Technical Report

Pattern Recognition and Image Processing Group
Institute of Computer Aided Automation
Vienna University of Technology
Favoritenstr. 9/183-2
A-1040 Vienna AUSTRIA
Phone: +43 (1) 58801-18351
Fax: +43 (1) 58801-18392
E-mail: {yll,krw}@prip.tuwien.ac.at
URL: http://www.prip.tuwien.ac.at/

PRIP-TR-74

July 26 , 2002

Reduction Factors of Pyramids on Undirected and Directed Graphs [1]

Yll Haxhimusa and Walter G. Kropatsch

Abstract

We present two new methods to determine contraction kernels for the construction of graph pyramids. The first method is restricted to undirected graphs and yields a reduction factor of at least 2.0. This means with our method the number of vertices in the subgraph induced by any set of contractible edges reduces to half or less by a single parallel contraction. Our second method also works for directed graphs. Our methods yield better reduction factors than Meer's stochastic decimation algorithm, in all tests. The lower bound of the reduction factor becomes crucial with large images. We also give a method to compare the structure of the image pyramid.

Keywords. Graph based image analysis, image graph pyramids, maximal independent vertex set, maximal independent edge set, maximal independent directed edge set, vertical path lenghts.

[1] This work was supported by the Austrian Science Foundation under grant P14445-MAT and P14662-INF.

Contents

1 Introduction

In a regular image pyramid (for an overview see [KLB99]) 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. 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 [BCR90]. In [MMR91, JM92] it was shown how these drawbacks can be avoided by irregular image pyramids, the so called adaptive pyramids. Irregular pyramids can perform most of the operations their regular counterparts are employed for [Ros87]. Each level represents a partition of the pixel set into cells, i.e. connected subsets of pixels. The construction of an irregular image pyramid is iteratively local [Mee89, Jol02, Jol01, BK93]:

- the cells have no information about their global position.
- the cells are connected only to (direct) neighbors.
- the cells cannot distinguish the spatial positions of the neighbors.

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). This union is controlled by so called *contraction kernels*. 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 indepth on the subject see the book of Jolion [JR94] and of Rosenfeld [Ros84].

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 $\overline{G_l}$, Figure 1(b). 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 Figure 1(b). The edges of $\overline{G_l}$ represent the borders of the cells on level l, solid lines in Figure 1(b), possibly including so called pseudo edges needed to represent neighborhood relations to a cell completely enclosed by another cell. Finally, the vertices of $\overline{G_l}$, circles in Figure 1(b), represent meeting points of at least three boundary segments of $\overline{G_l}$, solid lines in Figure 1(b). The sequence $(G_l, \overline{G_l})$, $0 \leq l \leq h$ is called (dual) graph pyramid.

The aim of this technical report is to combine the advantage of regular pyramids (logarithmic tapering) with the advantages of irregular pyramids (their purely local construction, universal segmentation, topology preservation, preservation of face degree etc.). The aim is reached

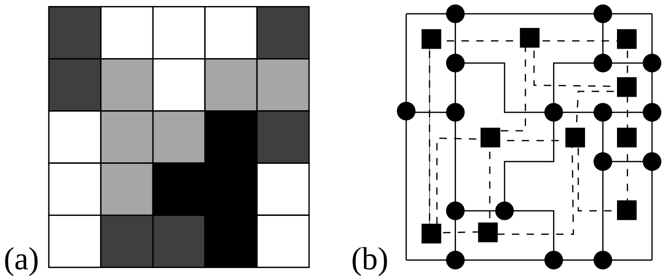


Figure 1: (a) Partition of pixel set into cells. (b) Representation of the cells and their neighborhood relations by a dual pair $(G, \overline{G})$ of plane graphs.

by replacing the selection method for contraction kernels proposed in [Mee89] by two new iteratively local methods. The method in Section 3 that now guarantees a reduction factor of 2.0 and the method in Section 4 . Experiments with selection methods show that:

- the old method ([Mee89]) does not lead to logarithmic tapering graph pyramids, as opposed to our methods, i.e. the reduction factors of graph pyramids build by the old method can get arbitrarily close to 1.0 This problem was encountered also in [MMR91].

- the sizes of the cells from the new method are much more uniform.

Not only stochastic decimation [Mee89] but also connected component analysis [KM95] gains from the new methods. The method in Section 4 turned out to produce logarithmic tapering graph pyramids also in case of monotonic dual graph contraction [GK00b].

The plan of the technical report is as follows. In Section 2 we give the main idea of the stochastic pyramid algorithm and in Section 6.1 we see that graph pyramids from maximal independent vertex sets may have a very poor reduction factor. Moreover, experiments show that small reduction factors are likely, especially when the images are large. We propose two modifications. The one in Section 3 guarantees a reduction factor of 2.0, but is applicable only if the edges may be contracted in both directions. The modification proposed in Section 4 also works in case of constraints on the directions. This modification yields the highest reduction factors in the case of stochastic graph pyramids, in all our tests. In Section 5 we give a short introduction of an idea proposed by Jolion in [Jol02]. An experimental comparison is given in Section 6. In Section 7 we give a method to compare the structure of the graph pyramids.

2 Maximal Independent Vertex Set

In the following the iterated local construction of the (stochastic) irregular image pyramid in [Mee89] is described in the language of graph pyramids. The main idea is to first calculate a so called *maximal independent vertex set* [1] [Chr75, TS92]. Let the vertex set and edge set of G_l be denoted by V_l and E_l, respectively. The incidence relation of V_l, denoted by $\iota_l(\cdot)$ maps each edge from E_l to its set of end vertices.

The neighborhood $\Gamma_l(v)$ of a vertex $v \in V_l$ is defined by

$$\Gamma_l(v) = \{\{v\} \cup \{w\} \in V_l \mid \exists e \in E_l \text{ such that } v, w \in \iota_l(e)\}.$$

A subset W_l of V_l is called maximal independent vertex set if:

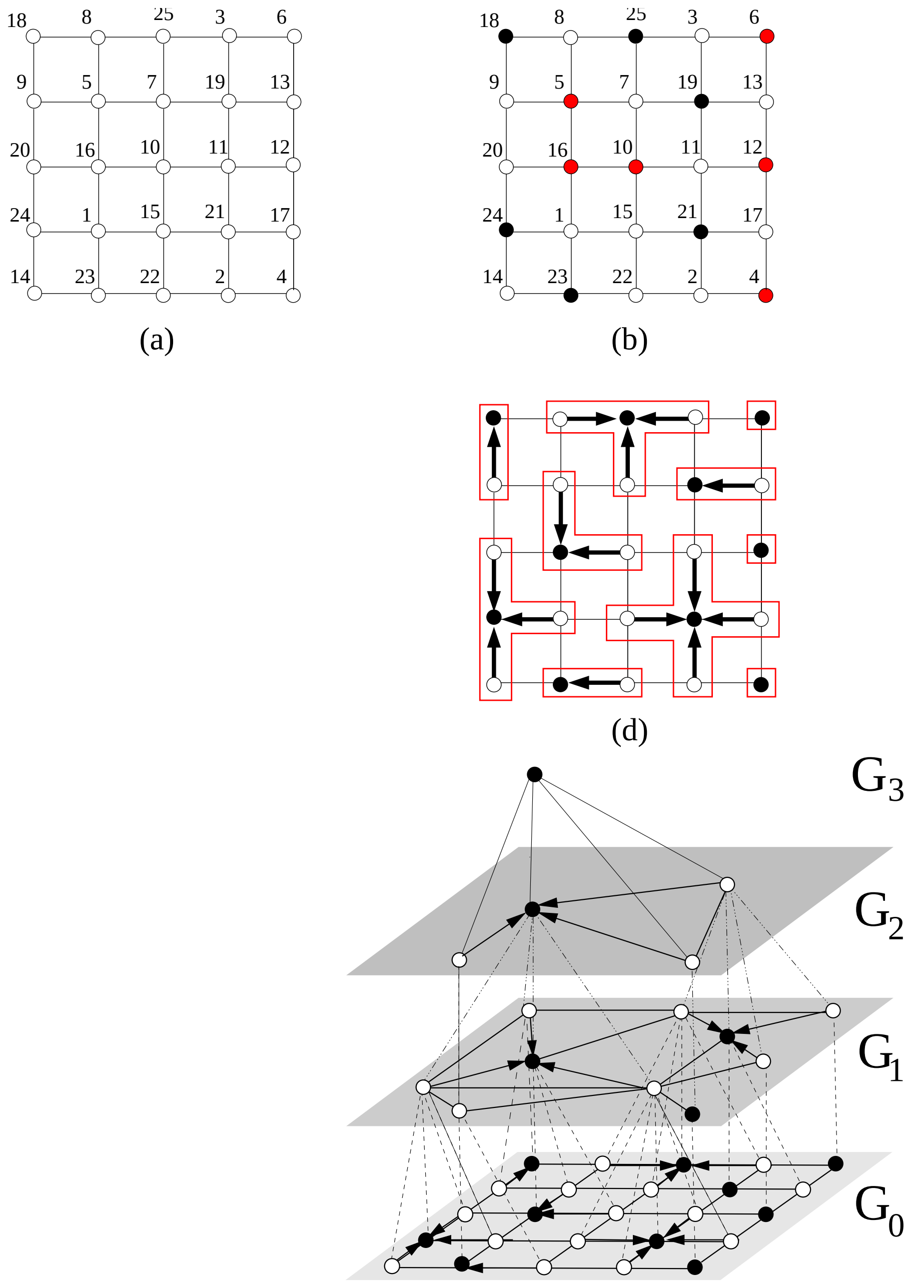
1. $w_1 \notin \Gamma_l(w_2)$ for all $w_1, w_2 \in W_l$,
2. for all $v \in V_l$ there exists a vertex $w \in W_l$ such that $v \in \Gamma_l(w)$.

The **1st** condition means that two surviving vertices cannot be in the neighborhood of each other, black vertices in Figure 2(d). The **2nd** condition says that every non-surviving vertex has in its neighborhood a surviving vertex, white vertices in Figure 2(d). Maximal independent vertex set (MIS) problem was solved using the heuristic in [Mee89]. The number of iterations to complete MIS converges in most of the cases very fast, so called iterations for correction [Mee89]. MIS [Mee89, Jol02] may be generated as follows.

MIS Algorithm:

1. Mark every element of V_l as *candidate*.
2. Iterate the following two steps as long as there are candidates.
   (a) Assign random numbers to the candidates of V_l.
   (b) Determine the candidates whose random numbers are greater than the random numbers of all neighboring candidates and mark them as *member* (of the maximal independent set) and as *non-candidate*. Also mark every neighbor of every new member as *non-candidate*.
3. In each neighborhood of a vertex that is not a member there will now be a member. Let each non-member choose the edge to be contracted to its neighboring member, say the one with the maximal random number (we assume that no two random numbers are equal).

The **2nd** step of the algorithm (the iteration for correction) is shown in more detail in a simple graph in Figure 2(a),(b),(c). The random numbers assigned to the vertices of graph are shown in Figure 2(a). In Figure 2(b) is shown the parallel process of finding the local maximum vertices and marking them as *members* (depicted with black) and as *non-candidates*, also marking as *non-candidate* all the vertices that are neighbors of the *members*, vertices shown with white. Since after this step there are some *candidates* (5, 6, 15, 10 and 12) status of which is undefined, we repeat the **2nd** step in order to find a status for these *candidates*. The complete maximal



(a) (b) (c)

(d)

(e)

Figure 2: (a),(b),(c) Second step of MIS. (d) The black vertices form a maximal independent vertex set. (d) The frames indicate a corresponding collection of contraction kernels. (e) A graph pyramid from maximal independent vertex sets.

independent set of vertices for this simple example is found in two iterations, Figure 2(c). The assignment of the non-members to their members determine a collection of *contraction kernels*: each non-member is contracted towards its member and all contractions can be done in a single parallel step, **3rd** step of the algorithm. In Figure 2(d) the contractions are indicated by arrows. A graph pyramid from maximal independent vertex sets can be seen in Figure 2(e), where G_i are graphs in levels $i = 0, ..., 3$. Note that we remove parallel edges and self-loops that emerge from the contractions, if they are not needed to encode inclusion of regions by other regions, in the example of Figure 2(e) we do not need loops nor parallel edges. This can be done by the dual graph contraction algorithm [Kro95]. Trivial contraction kernels do occur very often in MIS as can be seen in Figure 2(d), isolated black vertices. In Section 3 we introduce a method that finds contraction kernels that are non trivial.

3 Guaranteeing a Reduction Factor of 2.0

In the following we aim at a collection $\mathcal{C}$ of contraction kernels in a plane graph G such that

- each vertex of G is contained in exactly one kernel of $\mathcal{C}$, and
- each kernel $\mathcal{C}$ contains at least two vertices.

Definition 1 (Maximal Matching) *A set M of independent edges [2] in a graph $G = (V, E)$ is a maximal matching if every vertex in $U \subset V$ incidents with an edge in M, and M cannot be enlarged by an edge without loosing independence. The vertices in U are then called matched vertices (by M); all other vertices are called unmatched vertices.*

We assume that G is connected and planar, and that the data do not impose constraint on the selection, i.e. in large homogeneous regions. Clearly, the contraction of all kernels in $\mathcal{C}$ will reduce the number of vertices to half or less. In contrast to [Mee89] we start with independent *edge* sets or *matchings*, i.e. edge sets in which no pair of edges has a common end vertex. The *maximal independent edge set* (MIES), $\mathcal{C}$ is done in three steps.
MIES Algorithm:

1. A maximal matching M of edges from G is determined.
2. M is enlarged to a set M^+ that induces a spanning subgraph of G.
3. Larger trees of M^+ are split into smaller trees defining $\mathcal{C}$.

[1] also called maximal stable set; we distinct maximal from maximum independent set, which construction is NP-complete.

[2] Two edges are independent if they do not incident in the same vertex.

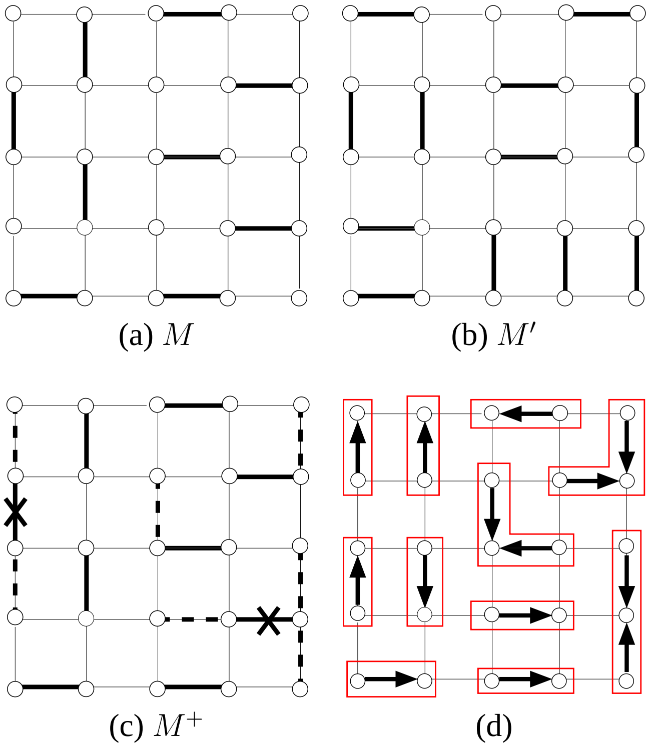


Figure 3: (a) M: a maximal matching with 9 edges. (b) M': a matching with 12 edges. (c) M^+: the matching from (a) enlarged by connecting formerly isolated vertices to the maximal matching. (d) (c) after breaking up trees of depth two into trees of depth one.

In the **1st** step, a maximal matching may be determined by an iteratively local process as specified as used in MIS algorithm, (Section 2). Note that a maximal matching of G is equivalent to a maximal independent vertex set on the edge graph of G [Die97, Chr75]. Note that M is only required to be maximal, the edge set M cannot be enlarged by another edge from G without loosing independence. As can be seen in Figure 3(a), a maximal matching M is not necessarily maximum: there is a matching M' (Figure 3(b)) that contains more edges than M.

The collection of contraction kernels defined by a maximal matching M may include kernels with a single vertex. Let v denote such an isolated vertex, isolated from M, and choose a non-self-loop e that has v as an end vertex. Since M is maximal, the other end vertex $w \neq v$ of e belongs to an edge that is contained in the matching. Let M^+ denote the set of edges that are in M or that are chosen to connect isolated vertices to M, **2nd** step of MIES.

The subgraph of G that is induced by M^+ spans G and its connected components are trees of depth one, two or three, Figure 3(c). A tree of depth three can be separated into two trees of depth one each by removing the unique edge, both end vertices of which belong to other edges of the tree (indicated by the crosses in Figure 3(c)); **3rd** step of MIES. Still, each vertex of G belongs to a tree (of depth one). The arrows in Figure 3(d) indicate possible directions of

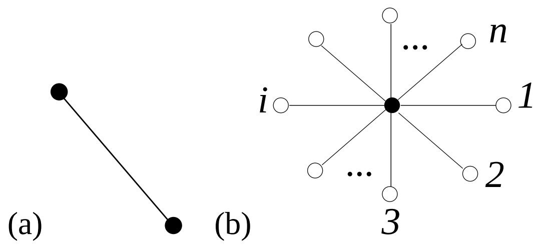


Figure 4: Restriction in choosing the surviving vertex. (a)Without restriction, (b) only the center vertex (depicted with black) can be chosen as the root of the tree.

contractions. Since each vertex of G is now contained in a non-trivial contraction kernel, we proved the following proposition:

Proposition 1 (Reduction factor of MIES at least 2.0) *The MIES algorithm yields contraction kernels for parallel contractions with a reduction factor of at least* 2.0.

Proof. Let $G = (V_l, E_l)$ be a planar connected graph with $|V_l|$ number of vertices. Let M be a maximal matching of G and let U be the set of vertices matched by M and $U' = M \setminus U$ be the set of unmatched vertices (isolated vertices). A vertex $v \in U$ is called matched by M if it is incident with an edge in M. Since the matching is maximal, no two connected vertices v, w may be unmatched. i.e. in the neighborhood of the unmatched vertex there is at least one matched vertex (since the graph is connected). This means that in the neighborhood of an unmatched vertex $w \in U'$ there is at least one **connecting edge** (of path length one) concatenating this unmatched vertex to the matched vertex $v \in U$. If there are more than one connecting edge, we select only one of them.

Edge set M is enlarged to M^+ by adding all connecting edges. Thus all vertices of V_l are incidented in edges of M^+. The subgraph of G that is induced by M^+ spans G and its connected components are trees of diameter one (the matched edge), two (isolated vertices connected to one endpoint of a matched edge only) or three (isolated vertices connected to both ends of the matched edge). Trees of diameter three are splitted into trees of diameter one by removing the **unique central edge** ue, both end vertices of which belong to other edges of the tree. The two endpoints of the matched edge of a tree of diameter three have a degree of at least two (they cannot be leafs). The split of the tree removes this matched edge which reduces the degree of both end vertices by one, but they are still greater than zero. Hence this edge deletion does not create a new isolated vertex. This ensures that all vertices V_l are also incidented in edges of the new set $M^{++} = M^+ \setminus \{ue\}$ and there are no isolated vertices left.

We conclude that the subgraph of G that is induced by M^{++} spans G and its connected components are trees of diameter one or two. If all the components are trees of depth one then after contraction we have $|V_{l+1}| = |V_l|/2$. If there is at least one tree of diameter two than $|V_{l+1}| < |V_l|/2$. We proved that in general case $|V_{l+1}| \leq |V_l|/2$.
$\sqcap$.

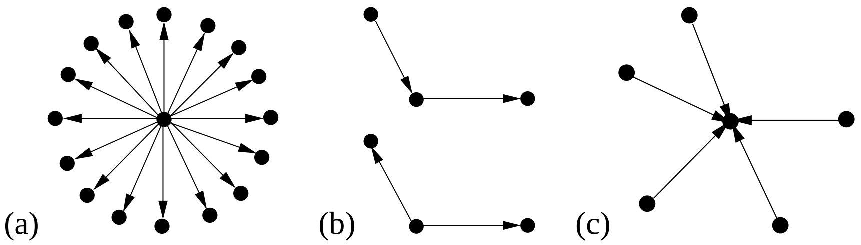


Figure 5: (a) The reduction factor of a star with n edges pointing away from the center is $(n+1)/n$. (b) Forbidden pairs of directed edges. (c) A legal configuration of directed edges

The surviving vertices can be in the neighborhood of each other, relaxing the **1st** condition of the MIS, which were proposed in [KM91]. Non-surviving vertices are in the neighborhood of at least one surviving vertex, fulfilling the **2nd** condition of MIS (Section 2). Note that in case of kernels with more than one edge, Figure 4(b), the roots within the kernel cannot be chosen arbitrarily. In Figure 4(a) any of the vertices can be chosen to be the root of the tree. But for the case in Figure 4(b) the black vertex can only be chosen as root of the tree, otherwise the (rooted) trees have depth greater than one. This would contradict our requirement about the depth of the contraction kernel. This is why the proposed method cannot be extended to applications in which there are a priori constraints on the directions of the contractions. However, the proposed method works for the stochastic case (no preconditions on edges to be contracted) and for connected component analysis [KM95], where the attributes of the end vertices are required to be identical.

4 Constraints on the Directions of the Contractions

In many graph pyramid applications such as line image analysis [BK99, KB98] and the description of image structure [GEK99, KG01, GK00b] 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. In particular, the condition depends on u being the source and v being the target. The edges that fulfill the condition are called *preselected* edges. From now on the plane graphs in the pyramid have directed edges. Undirected graphs can be converted into directed graphs by substituting the edges into pairs of reverse edges, i.e. for each edge e with source u and target v there exists an edge e' with source v and target u. However, the set of preselected edges may contain e without containing e'. The goal is to build contraction kernels with a “high” reduction factor from the set of preselected edges. The reduction will always be determined according to the directed graph induced by the preselected edges. For example, if the number of vertices in the induced subgraph is reduced to half, the reduction factor will be 2.0. From the example in Figure 5(a) it is clear that, in general, the reduction factor can be arbitrarily close to 1.0.

The contraction kernels are vertex disjoint rooted trees of depth one or zero (single vertices),

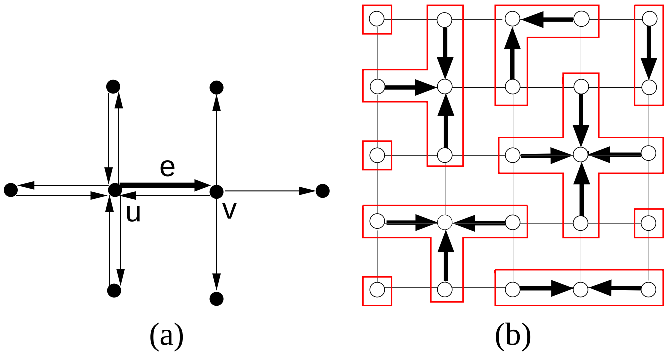


Figure 6: (a) The neighborhood $N(e)$. (b) Maximal independent edge set with respect to $N(e)$.

each edge of which is directed toward the root. A set C of directed edges forms such a collection of contraction kernels if and only if C contains none of the edge pairs depicted in Figure 5(b). Seen from a directed edge e with source u and target $v \neq u$ that one wants to contract (from u to v), no edge $e' \neq e$ with end vertex (source or target) equal to u or source equal to v may be contracted. An edge e to be contracted together with those edges that one may not contract form a directed edge neighborhood $N(e)$ of e. Figure 6(a) depicts $N(e)$ in case of u and v both having 4 neighbors. To find a *maximal (independent) set of directed edges* (MIDES) forming vertex disjoint rooted trees of depth zero or one, we proceed analogously to the generation of maximal independent vertex sets (MIS), as explained in the Section 2.

Let E_l denote the set of directed edges in the graph G_l of the graph pyramid. We proceed as follows:

MIDES Algorithm:

1. Mark every directed edge of E_l as *candidate*.

2. Iterate the following two steps as long as there are candidates:

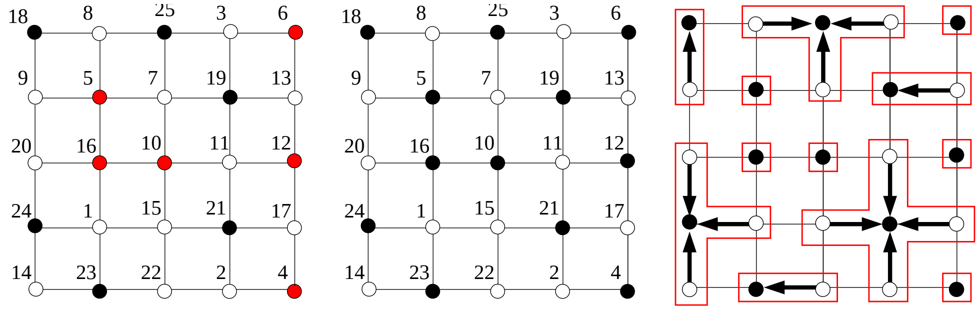
   (a) Assign random numbers to the candidates.

   (b) Determine the candidates e whose random numbers are higher (larger) than the random numbers in $N(e) \setminus \{e\}$ and mark them as *member* (of a contraction kernel). Also mark every $e' \in N(e)$ of every new member e as *non-candidate*.

As for the MIES, surviving vertices can be in the neighborhood of each other.

5 Data Driven Decimation Process

In this section we recall an idea proposed in [Jol02, Jol01] on how to build stochastic irregular image pyramid, the *data driven decimation process* (D3P). The main difference from the MIS



(a) (b) (c)

Figure 7: (a), (b) Second step of D3P and (c) contraction kernels.

algorithm presented in Section 2 is that the **2nd** step of MIS is relaxed, i.e. no iteration will be performed. Skipping out the iteration is motivated by the fact that the iteration is used only for completing the maximal independent set. It is assumed that being a local maximum is of importance [Jol02, Jol01].

Let the vertex set of G_l be denoted by V_l. Jolion [Jol02, Jol01] proceed as follows:

D3P Algorithm:

1. Mark every element of V_l as *candidate*.

2. Assign random numbers to the candidates of V_l.

   (a) Determine the candidates whose random numbers are greater than the random numbers of all neighboring candidates and mark them as *member* (of the maximal independent set) and as *non-candidate*. Also mark every neighbor of every new member as *non-candidate*.

   (b) Mark as *member* all the remaining candidates that do not have a *member* in the neighborhood.

3. In each neighborhood of a vertex that is not a member there will now be a member. Let each non-member choose its neighboring member, say the one with the maximal random number (we assume that no two random numbers are equal).

We will use again example shown in Figure 2 to explain the difference between MIS and D3P. All the *candidates* that have undefined status in **2nd** step in MIS algorithm are simply marked as *members*. Note the differences shown in Figure 2(c) and Figure 7(b). The *candidates* that have non-defined status (6, 5, 16, 10, 12, and 4) after the step **2**(a) are marked as *members* in step **2**(b), shown with black in Figure 7(b). In Figure 7(c) is given a set of contraction kernels build by the **3rd** step of the algorithm.

6 Comparing the Speed of Reduction

The experiment in this section compare the height of the pyramid, i.e. the reduction factors of vertices, edges and faces of stochastic graph pyramids using algorithms MIS, MIES, MIDES, and D3P.

Uniformly distributed random numbers are assigned to the vertices in the base level graphs. Then the pyramids have been built using the stochastic strategy. Finally we calculated statistics of all pyramids built by each specific selection (MIS, MIES, MIDES, D3P) to compare the properties of different strategies. The same observation resulted from the two different random number generators [MN98, MN99]. By changing the seed of the uniformly distributed random generator we generated 1000 graphs, on top of which we built stochastic graph pyramids. We reduce these graphs until we reach the top of the pyramid, the apex, which consist of one vertex and one face and no self loops.

In our experiments, Section 6.1, Section 6.2, Section 6.4, and Section 6.3 we used bi-directed grid graphs of size 10000, 22500 and 40000 vertices respectively, which correspond to image sizes of 100×100, 150×150 and 200×200 pixels, respectively.

Figures 9, 10, 11, and 12 summarize the results of the first 100 of 1000 tests for images of sizes 100×100, 150×150 and 200×200 pixels; solid lines connect the observed number of vertices/edges/faces of one particular pyramid, where the number of levels of the graph pyramid constitute the horizontal axis and the vertical axis shows the number of vertices v, edges e and faces f in logarithmic scale at the respective pyramid level. In this choice of coordinate axes a constant reduction factor becomes a straight line. Since we reduce the pyramid to a single vertex in the apex all the solid lines in all $v-$ and $f-$ diagrams end at the $10^0 (= 1)$. The solid lines of the e-diagrams end at the last but one level since the top does not contain any edge.

Statistical results using 1000 graphs are discussed in Subsection 6.5 and Tables 1, 2, and 3. The statistics of vertex degree [3] is given in Table 1. These parameters are of importance because they are directly related to the memory costs of the representation of graph [Jol02, Jol01]. In Table 2 the mean and standard deviation for height of the pyramid, and for the number of iteration for corrections are given. The mean and standard deviation of the decimation ratio for vertices and edges is given in Table 3. The software tool dgc_tool [4] [SHG02] was used to make these results.

6.1 Experiments with MIS

The number of levels needed to reduce the graph at the base level (level 0) to a graph consisting of a single vertex, with no self loops and a single face (top of the pyramid) are given in Figure 9, (v) for vertices, (e) for edges and (f) for faces. From Figure 9 we see that the height of the pyramid cannot be guaranteed to be logarithmic, except for some good cases. In the worst case the pyramid had 22 levels for 100×100 vertices and 41 levels for the graph with 200×200

[3] The number of edges incident to a vertex

[4] Can be found at ftp://ftp.prip.tuwien.ac.at/pub/dgc_tool/

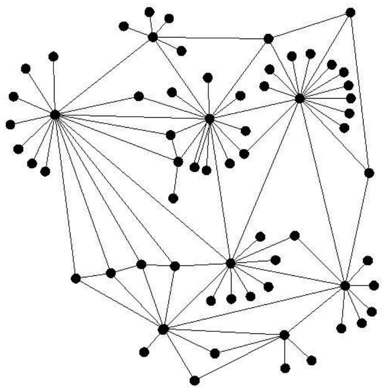


Figure 8: Evolution of vertices with large neighborhood using MIS.

vertices, respectively. Poor reduction factors are likely, as can be seen in Figure 9, especially when the images are large. This is due to the evolution of larger and larger variations between the vertex degrees in the contracted graphs (Table 1, $\mu(max) = 70.69$ and Figure 8). The absolute maximum indegree was 148. The *a priori* probability of a vertex being the local maximum depends on its neighborhood. The larger the neighborhood the smaller is the probability that a vertex will survive [Jol02]. MIS tends to have vertices with a large neighborhood (stars), as shown in Figure 8. This causes the reduction factor to be very poor at highest levels, also noted in [MMR91], which is mainly due to the good performance of the decimation at the first few levels where the graphs have large sizes and a small neighborhood size (e.g. each vertex has 4 neighbors in the base level). The mean reduction factor of vertices is 1.94 (Table 3).The collapse of the hight-order-star-like configuration into the apex causes the sudden break of the solid lines ("star-contraction") in the v and f -diagrams. The number of iterations necessary to complete the maximum independent set per level (iterations for correction [Mee89]) are 3 as reported by [Mee89]. The mean value of the height of the pyramid is 20.78 (Table 2).

To summarize, a constant reduction factor higher then 1.0 cannot be guaranteed and bad cases have a high probability (almost horizontal solid lines in Figure 9).

6.2 Experiments with MIES

The number of levels needed to reduce the graph at the base level by the MIES strategy to an apex are shown in Figure 10. The experiments show that the reduction factor, even in the worst case, is always higher than the theoretical lower bound 2.0, as indicated by the dashed line in Figure 10(v). The MIDES is more stable than MIS, as can be seen in Figure 10, where the slope of the solid lines have smaller variations. The mean and the variance of reduction factor for MIES is smaller then in case of MIS or D3P, which implies that the height of the pyramid ($\mu(height) = 14.01$) is also smaller than that for MIS ($\mu(height) = 20.78$) or D3P ($\mu(height) = 88.25$).

The mean number of iteration for correction per level was higher for MIES (Table 1). To summarize the reduction factor were always under the theoretical upper bound of 2.0.

6.3 Experiments with MIDES

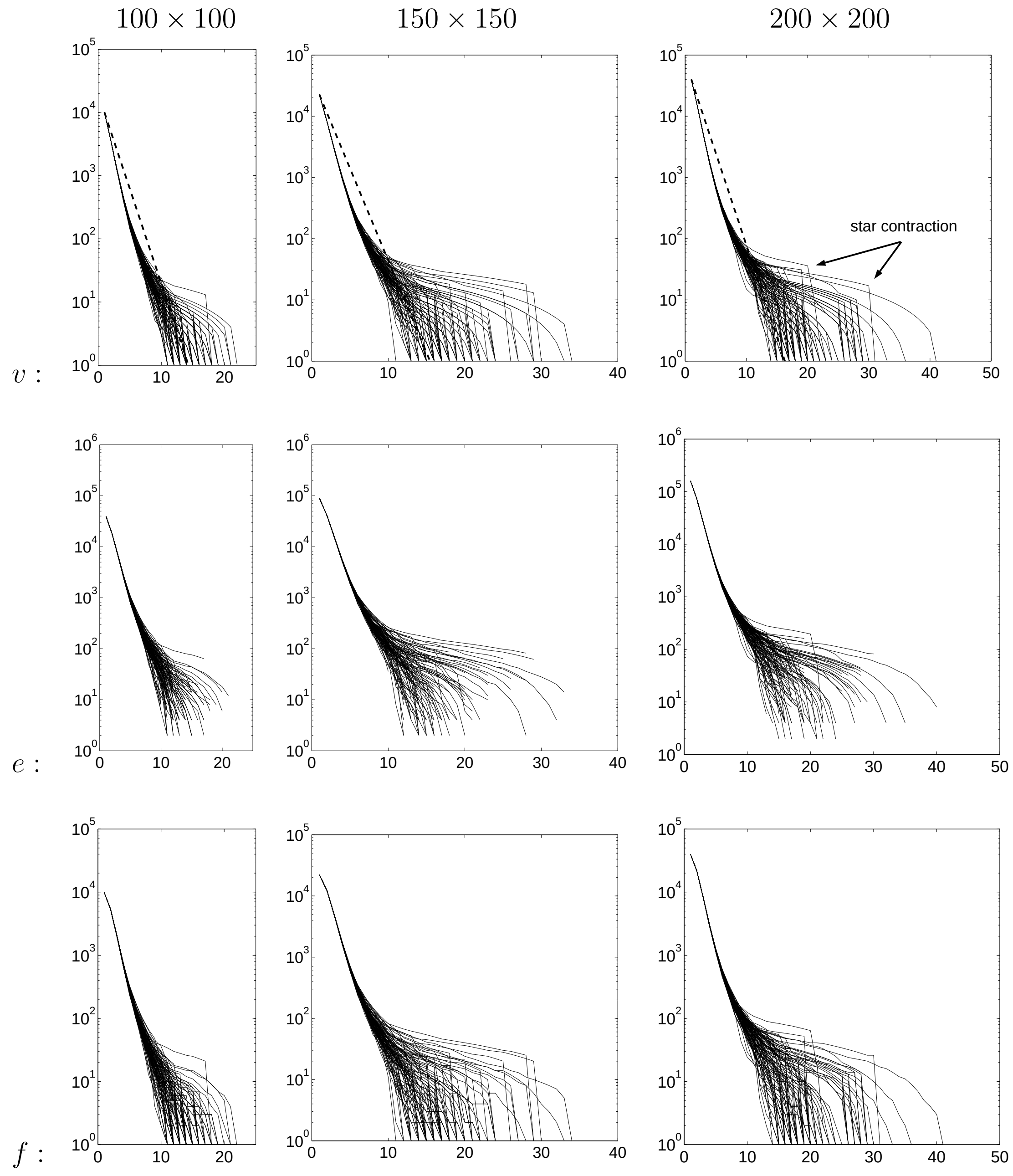
The Figure 11 depicts the number of levels required to get on top of the pyramid. We see that the reduction factor is higher than 2.0 (dashed line) even in the worst case. Also the maximum indegree of the vertices is much smaller ($\mu(max) = 13.29$) than for MIS ($\mu(max) = 70.69$, Table 1). For MIES and MIDES we have not encountered nodes with large neighborhood as for MIS. For the case of the graph with size 200×200 vertices, MIDES needed 13 levels in comparison to 15 levels in the worst case of MIES. The number of iterations needed to complete the maximum independent set was comparable with the one of MIS (Table 1). The MIDES algorithm shows a better reduction factor than MIES, as can be seen in Figure 11 and Table 3 ($\mu(v) = 2.62$).

To summarize the reduction factor was always better then the theoretical upper bound of 2.0 in all our tests. Bounded reduction factor cannot be guaranteed.

6.4 Experiments with D3P

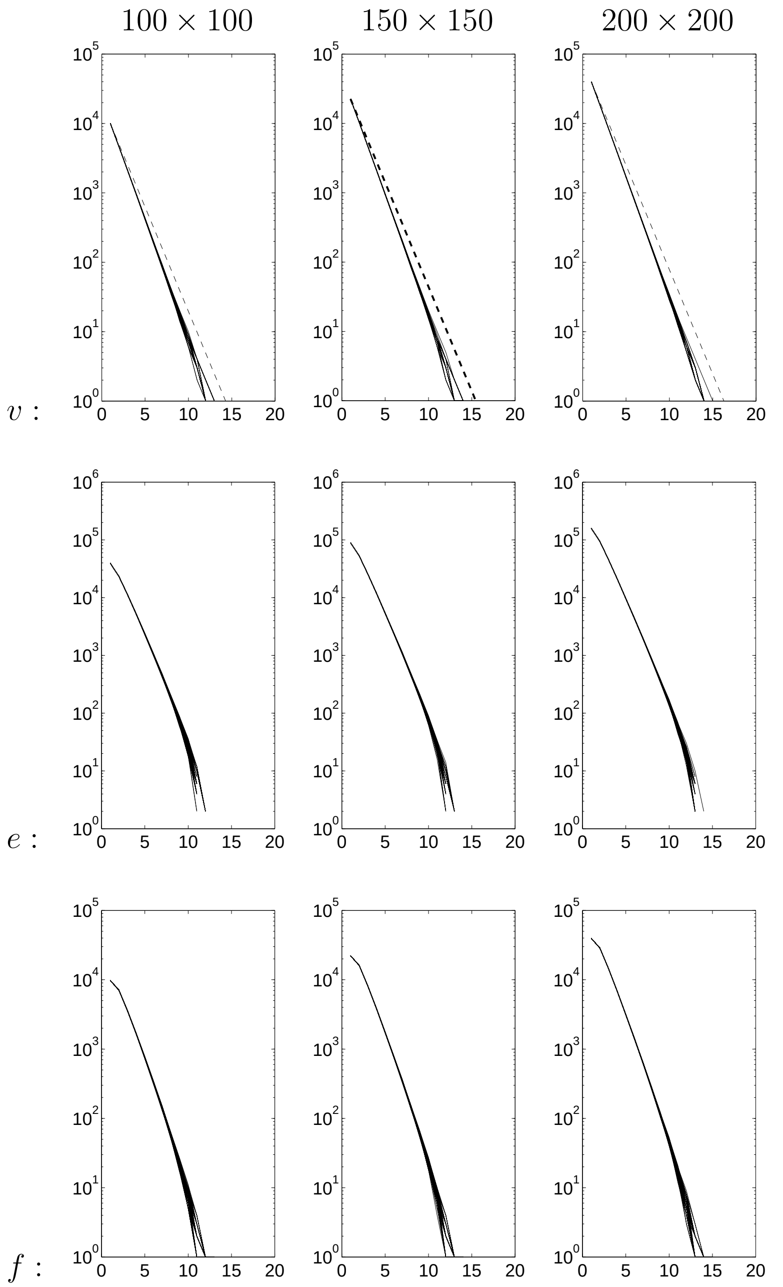
The Figure 12 gives the reduction factors (v) for vertices, (e) for edges and (f) for faces. The experiments show that poor reduction factors are likely, because of the large degree of vertices ($\mu(max) = 433.92$, Table 1). Also the height of the pyramid is related to the complexity of the vertices, which is why the D3P gives the highest pyramids ($\mu(height) = 88.25$). The number of iterations for correction is 1 because we do not iterate at all. As expected we have large neighborhoods for D3P.

To summarize, a constant reduction factor, like for the MIS, cannot be guaranteed and bad cases have a high probability, as can be seen in Figure 12.



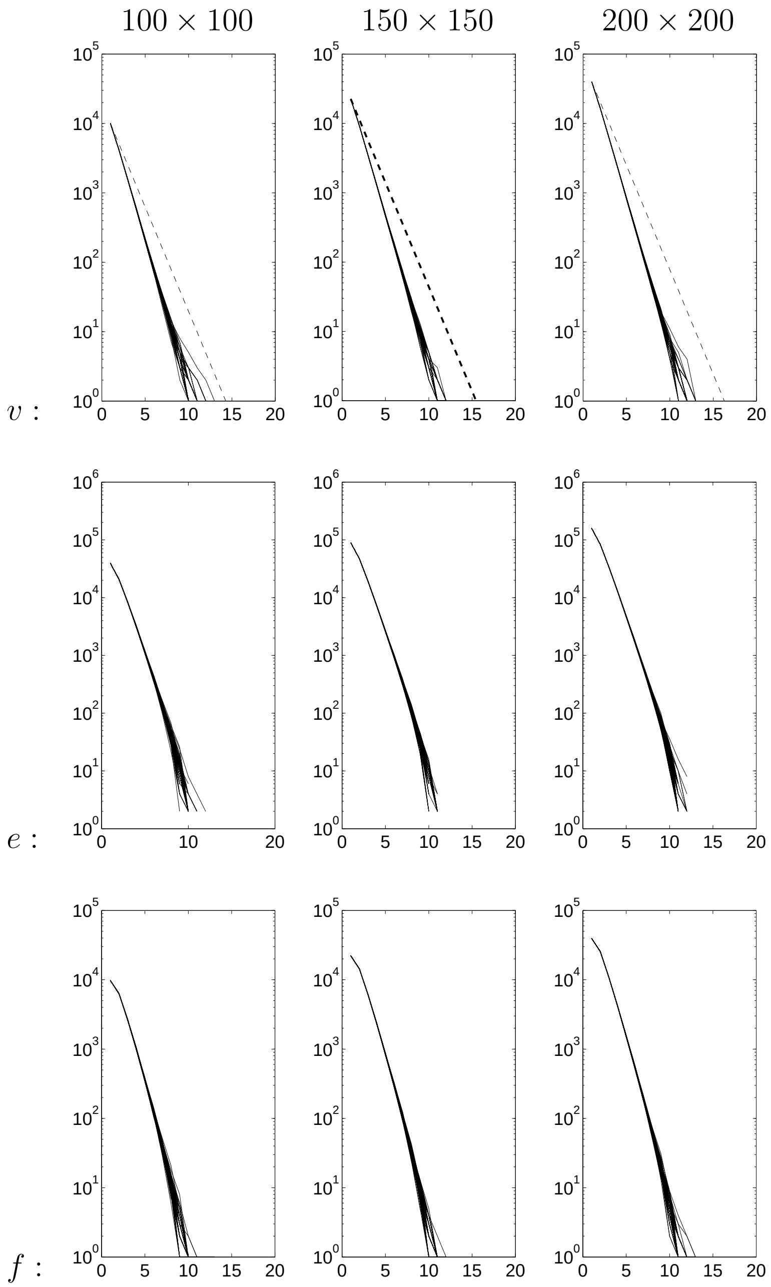
Legend: Number of vertices v , edges e , and faces f (y-axis) in levels (x-axis) of the first 100 MIS pyramids. The base levels are rectangular grid containing 100×100, 150×150 and 200×200 vertices. Dashed lines represent the theoretical reduction factor of 2.0.

Figure 9: **MIS** Algorithm.



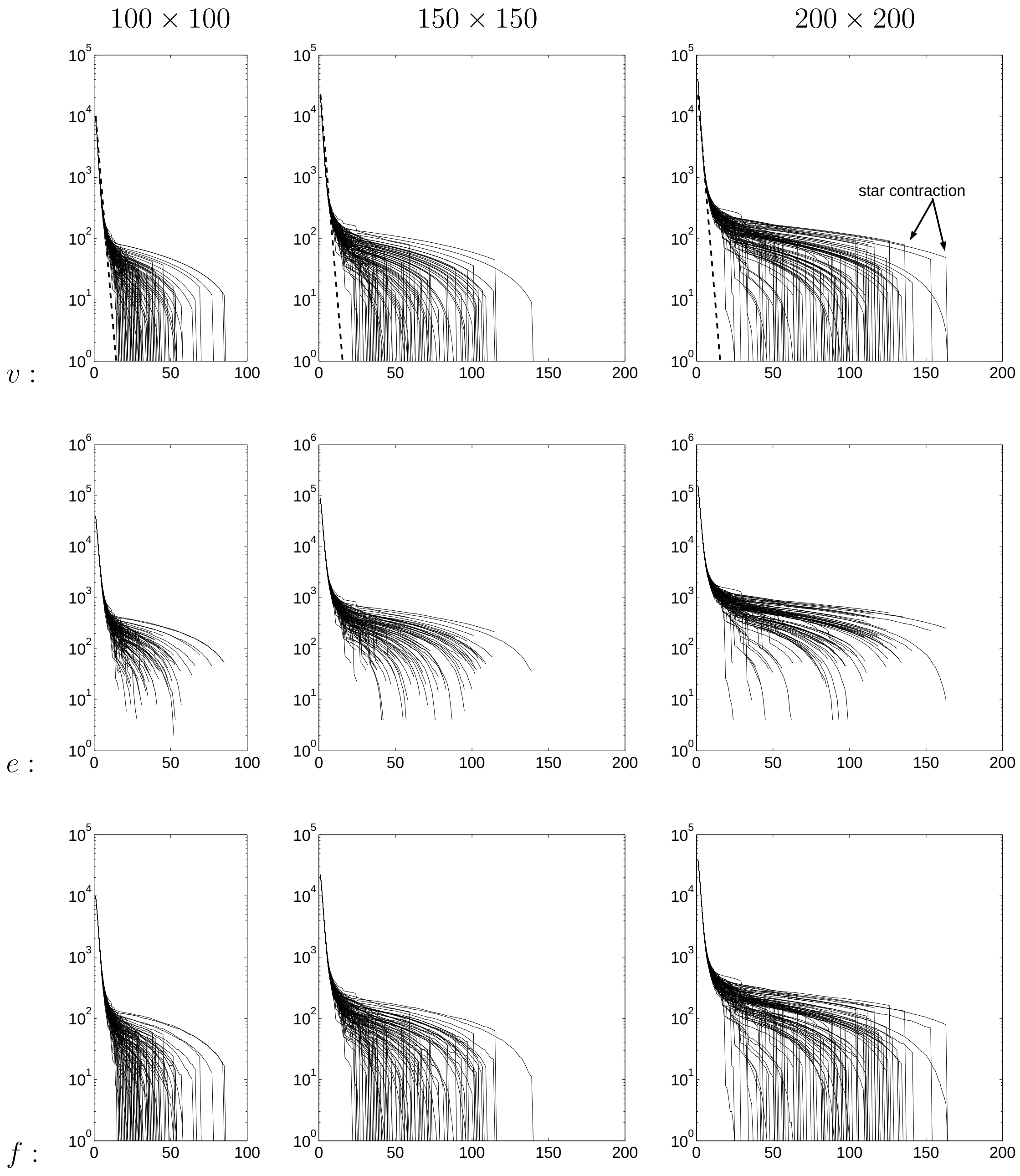
Legend: Number of vertices v, edges e, and faces f (y-axis) in levels (x-axis) of the first 100 MIES pyramids. The base levels are rectangular grid containing 100×100, 150×150 and 200×200 vertices. Dashed lines represent the theoretical reduction factor of 2.0.

Figure 10: **MIES** Algorithm.



Legend: Number of vertices v , edges e, and faces f (y-axis) in levels (x-axis) of the first 100 MIDES pyramids. The base levels are rectangular grid containing 100×100, 150×150 and 200×200 vertices. Dashed lines represent the theoretical reduction factor of 2.0.

Figure 11: **MIDES** Algorithm.



Legend: Number of vertices v, edges e, and faces f (y-axis) in levels (x-axis) of the first 100 D3P pyramids. The base levels are rectangular grid containing 100×100, 150×150 and 200×200 vertices. Dashed lines represent the theoretical reduction factor of 2.0.

Figure 12: **D3P** Algorithm.

6.5 Discussion of Results

In Table 1 a more extensive comparison is made using 1000 graphs of size 200×200. We extract three parameters, the maximum (*max*), the mean (*mean*) and the standard deviation (*std*) of the vertex degree for any graph in the contraction process. We average these values (μ) and find the respective standard deviation (σ) for all 1000 graphs. The maximum degree was encountered using D3P and MIS which is why this method have problems during contractions. Thus memory costs for D3P and MIS will be higher than for MIES and MIDES. Notice however that the mean degrees are similar for all algorithms. The height stability (in the sense of smaller variation) in the first case shows that MIS, MIES and MIDES do not depend on the data. The degrees for D3P exhibits more variations. This means that D3P better fit to the distribution of values associated with the vertices [Jol02, Jol01]. The last line in Table 1 shows the vertex complexity on base level G_0.

Table 2 gives statistics about *height* of the image pyramid and about number of iteration ($\#iterations$) to complete the maximal independent set. Table 3 gives the statistics about the decimation ratios of vertices and edges. Results in both tables were computed using 1000 graphs of size 200×200. Number of iterations for correction was almost the same for all methods except for the D3P, where it was 1 because we do not iterate at all. MIDES gave the best reduction factors ($\mu(v) = 2.62$) for all tests (Table 3). MIS, MIDES and D3P have the same algorithmic complexity for the worst case. The worst case happens when the neighboring vertices have increasing random numbers. We have not encountered the worst case in out test, since it is highly unlikely.

The *a priori* probability that a vertex survives depends on the size of its neighborhood. In Figure 13 vertex u will be favored by MIS and D3P. Vertex u survives with the probability of $1/2$, since it has only the vertex v in the neighborhood. Vertex v survives with the probability of $1/n$, where n is the size of its neighborhood. The center of the star will have smaller probability to survive than its leafs, which causes the poor reduction factor, since only one edge will be contracted. Since two surviving vertices cannot be neighbors, the center of the star will be pulled toward one of it leafs. But still there will be a vertex with large neighborhood. There

Table 1: Mean μ and standard deviation σ of maximum, mean and standard deviation of **vertex degrees** of the pyramids, respectively.

Method	$\mu(max)$	$\sigma(max)$	$\mu(mean)$	$\sigma(mean)$	$\mu(std)$	$\sigma(std)$
MIS	70.69	23.88	4.84	0.23	3.45	1.18
MIES	11.74	0.71	4.78	0.45	0.07	0.04
MIDES	13.29	1.06	4.68	0.14	0.57	0.08
D3P	433.92	123.41	5.17	0.14	8.74	1.83
G_0	8		7.95		0.28	

Table 2: Mean μ and standard deviation σ of **height of the pyramid** and of **number of iterations** to complete maximum independent set (iteration for correction).

Method	$\mu(height)$	$\sigma(height)$	$\mu(\#iterations)$	$\sigma(\#iterations)$
MIS	20.78	5.13	2.95	0.81
MIES	14.01	0.10	4.06	1.17
MIDES	12.07	0.46	2.82	1.07
D3P	88.25	31.72	1	0

Table 3: Mean μ and standard deviation σ of **decimation ratios** for vertices v and edges e.

Method	$\mu(v)$	$\sigma(v)$	$\mu(e)$	$\sigma(e)$
MIS	1.94	1.49	1.78	1.16
MIES	2.27	0.21	2.57	1.21
MIDES	2.62	0.36	3.09	1.41
D3P	1.55	5.06	1.11	0.41
Ideal	4			

are cases where the center of the star is the largest in its neighborhood. In these cases all its leafs will be contracted towards the star, yielding a very good reduction factor. The contraction of stars can be seen in Figures 9, and 12, where the solid lines descent rapidly. An arrow in Figure 9(v) and Figures 12(v) depict an example of star contraction.

The probability that edge e in Figure 13 will be contracted i.e. one of the end vertices will survive, is the same for all neighboring edges of e using MIES or MIDES. The existence of vertices with large neighborhood was not encountered in MIES and MIDES, which can be seen also in Figures 10, and 11, where there are no cases of rapidly descent lines.

Since no vertex is favored the size of receptive fields using MIES or MIDES will be better distributed, in the intermediate levels of the pyramid. There is no occurrence of very small or very big receptive fields as for MIS or D3P, where there are receptive field small as one pixel in higher levels of the image pyramid. Figure 14 shows receptive fields of comparable pyramid levels of MIS, D3P, MIES and MIDES. Each vertex of the same level received arbitrary but distinguishable gray values which was propagated down to the base of the pyramid. Hence the regions with the same gray value in Figure 14 correspond to the receptive field of one vertex of the high level. Without any constraint from the data there is no need to require big variations in the sizes of receptive fields. This kind of behavior was encountered also in Section 7.1.

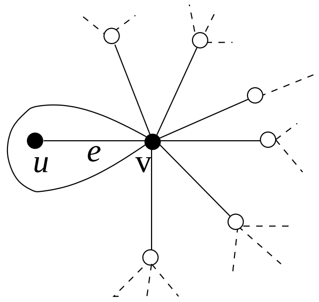


Figure 13: (a) Vertex u is favored by MIS and D3P.

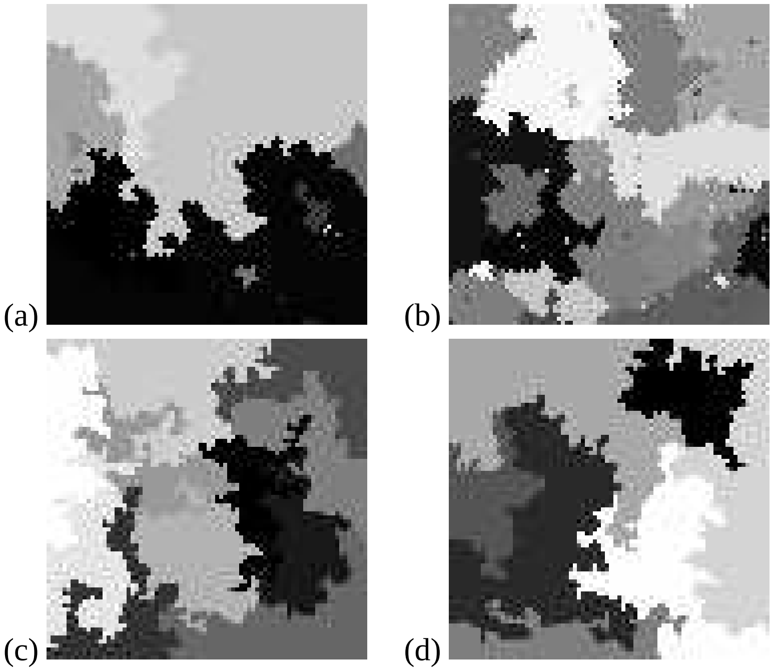


Figure 14: Receptive fields from MIS (a), D3P(b), MIES (c), and MIDES (d).

7 Path Lengths of Image Pyramid

In some applications some parts of image are of special interest, so we need to access data in a top-down process very often [GK00a]. We present in this section results of the study of the vertical path lengths (costs) in stochastic graph image pyramids. This will give us a way to measure the structure of the irregular pyramids built with different algorithms. 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 lengths. This information is important for quickly analyzing distinct objects in a top-down/bottom-up process. We are interested not only in comparing the vertex complexity and the height of the pyramid, as done in Section 6, but also to be able to compare the internal structure of image pyramids of the same height. Our goal is to build stochastic pyramids locally that are optimal in the sense of bottom-up and top-down processes.

First we built a graph pyramid in a bottom-up process using one of the algorithms MIS, MIES, MIDES or D3P to find contraction kernels (decimation parameters). Graphs are reduced using the dual graph contraction algorithm [Kro95]. Since we are building stochastic graph pyramids we end having always a singe vertex at the top of a pyramid.

As stated earlier, contraction kernels are rooted trees of depth one. The Figure 15(a) 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 ns, and they are contracted (arrows) towards the surviving son ss, 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:

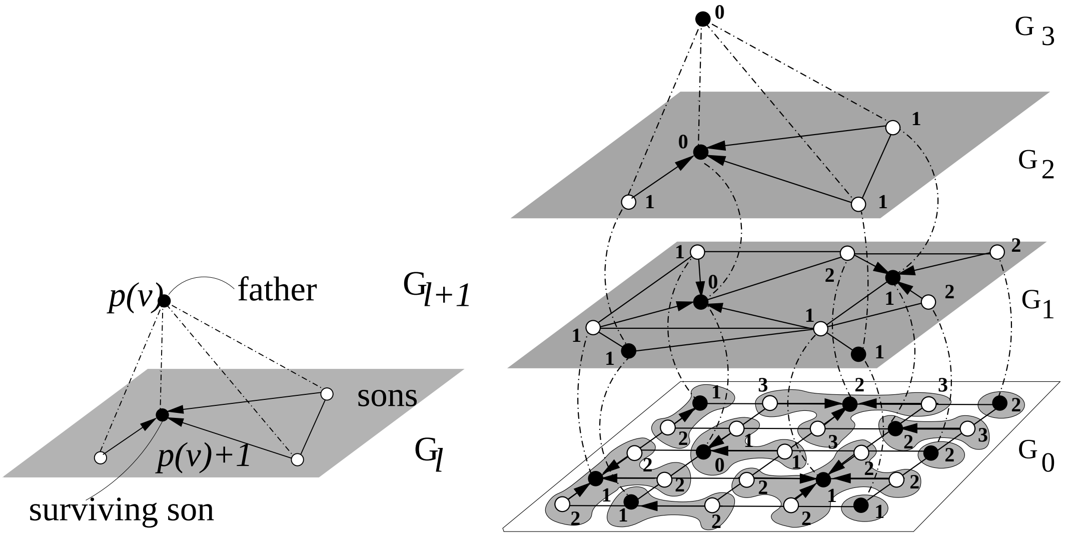
1. Let the vertices $v \in G_h$ at the top of the pyramid have path length 0, $p(v) = 0$.

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

   $\forall v \in G_{l+1}$: downpropagate the path length $p(v)$ 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 Figure 15(b). 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 [Kro95].

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 son is kept zero since it is 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



(a) Father-son relations (b) Path lengths

Figure 15: Graph pyramid built using DGC.

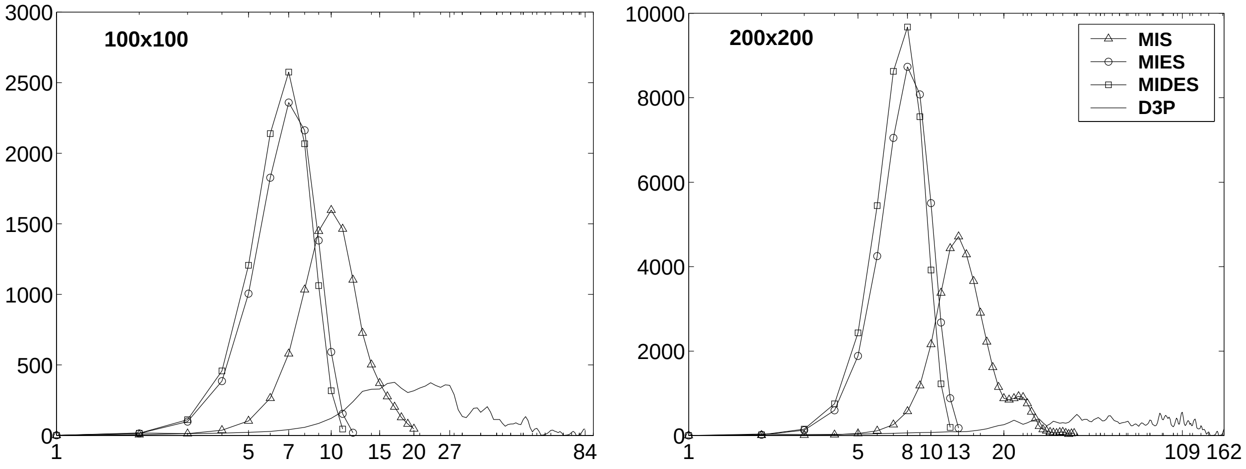
incremented path length of the father by one. In Figure 15(b) at G_2 a vertex has the same length as its father (black vertex with path length 0), and all non-surviving sons (white vertices) have the path length $0 + 1 = 1$. To arrive to the top of the pyramid from the edge with path length 3, three edges must be contracted.

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

7.1 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. We used one of the algorithms MIS, MIES, MIDES or D3P to build stochastic pyramids i.e to compute the path lengths. We contract these graphs using dual graph contraction [Kro95] until we reach the 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 Figure 16, (a) for 100×100 and (b) 200×200 image size. The two diagrams of Figure 16 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



(a)

(b)

Figure 16: μ of path lengths. x axis the path lengths, and y-axis number of vertices.

Table 4: The maximum **path length**.

	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

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 Figure16.

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 4 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. The MIS, and D3P path lengths are longer in both cases.

Vertical path lengths cannot be explained as the path lengths of equivalent contraction kernels (ECK) towards the root (the apex) of the pyramid. Figure 17 shows the vertical path lengths

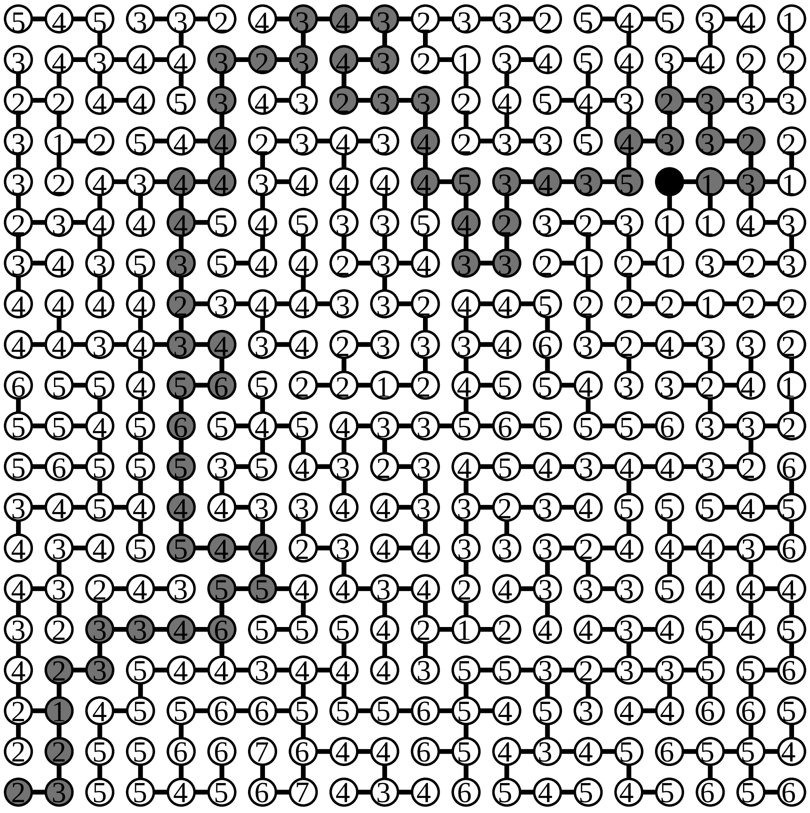


Figure 17: Path length and ECK.

(number inside the vertices) and the ECK at the top of the pyramid (the edges). As can be seen vertical path length is not the same as the number of steps (edges) needed to arrive from a vertex (any of them) to the apex (shown with black). In Figure 17 the corner vertex in gray will need 59 steps to arrive to the apex, which is quite large in comparison with 2 which is the vertical path length. So using ECK will imply larger costs to access the pixel at the base level.

To summarize, MIS and MIDES tend to find shorter paths. Longer path lengths imply bigger costs to access a vertex in the base from the apex of the pyramid.

8 Conclusion

Experiments with stochastic decimation using maximal independent vertex sets (MIS) showed a problematic behavior on large images. After an initial phase of strong reduction, the reduction decreases dramatically. This is due to the evolution of larger and larger variations between the vertex degrees in the contracted graphs. To overcome this problem we proposed a method, MIES, based on matchings which guarantees a reduction factor of 2.0. As in the case of independent vertex sets, the method based on matchings does not allow to control the directions of the contractions. The second method, MIDES, that we proposed and tested is based on directed edges and allows to control the directions of the contractions. The experiments showed a non-decreasing reduction that was even stronger than the one obtained from the method based on matchings. Path length algorithm was proposed to measure the structural complexity in irregular pyramids. The proposed methods showed also the smallest path length. Future work will focus on using MIES and MIDES for adaptive pyramids and on understanding and proving the good performance of the method based on directed edges.

9 Acknowledgment

This work was supported by the Austrian Science Foundation under grant P14445-MAT and P14662-INF. We would like to thank Roland Glantz, Georg Langs and Mamaar Saib for fruitfully discussions.

References

[BCR90] M. Bister, J. Cornelis, and Azriel Rosenfeld. A critical view of pyramid segmentation algorithms. *Pattern Recognition Letters*, 11(9):605–617, 1990.

[BK93] 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.

[BK99] Mark J. Burge and Walter G. Kropatsch. A minimal line property preserving representation of line images. *Computing*, 62:355 – 368, 1999.

[Chr75] N. Christofides. *Graph theory - an algorithmic approach*. Academic Press, New York, 1975.

[Die97] Reinhard Diestel. *Graph Theory*. Springer, New York, 1997.

[GEK99] 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.

[GK00a] 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.

[GK00b] Roland Glantz and Walter G. Kropatsch. Plane embedding of dually contracted graphs. In G. Borgefors, I. Nyström, and G. Sanniti di Baja, editors, *Discrete Geometry for Computer Imagery, DGCI'2000*, volume 1953 of *Lecture Notes in Computer Science*, pages 348–357, Uppsala, Sweden, 2000. Springer, Berlin Heidelberg, New York.

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

[Jol01] Jean-Michel Jolion. Data driven decimation of graphs. In *Proc. 3th IAPR Int. Workshop on Graph based Representation*, pages 105–114, Capri, Italy, 2001.

[Jol02] Jean-Michel Jolion. Stochastic pyramid revisited. *Pattern Recognition Letters*, accepted, 2002.

[JR94] J.M. Jolion and Azriel Rosenfeld. *A Pyramid Framework for Early Vision*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1994.

[KB98] Walter G. Kropatsch and Mark Burge. Minimizing the Topological Structure of Line Images. In Adnan Amin, Dov Dori, Pavel Pudil, and Herbert Freeman, editors, *Advances in Pattern Recognition, Joint IAPR International Workshops SSPR'98 and SPR'98*, volume Vol. 1451 of *Lecture Notes in Computer Science*, pages 149–158, Sydney, Australia, August 1998. Springer, Berlin Heidelberg, New York.

[KG01] Paul Kammerer and Roland Glantz. Using Graphs for Segmenting Crosshatched Brush Strokes. In Jean-Michel Jolion, Walter G. Kropatsch, and Mario Vento, editors, *3nd IAPR-TC-15 Workshop on Graph-based Representation*, pages 74–83. CUEN, 2001.

[KLB99] 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.

[KM91] Walter G. Kropatsch and Annick Montanvert. Irregular pyramids. Technical Report PRIP-TR-5, Technical University of Wien, Dept. Pattern Recognition and Image Processing 183/2, TU Wien, Austria, 1991.

[KM95] 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.

[Kro95] Walter G. Kropatsch. Building Irregular Pyramids by Dual Graph Contraction. *IEE-Proc. Vision, Image and Signal Processing*, 142(6):366 – 374, 1995.

[Mee89] Peter Meer. Stochastic image pyramids. *CVGIP*, 45:269 – 294, 1989.

[MMR91] 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.

[MN98] M. Matsumoto and T. Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8, No. 1:3–30., January, 1998.

[MN99] K. Mehlhorn and S. Näher. *The LEDA Platform of Combinatorial and Geometric Computing*. Cambridge University Press, Cambridge, U.K., 1999.

[Ros84] Azriel Rosenfeld, editor. *Multiresolution Image Processing and Analysis*. Springer, Berlin, 1984.

[Ros87] Azriel Rosenfeld. Pyramid algorithm for efficient vision. Technical Report CAR-TR-299, University of Maryland, Computer Science Center, 1987.

[SHG02] Maamar Saib, Yll Haxhimusa, and Roland Glantz. *Dgc_tool*: Building irregular graph pyramid using dual graph contraction. Technical Report TechRep. No.69, PRIP, Vienna University of Technology, 2002.

[TS92] K. Thulasiraman and M.N.S. Swamy. *Graphs: Theory and Algorithms*. J. Wiley & Sons, New York, USA, 1992.