principles. The survey is certainly not complete and can sometimes only scratch the surface of the problems. A compendium of references to the related literature should help to find specific information in more detail. An index lists the citations with the referencing page numbers and should provide a convenient way to localize the related sections in the text.

Since images are digitized and stored in computers many techniques have been developed to process and analyze them. Although a lot of interesting results could be achieved so far, we are not yet arrived at a stage where we could say that **the computer can see**. The challenge in the area of computer vision is certainly that humans are able to perform vision in so many tasks nearly without any effort.

The problems and goals of analyzing digital images by computer algorithms will be described in chapter 2. It shows the special need for hierarchical structures to convert the huge array of numbers, the image, into a sort of description of what is observed in the image. Chapter 3 introduces pyramidal data and processing structures and gives also a biologically plausible motivation. Many parallel numerical algorithms need only $\mathcal{O}(\log n)$ steps to compute results like sum, average, or variance (chapter 4). Special emphasis is put on properties of different pyramidal structures, on regular (chapter 5) and on irregular ones (chapter 6). Gray level pyramids contain low-pass filtered versions of the high resolution input image (chapter 7) and in Laplacian pyramids (chapter 8) the levels contain band-pass filtered images at multiple resolutions. The wavelet representation is a recent extension of Laplacian pyramids (chapter 9).

Curves are derived during the analysis process and represent shapes of different origins (chapter 10). Continuous (e.g. scale-space) and discrete (e.g. chain code) models are presented with emphasis on possibilities for hierarchical processing. The cells of the curve pyramids in chapter 11 represent curve segments ordered by their length. The concept of dual pyramids (chapter 12) combines both gray level and curve pyramids in a cooperational model.

2 Analysis of Digital Images

The general goal of image analysis (also known under the term Image Understanding in Artificial Intelligence) is to find semantic interpretations of images, in particular to localize and name objects contained in a scene and to assess their mutual relationships. The central problem is to recognize known objects reliably, independent of variations in position, orientation, and size, even when those objects are partially occluded. Although some information is lost by the image formation process, biological vision systems demonstrate that this task can be accomplished reliably and efficiently even under difficult viewing conditions.

To **see** the world, the human eye is equipped with approximately 130.000.000 receptors and 12.000.000.000 neurons in the brain. 'Technical' characteristics of the biological vision system are: 1-2 milliseconds are needed to forward data from one neuron to the next; and often 1 second is sufficient for a human to recognize a complex scene (like a busy street) and to react appropriately. With these time constraints it can be concluded that only several 100 massively parallel processing steps (neuron-to-neuron connections) must suffice for the task [Uhr86].

Further comparisons between human thinking processes with similar processes in computers are drawn in [Ger87]. This book gives a good overview of the neurophysiological foundations. The information processing of the neurons is explained in terms of notions from computer science. Neuron structures form hierarchies similar to pyramids as introduced in chapter 3. Coding and decoding of information is shown as the principle of mapping and storing events in the human brain. A similarity principle assigns similar events to similar codes (compare with 'matching' below). Both recognition and learning make use of this principle.

Most of the current approaches in image analysis can be described as a sequential multi-stage process (Fig. 1). For detailed information on image processing and analysis we refer to standard books like [Moi80], [RK82], [Mar82], [Ser82], or [Jae89]. This multi-stage process can be seen as a step-wise translation that is guided by the interpretation goal and constrained by its knowledge about the external world and about the available vocabularies at different processing levels. To correctly interpret a given image it is important to know the classes of OBJECTS that can be expected in the WORLD, and what the properties of their REFLECTING SURFACES are. In the image formation stage, the raw image data are collected by a sensing camera, which outputs an array of intensity values. This DIGITAL IMAGE is processed to produce a set of data primitives, often called FEATURES. Similar features are then assembled to more complex structures (e.g. REGIONS). A survey on image segmentation techniques can be found in [HS85]. SPECIFIC image PARTS can be derived from those regions by e.g. geometrical properties. They are useful components for OBJECT models and SCENE DESCRIPTION. In the final phase, these complex features are matched against existing object models in a data base (e.g. OBJECTS in the WORLD vs. OBJECTS in the SCENE).

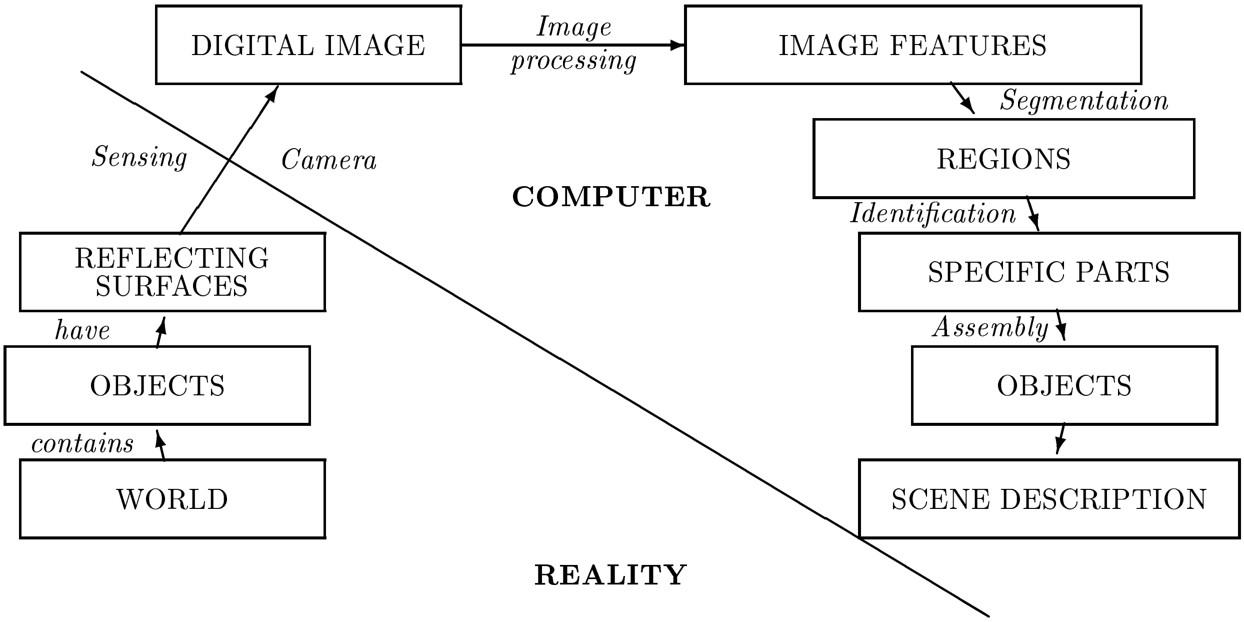


Figure 1: Processing stages in digital image analysis

Some of the pictorial entities, their information content, and the operations that can be performed at different processing levels are summarized in Table 1.

There exist many different methods to analyze digital images, many sceneries of application, many different vocabularies in which the final description should be expressed, but a very simple data structure at the input side. Hence the need to introduce abstract classes of operation, different levels of representation. Even the world shows many different aspects. To deal with this huge variety of data and to focus the processing to the final goal, mechanisms are necessary to agglomerate and compact the data, probably repeated in several stages, until the amount of data is small enough to make final decisions.

Abstraction is a powerful data fusion process. It enables individuums and processes to differentiate between **important** and less important informations, methods, representations etc. Such generalizations allow to treat all the elements of a general class in the same way. When successively applied, they imply a hierarchical structure with different levels

- of concepts for representing knowledge about the world, e.g. the conceptual hierarchy in [BR80],
- of representation (see below),
- of processing stages (see below), e.g. hierarchies of invariance in cognition [Bal86], and
- in the complexity of processing images.

Finally we address the problem of controling the processes in a vision system and the methods to accumulate evidence.

Table 1: Pictorial entities at different levels of processing

Entity	information content	examples for operations
Picture	imaging conditions, geometry	sampling, rectification
Pixel	gray value / color vector	enhancement, classification
Neighborhood	spatial locality	shrink, expand
(Step) edge	magnitude, orientation	edge detection and linking
Region	homogeneity, connectivity	segmentation
Boundary	shape	connecting continuous curve segments
Image Part	specific image properties	property measurement
Object Part	specific object properties	property matching
Object	functionality	relational matching
Situation	specific configuration of objects	interpretation
Scene	visible situations of the world	description

2.1 Levels of Abstraction

Abstraction is a scientific method for isolating facts, for generalization, or idealization. It has the characteristic to neglect details in favour of important features. Abstracted objects form generic classes which have common properties but differ in detail. Since there is no unique way for abstraction it may depend on functional goals (see [FB86]). Examples for abstraction in digital image analysis are listed in Table 2.

Table 2: Abstraction in digital image analysis

operation	neglected detail	preserved feature
projection	higher dimension	objects, properties
smoothing	high signal frequences	low signal frequences
reduced resolution	small (noise) regions	large regions
thresholding/quantization	grey value	binary shape
edge detection	absolute grey tone	significance, orientation
segmentation	individual pixel variations	size and homogeneity
identification	appearance model	generic properties and relations

Advantages of abstraction are: data reduction; reduced processing time by divide and conquer, entier generic classes are handled at once, differencing between class members only when necessary. As a consequence one tries to build abstraction hierarchies that have the above advantages and that allow also the necessary distinction between individual members.

Generalization has a long history in cartography. It is a necessary tool when maps at different scales have to be produced. Nowadays map data are stored in computers by geoinformation systems (GIS). In [Ric88] classical generalization is divided in two functional subareas: editorial generalization and technical generalization. Generalization in a GIS may involve the following:

1. feature selection (elimination)
2. linear simplification (smoothing)
3. symbolization
4. feature amalgamation
5. feature displacement

An image pyramid is a typical example for an abstraction hierarchy in image analysis. We shall develop an extended pyramid scheme in which one cell of the pyramid structure may contain more than just one grey value, e.g. edge information or a symbolic curve code.

2.2 Representation levels

The 'connection table' allows the transition between the different levels in [TR87]. Five different representation levels are identified from the real to the cognitive world:

- 2D image, 'image-based';
- 3D skeleton, 'feature-based', lexical level;
- connection table, 'part-based', syntactic level;
- object description language, 'model-based', semantic level;
- natural language, 'language-based'.

Basic descriptive notions are (comp. Fig. 1): objects - parts - primitive parts. The connection table describes the way in which parts form an object.

In [Kro88d] the fields of (scientific) visualization (image synthesis, a subarea of computer graphics) and image analysis, although different in their goals, are related through common intermediate levels of representation. Visualization generates an image from a computer stored description, digital image analysis is supposed to produce descriptions of a digital image. Descriptions at different levels of abstraction form the basis for both fields. Following levels are identified:

1. 2D digital image with pixels;
2. image segments such as region, edge, or texton [JB83];
3. image segments with specific properties such as generalized cylinders;
4. fragments, parts of objects, 'GEON' [Bie87];
5. objects, models;
6. functional areas [MM83];
7. natural language like in [TR87].

Images are a medium for communication. For evaluating the quality of a visual information transmission process four models are presented.

It is concluded that pyramids have several of the required properties of a visual system to approach a performance comparable with humans.

2.3 Processing levels in image analysis

Non-accidental properties allow 3-D space inference from 2-D image features in [Bie87]: e.g.

1. collinearity;
2. curvilinearity;
3. symmetry;
4. parallel curves;
5. curves terminate at a common point.

From contrasts in these five non-accidental relations, specific members of the set of **generalized cones** can be differentiated: **geons**. Two or three geons suffice to unambiguously represent most objects.

The processing stages of 'Recognition-by-Components' for object recognition are:

- edge extraction;
- detection of non-accidental properties;
- parsing at regions of concavity;
- determination of components;
- matching of components to object representations;
- object identification.

Tsotsos' model considers different levels of complexity [Tso87]. Analysis of the complexity level is necessary to ensure that the basic space and performance constraints observed in human vision are satisfied. The **'maximum power / minimum cost principle'** ranks the many architectures that satisfy the complexity level and allows the choice of the best one. A **receptive field** is defined as the area of the visual scene in which a change in the visual stimulus causes a change in the output of the processor to which it is connected (comp. chapter 5). Hexagonal images with hexagonal pixels are assumed and lead to a **hexagonal pyramid structure**. In summary, the characteristics of a visual processing architecture are:

- spatial parallelism;
- hierarchical organization;
- localization of receptive fields;

- visual stimuli lead to logically separable maps;
- abstraction of the input token arrays.

Predictions:

- processor columnar organization;
- tokens of visual parameters at high resolution can be obtained only by tuning of computing units and through the input abstraction hierarchy;
- token inseparability (or coarse coding).

2.4 Control mechanisms and evidence accumulation

Matsuyama characterizes the state-of-the-Art in the field [Mat88]: "The expertise stored in the systems is what we, image processing researchers, have acquired and accumulated through the development of image processing techniques." Four classes to combine complex image analysis processes are distinguished:

- consultation;
- program composition;
- design for image segmentation;
- goal directed image segmentation.

Two methods of representing image analysis strategies are proposed:

- software engineering;
- knowledge representation.

Many problems are encountered in designing image analysis processes:

1. Assessment of image quality.
2. Selection of appropriate operators.
3. Determination of optimal parameters.
4. Combination of primitive operators.
5. Trial-and-Error experiments.
6. Evaluation of analysis result.

Matsuyama differentiates between IUS (image understanding systems) and ESIP (expert systems for image processing) by means of the knowledge sources mainly used in the two different kind of systems.

The ESIP-architecture has the following major objective: formulate and describe visual information (e.g.image features, properties, and relations). ESIP should combine **qualitive symbolic reasoning** and **quantitative signal processing**. The reasoning consists of :

1. Analysis plan generation
2. Operator selection and parameter adjustment

Four types of ESIP are identified:

A: Consultation system: EXPLAIN. Initiated by a user request the system develops a processing plan as an ordered sequence of abstract image processing algorithms, which are then instantiated step by step allowing user interaction and backtracking.

B: Knowledge-based program composition system. Here, the user writes an "abstract program specification". For example, DIA-Expert uses the operation tree. The level of the operation tree means the level of abstraction. At each level a sequence of image analysis algorithms is described. The sequence of the bottom of the tree (i.e. leaf nodes) represents that of executable software modules in the program library.

C: Rule-based Design System for Image Segmentation Algorithms.

D: Goal-Directed Image Segmentation System. LLVE is originally the low level part of SIGMA [HLM86]. There, the user specifies the goal in a graph consisting of image features (as nodes) and transfer processes as arcs.

The software engineering strategy uses three types of heterogeneous compositions:

1. Composition of multiple analysis results.
2. Mask controlled operation.
3. Parameter optimization.

As an extension to LLVE's single resolution image features Matsuyama proposes as second strategy to represent the system's knowledge in a **hierarchical hypergraph**. Using a resolution parameter it can be compressed for storage purposes into a single layered hypergraph which is expanded into the multilayered hypergraph during search.

2.4.1 SIGMA

SIGMA is a system developed at the University of Maryland. The goal of this research [HLM86] is to develop a robust control strategy for constructing image understanding systems (IUS). This paper proposes a general framework based on the integration of "related" hypotheses. Hypotheses are regarded as predictions of the occurrences of objects in the image.

Related hypotheses are clustered together to accumulate evidence. A "composite hypothesis" is computed for each cluster. The goal of the IUS is to verify the hypotheses. An image understanding system, called SIGMA, has been constructed based on this framework and demonstrated its performance on an aerial image of suburban housing development.

2.4.2 PSEIKI

The paper [KHK88] surveys the planning and reasoning research being carried out in the Robot Vision Lab at Purdue. In particular, it describes the working of a new planning system called SPAR, which uses a constraint posting approach for simultaneously fulfilling the operational, geometric and uncertainty reduction goals, and the PSEIKI system for evidential reasoning in a tangled hierarchy. The authors also mention briefly their other related research in high precision assembly under forceltorque control and robotic manipulation with structural stereopsis for depth perception.

3 Image Pyramids

An important parameter of a digital image is its **resolution**. The spatial resolution is described in terms of the smallest dimension of the object that can just be discriminated. In an image with high resolution many detail objects can be observed, at low resolution only large objects are recognized. A *resolution cell* is the smallest most elementary areal constituent in a digital image. The area of a resolution cell is usually a square or a rectangle. Other shapes are possible and will be discussed in chapter 5. The position of a resolution cell within the 2D plane is determined by the coordinates (x, y) of its center. How many cells are needed to cover a given image area depends on the **size** of the cells. Table 3 compares the qualitative consequences of different cell sizes.

Table 3: Image qualities of different resolutions

	small cells	large cells
resolution	high	low
data amount	huge	smaller
computing times	(very) long	(relatively) short
details	rich and many	very few if at all
overview	bad	good
precision	high	low

Tanimoto [Tan86] defines a **pyramid** as **a collection of images of a single scene at different resolutions**. The images of this collection can be ordered according to their cell sizes and numbered as **levels** of the resulting ordered set of images.

Two terms describe the structure of a (regular) pyramid: the **reduction factor** and the **reduction window**. The reduction factor determines the rate by which the number of cells decrease from level to level. The reduction window associates to every (higher level) cell a set of cells in the level directly below. In general the reduction window covers the area of its

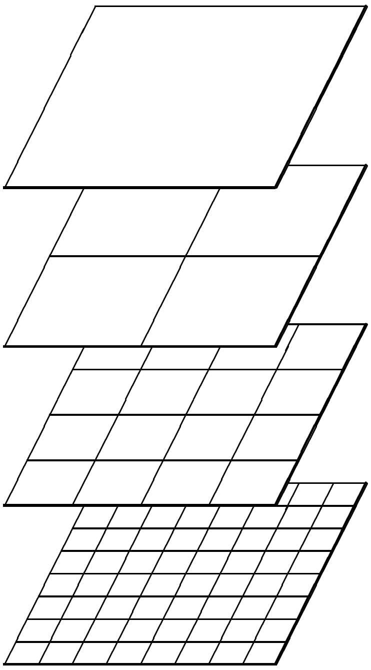


Figure 2: Classical 2 × 2/4 pyramid

associated cell. In square or rectangular images reduction windows are mostly rectangularly shaped and can be described by $(number\_of\_columns) \times (number\_of\_rows)$.

In the *classical pyramid* (Fig. 2) every 2 × 2 block of cells is merged recursively into one cell of the lower resolution. We formally describe this structure by 2 × 2/4 which specifies the 2 × 2 reduction window and the reduction factor of 4. This type of pyramid has been extensively studied (e.g. [TK80], [Ros84]).

A quadtree (e.g. [Sam85]) is a similar tree structure. The cells and connections of the quadtree form a substructure of the 2 × 2/4 classical pyramid. Pyramid and quadtree mainly differ in the fact that a quadtree has no communication links between cells of the same level.

Tanimoto's formal definitions refer to this type of pyramid [Tan88]. He defines a cell (Tanimoto uses the term *pixel*) in a pyramid as a triple (x, y, v) which is defined in a **hierarchical domain** of $L + 1$ levels:

$$\{(x, y, v) | 0 \leq x \leq 2^v, 0 \leq y \leq 2^v, 0 \leq v \leq L\} \tag{1}$$

Then a pyramid is any function whose domain is a hierarchical domain. This function assigns to every cell in the simplest case a value, but also structures of higher complexity can be stored. This definition of a pyramid will be generalized in the next section.

3.1 The contents of a pyramidal cell

If the value of the (pyramidal) function represents a gray value we call it a **gray level pyramid**. It was the model used in Hong's PhD-thesis [Hon82]. Different construction methods for such pyramids will be introduced in section 7.

Hartley [Har84b] allows a cell to contain more than just one (gray) value:

’. . . each cell in the pyramid contains a more complicated **model** *of the region which it represents.’* Examples of hierarchical (parametrical) models for extracting edges, (smooth) curves and lines, corners, Glass and flow patterns, and textures are used within the pyramid framework. The parameters of the models are best fitted in a least square optimization by building the necessary sums in the pyramid in logarithmic time complexity. (See section 4.) A main advantage of such pyramidal computations is pointed out: ”relative locality”.

Ahuja [AAS85] differentiates between two broad categories of image representations: (1) those which describe the interiors of the cells and (2) those which specify their borders. Most of the pyramidal approaches belong to category (1).

A concept that falls in category (2) has been introduced for curve representation in pyramids [Kro85a]. The details of this approach will be described in more detail in section 11. In this concept, a pyramidal cell is considered as a window through which the underlying image is observed. Objects that are completely within the field of view of a cell can be recognized and described by this cell. Objects (e.g. curves) that are only partly covered must cross the boundary of the **observation window**. Hence the entier object cannot be recognized at this resolution level (of the pyramid). Information necessary to connect the observed part of the object with the parts in the adjacent cells must be passed up to the next lower resolution level (or equivalently, to the next higher pyramid level). There, the cells cover a larger area and can join some parts of the level below. This process is repeated up to successively lower resolutions until the whole object is within the observation window of a cell.

More formally, this concept defines the cell that **represents** pictorial entities (e.g.primitive image parts, objects, configuration of objects) within a pyramid:

A pictorial entity of an image (stored in the base of a pyramid) should be represented in that cell of the pyramid that satisfies following conditions:

1. The region corresponding to that cell in the image space (that is a subarea of the base level in discrete reprsentations) covers the pictorial entity completely.
2. No smaller cell in the pyramid fulfills property (1).

This definition assigns to every pictorial entity a unique cell in the pyramid in which it should be represented. Unfortunately, in some pyramid structures a small rigid motion (shift, rotation) of the object may cause a completely different representation (the representation cell may be many levels below or above). We therefore require as pyramidal representation goal that the pyramid should order the pictorial entities by a specified property like size, length, etc. such that a certain property can be related directly to a bounded range of levels in which the entity must be represented if it appears in the image.

3.2 Operations in a pyramid machine

The operations in a pyramid are mostly **local**. The new value of a cell may be computed using as input either external parameters (constants) or values from the cell’s contents or values from the neighboring cells in the pyramidal structure. These are the neighbors within the same pyramid level, the cells of the next lower level which are within the reduction window of the cell (called **sons**), and the cells of the next higher level (called **fathers** or **parents**). Every cell except the top has at least one parent cell, but may have more than one in overlapping pyramids, where a cell may be part of several reduction windows (see section 5 for more detail on different pyramid structures).

An important class of operations is responsible for the bottom-up information flow within the pyramid: the **reduction function**. It computes the new value of a cell exclusively from the contents of its sons. Given an image in the base of the pyramid, application of a reduction function (e.g.average) to all first level cells fills this level. Once the cell of the first level received a value, the same process can be repeated to fill the second level and so on to the top cell.

Further operations depend on the architecture of the pyramid and will be treated in one of the two following subsections. Two architecturally different types of pyramid machines are distinguished: (1) a mesh-connected multi-processor architecture, a realization of a special, generalized cellular automaton [Ros81], and (2) a pipeline architecture.

3.2.1 Mesh-connected multi-processor architectures

In a mesh-connected multi-processor pyramid machine there exists one processing element per cell. Since Tanimoto's machine [Tan86] is a *bit pyramid* where every cell only contains a binary value, the basic operations are the binary operators AND, OR, and NEGATE with some additional instructions like LOAD, STORE, and COPY. The most interesting operation in this set is a **general matching operation**. For any cell a vector $\nu = [X(N_1), X(N_2), \ldots]$ of the values of all the neighboring cells N_i is built. The operand of the operation is a vector $P = [P_1, P_2, \ldots]$ taking values 0, 1, and D. Tanimoto defines an AND_Match and an OR_Match operation. They both combine first the elements of vectors ν and P logically by a point operation $\odot^*$. The resulting binary values are then summarized by either a logical AND or by a logical OR operation:

matching operation	point operation $\odot^*$	$x \odot^* y = 1$ if: (0 otherwise)	log. combination
AND_Match	$\wedge^*$	$x = D$ or $x = y$	$\bigwedge_i (P_i \wedge^* X(N_i))$
OR_Match	$\vee^*$	$x = D$ or $x \neq y$	$\bigvee_i (P_i \vee^* X(N_i))$

Other pyramid machines have been proposed and/or built [Mal86], [MCM$^+$86], [SH86]. They use similar operations in their machines.

3.2.2 Pipeline architectures

The RCA group [vS85] has built a pipelined preprocessor hardware to generate Gaussian and Laplacian pyramids using 2 programmable Multibus (Intel) boards with a memory update in 1/30 sec.

Burt [BASv86] identifies four basic (numerical) pyramid operations for this concept:

- filter convolution,

- point operations,
- image - image operations, and
- local integration.

Pyramid construction operations include

- filtering F,
- decimate or reduce steps R, and
- expansion E.

With these operations the levels $G_i, i = 0.., n$ of a (Gaussian) pyramid (see section 7) are generated by following iterative process:

$$G_0 := I; G_{i+1} \ := R(F(G_i)) \tag{2}$$

The pyramid machine is a pipeline with five computational units:

- a filter with kernel size 5×5;
- a decimator to reduce the resolution by a factor of 4;
- an expander, to insert zeros;
- an ALU for sum and difference of images; and
- several image frame stores.

A comparison with a bit serial mesh-connected multi-processor architecture shows that the mesh machine would need 5000 chips to do the same pyramid operations as 100 chips used for the pipelined approach.

3.3 Biological motivation

In 1984 Leonard Uhr [US84] introduced the notion 'Recognition Cone' to describe the characteristic of a recognition system similar to biological systems. The only conclusion that can be drawn from the efficiency and the known constraints of natural recognition systems is that such systems must process the data massively parallel. Uhr's recognition cone captures the necessary structure of the system: it receives at the bottom a huge amount of visual stimuli, processes them in parallel while reducing at the same time the data amount step by step until final recognition takes place by consulting only a few of the remaining data.

Rosenfeld [Ros87b] proposed an approach for recognizing unexpected objects in a scene where a pyramid forms the basic communication structure.

First he makes conjectures on how humans represent and describe:

1. 3D object representation by characteristic views;

2. ’Primitive’ parts: pieces of regions, pieces of boundaries;
3. Describe parts by local property values, or combinations of such values;
4. Relations for combinations are relative values of properties;
5. Classes of objects are characterized by unidimensional constraints.

Hence the general structure of the system:

1. Bottom: hypercube or pyramid;
2. Part properties are broadcast to
3. a set of object processors.

In [BASv86], Burt gives his motivations for image recognition in pyramid machines: ’..as pattern information becomes more complex, position information becomes less precise: there is a gradual shift from encoding ”where” to encoding ”what”.’ These ideas have been further developed and are summarized in Burt’s survey paper [Bur88] at the ICPR 1988. The following subsection brings some of the basic ideas of this paper. Although it refers to several concepts that will be introduced in the subsequent chapters, it is not split among the different chapters to keep the compactness of the concept.

3.4 The pyramid as a model for real time computer vision

Systems for real time computer vision must locate and analyze just that information essential to the task at hand, while ignoring the vast flow of irrelevant detail. Attention mechanisms support efficient, responsive analysis by focusing the system’s sensing and computing resources on selected areas of a scene. (Compare with O’Rourke who has implemented an attention control mechanism using a dynamic k-d-tree [O’R81].)

Three elements of attention in computer vision are described: foveation, tracking and high level interpretation.

In human vision, acuity falls roughly in inverse proportion to the visual angle, θ, from the fovea. The non-uniform distribution of acuity in the human eye represents a balance between conflicting requirements. It provides both a wide field of view and high acuity.

A rudimentary fovea is formed within the Laplacian pyramid (see section 8). Pyramid levels are indicated in one dimension as rows of sample points, with sample density reduced from level to level. After constructing the pyramid from an original image, pyramid data is discarded (set to zero) outside a central zone at each level, Figure 3. Because the region covered by each resolution level of the pyramid is double that of the next higher level, this implements a discrete approximation to a $1/\theta$ acuity distribution. In Figure 3 the width of the central zone is just 5 samples at each level, although 16 or 32 are typical in computer vision applications. This approximates what a human eye is able to resolve.

It is possible to represent image detail at much reduced resolution through the use of **integrated feature measures** [BHR81]. An original image is first decomposed into a set of band pass components through Laplacian pyramid construction. The band (level) best

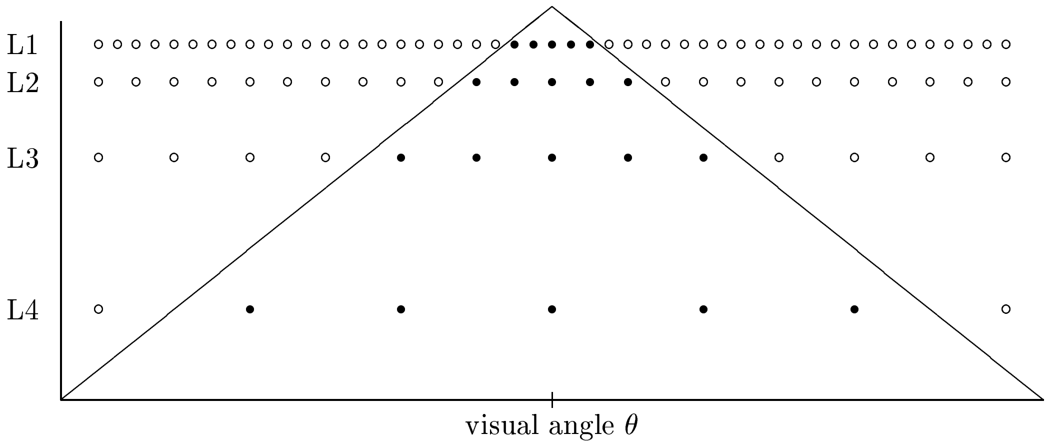


Figure 3: Formation of an electronic fovea

matched in scale to a feature of interest is selected for further processing. This is convolved with a filter, f, selective for the feature. Samples of the filtered image are then squared. The integration is achieved through construction of a second, Gaussian pyramid, with the filtered and squared image as its base level. A positive value of a sample of the integrated measure indicates that the feature selected by the filter is present in the image, but its position within the image is encoded at low resolution. When a critical feature or combination of features is detected at low resolution, the system follows a coarse-to-fine homing procedure to examine the indicated region of the image more closely.

In human vision there are two distinct classes of eye movements, **saccades**, trough which the eye brings critical regions of a scene into a foveal vision, and **pursuit movements**, through which it follows moving objects in order to stabilize their images on the retina. Tracking is a key element in the control of selective, attention-based analysis in computer vision, as it is in human vision. Tracking may be said to isolate selected regions of a scene in **time** just as foveation isolates regions in **space**.

Tracking begins with the estimation of background motion. Frames 1 and 2 are correlated to obtain a rough estimate of motion. Because correlation search is costly, analysis is performed first at reduced resolution. A pyramid is constructed for each frame, and a level k is chosen for analysis at which the sample distance, 2^k, is expected to be larger than displacement due to image motion. At this level, correlation needs to be computed only for displacements of plus or minus one sample distance in the horizontal and vertical directions. Frame 2 is shifted by an amount equal to the estimated motion, then it is compared to Frame 3. Since the estimate was obtained at reduced resolution, and hence is imprecise, the shifted Frame 2 will not exactly cancel motion between Frames 2 and 3. It will, however, reduce the magnitude of this motion. The correlation analysis is repeated for the shifted Frame 2 and Frame 3, but now at the next higher pyramid level, $k-1$. These steps are repeated for Frames 3-4, then 4-5, and so on.

The techniques described here illustrate two important aspects of tracking in dynamic vision. First, relatively crude computations within a feedback loop can achieve very precise estimates of background motion. Second, when background motion is nulled through tracking, subsequent detection of object motion is greatly simplified. When a human performs a vision task he 'knows where to look'.

One approach to implementing reasoning procedures is based on fast, hierarchical structured search. To implement this type of search we define an object representation called a **pattern tree**. The object pattern is decomposed into distinctive pattern components, or at various resolutions within a pyramid structure. The representation is constructed by first forming a full pyramid for the original pattern. Component patterns are then selected as small arrays of samples taken from this pyramid. Typically, large pattern components are represented at low resolution, while small components are respresented at high resolution. The links between components define their relative positions within the overall pattern.

Search is then formulated as a sequence of simple pattern matching steps.

Attention is presented here as essential control mechanisms for dynamic vision systems that perform real time tasks in an ever changing environment. Such systems are confronted with far more visual information than any practical system can process in real time. Attention mechanisms allow the system to function in a data rich world by directing system sensing and analysis resources to just that information critical to the system's current visual task.

3.5 German articles

A german introduction to image pyramids from the signal-theoretic point of view is given in [Jae89, chapter 7].

3.6 Bibliographical Notes

Good survey's about image pyramids and its applications can be found in [TK80], [AAB$^+$84], [OS87]; [Ros84] is a collection of papers dealing with pyramids.

Comparisons of pyramidal architectures with different other architectures like arrays and hypercubes are given in [Tan72], [Bes86], [Can86], [Cas86], [Duf86], [Fri86], [Sto86], [Uhr86], [Fou88], [Lev88], [SC88].

Many systems for image analysis use recently pyramid representations, e.g.

- Neveu for recogizing 2D objects [NDC86];
- PSEIKO [AK88a], [AK88b] for spatial reasoning;
- Jiang for detecting thresholds [JMP88];
- Kalvin [KPH88] for image segmentation;
- Shapiro [SL88] in a CAD-to-Vision system for relational representation.

Languages to program pyramidal architectures can be found in [Di 86], [Lev86].

Ebner [EF86b], [EF86a] applies a multigrid (which is the mathematically equivalent to multiresolution in image processing) method to digital elevation models.

Some research objectives of image pyramids have been compiled in [Kro88b].

This survey summarizes some of the actual benefits of image pyramids. 'Pyramid structures represent information about an image at a set of exponentially decreasing scales. The represented information takes the form of a series of abstractions that describe the contents of the image.'

Pyramids can perform numerical computations in a very efficient way. All the basic statistical computations such as summing, counting, averaging, take on the order of $log(n)$ steps. Extensions of the basic operations such as least square fitting, statistical moments, or the detection of 'coarse' features show similar efficiency.

Edge detection and contour coding is a step from numerical to symbolic representation and computation. The curve pyramid and Hartmann's hierarchical structure code [Har86b] are examples of pure symbolic reduction processes.

The concept of the 'dual pyramids' combines both numerical and symbolic computations and allows free local flow of information within the structure.

4 Numerical Computations in a Pyramid

The claim is made that pyramids are efficient computational structures [BHR81]. [BHR81] is a historical reference that elaborated the potential of pyramids for calculating image region properties efficiently.

Rosenfeld [Ros86b] gives an overview. He distinguishes between two main flow streams in the pyramid: bottom-up summarizing and top-down delineation. Computations include intensity (gray level pyramids), where modality detection and fitting are typical examples, and contour recognition, where blobs and ribbons need different treatment.

4.1 Calculating sum and mean

Let us illustrate the principles of operation by means of a simple example using a 1D (linear) pyramid. Our goal is to calculate the sum, the mean, and the variance of the 8 pixels in the base of our four level pyramid (Fig. 4). The process is very similar to **counting** the number of pixels of a binary image, as described in [SH86] for the GAM-pyramid. To compute the **sum**, we proceed bottom-up:

1. Four parallel additions $\boxed{+}$ produce the sums $\boxed{}$ at level 2;
2. two further additions $\boxed{+}$ are needed to fill level 3; and
3. the final result at the top level 4 is the sum of the two values $\boxed{12}$ and $\boxed{20}$ at level 3.

Seven additions are needed and the sum of our 8 ($= 2^3$) pixels is computed in 3 steps.

The results in intermediate levels represent sums of the receptive fields of all cells.

The **mean** $\underline{\mu}$ is defined as the sum divided by the number of elements. We can either calculate the size of the receptive field, store it in a separate entry of every cell, and build the quotient as illustrated in Fig. 4. An alternative to the above procedure, where large values

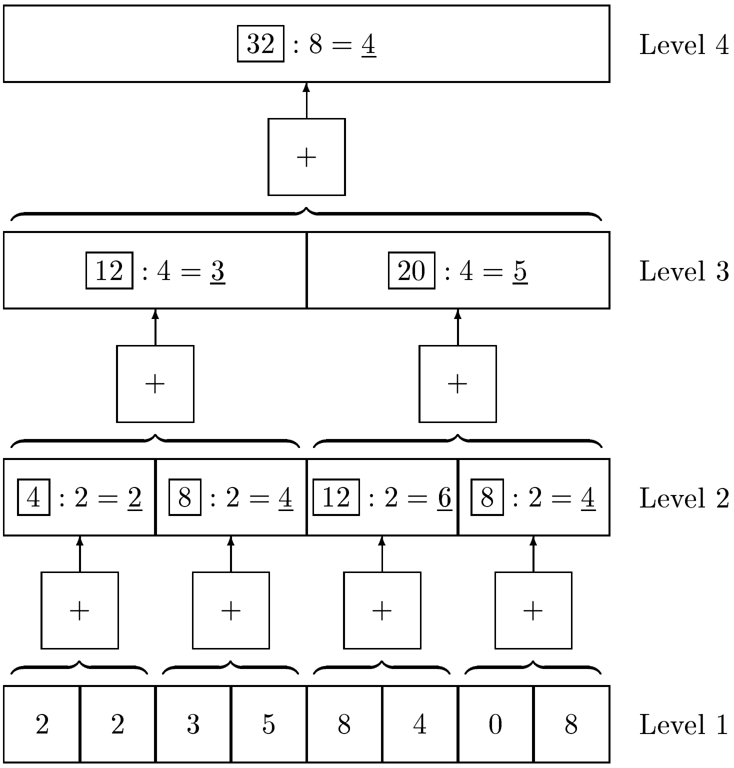

$\boxed{32} : 8 = \underline{4}$
Level 4
+
$\boxed{12} : 4 = \underline{3}$
$\boxed{20} : 4 = \underline{5}$
Level 3
+
+
$\boxed{4} : 2 = \underline{2}$
$\boxed{8} : 2 = \underline{4}$
$\boxed{12} : 2 = \underline{6}$
$\boxed{8} : 2 = \underline{4}$
Level 2
+
+
+
+
2
2
3
5
8
4
0
8
Level 1

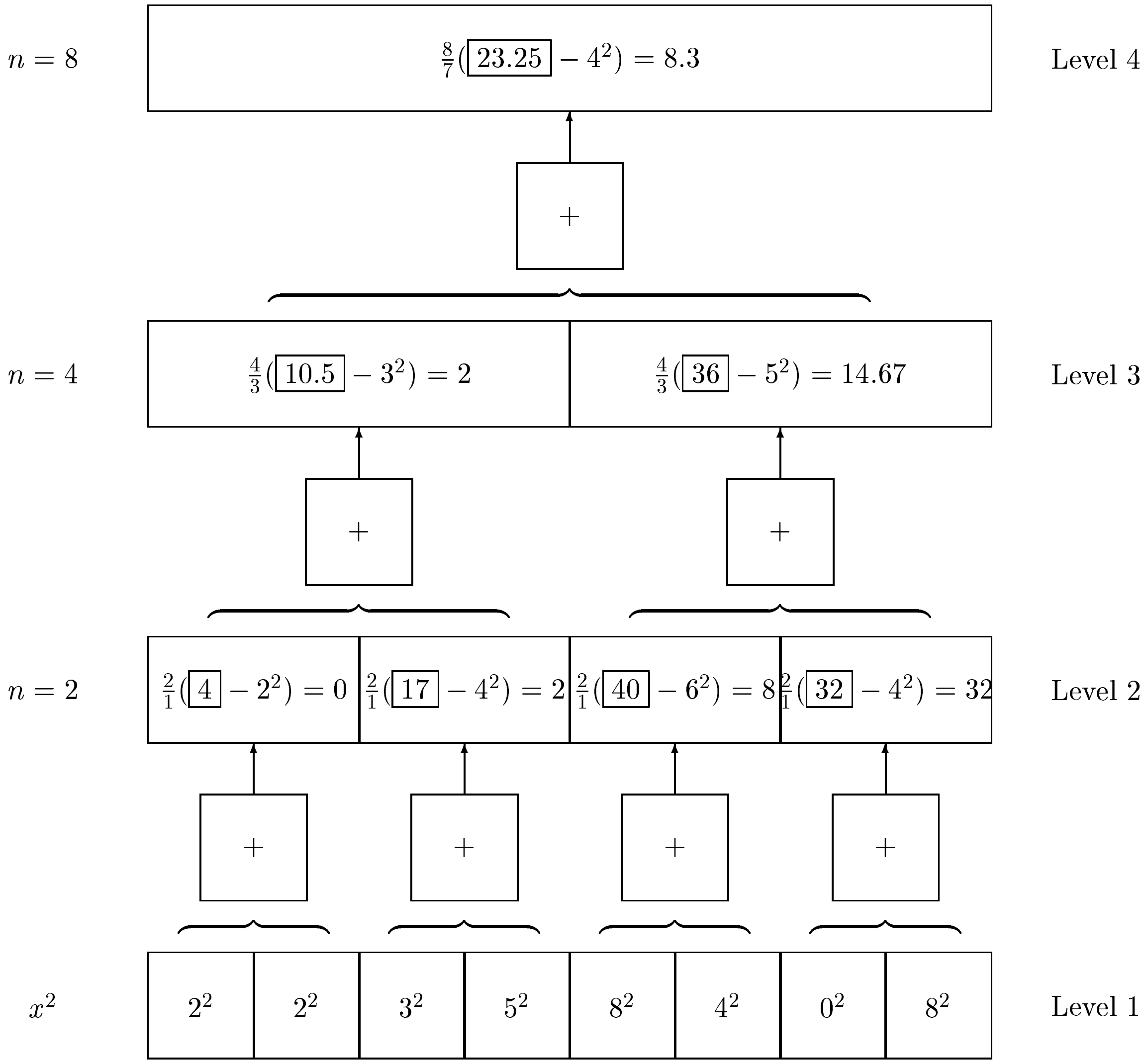
Figure 4: Computing sum $\boxed{\text{s}}$ and mean $\underline{\mu}$ in a pyramid

have to be stored, computes the mean $\underline{\mu}_i^{(j+1)}$ of level $j+1$ using the arithmetic mean of the means $\underline{\mu}_{2i}^{(j)}, \underline{\mu}_{2i+1}^{(j)}$ in the level j below:

$$\underline{\mu}_i^{(j+1)} = \frac{\underline{\mu}_{2i}^{(j)} + \underline{\mu}_{2i+1}^{(j)}}{2}. \tag{3}$$

Using the mean as reduction function, the levels above the base appear as reduced version of the original image.

4.2 Calculating the variance



$n = 8$
$\frac{8}{7}(\boxed{23.25} - 4^2) = 8.3$
Level 4
+
$n = 4$
$\frac{4}{3}(\boxed{10.5} - 3^2) = 2$
$\frac{4}{3}(\boxed{36} - 5^2) = 14.67$
Level 3
+
+
$n = 2$
$\frac{2}{1}(\boxed{4} - 2^2) = 0$
$\frac{2}{1}(\boxed{17} - 4^2) = 2$
$\frac{2}{1}(\boxed{40} - 6^2) = 8$
$\frac{2}{1}(\boxed{32} - 4^2) = 32$
Level 2
+
+
+
+
x^2
2^2
2^2
3^2
5^2
8^2
4^2
0^2
8^2
Level 1

Figure 5: Computing the variance σ^2 (equ. 5) in a pyramid

The **variance** σ^2 of the pixels measures the deviations from the mean. If all pixels have the same value, the variance is zero, the larger the difference to the mean the greater is σ^2.

The variance of n values $x_1, x_2, \ldots x_n$ with mean μ is defined by ([Kre65]):

$$\sigma^2 = \frac{1}{n-1}\left(\sum_{j=1}^{n} x_j^2 - n\mu^2\right) \tag{4}$$

$$= \frac{n}{n-1}\left(\boxed{\frac{1}{n}\sum_{j=1}^{n} x_j^2} - \mu^2\right) \tag{5}$$

The first term $\square$ in equ. 5 can be calculated in the same way as the mean (equ. 3) after squaring the values in the base. The result in every cell combines the precomputed values independently of the others (Fig. 5).

In the same way as sum, mean, and variance are computed in the pyramid, also **statistical moments** and **model fitting** can use the bottom-up summarization scheme that fills the cells in $\mathcal{O}(\log n)$ steps.

Tanimoto [Tan88] accomplishes bottom-up feature extraction with a **hierarchical election** scheme.

4.3 Calculating the bimodality

The last example of this section measures the **bimodality** of circular histograms [JR89]. Circular histograms describe measurements on a cyclic scale such as the slope of edges which varies modulo 360^0. A histogram is called bimodal if it is a mixture of two Gaussian subpopulations. The range of measurements is divided into two intervals, the maximum-distance (MD) partition, so as to maximize the Fisher distance between the resulting two subpopulations P_1 and P_2. The squared Fisher distance between P_1 and P_2 is defined as

$$FD^2 = \frac{n(\hat{\mu}_1 - \hat{\mu}_2)^2}{\hat{n}_1\hat{\sigma}_1^2 + \hat{n}_2\hat{\sigma}_2^2} \tag{6}$$

where $\hat{n}_1, \hat{n}_2, \hat{\mu}_1, \hat{\mu}_2, \hat{\sigma}_1^2, \hat{\sigma}_2^2$ are the sizes, the means, and the variances of P_1, P_2 respectively.

For a linear scale, there are $n-1$ possibilities to subdivide $[1, n]$ into two intervals. Hence $n-1$ Fisher distances are to be computed. For the cyclic scale, the computational cost increases to $\binom{n}{2}$ FD calculations. Therefore a coarse-fine strategy is proposed.

The resolution reduction groups the 2^k original values into 2^{k-h} intervals of the length 2^h. The reduction function $m : [0, 2^k) \mapsto [0, 2^k)$ requantizes the values into the midpoints of the subdivision intervals : $m(g) = \lfloor \frac{g}{2^h} \rfloor 2^h + 2^{h-1}$ (compare requantization by half-toning of Werman [WP88]).

Let f(i) be the distribution of the original values, $f_h(i) = f \circ m$ be the distribution of the reduced resolution. Then the coarse-fine strategy proceeds in two steps:

(a) Find the MD-partition $[r_h, s_h), [s_h, r_h)$, of f_h.

(b) Examine all partitions $[r_h + a, s_h + b)$ where a and b vary independently within $[-2^{h-1}, +2^{h-1})$ and choose the best MD-partition.

If the reduction factor 2^h becomes larger than the expected distance between the peaks of f, the coarse-fine strategy may not discover the correct bimodal partition. The required number of FD-computations reduces to $\binom{n}{2}\frac{1}{k^2} + \mathcal{O}(k^2)$.

5 Different Pyramid Structures

Besides the most frequently used 'classical' $2 \times 2/4$ and $4 \times 4/4$ pyramid structures, other types of structures have also been used recently. This section starts with some new regular structures, then investigates into other possible pyramid structures and compares some of their properties, among which there is their robustness with respect to perturbations in their structure. Before describing one of the newest research objectives in section 6, irregular pyramids, we compare pyramidal performance with some non-pyramidal architectures for image analysis.

5.1 The $2 \times 2/2$ pyramid

The first who worked with this type of pyramid was Crowley at CMU[1] [CP84], [CS84]. He computed a $2 \times 2/2$ Laplacian pyramid (he used the term DOLP[2]-transform) for detecting the peaks and ridges in the 'terrain' of gray values.

In [Kro85b]the major properties of the $2 \times 2/2$ pyramid structure are presented:

- The shape of the receptor field is an octagone, the side lenghts of which are derived in the number of border cells.
- Giving equal weights to all sons, the equivalent weighting in the base is formed by Gaussian-like parabolas.
- If all cell centers of the pyramid are down projected to the base plane, the projected points are part of a grid that is a refinement of the base level of the pyramid.

  The same refinement grid solves the representation problem in [GB88]: the one pixel wide medial axis (skeleton) of an arbitrary shape is uniquely defined on this 'derived' grid even if the corresponding region has a diameter of an even number of pixels.
- Every son has two parents, every parent four sons.
- Using wrap around, the pyramid levels can be conveniently stored in rectangular arrays of sizes $1 \times 1, 2 \times 1, 2 \times 2, 4 \times 2, 4 \times 4, 8 \times 4, 8 \times 8$,..
- The area of a cell is twice the area of a cell in the level below.
- Adjacent reduction windows have one cell in common (overlap).
- The grid axes of the higher levels are rotated by 45^0 with respect to the level below.

5.2 Properties of regular structures

Polygonal decompositions for hierarchical representation have been studied by Ahuja [Ahu83].

11 different hierarchical partitions of the two-dimensional plane are compared in [BDHJ83] by means of properties and relations among the cells:

[1] Carnegie Mellon University in Pittsburgh, Pennsylvania

[2] Difference Of Low-Pass

1. adjacency;
2. rotational symmetry;
3. aperture (corresponds to our term observation window);
4. circularity;
5. convexity;
6. orientation;
7. limit: if the shapes of higher level cells are not similar to cell shapes of the level below;
8. shape similarity;
9. regularity: if atomic cells are composed of regular polygons;
10. isohedrality;
11. democracy: if it is impossible for a cell to differentiate between its sons; compare with Tanimoto's hierarchical election scheme [Tan88].

It is concluded that no tiling satisfies all the criteria. But the square grid and the hexagonal grid display most of the desirable properties.

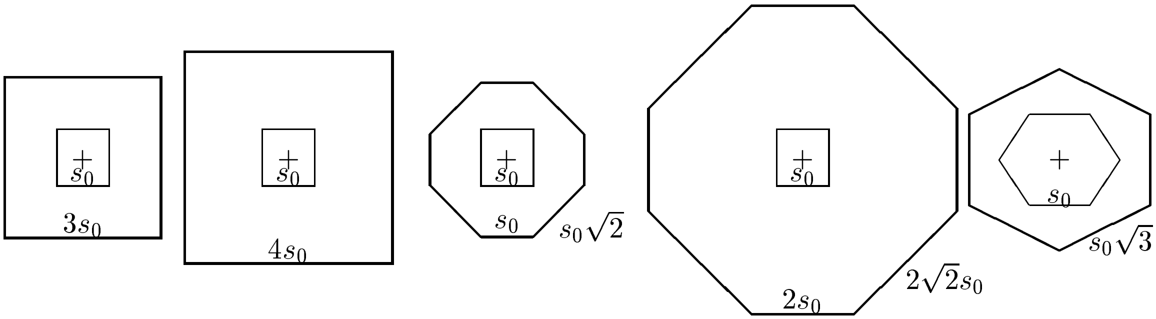
In [Kro88c] pyramids are compared by means of their receptive fields and some other related properties. In pyramids the contents of a large number of high resolution cells is propagated up to a successively smaller number of lower resolution cells. At the first level the contents of the base level's reduction windows is summarized, the second summarizes the local summarizations of the first level and so on. To interpret the contents of a high level cell it is important to know the domain in the base from which information is summarized in that cell. This region in the infinitely fine base plane is called the **receptive field** of the cell.

Non-overlapping structures form tree structures such that the receptive field is just the union of all leave nodes. The refinement of the starting cell is a partition of that region. Receptive fields of overlapping structures tend to grow, the refinement of a cell is larger than the cell. The importance of overlap in pyramids is pointed out in [Fer86]. It is also a key property for curve pyramids as we will see in chapter 11.

Fig. 6 shows the sizes and shapes of the receptive fields of five overlapping pyramid structures. Table 4 summarizes the structural properties of the eight most commonly used pyramids. It enumerates the characteristics of a cell's receptive field, its size in units of the cells area and its shape, the number of sons that two neighboring parents share (overlap), the number of sons of every cell except those in the base, and the number of parents and their frequency weight if there are more than one (e.g. $\frac{f}{n}$ means that every n-th cell has f parents).

Hartmann's hexagonal pyramid structure [Har84c] shows some interesting similarities with the $n \times n/2$ pyramids:

- Cell centers are positioned above each other as in the $3 \times 3/2$ pyramid.
- It has the same refinement rule as the $2 \times 2/2$ pyramid to generate the higher resolution from the lower resolution: insert inbetween every pair of brothers a new cell.



(a) 4 × 4/4 (b) 5 × 5/4 (c) 2 × 2/2 $(d)3 \times 3/2$ (e) 7/4

Figure 6: Receptive fields of 5 pyramid structures

Table 4: Properties of regular pyramid structures

Pyramid	receptive field		structural features		
	size	shape	overlap	sons	parents (weighted)
$(2m)^2/4$ (even)	$(2m-1)^2$	square	$2m(2m-1)$	$4m^2$	m^2
$(2m+1)^2/4$	$4m^2$	square	$4m^2+2m+1$	$(2m+1)^2$	$\frac{(m+1)^2}{4} + \frac{m(m+1)}{2} + \frac{m^2}{4}$
2 × 2/4	1	square	0	4	1
4 × 4/4	9	square	8	16	4
5 × 5/4	16	square	15	25	$\frac{9}{4} + \frac{6}{2} + \frac{4}{4}$
2 × 2/2	7	octagone	1	4	2
3 × 3/2	28	octagone	4	9	$\frac{5}{2} + \frac{4}{2}$
7/4 hexagon	3	hexagone	1	7	$\frac{1}{2} + \frac{2}{2}$

We finish with a list of papers describing applications of the respective structure:

2 × 2/4 **:** [Tan72], [Ram75], [TK80], [Shn81], [Shn82], [SW84], [Sam84], [US84], [SSW85], [Sam85], [Den86], [DMS86], [LLL86], [Par86], [SH86], [Uhr86], [Tan88], [Mal89] .

4 × 4/4 **:** [BHR81], [Bur81], [Hon82], [Har84b], [GHR85], [HR85], [Har85], [Fer86], [GJ86], [Gro86], [KD86], [Wel86], [Ros87b] .

5 × 5/4 **:** [BA83a], [BA83b], [AAB$^+$84], [ABv85], [vS85], [BASv86], [Bur88] .

2 × 2/2 **:** [CP84], [CS84], [Kro85b], [Kro85a], [Kro86e], [Kro86b], [Kro87b] .

3 × 3/2 **:** [Kro86a], [Kro86d], [Kro86c], [Her87], [KP87], [KH88], [Paa87] .

7/4 **hexagonal structure :** [Har84c], [DH86], [Har86b], [Har87a] .

5.3 The robustness of pyramids

The robustness of pyramidal algorithmus to pertubations of the **structure** of the pyramid is studied by Meer et al [MJBR88]. The perturbations are of three different natures:

- Gaussian noise added to the generating kernels‘s weights;
- random local offsets of child-parent links;
- stochastic pyramid structures.

The first two perturbations apply to regular ($2 \times 2/4$) pyramid structures. Burt [Bur81] showed that generating kernels should fulfil the following spatial constraints:

- normalisation (the sum of the weights is one)
- symmetry
- unimodality
- equal contribution to the next level.

Since adding noise to the kernels destroys Burt's constraints, several methods are given to reinstitute certain constraints on the noisy kernels. From those reinstitutions, normalisation appears to be the most important constraint and thus may help to improve the quality of low resolution pyramid levels.

Perturbations of child-parent links include offsets introduced at the children's level and offsets introduced at the parent's level. When the positions of the links were perturbed at the parent's level, normalisation was not preserved and the effects were severe. Reinstitution of normalisation improved the results in that case. The effects of perturbing the children's level were milder, because normalisation was preserved.

From the extensive experiments a general principle for constructing robust pyramid representations is concluded: the parent should control the computation so that he may normalize his weights.

Stochastic pyramid structures (see chapter 6 for more detail) are built by a stochastic recursive decimation process on the pixels of the base. Even on such severly perturbed pyramid structures two selected algorithms only show a mild degradation. In the bimodality detection algorithm, the pyramid fuses locally computed statistics. There is a reason why a stochastic implementation should lead to greatly different results than a regular one. In the second algorithm, object delineations, the low resolution representations show great discrepances to the original object but greatly the top-down delineation process successfully recovers from incorrect root definitions.

Multiresolution algorithms appear to be very robust and only weakly dependent on the structure on which they are implemented.

5.4 Pyramidal and non-pyramidal architectures

Structures with which the pyramid is often compared are the cellular array, the hypercube or n-cube, and the prism. They differ in the connections between the cells in the structure. The connections in a **cellular array** connect a cell with its three (triangular array), four (square grid with 4-connectivity), six (hexagonal array), or eight neighbors (square grid with 8-connectivity). An **n-cube** (also called hypercube) has 2^n cells and every cell is connected to n other cells. A **prism** [Ros85b] is a stack of cellular arrays, all of the same size. It has the same number of levels as the $(n \times n/4)$ pyramid resulting in a total of $n \cdot 4^n$ cells. Besides the usual neighbor connections every cell is connected to three cells in the level above.

Fountain [Fou88] evaluates the differences between the pyramid, the n-cube, and the (linear) array: The pyramid and the n-cube are rather evenly matched in cost, total processor performance and low-level processing times. In higher-level symbolic operations, the additional layers of the pyramid structure offer a significant improvement over the n-cube. The linear array is obviously in a lower category than the other two in cost and some aspects of performance.

Image pyramids are built in log(image_size) time with the consecutive levels having their size and resolution reduced by a constant factor. Similar structures with the representations decreasing only in resolution but not in size are also of interest. Meer [Mee88] simulates such constant size multiresolution representations of the input on image pyramids by increasing the number of values stored in the cells of the host structure. Constant size representations allow parallel processing in applications such as scale-space filtering and multiresolution edge detection.

Some important statements in [Mee88] are:

- At higher levels of image pyramids, the amount of noise is significantly decreased. Objects may become easier to detect. They allow fast data gathering across the input image.
- The resolution reduction in pyramids can be regarded as the discrete case of scale-space smoothing.
- Multiresolution edge detection is another class of applications requiring the same size for all the representations.

In pyramids a parent at level l carries a scalar value, the weighted average of its children. Instead of allocating a scalar value to each cell at level l, Meer allows it to carry a $Q_{min}(l) = s^l$ dimensional vector. Each (vector) component can be computed recursively by a pyramid PYR[q-s+1,s], where q-s+1 is the size of the reduction window, and s is the reduction factor. The PYR[q-s+1,s] pyramids are hosted by the initial PYR[q,s] structure.

A (one-dimensional) prism, PSM[q,s], is the concatenation of s^n interleaved (one-dimensional) PYR[q,s] pyramids. In his paper ([Mee88]), Meer simulates the essential levels of prism machines, the PSM[q-s+1,s] prism, on a PYR[q,s] pyramid. Another constant size multiresolution representation in [Mee88] is the simulation of a window with increasing length sliding on the input.

Peleg and Federbush [PF86] describe pyramids with variable reduction factors ('contraction'). Such structures may enhance the flexibility in using appropriate cell sizes for representation and recognition.

6 Irregular Pyramids

In irregular pyramids we relax the regularity constraints of regular structures, but keep the general contraction property of pyramids. Such structures are of interest because (1) also biological vision systems are not completely regular and (2) little perturbations may destroy the regularity of regular pyramids (as discussed above). There are several issues for generating irregular pyramids.

- Parallel graph contraction represents every level of the hierarchy by a (neighborhood-) graph. The nodes of the graph are the pyramidal cells. Two nodes are connected in the graph if the corresponding cells are neighbors. Higher level graphs are created by successively merging a certain number of nodes in the lower level graph.

- Decimation is another approach for building an irregular hierarchy. In this concept, the cells of a level are divided into two categories: those that **survive** the decimation process and form the cells of the next higher level, and those that are **eliminated**.

- Recently artificial neural networks have interested many researchers. In Pao's book [Pao89] their special abilities for adaptive pattern recognition are described in detail. The hierarchical neural network NEOCOGNITRON, as proposed by Fukushima [FMI83], shows great similarity to pyramidal structures. We therefore also mention an approach to fuse the irregular neural network and the regular pyramid.

- Fractals are sets which are extremely irregular at all scales. There are recently several applications in image analysis that relate fractals with image compression, image transmission, and texture segmentation.

6.1 Parallel graph contraction

Rosenfeld [Ros85a] explores different sorts of cellular hierarchies in a general theory. His focus is on methods that allow a parallel generation of the structure. By describing the interconnection network in terms of a graph, the problem is to find a parallel contraction scheme such that **the degree of the contracted graph remains bounded**. This important property would, for example, allow to simulate large networks of processors by smaller ones. For some special, regular cases parallel contraction is possible: hypercube, k-dimensional array, hexagonal array. However the degree of triangular array increases. The $2 \times 2/2$ pyramid appears as a special case of square grid hierarchies when (diagonal) pairs of cells at distance two are merged.

Melter [Mel86] also investigated into algebraic structures to characterize regular and semiregular partitions of the plane using 'rosettas'.

Abrahamson et al [ADKP87] present an efficient algorithm for contracting a tree structure in parallel. Let

$$T = \{V, f, s | f : V \mapsto V, \forall v \in V - \{r\} \overset{1}{\exists}\, w \in V : f(v) = w; s : V \mapsto (V \times V) \cup V \cup \emptyset\}$$

be a binary rooted tree with vertex set $V(T)$, with root $r \in V$, with father function $f(v)$, and with son function $s(v)$. A sequence of trees $T_1, T_2, \ldots T_k$ is called a **tree contraction sequence** of length k for T if

(i) $T_1 = T$;

(ii) $\forall i = 2, 3, \ldots k : V(T_i) \subseteq V(T_{i-1})$;

(iii) $|V(T_k)| \leq 3$; and

(iv) $\forall i = 2, 3, \ldots k$ if $v \in V(T_{i-1}) - V(T_i)$ then either

LEAF: v is a leaf of T_{i-1}; or

BYPASS: $s_{i-1}(v) = x \in V(T_{i-1})$ and $f_{i-1}(v) = f_i(x)$.

Two basic operations generate the tree contraction sequence:

PRUNE(v): Eliminate leaf v (satisfies condition LEAF); and

BYPASS($f(v)$): Eliminate father $f(v)$ of leaf v if condition BYPASS is satisfied.

Let T have n leaves. The presented algorithm constructs a contraction sequence in $\mathcal{O}(\log n)$ time using $\mathcal{O}(n/\log n)$ processors by numbering first all leaves from left to right. Then a leaf is removed in phase t if the rightmost 1 in its leaf index is in position t. In this way the difficulties associated with parallel path compression are avoided.

A tree contraction algorithm gives a method for solving several optimization problems, when the underlying graph is a tree, for example minimum covering set, maximum independent set and maximum matching.

6.2 Decimation

In Meer's stochastic pyramid [Mee89], the decimation is based on a randomized parallel algorithm. The surviving cells are supposed to obey two rules:

1. Surviving cells should not be too close.
2. Surviving cells that become neighbors should not be too far apart in the level below.

A fast converging parallel algorithm generates in two phases a subset of surviving cells that satisfy the above rules both in the 1D and the 2D case. The stochastic decision process starts by assigning every cell a uniformly distributed random number. Selecting only **local maxima** as surviving cells satisfies rule 1, but may leave gaps between these cells that are larger than desired. They are closed in the second phase by applying the same process to those connected subsets of the remaining non-surviving cells that are too far apart from any surviving cell.

The concept of **adaptive pyramid** [JM92] differs by the selection criterium. There the decimation process first tries to select significant cells depending on the cell's content and uses a random selection only where the data do not allow a decision.

The decimation rules can be formulated by means of a neighborhood (set) function $\Gamma(x)$ which defines all cells y that are neighbors of x. Let $\mathcal{L}_n$ denote all cells at level n in the pyramid, $x_n \in \mathcal{L}_n$ a cell at this level and let x_{n+1} at level $n+1$ correspond to x_n at level n. In this notation, the two decimation rules read as follows:

1. $y_{n+1} \in \Gamma(x_{n+1}) \Longrightarrow y_n \notin \Gamma(x_n)$
2. $\forall x_n \in \mathcal{L}_n \exists y_{n+1} \in \mathcal{L}_{n+1} : x_n \in \Gamma(y_n)$

Before this decimation process can be repeated at the decimated level $\mathcal{L}_{n+1}$, the neighborhood Γ at the decimated level $n+1$ must be defined. Every surviving cell x_{n+1} is assigned a *receptive field* $\mathcal{R}(x_{n+1}) \subset \Gamma(x_n)$ such that

$$\forall y_n \in \mathcal{L}_n \overset{1}{\exists}\, x_{n+1} \in \mathcal{L}_{n+1} : y_n \in \mathcal{R}(x_{n+1}).$$

We have investigated in the effects that these two rules have on the resulting structures by analyzing the minimum and maximum distances between neighbor cells at higher levels of the pyramid and by computing bounds for decimation ratios in 1D and 2D.

Since all decimated levels $\mathcal{L}_i$ are subsets of base level $\mathcal{L}_0$, we can measure the distance between every two cells x_i, y_i in any level i by their Euclidean distance in the base: $d(x_i, y_i) = d(x_0, y_0)$.

The factor by which the number of cells decreases is called the **decimation ratio**. It corresponds to the reduction factor of regular structures, but may vary from level to level.

The conclusions of this paper are:

- The distance between two neighbors P_i, Q_i at level $\mathcal{L}_i$ of the 1D irregular pyramid is bounded by following inequality:

$$2^i \leq d(P_i, Q_i) \leq 3^i \tag{7}$$

- The distance between two neighbors P_i, Q_i at level i of the 2D irregular pyramid is bounded by following bounds:

$$\begin{aligned} 2 \leq &\quad d(P_1, Q_1) &\leq 3\sqrt{2} \qquad (8)\\ \sqrt{5} \leq &\quad d(P_i, Q_i) &\leq 3^i\sqrt{2}, i > 1 \qquad (9) \end{aligned}$$

  and these bounds can be reached.

- A decimation ratio of 2 at levels greater than 0 is possible in the 2D irregular pyramid even if 8-connectivity is used in the base level. The smallest decimation ratio on a square lattice is two.

  The $2 \times 2/2$ pyramid is an example.

- A decimation ratio less than 2 and greater than $\frac{4}{3}$ can be realized if the non-surviving cells form a (regular) triangular network such that every surviving cell is located on one triangle side. For large networks this lower bound is closely approached.

These bounds for distances and decimation ratios don't take into account the uniform distribution of the random numbers used for decimation in stochastic pyramids. For stochastic pyramids Meer [Mee89] has found an average decimation ratio greater than 4 in his experiments.

The problem that distances in 2D irregular pyramids need not increase could probably be solved by a better selection of the receptive fields. However the modification of the receptive field selection cannot solve the problem of preserving the degree.

Another hierarchical decimation method is presented in [DP88]. There, the levels form triangular networks, and the decimation is based data-dependent criteria like in the adaptive pyramid. The neighborhood relations of the decimated levels are determined by a new version of the constrained Delaunay triangulation.

The efficiency of irregular pyramids has been shown very recently: the labeling of connected components of an image needs only $\mathcal{O}(\log n)$ parallel steps [MMR89]. This efficiency is made possible by decimating the components separately.

Another parallel algorithm for connected component labeling [MR89a] need a total time complexity of $\mathcal{O}((\log n)^3)$. It is implemented on NASA's MPP processor, which has the structure of an array. Samet's solution [ST88] involves **bintrees** [ST85a], [ST85b], which are generalizations of quadtrees and octrees.

6.3 Neural network

In [YK89b] *Neocognitron* [FMI83] is implemented in a pyramid scheme. The conventional cycle for calculating the lower resolution in a pyramid (e.g. low pass filtering and subsampling) is augmented by two functions from neural networks: pattern matching and sigmoidal contrast stretch. Four different reduction structures are compared for their performance and their computational complexity.

The processing structure has a considerable influence on the performance of the system [YK89a]. The size of both the patterns and the reduction windows have been found important for the learning and recognition performance of the pyramidal Neocognitron. Experiments with the nine digits 0, 1, 2, 3, 4, 5, 6, 7, 8 and various types and degrees of distortions are reported. Further results are to be expected from Yamaguchi's dissertation which should be finished by July 1990.

6.4 Fractals

Fractals have been introduced by Mandelbrot [Man83] as a new method to capture a variety of geometrical properties of nature.

Barnsley used fractals in his IFS system to describe shapes and images achieving extreme large compression rates (1:10000) [BS88]. An input image is broken up into segments through image-processing techniques. These image components are looked up in the IFS library using the Collage Theorem, and their IFS codes are recorded. A set of contractive affine transformations is chosen, so that every image component is approximated as well as possible by the union of the subimages.

When the image is to be reconstructed, the IFS codes are input to the random iteration algorithm. The accuracy of the reconstructed image depends only on the tolerance setting used during the collage mapping stage.

Earlier, Ahuja [AAS85] described a similar application in image coding and secure transmission:

1. Drop points randomly on image plane (Poisson distr.).
2. A Voronoi tesselation defines the cells around nuclei.
3. Assign gray/color value on region majority/size basis to nuclei.
4. Transmit parameters of random number generator and N colors.

Incorrect transmission occures mainly along object boundaries.

Müssigmann [Mue89b], [Mue89a] used the fractal dimension in the context of scale-space filtering (see section 10.4). He calculates the surface F of a gray level image in a certain neighborhood in dependency of the Gaussian smoothing parameter σ. The following power law could be observed for several textures:

$$F(\sigma) \sim \sigma^p \tag{10}$$

where p is the characteristic scaling exponent of the respective textures. The resulting 'fractal dimension images' show good discrimination between different Brodatz textures.

7 Gray Level Pyramids

In this section we concentrate on pyramids where the contents of the cells are gray values or color vectors. In other words, every level of the pyramid is a picture. To fill the cells of the pyramid bottom-up, different reduction functions can be used.

7.1 Reduction by linear filtering

Linear filters are used in digital image processing to remove certain unwanted effects (e.g. noise) from images while enhancing others. One way to filter an image is to convolve it with a spatial filter kernel that has the characteristics of the filter. In the continuous image space, both the image $I : \mathcal{R}^2 \mapsto \mathcal{R}$ and the filter $F : \mathcal{R}^2 \mapsto \mathcal{R}$ are continuous functions, the convolution $I * F$ is defined as

$$(I * F)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(\xi, \eta) F(x - \xi, y - \eta) d\xi d\eta \tag{11}$$

Filters are categorized by their characteristic in the frequency domain (Fourier space). Filters that suppress high frequencies like bright spots or lines and let pass through the low frequencies of the image spectrum are called **low-pass filters**. **High-pass filters** have the opposite characteristic. Filters that suppress frequencies outside a certain range of frequencies are called **band-pass filters**.

The discrete sampling theorem (i.e. see [RK82, Vol. 1, chapter 4.1]) relates the sampling (*Nyquist*) distance (e.g. the distance between the centers of adjacent cells) with the frequencies that can be represented and reconstructed from the discrete image. Moik's physical interpretation [Moi80, section 2.5.1] is useful in context with multiple resolutions: 'The sampling intervals must be equal or smaller than one-half the period of the finest detail within the image.'

Since a low-pass filter suppresses the high frequencies in an image, the result can be sampled with a larger sampling distance without loosing information by sampling. This forms the basis of gray level pyramids that are constructed by low-pass or smoothing filters (see [Bur81] for an early and[OS87] for a recent treatment of multi-resolution low-pass transforms).

A **Gaussian filter kernel** has some unique properties with respect to multiresolution representation [YP86] which will be analysed in more detail in section 10.4. The weights $G(t, \sigma)$ are highest at the center $(t = 0)$ and decrease depending on the parameter σ with increasing distance $|t|$ to the center:

$$G(t, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t}{\sigma}\right)^2} \tag{12}$$

It has been studied in [KH88] for building gray level pyramids and will be summarized in the following. Properties of the Gaussian smoothing filter $G(\sigma)$ (in this simplified notation t has been suppressed) for building gray level pyramids are investigated:

1. Convolving an image I: $G(\sigma) * (G(\sigma) * I) = G(\sigma\sqrt{2}) * I$ [CS84]. $G(\sigma\sqrt{2})$ is called the **equivalent weighting function** [BA83a].

2. The mean $\mu(G(\sigma) * I) \approx \mu(I)$ remains the same except for deviations caused by the boundary. This means that **the gray value** of large homogeneous regions **does not change** as long as they are large compared to the size of the reduction window.

3. The standard deviation is reduced in the filtered image.
   Since normal distributed noise is measured by the standard deviation, the amount of **noise is reduced** in the pyramid if $G(\sigma)$ is used as the reduction function.

4. Since repetitive application of $G(\sigma) * I$ is equivalent to convolution with a larger σ, the different levels of a pyramid can be built in two ways: either iteratively level by level from the bottom to the top or directly from the base using the equivalent weighting function. It is shown in [KH88] that the iterative computation needs much less time than the direct computation.

In a discrete image I the convolution is computed as the weighted sum of the gray values with the filter kernel $F_{(t_1+t_2+1)\times(t_3+t_4+1)}$ in the window $[-t_1, t_2] \times [-t_3, t_4]$:

$$(I * F)_{k,l} = \sum_{i=t_1}^{t_2} \sum_{j=t_3}^{t_4} I_{i,j} F_{k-i,l-j} \tag{13}$$

In a discrete filter, the above discrete convolution is calculated and stored at every pixel position by shifting the window pixel by pixel and row by row over the entier image.

When used as a reduction function in a pyramid, the convolution result need to be computed only at every n-th position according to the reduction factor. The result would be exactly the same if the filtered image would be decimated, e.g. by taking every n-th pixel. Wells [Wel86] describes an implementation of Gaussian filters for pyramids by making efficient use of separability of the Gaussian. A 2-dimensional filter with kernel $F_{n\times m}$ is called **separable** if it can be split in two 1-dimensional filters

$$F_{n\times m} = F_{n\times 1} \cdot F_{1\times m} \, .$$

Then the result can be obtained by convolving first the rows of I with $F_{1\times m}$ and, then, the columns of the first result with $F_{n\times 1}$.

The above considerations are valid not only for the classical pyramid structure but also for many other multiresolution structures (see section 5 for other pyramidal structures). Crowley [CS84] used a $2 \times 2/2$ structure.

7.2 Edge-preserving reduction

A Gaussian reduction function has nice noise cleaning properties, but it smooths at the same time sharp contours. Edge-preserving filters are designed to adapt the degree of smoothing to the presence of an edge ('Edge Preserving Smoothing' [NM79], [Har87b]). Since these functions give good results when used as a filter, we used them as reduction function. We compared three different reduction functions for generating a $3 \times 3/2$ gray level pyramid on SAR[3] images in [Her87] and [KP87]:

- a Gaussian filter,
- the Frost filter [FSSH82] , and
- the Lee filter [Lee81].

The Frost filter is an adaptive filter for SAR images to remove multiplicative noise: It is computed as a local convolution of the image with varying weights W that are recalculated depending on the local statistics μ (mean) and σ^2 (variance) and the local Euclidean distance $d_{i,j}$ from the application center:

$$V_{i,j} \;= e^{\left(-\frac{\sigma^2}{\mu^2} \cdot d_{i,j} \;\cdot f\right)} \tag{14}$$

The $V_{i,j}$ are locally normalized to yield the weights $W_{i,j}$.

The filter proposed by Lee [Lee81] is also an adaptive filter which reduces additive noise in radar images.

One of the five major visual observations in [KP87] seems to be characteristic for adaptive filters:

[3] synthetic apertur radar; see [Gol86] for a comprehensive treatment of SAR image formation

Such filters are normally tuned to produce sharp contrasts. Thin lines are often reduced to one pixel width. If used as a reduction function the filtering is followed by a subsampling process which may cut originally continuous lines into pieces. In some cases (diagonal) lines disappear completely in $3 \times 3/2$ pyramids.

Theoretical results in [Her87] include that, after a reduction step with any of the investigated reduction functions, the mean within a large homogeneous region remains approximately the same, and that the noise measured by the standard deviation is reduced. Repetitive application of Gaussian filters (with parameter σ_i) is equivalent with one application of a Gaussian with appropriate parameter $\sigma = \sqrt{\sum \sigma_i^2}$. This makes it possible to generate an arbitrary higher level of the pyramid directly from the base. However, the time complexity is much less for the iterative level-by-level computation in the pyramid than for direct computation. The effects on SAR images have been of special interest although the conclusions are general.

Further edge-preserving pyramid constructions make use of contour information and will be reported in chapter 12.

7.3 The maximum pyramid

Besides linear filters also other (non-linear) reduction functions have been used.

Blanford and Tanimoto [BT86] made extensive experiments on their pyramid machine. They describe the detection of bright spots in a $2 \times 2/4$ pyramid structure with two different bottom-up reduction functions: *maximum* and *average*. The bright spot is found in a top-down search pass that links the apex of the pyramid to one pixel in the base.

The non-overlapping structure of their pyramid produces results that depend on the lateral position of the object region. But the width of a region, which is defined as the side length of the largest pyramidal cell completely contained in the region, will vary by at most a factor of 2 when subjected to lateral translation within the pyramid. The average, when applied to a region of width $2^j, j > 0$, at level k produces a region of width 2^{j-1} at level $k-1$. The maximum as reduction function produces a region of width at least 2^{j-1} at level $k-1$. Averaging the first j levels and using the maximum above can detect bright spots of width 2^j or greater, $j > 0$, in lieu of regions of less width.

[BT88] is a revised version of [BT86]. This paper additionally investigates 'convex' combinations of average and maximum reduction functions and observes the effects on small, bright regions in contrast to large ones. A possible extension of the pure $2 \times 2/4$ structure is the $4 \times 4/4$ overlapping pyramid which could be used within the same framework. It improves the results while only increasing the computational costs by a constant factor.

7.4 Morphological reduction

The first two operations of mathematical morphology [Ser82], [JH89] are the operations of **dilation** and **erosion**. Let D denote the domain of all cell centers. The dilation of a set of cells $A \subseteq D$ with a set $B \subseteq D$ is defined by

$$A \oplus B = \{x | \text{for some } a \in A \text{ and } b \in B \text{ , } x = a + b\} \tag{15}$$

The erosion of a set of cells A by B is defined by

$$A \ominus B = \{x | \text{for every } b \in B \ , \ x + b \in A\} \tag{16}$$

Using $B = \{(0,0), (-1,0), (1,0), (0,-1), (0,1)\}$ in the 4-neighborhood or

$$B = \{(0,0), (-1,0), (1,0), (0,-1), (0,1), (1,1), (-1,1), (1,-1), (-1,-1)\}$$

in the 8-neighborhood, the 1's in a binary digital image I can be shrunk: $I^{-1} = I \ominus B$, they can be expanded by $I^1 = I \oplus B$. Multiple **shrink** and **expand** operations are defined recursively: $I^{-k-1} = I^{-k} \ominus B$ and $I^{k+1} = I^k \oplus B$.

Rosenfeld [Ros87a] investigated in calculating the shrinking and expansion operations using a cellular pyramid. His conclusions are:

1. A cellular pyramid can compute $I^{\pm(2^k-1)}$, resampled at intervals of 2^k, in $\mathcal{O}(k)$.
2. In one dimension, a cellular triangle can compute $I^{\pm(2^k-1)}$, without resampling, in k steps each involving $\mathcal{O}(k)$ computation.
3. In two dimensions, a cellular pyramid would need $\mathcal{O}(2^k)$ computation to compute $I^{\pm(2^k-1)}$ without resampling! Hence in this case, the pyramid has no speed advantage over a conventional cellular array.
4. For a grayscale image, expansion is generalized to local MAX (compare Blanford's and Tanimoto's maximum pyramid) and shrink to local MIN. In this case there is no speed advantage of the pyramid even in one dimension.

The purpose of the following approach is that it does not attempt to calculate a high resolution morphological operation, but rather uses morphological filtering instead of linear filtering to reduced the image's resolution.

The morphological filters **opening** ($A \circ B$) and **closing** ($A \bullet B$) are combinations of dilation and erosion:

$$\begin{aligned} A \circ B &= (A \ominus B) \oplus B & (17) \\ A \bullet B &= (A \oplus B) \ominus B & (18) \end{aligned}$$

Based on these definitions, Haralick and Zhuang [HLLZ87], [HZLL88] introduce a morphological reduction function. The reduction is performed in two steps: first, the input image is morphologically filtered, second, it is morphologically resampled to the reduced image size. A binary and a grayscale sampling theorem are formulated. It defines conditions on the filtering kernels that must be satisfied to preserve the relevant information after sampling. It further specifies to what precision an appropriately morphologically filtered image can be reconstructed after sampling.

7.5 Pyramid linking and segmentation

The concept presented in this section provides an efficient method for segmenting an image through the use of a gray level pyramid. The pyramid structure being used must be **overlapping**, e.g., the size of the reduction window must be larger than the reduction factor. The classical $4 \times 4/4$ is an example.

First, a gray level pyramid is built bottom-up as described in the previous sections. Although mostly Gaussian pyramids are used there is no principle restriction on what type of reduction function is to be used.

In a second (**linking**) step, every cell below the apex of the pyramid chooses one of its 'fathers' to which it is most similar in gray value. This choice is possible because in an overlapping pyramid, cells are covered by more than one cell of the next higher level.

In the third step, all the cells above the base level **recompute** their gray value based on the values of the sons that link to them.

This process is iterated until a stable state is reached. The final links constitute tree structures that have the property that all the cells that link to a common father are similar in gray value. There are different methods to cut the tree structure to get a segmented image. Typical features extracted by pyramid linking are compact homogeneous regions [HR84].

Grosky and Jain [GJ86] give an overview on **pyramid linking** methods and segmentation through pyramid structures. They focus on the 4 to 1 reduction of type $n \times n/4$ pyramids. Their segmentation method has following advantages (+) and disadvantages (−):

+: Unconnected segments (regions) will not be linked together.

−: Many long and thin regions cannot be segmented in parallel.

There are special cases in pyramid linking: a son may be different to all of its fathers (e.g., when it should become a root in the segmentation tree), a father may be disconnected from all its sons. Following situations may occur when a 2×2 reduction window is used (like in the $2 \times 2/4$ or $2 \times 2/2$ pyramids) and labels (or events) are passed up to the father (in the following nb-events(sons) is the number of different labels of the four observed (son-)cells):

1. **if** nb-events(sons)=0 **then** father:=no_event;
2. **if** nb-events(sons)=1 **then** event(father):=event(son);
3. **if** nb-events(sons)>0 (e.g. $2, 3, 4$) **then**
   - event(father):= select-one-event(sons); **or**
   - event(father):= merge-event(sons);
     Merge-event can be done:
     - numerically: sum, mean; or
     - symbolically: existence, new labels by substitution rules: e.g., event D is instantiated if events A, B, and C are present in the receptive field.

Wharton [Wha88] builds a $2 \times 2/4$ multispectral pyramid by cascaded averaging the spectral component vectors. The segmentation aims at spatially connected regions with homogeneous average spectral response. It is essentially the classical iterative pyramid linking algorithm with extensions for the spectral components. It differs in the choice of potential fathers, which depends on the current set of neighbors and their father's links. Finally a command interpreter (CI) allows the interactive modification of the result of automatic processing. Typical operations are merge regions into one region, split a region into two regions, and examine the contents of a specific region.

In [GHR85], pyramid linking is applied to segment so-called Glass pattern. They are constructed by displacing copies of random dot patterns and are often used to compare the ability of computer vision methods with human performance which is very good in perceiving structure in complex patterns. The presented approach is able to link the corresponding dots in $\mathcal{O}(\log n)$ computation steps.

Gross describes in [Gro86] (his PhD-thesis) and in [Gro87] (summary) several variations of the pyramid linking scheme. To arrive at a segmented image two stages are distinguished: **detection** and **delineation**.

In Gross' scheme, a cell of the pyramid contains three values: the gray value, the variance σ^2, and an interest measure. The later determines the presence of a blob (= compact region) in a cell of the pyramid and initiates the top-down delineation process. Two variants of the interest measure have been tested: $\frac{\sigma_F^2}{\sigma_S^2}$ and $\frac{\sigma_S^2}{\sigma_F^2}$, where σ_F^2 is the variance of the father cells and σ_S^2 the variance of the son-cell. The instance of a blob is detected if the interest measure of the cell is greater than the interest measure of all its fathers.

Table 5: Delineation methods

Name	Delineation method	Time	Quality
Local method	assign labels OBJECT and BACKGROUND top-down by closest-father rule.	$\mathcal{O}(\log n)$	not satisfactory
Global method	same as local method except that sons take the father's value they link to.	$\mathcal{O}(\log n)$	good
Updated method	after each level k has been labeled, the father's values/level is recomputed up to the seed level and then again down to level k.	$\mathcal{O}(\log^2 n)$	very good
Iterated global method	proceeds in two passes: 1. Top-down labeling and passing best father's value to the son; 2. bottom-up recomputing.	$\mathcal{O}(\log n)$	very good

In the delineation stage, the outline of the detected blob is determined. Earlier versions of the boundary localization are [BR85], [BR86]. Four top-down methods are tested in Gross' work: the local method, the global method, the updated method, and the iterated global method. Table 5 summarizes the methods, their time complexity, and their segmentation quality.

7.6 Hierarchical Hough-transforms

The use of hierarchies for computing the Hough-transform to identify straight lines, circular arcs, and, in general, arbitrary shapes in digital images did interest several researchers. Since the hierarchical approaches to compute the Hough-transform show similarities to pyramids we present some of them in this subsection.

Davis [Dav79] introduced the hierarchical generalized Hough-transform:

The classical Hough transform shape matching algorithms constitute a class of procedures for extracting analytically defined shapes from planar point sets.

The generalized Hough transform is a fast point pattern matching procedure that can be used to detect arbitrary specific shapes (rather than analytically defined classes of shapes). The generalized Hough transform solves the following point pattern matching problem: Given a set of object points $O \subseteq \mathcal{R}^n$, a set of feature points, $P \subseteq \mathcal{R}^m$ and a set of functions, F, with $f \in F$ being a mapping $f : \mathcal{R}^n \rightarrow \mathcal{R}^m$, find the $f \in F$ such that $v(f) =\mid P - f(O) \mid$ is minimal.

The generalized Hough transform can be extended to operate on the basis of hierarchical shape representations. The principal advantage of using hierarchical representation is increased control over the shape recognition process.

Davis' paper [Dav79] makes three points:

1. That Hough transform shape matching algorithms are instances of a general point pattern matching algorithm;
2. That the point pattern matching algorithms can be usefully extended to match hierarchical point patterns, and
3. That the generalized Hough transform can be further generalized to match patterns of geometric objects other than points , e.g., line segments.

O'Rourke [O'R81] used the K-d-tree for Hough space representation.

Li [LLL86] computed Hough-transforms for hyperplanes using K-tree, quadtree, or octree.

Tanimoto [Tan88] computes the pyramidal Hough-transform. Compared with the traditional Hough transform, the pyramidal version fails only in finding lines having global but not local support. But it is more than an order of magnitude faster than the classical Hough transform. Similarities between the hierarchical election scheme and neural networks are pointed out.

Sher and Rosenfeld [SR89] recently describe an implementation of a pyramid Hough transform on the connection machine.

7.7 Bibliographical notes

[Den86] and [DMS86] (in German) describe the use of the **dynamic pyramid** to estimate the local motion in a series of images.

Werman and Peleg show an additional possibility to compress an image in [WP86] and the revised version [WP88]: by reducing the gray level resolution (i.e. the number of gray levels) of the image. For this purpose a new image metric is used: **unfolding**. Gray levels are quantized in parallel with the reduction of resolution.

8 Laplacian Pyramids

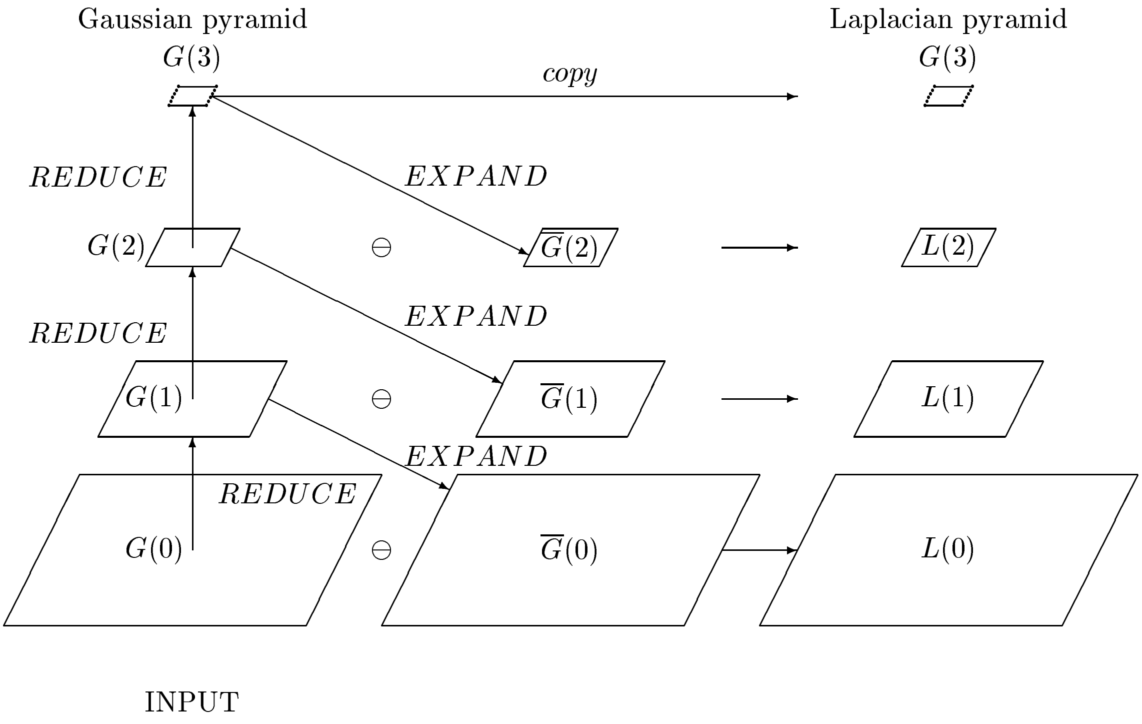


Figure 7: Building the RE Laplacian pyramid $L(0), L(1), L(2), G(3)$.

Burt [BA83a] describes a method for compressing, storing and transmitting images in a computationally efficient way.

Let $G(k)$ denote a $5 \times 5/4$ Gaussian pyramid, where k denotes the different levels and $G(0)$ is the base. The bottom-up building process is based on the reduction function $REDUCE$:

$$G(k) := REDUCE(G(k-1)) \, , \; k := 1, 2, \ldots \tag{19}$$

which has two tasks:

- Filtering with a 5×5 kernel which depends on one parameter -
  Here the important notion 'equivalent weighting function' is introduced.
- Sampling

It is stated that the whole process is 'faster than the Fourier transformation' (see also O'Gorman's evaluations below).

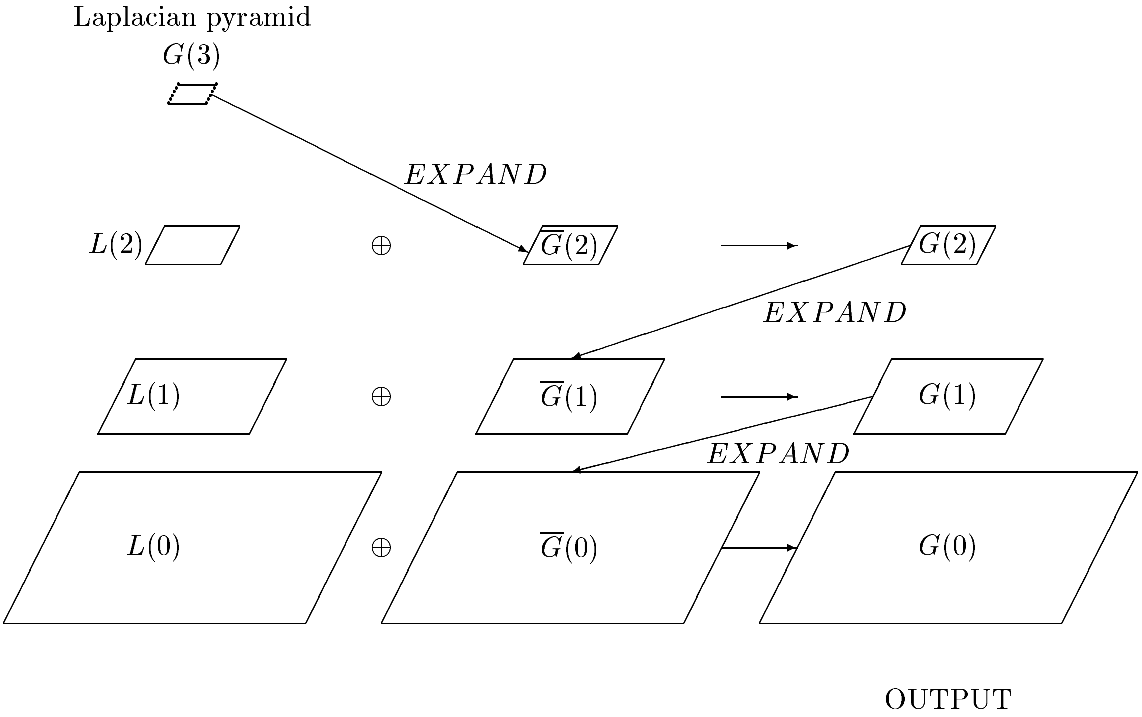


Figure 8: Reconstruction from the RE Laplacian pyramid.

After the Gaussian pyramid is built in this way, following steps are performed:

1. Define the interpolation function $EXPAND$ using the Gaussian filter which is the reverse function of $REDUCE$.
2. The 'reduce - expand' RE Laplacian pyramid is built by:
$$L(l) := G(l) - EXPAND(G(l+1)) \text{ for } l := 0, 1, \ldots, t-1 \qquad (20)$$
t is the top level. See Fig. 7.
3. Reconstruction of $G(k)$ is exact (Fig. 8):
$$G(k) := L(k) + EXPAND(G(k+1)) \text{ for } k := t-1, t-2, \ldots, 0. \qquad (21)$$
4. Hence storing $G(t), L(t-1), L(t-2), \ldots, L(0)$ is sufficient for exact reconstruction of the original image $G(0)$.
5. Quantisation of $L(l)$ shows only small and visually not recognizable errors, but large reduction rates: < 1 Bit/Pixel.

An alternate way of constructing Laplacian pyramids is also defined in terms of Burt's operations [BASv86] : the 'filter - subtract - decimate' FSD Laplacian by

$$L(l) := G(l) - F(G(l)) \text{ for } l := 0, 1, \ldots \qquad (22)$$

8.1 Image compression by $5 \times 5/4$ and $3 \times 3/2$ Laplacian pyramids

In [MK89b], [May89], [MK89a] Burt's image coding scheme has been compared for two different pyramid structures: $5 \times 5/4$ (Burt) and $3 \times 3/2$ (Mayer, Kropatsch). By using an efficient scheme for representing the odd levels of the $3 \times 3/2$ pyramid which would occupy 100% more storage if stored uncompressed because of the 45^0 rotation,we could achieve compression rates of 1:10 up to 1:20. Very similar results in quality and time complexity are reported by Burt and were also verified by simulating the $5 \times 5/4$ structure.

This result was estonishing because the reduction factor of 2 causes the $3 \times 3/2$ pyramid to have double as many levels as the $5 \times 5/4$ pyramid. More resolution levels have two advantages: they allow a much finer discrimination of object sizes and image reconstruction gives a much smoother impression.

If the receiving station of an image transmission expands the early transmitted low resolution images with the same kernel that has been used to achieve best compression then both pyramids show artefacts. In the $5 \times 5/4$ pyramid edges along the horizontal and vertical axes may be exagerated, in the $3 \times 3/2$ pyramid bright and dark spots at certain positions are introduced. These artefacts can be avoided by using a different expansion kernel without negative values.

8.2 Filtering in the space domain versus the frequency domain

In [OS87], filtering in the pyramid (space) is compared with filtering in the frequency domain. Besides the fact that 'processing in the Fourier domain is often inappropriate for images', because the data of a *picture* image as opposed to a 2-dimensional *signal* is usually **not periodic on the global scale**, the computational cost in the number of multiplies is compared:
a multiple low-pass transform can be computed either in the frequency domain by fast Fourier transform (FFT) which takes $C^L_{FFT} \approx N^2 \left(\frac{8}{9} + \frac{16}{3}\log_2 N\right)$ multiplications; or in the pyramid (space) domain involving convolution with a $p \times p$ kernel: $C^L_{S1} \approx \frac{p^2N^2}{3}$, which can be further reduced for symmetric and separable filters down to: $C^L_{S4} \approx \frac{pN^2}{3}$.

Similar results are obtained for the multiple band-pass (Laplacian) transform: In the frequency domain: $C^B_{FFT} \approx N^2 \left(\frac{26}{9} + \frac{28}{3}\log_2 N\right)$; and in the space domain using a $p \times p$ kernel: $C^B_{S1} \approx \frac{2p^2N^2}{3}$, which can be further reduced for symmetric and separable filters down to: $C^B_{S4} \approx \frac{2pN^2}{3}$.

8.3 Further applications

In [BA83b] Burt and Adelson used their Laplacian pyramid scheme to compute **multiresolution splines** of images that are useful for building image mosaics with smooth transitions. Let A, B be the two images and M a binary mask indicating where values from A should be taken (e.g.0) and where from B (e.g.1):

1. Laplacian pyramid LA of image A is built.
2. Laplacian pyramid LB of image B is built.

3. Gaussian pyramid GM of binary mask M is built.

4. $LS := LA + GM \cdot (LB - LA)$.

5. Image S is a reconstruction of LS.

In [ABv85], the Laplacian pyramid is used to indicate a significant change in a time-series of images. Let $I(t)$ denote the image taken at time t, let l_1 denote the level at which the change shall occur. Following procedure is able to initiate an **alarm** when an unusual situation occures in the field of view:

1. $D(t) := I(t) - I(t-1)$;

2. build Laplacian pyramid $L(l,t), l := 1, 2, \ldots, l_1$ with $L(0,t) := D(t)$;

3. square level l_1: $L(l_1,t)^2$;

4. build Gaussian pyramid $G(k,t), k := 1, 2, \ldots, k_{top}$ with $G(0,t) := L(l_1,t)^2$;

5. threshold $G(k,t), k := 1, 2, \ldots, k_{top}$: alarm.

9 The Wavelet Representation

The **Wavelet Representation** is a new method for image representation [Mal89]. It can be seen as an extension of Burt and Adelson's Laplacian pyramid concept [BA83a].

An image is approximated at a spectrum of resolutions in a wavelet orthonormal basis. It is shown in [Mal89] that a multiresolution approximation can be characterized by a unique **scaling function** $\phi(x)$. $\phi(x)$ is used to derive a discrete linear filter H and its mirror filter $\tilde{H}$, the reduction function of the $2 \times 2/4$ pyramid transform.

Given a sequence of increasing resolutions $(r_j)_{j \in Z}$, the details of an image at resolution r_j are defined as the difference of information between its approximation at the resolution r_j and its approximation at the lower resolution r_{j-1}. These image details must be uncorrelated. They are computed by decomposing the image function in a wavelet orthonormal basis which is uniquely characterized by the orthogonal wavelet $\psi(x)$. Similarly to $\phi(x)$, $\psi(x)$ gives rise to discrete linear filter G and its mirror filter $\tilde{G}$. $\tilde{G}$ is used to compute the *detail image levels* of the orthogonal wavelet representation (see Fig. 9). For example, this algorithm decomposes a 512×512 image into the three detail images $D^1_{2^{-1}}f, D^2_{2^{-1}}f, D^3_{2^{-1}}f$ all of size 256×256, three images $D^1_{2^{-2}}f, D^2_{2^{-2}}f, D^3_{2^{-2}}f$ all of size $128 \times 128 \ldots$, and the top level image $A^d_{2j}f$ of size $2^{9+j} \times 2^{9+j}, -9 \leq j \leq 0$. The detail images D^1, D^2, D^3 can be interpreted as spatially oriented frequency channels, e.g. D^1 shows the vertical high frequencies (horizontal edges), D^2 shows the horizontal high frequencies (vertical edges), and D^3 shows the high frequencies in both directions (the corners).

The wavelet basis provides a representation of the image that is midway between the frequency (Fourier) domain and the space domain [Mal89].

Given the wavelet representation of an image, the original image can be reconstructed using filters H and G similarly than from a Laplacian pyramid (see algorithm in Fig. 10).

The following assumptions are made for the wavelet representation:

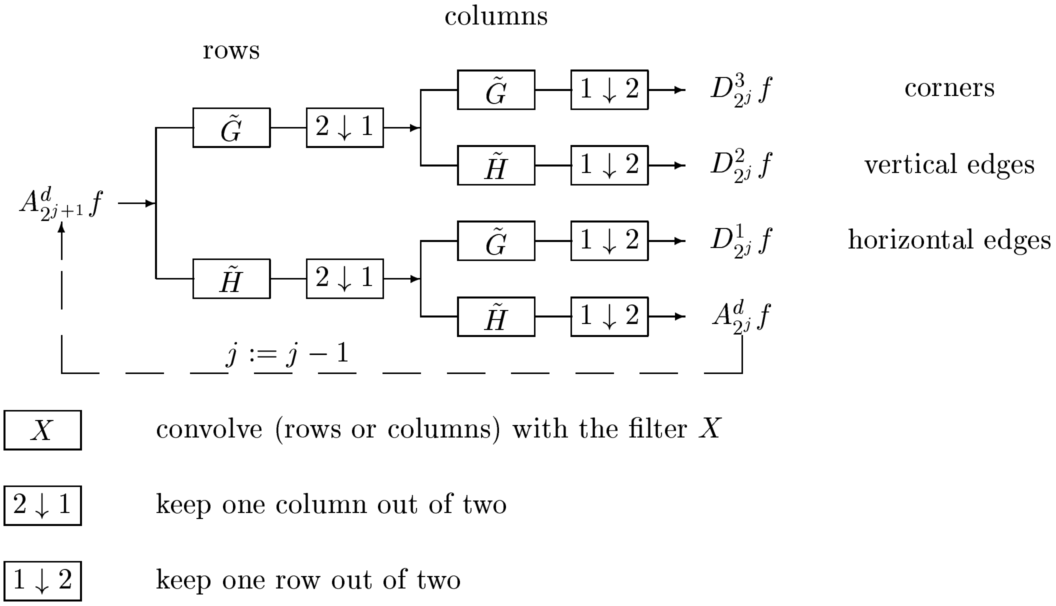


Figure 9: Mallat's wavelet decomposition of an image

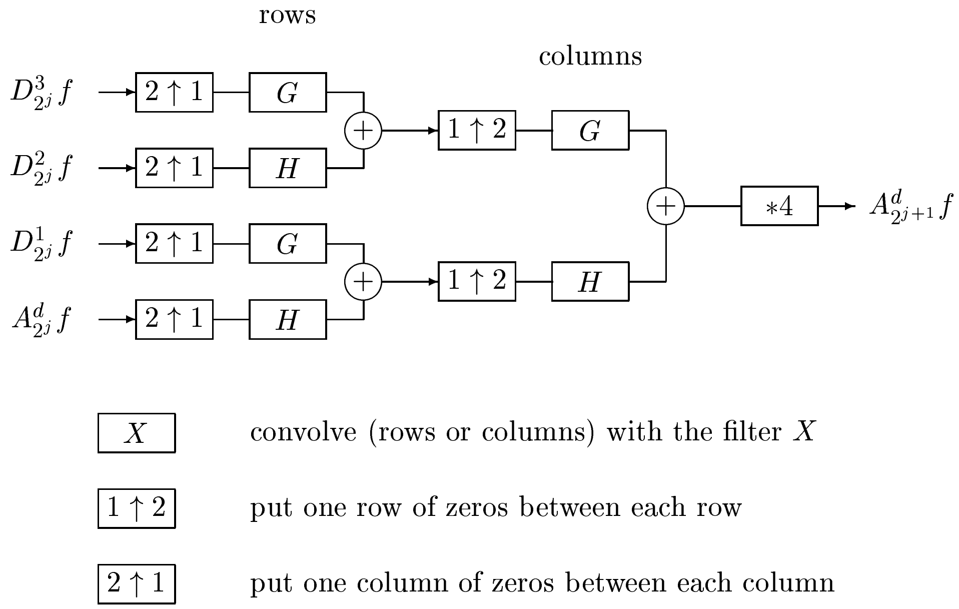


Figure 10: Image reconstruction from Mallat's wavelet decomposition

1. The continuation of the original image outside of the field of view is symmetric with respect to horizontal and vertical borders.
2. The 2D scaling function $\Phi(x, y) = \phi(x)\phi(y)$ gives extra importance to the horizontal and the vertical directions in the image.

Notice the difference to the Laplacian pyramid:

- The difference levels D^i are generated by the separate filter $\tilde{G}$ and not by the difference between two resolutions as in the Laplacian pyramid.
- The highest resolution difference levels of a 512×512 image are of size 256×256. The highest level of the corresponding Laplacian pyramid has 512×512 pixels.
- The 2D wavelet representation needs the same amount of space as the original image whereas the Laplacian pyramid needs 1/3 more than the original image.

10 Curves

A curve in an image may describe two different types of features:

- a boundary of a region or
- the skeleton of an elongated thin region (e.g. rivers or roads on satellite images; see section 10.3.2).

The recognition of curves often involves two steps: *First*, local instances of the curves are detected (section 10.1), e.g. edges, short curve segments. The primal sketch compiles the results of this first stage (section 10.2). *Second*, these elementary segments are connected to form sequences that may be matched with higher level curve models.

The shape of a region can be described by a curve. Different criteria judge the quality of a shape representation (section 10.3). Representations often depend on a special global parameter: the scale. Varying this parameter continuously modifies the representation and creates the scale-space (section 10.4). Different basic shape elements can be used in order to classify shapes locally. More global properties are then derived from the specific configurations of shape elements (section 10.5). One property of digital curves has been studied extensively: the straightness (section 10.6). Corners can be detected locally as non-straight curve segments (section 10.7).

10.1 Curve detection

There are numerous methods for detecting edges in digital images. Here we want to concentrate on methods that are somehow related to the goal of detecting curve segments and that allow the integration into a multiresolution approach.

10.1.1 In two dimensions

Julesz performed extensive experiments and found a subset of visual features that can be recognized by humans extremly fast [JB83]. The preattentive visual system can detect differences in a few local features almost instantaneously, regardless of where they occur. These features, called TEXTONs, are

- elongated blobs,
- ends-of-lines, and
- crossings of line segments.

Julesz conclusions give indications what the primitive elements of a vision systems could be and what the characteristic of the processes is:
'Preattentive texture perception is essentially a local process.'
'The preattentive system utilizes globally only the textons in the simplest possible way *by counting their numbers(densities).*'
'It does not perform Fourier analysis.'

Some of these characteristics are directly built into recognition algorithms. For example: Edelman's interesting pyramid algorithms MAC-1 and MAC-2 [Ede87] try to realize the author's hypothesis: 'The computational basis of human perception of connectivity in thin figures is some kind of terminator counting'.

The early work of Ramer [Ram75] shows some interesting analogies to pyramidal structures: Line structures are extracted from gray level images in two phases:

1. A nonlinear transformation produces a 2D **stroke array**. The structure of this transformation is an $(8 \times 8 - 12)/16$ overlapping reduction. The octagonal stroke domain, which corresponds to the reduction window in pyramids, is embedded in an 8×8 window from which three cells are excluded at each corner.

2. The strokes are assembled into **streaks** using graph searching techniques. A streak is an ordered set more than two strokes such that every two strokes adjacent in the ordered set are 8-neighbors in the stroke array. The cellular representation uses the 3-bit Freeman chain code to point to the next stroke of the streak.

   Streaks are classified into **prime** (nonblurred and nonparasitic), **parasitic** (streaks parallel to longer streaks), and **blurred** streaks. Prime streaks are used to form junctions.

Burns [BHR86] extends the observation window for detecting a line on both sides of the line: 'line support regions'. He tested his approach with four natural examples: a building on dark background, a person in front of a house, a house that is partly covered by a tree, and an areal photograph.

Rosenfeld and Sher [RS87] fit a straight line to sets of edge pixels. Since the first fit L_0 depends on variations in direction (wiggles), a second fit includes both the magnitude $m(P)$ and the deviation in direction $\theta(P)$ of the edge pixels from the line L_0:

1. The standard SOBEL edge detector ([RK82, Section 10.2]) gives edges P with $(m(P), \theta(P))$;

2. fit line $L_0 := min \sum_P m(P) d^2(P, L_0)$, where $d(P, L_0)$ is the perpendicular distance of pixel P from line L_0; set $i := 0$;
3. $\theta_i :=$ slope(L_0) ;
4. fit $L_{i+1} := min \sum_P m(P) cos(\theta(P) - \theta_i) \cdot d^2(P, L_{i+1})$;
5. $i := i + 1$ and iterate (3., 4.).

For a non-noisy edge, the result is insensitive to wiggles.

Sher [She87] reports on advanced likelihood generators that evaluate the presence of a boundary curve. The basic operation involves counting mistakes:

- false positives: wrong boundary reported;
- false negatives: real boundary missed.

The operator is tuned ($\sigma = 12$) to the expected noise.

Wojcik's [Woj87] concept for detecting and representing curve segments shows similarities to the previously introduced concept of the observation window and to the principle of representing curves in pyramids as used in chapter 11.

A circular (observation) window is shifted along an image contour. Then adaptive thresholding delivers binary values, from which the Euler number is calculated: I_w denotes the number of objects, I_0 the number of background regions, and I_p the number of intersections with the window. Equivalence classes are built and form the primitive entities for describing the curves:

tip	if $I_w = 1 \wedge I_0 = 1 \vee I_p = 1$
segment	if $I_w = 1 \wedge I_0 = 2 \vee I_p = 2$
fork	if $I_w = 1 \wedge I_0 = 3 \vee I_p = 3$
junction	if $I_w = 1 \wedge I_0 = 4 \vee I_p = 4$
two segments	if $I_w = 1 \wedge I_0 = 3$
discontinuity	if $I_w = 2 \wedge I_0 = 3$

Wojcik makes four assumptions:

- a line is compact;
- the width is smaller than (a given) radius;
- the length is greater than the diameter;
- the line crosses the center of the window.

As a result following line attributes are calculated: mass, position, length, slope, index.

10.1.2 In three dimensions

Since most images are taken from the real three dimensional (3D) world one of the major problems in vision is the estimation of the 3D positions of the objects in the 2D image (e.g. [Har86a]). This problem is ill-posed and needs more information or assumptions to be solved based on single objects (e.g. a second stereo image, knowledge about the 3D shapes of the observed objects, etc.). Naito and Rosenfeld [NR88] are able to estimate 3D positions from orthoprojected line drawings.

Two assumptions:

1. many identical objects;
2. arranged in random orientations.

Four features: longest line; special positions; angle between two branching line segments; most frequent angle.

Four specific models:

- base line and height (example house plant);
- Length and angle (example pine branch);
- triangle (example flowers);
- angle (example tree).

10.2 Primal sketch

This section shortly summarizes approaches related with David Marr's Primal Sketch.

Marr's [Mar76] *Primal Sketch* is a symbolic, two-dimensional description of the significant gray-level changes in an image. It includes the type, the position, the orientation, and the fuzziness of the edge for each area of gray level change.

How good a description is can be determined from the quality of an image reconstructed only using this description. [Mar80] illustrates the possibility to reconstruct the image from the *Primal Sketch* to a reasonable degree.

10.2.1 Terrain related primitives

Based on Marr's Primal Sketch [Mar76], [Mar80] further primitive categories are added to the vocabulary. Various primitives are derived from an interpretation of the gray level image as a gray tone intensity surface comparable to a digital elevation model.

The **facet model** [Har84a] is used in [WLH85] to derive the following topographic categories from a digital image: **peak, pit, ridge, ravine, saddle, flat, hillside**. The facet model needs to interpolate a continuous surface from the discrete grid data. A generalized spline approach and the discrete cosine transform perform this task.

Very similar categories of terrain classes are used in [BF87]: **peak, pit, pass, ridge, ravine, slope, break, flat**. The process constructs a general hierarchical network of drain and ridge lines.

A further grouping and terrain related semantics are introduced in [FP87] for modelling 'real' digital terrains:

critical regions: hill, dale, catchment area, water shed, drainage basin, lake, patch;

critical lines: ridge, channel, cliff, break;

critical points: peak, pit, pass, ridge junction, channel junction.

Earlier, Crowley [CP84] represented the **peaks and ridges** of the DOLP transform of a gray value image in the $2 \times 2/2$ pyramid.

10.2.2 Shape related primitives

Asada's [AB86] *Curvature Primal Sketch* describes curves in scale space (see section 10.4). His multiscale approach segments the curve in three categories: **corner, end, smooth join**.

Watt's [Wat87] algorithm MIRAGE extends Marr's concept of the *Primal Sketch* [Mar76]. Let $I(r,\theta)$ denote an image in polar coordinates.

1. Filter with Laplacian of Gaussians with parameter σ_j:
   $R_j(r,\theta) := I(r,\theta) * (1 - \frac{r^2}{2\sigma_j^2})e^{-\frac{r^2}{2\sigma_j^2}}$
2. Split R_j into 2 images:
   $R_j^+ = \begin{cases} R_j & \text{for } R_j > 0, \\ 0 & \text{otherwise,} \end{cases} \quad R_j^- = \begin{cases} -R_j & \text{for } R_j < 0, \\ 0 & \text{otherwise.} \end{cases}$
3. Build sums of the positive and negative signals separately:
   $T^+(r,\theta) = \sum_j R_j^+,\ T^-(r,\theta) = \sum_j R_j^-.$
4. Comparing the two signals T^+ and T^- there are three possibilities:

   (i) $T^+ = T^- = 0.$

   (ii) $T^+ = 0$ or $T^- = 0$ but $T^+ \neq T^-$.

   (iii) $T^+ \neq 0$ and $T^- \neq 0$.

   Based on these classes, **edges** and **lines** (bars) are detected.
5. Statistics on T^+ and T^- are used to calculate the edge's and bar's attributes of **blur, location** , and **contrast**.

As a model for earliest stages of human low-level vision, MIRAGE contributes the following types of functions:

1. The gray level range is mapped so that its mean is zero and its dispersion is restricted.

2. Important features are emphasized and then detected. Their attributes are preserved and may be measured.

3. Features are grouped intelligently according to **spatial scale**.

4. The data rate into slow geometrical processes is controlled.

10.3 Shape representations

The paper of Fischler and Bolles [FB86] addresses the problem of partitioning (perceptual organization). Most of the techniques can be characterized by one of the four paradigms for partitioning a curve :

1. Local discontinuity detection;

2. best global description;

3. confirming evidence;

4. recursive simplification (producing a hierarchy of data sets).

Following two general principles must be satisfied by an effective technique:

1. Stability (see below) and

2. complete, concise, and complexity limited explanation.

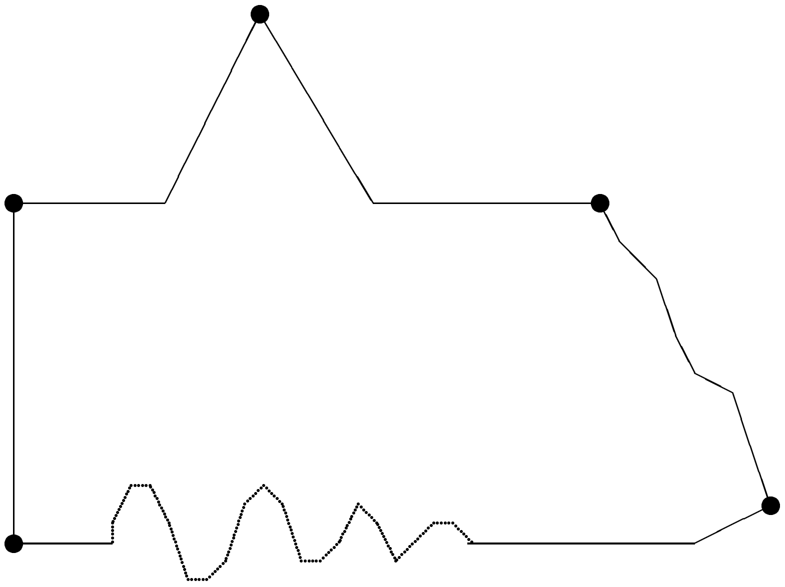


Figure 11: Describe shape by five points!

They illustrate the concepts by the example of five points on a line drawing (Fig. 11). Sometimes, the complex configurations that may appear in an image have to be described by a limited vocabulary (only five points). Depending on the task that a human was given, the five points have been placed differently. The three tasks were:

1. Reconstruction (placement in Fig. 11);
2. Recognition;
3. Segmentation.

Two key points summarize the basic ideas:

Key point 1: The partition problem does not have a unique definition, but it is parameterized: purpose, data representation, trade off between different error types (false alarms versus misses), etc.

Key point 2: Psychologically acceptable partitions imply an explanation. This explanation must satisfy criteria for accuracy, complexity, and believability. These criteria are formulated by a set of principles which can guide the construction of an effective partitioning algorithm.

Woodham [Woo87] defines the notions 'representation', 'description', 'stable'. A **representation** identifies a formalism, or language, for encoding a general class of shapes. A **description** means a specific expression in the formalism that identifies an instance of a particular shape, or a class of shapes, in the representation. A computation is **stable** if small changes in the input produce correspondigly small changes in the output.

Woodham's criteria for shape representation are: local support, stable, rich, multiple scales (!), object based semantics, human performance. A prototype bin-picking system is based on the concepts of photometric stereo, the 'Extended Gaussian image' and mixed volumes.

Mackworth [Mac87] refined Woodham's criteria and gave 12 criteria for shape representation:

1. computable efficiently;
2. local;
3. stable;
4. unique: e.g. $F : object \rightarrow representation$ is a function;
5. complete: F is total, every object has a representation;
6. invertible (rich, information preserving), F is one-to-one;
7. invariant under (shift, rotation, magnification) transformations;
8. scale sensitive (coarse-to-fine);

9. composite, explicite part-whole composition structure;
10. matchable, comparing two shape descriptions should return a description of their difference;
11. generic (through parameters or instantiation rules);
12. refinable, generic → specific.

Four domains for explicite shape representation are differentiated:

- $f(x)$
  scale space (see chapter 10.4): $F(x, \sigma) := G(x, \sigma) * f(x)$
  Steepest slope $F_{xx} = 0$ and $F_{xxx} \neq 0 \Rightarrow$ 'edge'.

  Note that $F_{xx}(x, \sigma) = G_{xx}(x, \sigma) * f(x)$.

  Monotonic property: filtering in scale space does not create generic zero crossings ([BWBD86] and [YP86]). This key property allows to define the interval tree in scale space. The scale space image uniquely characterizes the curve, hence it is invertible.

- $x(t), y(t)$
  This includes the MAT (medial axis transformation of Blum), the Brady ribbon (cf. smoothed local symmetries) and CODONS. See below for more detail.

  Curvature scale space: $\kappa(t) * G(t, \sigma)$, where t is the arc length and κ denotes the curvature.

  Generalized scale space image: $X(t, \sigma) = x(t) * G(t, \sigma)$ and $Y(t, \sigma) = y(t) * G(t, \sigma)$.

  Extended circular image [HW86].

- $z(x, y)$
  Haralick's *facet* model [Har84a];
  Shape-from-shading;
  Fractal modelling (compare with [BS88]);
  Shape-from-contour (block's world), Huffman-Clowes (-Waltz) labelling.

- $x(p, q), y(p, q), z(p, q)$
  Generalized cylinder
  Extended Gaussian image
  Superquadrics.

Leu and Chen present an approach for polygonal approximation of 2D shapes through boundary merging [LC88]. They differentiated between two types of shape representations:

a) region based: quad-tree, MAT, moment invariants, subregion decomposition.

b) boundary based: chain coding, polygonal/B-spline approximation, Fourier descriptors, Hough transformation.

Shape representations should

- preserve the shape information (to allow reconstruction, '*invertible*');
- allow fast conversion with moderate memory requirements ('*computable*');
- be insensitive to distortions ('*stable*'); and
- not be too sensitive to local noise.

Critical comments to their approach show general problems of shape representation methods:

1. It may encounter problems, when large smooth curves are approximated. Imagine a long circular arc (big radius). A fine (regular) approximation will have lots of runs with the same length and the same angle inbetween too neighboring runs. Hence the LMDAs[4] are *all* the same! (Or nearly the same.) It depends essentially on the scanning direction and on the curve before and after the circular arc. *Symmetry* is lost!
2. Another important feature of closed shapes is their *size* measured by the area. Since all points of the approximated polygon are *on* the original boundary, the area of the shape can shrink considerably, especially for convex shapes.
3. *Intersections* are also an important visual feature (cf. Julesz' TEXTONS). Since the approximated boundary moves spatially, a non-self-intersecting curve can become self-intersecting in the image plane.

10.3.1 CODONS

Richards and Hoffman [RH85] describe closed 2D shapes by a complete set of primitive shapes (CODONs) based on the differential properties of 2D curves. Their CODONs segment curves at concave cusps (minima of negative curvature) and describe them syntactically by six primitive elements by means of the configurations of curvature zero crossings and the sign of curvature:

∞: straight line;

0^+: positive curvature;

0^-: negative curvature;

1^+: one curvature zero crossing before reaching the curvature maximum;

1^-: one curvature zero crossing after reaching the curvature maximum;

2: two curvature zero crossings.

The notation is based on the observation that all curve segments lying between minima of curvature must have zero, one, or two points of zero curvature (points of inflection). The superscript in CODONs 0 and 1 indicate the sign of curvature at the CODON entry.

[4] The maximum arc-to-chord deviation of a locally minimum deviation arc (LMDA) is smaller than that of its neighboring arcs.

10.3.2 Generalized ribbons

'Ribbonlike' planar shapes can be defined by specifying a planar curve, also called spine or axis, and a geometric figure such as a disk or a line segment that 'sweeps out' the shape by moving along the curve, changing size as it moves. Such shape descriptions have been considered by Blum, Brooks, Brady, and others. Blum used a **disk** as generator, his representation is also known under the term 'medial (or symmetric) axis tranfsformation'. Brooks' generalized ribbons originate from a **line segment** that moves along the curve maintaining a **fixed angle** with the tangent of the spine. The resulting shapes are 2D versions of '*generalized cylinders*' sometimes called '*generalized cones*'. Brady's representation is based on 'local symmetry'. The generator is also a **line segment**, but it is required to make **equal angles** with the sides of the shape. Rosenfeld [Ros86a] compares those three representation classes with respect to generation and recovery. It is shown that the three classes of generalized 2D ribbons are strictly nested for straight spines if the ends are ignored:

$$Blum \subset Brooks \subset Brady. \tag{23}$$

10.3.3 Relating the skeleton with the boundary

In general, MAT representations do not satisfy the stability criterium. Ho and Dyer [HD86] smooth the shape to make it less sensitive to little perturbations. The points of the medial axis transform (MAT) are weighted by prominence measurements that are based essentially on correspondence between the MAT point and the curve segments of the shape that are uniquely associated with it. In this way 'major' and 'minor' axis are identified on the skeleton. The shape is smoothed by removal of minor axes.

Leyton's [Ley87] contribution relates symmetric axis with boundary description by CODONs [RH85]. 'The ends of the (Brady-)Ribbon [Ros86a] (symmetric axis) are close to the curvature extrema of CODONs.' CODONs are further subdivided into simpler segments called *limbs* , which have the shape of a spiral (= curve of monotonically changing curvature). Symmetry is identified as a crucial organizing principle of shape.

10.4 Scale-space representations

To generate a description of a continuous signal, the signal is fed into a detector (e.g. a filter) that produces the features (e.g. edges, peaks, curvature extrema) of the required description. It is interesting to note that, even in the continuous case, the description does not depend on the signal alone, but also on the **scale of measurement**, i.e. the size of the detector. The scale-space is spanned by the coordinate axes of the input signal (e.g. 1D, or 2D) plus an additional dimension, the scale σ, that is the continuously varying size of the detector. At a small detector size, many (detail) features are detected. With larger detectors, the degree of smoothing reduces the number of detected features. The goal of scale-space representations is to use detectors that obey the **monotonicity condition**: when moving from fine to coarse scale, no new features are detected; and when moving from coarse to fine scale, existing features never disappear.

We shall introduce in the following two papers that show that under certain assumptions there exists a **unique** smoothing kernel. Then we describe papers with two different characteristics in two sections:

- A (1D-) detector is applied to every dimension of the signal separately.
- A (2D-) spatial detector is used to derive scale-space representations.

10.4.1 The uniqueness of the Gaussian kernel

Babaud et al [BWBD86] consider as detector category filters $F(t,\sigma) : \mathcal{R} \times \mathcal{R} \mapsto \mathcal{R}$. The features are defined as first-order extrema of a twice differentiable (signal) function $f : \mathcal{R} \mapsto \mathcal{R}$. They formulate the uniqueness theorem (Table 6).

Table 6: Uniqueness theorem of Babaud et al

Under following assumptions:

1. σ is a **bandwidth** parameter for F:
$$\exists \text{kernel}\, h\, \forall t \in \mathcal{R}, \sigma > 0 : F(t,\sigma) = \frac{1}{\sigma} h(\frac{t}{\sigma}).$$
2. F is **symmetrical** in t:
$$\forall t \in \mathcal{R}, \sigma > 0 : F(-t,\sigma) = F(t,\sigma).$$
3. F is normalized:
$$\forall \sigma > 0 : \int_{-\infty}^{\infty} F(t,\sigma) dt = \int_{-\infty}^{\infty} h(v) dv = 1.$$
4. $$\exists p \in \mathcal{N}(\text{integers}) : h^{(2p)}(0) \neq 0.$$

The only filter kernel F in the convolution

$$\phi(t,\sigma) = f(t) * F(t,\sigma) = \int_{-\infty}^{\infty} f(u) F(t-u,\sigma) du \tag{24}$$

that guarantees the monotonicity condition

$$\phi_\sigma \phi_{tt} > 0 \tag{25}$$

wherever $\phi_t = 0$ and $\phi_{tt} \neq 0$, is the Gaussian

$$G(t,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t}{\sigma}\right)^2} \tag{26}$$

Yuille and Poggio [YP86] come to the same conclusion. However, their formulation is valid for any dimension (simply take the variable of image I as an n-dimensional vector):

'If $L(x)$ is a differential operator in any dimension that commutes with the diffusion equation, then the only filter that does not create generic zero crossings as the scale increases, e.g. solutions of

$$L(F * I) = const,$$

is the Gaussian.'

Their five assumptions about the filter F are imposed as conditions for 'nice scaling behaviour'. Although they are equivalent to Babaud's assumptions, they are presented from a different point of view:

1. Filtering is a convolution:
$$F * I(x) = \int F(x - \zeta, \sigma) I(\zeta) d\zeta.$$
2. The filter has no preferred scale σ:
$$F(x, \sigma) = (1/\sigma^2) f(x/\sigma).$$
3. The filter recovers the whole image at sufficiently small scales:
$$\lim_{\sigma \to 0} F(x, \sigma) = \delta(x).$$
$\delta(x)$ denotes the Dirac delta function.
4. The position of the center of the filter is independent of σ.
5. The filter goes to zero as $|x| \to \infty$ and as $\sigma \to \infty$.

Theorems:

1. In 1D, with the second derivative, the Gaussian is the only filter obeying the five conditions which never creates zero crossings as the scale increases.
2. In 2D, with the Laplacian operator, the Gaussian is the only filter obeying the five conditions which never creates zero crossings as the scale increases.
3. In 2D, with the directional derivative along the gradient, there is no filter obeying the five conditions which never creates zero crossings as the scale increases.

10.4.2 1D-smoothing of parametrical curves

A parametrical curve

$$\Gamma = \{(x(w), y(w)) \mid w \in [0, 1)\} \tag{27}$$

with normalized arc length w is evolved by $G(u, \sigma) = \frac{1}{\sigma\sqrt{2\pi}}\ e^{-u^2/2\sigma^2}$ to yield the evolved curve $\Gamma_\sigma = \{(X(u, \sigma), Y(u, \sigma)) \mid u \in [0, 1)\}$. The curvature of Γ_σ

$$\kappa(u, \sigma) = \frac{[X_u(u, \sigma) Y_{uu}(u, \sigma) - X_{uu}(u, \sigma) Y_u(u, \sigma)]}{(X_u(u, \sigma)^2 + Y_u(u, \sigma)^2)^{3/2}} \tag{28}$$

defines the *curvature scale-space image* of Γ by $\kappa(u, \sigma) = 0$.

In the above parametrical curve representation, curves in scale-space correspond to curvature zero-crossings. They are used in [MM86b] for shape matching with an adaptation of the Uniform Cost Algorithm, a special case of the A^* algorithm. The examples include the shoreline of Africa.

The comment in [Gos86] and the reply [MM86a] refer to an important difference between two ways of smoothing: 'Convolving a region boundary with a 1D Gaussian **is not the same as** convolving the region as a solid with a 2D Gaussian and extracting the obtained region boundary'. We shall see the effects of 2D smoothing in Bergholm's approach in the next section.

Although w is the normalized arc length parameter of Γ in equation 27, the parameter u is not normalized with respect to the smoothed curve of Γ_σ. The *renormalized curvature scale-space image* [MM88] of Γ is defined by $\kappa(\omega, \sigma) = 0$ where ω is the normalized arc length parameter after reparametrizing Γ_σ. The renormalized curvature scale-space enhances the utility for shape matching of similar curves if they contain radically different scale related phenomena.

Following scaling properties of the curvature scale-space image are reported in [MM88]: The monotonic property of planar curves Γ in C_2 does not hold in general, but if all curves Γ_σ are in C_2, then all extrema occuring at regular points on contours in the curvature scale-space of Γ are maxima. Furthermore, invariance under affine transformation, connectivity and closure preservation, and the interrelationship between cusps of Γ_σ and self intersecting $\Gamma_{\sigma-\delta}$ are proven.

In Asada and Brady's [AB86] curvature primal sketch (compare section 10.2), scale-space is sampled at scales 4, 5, 7, 11, 15, and 22, which are approximations of $4(\sqrt{2})^k, k = 0, 1, 2, 3, 4, 5$. Two primitive features, **corner, smooth join**, and three compound features, **end, crank, bump/dent**, constitute the vocabulary that is produced by interpreting the significant changes in curvature at various scales.

Reconstruction of a curve from its scale-space representation is possible [Mok88]: A single point on one curvature zero-crossing contour in the curvature scale-space image of planar curve Γ determines Γ uniquely up to constant scaling, rotation and translation (except on a set of measure zero).

The *torsion scale-space* description of a space curve Γ represents that curve up to a class represented modulus a scale factor by the function $\beta(u) = \tau(u)\kappa^2(u)$ where $\tau(u)$ and $\kappa(u)$ are the torsion and curvature functions of Γ respectively.

The results presented in [Mok88] indicate that

- a polynomially represented planar curve in C_1 can be reconstructed using four points of its curvature scale-space image at one scale and
- a polynomially represented space curve in C_1 can be reconstructed modulus the class represented by $\beta(u)$ using seven points of its torsion scale-space image at one scale.

In principle, the scale-space need not be built on top of the original curve, any reversible transformation of the curve can be used as well. As an example let us mention briefly the **extended circular image** [HW86]. There, a simple, convex, closed curve is given by the radius of curvature R as a function of normal direction ψ, $R(\psi) = \frac{dw}{d\psi}$, where w is the arc-length. Horn and Weldon's approach differs from the others by (1) the use of the radius of curvature R instead of the curvature, κ, and by (2) the use of normal angle ψ instead of the arc-length w as independent variable. They show in [HW86] that filtering the 'extended circular image' preserves the closure of the curve.

10.4.3 Detecting features in 2D smoothed images

In this section we consider the behaviour of features in a scale-space that is created by 2D-smoothing of an image I. The features of interest are edges and boundaries.

Bischof and Caelli [BC88] address the problem of parsing boundaries through scale-space. They define the spatial stability for boundaries:
(A1) A boundary is a region of steep gradient and high contrast.
(A2) A boundary is **well-defined** if it has no neighboring boundaries.
Spatial stability has a "noise cleaning" effect. 'All zero-crossing curves are either closed or cross the image boundary (see ref. [24] there).'

Bergholm [Ber87] studies deformations of edges and contours that are produced by 2D-Gaussian smoothing. His 'edge focusing' method has following characteristics:

1. It smooths the image with a 2D Gaussian.
2. It detects edges as maxima along the gradients.
3. Scale-space reduces resolution continuously.
4. The deformation of four elementary edge structures, step edge, double edge (often called 'line'), corner edge (L-junction), edge box (blob), can be described by combination of Gaussians.
5. Blurring may transform an edge contour in four ways:
   (a) rounding-off corners,
   (b) expansion of the two edges of a double edge profile,
   (c) isolated closed contours are transformed into a circle, and
   (d) merging of separated closed contours into one contour if they are not isolated.
6. It creates a coarse (edge) image and detect edges there.
7. It tracks edges at higher resolution (e.g. until $\sigma = 0.7$) only in the neighborhood of coarse edges.

Results are very similar to structural noise cleaning in the $2 \times 2/2$ curve pyramid (see chapter 11).

In [SB88] this approach is extended for the extraction of diffuse edges. Two categories of edges are distinguished: 'diffuse' and 'non-diffuse' (or 'true'). Non-diffuse edges are 'object edges' that correspond to object boundaries. Diffuse edges are mainly due to illumination patterns such as shadows. The presented method is based on 'edge focusing' [Ber87]. Two blurring transformations are introduced in addition to

(1) rounding-off,

(2) expansion,

(3) transformation into circles,

(4) merging,

by diffuse edges:

(5) attraction of weak elements,

(6) wiggling.

Two observations in scale-space allow the algorithm to distinguish between the two categories of edges: First, diffuse edges tend to split up from coarse to fine resolutions and the pixels seem to move in two directions out from the edge. Secondly, true edges move very little and only in one direction.

The method compares two different levels of resolution (5 and 9, or 4 and 8, in the focusing numbering):

1. Count edges in the neighborhood of the coarse level (CE).
2. Compare coarse and fine resolution edges: some correspond and some are 'new' neighbors.
3. Count edges in the fine resolution (FE).
4. For diffuse edges $FE >> CE$.
5. Map the results into the finer resolution.
6. Track edges to find continuous edge segments.
7. Compare edge segments whether it is part of a shadow or an edge.

Conclusion: 'In a fine resolution, short segments are almost all shadow-pixels.' The two examples show good results.

An approach where scale-space continuation enlarges the capture region around features of interest are **SNAKES**[KWT87]. A snake is an energy-minimizing spline. Forces control the continuity: internal energy, snake pit, image forces are the line functional, the edge functional, scale-space, and the termination functional.

10.5 Discrete representations

When a curve is to be represented by discrete elements there are many variations from representations closely related with the underlying raster up to representations that capture the major properties of the curve.

Chain codes have been introduced by Freeman [Fre61], [Fre74] and are widely used as shape representations. They specify relative movements within a connected sequence of pixels.

In [WR87] chain codes are used to generate lines and circles. Other types of chain codes are also possible. We shall introduce the **RULI-chain code** in chapter 11. RULI chains have the advantage that they can be generalized into a lower resolution by applying syntactical substitutions.

A step further away from the raster are polygonal representations. These are sequences of straight line segments such that the next segment starts at the end point of its predecessor. So-called **vectorizers** perform (e.g. [Pav84], [Pav86]) the raster-to-vector conversion.

Polygons are widely used in computer graphics. Dunham and Glanz [DG86] present optimal approximations of planar shapes by polygons. Samet [SW84] stores polygons efficiently in quadtrees. Ayache and Faugeras [AF86] have built a system HYPER in which shapes are matched using polygonal approximations.

The objective in [KPK87] is to segment a shape which is given by a closed polygon. The presented algorithm segments the shape based on collinear segments. It defines two types of **cojoins**: type I (concave) and type T (convex).

1. Extract three kinds of collinear line segments: forward collinear, backward collinear, anti-collinear. The result is a set of MCCS (minimal cojoined convex subpolygons).

2. Resolve conflict cojoins: a) inter-conflict (crossing), b) share-conflict.

10.5.1 Splines

The two approaches in this section allow to describe **smooth curves**.

The authors of [BPD88] propose a method for coding a binary image contour using Bézier approximation. Bézier curves have the advantage of being simple and independent of the coordinate axes.

A set of key pixels is detected on the contour. It decomposes the contour into arcs and straight line segments. Key pixels are close to the points of maxima and minima of curvature. The segments between every two key pixels can then be classified as either an arc or a straight line L. An arc may again be of two types, with or without an inflection point (cf. CODONs of [RH85]). An arc with inflection point is splitted into two arcs CC without inflection point at the inflection point. The coding stores the key points and an additional point for arcs CC to allow quadratic Bézier approximation at the reconstruction.

Straight lines L are reconstructed using Bresenham's algorithm [Bre65]. Arcs CC are reconstructed using a forward difference scheme.

Reported compression rates are significantly less than in the CRLC (contour run length coding) and in the DLSC (discrete line segment coding) methods.

Pham [Pha89] introduces a general form of **conic B-splines** which allows representation of **circular, elliptic**, and **hyperbolic arcs** in addition to parabolic arcs.

A conic B-spline is defined by a sequence of control vertices $\{V_i\}$ and knots $\{K_i\}$:

$$(V_0, K_0, V_1, K_1, V_2, \ldots)$$

Since the curve goes through the knots K_i, and its tangent at the knot is determined by the line segment (V_i, V_{i+1}), it must be located on this line segment.

Conventional B-splines interpolate square Bézier curves through points (K_{i-1}, V_i, K_i) by

$$C_i(t) = B_{-1}(t)K_{i-1} + B_0(t)V_i + B_1(t)K_i \tag{29}$$

where Bézier's basis functions $B_k(t)$ are defined as follows:

$$\begin{aligned} B_{-1}(t) &= (1-t)^2 && (30) \\ B_0(t) &= 2t(1-t) && (31) \\ B_1(t) &= t^2 && (32) \end{aligned}$$

Pham uses different basis functions $\bar{B}_{-1} = B_{-1}/2, \bar{B}_0 = B_{-1}/2 + B_0 + B_1/2, \bar{B}_1 = B_1/2$. They allow him to express the same curve $C_i(t)$ in terms of control vertices (V_{i-1}, V_i, V_{i+1}):

$$C_i(t) = \bar{B}_{-1}(t)V_{i-1} + \bar{B}_0(t)V_i + \bar{B}_1(t)V_{i+1} \tag{33}$$

Note that $K_i = (V_{i-1} + V_i)/2$ relate the two representations.

Using coordinates V_i^h in **homogeneous space**, a circular arc $C^{(h)}(t) = (1-t^2, 2t, 1+t^2)$ in the 2D plane can be represented by a conic B-spline with homogeneous vertices $V_0^{(h)} = (1,-1,1), V_1^{(h)} = (1,1,1), V_2^{(h)} = (-1,3,3)$. The corresponding knots are $K_0^{(h)} = (1,0,1), K_1^{(h)} = (0,2,2)$. In the 2D plane the sequence V_0, K_0, V_1, K_1, V_2 becomes $(1,-1)$, $(1,0)$, $(1,1)$, $(0,1)$, $(-\frac{1}{3},1)$. Note that the reconstructed knot $K_2 \neq (V_1 + V_2)/2$. Therefore the circular arc cannot be reconstructed from V_0, V_1, V_2 alone.

The curve fitting process consists of following steps:

1. Specify the control sequence $(V_0, K_0, V_1, K_1, V_2, \ldots)$.
2. Compute the homogeneity factors h_i recursively by:

$$\begin{aligned} h_0 &:= 1 && (34) \\ h_{i+1} &:= h_i \frac{|V_i - K_i|}{|K_i - V_{i+1}|} && (35) \end{aligned}$$

3. $V_i^{(h)} := (V_0 h_i, h_i)$
4. Use De Boor's algorithm to calculate polynomial B-spline points $P_j^{(h)} = (\bar{P}_j, h_j)$.
5. $P_j := \bar{P}_j / h_j$

10.5.2 Moments

Sluzek [Slu88] addresses the problem of contour matching by means of partial moments of curves. It can handle also the problem of partially occluded contours. No results with 'real' data are reported.

10.6 Digital straight lines

As we have seen already, straight lines play an important role as an intermediate shape representation. Many researchers have been interested in methods for detection of straightness from chain codes [Ros74], [Ron85], [Pha86], [LL88] and for efficient generation of straight lines in a raster [Bre65], [Pit82], [Bre85], [CP85] .

10.6.1 A sufficient condition for digital straightness

A necessary and sufficient condition for chain coded curves to be straight is derived in [KT89] and [Toc87]. An algorithm is given that checks the straightness of both a Freeman chain coded (FCC) and a RULI chain coded sequence.

Two classes of codes are defined by comparing neighboring codes: **repetition** and **single** codes. A code is called a repetition code if one of its neighbors in the sequence is the same, if both are different, it is called single. Repetition and single code of a chain coded straight line must be uniquely defined. Freeman [Fre74] found further that the two different codes must have neighboring orientations, e.g. FCC differ by one.

Sequences of identical codes are collected to 'runs' separated by single codes. The lengths of these runs form a new sequence of integer numbers. Neighboring elements of such a sequence can be compared in the same way as the original chain codes. Hence repetition and single codes can be determined in this reduced sequence as well. It is shown that uniqueness of repetition and single codes is also required for the reduced sequence if it represents a straight line. The difference of the two must not exceed 1.

This process is repeated until the reduced sequence is so simple that the straightness can be decided immediately, e.g. if all elements are the same. It is proven that this procedure eliminates all non-straight lines during evaluation.

One single iteration depends only on the number of the sequence's elements and the reduced sequence is at most half as long. Therefore the algorithm takes on the order of $n + n/2 + n/4 + n/8 + \cdots < 2n$ steps if n is the number of original chain codes.

Using the repetition and single codes of all iterations the possible slopes of the line can be computed.

Some correspondences with the different type of reduction in the curve pyramid (chapter 11) are additionally reported in [Toc87]. However, the final goal of a parallel algorithm for straightness detection in less than $\mathcal{O}(n)$ time is not reached.

10.6.2 Other straight line algorithms

In [SBT85] the k-curvature method is used to derive the average period of a chain coded straight line.

The considerations in [DS86] are based on chain code of straight lines. The authors estimate the length of the line by measurements on the chain string. The algorithm has 3 steps:

1. digitization → chain string;
2. characterization: (n,p,q,s) ;
3. calculation.

Krishnaswamy and Kim [KK87] define SLOPE, PARALLELISM, and PERPENDICULARITY of digital LINE SEGMENTS. Related notions are also defined: dig. image, dig. arc. These are embedded in a good overview of discrete geometry, which comes along with some useful references to this subject. The presented algorithm for SLOPE has complexity $\mathcal{O}(N)$.

Melter and Rosenfeld [MR89b] address the stability of repeated digitization and recovery of straight lines. A continuous line of the Euclidean plane is digitized by **rounding**. It creates a set of lattice points, also called a scatter diagram. For convenience it is assumed that the points have distinct abscissas. A **least squares line** for the scatter diagram is a line which minimizes the sum of the squares of the vertical distances to the data points.

For a line $y = \frac{x}{2n+1}, n \in \{1, 2, \ldots\}$ the least square approximation converges to the original line as more points are digitized. Furthermore, let α_k be the slope of a least squares line of a line $y = \alpha x$. Then for every positive integer m, $(\alpha + m)_k = \alpha_k + m$.

Conversely, the three points $(0,0), (1,a), (2,b)$ contained in a digital line reoccure if their least squares line is digitized.

The linearity of a scatter diagram is measured by the **correlation coefficient** r_{xy}. Let S be the digitization obtained by rounding the line $y = \alpha x$ for abscissas $(0, 1, 2)$. Then $r^2_{xy}(S) \geq \frac{3}{4}$, or the three points lie along a horizontal line. The correlation coefficient for the line $y = \frac{x}{2n+1}, n \in \{1, 2, \ldots\}$ approaches 1 for increasing size of the scatter diagram.

A set of lattice points with k consecutive abscissas is defined a **noisy line segment** if the correlation coefficient of every three consecutive points is either $\geq \sqrt{3}/2$ or $\leq -\sqrt{3}/2$ (sign fixed for the line), or the three points lie along a horizontal line.

It is shown that noisy lines are a generalization of digital lines.

A new type of **digital connectedness** is also discussed; it is intermediate between the usual 4- and 8-connectedness. A class of paths in the digital plane slightly more restrictive than 8-paths is defined. The length of the shortest such path between two points defines a metric.

10.7 Corners

Polygons have been introduced as connected sequences of straight line segments. Vectorization in this concept searches for chains of maximum length that could be the result of digitizing straight lines. An alternative defintion specifies a sequence of points that are connected by straight lines. This definition suggests a different vectorization strategy:

1. Find **corners** as points where the digital curve changes its direction significantly and
2. connect them by straight line segments.

Obviously this approach has problems in detecting and representing smooth curves at high resolution.

We have shown in [FK89] that curvature extrema can be related to corners in multiple resolutions. Consider following example: a curve consisting of three elements, L_1, C_2, L_3. Let the circular arc C_2 connect smoothly two straight line segments L_1, L_3. Four parameters describe this configuration: the radius r of C_2, the angle θ between L_1 and L_3, and the lengths $l_1 > 2r, l_3 > 2r$ of L_1, L_3 respectively. If r is much larger than the side length of a resolution cell, C_2 will be recognizable in the corresponding image. If we reduce the resolution, the angle θ remains the same as long as both L_1 and L_3 are represented. But the other three parameters, r, l_1, l_3, will decrease with respect to the reduced resolution. A smaller radius implies a larger curvature. If we continue reducing the resolution until the size of a cell approaches $(2r)^2$, C_2 will disappear, and L_1 and L_3 will form a *corner* with angle θ. This principle relates curvature extrema and corners through different resolutions.

The results of this work have been published in German [FK89], and in a Diploma thesis [Fer89]. An English article [FK94] has been submitted for publication. The key points of this work can be summarized as follows:

1. In a reduced resolution, curvature extrema may disappear. This causes symbolic curve representations like CODONs to change. The modifications are local and have the effect of shrinking the length of the description. The order in which curvature extrema disappear is determined by the length of the two adjacent limbs.

2. Corners are locally detected as **non-straight** configurations of RULI-chains.

3. Corners are linked in adjacent levels of a curve pyramid. In contrast to continuous scale-space (chapter 10.4), corners need not always have a corresponding corner in the pyramid level directly below. Artefacts caused by digitization of the continuous scale space are identified as such.

4. Two sufficient conditions of the corners before digitization determine if the corners would be recognized in any possible placement and orientation. The corresponding limits depend on the size of the detector k ($k = 3$ and $k = 5$ elements have been used in the experiments):

   - Upper limits $\Theta(k)$ for the corner's angles ($\theta < \Theta(k)$ in the above example);
   - Lower limits $l_{min}(k)$ for the length of the corner's legs ($l_1 > l_{min}(k), l_3 > l_{min}(k)$ in the above example).

5. Curvature extrema become corners in a bottom-up reduction process and remain corners until they disappear.

6. Efficient top-down **matching** strategies are described and tested by simple examples (contour of a plier, a contour from [FB86] and [PR87]).

The following algorithm by Aviad [Avi88] combines local corner detection with confirming evidence.

10.7.1 Corner detection by imperfect sequences

Aviad [Avi88] addresses the problem of locating right angle corners in unsmooth digital lines, employing the concept of **imperfect sequences**. Four local classifiers (see Fig. 12) produce four sequences of 0 and 1. In a noise-free curve there is at least one classifier that has a different response on the two sides of the corner.

Next, each of the four sequences is partitioned by delineating imperfect sequences. Let $s_0, s_1, \ldots s_n$ denote such a sequence. In a noisy sequence, s_i changes many times due to noise. An Imperfect Sequence Detector (ISD) guesses when the sequence really changes the state. Following assumptions are made:

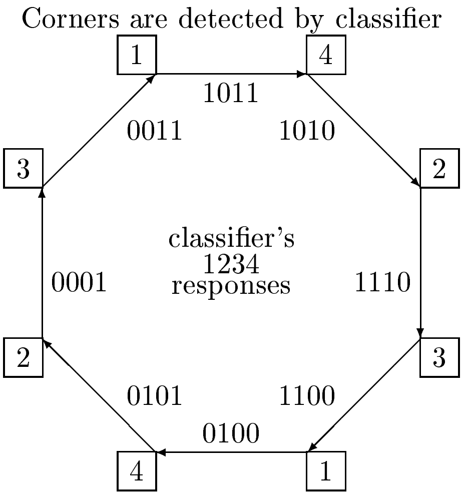


Figure 12: Aviad detects corners in 8 directions with 4 binary classifiers

1. Sampling is known faster than the state changes.
2. Two thresholds k_0, k_1 are known, such that sequences $\underbrace{00\ldots0}_{k_0}, \underbrace{11\ldots1}_{k_1}$ indicate beyond reasonable doubt the presence of state 0 and 1, respectively.

The ISD algorithm can be formulated as follows:

```
THRESHOLD(0):=$k_0(k_0 + 1)/2$;
THRESHOLD(1):=$k_1(k_1 + 1)/2$;
SCORE(0):=SCORE(1):=0;
INC(0):=INC(1):=1;
k:=1;
BEGIN(k):=1;
ACCEPTED:= none;
for i:=1 until n do

    PREV:=SCORE($s_i$);
    SCORE($s_i$):=SCORE($s_i$) + INC($s_i$);
    INC($s_i$):=INC($s_i$) + 1;
    INC($1 - s_i$):=1;
    if SCORE($s_i$) $>$ THRESHOLD($s_i$) then
        ACCEPTED:=$s_i$;
        k:=k + 1;
        SCORE(0):=SCORE(1):=0;
        INC(0):=INC(1):=1;

    if ACCEPTED=$s_i$ and SCORE($s_i$) $>$ SCORE($1 - s_i$) then
```

```
        SCORE(0):=SCORE(1):=0;
        INC(0):=INC(1):=1;

    if PREV = 0 and ACCEPTED $\neq s_i$ then
        BEGIN(k):=i;
```

The variables SCORE and INC update the support in accordance with confirming or contradicting samples s_i. If one hypothesis is accepted (ACCEPTED:=0 or 1), the other is rejected. If the SCORE of the currently ACCEPTED hypothesis exceeds the SCORE of the competing $(1 - s_i)$ hypothesis, the confidence buildup in the unaccepted hypothesis is overridden. The beginnings of the sequences are recorded in array BEGIN. A new beginning is set whenever the current sample s_i is different from the current hypothesis and the competing hypothesis has no support from the PREVious step.

The partition points BEGIN(k) of all four sequences are collected as corner candidates. Those that form angles of at most 135^0 are selected as the final break points of the curve.

Experiments with the new corner detector are reported and compared with the results of Fourier approximations. The new corner detector performs very well in detecting and localizing significant corners.

Note that the parameters k_0 and k_1 measure the required minimum length between two corners (compare [FK89]). Furthermore, the angle constraint is checked also before the final break points are selected.

10.7.2 Other corner detectors

Freeman [Fre77] detects corners in chain coded curves.

In [HR85], curve segments are linked in a pyramid to detect significant corners.

In [CH88] corners of curves are detected in digital images by measuring the degree of bending via an extended (k-step) 3×3 mask. The bending values are identified as a measure of prominence of a corner.

Curvature estimation is the focus of [O'G88b] and [O'G88a]. In [O'G88a] analytical and empirical comparisons have been made of the difference of slopes (DOS) and Gaussian smoothing methods for curvature estimation. For the DOS approach, the DOS^+ method with parameter values of $(M \to +0, M > 0)$ yields the best SNR[5] results for lines of small signal angle and high line noise. The DOS^+ method yields similar results to the Gaussian smoothing method. However for small signal angle and high noise, both analysis and examples show the DOS^+ method to perform better for signal detection. In [O'G88b] the DOS^+ method is described in more detail for detecting curve and corner features from the curvature plot as functions of M and L.

[5] Signal-to-Noise-Ratio

11 The Curve Pyramid

Problems are encountered when thin structures such as curves and lines are to be detected in pyramids. In the intensity pyramid, curves disappear after a few reduction steps because the reduction averages the intensity over compact regions (receptive fields). Therefore an edge detector is applied to the intensity pyramid yielding an edge pyramid (e.g. [Shn81], [Har85]). The problem in an edge pyramid is to combine the edges simultaneously detected at different resolutions.

Kelly was the first [Kel71] to use a two level ($5 \times 5/25$) multiresolution hierarchy for edge detection. He finds edges and lines first in the reduced resolution and uses these as a plan to constrain the search in the higher resolution.

Others extract linear features [Shn82] and contours [TASM83] using pyramids or quadtrees [SSW85].

Hong and Rosenfeld [HSHR83] use **good continuation** as criterium to integrate local features. Hartley [HR85] smoothly joins polynomial curve models. The connectivity of curves was the primary goal in the concept of the curve pyramid which is presented in the sequel.

11.1 The $2 \times 2/2$ curve pyramid and the RULI-chain code

The concept of representing linear structures of digital images by **curve relations** is introduced in [Kro85a]. A cell of the underlying square grid is considered as the observation window through which a curve is observed. A curve under such an observation window connects the two points where the curve crosses the boundary of the cell. Since these two points are located on two of the four sides of the square, N, E, S, or W, also the two sides are related by that curve (see example in Fig. 13).

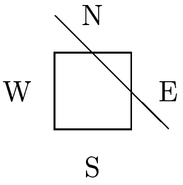


Figure 13: A NE curve relation

The set of *curve relations* of all observed curves is stored in every cell. It approximates the observed curves.

The accuracy of the approximation is determined by the resolution of the cell. A lower resolution representation is derived deterministically by two steps (Fig. 14):

- subdivision of the cell contents by introducing a **diagonal** and
- merging curve relations of two neighbor cell by **transitive closure**.

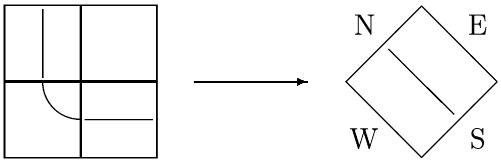


Figure 14: Reduction in the 2 × 2/2 curve pyramid

Iterative application of this generalization procedure yields the 2 × 2/2 curve pyramid (see Fig. 20, page 76).

There exists a chain code representation equivalent to the 2D concept of the curve relations: the **RULI chain code**. The four elements of this chain code $\{R, U, L, I\}$ determine the relative change in moving direction: go right (R), make a U-turn (U), go left (L), continue in the same direction (I). A RULI-chain is invariant to shifts and rotations by 90^0. The generalization of a curve represented by this RULI chain code is formulated by a 5 step formal substitution grammar. It is shown that both generalizations are identical for single curves. The grammar is further used to prove that the number of codes of a non-closed curve always shrinks after generalization *'length reduction property'* .

In [Kro86e] three types of square grid pyramids are checked whether they are suited for multiresolution curve representation: the 2 × 2/4 pyramid, the 4 × 4/4 pyramid, and the 2 × 2/2 pyramid. Such a representation must satisfy the following goal: **the highest level up to which a curve is still represented before it disappears must be related to the area that is traversed by the curve**. Non-overlapping pyramid structures do not satisfy this criterium. The *length reduction property* of the 2 × 2/2 curve pyramid is extended to **closed curves**. The number of code elements of closed chains is shown to shrink at least every second step.

The articles [Kro86b] and [Kro87a] summarize the motivation and the properties of **curve relations** and the **RULI chain code**. They contain the formal definitions and the formal proof of the *length reduction property.*

Noise can have two major effects on curves in digital images: it creates a lot of new short curve segments and it cuts long curves into smaller pieces. Using the length reduction property of the 2 × 2/2 curve pyramid, the short noisy curve segments are eliminated in a top-down process that deletes all curve relations in the lower levels that are not represented in the level above [Kro87b].

Experiments with the synthetic image from [Kro86c] are reported. The only parameter of this '**structural noise filtering process**' is the level up to which the pyramid is built bottom-up and which is also the starting level for the top-down deletion. In the experiments this parameter varied from 1 to 10. The results are compared to the uncorrupted image and show that the number of deleted noisy curve segments increase by a factor of two on the average when one more level is used.

11.2 Hartmann's hierarchical structure code

Hartmann's pyramid is built on a hexagonal grid (see chapter 5). In the early version the cells contained a hierarchical contour code HCC describing a contour segment that has been detected by a Laplacian operator [Har84c]. The cells' contents has been extended to hold also regions in the HSC (hierarchical structure code [Har86b]).

One HSC element in the actual version [Har87a] is described by

$$< t; m; \varphi | k; n >,$$

where

t is one of 7 types:

'e' edge,

'b' bright line,

'd' dark line,

'h' small bright region,

'h***'** nose-shaped bright region,

'l' small dark region,

'l***'** nose-shaped dark region;

m identifies one of 42 detector shapes;

φ describes one of the 6 possible detector orientations;

k is the detector size; and

n is the reduction level to which the detector was applied.

Hartmann's pyramid has been used for several years and many applications. To name one example, [DH86] shows similar strategies for model-based recognition than Burt's pattern tree.

11.3 The multiresolution intensity axis of symmetry

Gauch et al [GOP], [GP88] use the *symmetric axis transformation*(SAT) according to Blum (see also [Ros86a]) for figure based shape descriptions.

- Advantages of SAT:
  - branching structure of object = branching structure of axis
  - bending and flaring of object ↔ changes in curvature and radii of axis
  - it is unique: it allows reconstruction.
- Drawbacks of SAT: it is sensitive to noise and to small detail;

The authors' solution is a multiresolution symmetric axis.

It is based on following ideas: Lower resolution yields simplification. As does also the branching structure of the SAT. The importance of a branch is determined by its annihilation resolution (that is the level above which it does not appear any more). This definition of importance implies a hierarchical ordering of the SAT-branches. Note the analogy to the approach taken in [FK89], [FK94] (chapter 10.7).

There are two possibilities to reduce resolution with advantages and drawbacks:

Reduce resolution by	advantage	drawback
1D boundary blurring	preserves topology	does not preserve figural similiarity
2D figure blurring	good figural similarity	changes topology

The boundary curvature reflects the bending of the object: see CODONs [RH85] , and the CODON duals [Ley87] (e.g. parts on the SAT corresponding to CODON).

The authors' strategy "labels each CODON with a measure of its importance." The CODON generalization is achieved by taking the annihilation level of resolution as importance. That creates a hierarchy of CODONs.

Through Leyton's duality between SAT ↔ CODONs, a correspondence between SAT- and CODON-hierarchies can be established:

$\boxed{\text{scale of annihilation of SAT-axis branch} = \text{scale of annihilation of CODON}}$

A symmetric axis pile is defined as a collection of level curves for every possible (gray-) level. Adding also the scale creates the *multiresolution symmetric axis pile.*

At critical points the symmetric axis sheet changes abruptly:

interpretation	Critical points
sheet terminations	local extreme spots
loop terminations	local extreme rings
axis tears	saddle

In the experiments, 2D Gaussian intensity blurring was used to generate the pyramid. Following applications are reported:

- Segmentation of binary images.
- Segmentation of gray scale images into a set of objects $R(x, y)$ such that the image $I(x, y) = \int \int R(x, y) dxdy$.

11.4 Further approaches

Huertas and Medioni [HM86] have a goal similar to our structural noise filtering:

1. Linear decomposition of Laplacian-of-Gaussians (LoG) → zero crossings;
2. zero crossings → edges + orientation
3. At edges: Facet-Model-Interpolation → refined contours.

The authors mention a possible resolution reduction for 'global' or 'macro' edges.

Kjell and Dyer [KD86] determine "long straight edge segments" at orientations of $45^0 i$. A $4 \times 4/4$ pyramid is used for the hierarchical region growing procedures:
SEGMENT → (LINK (bottom up linking), LABEL (top down label propagation)),
SMOOTH → (BUILD (bottom up weighted average), PROJECT (top down weighted average)).
They conclude: 'Features based on spatial properties of long, straight, extended edge segments are reliable features for texture description and segmentation.'

Meer, Sher, and Rosenfeld [MSR88] present the **chain pyramid** as a solution to the curve representation problem in pyramids. Curves are represented by doubly lined lists in the cells of a $4 \times 4/4$ pyramid. A probabilistic allocation algorithm avoids the overload in the higher levels. Algorithms for preserving the local connectivity during the bottom-up building process (reduction function), for smoothing of multiscale curves, and for gap bridging between contour fragments are presented. No consideration is given to the monotonic curve property e.g. if curvature extrema are preserved during reduction or not.

An alternative to the RULI-chain code is presented in [MS86]. A multiple grid chain code (MG-code) is defined on 4 grids. It has better coding efficiency than Freeman's chain code.

12 Dual Pyramids

In [Kro86c] extensive experiments have been performed with a synthetic image containing a large bright square, three large characters and several long lines. One version of the image has been corrupted by strong additive noise. The results of the experiments are compared with the uncorrupted version.

The gray value pyramid built with a Gaussian reduction function destroys the thin lines after a few steps, while large homogeneous regions remain visible although their boundaries are smoothed. Detecting contours at higher levels of the gray value pyramid produces many parallel contours at the higher levels. However, detecting contours at the high resolution and building the curve pyramid keeps the contours thin. Noise introduces a lot of short curve segments. The higher levels of the curve pyramid show much less noise because short curve segments disappear after a few reduction steps (compare chapter 11).

Hence the idea to combine the capabilities of Gaussian pyramid in detecting homogeneous compact regions with the advantages of the curve pyramid that can efficiently recognize long curves. The requirement to allow full information flow between both pyramids, not only bottom-up and top-down in each pyramid but also between all corresponding levels of both pyramids, led to the following dual structures.

12.1 Dual grids and pyramid structures

Two dual classes of square grids are related by local (square) operations: a side of the window may have an odd or an even number of pixels. The center of an odd sided window (e.g. 3×3, Fig. 15) is located at the same position as the center pixel of the window. Hence the same square grid can be used to store the result.

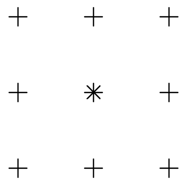


Figure 15: Geometric position (”×”) of a 3 × 3 local operation (”+”).

The centers of a local square operation with an **even** sided window (e.g. 2 × 2, Fig. 16) form a new grid which is the **dual** of the original.

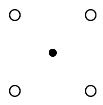


Figure 16: Geometric position (”•”) of a 2 × 2 local operation (”◦”).

Spacing and orientation of the new grid is not affected by any local operation that is applied at every pixel location in the input image.

When an even sided local operation is applied to the dual grid the result is located on the original grid. As a matter of fact, two successive even sided local operations can be performed in one step by one equivalent odd sided local operation which, as discussed above, is defined on the same grid.

That means that two grids are sufficient to position the result of any sequence of local square operation correctly with respect to the refered image plane. The two derived grids complete each other in a dual way. Every grid position in one grid is the center of a square formed by the positions of the corresponding four nearest pixels in the other grid. When the points in each grid are connected by edges, they form graphs which are **dual** to each other. Points correspond to squares and horizontal edges to vertical edges and vice-versa.

Figure 17 shows the projection of a part of the two dual grids into the image plane.

This duality is the basis for our two complementary pyramids. We use both types of windows for reducing the pyramid levels. The resulting pyramids are called ”odd” and ”even” according to the type of the reduction window.

Let us define following grid transformations: H transforms a level $PO(k)$ of the odd pyramid into the corresponding level $PE(k)$ of the even pyramid. S transforms a grid $PE(k)$ of the even pyramid into the coarser grid $PE(k+1)$ of level $k+1$. R reduces $PO(k)$ to $PO(k+1)$ in the odd pyramid similar to S. Figure 18 shows a diagram of the described situation.

If we require the diagram in Figure 18 to be commutative, the composite transformations

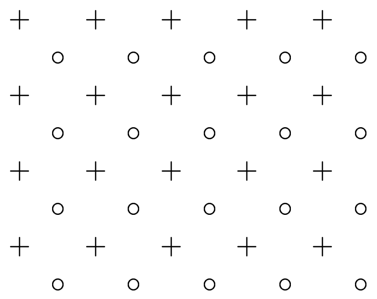


Figure 17: The dual grids: "+" and "o".

$$\begin{tikzcd} \boxed{PO(k)} \arrow[r, "H"] \arrow[d, "R"] & \boxed{PE(k)} \arrow[d, "S"] \\ \boxed{PO(k+1)} \arrow[r, "H"] & \boxed{PE(k+1)} \end{tikzcd}$$

Figure 18: Grid transformations between two successive levels.

must satisfy

$$H \circ S = R \circ H. \tag{36}$$

With this constraint the transformation R can be derived from H and S. A combined projection of grids $PO(k), PE(k), PO(k+1)$, and $PE(k+1)$ is presented in Figure 19. The structure dual to the $2 \times 2/2$ pyramid with the smallest reduction window is a $3 \times 3/2$ pyramid (Fig. 20).

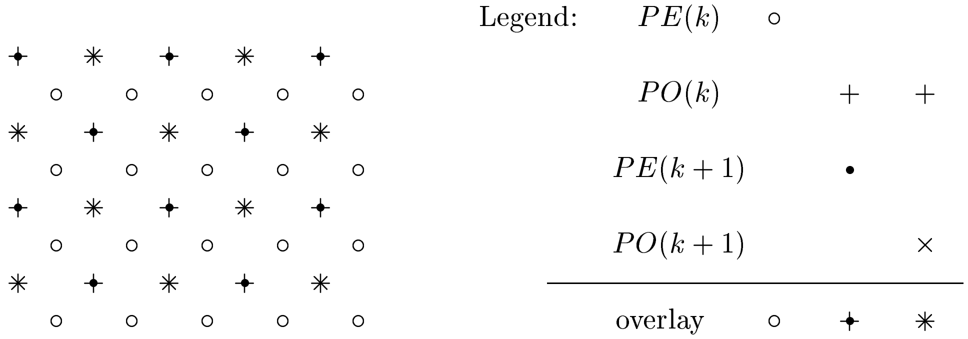


Figure 19: Overlaid projection of levels k and $k+1$ of odd (PO) and even (PE) pyramids.

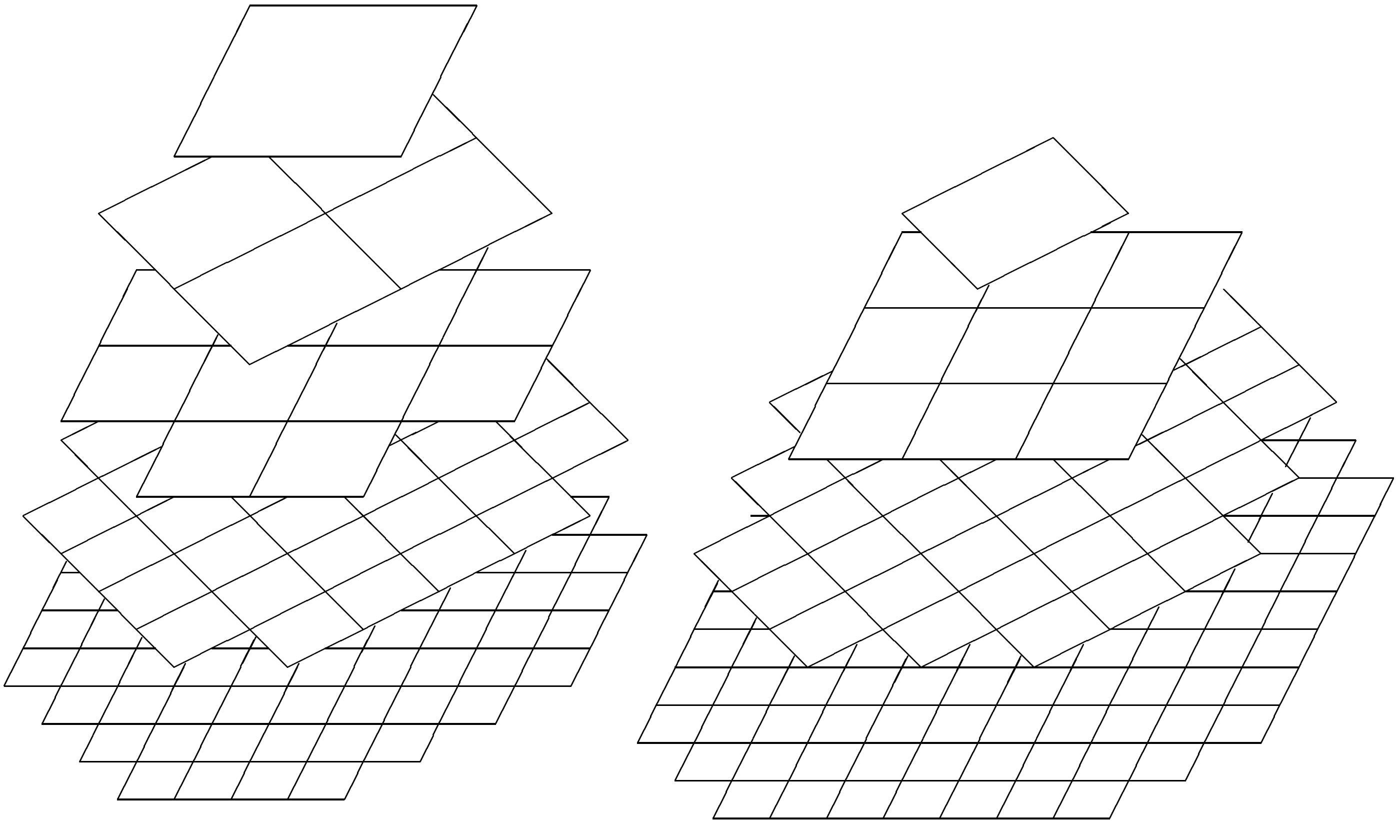


Figure 20: Dual 2 × 2/2 and 3 × 3/2 pyramids.

This concept of dual pyramids was presented in the paper [Kro86a] for the first time. The geometric correspondences and the indexing functions are calculated in this paper. Besides the geometric duality of the two pyramids also the contents of the cells are chosen complementary to each other: the *even* 2 × 2/2 pyramid stores curve relations and the *odd* 3 × 3/2 pyramid stores gray values. First examples of the information flow in and between the two pyramids are given in this paper. Further experiments are summarized in the following section.

12.2 Cooperation between dual pyramids

This section summarizes the principle idea and the experiences made with computer simulations. Following four papers and a Diploma thesis describe the experiments and the results: [Kro86d], [Kro86c], [Kro88a], [PK88], [Paa87] (Diploma thesis, in German).

Thin structures in gray level images must be detected at a high resolution. Local (2 × 2) contour operators generate the base of the dual curve pyramid. Short curve segments created by noise can be eliminated in the curve pyramid by the approach described in [Kro87b]. The remaining (long) contour segments partition the 3 × 3 local reduction window of the dual intensity pyramid into one or more connected regions (Fig. 21). Only pixels that are connected with the center pixel are fed into the reduction function. This smooths the noise in homogeneous regions and preserves at the same time the contrast along the boundaries.

This was the first of two useful applications emphasized in [Kro86d]: contour preserving gray level reduction. The second application, gray value based contour refinement, is still under investigation.

Gerhard Paar's [Paa87] thesis reports on computer simulations of the dual pyramids. The aim was to study properties of objects in the pyramids while different components for pyramid building are used. The main goal was to preserve sharp object contours.

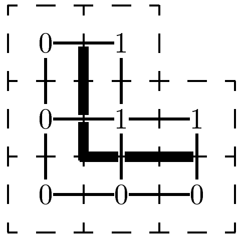


Figure 21: Overlaid dual gray values and curve relations.

Following components of the dual concept are compared:

Digital filters: Gauss, Frost [FSSH82], Lee [Lee81] (see also chapter 7), Median [RK82, Vol.1, section 6.4.4], KNN (K Nearest Neighbors)[DR78], and SNN (Symmetric Nearest Neighbor) [Har87b].

Contour operators: MAX (called B2(4) in [Kro86d]), THR thresholds the magnitude of the edges, LOK thresholds the gray levels.

Curve reduction and verification: as described in [Kro87b] and chapter 11.

Determination of connected center region: curves of the curve pyramid segment the local reduction window of the dual intensity pyramid (e.g. Fig. 21). The connected center region is a subset of the pixels in the reduction window that is not divided by any curve.

Selective reduction: Only pixels of the connected center region are fed into the digital filter used for reduction.

26 different combinations of the above components have been applied to both synthetic and real images. The results are evaluated optically and by objective methods like histograms, mean and standard deviation of selected regions, and radial histograms (circular projections). The experiences are summarized in [Kro88c] and [PK88].

In [PK88] we highlight in addition the theoreticel background of the dual concept. It combines in the pyramidal framework both **numerical** and **symbolic** image representations and allows transitions between the different representation at different resolutions. The $2\times2/2$ curve pyramid represents the multiscale symbolic scheme and the $3 \times 3/2$ intensity pyramid is used for numerical computations.

13 Conclusion

Classical pyramids use constant reduction factors of four between the levels of the pyramid. That means that every lower resolution has only 1/4 of the pixels. But also other multiresolution structures with lower reduction factors are possible. They have the advantage that the resolution is reduced gradually. Objects of sizes that fall inbetween two successive levels of the classical pyramid (e.g. too large for the higher resolution but too small for the lower resolution) may find the appropriate level in a $n \times n/2$ pyramid.

Dual representations enhance the description capabilities by combining numerical data and symbolic data in the concept of dual pyramids.The usage of symbolic representations within a pyramidal framework is relatively new. The symbols used are still very simple image primitives like curve segments. But since image interpretation has the objective to describe the image in terms of a high level vocabulary, symbolic pyramid representations are a step in the direction of high level conceptual hierarchies as proposed by R. Bajcsy [BR80]. The combination of such rich representations with the robustness of irregular pyramids let us expect to come closer to our goal of having methods that produce reliable image analysis results in reasonable time.

Index

This list of citations refers to the page numbers in the text.

References

[AAB$^+$84] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing. *RCA Engineer*, Vol. 29-6, Nov./Dec. 1984.

[AAS85] Narendra Ahuja, Byong An, and Bruce Schachter. Image representation using Voronoi tesselation. *Computer Vision, Graphics, and Image Processing*, Vol. 29:pp.286–295, 1985.

[AB86] H. Asada and M. Brady. The curvature primal sketch. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(No.1):pp.2–14, January 1986.

[ABv85] C. H. Anderson, P. J. Burt, and G. S. van der Wal. Change detection and tracking using pyramid transform techniques. *Intelligent Robots and Computer Vision*, SPIE Vol.579:pp.72–78, Sept.16-20 1985.

[ADKP87] K. Abrahamson, N. Dadoun, D.G. Kirkpatrick, and T. Przytycka. A simple parallel tree contraction algorithm. Technical Report TR–87–30, Dept. of CS, Univ. of British Columbia, Vancouver, August 1987.

[AF86] Nicholas Ayache and Olivier D. Faugeras. HYPER: A new approach for the recognition and positioning of two-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(No. 1):pp.44–54, January 1986.

[Ahu83] Narendra Ahuja. On approaches to polygonal decomposition for hierarchical representation. *Computer Vision, Graphics, and Image Processing*, Vol. 24(No. 2):pp.200–214, Nov. 1983.

[AK88a] K. M. Andress and Avinash C. Kak. Evidence accumulation and flow of control in a hierarchical spatial reasoning system. *The AI Magazine*, Vol.9(No.2):pp.75–94, 1988.

[AK88b] K. M. Andress and Avinash C. Kak. The pseiko report - version 2. Technical Report TR-EE 88-9, SCHOOLof Electrical Engineering, Purdue University, 1988.

[Avi88] Zlotnick Aviad. Locating corners in noisy curves by delineating imperfect sequences. Technical Report CMU–CS–88–199, Dept. of CS, Carnegie-Mellon University, December 1988.

[BA83a] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, Vol. COM-31(No.4):pp.532–540, April 1983.

[BA83b] P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, Vol. 2(No.4):pp.217–236, October 1983.

[Bal86] Dana H. Ballard. Interpolation coding: A representation for numbers in neural models. Technical Report TR–175, Dept. of CS, Univ. of Rochester, Sept. 1986.

[BASv86] Peter J. Burt, C. H. Anderson, J. O. Sinniger, and G. van der Wal. A pipelined pyramid machine. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 133–152. Springer-Verlag Berlin, Heidelberg, 1986.

[BC88] Walter F. Bischof and Terri Caelli. Parsing scale-space and spatial stability analysis. *Computer Vision, Graphics, and Image Processing*, Vol. 42(No. 2):pp.192–205, May 1988.

[BDHJ83] S. B. M. Bell, B. M. Diaz, F. Holroyd, and M. J. Jackson. Spatially referenced methods of processing raster and vector data. *Image and Vision Computing*, Vol. 1(No. 4):pp.211–220, Nov. 1983.

[Ber87] Fredrik Bergholm. Edge focusing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(No. 6):pp.726–741, November 1987.

[Bes86] Ph. W. Besslich. Pyramidal transforms in image processing and computer vision. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 215–246. Springer-Verlag Berlin, Heidelberg, 1986.

[BF87] Giuseppe Bevacqua and Ruhal Floris. A surface specific-line tracking and slope recognition algorithm. *Computer Vision, Graphics, and Image Processing*, Vol. 40(No. 2):pp.219–227, November 1987.

[BHR81] P. J. Burt, T.-H. Hong, and Azriel Rosenfeld. Segmentation and estimation of image region properties through cooperative hierarchical computation. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-11(No.12):pp.802–809, December 1981.

[BHR86] J. B. Burns, A. R. Hanson, and E. M. Riseman. Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(No.4):pp.425–455, July 1986.

[Bie87] Irving Biederman. Matching image edges to object memory. In *Proceedings of the First International Conference on Computer Vision*, pages 384–392, London, England, 1987.

[BPD88] S.N. Biswas, S.K. Pal, and D. Dutta Majumder. Binary contour coding using bézier approximation. *Pattern Recognition Letters*, Vol. 8(No. 4):pp. 237–249, November 1988.

[BR80] R. Bajcsy and D. A. Rosenthal. Visual and conceptual focus of attention. In S. Tanimoto and A. Klinger, editors, *Structured Computer Vision*, pages 133–149. Academic Press, 1980.

[BR85] Ernest S. Baugher and Azriel Rosenfeld. Boundary localization in an image pyramid. Technical Report TR-1488, University of Maryland, Computer Science Center, May 1985.

[BR86] Ernest S. Baugher and Azriel Rosenfeld. Boundary localization in an image pyramid. *Pattern Recognition*, Vol. 19:pp.373–395, May 1986.

[Bre65] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, Vol. 25(No.1):pp. 25–30, 1965.

[Bre85] J. E. Bresenham. Run length slice algorithm for incremental lines. In *Fundamental Algorithms for Computer Graphics*, NATO ASI Series, pages 59–104. Springer Verlag, New York/Berlin, 1985.

[BS88] Michael F. Barnsley and Alan D. Sloan. A better way to compress images. *Byte*, Vol. 13(No. 1):pp.215–223, January 1988.

[BT86] Ronald P. Blanford and Steven L. Tanimoto. Bright-spot detection in pyramids. In *Proc. Eighth International Conference on Pattern Recognition*, pages 1280–1282, Paris, France, October 1986. IEEE Comp.Soc.

[BT88] Ronald P. Blanford and Steven L. Tanimoto. Bright spot detection in pyramids. *Computer Vision, Graphics, and Image Processing*, Vol. 43(No. 2):pp.133–149, August 1988.

[Bur81] P. J. Burt. Fast filter transforms for image processing. *Computer Graphics and Image Processing*, Vol. 16:pp.20–51, 1981.

[Bur88] Peter J. Burt. Attention mechanisms for vision in a dynamic world. In *Proc. 9th International Conference on Pattern Recognition*, pages 977–987, Rome, Italy, November 1988. IEEE Comp.Soc.

[BWBD86] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda. Uniqueness of the Gaussian kernel for scale-space filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8(No.1):pp.26–33, January 1986.

[Can86] V. Cantoni. I.P. hierarchical systems: Architectural features. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 21–40. Springer-Verlag Berlin, Heidelberg, 1986.

[Cas86] S. Castan. Architectural comparisons. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 91–108. Springer-Verlag Berlin, Heidelberg, 1986.

[CH88] Fang-Hsuan Cheng and Wen-Hsing Hsu. Parallel algorithm for corner finding on digital curves. *Pattern Recognition Letters*, 8(No. 1):pp.47–53, July 1988.

[CP84] J. L. Crowley and A. Parker. A representation of shape based on peaks and ridges in the difference of low–pass transform. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI–6:pp.156–170, 1984.

[CP85] C. M. A. Castle and M. L. V. Pitteway. An application of Euclid's algorithm to drawing straight lines. In *Fundamental Algorithms for Computer Graphics*, NATO ASI Series, pages 135–139. Springer Verlag, New York/Berlin, 1985.

[CS84] J. L. Crowley and R. M. Stern. Fast computation of the difference of low–pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI–6:pp.212–222, 1984.

[Dav79] Larry S. Davis. Hierarchical generalized Hough transforms and line-segment based generalized Hough transforms. Technical report, University of Texas, Austin, Laboratory for Image and Signal Analysis, 1979.

[Den86] J. Dengler. Local motion estimation with the dynamic pyramid. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 289–298. Springer-Verlag Berlin, Heidelberg, 1986.

[DG86] S. R. Dunham and F. H. Glanz. Optimum uniform piecewise linear approximation of planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(No.1):pp.55–66, January 1986.

[DH86] S. Drüe and G. Hartmann. Modellgestützte Erkennung hierarchisch codierter Objekte. In G. Hartmann, editor, *Mustererkennung 1986*, Informatik Fachberichte 125, pages 245–249. Springer Verlag, 1986.

[Di 86] V. Di Gesu'. A high level language for pyramidal architectures. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 329–340. Springer-Verlag Berlin, Heidelberg, 1986.

[DMS86] J. Dengler, H. P. Meinzer, and M. Schmidt. Lokale Bewegungsanalyse mit der Dynamischen Pyramide. In G. Hartmann, editor, *Mustererkennung 1986*, Informatik Fachberichte 125, pages 276–281. Springer Verlag, 1986.

[DP88] Leila De Floriani and Enrico Puppo. Constrained delaunay triangulation for multiresolution surface description. In *Proc. 9th International Conference on Pattern Recognition*, pages 566–569, Rome, Italy, November 1988. IEEE Comp.Soc.

[DR78] Larry S. Davis and Azriel Rosenfeld. Noise cleaning by iterated local averaging. *Trans. Syst. Man Cybern.*, SMC–8:pp.705–710, 1978.

[DS86] L. Dorst and A. W. M. Smeulders. Best linear unbiased estimators for properties of digitized straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(No.2):pp.276–282, March 1986.

[Duf86] M. J. B. Duff. Pyramids - expected performance. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 59–74. Springer-Verlag Berlin, Heidelberg, 1986.

[Ede87] Shimon Edelman. Line connectivity algorithms for an asynchronous pyramid computer. *Computer Vision, Graphics, and Image Processing*, Vol. 40(No. 2):pp.169–187, November 1987.

[EF86a] Heinrich Ebner and Dieter Fritsch. High fidelity digital elevation models - elements of land information systems. In *XVIII. FIG Congress*, Toronto, Canada, June 1986.

[EF86b] Heinrich Ebner and Dieter Fritsch. The multigrid method and its application in photogrammetry. *International Archives of Photogrammetry and Remote Sensing*, Vol. 26, Part 3/3, 1986.

[FB86] M. A. Fischler and R. C. Bolles. Perceptual organization and curve partitioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(No.1):pp.100–105, January 1986.

[Fer86] M. Ferreti. Overlapping in compact pyramids. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 247–260. Springer-Verlag Berlin, Heidelberg, 1986.

[Fer89] Cornelia Fermüller. Hierarchisches Vergleichen von Konturen. Master's thesis, Technische Universität Graz, 1989. Diplomarbeit.

[FK89] Cornelia Fermüller and Walter G. Kropatsch. Hierarchische Kontur-Beschreibung durch Krümmung. In Axel Pinz, editor, *Wissensbasierte Mustererkennung*, OCG-Schriftenreihe, Österr. Arbeitsgemeinschaft für Mustererkennung, pages 171–187. Oldenbourg, 1989. Band 49.

[FK94] Cornelia Fermüller and Walter G. Kropatsch. A Syntactic Approach to Scale-Space-Based Corner Description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16(No. 7):pp. 748–751, July 1994.

[FMI83] K. Fukushima, S. Miyake, and T. Ito. *Neocognitron*: a neural network model for a mechanism of visual pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-13(Nb. 3):pp.826–834, September/October 1983.

[Fou88] T.J. Fountain. Array architectures for iconic and symbolic image processing. *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 2(No. 3):pp. 407–424, June 1988.

[FP87] M. Feuchtwanger and T. K. Poiker. The surface patchwork, an intelligent approach to terrain modeling. In *Proceedings of the 5th NorthWest Conference on Surveying and Mapping*, Whistler, B. C., June 1987.

[Fre61] H. Freeman. On the encoding of arbitrary geometric configurations. *IEEE Trans. Electron. Comput.*, EC-10:pp.260–268, 1961.

[Fre74] H. Freeman. Computer processing of line-drawing images. *Computing Surveys*, Vol. 6(No.1):pp.57–97, March 1974.

[Fre77] H. Freeman. A corner finding algorithm for chain coded curves. *IEEE Tr. on Computers*, Vol.C-23(No 3):pp.297–304, March 1977.

[Fri86] G. Fritsch. General purpose pyramidal architectures. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 41–58. Springer-Verlag Berlin, Heidelberg, 1986.

[FSSH82] V. S. Frost, J. A. Stiles, K. S. Shanmugan, and J. C. Holtzman. A model for radar images and its application to adaptive filtering of multiplicative noise. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI–4:pp.157–165, 1982.

[GB88] WeiXin Gong and Gilles Bertrand. A fast skeletonization algorithm using derived grids. In *Proc. 9th International Conference on Pattern Recognition*, pages 776–778, Rome, Italy, November 1988. IEEE Comp.Soc.

[Ger87] Peter R. Gerke. *Wie denkt der Mensch? - Informationstechnik und Gehirn.* Springer Verlag, New York, Heidelberg, Berlin, 1987.

[GHR85] A. D. Gross, R. L. Hartley, and Azriel Rosenfeld. A fast parallel algorithm for dot linking in glass patterns. *Pattern Recognition Letters*, Vol. 3:pp.263–270, 1985.

[GJ86] William I. Grosky and Ramesh Jain. A pyramid-based approach to segmentation applied to region matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(No.5):pp.639–650, September 1986.

[Gol86] Andrew D. Goldfinger. SAR as a tool for remote sensing. In *Photogrammetric and Remote Sensing Systems for Data Processing and Analysis*, pages 292–306. ISPRS, American Society for Photogrametry and Remote Sensing, Mai 1986. Vol.26, part.2.

[GOP] John M. Gauch, William R. Oliver, and Stephen M. Pizer. Multiresolution shape descriptions and their applications in medical imaging. , .

[Gos86] Ardeshir Goshtasby. Comments on 'scale-based description and recognition of planar curves and two-dimensional shapes'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(No.5):pp.674–675, September 1986.

[GP88] John M. Gauch and Stephen M. Pizer. Image description via the multiresolution intensity axis of symmetry. In *Proceedings of the Second International Conference on Computer Vision*, pages 269–274, Tampa, Florida, December 1988.

[Gro86] A. D. Gross. *Multiresolution Object Detection and Delineation.* PhD thesis, University of Maryland, Computer Science Center, January 1986.

[Gro87] A. D. Gross. Multiresolution object detection and delineation. *Computer Vision, Graphics, and Image Processing*, Vol. 39(No. 1):pp.102–115, July 1987.

[Har84a] Robert M. Haralick. Digital step edges from zero crossing of second directional derivative. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6:pp.58–68, 1984.

[Har84b] R. L. Hartley. *Multi–Scale Models in Image Analysis.* PhD thesis, University of Maryland, Computer Science Center, 1984.

[Har84c] G. Hartmann. Principles and strategies of hierarchical contour coding. In *Proceedings of the Seventh International Conference on Pattern Recognition*, pages 1087–1089, Montreal, Canada, 1984.

[Har85] R. L. Hartley. A Gaussian–weighted multiresolution edge detector. *Computer Vision, Graphics, and Image Processing*, Vol. 30:pp.70–83, May 1985.

[Har86a] Robert M. Haralick. Computer vision theory: The lack thereof. *Computer Vision, Graphics, and Image Processing*, Vol. 36(No. 2/3):pp.372–386, November/December 1986.

[Har86b] G. Hartmann. Unified description and recognition of continuous contours and regions. In *Proceedings of the Eighth International Conference on Pattern Recognition*, pages 1201–1203, Paris, France, October 1986.

[Har87a] G. Hartmann. Recognition of hierarchically encoded images by technical and biological systems. *Biological Cybernetics*, Vol. 57:pp.73–84, 1987.

[Har87b] D. Harwood. A new class of edge-preserving smoothing filters. *Pattern Recognition Letters*, Vol. 6(No. 3):pp.155–162, August 1987.

[HD86] S.-B. Ho and C. R. Dyer. Shape smoothing using medial axis properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(No.4):pp.512–520, July 1986.

[Her87] Harald Herbst. Die Auswirkungen verschiedener Reduktionsfunktionen auf Bildeigenschaften in Radar–Bildpyramiden. Master's thesis, Technische Universität Graz, 1987. Diplomarbeit.

[HLLZ87] Robert M. Haralick, Charlotte Lin, James S. J. Lee, and Xinhua Zhuang. Multiresolution morphology. In *Proceedings of the First International Conference on Computer Vision*, pages 516–520, London, England, June 1987.

[HLM86] Vincent Shang-Shouq Hwang, Davis S. Larry, and Takashi Matsuyama. Hypothesis integration in image understanding systems. *Computer Vision, Graphics, and Image Processing*, Vol. 36:pp.321–371, November 1986.

[HM86] Andres Huertas and Gerard Medioni. Detection of intensity changes with subpixel accuracy using Laplacian-Gaussian masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(No.5):pp.651–664, September 1986.

[Hon82] T.-H. Hong. *Pyramid Methods in Image Analysis.* PhD thesis, University of Maryland, Computer Science Center, 1982.

[HR84] T.-H. Hong and Azriel Rosenfeld. Compact region extraction using weighted pixel linking in a pyramids. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6(2):pp.222–229, 1984.

[HR85] R. L. Hartley and Azriel Rosenfeld. Hierarchical curve linking for corner detection. In S. Levialdi, editor, *Integrated Technology for Parallel Image Processing*, pages 101–119. Academic Press, London, 1985.

[HS85] Robert M. Haralick and Linda G. Shapiro. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, Vol. 29(No. 1):pp.100–132, January 1985.

[HSHR83] T.-H. Hong, M. Shneier, R. L. Hartley, and Azriel Rosenfeld. Using pyramids to detect good continuation. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-13(4):pp.631–635, 1983.

[HW86] B. K. P. Horn and E. J. Jr. Weldon. Filtering closed curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(No.5):pp.665–668, September 1986.

[HZLL88] Robert M. Haralick, Xinhua Zhuang, Charlotte Lin, and James Lee. The digital morphological sampling theorem. Technical report, Department of Electrical Engineering, University of Washington, Seattle, WA 98195, February 1988.

[Jae89] Bernd Jähne. *Digitale Bildverarbeitung.* Springer, Berlin, 1989.

[JB83] B. Julesz and J. R. Bergen. Textons, the fundamental elements in preattentive vision and perception of textures. *The Bell System Technical Journal*, Vol. 62(No. 6):pp.1619–1645, July-August 1983.

[JH89] Hyonam Joo and Robert M. Haralick. Application of mathematical morphology to machine vision. In H. Burkhardt, K.H. Höhne, and B. Neumann, editors, *Informatik Fachberichte 219: Mustererkennung 1989*, pages 1–27, Hamburg, 2.-4.10. 1989. 11.DAGM - Symposium, Springer Verlag.

[JM92] Jean-Michel Jolion and Annick Montanvert. The adaptive pyramid, a framework for 2D image analysis. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 55(3):pp.339–348, May 1992.

[JMP88] T. Jiang, M.B. Merickel, and E.A. Jr. Parrish. Automated threshold detection using a pyramid data structure. In *Proc. 9th International Conference on Pattern Recognition*, pages 689–692, Rome, Italy, November 1988. IEEE Comp.Soc.

[JR89] Jean-Michel Jolion and Azriel Rosenfeld. Coarse-fine bimodality analysis of circular histograms. *Pattern Recognition Letters*, Vol. 10(No. 3):pp. 201–207, September 1989.

[KD86] B. P. Kjell and C. R. Dyer. Segmentation of textured images by pyramid linking. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 273–288. Springer-Verlag Berlin, Heidelberg, 1986.

[Kel71] M. D. Kelly. Edge detection in pictures by computer using planning. In B. Meltzer and D. Michie, editors, *Machine Intelligence 6*, pages 397–409, Edinburgh, Scotland, 1971. Edinburgh University Press.

[KH88] Walter G. Kropatsch and H. Herbst. Effiziente Störungsreduktion in der Gauß-Pyramide. In Franz Pichler and Axel Pinz, editors, *Statistik und Mustererkennung*, OCG-Schriftenreihe, Österr. Arbeitsgemeinschaft für Mustererkennung, pages 22–47. Oldenbourg, 1988. Band 42.

[KHK88] A.C. Kak, S.A. Hutchinson, and Andress K.M. Planning and reasoning in sensor based robotics. Presented paper at the 10. DAGM-Symposium Mustererkennung, Zürich, 27.9.-29.9., 1988.

[KK87] R. Krishnaswamy and C. E. Kim. Digital parallelism, perpendicularity, and rectangles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(No. 2):pp.316–321, March 1987.

[KP87] Walter G. Kropatsch and G. Paar. Aufbau einer Pyramide auf Radarbildern. In G. Pernul and A Min Toja, editors, *Berichte aus Informatikforschungsinstitutionen*, volume Band 37 of *OCG–Schriftenreihe*, pages 221–232, Wien–München, 1987. R. Oldenburg.

[KPH88] Alan Kalvin, Shmuel Peleg, and Robert Hummel. Pyramid segmentation in 2D and 3D images using local optimization. In *Proc. 9th International Conference on Pattern Recognition*, pages 276–278, Rome, Italy, November 1988. IEEE Comp.Soc.

[KPK87] Ho Sung Kim, Kyo Ho Park, and Myunghwan Kim. Shape decomposition by collinearity. *Pattern Recognition Letters*, Vol. 6(No. 5):pp.335–340, December 1987.

[Kre65] E. Kreyszig. *Statistische Methoden und ihre Anwendungen.* Vandenhöck and Ruprecht, Göttingen, 1965.

[Kro85a] Walter G. Kropatsch. Hierarchical curve representation in a new pyramid scheme. Technical Report TR-1522, University of Maryland, Computer Science Center, June 1985.

[Kro85b] Walter G. Kropatsch. A pyramid that grows by powers of 2. *Pattern Recognition Letters*, Vol. 3:pp.315–322, 1985.

[Kro86a] Walter G. Kropatsch. Complementary pyramids. Presented paper at the NATO ARW in Maratea on 'Pyramidal Systems for Image Processing and Computer Vision', May 5-9, 1986.

[Kro86b] Walter G. Kropatsch. Curve representations in multiple resolutions. In *Proc. Eighth International Conference on Pattern Recognition*, pages 1283–1285. IEEE Comp.Soc., 1986.

[Kro86c] Walter G. Kropatsch. Ein Konzept mit zwei sich ergänzenden Pyramiden. In Walter G. Kropatsch and P. Mandl, editors, *Mustererkennung'86*, OCG–Schriftenreihe der österreichischen Arbeitsgemeinschaft für Mustererkennung, pages 158–171. R. Oldenburg, 1986. Band 36.

[Kro86d] Walter G. Kropatsch. Grauwert und Kurvenpyramide, das ideale Paar. In G. Hartmann, editor, *Mustererkennung 1986*, Informatik Fachberichte 125, pages 79–83. Springer Verlag, 1986.

[Kro86e] Walter G. Kropatsch. Kurvenrepräsentation in Pyramiden. In Walter G. Kropatsch and P. Mandl, editors, *Mustererkennung'86*, OCG-Schriftenreihe, Österr. Arbeitsgruppe für Mustererkennung, pages 16–51. Oldenbourg, 1986. Band 36.

[Kro87a] Walter G. Kropatsch. Curve Representations in Multiple Resolutions. *Pattern Recognition Letters*, Vol. 6(No. 3):pp.179–184, August 1987.

[Kro87b] Walter G. Kropatsch. Elimination von "kleinen" Kurvenstücken in der $2 \times 2/2$ Kurvenpyramide. In E. Paulus, editor, *Mustererkennung 1987*, Informatik Fachberichte 149, pages 156–160. Springer Verlag, 1987.

[Kro88a] Walter G. Kropatsch. Preserving contours in dual pyramids. In *Proc. 9th International Conference on Pattern Recognition*, pages 563–565, Rome, Italy, November 1988. IEEE Comp.Soc.

[Kro88b] Walter G. Kropatsch. Pyramid research for image analysis. In Dmitrij Csetverikov, editor, *The 2nd Hungarian Workshop on Image Analysis*, Studies 206, pages 115–118, Budapest, Hungary, June 1988. MTA, Computer and Automation Institute, Hungarian Academy of Sciences.

[Kro88c] Walter G. Kropatsch. Rezeptive Felder in Bildpyramiden. In H. Bunke, Olaf Kübler, and P. Stucki, editors, *Mustererkennung 1988*, Informatik Fachberichte 180, pages 333–339. Springer Verlag, 1988.

[Kro88d] Walter G. Kropatsch. Wechselwirkungen zwischen Visualisierung und Bildanalyse. In W. Barth, editor, *Visualisierungstechniken und Algorithmen*, pages 99–108. Springer Verlag, 1988. Informatik Fachberichte 182.

[KT89] Walter G. Kropatsch and H. Tockner. Detecting the straightness of digital curves in $\mathcal{O}(n)$ steps. *Computer Vision, Graphics, and Image Processing*, Vol. 45(No. 1):pp.1–21, January 1989.

[KWT87] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. In *Proceedings of the First International Conference on Computer Vision*, pages 259–268, London, England, June 1987.

[LC88] Jia-Guu Leu and Limin Chen. Polygonal approximation of 2D shapes through boundary merging. *Pattern Recognition Letters*, 7(No. 4):pp.231–238, April 1988.

[Lee81] J. S. Lee. Refined filtering of image noise using local statistics. *Computer Vision, Graphics, and Image Processing*, Vol. 15:pp.380–389, 1981.

[Lev86] S. Levialdi. Programming image processing machines. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 311–328. Springer-Verlag Berlin, Heidelberg, 1986.

[Lev88] Stefano Levialdi. Computer architectures for image analysis. In *Proc. 9th International Conference on Pattern Recognition*, pages 1148–1158, Rome, Italy, November 1988. IEEE Comp.Soc.

[Ley87] Michael Leyton. Symmetry-curvature duality. *Computer Vision, Graphics, and Image Processing*, Vol. 38(No. 3):pp.327–341, June 1987.

[LL88] Shu-Xiang Li and Murray H. Loew. Analysis and modeling of digitized straight-line segments. In *Proc. 9th International Conference on Pattern Recognition*, pages 294–296, Rome, Italy, November 1988. IEEE Comp.Soc.

[LLL86] H. Li, M. A. Lavin, and R. J. Le Master. Fast Hough transform: A hierarchical approach. *Computer Vision, Graphics, and Image Processing*, Vol. 36(Nos. 2/3):pp.139–161, November/December 1986.

[Mue89a] Uwe Müßigmann. Texturanalyse, Fractale und Scale Space Filtering. In H. Burkhardt, K.H. Höhne, and B. Neumann, editors, *Informatik Fachberichte 219: Mustererkennung 1989*, pages 60–67, Hamburg, 2.-4.10.1989 1989. 11.DAGM - Symposium, Springer Verlag.

[Mue89b] Uwe Müßigmann. Texture analysis, fractals and scale space filtering. In *Proceedings of the 6th Scandinavian Conference on Image Analysis*, pages 987–994, Oulu, Finland, June 1989.

[Mac87] Alan K. Mackworth. Update on computational vision: Shape representation, object recognition and constraint satisfaction. Technical Report TR-87-29, Univ. of British Columbia Vancouver, CS Dept., July 1987.

[Mal86] F. Maloberti. Silicon implementation of multiprocessor pyramid architecture. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 357–372. Springer-Verlag Berlin, Heidelberg, 1986.

[Mal89] Stephane G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI–11(No. 7):pp. 674–693, July 1989.

[Man83] Benoit B. Mandelbrot. *The fractal geometry of nature.* W. H. Freeman and Company, New York, 1983.

[Mar76] David Marr. Early processing of visual information. *Philosophical Transactions of the Royal Society London*, Ser. B(275):pp.483–524, 1976.

[Mar80] David Marr. Visual information processing: The structure and creation of visual representations. *Philosophical Transactions of the Royal Society London*, Ser. B(290):pp.199–218, 1980.

[Mar82] David Marr. *Vision.* Freeman, W.H., San Francisco, 1982.

[Mat88] Takashi Matsuyama. Expert systems for image processing - knowledge-based composition of image analysis processes. In *Proc. 9th International Conference on Pattern Recognition*, pages 125–133, Rome, Italy, November 1988. IEEE Comp.Soc.

[May89] Harald Mayer. Progressive Übertragung und komprimierte Speicherung von digitalen Bildern. Master's thesis, Technische Universität Graz, 1989. Diplomarbeit.

[MCM$^+$86] A. Merigot, P. Clermont, J. Mehat, F. Devos, and B. Zavidovique. A pyramidal system for image processing. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 109–124. Springer-Verlag Berlin, Heidelberg, 1986.

[Mee88] Peter Meer. Simulation of constant size multiresolution representations on image pyramids. *Pattern Recognition Letters*, Vol. 8(No. 4):pp. 229–236, November 1988.

[Mee89] Peter Meer. Stochastic image pyramids. *Computer Vision, Graphics, and Image Processing*, Vol. 45(No. 3):pp.269–294, March 1989.

[Mel86] Robert A. Melter. Tesselation graph characterization using rosettas. *Pattern Recognition Letters*, Vol.4:pp.79–85, 1986.

[MJBR88] Peter Meer, Song-Nian Jiang, Ernest S. Baugher, and Azriel Rosenfeld. Robustness of image pyramids under structural perturbations. *Computer Vision, Graphics, and Image Processing*, Vol. 44(No. 3):pp.307–331, December 1988.

[MK89a] Harald Mayer and Walter G. Kropatsch. Progressive Bildübertragung mit der $3 \times 3/2$ Pyramide. In H. Burkhardt, K.H. Höhne, and B. Neumann, editors, *Informatik Fachberichte 219: Mustererkennung 1989*, pages 160–167, Hamburg, 2.-4.10.1989 1989. 11.DAGM - Symposium, Springer Verlag.

[MK89b] Harald F. Mayer and Walter G. Kropatsch. Kompakte Bildkodierung mit der $3 \times 3/2$ Pyramide. In Axel Pinz, editor, *Wissensbasierte Mustererkennung*, OCG-Schriftenreihe, Österr. Arbeitsgemeinschaft für Mustererkennung, pages 195–210. Oldenbourg, 1989. Band 49.

[MM83] David M. McKeown and John McDermott. Toward expert systems for photo interpretation. In *Proc. of Trends and Applications*, pages 33–39. IEEE Comp.Soc., 1983.

[MM86a] F. Mokhtarian and A. Mackworth. Authors' reply. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(No.5):pp.675, September 1986.

[MM86b] F. Mokhtarian and A. Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(No.1):pp.34–43, January 1986.

[MM88] Alan K. Mackworth and Farzin Mokhtarian. The renormalized curvature scale space and the evolution properties of planar curves. In *Proceedings, CVPR'88*, pages 318–326, Ann Arbor, Michigan, June 1988. IEEE Comp.Soc.

[MMR89] Annick Montanvert, Peter Meer, and Azriel Rosenfeld. Hierarchical image analysis using irregular tesselations. Technical Report TR-2322, University of Maryland, Computer Science Center, September 1989.

[Moi80] J. G. Moik. *Digital Processing of Remotely Sensed Images.* NASA SP ;431. NASA, Washington D.C., 1980.

[Mok88] Farzin Mokhtarian. Fingerprint theorems for curvature and torsion zero-crossings. Technical Report 88-9, Dept. of CS, Univ. of British Columbia, April 1988.

[MR89a] Mareboyana Manohar and H. K. Ramapriyan. Connected component labeling of binary images on a mesh connected massively parallel processor. *Computer Vision, Graphics, and Image Processing*, Vol. 45(No. 2):pp.133–149, February 1989.

[MR89b] Robert A. Melter and Azriel Rosenfeld. New views of linearity and connectedness in digital geometry. *Pattern Recognition Letters*, Vol. 10(No. 1):pp. 9–16, July 1989.

[MS86] T. Minami and K. Shinohara. Encoding of line drawings with a multiple grid chain code. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(No.2):pp.269–275, March 1986.

[MSR88] Peter Meer, C. Allan Sher, and Azriel Rosenfeld. The chain pyramid: Hierarchical contour processing. Technical Report TR-2072, University of Maryland, Computer Science Center, July 1988.

[NDC86] Ch. F. Neveu, CH. R. Dyer, and R. T. Chin. Two dimensional object recognition using multiresolution models. *Computer Vision, Graphics, and Image Processing*, Vol. 34(Nb.1):pp.52–65, 1986.

[NM79] M. Nagao and Takashi Matsuyama. Edge preserving smoothing. *Computer Graphics and Image Processing*, Vol. 9:pp.394–407, 1979.

[NR88] Seiichiro Naito and Azriel Rosenfeld. Shape from random planar features. *Computer Vision, Graphics, and Image Processing*, Vol. 42(No. 3):pp.345–370, June 1988.

[O'G88a] Lawrence O'Gorman. An analysis of feature detectability from curvature estimation. In *Proceedings, CVPR' 88*, pages 235–240, Ann Arbor, MI, June 1988. Computer Society Conference on Computer Vision and Pattern Recognition, Computer Society Press.

[O'G88b] Lawrence O'Gorman. Curvilinear feature detection from curvature estimation. In *Proc. 9th International Conference on Pattern Recognition*, pages 1116–1119, Rome, Italy, November 1988. IEEE Comp.Soc.

[O'R81] J. O'Rourke. Automatic visual attention control via a dynamic data structure. In *Conference on Information Science*, pages 301–306. John Hopkins University, May 1981.

[OS87] L. O'Gorman and A. C. Sanderson. A comparison of methods and computation for multi–resolution low– and band pass tranforms for image processing. *Computer Vision, Graphics, and Image Processing*, Vol. 37(No. 3):pp.386–401, March 1987.

[Paa87] Gerhard Paar. Beibehaltung von Objekträndern in Bildpyramiden. Master's thesis, Technische Universität Graz, 1987. Diplomarbeit.

[Pao89] Yoh-Han Pao. *Adaptive Pattern Recognition and Neural Networks*. Addison-Wesley Publishing Company, Inc., Reading, MA, 1989.

[Par86] M. S. Parsons. Generating lines using quadgraph patterns. *Computer Graphics Forum*, Vol. 5:pp. 33–39, 1986.

[Pav84] T. Pavlidis. A hybrid vectorization algorithm. In *Proceedings of the Seventh International Conference on Pattern Recognition*, pages 490–492, Montreal, Canada, 1984.

[Pav86] Theo Pavlidis. A vectorizer and feature extractor for document recognition. *Computer Vision, Graphics and Image Processing*, Vol.35(No.1):pp. 111–127, 1986.

[PF86] Shmuel Peleg and Orna Federbush. Custom made pyramids. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 165–172. Springer-Verlag Berlin, Heidelberg, 1986.

[Pha86] S. Pham. Digital straight segments. *Computer Vision, Graphics and Image Processing*, Vol.36(No.1):pp. 10–30, 1986.

[Pha89] Binh Pham. Conic B-splines for curve fitting: A unifying approach. *Computer Vision, Graphics, and Image Processing*, Vol. 45(No. 1):pp. 117–125, January 1989.

[Pit82] M. L. V. Pitteway. Bresenham's algorithm with run-line coding shortcut. *Computer Journal*, Vol.25(No.1):pp. 114–115, 1982.

[PK88] G. Paar and Walter G. Kropatsch. Hierarchical cooperation between numerical and symbolic image representations. In R. Mohr, Theo Pavlidis, and A. Sanfeliu, editors, *Structural Pattern Analysis*, pages 113–130. World Scientific Publ. Co., 1988.

[PR87] Tsai-Yun Phillips and Azriel Rosenfeld. A method of curve partitioning using arc-chord distance. *Pattern Recognition Letters*, Vol. 5(No. 4):pp.285–288, April 1987.

[Ram75] Urs Ramer. Extraction of line structures from photographs of curved objects. *Computer Graphics and Image Processing*, Vol.4:pp.81–103, 1975.

[RH85] Whitman Richards and Donald D. Hoffman. CODON constraints on closed 2D shapes. *Computer Vision, Graphics and Image Proceesing*, Vol.31:pp.265–281, 1985.

[Ric88] Dianne E. Richardson. Database design considerations for rule-based map feature selection. *ITC Journal*, Vol. 1988-2:pp. 165–171, 1988.

[RK82] Azriel Rosenfeld and A. C. Kak. *Digital Picture Processing*, volume Vol. 1 and 2. Academic Press, New York, second edition, 1982.

[Ron85] Christian Ronse. A simple proof of Rosenfeld's characterization of digital straight line segments. *Pattern Recognition Letters*, Vol.3:pp.323–326, September 1985.

[Ros74] Azriel Rosenfeld. Digital straight line segments. *IEEE Transactions on Computers*, Vol. C-23(No.12):pp.1264–1269, December 1974.

[Ros81] Azriel Rosenfeld. Generalized cellular automata. In Morio Onoe, Kendall Jr. Preston, and Azriel Rosenfeld, editors, *Real-Time/Parallel Computing, Image Analysis*, pages 63–71. Plenum Press, New York, 1981.

[Ros84] Azriel Rosenfeld, editor. *Multiresolution Image Processing and Analysis.* Springer, Berlin, 1984.

[Ros85a] Azriel Rosenfeld. Arc colorings, partial path groups, and parallel graph contractions. Technical Report TR-1524, University of Maryland, Computer Science Center, July 1985.

[Ros85b] Azriel Rosenfeld. The prism machine: an alternative to the pyramid. *J. Parallel Distributed Computing*, Vol. 2:pp.404–411, 1985.

[Ros86a] Azriel Rosenfeld. Axial representations of shape. *Computer Vision, Graphics and Image Processing*, Vol. 33(No.2):pp.156–173, Feb. 1986.

[Ros86b] Azriel Rosenfeld. Some pyramid techniques for image segmentation. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 261–271. Springer-Verlag Berlin, Heidelberg, 1986.

[Ros87a] Azriel Rosenfeld. A note on shrinking and expanding operations in pyramids. *Pattern Recognition Letters*, Vol. 6(No. 4):pp.241–244, September 1987.

[Ros87b] Azriel Rosenfeld. Recognizing unexpected objects: A proposed approach. *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 1(No.1):pp.71–84, April 1987.

[RS87] Azriel Rosenfeld and Allen C. Sher. Direction-weighted line fitting to edge data. *Pattern Recognition Letters*, Vol. 5(No. 4):pp.289–292, April 1987.

[Sam84] Hanan J. Samet. The quadtree and related hierarchical data structures. *Computing Surveys 16*, pages pp.187–260, 1984.

[Sam85] Hanan Samet. Data structures for quadtree approximation and compression. *Communications of the ACM*, Vol. 28(No. 9):pp.973–993, September 1985.

[SB88] Finn Sjöberg and Fredrik Bergholm. Extraction of diffuse edges by edge focusing. *Pattern Recognition Letters*, Vol. 7(No. 3):pp.181–190, March 1988.

[SBT85] R. Shoucri, R. Benesch, and S. Thomas. Note on the determination of digital straight line from chain codes. *Computer Vision, Graphics, and Image Processing*, Vol. 29(No. 1):pp.133–139, January 1985.

[SC88] Jorge L.C. Sanz and Robert E. Cypher. Algorithms for massively parallel image processing architectures. In *Proc. 9th International Conference on Pattern Recognition*, pages 412–419, Rome, Italy, November 1988. IEEE Comp.Soc.

[Ser82] Jean Serra. *Image Analysis and Mathematical Morphology.* Academic Press Inc., New York, 1982.

[SH86] David H. Schaefer and Ping Ho. Counting on the GAM pyramid. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 125–132. Springer-Verlag Berlin, Heidelberg, 1986.

[She87] D. Sher. Advanced likelihood generators for boundary detection. Technical Report TR-197, Univ. of Rochester, CS Dept., January 1987.

[Shn81] M. Shneier. Two hierarchical linear feature representations: Edge pyramids and edge quadtrees. *Computer Graphics and Image Processing*, Vol. 17:pp.221–224, 1981.

[Shn82] M. Shneier. Extracting linear features from images using pyramids. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-12:pp.569–572, 1982.

[SL88] Linda G. Shapiro and Haiyuan Lu. The use of a relational pyramid representation for view classes in a CAD-to-vision system. In *Proc. 9th International Conference on Pattern Recognition*, pages 379–381, Rome, Italy, November 1988. IEEE Comp.Soc.

[Slu88] Andrzej Sluzek. Using moment invariants to recognize and locate partially occluded 2D objects. *Pattern Recognition Letters*, Vol. 7(No. 4):pp.253–257, April 1988.

[SR89] Allen C. Sher and Azriel Rosenfeld. A pyramid Hough transform on the connection machine. Technical Report TR-2182, University of Maryland, Computer Science Center, February 1989.

[SSW85] Hanan J. Samet, Clifford A. Shaffer, and Robert E. Webber. The segment quadtree: a linear quadtree-based representation for linear features. In *Proceedings of IEEE Comp.Soc. Conf. on Computer Vision and Pattern Recognition*, pages 385–389, San Francisco, CA, June 10-13 1985.

[ST85a] Hanan Samet and Markku Tamminen. Converting CSG trees to bintrees including time. Technical Report TR-1467, University of Maryland, Computer Science, January 1985.

[ST85b] Hanan J. Samet and Markku Tamminen. Bintrees, CSG trees, and time. In B. A. Barsky, editor, *SIGGRAPH'85 Conf. Proceedings*, pages 121–130, San Francisco, CA, July 1985.

[ST88] Hanan Samet and Markku Tamminen. Efficient component labeling of images of arbitrary dimension. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-10, 1988.

[Sto86] Quentin F. Stout. Hypercubes and pyramids. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 75–90. Springer-Verlag Berlin, Heidelberg, 1986.

[SW84] Hanan J. Samet and Robert E. Webber. Storing a collection of polygons using quadtrees. *ACM Transactions on Graphics*, Vol. 4:pp.182–222, 1984.

[Tan72] Steven L. Tanimoto. Programming techniques for hierarchical parallel image processors. In Kendall Jr. Preston and Leonard Uhr, editors, *Multicomputers and Image Processing - Algorithms and Programs*, pages 421–429. Academic Press, New York, 1972.

[Tan86] Steven L. Tanimoto. Paradigms for pyramid machine algorithms. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 173–194. Springer-Verlag Berlin, Heidelberg, 1986.

[Tan88] Steven L. Tanimoto. From pixels to predicates in pyramid machines. In Jean-Claude Simon, editor, *Proceedings of the COST-13 workshop 'From the Pixels to the Features'*. AFCET, Bonas, France, August 1988.

[TASM83] David Lee Tuomenoksa, George B. Adams III, Howard Jay Siegel, and O. Robert Mitchell. A parallel algorithm for contour extraction: Advantages and architectural implications. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 336–344, Washington D.C., June 1983.

[TK80] Steven L. Tanimoto and A. Klinger, editors. *Structured Computer Vision: Machine Perception through Hierarchical Computation Structures.* Academic Press, New York, 1980.

[Toc87] Helmut Tockner. Syntaktische Untersuchung und Erkennung von Geraden im RULI Chain Code. Master's thesis, Technische Universität Graz, 1987. Diplomarbeit.

[TR87] Staffan Truvé and Whiteman Richards. From Waltz to Winston (via the connection table). In *Proceedings of the First International Conference on Computer Vision*, pages 393–404, London, England, 1987.

[Tso87] John K. Tsotsos. A 'Complexity Level' Analysis of Vision. In *Proceedings of the First International Conference on Computer Vision*, pages 346–355, London, England, 1987.

[Uhr86] Leonard Uhr. Parallel, hierarchical software/hardware pyramid architectures. In Virginio Cantoni and Stefano Levialdi, editors, *Pyramidal Systems for Image Processing and Computer Vision*, volume F25 of *NATO ASI Series*, pages 1–20. Springer-Verlag Berlin, Heidelberg, 1986.

[US84] Leonard Uhr and L. Schmitt. The several steps from icon to symbol using structured cone/pyramids. In Azriel Rosenfeld, editor, *Multiresolution Image Processing and Analysis*, pages 86–100. Springer Verlag, Berlin, Heidelberg, New York, Tokyo, 1984.

[vS85] Gooitzen S. van der Wal and Joseph O. Sinniger. Real time pyramid transform architecture. In *Intelligent Robots and Computer Vision*, pages 300–305, 1985. SPIE Vol.579.

[Wat87] R. J. Watt. An outline of the primal sketch in human vision. *Pattern Recognition Letters*, Vol.5(Nb. 5):pp.139–150, February 1987.

[Wel86] W. Wells III. Efficient synthesis of Gaussian filters by cascaded uniform filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI–8(Nb. 2):pp.234–239, March 1986.

[Wha88] Stephen W. Wharton. Pips: a procedure for interactive pyramid segmentation. In *IGARSS'88 - Remote Sensing: Moving towards the 21st Century*, volume Vol.1, pages 197–200, Sept.12–16 1988.

[WLH85] Layne T. Watson, Thomas J. Laffey, and Robert M. Haralick. Topographic classification of digital image intensity surfaces using generalized splines and the discrete cosine transformation. *Computer Vision, Graphics, and Image Processing*, Vol. 29(No. 2):pp.143–167, February 1985.

[Woj87] Zbigniew Wojcik. Rough approximation of shapes in pattern recognition. *Computer Vision, Graphics, and Image Processing*, Vol. 40(No. 2):pp.228–249, November 1987.

[Woo87] Robert J. Woodham. Stable representation of shape. Technical Report TR-87-5, Univ. of British Columbia Vancouver, CS Dept., February 1987.

[WP86] Michael Werman and Shmuel Peleg. Halftoning as optimal quantization. In *Proceedings of the 8th (IEEE) International Conference on Pattern Recognition*, pages 1114–1116, Paris, France, October 1986.

[WP88] Michael Werman and Shmuel Peleg. Gray level requantization. *Computer Vision, Graphics, and Image Processing*, Vol. 43(No. 1):pp.81–87, July 1988.

[WR87] Xiaolin Wu and Jon G. Rokne. Double-step incremental generation of lines and circles. *Computer Vision, Graphics and Image Processing*, Vol.37(No.3):pp. 331–344, March 1987.

[YK89a] Toru Yamaguchi and Walter G. Kropatsch. Structural implications of the performance of a neural network. In Axel Pinz, editor, *Wissensbasierte Mustererkennung*, OCG-Schriftenreihe, Österr. Arbeitsgemeinschaft für Mustererkennung, pages 140–148. Oldenbourg, 1989. Band 49.

[YK89b] Toru Yamaguchi and Walter G. Kropatsch. A vision by neural network or by pyramid. In *Proceedings of the 6th Scandinavian Conference on Image Analysis*, pages 104–111, Oulu, Finland, June 1989.

[YP86] A. L. Yuille and T. A. Poggio. Scaling theorems for zero crossings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8(No.1):pp.15–25, January 1986.