Abstraction Pyramids on Discrete Representations

Walter G. Kropatsch[*]

Institute for Computer-Aided Automation
Pattern Recognition and Image Processing Group
Vienna Univ. of Technology- Austria

Abstract. We review multilevel hierarchies under two special aspects: their potential for abstraction and for storing discrete representations. Motivated by claims to 'bridge the representational gap between image and model features' and by the growing importance of topological properties we discuss several extensions to dual graph pyramids and to topological maps: structural simplification should preserve important topological properties and content abstraction could be guided by an external knowledge base.

1 Introduction

At a panel of the last International Workshop on Visual Form (IWVF4) Sven Dickinson asked the following question referring to several research issues in the past and also in the future: "How do we bridge the representational gap between image features and coarse model features?" He identifies the one-to-one correspondence between

- salient image features (pixels, edges, corners,...) and
- salient model features (generalized cylinders, polyhedrons, invariant models,...)

as **limiting assumption** that makes prototypical or generic object recognition impossible. He suggested to bridge and not to eliminate the representational gap, and to focus efforts on:

- region segmentation
- perceptual grouping
- image abstraction

Let us take these goals as a guideline to re-consider research efforts in the area of multiresolution discrete representations under the special viewpoint of abstraction and of representations that are discrete in nature.

Regions as aggregations of primitive pixels play an extremely important role in nearly every image analysis task. Their internal properties (color, texture,

---

shape, ...) help to identify them and their external relations (adjacency, inclusion, similarity of properties) are used to build groups of regions having a particular meaning in a more abstract context. The union of regions forming the group is again a region with both internal and external properties and relations.

A representational concept that supports processes working at multiple levels of abstraction with the possibility to access semantic knowledge from the external world are extremely complex. We would like to highlight a few approaches that may have the potential to be extended into a future complex vision system bridging the representational gap identified by Dickinson.

At the last DGCI Udupa [35] considered surfaces as basic descriptive elements for representing boundaries between volumetric regions in 3D. He identifies three 'important' properties of such surfaces:

1. connected (topology)
2. oriented (combinatorial maps)
3. closed (Jordan boundary)

We would like to address some of these issues in the context of gradually generalizing our discrete image data across levels where geometry dominates up to levels of the hierarchy where topological properties become important.

Based on experiences with multiresolution pyramids we present several conceptual extensions with the aim of stimulating further research and collaboration which is necessary to overcome the intrinsic complexity of the proposed system architecture by joint efforts leaving enough room for original contributions.

The paper is organized as follows: After considering the formal definition of abstraction (section 2) and the consequences for representations we review discrete representations including a 'natural' example of vision based on an irregular sampling (section 3). Image pyramids are the main focus of section 4 where we present the basic ideas and properties of dual graph pyramids and of multilevel topological maps. Abstraction in such multilevel structures can be done either by modifying the contents of a representational cell (section 5) or by 'simplifying' the structural arrangement of the cells while major topological properties are preserved (section 6). In this last section we present two simple 3D configurations which turned out to be hard to distinguish by current representations.

2   Visual Abstraction

By definition abstraction extracts essential features and properties while it neglects unnecessary details. Two types of unnecessary details can be distinguished:

- redundancies
- data of minor importance

Details may not be necessary in different contexts and under different objectives which reflect in different types of abstraction. In general, three different types of abstractionare distinguished:

isolating abstraction: important aspects of one or more objects are extracted from their original context.

generalizing abstraction: typical properties of a collection of objects are emphasized and summarized.

idealizing abstraction: data are classified into a (finite) set of ideal models, with parameters approximating the data and with (symbolic) names/notions determining their semantic meaning.

These three types of abstraction have strong associations with well known tasks in computer vision: recognition and object detection tries to **isolate** the object from the background; perceptual grouping needs a high degree of **generalization**; and classification assigns data to **'ideal'** classes disregarding noise and measurement inaccuracies.

In all three cases abstraction drops certain data items which are considered less relevant. Hence the **importance** of the data needs to be computed to decide which items to drop during abstraction. The importance or the relevance of an entity of a (discrete) description must be evaluated with respect to the purpose or the goal of processing. The system may also change its focus according to changing goals after knowing certain facts about the actual environment, other aspects that were not relevant at the first glance may gain importance. Representational schemes must be flexible enough to accommodate such attentional shifts in the objectives.

3 Discrete Representations

A discrete representation is associated with a countable variable which can be mapped into $\mathbb{Z}^n$ [12]. A digital image is the result of sampling a continuous image at discrete locations **the sampling points**. Usually this is a finite subset of 'pixels' of the discrete grid $\mathbb{Z}^2$. This discretization process maps any object of the continuous image into a discrete version if it is sufficiently large to be captured by the sensors at the sampling points. Resolution relates the unit distance of the sampling grid with a distance in reality. There exist different concepts to model the conversion between continuous and discrete representations. Recently Brimkov etal [5] have introduced new schemes for object discretizations in higher dimensions: k-neighbors in $\mathbb{Z}^n$, minimal cover and super-cover. Besides their relevance for visualization purposes such concepts allow also the continuous interpretation of discrete measurements. The properties of the continuous object, i.e. color, texture, shape, as well as its relations to other (nearby) objects are mapped into the discrete space, too. The most primitive discrete representation assigns to each sampling point a measurement, be it a gray or color value from a finite set or a binary value.

Hence a digital image is a finite set of integer triples $(ix, iy, ig) \in \mathbb{Z}^3$. In order to express the connectivity or other geometric or topological properties this set must be enhanced by a neighborhood relation. In the regular square grid arrangement of sampling points it is implicitly encoded as 4- or 8-neighborhood with the well known problems in conjunction with Jordan's curve theorem. Note

that ALL the information about the image's content is stored at the sampling points!

The neighborhood of sampling points can be represented explicitly, too: in this case the sampling grid is represented by a graph consisting of vertices corresponding to the sampling points and of edges connecting neighboring vertices. Although this data structure consumes more memory space it has several advantages, among which we find the following:

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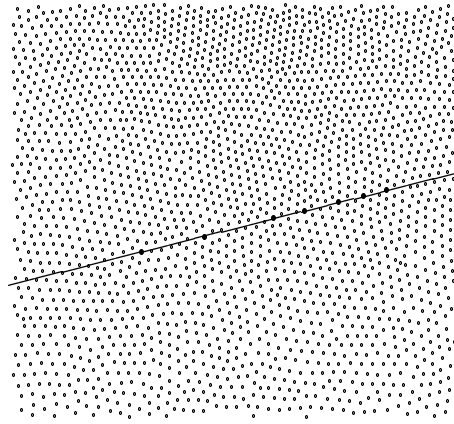- The edges may be determined either automatically or depending on the data.

Since sub-sampling of a discrete representation is also a discrete representation, the arrangement of a human retina can be considered as discrete, too. Fig. 1(a) shows a small portion of the sampling points of a monkey's retina, which is similar to the one of a human eye. In Fig. 1(b-f) we opposed a simple sampling concept both with the natural but irregular grid and an artificial but regular grid of similar resolution. Sampling points being within a certain distance to the line have been filled and the radius has been varied. Let this distance be determined by a circle intersecting the continuous line. Optically similar effects can be observed: with a small radius the sequence of bold points has many gaps (Fig. 1b,c,e), the line is not connected. Increasing the radius, the density of black points increases also and the gaps are closed. With a large radius the line becomes thick since also sampling points not directly along the line reach the line (Fig. 1d,f). These effects are typical for discrete lines and they are overcome in many different ways, but most approaches consider the regular grid. The problem arising with irregular grids is that there is no implicit neighbor definition! Usually Voronoi neighbors determine the neighborhood graph. It would be interesting whether concepts for discrete straight lines, discrete planes in 3D and hyper-planes in nD, discrete circles, spheres and hyper-spheres could be recovered also from irregular grids and if the involved computational processes are feasible.

The retina example demonstrated that the neighborhood in irregular grids needs to be represented explicitly. This creates a new representational entity: the binary relation of an edge in the neighborhood graph. Together with the fact that a 2D image is embedded in the continuous image plane the line segments connecting the end points of edges partition the image plane into connected faces which are part of the **dual graph**. In n dimensions $n+1$ basic entities are sufficient to describe a discrete configuration embedded in the space spanned by the n coordinate axes, e.g. the cells of abstract cellular complexes [24]. Let us shortly remind some of the most frequent names used by different authors:
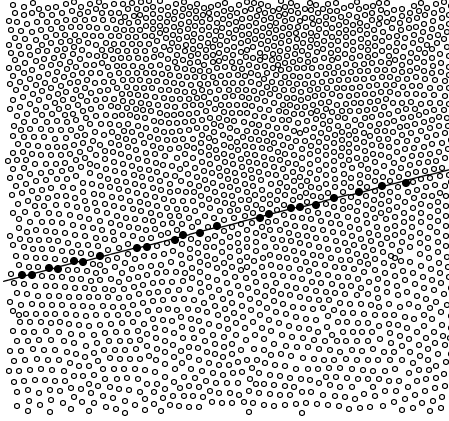
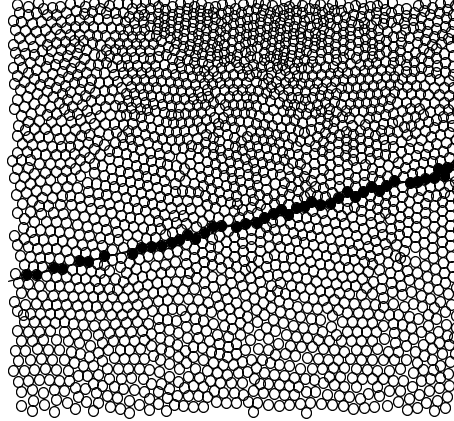| dimension	geometry	spel [35]	graph
0	point	pixel,pointel	vertex
1	line	linel	edge
2	face, region	surfel	face
3	volume	voxel	
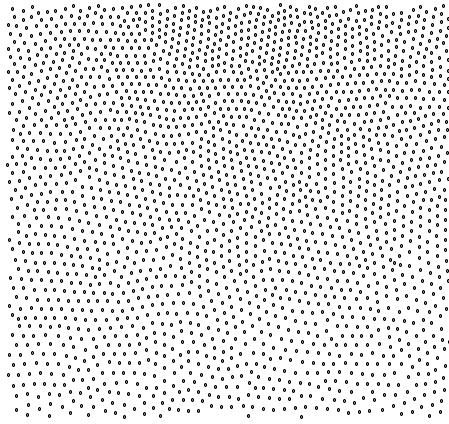
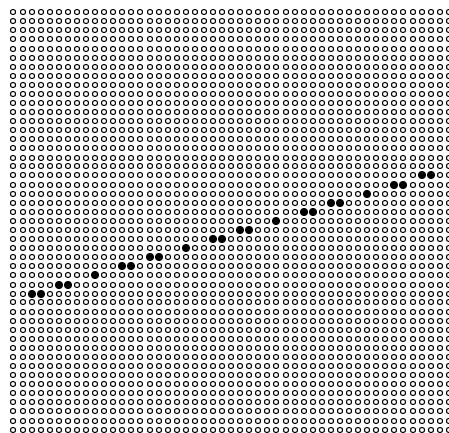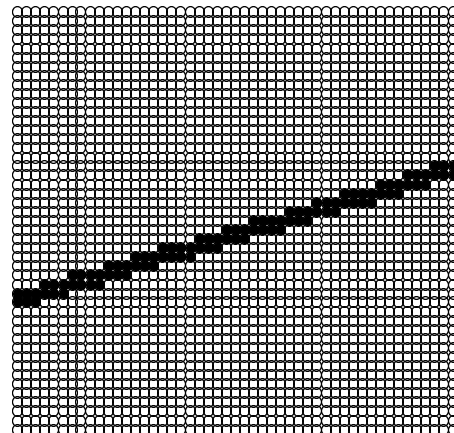(a) Sampling grid of retina   (b) Retina samples line with very small radius

(c) Retina samples line with small radius   (d) Retina samples line with large radius

(e) Regular sampling with small radius   (f) Grid samples line with large radius

Fig. 1. Sampling a line with irregular and regular grids

The reason to review these basic representational entities is to point out that in nearly all discretization concepts only one of these entities (be it the point or the 'dual' correspondent, e.g. the face in 2D or the voxel in 3D) carries the measured information. Only few representational schemes allow the other entities to store information appropriate for the particular dimension of the manifold.

4 Pyramids

In this section we summarize the concepts developed for building and using multiresolution pyramids [33, 21, 28] and put the existing approaches into a general framework. The focus of the presentation is the representational framework, its components and the processes that transfer data within the framework.
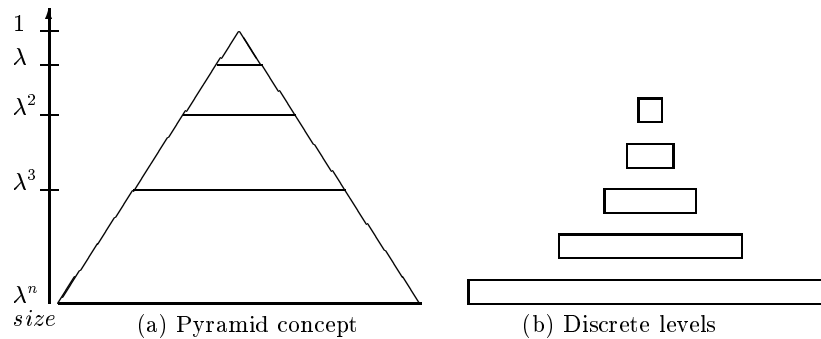
(a) Pyramid concept   (b) Discrete levels

Fig. 2. Multiresolution pyramid

A pyramid (Fig. 2) describes the contents of an image at multiple levels of resolution. The base level is a high resolution input image. Successive levels reduce the size of the data by a constant **reduction factor** $\lambda > 1.0$ while constant size local **reduction windows** relate one cell at the reduced level with a set of cells in the level directly below. Thus local independent (and parallel) processes propagate information up and down in the pyramid. The contents of a lower resolution cell is computed by means of a **reduction function** the input of which are the descriptions of the cells in the reduction window. Sometimes the description of the lower resolution needs to be extrapolated to the higher resolution. This function is called the **refinement** or **expansion function**. It is used in Laplacian pyramids [11] and wavelets [30] to identify redundant information in the higher resolution and to reconstruct the original data. The number of levels n is limited by the reduction factor λ: $n \leq \log(image\_size)/\log(\lambda)$. The main computational advantage of image pyramids is due to this logarithmic complexity.

We intent to extend the expressive power of these efficient structures by several generalizations. The reduction window and the reduction factor relate

two successive levels of a pyramid. In order to interpret a derived description at a higher level this description should be related to the original input data in the base of the pyramid. This can be done by means of the **receptive field** (RF) of a given pyramidal cell c_i: $RF(c_i$ collects all cells (pixels) in the base level of which c_i is the ancestor .

Since our goal is to bring up the 'relevant data' for solving a particular task let us give the term 'resolution' a more general meaning beyond the pure geometric definition.

This is the base of several pyramidal approaches, two of which are chosen as representatives: irregular graph pyramids and topological maps.

4.1   Irregular Graph Pyramids

A graph pyramid is a pyramid where each level is a graph $G(V, E)$ consisting of vertices V and of edges E relating two vertices. In the base level pixels are the vertices and two vertices are related by an edge if the two corresponding pixels are neighbors. This graph is also called the neighborhood graph. The content of the graph is stored in attributes attached to both vertices and edges. Initially only the attributes of the vertices receive the gray values of the pixels. In order to correctly represent the embedding of the graph in the image plane [19] we additionally store the dual graph $\overline{G}(\overline{V}, \overline{E})$ at each level. Let us denote the original graph as the primal graph.

In general a graph pyramid can be generated bottom-up as follows:

while further abstraction is possible do

1. determine contraction kernels
2. perform dual graph contraction and simplification of dual graph
3. apply reduction functions to compute content of new reduced level.

The complete formalism of **dual graph contraction** is described in [28]. Let us explain it here by means a small window of our line example (Fig. 1). The first step determines what information in the current top level is important
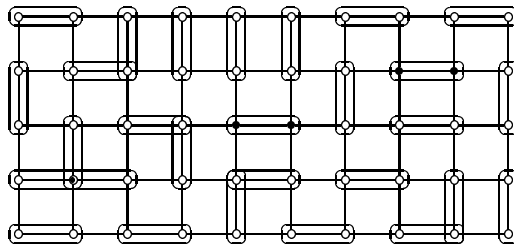

Fig. 3. Neighborhood graph G_0 and contraction kernel N_{01}

and what can be dropped. A contraction kernel is a (small) sub-tree of the top

level the root of which is chosen to survive. Fig. 3 shows the window and the selected contraction kernels each surrounded by an oval. Selection criteria in this case contract only edges inside connected components except for isolated black vertices which are allowed to merge with their background.

All the edges of the contraction trees are dually contracted during step 2. Dual contraction of an edge e (formally denoted by $G/\{e\}$) consists of contracting e and removing the corresponding dual edge $\overline{e}$ from the dual graph (formally denoted by $\overline{G} \setminus \{\overline{e}\}$). This preserves duality and the dual graph needs not be constructed from the contracted primal graph G' at the next level.

Since the contraction of an edge may yield multi-edges and self-loops there is a second phase of step 2 which removes all redundant multi-edges and self-loops. Note that not all such edges can be removed without destroying the topology of the graph: if the cycle formed by the multi-edge or the self-loop surrounds another part of the data its removal would corrupt the connectivity! Fortunately this can be decided locally by the dual graph since **faces of degree two** (having the double-edge as boundary) and **faces of degree one** (boundary = self-loop) cannot contain any further elements in its interior. Since removal and contraction are dual operations, the removal of a self-loop or of one of the double edges can be done by contracting the corresponding dual edges in the dual graph. The dual
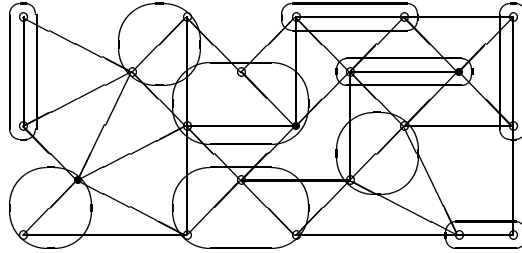

Fig. 4. Dually contracted graph G_1 and contraction kernel N_{12}

contraction of our example remains a simple graph G_1 without self-loops and multi-edges (Fig. 4).

Step 2 generates a reduced pair of dual graphs. Their contents is derived in step 3 from the level below using the reduction function. In our example reduction is very simple: the surviving vertex inherits the color of its son. In the only case where the contraction kernel contains two different colors, the isolated vertex is always chosen as surviving vertex.

The result of another dual contraction is shown in Fig. 5. The selection rules and the reduction function are the same as in the first iteration. The result shows that the bridges between the gaps of the original sampling has been closed and the three surviving black vertices are connected after two iterations. This fact could be used in a top-down verification step which checks the reliability of closing the gap in the more general context.
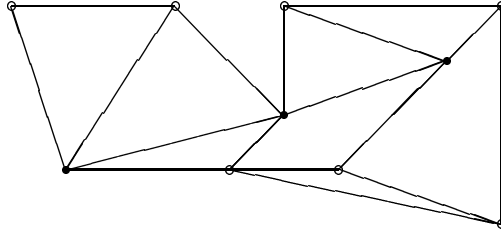
Fig. 5. Graph G_2 after two steps of dual graph contraction

There are lots of useful properties of the resulting graph pyramids. If the plane graph is transformed into a combinatorial map the transcribed operations form the combinatorial pyramid [6, 7]. This framework allowed us to prove several of the above mentioned properties and links dual graph pyramids with topological maps which extend the scope to three dimensions.

The following table summarizes dual graph contraction in terms of the control parameters used for abstraction and the conditions to preserve topology:

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

4.2 The Topological Map

Fiorio, Bertrand and Damiand developed a method that derives the topologically correct region adjacencies both in 2D and in 3D [1, 13]. Their base representations are combinatorial and generalized maps. The derivation proceeds in several levels similar to a pyramid with the difference that the lower levels are not used any more after the higher level has been created. In 2D following levels are identified:

Level	representation	merge	conditions
0	complete inter-pixel map		
	$\downarrow$	adjacent faces	same label
1	line map		
	$\downarrow$	lines (l_1, v, l_2)	deg $v = 2$ and colinear(l_1, l_2)
2	border map		
	$\downarrow$	lines (l_1, v, l_2)	deg $v = 2$
3	topological map		

In addition to the topological map an inclusion tree is generated and maintained to cope with holes in regions. The resulting representation is computed by a sequential scan line procedure comparing 2×2 windows with prestored 'pre-codes' each requiring a special treatment. The current approach has no limit on the size of the reduction window in the sense that any number of elements ('darts') of level i may be merged into a single element at level $i + 1$. However a large set of fused elements could be decomposed into a hierarchy of locally independent fusions to allow parallel implementation.

The extension into three dimensions is straight forward and contains six different levels [13]:

Level	representation	merge	conditions
0	complete inter-voxel map		
	$\downarrow$	adjacent volumes	same label
1	line map		
	$\downarrow$	faces (f_1, l, f_2)	$\deg l \leq 2$, coplanar(f_1, f_2)
2	level 2 map		
	$\downarrow$	lines (l_1, v, l_2)	$\deg l = 2$, colinear(l_1, l_2)
3	border map		
	$\downarrow$	faces (f_1, l, f_2)	$\deg v \leq 2$
4	level 4 map		
	$\downarrow$	lines (l_1, v, l_2)	$\deg v = 2$, $l_1 \neq l_2$
5	topological map		

Besides the inclusion tree that captures the 3D holes, fictive edges are maintained to prevent disconnection of faces or lines. Additional conditions are implicitly expressed to prevent disconnections and suppression of basic elements. The process proceeds similar to the 2D case: $2 \times 2 \times 2$ windows are compared with a set of 'pre-codes' and the corresponding procedure is executed for each code. The levels and several categorizations permit to reduce the number of cases drastically from the ones that would be needed for the exhaustive enumeration.

5   Abstraction in Pyramids

In order to discuss the role of abstraction in general multiresolution hierarchies. Let us consider the structure of the representation and the content stored in the representational units separately. In our generalization we allow the resolution cell to take other simply connected shapes and to describe the content by a more complex 'language'.

The first generalization is a consequent continuation of the observations of Bister etal [4] to overcome the limited representational capabilities of rigid regular pyramids. It necessitates to consider in more detail the *structure* of a resolution level which was implicitly coded as a matrix in regular sampling grids. A similar conclusion was expressed by DeFloriani etal [14] where a 'multi-complex' was presented as a unifying framework for many multiresolution regular cell complexes and extensions to cope with non-regular shapes are envisioned. Since irregular structures reduce the importance of explicitly representing geometry,

topological aspects become relevant. These aspects will be discussed in more detail in section 6.

The second generalization started with the works of Hartley [20] who allowed a resolution cell to contain more than one (gray) value and interpreted them as the parameters of a globally defined model.

5.1 Content Models and Reduction Functions

In the topological map approach each cell contains a label identifying the membership of the cell to the class of all those cells having the same label. In this case the contents of the cells merged during the reduction process can be propagated by simple inheritance: the fused cell 'inherits' its label from its children and it does not matter from which since all have the same label.

In classical gray level pyramids the contents of a cell is a gray value which is summarized by the mean or a weighted mean of the values in the reduction window. Such reduction functions have been efficiently used in Gaussian pyramids. Laplacian pyramids, Ratio pyramids and wavelet pyramids identified the loss of information that occurs in the reduced level and stored the missing information in the hierarchical structure where it could be retrieved when the original base level is reconstructed. All these approaches use **one single globally defined model** which must be flexible enough to adapt its parameters to approximate the data.

In our generalization we would like to go one step further and allow **different models** to be used in different resolution cells as there are usually different objects **at different locations** of an image. The models could be identified by a name or a symbol and may be interrelated by semantic constraints.

Simple experiments have been done with images of line drawings. This research used the experiences gained with a regular $2 \times 2/2$ curve pyramid [25] and the chain pyramid [31] in the more flexible framework of graph pyramids. The model describes symbolically the way how a curve intersects the discrete segments of the boundary of a cell and the reduction function consists in the transitive closure of the symbols collected in the reduction window. The concept works well in areas where the density of curves was low, through the rigidity of the regular pyramid ambiguities arise when more curves appeared within the same receptive field. This limitation could be effectively overcome with irregular pyramids in which we could limit the receptive field of a cell to a single curve. Fig. 6 gives an impression of the complexity of the data that have been processed in the minimum line property preserving (MLPP) pyramid in order to find and identify components in a technical drawing. More details can be found in [8].

The content abstraction in this representation has following features:

- models are identified by (discrete) names (empty cell, line end, line crosses edge, junction), no parameters were used;
- adjacent models have to be consistent ('good continuation');
- contraction kernels were selected such that only one consistent curve is covered in one receptive field;
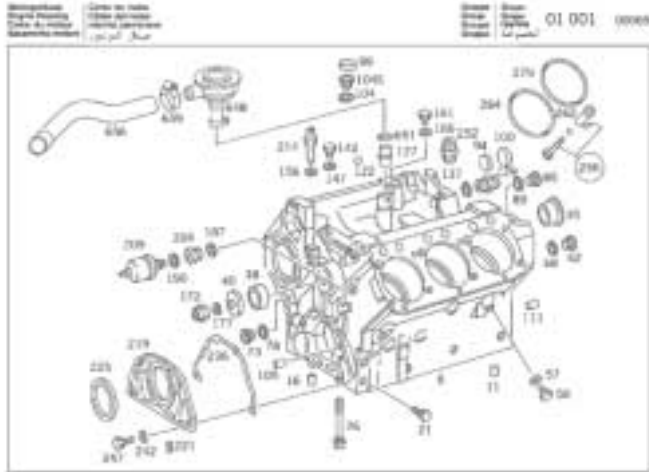
Fig. 6. Technical drawing used to build an MLPP-pyramid

– a few symbolic rules of the form:

$$\text{new model} \mapsto \text{local generic sub-graph with model assignments} \quad (1)$$

governed this selection process;
– the reduced content appears in the left hand side of the rule (1).

One may notice a certain similarity between the selection rule (1) and the use of pre-codes in the efficient computation of the topological maps [2, 1, 13].

In both cases the knowledge about the models and in what configurations they are allowed to occur needs to be stored in a knowledge base. In order to determine which are the best possible abstractions the local configurations at a given level of the pyramid must be compared with the possibilities of reduction given in the knowledge base. This would typically involve matching the local configuration with the right-hand sides of rules stored in the knowledge base. Such a match may not always be perfect, one may allow a number of outliers. The match results in a goodness of match, which can be determined for all local configurations. The selection can then choose the locally best candidates as contraction kernels and reduce the contents according to the generic models which matched the local configuration. The goodness of match may also depend on a global objective function to allow the overall purpose, task or intention to influence the selection process.

5.2 The Knowledge Base

The knowledge to be used in the pyramid can be organized in many different ways. It is needed in selecting surviving cells and as the model for the reduction

function. Since the aim is to achieve abstraction it should provide also, besides a goodness of fit, a measure of relevance or importance which may depend on varying goals of the system. These variations in the goals further need to be communicated to the system. Hence the language of interaction must be part of the knowledge base.

It is certainly impossible to exhaustively enumerate all the possibilities to organize the knowledge. We list a few exemplary ways (1., 2.) one can find in existing approaches and propose a structure (3.) which fits to the concept sketched above.

1. The most common way to enter semantic knowledge into a system is by **implicit coding**. In the pyramid building process it appears as the parameters that control the selection function or the frequently used filters in the reduction function. Although often computationally efficient implicit coding has no flexibility to adapt to the data and any change requires a modification of the program code. This type of knowledge representation can be found in most regular pyramid approaches which are based on linear filters.

2. In order to separate the knowledge from the process working on the data, globally coded rules or states are associated with specific procedures to treat the data. This knowledge is pre-compiled and is accessed through indices based on locally computed features. As examples can be mentioned the pre-codes for building the topological map and the syntactical rules used for the line drawing application (Fig. 6). These types of knowledge representations are more flexible than the implicit coding since the knowledge base can be extended or adapted to the special needs of an application. However the knowledge is used mostly in a deterministic way without assigning special priorities to items important for the current task. Furthermore the knowledge is in all cases compiled manually which limits the scope of the application. A few 'rules' have been used for the line drawing example, application dependent rules would need an enormous amount of effort. Similarly for the definition of pre-codes of the topological map: the 2D construction needs only 12 pre-codes which grows to more than 4000 cases in 3D. Although this number could be substantially reduced using the six levels and further categorization the extension to higher dimensions seems to be prohibitive.

3. The concept presented in the previous section suggests that the knowledge base can provide pre-stored local configurations that can be used to identify potential local contraction kernels. There are several approaches pointing in this direction, e.g. the one presented by Kittler etal [22] where three interrelated levels express the knowledge of the system: the measurements, the image features, and the object cues. In addition each 'configuration' must be associated with the following:

   - A function computing the importance based on the goodness of match between the data configuration and the pre-stored configuration.
   - The more abstract description needs to be identified, e.g. by a name or a symbol.

– The reduction function associated with the new identity calculates the specific parameters (attributes of the survivors) from the attributes of the data configuration.

Such a knowledge base could be realized as a formal (graph) grammar. However, as the examples already demonstrated, in order to be effective the knowledge needs a non-negligible degree of complexity. This has consequences for both the knowledge retrieval and updating: Knowledge retrieval requires a high degree of internal organization to quickly access all those configurations that must be checked in a particular case. Exhaustive search may not be feasible. One could imagine a structure similar to the data pyramid since also the abstract terms have neighbors, e.g. associations in an abstract sense, more specific terms they are derived from and more abstract terms they are part of. Let us call this the **abstraction pyramid** in contrast to the **data pyramid** having the image data in the base. In this case the possible configurations to be checked in a particular part of the data pyramid could be local neighbors in the abstraction pyramid of the abstract term associated with the data cell. Note that a simplified version has been proposed for regular pyramids by P. Burt: the pattern tree [10].

The second consequence of the high complexity of the knowledge base concerns the updating: both interactive user interfaces and learning strategies could be integrated with the concept of the abstraction pyramid. However there is still a wide field of research necessary before such systems could be used in 'real' applications.

6   Preserving Topology

Objects mapped into images remain connected if they are not occluded by other objects nor disturbed by noise. Neither the projection nor the discretization separate the two corresponding adjacent regions in the image. A similar property holds for the adjacency of objects. Hence the **connectivity** and the **adjacency** of **regions** and of **boundary segments** is a very important property which should not be lost by abstraction. Several authors studied operations that allow the modification of the data structure, e.g. its reduction, while the topological properties of the represented objects and their background is preserved (e.g. [23, 32, 3, 26, 15, 9]).

In the following we first look at the simpler cases in two dimensions and refer to the dual graph pyramid. Then some considerations about the reduction operations in 3D based on the recent results of Damiand are discussed.

6.1   Preserving Topology in 2D

Table 1 summarizes the necessary primitive operations: The Euler number characterizes the topology of a description given in terms of points ($\#P$), lines ($\#L$) and faces ($\#F$). Since we aim at preserving its value the sum of the changes must be zero:
$$\Delta\#P - \Delta\#L + \Delta\#F = 0.$$

Table 1. Topology Preserving Operations in 2D

	Points	Lines	Faces	Config.	PRE-CONDITION	CCL
Euler	$\#P$	$-\#L$	$+\#F$		$=$ const.	
Incr.	$\Delta\#P$	$-\Delta\#L$	$+\Delta\#F$		$= 0$	Euler
Contract(l, p_0)	-1	-1	0	(p_1, l, p_0)	$p_1 \neq p_0$	same label
Remove(l, f_0)	0	-1	-1	(f_x, l, f_0)	$f_x \neq f_0$	$\deg(f_0) \leq 2$
Any Incr.	$(-a$	$-b$	$-c)$		$b = a + c;$	
by a contr.	$(-1$	$-1)$		$\times a$		
by c remov.		$(-1$	$-1)$	$\times c$		

First we observe the changes introduced by **contracting an edge** l bounded by the two points p_0, p_1. This eliminates one of the points (i.e. p_0) and the edge l, hence it does not change the Euler characteristic. The only pre-condition is to avoid contracting a self-loop.

If we **remove an edge** l, the number of points remains the same, but two faces f_0, f_x are merged into one (f_x). That reduces the number of faces by one. If we would have the same face on both sides of the edge, i.e. $f_x = f_0$, the edge would be a bridge in G the removal of which would disconnect G. If one of the end points of l would have degree 1, the removal of its only connection to the remaining structure would isolate it. Both cases are excluded from removal by the pre-condition $f_x \neq f_0$. The second pre-condition $\deg(f_0) \leq 2$ identifies a redundant self-loop or a redundant multi-edge: in the later case f_0 is bounded by two parallel edges connecting the same end-points. This configuration is simplified in the second phase of dual graph contraction.

What about other operations? It is clear that the elimination of an edge must be accompanied by the removal of either a point or a face to preserve the Euler number. So we cannot have less elements involved in a topology preserving modification. But we can also show the following:
Contraction and removal are the ONLY operations needed to reduce the structure while preserving the topology. Any other topology-preserving operation can be achieved by appropriate combinations of contraction and removals. If we want to remove a number a of points and a number c of faces we have to remove also a number $b = a + c$ of edges to preserve the Euler number. This can be achieved by a contractions and c removals.

Pre-conditions for individual operations can be extended to sets of operations to allow a different order of execution or even parallelism: The requirement for contraction kernel to form a FOREST is such an extension. If the edges of a cycle would be contracted the last one need to be a self-loop which cannot be contracted. Hence sets of edges to be contracted must be acyclic.

6.2 What Remains after Repetitions?

We can repeat contracting edges the end point of which carry the same label and remove all unnecessary self-loops and multi-edges until no further contraction

nor removal is possible. Note that a very similar strategy is used to create the border map in [2, 1] and the topological map in [13]. At convergence we have the following conditions:

1. all edges (p_1, l, p_2) with different end points have different labels: $lab(p_1) \neq lab(p_2)$.
2. A surviving self-loop (p, l, p) separates two different faces, (f_1, l, f_2) and the inner face has degree $\deg(f_1) > 2$. Since any tree of the dual graph would have been eliminated by the rule $\deg(f_1) \leq 2$ starting from the leafs up to the root, there must be a cycle $C \in \overline{G}$ and inside this cycle there exists a point $p_3 \in C : lab(p_3) \neq lab(p)$.
3. All faces have three or more sides: $\deg(f) \geq 3$.
4. Pseudo or fictive edges are self-loops $(p_0 = p_1)$ which cannot be contracted in the primal graph and which separate two faces with $\deg(f_0) > 2$ and $\deg(f_1) > 2$. Such edges were first observed in [29] as an artifact of topology preserving contraction. They connect the boundary of a hole to the surrounding 'main land'. Holes can be equivalently represented by an inclusion tree as in [13].
5. Fictive edges appear arbitrarily placed and depend only on the order of contractions and removals. Similar observations can be found in the topological 3D-map [13] where fictive edges appear as the last ones before disconnecting a face or a boundary during the merging of faces and lines (for the process of successive region merging see [1, 2]).
6. For each hole there remains exactly one fictive edge (as indicated by the Betti number [17]).
7. Fictive edges are not delineated between two regions as all other edges. Hence they can be continuously deformed and their end points can be moved along the boundary as long as the edge remains fully embedded inside f. Other fictive edges are not excluded from being traversed by the end point! We conjecture that an arrangement of fictive edges can be transformed into any other legal arrangement of fictive edges. Algorithms for continuous deformation [34] or [18] may find a new application for re-arranging fictive edges.

6.3 Preserving Topology in 3D

The primitive operations to build the 3D topological map merge voxels (V-Fusion), faces (F-Fusion) and linels (L-Fusion). In analogy to the 2D table Table 2 summarizes the necessary primitive operations in 3D:

1. V-Fusion, F-Fusion and L-Fusion are the ONLY operations needed, the reasoning is the same as in 2D. Any other topology-preserving operation can be achieved by appropriate combinations.
2. Pre-conditions in 3D are non-trivial except for volumes: a line may delimit more than 2 faces, and a point may be the intersection of more than 2 lines. Damiand [13] lists additional constraints: no disconnection and no suppression of any face or line should be possible.

Table 2. Topology Preserving Operations in 3D

	Pts.	Lin.	Fac.	Vol.	Config.	PRE-CONDITION	CCL
Euler	$\#P$	$-\#L$	$+\#F$	$-\#V$		$=$ const.	
Incr.	$\Delta\#P$	$-\Delta\#L$	$+\Delta\#F$	$-\Delta V$		$=0$	Euler
V-Fusion			-1	-1	(v_1, f, v_2)	$v_1 \neq v_2$	same label
F-Fusion		-1	-1		(f_1, l, f_2)	$f_1 \neq f_2$	$\deg l \leq 2$
L-Fusion	-1	-1			(l_1, p, l_2)	$l_1 \neq l_2$	$\deg p \leq 2$

3. It remains to be checked whether the pre-conditions for individual operations can be extended to sets of operations.
4. Semantic control (i.e. checking the same label as for CCL) occurs only in the initial fusions, all other operations are automatic simplifications.
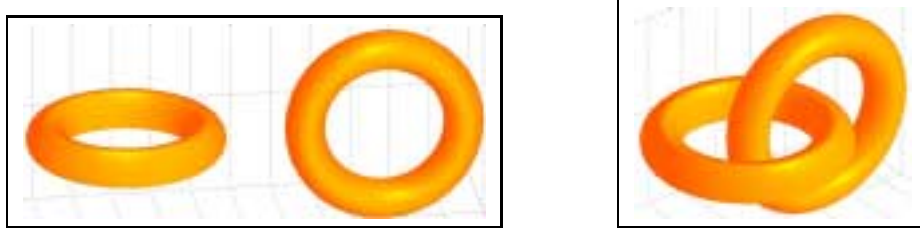
6.4 What should remain in 3D after repetition?

Fig. 7. Are the two 3D configuration the same?

It is not surprising that the complexity of minimal topological cases to be considered grows with dimension. In 2D only holes have to be considered and correctly represented by either an inclusion tree or additional fictive edges. In 3D, tunnels may be present additionally both in the foreground and in the background.

Configurations like the two tori as in Fig. 7 need to be distinguished. The answer 'there are two tori' is not wrong but it is unsatisfactory since it does not distinguish the two depicted configurations. Most representations do not even have the necessary relation to express the interlacing between the two tori in the second case. In addition they are not connected. So the description most involve the topology of the surrounding background.

Let us sketch a possible solution using fictive elements. A fictive surface intersecting the tunnel could be used to make a torus (genus 1) homotopic to a sphere (genus 0). This fictive surface is not fixed geometrically in space but another object, like the second torus, would intersect it and create a hole. The boundary of the hole would be connected by a fictive edge to the outer boundary

Fig. 8. Two tori with two fictive surfaces and one fictive edge

of the fictive surface added to the torus. An identical reasoning applies to the second torus creating in total two fictive surfaces and one fictive edge as depicted in Figure 8.

The consequences for abstract representations open a wide range for further research and may address deep mathematical problems in different fields: knot theory, Morse complexes, algebraic topology, . . . (see [17, 16]).

7   Conclusion

We motivated our discussion by Dickinson's and Udupa's claims to *'bridge the representational gap'*, to *'focus on image abstraction'*, and to study *'topological properties'* in the introduction. We first discussed the basic fields, abstraction and discrete representation, in more detail. It seems that there are much less concepts working on discrete irregular grids than on their regular counterparts. We then recalled two pyramidal approaches having the potential to cope also with irregular grids. These pyramids have some useful properties, e.g.

1. they show the necessity to use multi-edge and self-loop to preserve the topology;
2. they allow to combine primitive operations at one level (i.e. collected by the contraction kernel) and across several levels of the pyramid (i.e. equivalent contraction kernels [27]);
3. repeated contraction converges to specific properties which are preserved during contraction;
4. termination criteria allow to stop abstraction before a certain property is lost.
5. pseudo/fictive elements characterize topological relations, a fictive edge characterizes a hole, a fictive face characterizes a tunnel, . . . ;

7.1   Open Problems

There are numerous open problems partly addressed in the paper. Let us just enumerate a few important issues:

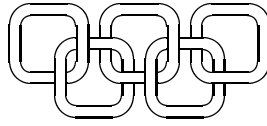1. extensions to 3D, 4D, 5D (see [35] for current data sources)

Fig. 9. The Olympic problem

2. how to represent multiple interlaced tori, chains, the Olympic rings (Fig. 9) in a topologically correct representation?
3. Re-insertion of removed edges/darts (like reconstruction with wavelets): after reducing a level of the pyramid the data remaining in the level below could be checked for redundancies and store only the differences needed for loss-less reconstruction;
4. repeated contraction has several control parameters which allow adaptation to specific applications:
   - different selection criteria
   - different termination criteria
   - different attributes
   - different reduction functions
5. these control parameters could be organized in a knowledge base to allow further user interaction and automation through learning.

References

[1] Y. Bertrand, G. Damiand, and C. Fiorio. Topological Encoding of 3D Segmented Images. In G. Borgefors, I. Nyström, and G. Sanniti di Baja, editors, *Proceedings DGCI'00, Discrete Geometry for Computer Imagery*, volume Vol. 1953 of *Lecture Notes in Computer Science*, pages 311–324, Uppsala, Sweden, 2000. Springer, Berlin Heidelberg, New York.

[2] Y. Bertrand, C. Fiorio, and Y. Pennaneach. Border Map: A Topological Representation for nD Image Analysis. In G. Bertrand, M. Couprie, and L. Perroton, editors, *Discrete Geometry for Computer Imagery, DGCI'99*, volume Vol. 1568 of *Lecture Notes in Computer Science*, pages 242–257, Marne-la-Vallée, France, 1999. Springer, Berlin Heidelberg, New York.

[3] J. C. Bezdek and N. R. Pal. An index of topological preservation for feature extraction. *Pattern Recognition*, 28(3):381–391, March 1995.

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

[5] V. E. Brimkov, E. Andres, and R. P. Barneva. Object Discretization in Higher Dimensions. In G. Borgefors, I. Nyström, and G. Sanniti di Baja, editors, *Proceedings DGCI'00, Discrete Geometry for Computer Imagery*, volume Vol. 1953 of *Lecture Notes in Computer Science*, pages 210–221, Uppsala, Sweden, 2000. Springer, Berlin Heidelberg, New York.

[6] L. Brun and W. G. Kropatsch. Contraction Kernels and Combinatorial Maps. In J.-M. Jolion, W. G. Kropatsch, and M. Vento, editors, *Graph-based Representations in Pattern Recognition, GbR 2001*, pages 12–21. CUEN, 2001. ISBN 88 7146 579-2.

[7] L. Brun and W. G. Kropatsch. Introduction to Combinatorial Pyramids. In G. Bertrand, A. Imiya, and R. Klette, editors, *Digital and Image Geometry*, pages 108–128. Springer, Berlin, Heidelberg, 2001.

[8] M. Burge and W. G. Kropatsch. A Minimal Line Property Preserving Representation of Line Images. *Computing, Devoted Issue on Image Processing*, 62:pp. 355–368, 1999.

[9] J. Burguet and R. Malgoures. Strong Thinning and Polyhedrization of the Surface of a Voxel Object. In G. Borgefors, I. Nyström, and G. Sanniti di Baja, editors, *Proceedings DGCI'00, Discrete Geometry for Computer Imagery*, volume Vol. 1953 of *Lecture Notes in Computer Science*, pages 222–234, Uppsala, Sweden, 2000. Springer, Berlin Heidelberg, New York.

[10] P. 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.

[11] 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.

[12] J.-M. Chassery and A. Montanvert. *Géométrie discrète en analyse d'images*. Traité des Nouvelles Technologies, série – Images. HERMES, Paris, France, 1991.

[13] G. Damiand. *Définition et étude d'un modèle topologique minimal de représentation d'images 2d et 3d*. PhD thesis, LIRMM, Université de Montpellier, 2001.

[14] L. De Floriani, P. Magillo, and E. Puppo. Multiresolution Representation of Shapes Based on Cell Complexes. In G. Bertrand, M. Couprie, and L. Perroton, editors, *Discrete Geometry for Computer Imagery, DGCI'99*, volume Vol. 1568 of *Lecture Notes in Computer Science*, pages 3–18, Marne-la-Vallée, France, 1999. Springer, Berlin Heidelberg, New York.

[15] T. K. Dey, H. Edelsbrunner, S. Guha, and D. V. Nekhayev. Topology Preserving Edge Contraction. *??*, pages pp. ??–??, 1999.

[16] H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse Complexes for Piecewise Linear 2-Manifolds. In *SCG'01, June 3-5, 2001, Medford, Mass. USA*, pages ??–?? ACM 1-58113-357-X/01/0006, 2001.

[17] R. Forman. Combinatorial Differential Topology and Geometry. *??*, pages pp. ??–??, 1999?

[18] S. Fourey and R. Malgouyres. Intersection Number of Paths Lying on a Digital Surface and a New Jordan Theorem. In G. Bertrand, M. Couprie, and L. Perroton, editors, *Discrete Geometry for Computer Imagery, DGCI'99*, volume Vol. 1568 of *Lecture Notes in Computer Science*, pages 104–117, Marne-la-Vallée, France, 1999. Springer, Berlin Heidelberg, New York.

[19] R. Glantz and W. G. Kropatsch. Plane Embedding of Dually Contracted Graphs. In G. Borgefors, I. Nyström, and G. Sanniti di Baja, editors, *Proceedings DGCI'00, Discrete Geometry for Computer Imagery*, volume Vol. 1953 of *Lecture Notes in Computer Science*, pages 348–357, Uppsala, Sweden, 2000. Springer, Berlin Heidelberg, New York.

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

[21] J.-M. Jolion and A. Rosenfeld. *A Pyramid Framework for Early Vision*. Kluwer Academic Publishers, 1994.

[22] J. Kittler, K. Messer, W. Christmas, B. Levienaise-Obadia, and D. Koubaroulis. Generation of semantic cues for sports video annotation. In *Proceedings IEEE International Conference on Image Processing, ICIP2001*, pages pp.26–29, Thessaloniki, Gr., 2001.

[23] T. Kong and A. Rosenfeld. Digital topolgy: a comparison of the graph-based and topological approaches. In G. Reed, A. Roscoe, and R. Wachter, editors, *Topology and Category Theory in Computer Science*, pages 273–289, Oxford, 1991. Oxford University Press.

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

[25] W. 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.

[26] W. G. Kropatsch. Property Preserving Hierarchical Graph Transformations. In C. Arcelli, L. P. Cordella, and G. Sanniti di Baja, editors, *Advances in Visual Form Analysis*, pages 340–349. World Scientific Publishing Company, 1997.

[27] W. G. Kropatsch. From equivalent weighting functions to equivalent contraction kernels. In E. Wenger and L. I. Dimitrov, editors, *Digital Image Processing and Computer Graphics (DIP-97): Applications in Humanities and Natural Sciences*, volume 3346, pages 310–320. SPIE, 1998.

[28] W. G. Kropatsch, A. Leonardis, and H. Bischof. Hierarchical, adaptive and robust methods for image understanding. *Surveys on Mathematics for Industry*, No. 9:pp. 1–47, 1999.

[29] W. G. Kropatsch and H. Macho. Finding the structure of connected components using dual irregular pyramids. In *Cinquième Colloque DGCI*, pages 147–158. LLAIC1, Université d'Auvergne, ISBN 2-87663-040-0, September 1995.

[30] S. 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.

[31] P. Meer, C. A. Sher, and A. Rosenfeld. The chain pyramid: Hierarchical contour processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(No.4):pp.363–376, April 1990.

[32] P. F. Nacken. Image segmentation by connectivity preserving relinking in hierarchical graph structures. *Pattern Recognition*, 28(6):907–920, June 1995.

[33] A. Rosenfeld, editor. *Multiresolution Image Processing and Analysis*. Springer, Berlin, 1984.

[34] A. Rosenfeld and A. Nakamura. Local deformations of digital curves. *Pattern Recognition Letters*, 18:613–620, 1997.

[35] J. K. Udupa. Go Digital, Go Fuzzy. In G. Borgefors, I. Nyström, and G. Sanniti di Baja, editors, *Proceedings DGCI'00, Discrete Geometry for Computer Imagery*, volume Vol. 1953 of *Lecture Notes in Computer Science*, pages 284–295, Uppsala, Sweden, 2000. Springer, Berlin Heidelberg, New York.