Paths Lengths in Stochastic Graph Image Pyramid[1)]

Yll Haxhimusa and Walter G. Kropatsch
Pattern Recognition and Image Processing Group 183/2
Institute for Computer Aided Automation
Vienna University of Technology
Favoritenstr. 9, A-1040 Vienna, Austria
{yll, krw}@prip.tuwien.ac.at

Abstract:
In many applications some parts of an image are of special interest. We present in this paper results of the analysis of the vertical path lengths in stochastic graph image pyramids. Such path lengths evaluate the efficiency of the pyramid structure needed e.g. for algorithms which derive object properties from the object pixels in the image. Our aim is to build stochastic image pyramid locally that are optimal in the sense of bottom-up and top-down processes.

1 Introduction

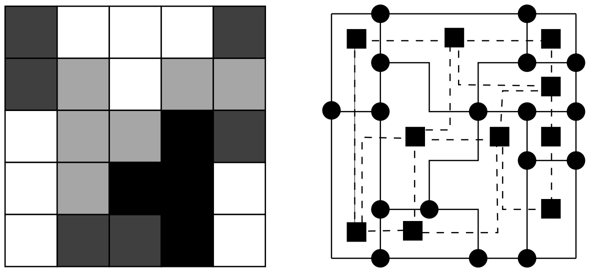
In a regular image pyramid (for an overview see [16]) the number of pixels at any level l, is r times higher than the number of pixels at the next reduced level $l + 1$. The so called reduction factor r is greater than one and it is the same for all levels l, moreover the reduction window is also of the same size. If s denotes the number of pixels in an image I, the number of new levels on top of I amounts to $log_r(s)$. Thus, the regular image pyramid may be an efficient structure to access image objects in a top-down process.

However, regular image pyramids are confined to globally defined sampling grids and lack shift invariance [2]. In [18, 12] it was shown how these drawbacks can be avoided by irregular image pyramids, where the data partition the image into connected regions each of which is contracted to a single vertex. Since data do not impose any particular internal structure of these regions there are differences in efficiency. Irregular pyramids can perform most of the operations their regular counterparts are employed for [20].

The construction of an irregular image pyramid is iteratively local [17, 11, 10, 1]:

- the cells have no information about their global position.

[1)]This paper has been supported by the Austrian Science Fund under grants P14445-MAT and P14662-INF



(a) **Graylevel image** (b) **Dual graphs** $(G, \overline{G})$

Figure 1: Partition of pixel set into cells and representation of the cells and their neighborhood relations.

- the cells are connected only to (direct) neighbors.
- the cells cannot distinguish the spatial positions of the neighbors.

In some applications some parts of the image are of special interest, so we need to access data in a top-down process very often [6]. Our aim is to build stochastic image pyramids locally that are optimal in the sense of bottom-up and top-down processes.

We restrict ourselves to irregular stochastic image pyramids with an apex, i.e. the top of the pyramid, level h contains only one cell. We represent the levels as dual pairs $(G_l, \overline{G_l})$ of plane graphs G_l and $\overline{G_l}$ [14]. Each level represents a partition of the pixel set into cells, i.e. connected subsets of pixels. On the base level (level 0) of an irregular image pyramid the cells represent single pixels and the neighborhood of the cells is defined by the 4-connectivity of the pixels. A cell on level $l+1$ (parent) is a union of neighboring cells on level l (children), Fig. 2a. This union is controlled by so called *contraction kernels* (CK). Every parent computes its values independently of other cells on the same level. This lead to the property that an image pyramid is built in $O[log(image\_diameter)]$ time. For more in depth on the subject see the book of Jolion [13] and of Rosenfeld [19].

Neighborhoods on level $l+1$ are derived from neighborhoods on level l. Two cells c_1 and c_2 are neighbors if there exist pixels p_1 in c_1 and p_2 in c_2 such that p_1 and p_2 are 4-neighbors. We assume that on each level $l+1$ ($l \geq 0$) there exists at least one cell not contained in level l. In particular, there exists a highest level h . Furthermore, we restrict ourselves to irregular pyramids with an apex, i.e. level h contains only one cell.

A level consists of dual pair $(G_l, \overline{G_l})$ of plane graphs G_l and $\overline{G_l}$, Fig. 1b. The planarity of graphs restricts us in using only the 4-connectivity of the pixels. The vertices of G_l represent the cells on level l and the edges of G_l represent the neighborhood relations of the cells on level l, depicted with square vertices and dashed edges in Fig. 1b. The edges of $\overline{G_l}$ represent the borders of the cells on level l, solid lines in Fig. 1b, possibly including pseudo edges needed to represent neighborhood relations to a cell completely enclosed by another cell [14]. Finally, the vertices of $\overline{G_l}$, circles in Fig. 1b, represent meeting points of at least three boundary segments of $\overline{G_l}$, solid lines in Fig. 1b.

The sequence $(G_l, \overline{G_l})$, $0 \leq l \leq h$ is called (dual) graph pyramid (Fig. 2c).

The plan of this paper is as follows. In Section 2 we will present shortly the algorithms used and in Section 3 we will define the path lengths. Experimental results are given in Section 4.

2 Building Image Pyramids

In the following the iterated local construction of the stochastic irregular image pyramid in [17] and [9] is described shortly in the language of graph pyramids. We use these algorithms in our tests.

The main idea of Meer's stochastic decimation algorithm [17] is to first calculate a so called *maximal independent vertex set* (**MIS**) [4]. There are two conditions to be fulfilled in order to have MIS:

- two surviving vertices cannot be neighbors of each other, and
- every non-surviving vertex has in its neighborhood a surviving vertex.

See [17, 11, 8] for details how to build MIS.

MIS is built as follows: All the vertices of the graph $G_l(V_l, E_l)$ are marked as *candidate* of MIS. After which a random number is assigned to every vertex of V_l. A vertex is marked as *member* (of MIS) if it is a local maximum in its neighborhood[1)], i.e. having the largest random number. The *member* and all its neighboring vertices are marked as *non-candidates*. In general some iteration for correction must be done to complete the maximal independent vertex set (three iterations in [17, 9]). The *members* of MIS determine the vertices that will survive in the next level.

The assignment of the non-survivors to their survivors determines a collection of *contraction kernels* $\mathcal{C}$ (CK): each non-survivor is contracted towards its survivor (e.g. the one with the largest random number) and all contractions can be done in a single parallel step . This collection of CKs are used by dual graph contraction (DGC) algorithm [14] to build the next level of the graph pyramid. The CKs are required to form spanning forests of maximally depth two. Equivalent contraction kernels (ECKs) combine two or more CKs into a single CK which generates the same results in one single dual contraction. Contraction parameters of any individual pyramid level can be reconstructed from ECK of the pyramid's apex [15]. ECK is a rooted spanning tree. Because the constant reduction factor in MIS cannot be guaranteed, a new method is proposed in [7, 9] to overcome this problem.

The main difference of the data driven decimation process (**D3P**) proposed in [11] w.r.t to the MIS algorithm is that no iteration for correction is performed. A vertex in G_l survives if it is a local maximum or if it does not have any survivors in its neighborhood. Leaving out the iteration is motivated by the fact that the iteration is used only for completing the maximal independent set. It

[1)]The neighborhood $\Gamma_l(v)$ of a vertex $v \in V_l$ is defined by $\Gamma_l(v) = \{v\} \cup \{\{w\} \in V_l \mid \exists e = (v, w) \in E_l\}$.

is assumed that being a local maximum is of importance [11]. As for the MIS also the D3P cannot guarantee a constant reduction factor [9].

Maximal independent edge set algorithm (**MIES**) [8] aims at a collection $\mathcal{C}$ of CK in a plane graph G_l such that each vertex of G_l is contained in exactly one kernel of $\mathcal{C}$, and each kernel $\mathcal{C}$ contains at least two vertices. We assume that G_l is connected and this is preserved by the DGC algorithm. This is achieved by transforming the graph G to the line graph[2)] $\underline{G}$ and applying the MIS algorithm on $\underline{G}$. After this we eliminate all the isolated surviving vertices (CKs of depth zero) by augmenting them with one of the CK in the neighborhood. If the resulting CK is of depth three we split it into two CKs of depth one. Clearly, the contraction of all kernels in $\mathcal{C}$ will reduce the number of vertices to half or less, i.e. the constant reduction of at least 2 can be guaranteed. The MIES algorithm is only applicable where there are no constraints on direction of contraction [7, 9].

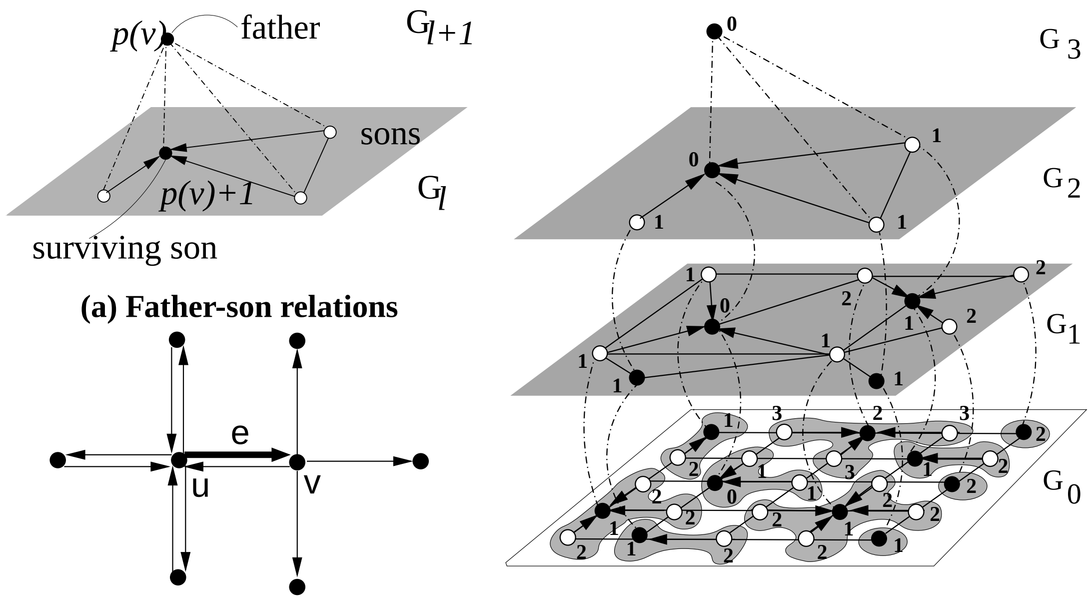
In many graph pyramid applications such as line image analysis [3] and the description of image structure [5] a directed edge e with source u and target $v \neq u$ must be contracted (from u to v), only if the attributes of e, u, and v fulfill a certain condition. This means that the graph G is directed. The edges that fulfill the condition are called *preselected* edges, and only these edges are considered as candidates for contraction. On this set of preselected edges a maximal independent directed edge set is found (**MIDES**) analogously to MIS. All edges of G are marked as *candidate*. The edge with the largest random number in its neighborhood (edge e in Fig. 2b) is marked as *member* and as *non-candidate*. All edges in the neighborhood of the *member* are marked as *non-candidate* (all edges shown in Fig. 2b, except e) . Some iteration are needed to mark all the edges as *non-candidate*. The collection of the *member* edges determine the CKs. After which we can proceed with the DGC to build the graph pyramid. This algorithm showed the best reduction factor [9, 8].

3 Path Lengths in Image Pyramids

Path length shows steps to bring an attribute from every vertex in the base (bottom-up) to the top of the pyramid, and from a top to a vertex in the base (top-down). We call this vertical path length. This information is important for quickly analyzing distinct objects in a top-down/bottom-up process. We are not interested in comparing the vertex complexity and the height of the pyramid, but to be able to compare the internal structure of image pyramids of the same height, i.e. structural complexity.

In the following we define the vertical path lengths and the way they are determined. First we build a graph pyramid bottom-up using one of the algorithms MIS, MIES, MIDES or D3P to find CKs (decimation parameters). Graphs are reduced using the dual graph contraction algorithm [14]. Since we are building stochastic image pyramids the top of a pyramid is always a single vertex.

[2)] The vertices of G becomes edges in $\underline{G}$ and vice versa.



(a) Father-son relations

(b) Neighborhood $N(e)$

(c) Path lengths

Figure 2: Oriented Graph Pyramid built using DGC.

Fig. 2a shows a rooted tree on level G_l (the sons) and their relation (dashed lines) with the vertex on G_{l+1} (the father); white vertices on G_l are the non-surviving sons, and they are contracted (arrows) towards the surviving son, depicted with black.

Vertical paths connect the apex with the base of the pyramid following *father-son* relations from level to level. Path lengths $p \in \mathbb{N}$ of vertices at G_0 (level 0) can be found in a top-down process as follows:

Path Length Algorithm:

1. Let the vertex $v \in G_h$ at the top of the pyramid have path length 0, $p(v) = 0$.

2. Iterate until base level, $l = h - 1, .., 0$.

   $\forall v \in G_{l+1}$: downpropagate the path length $p(v)$ of the vertices of the vertex v at level $l+1$ to its surviving son ss at level l below, so the path length of ss is $p(ss) = p(v)$. All non-surviving sons $ns \in V_l$ of v at level l have path length $p(ns) = p(v) + 1$.

An example of vertical path is shown in Fig. 2c. We start at the top of the pyramid. For stochastic image pyramid there is only one vertex at the top of the pyramid, vertex h at the top has path length 0. An one-to-one relationship between sons (vertices at l) and fathers (vertices at $l+1$) is created during construction of the image pyramid [14].

The number of vertices $|V_{l+1}|$ in level $l+1$ is the same as the number of surviving vertices in $|V_l|$ (surviving sons) in level l. Costs for inheritance from fathers to surviving sons is kept zero since it involves a simple copy of the attributes. Costs for contracting an edge are set to 1 since reduction involves the merging operation for the attributes of the two end vertices. This means that the surviving son will have the path length of the father and non-surviving sons the incremented path length of the father by one. In Fig. 2c at G_2 a vertex has the same length as its father (black vertex with path length 0), and all non-surviving sons (white vertices) have the path length $0 + 1 = 1$. To arrive to the top of the pyramid from the edge with path length 3, three edges must be contracted. Note that there is a vertex in the base level, the representative of the apex, which has a 0 path length.

The path length shows costs to arrive to the top of the pyramid from every vertex in the base. This information is important for quickly analyzing distinct objects in a top-down process.

4 Experimental Results and Discussion

Uniformly distributed random numbers are assigned to the vertices in the base level grid graphs. By changing the seed of the uniformly distributed random generator we generated 100 graphs, on top of which we built stochastic graph pyramids using one of the algorithms MIS, MIES, MIDES or D3P to compute the path lengths. We contract these graphs using DGC [14] until we reach on top of the pyramid. In our experiments we used grid graphs of size 10000 and 40000 vertices respectively, which correspond to image sizes of 100×100 and 200×200 pixels.

The result of the mean value of number of vertices per path length over 100 pyramids are given in Fig. 3, (a) for 100×100 and (b) 200×200 image size. The two diagrams of Fig. 3 show on the x-axis in a logarithmic scale the length of the vertical paths and on the y-axis the number of vertices of the base level. Each base vertex has a certain 'vertical distance' to the apex, the number of vertices having the same vertical distance can be accumulated in the histogram of vertical path lengths. Every pyramid generates such a histogram using the **path length algorithm**. Histograms generated by a particular selection strategy can be averaged and are shown for the MIS, MIES, MIDES and D3P strategies in Fig.3.

Path lengths i.e. costs for MIES and MIDES are smaller, even when the image size were 4 times larger. MIS and D3P have tendency to have longer path lengths. The Table 1 shows the maximal path length and in brackets the most frequent path length. For MIES and MIDES almost 50% of vertices have path length of 6, 7, and 8 for 100×100, and 7, 8, and 9 for 200×200 image, respectively. These values are comparable with the $log(diameter)$, which would be the height of the regular pyramid, a property we are trying to achieve.

To summarize, MIS and D3P path lengths are longer in both cases. MIES and MIDES tend to find shorter path length. Longer path lengths imply bigger costs to access a vertex in the base from the apex of the pyramid.

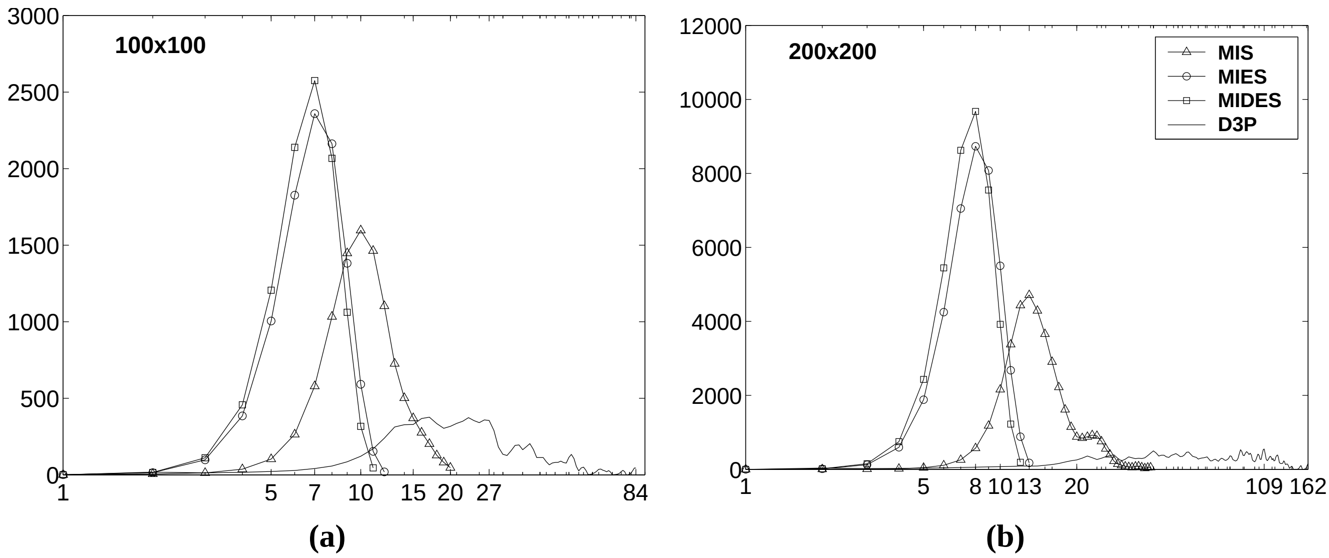


Figure 3: μ of path lengths. x axis the path lengths, and y-axis number of vertices.

	Image size	
	100×100	200×200
MIS	20 (10)	39 (13)
MIES	12 (7)	13 (8)
MIDES	11 (7)	12 (8)
D3P	84 (27)	160 (109)
diameter	200	400
log(diameter)	8	9

Table 1: The maximum path length.

5 Conclusion

The path length algorithm measures the structural complexity in irregular image pyramids and gives costs to arrive to the top of the pyramid (vertical path length) from every vertex in the base. The experiment showed that MIES and MIDES have shown shorter path lengths and better reduction factors [9, 8]. These algorithms are used to build stochastic image pyramids optimally, i.e. faster access to vertices at the base level from the top and with the smallest height and vice versa. This information is important for quickly analyzing distinct objects in an iterated top-down/bottom-up process.

References

[1] Horst Bischof and Walter G. Kropatsch. Hopfield networks for irregular decimation. In Wolfgang Pölzleitner and Emanuel Wenger, editors, *Image Analysis and Synthesis*, pages 317–327. OCG-Schriftenreihe, Österr. Arbeitsgemeinschaft für Mustererkennung, R. Oldenburg, 1993. Band 68.

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

[3] Mark J. Burge and Walter G. Kropatsch. A minimal line property preserving representation of line images. *Computing*, 62:355 – 368, 1999.

[4] N. Christofides. *Graph theory - an algorithmic approach*. Academic Press, New York, 1975.

[5] Roland Glantz, Roman Englert, and Walter G. Kropatsch. Representation of Image Structure by a Pair of Dual Graphs. In Walter G. Kropatsch and Jean-Michel Jolion, editors, *2nd IAPR-TC-15 Workshop on Graph-based Representation*, pages 155–163. OCG-Schriftenreihe, Band 126, Österreichische Computer Gesellschaft, May 1999.

[6] Roland Glantz and Walter G. Kropatsch. Guided Relinking of Graph Pyramids. In Francesc J. Ferri, José M. Iñesta, Adnan Amin, and Pudil Pavel, editors, *Joint IAPR International Workshops on SSPR'2000 and SPR'2000*, volume 1876 of *Lecture Notes in Computer Science*, pages 367–376, Alicante, Spain, August 2000. Springer, Berlin Heidelberg, New York.

[7] Yll Haxhimusa, Roland Glantz, Maamar Saib, Langs, and Walter G. Kropatsch. Logarithmic Tapering Graph Pyramid. In *In the Proceedings of DAGM2002*, Swiss, 2002. To appear.

[8] Yll Haxhimusa, Roland Glantz, Maamar Saib, Langs, and Walter G. Kropatsch. Reduction Factors of Image Pyramid on Udirected and Directed Graph. In *Proceedings of the 7th. Computer Vision Winter Workshop*, 2002.

[9] Yll Haxhimusa and Walter G. Kropatsch. Reduction Factors of Pyramids on Undirected and Directed Graphs. Technical Report No.74, PRIP, Vienna University of Technology, 2002.

[10] Jean-Michel Jolion. Data driven decimation of graphs. In *Proc. 3th IAPR Int. Workshop on Graph based Representation*, pages 105–114, Capri, Italy, 2001.

[11] Jean-Michel Jolion. Stochastic pyramid revisited. *Pattern Recognition Letters*, accepted, 2002.

[12] 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.

[13] J.M. Jolion and Azriel Rosenfeld. *A Pyramid Framework for Early Vision*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1994.

[14] Walter G. Kropatsch. Building Irregular Pyramids by Dual Graph Contraction. *IEE-Proc. Vision, Image and Signal Processing*, 142(6):366 – 374, 1995.

[15] Walter G. Kropatsch. Equivalent contraction kernels to build dual irregular pyramids. *Advances in Computer Science*, Advances in Computer Vision:pp. 99–107, 1997.

[16] Walter G. Kropatsch, Aleš Leonardis, and Horst Bischof. Hierarchical, Adaptive and Robust Methods for Image Understanding. *Surveys on Mathematics for Industry*, No.9:1–47, 1999.

[17] Peter Meer. Stochastic image pyramids. *CVGIP*, 45:269 – 294, 1989.

[18] Annick Montanvert, Peter Meer, and Azriel Rosenfeld. Hierarchical image analysis using irregular tesselations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):pp.307–316, April 1991.

[19] Azriel Rosenfeld, editor. *Multiresolution Image Processing and Analysis*. Springer, Berlin, 1984.

[20] Azriel Rosenfeld. Pyramid algorithm for efficient vision. Technical Report CAR-TR-299, University of Maryland, Computer Science Center, 1987.