Removing Redundancies in Binary Images*

Majid Banaeyan[1], Darshan Batavia[1], and Walter G. Kropatsch[1]

TU Wien, Pattern Recognition and Image Processing Group 193/03, Austria
{majid,darshan,krw}@prip.tuwien.ac.at

Abstract. Every day a huge amount of digital data is generated. Processing such big data encourages efficient data structure and parallelized operations. In this regard, this paper proposes a graph-based method reducing the memory requirement of the data storage. Graphs as a versatile representative tool in intelligent systems and pattern recognition may consist of many nonessential edges accumulating memory. This paper defines the structure of such redundant edges in the neighborhood graph of a 2D binary image. We introduce a novel approach for contracting the edges that simultaneously assists in determining the structurally redundant edges. In addition, finding a set of independent edges, the redundant edges are removed in parallel with the complexity $\mathcal{O}(1)$. Theoretically, we prove that the maximum number of redundant edges is bounded by half of all edges. Practical results show the memory requirement decreases significantly depending on the input data in different categories of binary image data sets. Using the combinatorial map as the data structure, first the topological structure of the graph is preserved. Second, the method can be extended to higher dimensions (nD).

Keywords: Redundant Edges· Connected Component Labeling · Binary Image · Combinatorial Map.

1 Introduction

The amount of available data in intelligent systems has increased dramatically in recent years [19, 20], and this situation will continue to become more extreme with the development of technologies [7, 10]. Such circumstances necessitate the development of sophisticated schemes promoting better structural representation. The structure of the data helps to preserve the topology of the image and assists to achieve a compact representation of the data. This makes structure a crucial part of data analysis. The structure of data gives the information about the intrinsic relationships between the subset of the data. Helman et al. [9] stated that extraction of relevant structure helps to reduce the data storage and assists in better visualization. In their case the amount of storage required was approximately one-tenth of the actual storage required for the data. Elimination of the redundant data [18] plays a key role in achieving a compact representation of data and saving the storage memory. Besides, it largely depends upon the representation technique, the data structure used for storage of representation, the

* Supported by Pattern Recognition and Image Processing Group (PRIP), Austria.

algorithm's compliance with parallel processing, etc. This paper covers the points related to a structure preserving algorithm for binary images. More specifically we will look into the elimination of the structurally redundant data (see Section 3) with a graph based representation (see Section 2.1) using the combinatorial maps (see Section 2.3) as the data structure.

2 Motivations and Definitions

2.1 Graph-based representation

Graphs have the capabilities to represent both structured data (like images, videos, grids) as well as unstructured data (like climate data, point cloud). Narrowing down to images, graphs based representation are simple and effective. A digital image can be easily represented using a 4-adjacent neighborhood graph. Let $G = (V, E)$ be the Region Adjacency Graph (RAG) of image P where V corresponds to the vertex set and E corresponds to the edge set. The vertex $v \in V$ associates with the pixels in image P and the edge $e \in E$ connects the corresponding adjacent vertices. Let the gray-value of vertex $g(v) = g(p)$ where $p \in P$ is a pixel in the image corresponding to vertex v. Let $contrast(e)$ be an attribute of an edge $e(u, v)$ where $u, v \in V$ and $contrast(e) = |g(u) - g(v)|$. Since we are working with binary images only, the pixels (and corresponding vertices can) have either of the two values 0 and 1. Similarly the edge contrast can have only two possible values 0 and 1. The edges in the neighborhood graph can be classified into the following two categories:

Definition 1 (Zero-edge). *An edge $e \in E$ is a* ***zero-edge****, e_0, iff the contrast between its two endpoints is zero.*

Definition 2 (One-edge). *An edge $e \in E$ is a* ***one-edge****, e_1, iff the contrast between its two endpoints is one.*

The set of edges classified as e_0 is denoted as E_0 and the set of edges classified as e_1 is denoted as E_1. The edge set $E = E_0 \cup E_1$.

2.2 Image Pyramid

Image Pyramids consist of a series of successively smaller images produced from a base image. They are efficient hierarchical structures which are able to propagate local information from the base level into a global one at the top of the pyramid. Generally, two types of the pyramid, namely regular and irregular pyramid exist.

In **regular pyramids** [12] the resolution is decreased in regular steps and therefore the size of the pyramid is fixed. On the contrary, in irregular pyramids the size of the pyramid is not fixed and it is adapted to the image data. In addition, unlike the regular ones, the irregular pyramids are shift- and rotation-invariant [16] that make them useful to use in a variety of tasks, such as image segmentation [6] and object recognition. It should be noticed that the irregular image

pyramid is interpreted as the irregular graph pyramid when its pixels and the neighborhood relations between adjacent pixels correspond to the vertices and the edges of the graph, respectively.
Irregular pyramids [15, 14, 11, 13] are a stack of successively reduced graphs where each graph is constructed from the graph below by selecting a specific subset of vertices and edges. For generation of irregular pyramids, we use the two fundamental operations on graphs: edge **contraction** and edge **removal** (Fig. 1). The edge contraction operation contracts an edge connecting two vertices, and the two vertices are merged into one. All edges that were incident to the merged vertices will be incident to the resulting vertex after the operation. The edge removal operation removes an edge from the graph, without changing the number of vertices or affecting the incidence relationships of other edges.
In each level of the pyramid, the vertices and edges that disappear in a level above are called *non-surviving* and those that appear in the upper level *surviving* ones.

Definition 3 (Contraction Kernel (CK)). *A spanning tree of a connected component.*

A contraction kernel contracts the non-surviving vertices to their corresponding surviving vertex such that each connected component indicated by one surviving vertex.

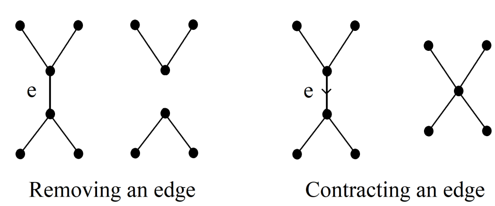


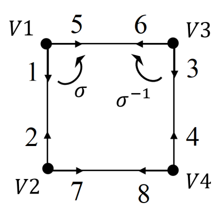
Fig. 1: Two different operations on an edge.

There are different structures to build the irregular pyramid such as simple graphs [5], dual graphs [11] and combinatorial maps (CM) [4]. In the simple graph the produced region adjacency graph (RAG) cannot distinguish between different topological configurations [13], in particular between inclusion and multiple adjacency relationships of regions [5]. The problem with dual graphs is that they cannot unambiguously represent a region enclosed in another one on a local level [5]. Therefore, in this paper the CM, as a planar embedding of RAG, is used which not only solves the mentioned problems but also provides an efficient structure that preserves topological relations between regions and can be extended to higher dimensions (nD).
A ***plane*** graph is a graph embedded in the plane such that no two edges intersect. In the plane graph there are connected spaces between edges and vertices and every such connected area of the plane is called a *face*. The *degree* of the

face is the number of edges bounding the face. In addition a face bounded by a cycle is called an *empty face*. In a non-empty face traversing the boundary would require to visit vertices or edges twice.

2.3 Combinatorial pyramid

A combinatorial pyramid is a hierarchy of successively reduced combinatorial maps. A combinatorial map (CM) is similar to a graph but explicitly stores the orientation of edges around each vertex. The combinatorial map (G) is defined by a triple $G = (D, \alpha, \sigma)$ where the D is a finite set of darts. A dart is defined as the half edge and it is the fundamental element in the CM's structure. The α is an *involution* on the set D, provides a one-to-one mapping between darts forming the same edge such that $\alpha(\alpha(d)) = d$. The σ is a *permutation* on the set D and encodes consecutive darts around the same vertex while turning counterclockwise [17]. Note that the clockwise orientation is denoted by σ^{-1}.
Fig. 2 left, shows a set of adjacent darts with their σ relations in a face of degree 4. Fig. 2, right, shows the encoding of the darts. For instance, consider $e = (1, 2)$ where $\alpha(1) = 2$, $\alpha(2) = 1$, $\sigma(1) = 5$.



Encoding of darts in Combinatorial Map

Darts (D)	1	2	3	4	5	6	7	8
α	2	1	4	3	6	5	8	7
σ	5	7	6	8	1	3	2	4

Fig. 2: Combinatorial Map.

3 Structurally Redundant Edges

The definition of the term redundant edges differs depending on the application, the representation and the data structure used for the implementation. In our case, we are dealing with binary images. In order to obtain the structure of the binary image, the relevant edges consist of a tree that spans the connected components and the edges that interconnect the components. To detect the redundant edges, it is needed to define an efficient method for selecting the CK. Note that a connected component consists of edges with zero contrast (e_0) only, and the edges with contrast one (e_1) connect two different connected components together. Therefore, in a binary neighborhood graph, the contraction kernel is selected among only e_0s.

3.1 Selecting the Contraction Kernel

Selecting the contraction kernel (CK) has a key role in detecting the redundant edges in the neighborhood graph. To this aim, a totally ordered set is defined over the indices of vertices. Consider the binary image has M rows and N columns such that $(1, 1)$ is the coordinate of the pixel $(p \in P)$ at the upper-left corner and (M, N) at the lower-right corner. The corresponding 4-adjacent neighborhood graph of the binary image has MN vertices. An index $Idx(.,.)$ of each vertex is defined:

$$Idx : [1, M] \times [1, N] \mapsto [1, M \cdot N] \subset \mathbb{N} \tag{1}$$

$$Idx(r, c) = (c - 1) \cdot M + r \tag{2}$$

Where r and c are the row and column of the pixel, respectively. Fig.3 shows the neighborhood graph of a 7 by 7 binary image where indices are from 1 to 49. Since the set of integers is totally ordered each vertex has a unique index. The important property of such totally ordered set is that every subset has exactly one minimum and one maximum member (integer number). This property provides a unique orientation between non-surviving and surviving vertices.

Consider a non-surviving vertex v. In order to find the surviving vertex, v_s, an incident e_0 must be found in its neighborhood. Such a neighborhood $\mathcal{N}(v)$ is defined as follows:

$$\mathcal{N}(v) = \{v\} \cup \{w \in V | e_0 = (v, w) \in E_0\} \tag{3}$$

if such neighborhood exists ($|\mathcal{N}(v)| > 1$) the surviving vertex is:

$$v_s = \text{argmax}\{Idx(v_s)|\ v_s \in \mathcal{N}(v), |\mathcal{N}(v)| > 1\} \tag{4}$$

Definition 4 (Orientation of a e_0). *A $e_0 = (v, w) \in E_0$ is oriented from v to w if w has the largest index among the neighbors, $Idx(w) = \max\{Idx(u)|u \in \mathcal{N}(v)\}$. All edges to the other neighbors remain non-oriented.*

By such definition, a chain of oriented e_0s connects each non-surviving vertex to its corresponding survivor vertex. In Fig. 3 the oriented e_0s are identified by an arrow over each e_0. The three vertices (25, 33 and 49) are surviving vertices while the remaining vertices are non-surviving.

Proposition 1. *Selecting the CK partitions vertices into non-surviving and surviving vertices.*

Proof. If the $|\mathcal{N}(v)| = 1$, either there is no e_0s around v or the index of v is bigger than the indices of neighboring e_0s. Therefore the v is a surviving vertex. In case the $|\mathcal{N}(v)| > 1$, since the indices are totally ordered, there is a maximum in the neighborhood of v which is selected as survivor and the v becomes the nun-surviving vertex. □

Proposition 2. *Every non-surviving vertex has a unique surviving vertex.*

Proof. Each tree of oriented e_0s has one unique maximum as the index of the surviving vertex. □

Property 1. With the choice of $Idx(.)$ and the coordinate axes in (1) a non-surviving vertex contracts either to its adjacent right vertex or to its down vertex where the right vertex has the higher priority.

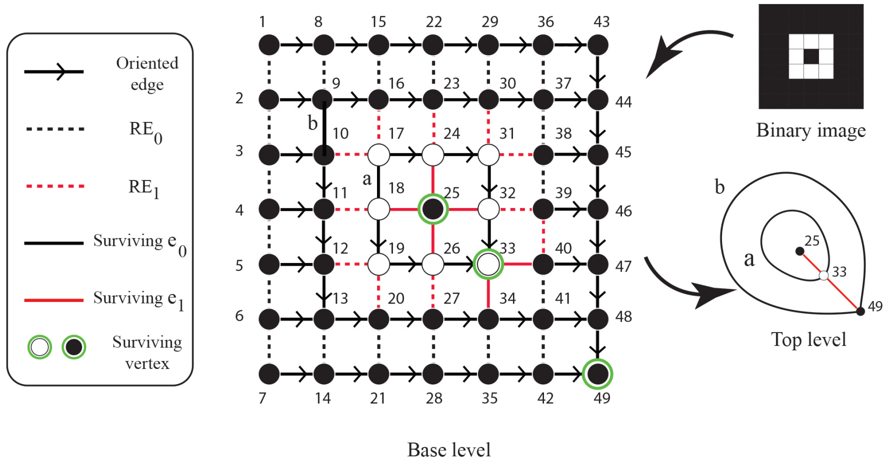


Fig. 3: Combinatorial Map.

3.2 Redundant Edges

Connectivity is an essential property in the structure of a hierarchical graph pyramid. Nevertheless, there may be some edges the removal of which does not harm the connectivity. We define such edges as *redundant edges*.

Definition 5 (Redundant-Edge (RE)). *In an empty face, the non-oriented edge incident to the vertex with lowest Idx is redundant iff:*

- *The empty face is bounded by only non-oriented edges with the same contrast value.*
- *The empty face is bounded by non-oriented edges with the same contrast value and oriented edges.*

Based on the RE definition, an empty self-loop is redundant. In addition, in an empty face of degree 2 (double edge), one of the edges is redundant. Fig. 4 illustrates an empty face of different degrees where in each empty face the redundant edge is indicated by RE.

Proposition 3. *The upper bound of the number of redundant edges (REs) is equal to half of the edges of the grid at the base level.*

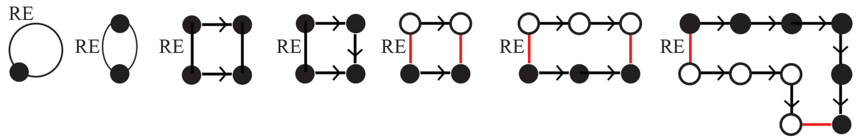


Fig. 4: Example of redundant edge (RE) in different empty faces.

Proof. In a grid M by N, the number of vertices is MN and the number of edges is $2MN - M - N$. To preserve the connectivity the smallest graph is a spanning tree of vertices with $MN - 1$ edges. Therefore, the maximum number of REs is:

$$Max|REs| = (2MN - M - N - (MN - 1)) = MN - M - N + 1 \quad (5)$$

$$\lim_{M\to\infty\ N\to\infty} (Max|REs|/E) = (MN - M - N + 1)/(2MN - M - N) = 1/2 \quad (6)$$

As the result, by growing M and N, the maximum number of REs becomes maximally half of all the edges (E) at the base level. □

Proposition 4. *In every face of degree n ($n > 1$) is bounded by only e_0s, one of the non-oriented e_0s is redundant.*

Proof. By contracting an edge, every face of degree $n > 2$ after $n-2$ consecutive contractions becomes a face of degree 2 which has a RE (definition 5). □

Proposition 5. *In every face of degree n ($n > 1$) is bounded by only e_1s and oriented e_0s, there is a redundant one-edge (RE_1) .*

Proof. Contracting all oriented e_0s results in a face of degree 2 containing two e_1s between the same endpoints. Hence, one of the e_1s is redundant. □

Since edges classify into E_0 and E_1, the REs are partitioned into *Redundant Zero-Edges* (RE_0) and *Redundant One-Edges* (RE_1) as well:

$$REs = RE_0s \ \dot{\cup} \ RE_1s \quad (7)$$

In Fig. 3, the RE_0s are shown by black dashed-lines and the RE_1s are shown by red dashed-lines. Furthermore, the RAG at the top of the pyramid shows the connections between three different connected components. Using the combinatorial map structure, the inclusion relation is preserved because it is represented by the loop **a** around the vertex 25.
It should be noted that a neighboring graph at the base level may not have any redundant edge. Consider a 4-connected graph that its vertices form a checkerboard pattern. In such the case, all edges have contrast one that it means there is no zero-edge and thus no RE_0. Furthermore, based on the proposition 5, no two one-edges connect the same vertices and thus there is no RE_1 as well.

3.3 Removing Redundant Edges in Parallel

In order to remove the REs, a dependency between edges is considered. We define such dependency relation to detect a set of REs where by simultaneously removing, the combinatorial structure is not harmed. To this aim, first a set of dependent darts is defined as follows:

Definition 6 (Dependent Darts). *All darts of a σ-orbit sharing an endpoint are dependent darts.*

Next, by considering the corresponding edge of each dart, $e = (d, \alpha(d))$, the set of dependent darts results in the set of **dependent edges**. Consequently, two edges not sharing an endpoint are independent. In this manner, the only case of the dependency between REs occurs when the REs share an endpoint. In the grid at the base level the REs may be connected horizontally or vertically and thus are dependent. However, consider a horizontal edge in an odd row of the grid. This edge is independent to all other horizontal edges of other odd rows. Similarly, a vertical edge in an odd column is independent of all other vertical edges of other odd columns. Such independency exists between edges in even rows and even columns as well. Fig. 5 shows the set of independent edges at the base. Therefore, all the edges in grid are classified to four independent classes of edges. Consequently, removing all edges belonging to each independence class

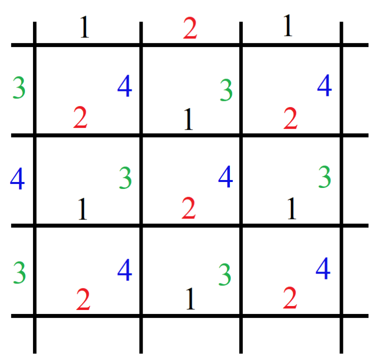


Fig. 5: The four independent classes of edges in the grid at the base.

(1, 2, 3 or 4) occurs simultaneously. This means, all the REs are removed in only four steps where each step has the complexity $\mathcal{O}(1)$. Therefore removing the redundant edges is performed in parallel.

4 Memory Consumption

The topological structure is well captured in the combinatorial map. A combinatorial pyramid is a hierarchy of successively reduced combinatorial maps [4].

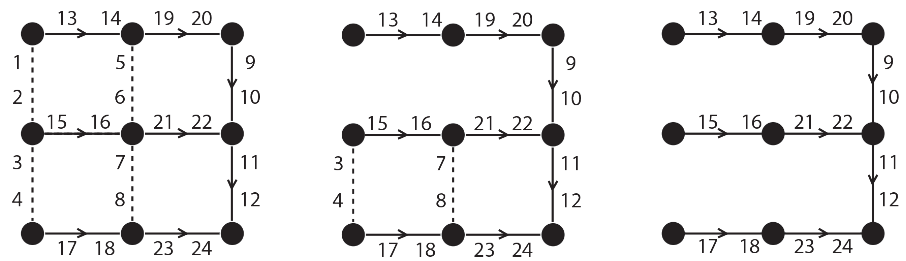
The pyramid needs to store the combinatorial map of each level that results in high memory consumption. To avoid such expensive memory requirement, we use a *canonical encoding* [17] where the memory consumption of the pyramid is equal to the size of the base level.
In the canonical encoding of the combinatorial pyramid, all the darts are stored in a single array that preserves the history of pyramid construction. The number of darts at the base level is equal to $2 \cdot (2MN - M - N)$ in a binary image M by N. Since nearly half of the edges at the base level (Proposition . 3) are redundant (RE), their removal decreases the memory requirements.
In addition, in the canonical pyramid, removing the darts performs in a sequential manner. In contrast, using the independent set of edges (Section 3.3) we are able to remove the independent set of corresponding darts in parallel. Therefore, in the canonical array such darts are removed in parallel. Fig . 6(a) illustrates the combinatorial map (CM) of a graph and Fig . 6(d) shows its canonical encoding. The REs are shown by dashed-lines. Four REs are corresponding to darts 1 to 8. The darts at the first row (1,2,5,6) are removed in one step (Fig . 6(b,f)). Afterwards, darts at the second row (3,4,7,8) are removed simultaneously (Fig. 6(c,g)). This results in, the smaller array of the canonical encoding shown in Fig. 6(h).

5 Comparisons and Results

To highlight the advantages of the proposed method, we compare the memory storage required with and without removing the REs. The comparison is done with the originally proposed canonical representation [17]. It was used by [6, 1, 3] for the implementation of topology preserving irregular image pyramids of gray scaled and RGB images. In addition, recently the canonical encoding was used in connected component labeling [2]. Since for our current research, the input images are restricted to binary images, it is easy to identify the connected components unlike the gray scale images. Considering the structure of the image, the number of REs that can be eliminated are significantly higher than that in a gray scale or RGB image.

In a combinatorial map, the involution α between the darts remain the same even after performing the contraction and/or removal operations. The α relations can be encoded into the even and odd numbering of darts for each edge. Thus all the modifications related to the contraction and the removal operation on the graphs are performed by modifying the σ-permutation. In the canonical representation, the minimum storage required to store and to modify the σ-permutation is equal to the number of darts i.e. twice the number of edges. By using the proposed method, we eliminate the edges that are structurally redundant and consequently reduce the storage space of darts and its permutation.

The algorithm was tested on several classes of images from the YACCLAB [8] dataset. Table 1 displays the outcome of the proposed method. The first column shows the name of the image class in the data set and an example from it, while the second column displays the 'size' of the image. The number of images



(a) CM of a graph

(b) Removing REs of first row

(c) Removing REs of second row

d	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
σ(d)	13	3	15	17	19	16	4	18	20	22	10	24	1	5	2	7	4	23	14	9	6	11	8	12

(d) Canonocal encoding of (a)

d	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
σ(d)	13	3	15	17	19	16	4	18	20	22	10	24	1	5	2	7	4	23	14	9	6	11	8	12

Redundant darts

(e) Redundant darts in canonical encoding of (a)

d	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
σ(d)	13	3	15	17	19	16	4	18	20	22	10	24	13	19	3	7	4	23	14	9	16	11	8	12

Redundant darts

(f) Canonical encoding of (b)

d	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
σ(d)	13	3	15	17	19	16	4	18	20	22	10	24	3	19	15	21	17	23	14	9	16	11	18	12

Redundant darts

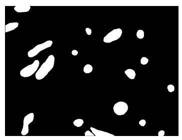
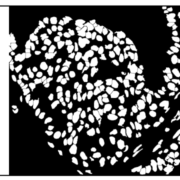
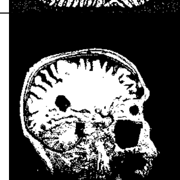
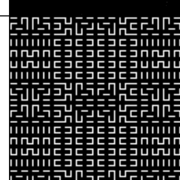
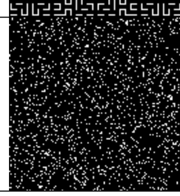
(g) Canonical encoding of (c)

d	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
σ(d)	20	22	10	24	3	19	15	21	17	23	14	9	16	11	18	12

(h) CM after removing all redundan darets

Fig. 6: Memory usage in the canonical encoding.

Table 1: Results over images of different categories from (YACCLAB[8]).

Database	Example	Size	#Images	$\lvert V_s\rvert / \lvert V\rvert$	$\lvert RE_{min}\rvert$	$\lvert RE_{\mu}\rvert$	std($\lvert RE\rvert$)
Mitochondria		768×1024	495	0.33%	49.01%	49.44%	0.0073
Medical		890×886	189	2.74%	41.18%	46.87%	0.0145
Finger-print		300×300	962	3.50%	42.50%	46.05%	0.0108
MRI		256×256	1170	2.72%	44.42%	46.49%	0.0114
3dpes		704×576	2400	0.07%	49.81%	49.84%	0.0019
Hilbert		127×127	512	2.43%	41.31%	45.25%	0.0108
Random		64×64	89	18.90%	23.18%	27.66%	0.0407

('#Images') from each class, on which the implementation was performed is displayed in the third column. The forth column gives the percentage of vertices that survive ('$|V_s|/|V|$'). Since there is a significant variation in the size of the image, the number of REs are expressed in terms of percentage of the actual number of edges. The last three columns display the lowest ('$|RE_{min}|$'), and the average number of REs ('$|RE_\mu|$') along with the standard deviation ('std($|RE|$)') over all images from each dataset.

The redundancy in the random images is notably lower than that in the other class of images. This can be observed in the number of surviving vertices as well. This happens due to the fact that the number of isolated vertices (vertices surrounded by e_1s only) are higher, making the connected component smaller in size. The worst case occurs in a checkerboard pattern where all the vertices are isolated making each region containing a single pixel. In such a case, none of the edges are redundant. In contrast, an image with only black (0) or only white (1) color will have 50% of the REs.

6 Conclusion and Future Works

The paper presents a new formalism to define redundant edges in the neighborhood graph of a 2D binary image. By proposing the new method for selecting the contraction kernels these redundant edges are efficiently detected and removed *before* the contraction operation. We prove that the amount of redundant edges may reach up to half of the edges at the base level with a grid like structure. The experiments show that most classes of images have 45%-49% of redundant edges (except for artificially generated random binary images). As a result, the memory consumption is reduced by 45%-49% while using combinatorial map as the data structure. Furthermore, all the redundant edges can be removed in parallel with a constant algorithmic complexity $\mathcal{O}(1)$. For the future work, we are going to develop the method for gray-scale images. Secondly, by using the combinatorial structure we will work on extending the redundant edges to higher dimensions (nD).

References

1. Banaeyan, M., Huber, H., Kropatsch, W.G., Barth, R.: A novel concept for smart camera image stitching. In: Čehovin, L., Mandeljc, R., Štruc, V. (eds.) Proceedings of the 21st Computer Vision Winter Workshop 2016. pp. 1–9 (2016)
2. Banaeyan, M., Kropatsch, W.G.: Pyramidal connected component labeling by irregular graph pyramid. In: 2021 5th International Conference on Pattern Recognition and Image Analysis (IPRIA). pp. 1–5 (2021). https://doi.org/10.1109/IPRIA53572.2021.9483533
3. Batavia, D., Gonzalez-Diaz, R., Kropatsch, W.G.: Image = structure + few colors. In: Structural, Syntactic, and Statistical Pattern Recognition - Joint IAPR International Workshops, S+SSPR 2020, Padua, Italy, January 21-22, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12644, pp. 365–375. Springer (2020)

4. Brun, L., Kropatsch, W.: Introduction to combinatorial pyramids. In: Digital and Image Geometry. pp. 108–128. Springer (2001)
5. Brun, L., Kropatsch, W.G.: Hierarchical graph encodings. In: Lézoray, O., Grady, L. (eds.) Image Processing and Analysis with Graphs: Theory and Practice, pp. 305–349. CRC Press (2012)
6. Cerman, M., Janusch, I., Gonzalez-Diaz, R., Kropatsch, W.G.: Topology-based image segmentation using LBP pyramids. Machine Vision and Applications pp. 1–14 (2016)
7. Chen, L., Zheng, L., Xia, D., Cai, X., Sun, D., Liu, W.: Recognizing and analyzing private car commuters using big data of electronic registration identification of vehicles. IEEE Transactions on Intelligent Transportation Systems pp. 1–15 (2022). https://doi.org/10.1109/TITS.2022.3142778
8. Grana, C., Bolelli, F., Baraldi, L., Vezzani, R.: YACCLAB - Yet Another Connected Components Labeling Benchmark. In: 2016 23rd International Conference on Pattern Recognition (ICPR). pp. 3109–3114. Springer (Dec 2016). https://doi.org/https://doi.org/10.1109/ICPR.2016.7900112
9. Helman, J., Hesselink, L.: Visualizing vector field topology in fluid flows. IEEE Computer Graphics and Applications **11**(3), 36–46 (1991). https://doi.org/10.1109/38.79452
10. Huang, W., Liang, H., Lin, L., Wang, Z., Wang, S., Yu, B., Niu, R.: A fast point cloud ground segmentation approach based on coarse-to-fine markov random field. IEEE Transactions on Intelligent Transportation Systems pp. 1–14 (2021). https://doi.org/10.1109/TITS.2021.3073151
11. Kropatsch, W.G.: Building irregular pyramids by dual graph contraction. IEE-Proc. Vision, Image and Signal Processing **Vol. 142**(No. 6), pp. 366–374 (1995)
12. Kropatsch, W.G., Leonardis, A., Bischof, H.: Hierarchical, adaptive and robust methods for image understanding. Surveys on Mathematics for Industry **No. 9**, pp. 1–47 (1999)
13. Kropatsch, W.G., Macho, H.: Finding the structure of connected components using dual irregular pyramids. In: Cinquième Colloque DGCI. pp. 147–158. LLAIC1, Université d'Auvergne, ISBN 2-87663-040-0 (1995)
14. Kropatsch, W.G., Reither, C., Willersinn, D., Wlaschitz, G.: The dual irregular pyramid. In: Chetverikov, D., Kropatsch, W.G. (eds.) Proceedings CAIP'93, Computer Analysis of Images and Patterns. vol. Vol. 719, pp. 31–40. Springer (1993)
15. Meer, P.: Stochastic image pyramids. Computer Vision, Graphics, and Image Processing **45**(3), 269–294 (1989)
16. Montanvert, A., Meer, P., Rosenfield, A.: Hierarchical image analysis using irregular tessellations. IEEE transactions on pattern analysis and machine intelligence **13**(4), 307 (1991)
17. Torres, F., Kropatsch, W.G.: Canonical encoding of the combinatorial pyramid. In: Proceedings of the 19th Computer Vision Winter Workshop. pp. 118–125 (2014)
18. Wang, C., Wang, K., Liu, W.: Personalized recommendation via enhanced redundant eliminated network-based inference. In: 2019 First International Conference on Digital Data Processing (DDP). pp. 1–6 (2019). https://doi.org/10.1109/DDP.2019.00011
19. Wang, N., Haihong, E., Song, M., Wang, Y.: Construction method of domain knowledge graph based on big data-driven. In: 2019 5th International Conference on Information Management (ICIM). pp. 165–172 (2019). https://doi.org/10.1109/INFOMAN.2019.8714664

20. Zhang, D., Jiang, Y.: Design of urban intelligent traffic congestion situation monitoring system based on big data. In: 2020 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS). pp. 12–15 (2020). https://doi.org/10.1109/ICITBS49701.2020.00011