



# Image = Structure + Few Colors

Darshan Batavia<sup>1</sup>(✉), Rocio Gonzalez-Diaz<sup>2</sup>, and Walter G. Kropatsch<sup>1</sup>

<sup>1</sup> TU Wien, Pattern Recognition and Image Processing Group 193/03,  
Vienna, Austria

{darshan,krw}@prip.tuwien.ac.at

<sup>2</sup> Applied Math I, University of Seville, Seville, Spain  
rogodi@us.es

**Abstract.** Topology plays an important role in computer vision by capturing the structure of the objects. Nevertheless, its potential applications have not been sufficiently developed yet. In this paper, we combine the topological properties of an image with hierarchical approaches to build a topology preserving irregular image pyramid (TIIP). The TIIP algorithm uses combinatorial maps as data structure which implicitly capture the structure of the image in terms of the critical points. Thus, we can achieve a compact representation of an image, preserving the structure and topology of its critical points (maxima, the minima and the saddles). The parallel algorithmic complexity of building the pyramid is  $O(\log d)$  where  $d$  is the diameter of the largest object. We achieve promising results for image reconstruction using only a few color values and the structure of the image, although preserving fine details including the texture of the image.

## 1 Introduction

Critical points and curves connecting them are an effective means of communicating topological information, which governs the structure of an image. An image compact representation can be achieved by using a surface topology based data structure as mentioned in [11]. For example, Nackman in [17] represented a surface in form of graphs of critical points, subdividing the surface into *slope districts*. Other approaches include: Reeb graphs [21], hierarchical decomposition of Morse-Smale complexes into piecewise linear 2-manifolds [9].

Compact representation has become necessary with the increase of digital data, resulting in down-sampling for compactness and up-sampling for surface reconstruction/approximation. Gaussian and Laplacian pyramids [6] are the most basic regular image pyramid methods. Irregular sampling and image pyramids are an excellent tool for topological representation. Cerman *et al.*, in [7], developed a topology-based image segmentation algorithm. In [18], Maia *et al.* use hierarchical watershed for image segmentation. The most common method for topology simplification using graph representation is by repeated application of the fundamental operation *edge contraction* [12]. Simplification of data is broadly divided into two major types: the topological simplification and the geometric simplification. Persistence measure is the most famous technique used

in 2D to decide the priority for simplification [4, 10]. For example, in [28], an efficient image down-sampling and up-sampling technique based on interpolation is developed. Other approaches include [8] and [13]. Besides down-sampling, up-sampling is equally important for applications like image super-resolution, image enhancement and denoising. In [22], the anchored neighborhood regression for learning-based super-resolution is used. Other related researches related to image super-resolution using simple functions are [25] and [26].

**Motivation and Contribution:** In this paper, we propose a *Topology-preserving Irregular Image Pyramid (TIIP)* algorithm and a hierarchical method to build an irregular image pyramid that preserves the critical points and their connections. For a surface, the topology of its contours changes at the function values of its critical points. For example, the surface contours will collapse to a point at a non-degenerated extremum and multiple contours will intersect at a saddle point. Therefore, we preserve the critical points and their function values to preserve the topology of an image. The algorithm operates on combinatorial maps [5] which implicitly encode the structure of the image on the higher level of the pyramid with a compact representation. The use of combinatorial maps supports parallel processing [23] with time complexity of  $O(\log(d))$ , where  $d$  is the diameter of the longest object in the image. The approach is reinforced by a concrete theory of cellular decomposition into cells called slope regions<sup>1</sup> [2, 15]. The TIIP algorithm is explained in two distinct algorithms: (a) the bottom-up construction of the pyramid (*REDUCE*) and (b) the top-down expansion of the higher level of the pyramid (*EXPAND*), terminologies as introduced in [6]. The TIIP algorithm achieved a perceptually superior reconstruction especially in the focused region and preserved the texture information maintaining structure and features similar to the original image. Thus an image can be efficiently reconstructed by its structure and a few colors.

This paper is organized as follows: Sect. 2 introduces the basic definitions and terminologies required together with the proposed TIIP algorithm. In Sect. 3, we show results for image reconstruction and its comparison with other algorithms. In Sect. 4, we end with some conclusions and future work.

## 2 Reduce and Expand Operations in the TIIP Algorithm

We introduce now a method to efficiently build an irregular image pyramid which preserves structural and topological information. We first explain the bottom-up REDUCE operation (TIIP Algorithm 1) for building the topology preserving image pyramid. Later, the top-down EXPAND operation is explained, depending on the application: (1) image segmentation and (2) image recovery.

A discrete 2D image  $P$  where the gray value of a pixel  $p$  is denoted by  $g(p)$ , can be represented as a 4-neighborhood graph  $G_0(V, E)$ . The labels of a segmentation are often stored in form of a label image where each region has a distinct

---

<sup>1</sup> Slope regions are the surfaces in which every pair of points can be connected by a monotonic curve.

label. The adjacencies of the regions are described by the *region adjacency graph* (RAG)  $G = (V, E)$ . Taking every pixel of the image as a (smallest) region, the neighborhood graph can also be interpreted as a RAG. Every pixel  $p$  in the image  $P$  corresponds to a vertex  $v \in V$  with gray value  $g(v) := g(p)$ . Vertex  $v$  is connected to its adjacent vertices<sup>2</sup> by edges of  $G$ , being the degree of  $v$ , denoted by  $\text{degree}(v)$ , the number of edges incident to  $v$ . An edge  $e \in E$  is a *boundary edge* if  $e$  is in the border of the unbounded face. The endpoints of a boundary edge are *boundary vertices*. The *orientation of an edge*  $e = (v, w) \in E$  is directed from vertex  $v \in V$  to vertex  $w \in V$  iff  $g(v) > g(w)$ . When  $g(v) = g(w)$ , the edge  $e$  is not oriented. The *weight* of edge  $e$  is its contrast  $g(v) - g(w)$ . A connected subgraph of  $G$  having the same gray value for all the vertices is referred to as a *plateau region* where every pair of vertices  $v, w \in V$  of the subgraph satisfies  $g(v) = g(w)$ . A *path*  $\pi(v_1, v_2, \dots, v_r) = (V_\pi, E_\pi)$  is a non empty subgraph of  $G$ , where  $V_\pi = \{v_1, v_2, \dots, v_r\} \subseteq V$  and  $E_\pi = \{(v_1, v_2), (v_2, v_3), \dots, (v_{r-1}, v_r)\} \subseteq E$ . The path  $\pi$  is *monotonic* if all the oriented edges of  $E_\pi$  have the same orientation, e.g. from  $v_1$  to  $v_r$  or from  $v_r$  to  $v_1$ . The path  $\pi$  is a *level curve* if  $g(v_i) = g(v_{i+1})$  for all  $i$ . A level curve can be a part of monotonic paths. A face is a *slope region* if the edges in its border can be divided in two disjoint sets forming two monotonic paths with same orientation.

The orientation of edges can be used to categorize a vertex  $v \in V$  into critical (maximum, minimum, saddle), or non-critical (slope). A vertex  $v \in V$  is a *local maximum* (*2-max*) if all the edges incident to  $v$  are oriented outwards. Analogously, a boundary vertex  $v \in V$  is a local boundary maximum (*1-max*) if the two boundary edges incident to  $v$  are oriented outwards. A vertex  $v \in V$  is a *local minimum* (*2-min*) if all the edges incident to  $v$  are oriented inwards. Analogously, a boundary vertex  $v \in V$  is a local boundary minimum (*1-min*) if the two boundary edges incident to  $v$  are oriented inwards. A vertex  $v \in V$  is a *degenerated critical vertex* if all the edges incident to  $v$  are non-oriented. A non-boundary vertex  $v \in V$  is a slope vertex if there are exactly two changes in the orientation of edges incident to  $v$ , when traversed circularly (clockwise or counter-clockwise direction). In this case, a slope vertex is a *singular slope vertex* if all the oriented edges incident to  $v$  have the same orientation except one. Otherwise, it is a *regular slope vertex*. Observe that degenerated critical vertices always belong to plateau regions. A boundary vertex that is neither a local boundary maximum nor a local boundary minimum is considered a singular slope vertex. A vertex  $v \in V$  is a *saddle vertex* if it is not a local maximum, nor a local minimum, neither a plateau nor a slope vertex.

**REDUCE Operation.** It basically comprises edge contraction [14] and edge removal operations on the graph  $G_0$ , forming a graph pyramid. Significance of steps in Algorithm 1 (the enumerations are correlated to the algorithm step numbers).

---

<sup>2</sup>  $\mathcal{N}(v)$  denotes the set of vertices adjacent to  $v$ .

---

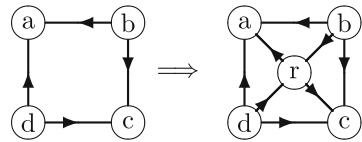
**Algorithm 1.** Topology preserving Irregular Image Pyramid

---

- 1: **Input:** A digital image  $P$ .
  - 2: **Initialize:** Generate the 4-connected neighborhood graph  $G_0$ .
  - 3: Insertion of hidden saddle vertices.
  - 4: Contraction of level curves connecting boundary vertices.
  - 5: Contraction of plateau regions.
  - 6: Categorising the vertices into critical and non-critical.
  - 7: Replicating regular slope vertices.
  - 8: **while** #non-critical vertices > 0 **do**
  - 9:   **if** #regular slope vertices > 0 **then**
  - 10:     replicate regular slope vertices.
  - 11:   **end if**
  - 12:   I. **Select contraction kernel** giving priority to edges with lowest weight.
  - 13:   II. **Decide surviving vertices** for edges connecting:
    - (a) non-critical - non-critical vertices.
    - (b) critical - non-critical vertices.
  - 14:   **Edge contraction**
  - 15:   **Reduction function:** Recomputing the edge weights.
  - 16:   **Simplification:** Edge removal such that the resulting region is a slope region.
  - 17: **end while**
  - 18: Contraction of edges connecting critical vertices (if required).
  - 19: **end**
- 

**3. Insertion of Hidden Saddle Vertices:**

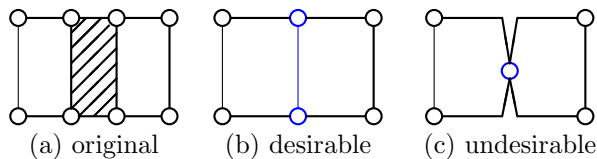
If  $\pi = (V_\pi, E_\pi)$  being  $V_\pi = \{a, b, c, d\}$  and  $E_\pi = \{(a, b), (b, c), (c, d), (d, a)\}$ , is a closed path of  $G_0$ , satisfying that  $g(a) < g(b)$ ,  $g(a) < g(d)$ ,  $g(c) < g(b)$  and  $g(c) < g(d)$ , then a new vertex  $r$  called *hidden saddle vertex* together with edges  $(r, a), (r, b), (r, c), (r, d)$  are added to the RAG (see Fig. 1) with a new gray value  $\max(g(a), g(c)) < g(r) < \min(g(b), g(d))$  (see [7, 16]).



**Fig. 1.** Inserting a hidden saddle.

**4. Contraction of Level Curves Connecting Boundary Vertices:**

The shaded region in Fig. 2.a exhibits a plateau region. It connects two disconnected parts of the image boundary. In such cases, contracting the plateau region will result into partitioning the image into two regions connected through a single cut vertex as shown in Fig. 2.c. This result is undesirable because the connections of all the vertices through the plateau region is now concentrated in a single vertex and then the disconnected parts of the image boundary are now connected. To avoid such undesirable occurrences,



**Fig. 2.** A plateau region and two contractions of it.

is undesirable because the connections of all the vertices through the plateau region is now concentrated in a single vertex and then the disconnected parts of the image boundary are now connected. To avoid such undesirable occurrences,

we first contract the level curves on the boundary and then proceed by contracting the plateau region not allowing any two remaining vertices on the boundary to be contracted. The resulting level, curve showed in Fig. 2.b, preserves the connection through the plateau region.

**5. Contraction of Plateau Regions:** In this step (which is similar to the superpixel hierarchy), we cluster the vertices with the same attribute (gray value) and represent them by a single vertex. Details of the contraction of plateau regions spread across the image are explained in [3, Section 2.1]. In the case of binary images and segmented images, this step usually contains the maximum number of edge contractions and will result in a connected component labelling.

**6. Categorising the Vertices into Critical and Non-critical:** To preserve the topological properties, it is important to categorize the vertices into critical (local maximum, local minimum, saddle) and non-critical (slope) vertices. This step is executed after the contraction of plateau regions so that we avoid the misclassification of the degenerated critical vertices. We use the orientation of the edges incident to a vertex to categorize it as mentioned in Sect. 2.

**7. Replicating the Regular Slope Vertices:** The replication operation of a regular slope vertex  $v$  consists of an edge de-contraction [23], i.e., replacing  $v$  by an edge  $e$  with endpoints  $x$  and  $y$  having the same gray value than  $v$  (the edges incident to  $v$  being now incident to one of the two endpoints of  $e$ ) followed by the removal of  $e$ . After the replication operation of  $v$ , the two vertices  $x$  and  $y$  are also slope vertices with lower degrees. The process is repeated until all regular slope vertices become singular slopes vertices.

**12. I. Select the Contraction Kernel:** With combinatorial maps as the data structure, we can contract multiple edges in parallel, as a result of which we can have larger contraction kernels consequently lowering the height of the pyramid. The constraints for computation of the contraction kernels is application dependent. For the contraction process, this paper gives priority to the edges with the lower edge weight. This constraint can be included in the computation of the contraction kernel. Contraction of edges with lower weights preserves the high frequency components of the data preserving sharp edges and contours.

**12. II. Decide Surviving Vertices:** Below are the few aspects which we take into account before the edge contraction operation:

- (a) To guarantee that critical vertices are preserved and that the vertices do not change from critical to non-critical or vice-versa, an edge is selected for contraction if it is incident to a singular slope vertex  $v$  and have a different orientation than the rest of the edges incident to  $v$ .
- (b) Before contraction of an edge connecting a critical and a non-critical vertex, we mark the critical vertex as the surviving vertex and the non-critical vertex as the non-surviving vertex.
- (c) Before contraction of an edge connecting two non-critical vertices, decision of the surviving and the non-surviving vertex is rather application dependent. For example, in the case of Connected Component Labelling (CCL) of

a binary image, it is not significant to categorize the surviving and non-surviving vertices. In contrast, for visualization and analysis, the vertex closer to the centroid (visual center for concave polygon) would be the surviving vertex. As a result the vertex corresponding to the centroid will represent the segment.

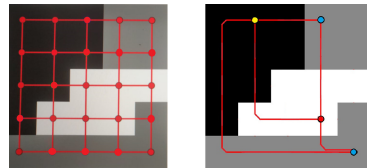
**13. Edge Contraction:** In this step, we contract the edges selected in step 12 of Algorithm 1. Given an edge  $e$  connecting two vertices  $s, n$  of a RAG  $G_k = (V_k, E_k)$ , the contraction of  $e$  will result in merging the survivor vertex  $s$  and the non-survivor vertex  $n$  (see [14]). After the contraction of  $e$ , all the edges, except  $e$ , previously incident to vertex  $n$  will now be incident to vertex  $s$  in graph  $G_{k+1}$ .

**14. Reduction Function:** It computes the weights of the edges previously connected to the non-surviving vertices. While contracting an edge  $e = (s, n)$ , where  $s$  is a surviving vertex and  $n$  is a non-surviving vertex, if  $s$  is critical, we preserve the gray value of  $s$ ; if  $s$  is non-critical, we compute the gray value of  $s$  by  $g(s): \max\{g(n) \leq g(s) | n \in \mathcal{N}(s)\} \leq g(s) < \min\{g(n) > g(s) | n \in \mathcal{N}(s)\}$ .

**15. Simplification:** The simplification operation removes the redundant edges (merging the two respective faces sharing the same edge) which leads to an increase in the degree of the faces in the RAG. The empty self-loops<sup>3</sup> are always removed. In [3], a more generalized version of of this step is shown. Edge removal simplifies the graph for visualization and also eliminates the redundant information. We do not remove boundary edges to preserve the image boundary.

**17. Contraction of Edges Connecting Critical Vertices:** Unlike edge contraction mentioned in steps 4, 5 and 13 of Algorithm 1, in this step we contract the edges connecting the critical vertices. It is usually observed in a natural image that approximately 30% of the total vertices at the base level of the image pyramid are critical. For further reduction in information, we need to contract edges connecting critical vertices, and this may affect the topology of the data. The selection criteria for the contraction kernel depends on the application. For example: if the given image is noiseless and smooth and the application is segmentation, then the contraction of edges with lower weight is preferred. In contrary, if the input is a natural image with salt and pepper noise, then the edges with higher weight are preferred. Topological persistence can be used for the selection process [10].

**Graph of Critical Vertices Only:** All the faces in a RAG are slope regions. Besides, faces after edge contraction, edge removal and vertex replication continue to be slope regions. At the end of the process, only critical vertices survive as the following result states. See the naive example showed in Fig. 3.



**Fig. 3.** Base and top level of a pyramid.

<sup>3</sup> An empty self-loop is a self-loop that does not encapsulate a subgraph inside.

**Theorem 1.** *Algorithm 1 (without step 17) computes a graph pyramid with the property that all the vertices on the top are critical and the number of critical vertices is constant along the pyramid after plateaus are contracted.*

Proof. Let us prove that critical vertices are preserved in the all pyramid. Basically, there are three steps applied recursively to compute the graph pyramid, namely, regular slope vertices replication, edge contraction (step 13) and simplification. Let us see that these operations do not modify the critical vertices. With edge contraction, the number of slope vertices is reduced, and hence, critical vertices are not modified. With simplification, which is an edge removal between two vertices which are not both critical, the degree of the two endpoints of the removed edge decreases, with the possibility that such vertices change their category to singular slope vertices and they can be removed in the next step in the contraction process. With the replication process, which is applied on regular slope vertices only and it does not modify critical vertices, a regular slope vertex is divided in two slope vertices. It increases the number of slope vertices but reduces their degrees (recall that the incident edges connected to the regular slope vertex before the replication process are later distributed between the two new slope vertices). Therefore a new singular slope can show up and be removed in the next contraction step. Observe that given a regular slope vertex  $v$ , if  $n$  edges incident to  $v$  are oriented inwards and  $m$  edges incident to  $v$  are oriented outwards with  $n < m$ , then the replicating process can be applied  $n - 1$  times to obtain  $n$  singular slope vertices that are contracted in the next step. Let  $F_k$  be the number of edges of  $G_k$  with at least one endpoint being non-critical. Observe that to obtain  $G_{k+1}$ , the replication process does not increase  $F_k$  whereas the contraction process reduces  $F_k$  to one unit for each singular slope vertex. Therefore,  $F_{k+1} < F_k$  and the process finishes when there are no edges in the graph with at least one endpoint being non-critical, then the graph has only critical vertices and they are the original critical vertices because the applied operations did not modify them during all the process.  $\square$

**EXPAND Operation:** In contrast to the REDUCE operation, in the EXPAND operation, we project the information from the higher level to the lower level of the pyramid. After eliminating the inserted hidden saddles (introduced in step 3 of Algorithm 1), the RAG can be reconverted into a reconstructed image. For the purpose of image reconstruction, we aim to reconstruct a perceptually superior image with a high structural similarity as compared to the original image. For the EXPAND operation, we define edge de-contraction and insertion [7] which are the inverse operations for edge contraction and removal used in the REDUCE operation. The top level of the pyramid conceives all the critical vertices (as proved in Theorem 1) and the monotonic paths connecting critical vertices. Knowing the fact that the monotonic paths are bounded by the critical vertices, the monotonic path connecting two critical vertices can be interpolated between the intensity range of the two critical vertices for promising reconstructed images. To further reduce the error, the weights of the contracted edges can be stored externally for a perfect reconstruction of an image as explained

in Burt’s Laplacian pyramid [6]. Certainly there is a trade-off between accuracy and required memory.

### 3 Experiments and Results

In this section we analyse the performance of our TIIP algorithm for image reconstruction and compare it with the state-of-art CNN models for image super-resolution. Since super resolution algorithms are tested by first down-scaling the image through the regular pyramids and then up-scaling, it was closest to the application of image reconstruction mention in this paper. Hence the TIIP algorithm is compared with the super resolution algorithms. The methods for comparison include RCAN [29], DRLN+ [1], A+ANR [22], and EnhanceNet [20]. For evaluations of a (regular grid based) down-scaling factor of  $3\times$  was adopted and the parameters used in [1,22] are untouched. For TIIP algorithm, the irregular pyramid was constructed until the surviving vertices were approximately 33% of the total number of pixels in the original image. In this paper we have chosen minimal contrast as the criteria for selection of the contraction kernels. The chosen reduction function preserves the contrast between the endpoints of the surviving edges. This choice preserves the critical vertices and the topology of the image. The image was reconstructed by implementing the EXPAND operation on the reduced graph and the pixel intensity (RGB) information of the surviving vertices only.

The algorithms were tested on all the 100 images of BSD100 - Berkeley segmentation data set [19] which is widely used publicly available data set for various image processing and computer vision tasks. For quality assessments, we used Structural Similarity Index Measure (SSIM) [24] and Feature Similarity Index Measure (FSIM) [27] which are perceptual metric based on the visible structures in the image. In contrast to these assessment, to measure the global degradation of image, we calculated Peak Signal to Noise Ratio (PSNR). Table 1 is showed to communicate the advantages of preserving the structural properties through irregular pyramids over regular pyramids<sup>4</sup>.

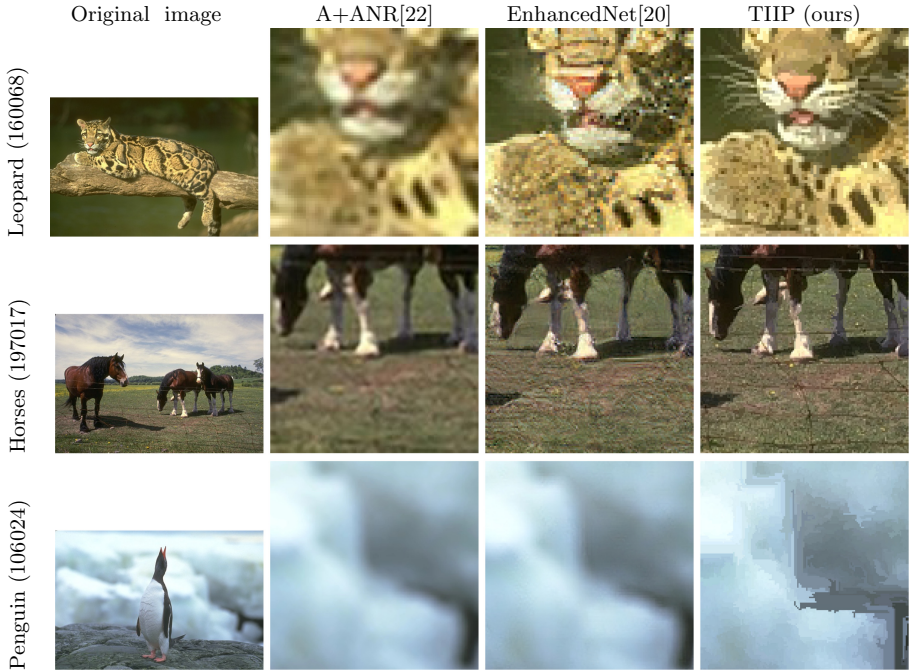
**Table 1.** Quality assessment of reconstructed images on BSD100 ( $3\times$ ).

Quality measure \ method	A+ANR [22]	EnhanceNet [20]	RCAN [29]	DRLN+ [1]	TIIP
SSIM	0.75	0.73	0.811	0.812	0.92
FSIM	0.86	0.84	–	–	0.93
PSNR	26.64	27.50	29.3	29.4	36.26

The example *Leopard* and *Horses* in Fig. 4 clearly shows that TIIP successfully preserves the high frequency texture information, such as the thin lines

<sup>4</sup> Quality measures of RCAN and DRLN+ algorithms are not given because the code did not compile. The FSIM measures are not mentioned in [1,29].





**Fig. 4.** For each image, original (from BSD 100) and reconstructed images from A+ANR [22], EnhanceNet [20] and TIIP algorithms with their zoom visuals.

due to hairs and the fencing. While, in the example of *Penguin* there appears pastelized especially in the smooth areas where the focus is lower. The results are also reflected in the Table 1 which shows that TIIP algorithm successfully preserves the structure and the features of the image.

*Observations from the Experiments:* 1. Irregular pyramids are more suitable to preserve and analyse the structural information of the image compared to regular pyramids. In contrary, low resolution images cannot be displayed properly since a non-grid graph is used to represent the image. 2. The TIIP algorithm performs better on images (or patch of image) with high texture, because they contain a higher number of critical points which are preserved by TIIP. In contrary, there is a considerable amount of perceptual inconsistencies in smooth regions of images which are out of focus. The inconsistent regions are usually comprised of the background and homogeneous regions of an image.

## 4 Conclusions and Future Work

The most important steps, controlling the hierarchy in the proposed algorithm, are the selection of the contraction kernels and the choice of the reduction function. In this paper, we have chosen minimal contrast for selecting the contraction kernels. The chosen reduction function preserves the contrast between the

endpoints of the surviving edges. This choice allows us to preserve the critical vertices and the topology of the image. As a result, the algorithm generates superior results by reconstructing the fine details and the high frequency texture information of the image, which is typically lost after smoothing. Thus an image is equivalent to the combination of its structure and few colors. The hierarchy provides a (multi-resolution) structural overview of the image.

Future work includes speeding up the algorithm by implementing it on a GPU. We also plan to derive application-specific constraints and combine them with the TIIP algorithm to achieve better results for image segmentation, connected component labelling, etc. By merging machine learning with the TIIP algorithm, we anticipate learning the edge weights corresponding to the edges and derive a promising contraction and removal kernel, thereby achieving even better application-specific results.

## References

1. Anwar, S., Barnes, N.: Densely residual laplacian super-resolution. *IEEE Trans. Pattern Anal. Mach. Intell.* (2020)
2. Batavia, D., Hladůvka, J., Kropatsch, W.G.: Partitioning 2D images into prototypes of slope region. In: Vento, M., Percannella, G. (eds.) *CAIP 2019*. LNCS, vol. 11678, pp. 363–374. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-29888-3\\_29](https://doi.org/10.1007/978-3-030-29888-3_29)
3. Batavia, D., Kropatsch, W.G., Casablanca, R.M., Gonzalez-Diaz, R.: Congratulations! Dual graphs are now orientated!. In: Conte, D., Ramel, J.-Y., Foggia, P. (eds.) *GbrPR 2019*. LNCS, vol. 11510, pp. 131–140. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-20081-7\\_13](https://doi.org/10.1007/978-3-030-20081-7_13)
4. Bremer, P.T., Hamann, B., Edelsbrunner, H., Pascucci, V.: A topological hierarchy for functions on triangulated surfaces. *IEEE TVCG* **10**(4), 385–396 (2004)
5. Brun, L., Kropatsch, W.: Irregular pyramids with combinatorial maps. In: Ferri, F.J., Iñesta, J.M., Amin, A., Pudil, P. (eds.) *SSPR /SPR 2000*. LNCS, vol. 1876, pp. 256–265. Springer, Heidelberg (2000). [https://doi.org/10.1007/3-540-44522-6\\_27](https://doi.org/10.1007/3-540-44522-6_27)
6. Burt, P.J., Adelson, E.H.: The Laplacian pyramid as a compact image code. *IEEE TC* **31**(4), 532–540 (1983)
7. Cerman, M., Janusch, I., Gonzalez-Diaz, R., Kropatsch, W.G.: Topology-based image segmentation using LBP pyramids. *MVA* **27**(8), 1161–1174 (2016)
8. Dai, Q., Chopp, H., Pouyet, E., Cossairt, O., et al.: Adaptive image sampling using deep learning and its application on x-ray fluorescence image reconstruction. *IEEE TM* **22**(10), 2564–2578 (2019)
9. Edelsbrunner, H., Harer, J., Zomorodian, A.: Hierarchical morse-smale complexes for piecewise linear 2-manifolds. *DCG* **30**(1), 87–107 (2003)
10. Edelsbrunner, H., Letscher, D., Zomorodian, A.: Topological persistence and simplification. In: *Proceeding of International Conference on IEEE FOCS*, pp. 454–463 (2000)
11. Helman, J.L., Hesselink, L.: Visualizing vector field topology in fluid flows. *IEEE CGA* **11**(3), 36–46 (1991)
12. Hoppe, H.: Progressive meshes. In: *Proceeding of CGIT*, pp. 99–108. ACM (1996)

13. Iliadis, M., Spinoulas, L., Katsaggelos, A.K.: Deep fully-connected networks for video compressive sensing. *Digital Signal Process.* **72**, 9–18 (2018)
14. Kropatsch, W.G.: From equivalent weighting functions to equivalent contraction kernels. In: *International Conference on DIP*, vol. 3346, pp. 310–320 (1998)
15. Kropatsch, W.G., Casablanca, R.M., Batavia, D., Gonzalez-Diaz, R.: On the space between critical points. In: *Couprrie, M., Cousty, J., Kenmochi, Y., Mustafa, N. (eds.) DGCI 2019. LNCS*, vol. 11414, pp. 115–126. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-14085-4\\_10](https://doi.org/10.1007/978-3-030-14085-4_10)
16. Latecki, L.J.: 3D well-composed pictures. *CVGIP* **59**(3), 164–172 (1997)
17. Lee, R.N.: Two-dimensional critical point configuration graphs. *IEEE TPAMI* **4**, 442–450 (1984)
18. Maia, D.S., Cousty, J., Najman, L., Perret, B.: Recognizing hierarchical watersheds. In: *Couprrie, M., Cousty, J., Kenmochi, Y., Mustafa, N. (eds.) DGCI 2019. LNCS*, vol. 11414, pp. 300–313. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-14085-4\\_24](https://doi.org/10.1007/978-3-030-14085-4_24)
19. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proceeding of ICCV*, vol. 2, pp. 416–423. IEEE (2001)
20. Sajjadi, M.S., Scholkopf, B., Hirsch, M.: EnhanceNet: single image super-resolution through automated texture synthesis. In: *Proceeding of IEEE ICCV*, pp. 4491–4500 (2017)
21. Shinagawa, Y., Kunii, T.L., Kergosien, Y.L.: Surface coding based on morse theory. *IEEE Comput. Graph. Appl.* **11**(5), 66–78 (1991)
22. Timofte, R., De Smet, V., Van Gool, L.: A+: adjusted anchored neighborhood regression for fast super-resolution. In: *Cremers, D., Reid, I., Saito, H., Yang, M.-H. (eds.) ACCV 2014. LNCS*, vol. 9006, pp. 111–126. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-16817-3\\_8](https://doi.org/10.1007/978-3-319-16817-3_8)
23. Torres, F., Kropatsch, W.G.: Canonical encoding of the combinatorial pyramid. In: *Proceeding of CVWW*, pp. 118–125 (2014). ISBN: 978-80-260-5641-6
24. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE TIP* **13**(4), 600–612 (2004)
25. Yang, C.Y., Yang, M.H.: Fast direct super-resolution by simple functions. In: *Proceeding of IEEE ICCV*, pp. 561–568 (2013)
26. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: *Boissonnat, J.D., et al. (eds.) Curves and Surfaces 2010. LNCS*, vol. 6920, pp. 711–730. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-27413-8\\_47](https://doi.org/10.1007/978-3-642-27413-8_47)
27. Zhang, L., Zhang, L., Mou, X., Zhang, D.: FSIM: a feature similarity index for image quality assessment. *IEEE TIP* **20**(8), 2378–2386 (2011)
28. Zhang, Y., Zhao, D., Zhang, J., Xiong, R., Gao, W.: Interpolation-dependent image downsampling. *IEEE TIP* **20**(11), 3291–3296 (2011)
29. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: *Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS*, vol. 11211, pp. 294–310. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-01234-2\\_18](https://doi.org/10.1007/978-3-030-01234-2_18)