CPRW'97, Milovy, CZ, February 17-20, 1997

From Equivalent Weighting Functions To Equivalent Contraction Kernels*

Walter G. Kropatsch
Vienna Univ. of Technology, Inst. of Automation, 183/2
Pattern Recognition + Image Processing Group
Treitlstr.3 / A-1040 WIEN / Austria

Abstract

Burt introduced 1983 'equivalent weighting function': *"Iterative pyramid generation is equivalent to convolving the image g_0 with a set of 'equivalent weighting functions' h_l:"* $g_l = h_l * g_0 = h * g_{l-1}, l > 1$. It allowed him to study the effects of iterated reduction (e.g. the low-pass character of Gaussian pyramids) using the single parameter h_l without giving up the efficient iterative computation.

A similar concept applies to graph pyramids built by dual graph contraction. This new algorithm reduces the number of vertices and of edges of a pair of dual image graphs while, at the same time, the topological relations among the 'surviving' components are preserved. Repeated application produces a stack of successively smaller graphs: a pair of dual irregular pyramids. The process is controlled by selected decimation parameters which consist of a subset of surviving vertices and associated contraction kernels. These play a similar role for graph pyramids than the convolution kernels of Gaussian pyramids. Equivalent contraction kernels (ECKs) combine two or more contraction kernels into one single contraction kernel which generates the same result in one single dual contraction. The basic concepts are elaborated and discussed. The new theory opens a large variety of possibilities to explore the domain of 'all' graph pyramids.

1 Introduction

A raw digital image consists of a 2D spatial arragement of pixels each of which results from measuring the light at a specific location of the image plane. Currently most of the artificial sensors (e.g. CCD cameras) have the rigid structure of an orthogonal grid, whereas most natural vision systems are based on non-regular arrangements of sensors [1]. Although arrays are certainly easier to manage technically, topological relations seem to play an even more important role for vision tasks in natural systems than precise geometrical positions.

A second aspect concerns the projection from the real (3D-) world into the 2D image. Surfaces of 3D-objects reflect the light in a very specific way that somehow 'codes' the structure of the object: reflectivity is a property of material and does not vary much along

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

object surfaces, it changes abruptly between different surfaces or from the object to its background [15]. Image properties can be formally described (see Minsky and Papert [18]) by predicates $\psi(X), X \subset I$ where X is the subset of pixels belonging to the image of an object.

The topological structure on a visible surface patch is preserved in the image while its geometry may be severly distorted. But also the arrangement of different objects in the 3D-world will be mapped to the regions in the image, be it regularly or irregularly sampled. Hence the idea pursuit in this paper to start with arbitrarily but densely sampled measurements of which only the topology is known and to successively shrink the number of descriptive elements until the structure of the imaged scene becomes evident.

The third aspect addresses computer vision models. They have in general a parametric and a structural component. While parameter optimization models quantitative image properties well, the qualitative image and scene properties rely more on the structural component.

The presented approach addresses a representation of pure structure, a hierarchy of plane graphs, with a clear interface, the decimation parameters, to control generation and modification of the structure. Dual graph contraction is the basic process [10] that builds an irregular 'graph' pyramid by successively contracting a dual image graph of one level into the smaller dual image graph of the next level. Dual image graphs are typically defined by the neighborhood relations of image pixels or by the adjacency relations of the region adjacency graph. The above concept has been used for finding the structure of connected components [14, 13]. It also embeds Meer's stochastic pyramid [17], the adaptive pyramid [6], and a further variant of Meer's approach, Mathieu's optimal stochastic pyramid [16] which produced excellent segmentation results by decimating a minimal spanning tree instead of the original graph.

The paper is organized as follows. We first review classical image pyramids and recall the effect of Burt's equivalent weighting function (section 2). In the sequel graphs take over the role of the regular grid structure of image arrays (section 3). Then we summarize and illustrate the procedure of dual graph contraction in Section 4. The observation that the parameters that control the process form forests is then generalized by the concept of contraction kernels. Repeated dual contractions can be replaced by a single dual contraction using equivalent contraction kernels (ECKs, section 4.2). ECKs are able to compute any level of an irregular (graph-) pyramid directly from the base. Decimation parameters can be designed now at the base without the need to first generate the lower pyramid levels. The conclusion (Section 5) summarizes the major advantages of both regular and irregular pyramids.

2 Image Pyramids

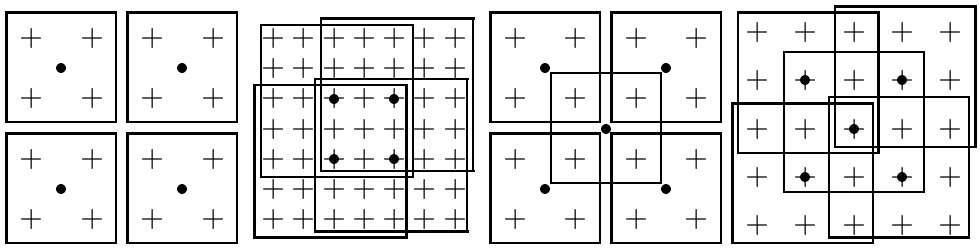
Classical image pyramids have been introduced 1981/82 [4, 20] as a stack of images of decreasing resolutions. Since then several modifications and additions have been made to the original concept [9] while main properties are still valid.

Let us define a digital image as a function g_k of a finite set L_k of resolution cells [5] that cover a limited area of the continuous image plane into a limited range of grey values. The different images $g_k : L_k \mapsto [0, 255]$ in the pyramidal stack are ordered according to their

spatial resolution[1]

$$L_0 \subset L_1 \subset \ldots \subset L_n.$$

In this ordering the individual images are called **levels** and are numbered from bottom L_0 (highest resolution) to top L_n. Based on the neighborhood relations of an image, the notion of **reduction window** is introduced for regular (e.g. square grid) structures. It relates every cell $c_{k+1} \in L_{k+1}$ to a set of cells $w(c_{k+1}) \subset L_k$, the **children** of cell c_{k+1}. In a regular pyramid all (interior) cells have the same number of neighbors and children.



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

Figure 1: Regular pyramids: levels k (+) and $k+1$ (•), reduction window

A second parameter describes different types of pyramids: the **reduction factor** λ. It captures the rate by which the number of cells decreases

$$|L_{k+1}| \leq \frac{|L_k|}{\lambda} \text{ for } 0 \leq k < n$$

In the formal notation "$2 \times 2/4$", 2×2 defines the size of the reduction window, and the reduction factor 4 expresses that level $k+1$ contains only a quarter of pixels of level k. The notation *(reduction window)/ reduction factor* characterizes the type of the regular pyramid. Fig. 1 overlays two adjacent levels of four different types of regular pyramids. Note that adjacent reduction windows may overlap (see Fig. 1b, c, d).

If the information in a pyramid is transmitted only across parent-child links every cell $c_k \in L_k, k > 0$ summarizes information from cells in the levels $L_i, 0 \leq i < k$. In the level L_{k-1} directly below the cell c_k they form the reduction window $w(c_k)$. Every cell of the reduction window in turn has children in lower levels. An **equivalent window** $w^j(c_k)$ covers all cells at a given level $j < k$ that link to the same cell c_k. Formally w^j can be defined recursively:

$$\begin{aligned} w^0(\bullet) &:= w(\bullet) \\ w^{i+1}(p) &:= \bigcup_{q \in w(p)} w^i(q) \qquad \text{for } 0 \leq i < n \end{aligned} \tag{1}$$

$w^j(c_k)$ defines the domain of measurements that are summarized as contents of c_k.

The information stored in the cells of a pyramid range from single gray values up to very complex descriptions of 'what the cell sees'. We consider also parameters of one or more models, selected symbols from a vocabulary, simple symbolic descriptions defined by a

[1]The geometrical center of a cell may be shifted to the center of gravity of its reduction window, i.e. in Fig. 1(a, c).

formal language, and attributed graphs. The processes working on pyramidal representations compute or modify either the contents of the cells or the structure of the pyramid.

Reduction functions $g_{k+1} := R(g(w(c_{k+1})))$ compute the contents of cells c_{k+1} from the contents of cells in the reduction windows. Typical functions used are convolutions with predefined (e.g. Gaussian) kernels, filters, interpolation, morphological operations, model fitting (optimization), ... In a convolution the reduction function is a weighted sum

$$R(g(w(c_{k+1}))) := \Sigma_{q \in w(c_{k+1})} c_i g(q)$$

where c_i are predefined constant weights associated with q in the ordering of $w(\bullet)$. These weights are called convolution kernels in the square grid arrangement.

In regular structures the **top-down refinement** of a cell c_k into equivalent windows $w^j(c_k)$ can be continued recursively. If the size of the cells is reduced (geometrically) according to the reduction factor, the equivalent windows form shapes that converge towards geometrical shapes (squares for $n \times n/4$ type pyramids, octagones for $n \times n/2$ type pyramids) that depend only on the size of cell c_k and the type of pyramid [8]. We call this geometrical region of the continuous image plane the **receptive field** of cell c_k. Since equivalent windows are monotonically increasing they are all contained in the receptive field. It also characterizes the type of a regular pyramid.

2.1 Equivalent Weighting Functions

In any regular pyramid the number of neighbors is fixed. As a consequence any search in the neighborhood of a cell has constant complexity independent of the type of pyramid. All processes using only the contents of adjacent cells are **local**. In most cases the computation of a pyramidal cell uses only data from its parents or from its children. Hence all processes at one level may work simultaneously.

Both reduction and refinement propagate data up and down in the pyramid. Hence their computational complexity depends only on the number of levels in the pyramid, e.g. on the $\log(image - diameter)$.

This computational efficiency may even affect sequential processing: It can be shown that the recursive computation of the n'th level of any overlapping pyramid needs less operations than direct computation by equivalent weighting functions. Burt [3] introduced the 'equivalent weighting function': *"Iterative pyramid generation is equivalent to convolving the image g_0 with a set of 'equivalent weighting functions' h_l:"* $g_l = h_l * g_0 = h * g_{l-1}, l > 1$. It allowed him to study the effects of iterated reduction (e.g. the low-pass character of Gaussian pyramids) using the single parameter h_l without giving up the efficient iterative computation. The following example shows on the right side the weighting function that computes the second level of a $3 \times 3/4$ Gaussian pyramid (weights on the left side) directly from the base:

$$\left(\frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \right)^2 \equiv \frac{1}{256} \begin{pmatrix} 1 & 2 & 3 & 4 & 3 & 2 & 1 \\ 2 & 4 & 6 & 8 & 6 & 4 & 2 \\ 3 & 6 & 9 & 12 & 9 & 6 & 3 \\ 4 & 8 & 12 & 16 & 12 & 8 & 4 \\ 3 & 6 & 9 & 12 & 9 & 6 & 3 \\ 2 & 4 & 6 & 8 & 6 & 4 & 2 \\ 1 & 2 & 3 & 4 & 3 & 2 & 1 \end{pmatrix} \quad (2)$$

Note that the resulting pyramid is of type $7 \times 7/16$.

3 From pixels to graphs

Fig. 2 illustrates how a pixel grid can be represented by a neighborhood graph $G(V, E)$ and its dual face graph $\overline{G}(F, \overline{E})$. Vertices V correspond to pixels, geometrical properties like coordinates or gray value are stored as attributes. Two vertices are joined by an edge if the corresponding pixels are neighbors[2]. The faces of the dual graph are placed inside the square formed by 4 vertices. Two faces are joined by a dual edge if the corresponding squares are adjacent in G. There is a one-to-one relationship between E and $\overline{E}$. This graph representation extends the scope to any (planar) tesselation of the image plane.

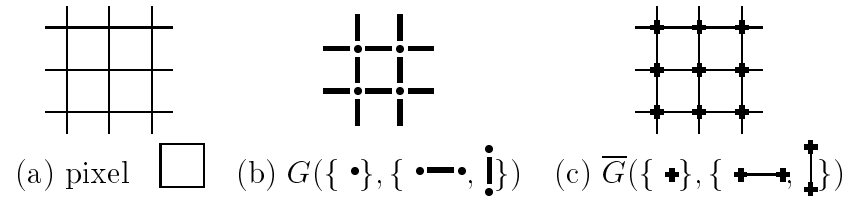


Figure 2: (a) pixel grid, (b) neighborhood graph $G(V, E)$, (c) dual face graph $\overline{G}(F, \overline{E})$

3.1 Computational Complexity

We now generalize the computations of a reduction by a transformation $\tau : X \mapsto X'$ that reduces the data size by a factor $\lambda > 1$

$$|X'| < |X|/\lambda \tag{3}$$

while the reduced data set X' preserves some image properties to compute:

$$\psi(X') = \psi(X) \tag{4}$$

Such properties may be the range of grey values, the average grey value, but also the connectivity of regions or their convexity. The application of τ can be repeated on the reduced data set X' until any further reduction would destroy property ψ. Assume that $X^{(n)} = \tau^n(X)$ is the result after n repetitions. From (3) follows that the overall size reduction is bounded by $\lambda^n < \frac{|X|}{|X^{(n)}|}$ and also the number n of repetitions:

$$n < \frac{\log |X| - \log |X^{(n)}|}{\log \lambda} \tag{5}$$

Together with constraints (3) and (4) and the assumption that τ needs only local parallel computations $\psi(X) = \psi(X^{(n)})$ can be computed in $\mathcal{O}(\log |X|)$ parallel steps.

Classical regular pyramids [7] possess property (3), but, in general, counter examples for property (4) can be constructed [2]. It has been demonstrated [12] that adaptive pyramids in which transformation τ depends on the data X overcome the problem. The price to pay is the loss of regularity.

[2] 4-neighbors are used because they make G planar.

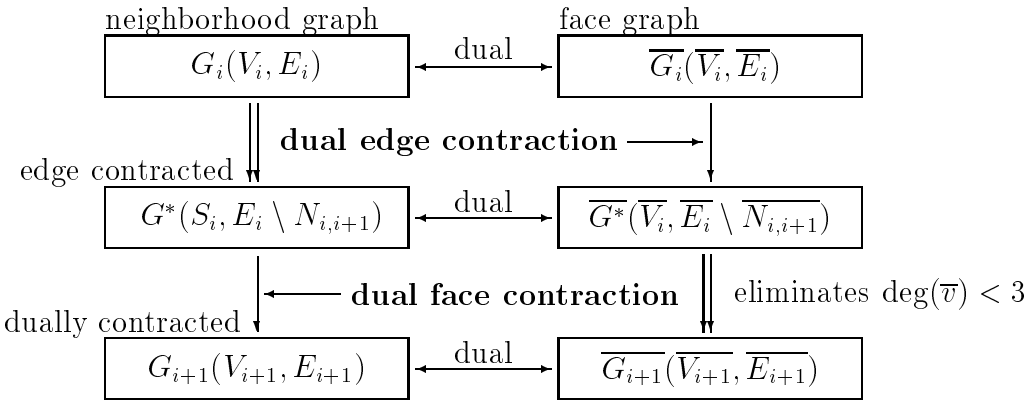


Figure 3: Dual Graph Contraction: $(G_{i+1}, \overline{G_{i+1}}) = C[(G_i, \overline{G_i}), (S_i, N_{i,i+1})]$

4 Dual Graph Contraction

Irregular (or graph) pyramids are constructed bottom-up level by level by repeatedly contracting the image graph in the base: Dual graph contraction proceeds in two basic steps (Fig. 3): dual edge contraction and dual face contraction. The base of the pyramid consists of the pair of dual image graphs $(G_0, \overline{G_0})$. Following *decimation parameters* $(S_i, N_{i,i+1})$ determine the structure of an irregular pyramid [11][Def.5]: a subset of *surviving vertices* $S_i = V_{i+1} \subset V_i$, and a subset of *primary non-surviving edges*[3] $N_{i,i+1} \subset E_i$. Every non-surviving vertex, $v \in V_i \setminus S_i$, must be connected to one surviving vertex in a unique way. The relation between the two pairs of dual graphs, $(G_i, \overline{G_i})$ and $(G_{i+1}, \overline{G_{i+1}})$, as established by dual graph contraction with decimation parameters $(S_i, N_{i,i+1})$ is expressed by function $C[.,.]$:

$$(G_{i+1}, \overline{G_{i+1}}) = C[(G_i, \overline{G_i}), (S_i, N_{i,i+1})] \tag{6}$$

The contraction of a primary non-surviving edge consists in the identification of its endpoints and in the removal of both the contracted edge and its dual edge. Fig. 4 shows the normal situation (a), the situation where the dual edge contraction creates multiple edges (b) and self-loops (c). Redundancies (lower parts) in cases (b) and (c) are decided through the corresponding dual graphs and removed by dual face contraction[4]). Dual face contraction simplifies most of the multiple edges and self-loops, but not those inclosing any surviving parts of the graph (see [11]). Two steps of dual graph contraction shows the example of Fig. 5. They can be formally written as $(G_1, \overline{G_1}) = C[(G_0, \overline{G_0}), (S_0, N_{0,1})]$, and $(G_2, \overline{G_2}) = C[(G_1, \overline{G_1}), (S_1, N_{1,2})]$. Note that graph G_2 in this example contains both a self-loop and a double edge. [11] compares three different types of graph contractions.

4.1 Contraction kernels

To define the parameters that control the process of dual graph contraction we observe that the subgraphs in our example graph (Fig. 5d, e, f, levels $i = 0, 1, 2$ resp.). form small tree structures $T(s)$ that collaps into surviving vertex s of the contracted graph. $T(s)$ is

[3] Secondary non-surviving edges are removed during dual face contraction.

[4] In figures, $S_i = \{\bullet\}$, $\overline{V_{i+1}} = \{\blacksquare\}$, $V_i \setminus S_i = \{\circ\}$, $\overline{V_i} \setminus \overline{V_{i+1}} = \{\square\}$ and $(\bullet, \circ) \in N_{i,j}$ are indicated by $\longrightarrow$.

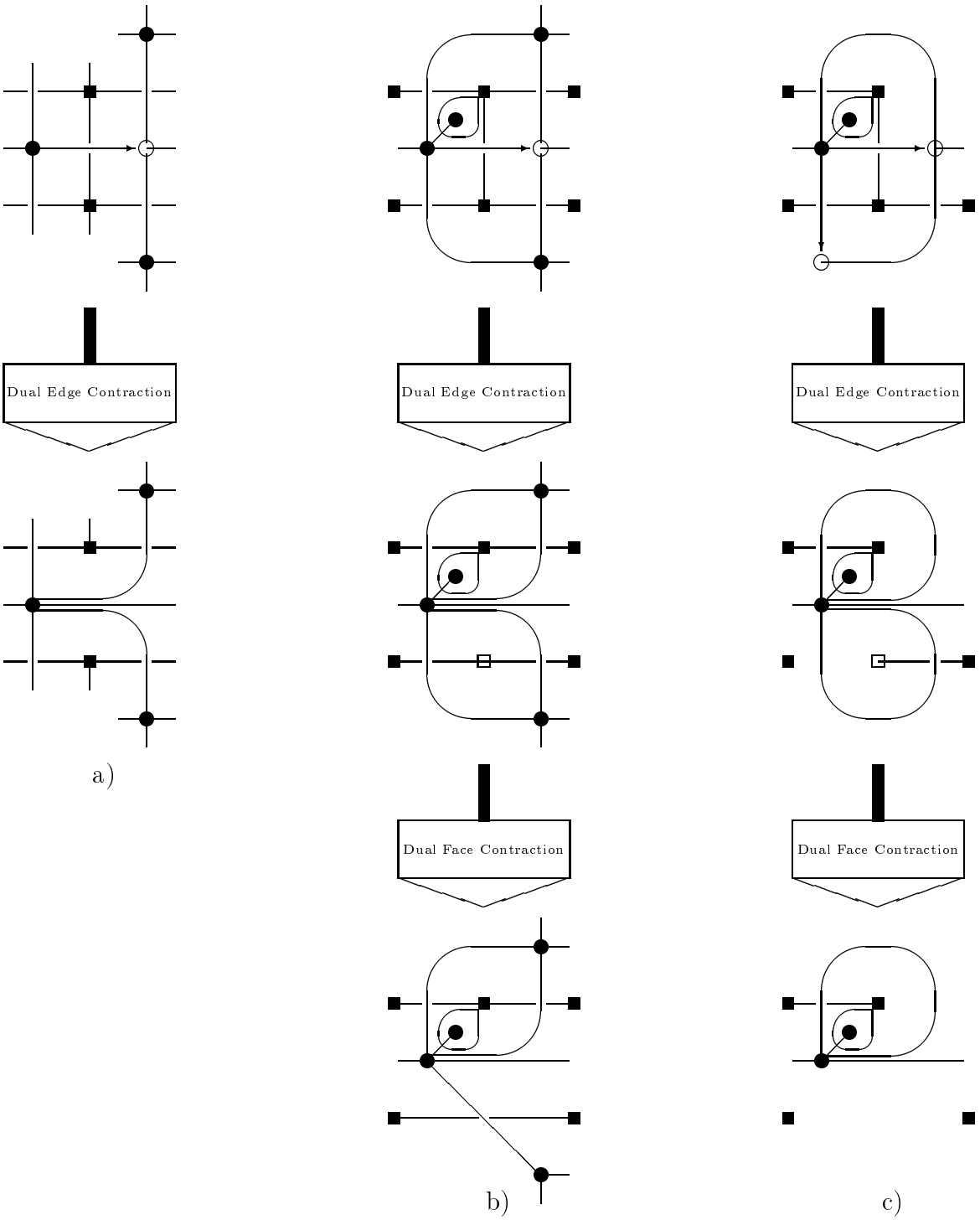
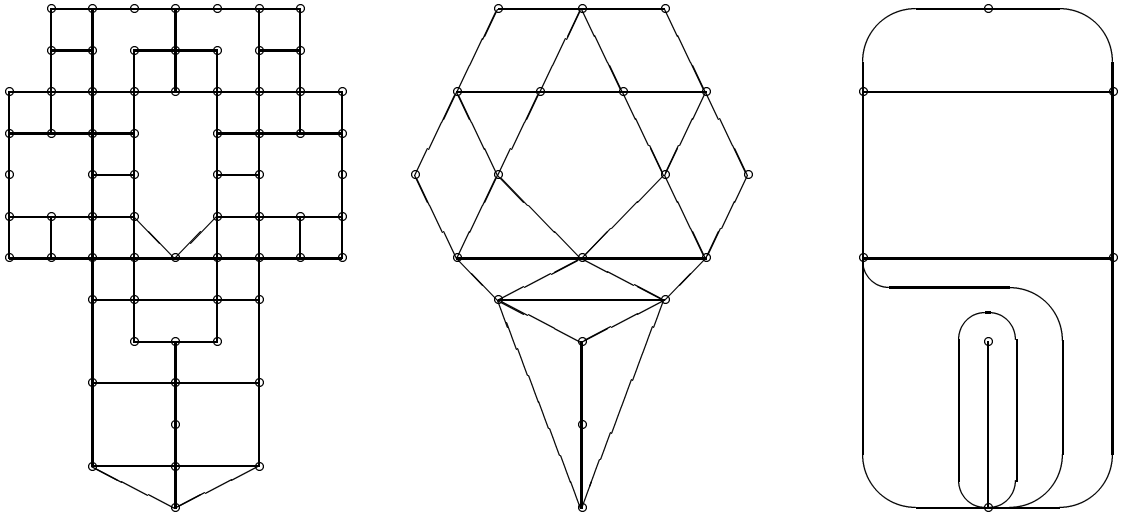
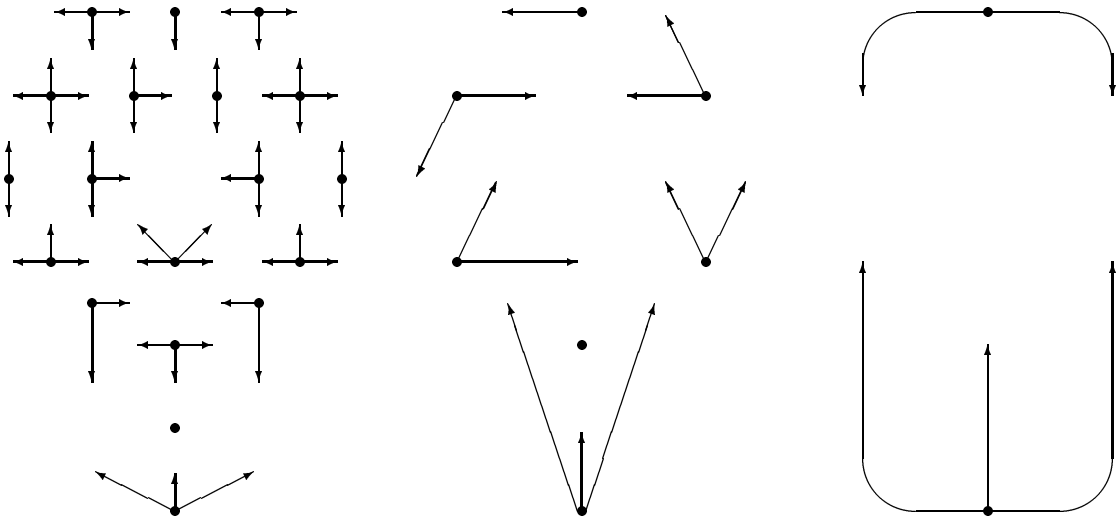


Figure 4: Three Cases of Dual Graph Contraction



(a) (V_0, E_0) (b) (V_1, E_1) (c) (V_2, E_2)



(d) $(S_0, N_{0,1})$ (e) $(S_1, N_{1,2})$ (f) $(S_2, N_{2,3})$

Figure 5: Example of a dual irregular pyramid and decimation parameters

a *spanning tree* of the connected component of the surviving root vertex, or equivalently, (V, N) is a spanning forest of graph $G(V, E)$.

Definition 1 *A decimation of a graph $G(V, E)$ is specified by a selection of* surviving *vertices $S \subset V$ and a selection of* primary non-surviving edges *$N \subset E$ such that following two conditions are fulfilled:*

1. *Graph (V, N) is a spanning forest of graph $G(V, E)$.*
2. *The surviving vertices $S \subset V$ are the roots of the forest (V, N).*

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

The connectivity structure of the contracted graph is established by paths connecting two surviving vertices:

Definition 2 *Let $G(V, E)$ be a graph with decimation parameters (S, N). A path in $G(V, E)$ is called a* connecting path *between two surviving vertices $v, w \in S$, denoted $CP(v, w)$, if it consists of three subsets of edges E (Fig. 6):*

1. *The first part is a possibly empty branch of contraction kernel $T(v)$.*
2. *The middle part is an edge $e \in E \setminus N$ that bridges the gap between the two contraction kernels $T(v)$ and $T(w)$. We call e the* bridge *of the connecting path $CP(v, w)$.*
3. *The third part is a possibly empty branch of contraction kernel $T(w)$.*

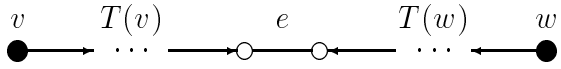


Figure 6: Decomposition of connecting path $CP(v, w)$

Connecting paths $CP_k(v, w)$ in $G_k(V_k, E_k)$ are strongly related to the edges $e_{k+1} = (v, w) \in E_{k+1}$ in the contracted graph $G_{k+1}(V_{k+1}, E_{k+1})$: Two different surviving vertices that are connected by a connecting path in G_k are connected by an edge in E_{k+1}. For every edge $e' = (v, w) \in E_{k+1}$ there exists a connecting path $CP_k(v, w)$ in G_k. Dual edge contraction can be implemented by (1) simply renaming all the non-surviving vertices to their surviving parent vertex, (2) deleting all non-surviving edges N and (3) their duals $\overline{N}$.

4.2 Equivalent contraction kernels

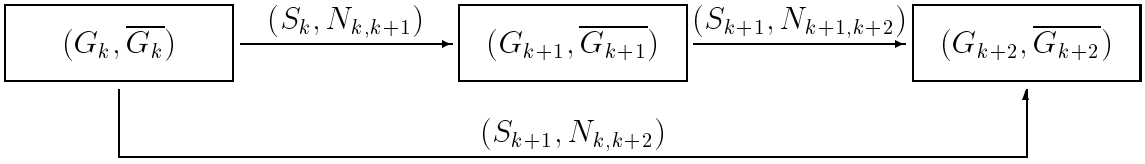
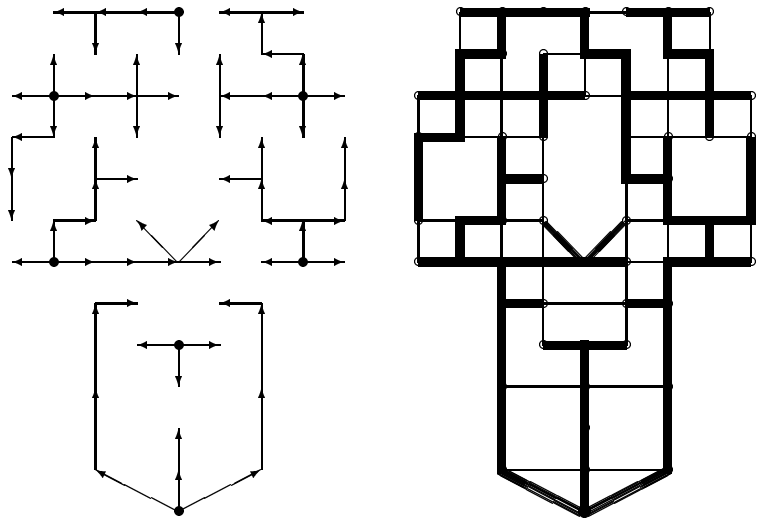


Figure 7: Equivalent contraction kernel

The combination of two (and more) successive reductions in an equivalent weighting function allowed Burt to calculate any level of the pyramid directly from the base. Similarly we combine two (and more) dual graph contractions (see Fig. 7) of graph G_k with decimation parameters $(S_k, N_{k,k+1})$ and $(S_{k+1}, N_{k+1,k+2})$ into one single **equivalent contraction kernel** (ECK) $N_{k,k+2} = N_{k,k+1} \circ N_{k+1,k+2}$ (for simplicity G_i stands for $(G_i, \overline{G_i})$):

$$\begin{aligned} C[C[G_k, (S_k, N_{k,k+1})], (S_{k+1}, N_{k+1,k+2})] &= C[G_k, (S_{k+1}, N_{k,k+2})] \\ &= G_{k+2} \end{aligned} \tag{7}$$

Equivalent contraction kernels (see example in Fig. 8) are constructed in the following way:



(a) (b)

Figure 8: Example of equivalent contraction kernels: (a) $(S_1, N_{0,2}) = (S_0, N_{0,1}) \circ (S_1, N_{1,2})$, (b)of the apex: $G_0 \cup N_{0,4}$

1. Assume that the dual irregular pyramid $((G_0, \overline{G_0}), (G_1, \overline{G_1}), \ldots, (G_{k+2}, \overline{G_{k+2}}))$ is the result of $k+2$ dual graph contractions. The structure of G_{k+2} is fully determined by the structure of G_{k+1} and the decimation parameters $(S_{k+1}, N_{k+1,k+2})$.

2. Furthermore, the structure of G_{k+1} is determined by G_k and the decimation parameters $(S_k, N_{k,k+1})$. $S_{k+1} = V_{k+2}$ are the vertices surviving from G_k to G_{k+2}. The searched contraction kernels must be formed by edges $N_{k,k+2} \subset E_k$. This is true for $N_{k,k+1}$ but not for $N_{k+1,k+2} \subset E_{k+1}$ if we would simply overlay the two sets of decimation parameters. An edge $e_{k+1} = (v_{k+1}, w_{k+1}) \in N_{k+1,k+2}$ corresponds to a connecting path[5] $CP_k(v_{k+1}, w_{k+1})$ in G_k. By definition 2, $CP_k(v_{k+1}, w_{k+1})$ consists of one branch of $T_k(v_{k+1})$, one branch of $T_k(w_{k+1})$, and one surviving edge $e_k \in E_k$ connecting the two contraction kernels $T_k(v_{k+1}), T_k(w_{k+1})$.

[5]If there are more than one connecting paths, one must be selected.

Definition 3 *Function* **bridge**: $E_{k+1} \mapsto E_k$ *assigns to each edge* $e_{k+1} = (v_{k+1}, w_{k+1}) \in E_{k+1}$ *one of the bridges* $e_k \in E_k$ *of the connecting paths* $CP(v_{k+1}, w_{k+1})$*:*

$$bridge(e_{k+1}) := e_k. \tag{8}$$

Two disjoint tree structures connected by a single edge become a new tree structure. The result of connecting all contraction kernels T_k by bridges fulfills the requirements of a contraction kernel:

$$N_{k,k+2} := N_{k,k+1} \quad \cup \bigcup_{e_{k+1} \in N_{k+1,k+2}} \text{bridge}(e_{k+1}) \tag{9}$$

The contraction kernels $(V_2, N_{0,2})$ in Fig. 8a are equivalent to the successive contraction with kernels of Fig. 5d and e.

3. The above process can be repeated on the remaining contraction kernels until the base level 0 contracts in one step into the apex $V_n = \{v_n\}$. The edges of the corresponding spanning tree are contained in $N_{0,n}$. Fig. 8 shows spanning tree $N_{0,4}$ overlaid with the base graph G_0.

5 Conclusion

Rosenfeld [19] exposed the properties of two "variable-resolution representations" in 1982: pyramids and quadtrees. Since then different reduction functions have greatly enhanced the computational power of pyramids. However in the pure bottom-up processing mode, **pyramids do not compute anything new**, which could not be computed from the cells of the receptive field. *Equivalent weighting functions* allow the computation of any pyramid level directly from the base. The same role play the *equivalent contraction kernels* for irregular pyramids. Why then use pyramids any more?

- Pyramids decompose an often very complex computation into a few relatively simple processing steps.
- Global operators decompose into a few local operators which can be applied in parallel.
- Pyramids are computationally extremely efficient.
- The limitation of a cell's storage capacity enforces abstraction, less important data are neglected in higher levels.
- Exterior knowledge sources can be integrated at different levels of abstraction. It could define the language of interpretation.
- Irregular pyramids can adapt their structure to the data rather than the parameters of the underlying models.
- The graph structures of irregular pyramids provide a smooth transition to mid- and high-level vision.

- Decimation parameters control dual graph contraction, a process that iteratively builds an irregular (graph) pyramid. The concept of contraction kernel preserves the graph's structural properties, its connectivity, its planarity, and the face degrees of its dual graph.
- Equivalent contraction kernels (ECKs) allow to skip the construction of intermediate pyramid levels. The contents of aggregations of cells can be computed efficiently and in parallel through the tree structure of the contraction kernels. The ECK of the apex is a spanning tree of the base graph.

References

[1] Peter K. Ahnelt and Dietmar Pum. The foveal photoreceptors. a model for the comparison of physiologic and algorithmic image compressions. In Peter Mandl, editor, *Modelling and New Methods in Image Processing and in Geographical Information Systems*, OCG-Schriftenreihe, Österr. Arbeitsgemeinschaft für Mustererkennung, pages 223–232. Oldenbourg, 1991. Band 61.

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

[3] 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.

[4] 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.

[5] Robert M. Haralick and Linda G. Shapiro. Glossary of Computer Vision Terms. *Pattern Recognition*, Vol. 24(No. 1):pp. 69–93, 1991.

[6] 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.

[7] Jean-Michel Jolion and Azriel Rosenfeld. *A Pyramid Framework for Early Vision.* Kluwer Academic Publishers, 1994.

[8] 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.

[9] Walter G. Kropatsch. Image Pyramids and Curves - An Overview. Technical Report PRIP-TR-2, Institute f. Automation 183/2, Dept. for Pattern Recognition and Image Processing, TU Wien, Austria, 1991.

[10] Walter G. Kropatsch. Building Irregular Pyramids by Dual Graph Contraction. Technical Report PRIP-TR-35, Institute f. Automation 183/2, Dept. for Pattern Recognition and Image Processing, TU Wien, Austria, 1994. Also available through http://www.prip.tuwien.ac.at/ftp/pub/publications/trs/tr35.ps.gz.

[11] Walter G. Kropatsch. Building Irregular Pyramids by Dual Graph Contraction. *IEE-Proc. Vision, Image and Signal Processing*, Vol. 142(No. 6):pp. 366–374, December 1995.

[12] Walter G. Kropatsch and Souheil BenYacoub. A revision of pyramid segmentation. In Walter G. Kropatsch, editor, *13th International Conference on Pattern Recognition*, volume II, pages 477–481. IEEE Comp.Soc., 1996.

[13] Walter G. Kropatsch and Herwig 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.

[14] Herwig Macho and Walter G. Kropatsch. Finding Connected Components with Dual Irregular Pyramids. In Franc Solina and Walter G. Kropatsch, editors, *Visual Modules, Proc. of 19th ÖAGM and 1st SDVR Workshop*, pages 313–321. OCG-Schriftenreihe, Österr. Arbeitsgemeinschaft für Mustererkennung, R. Oldenburg, 1995. Band 81.

[15] 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.

[16] Christophe Mathieu, Isabelle E. Magnin, and C. Baldy-Porcher. Optimal stochastic pyramid: segmentation of MRI data. *Proc. Med. Imaging VI: Image Processing*, SPIE Vol.1652:pp.14–22, Feb. 1992.

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

[18] Marvin Lee Minsky and Seymour A. Papert. *Perceptrons: An Introduction to computational geometry*. The MIT Press, expanded edition edition, 1988.

[19] Azriel Rosenfeld. Quadtrees and Pyramids: Hierarchical representation of images. Technical Report TR-1171, University of Maryland, Computer Science Center, May 1982.

[20] Azriel Rosenfeld and A. C. Kak. *Digital Picture Processing*, volume Vol. 1 and 2. Academic Press, New York, second edition, 1982.