Pattern Recognition Letters 190 (2025) 8–14

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Adapting a total vertex order to the geometry of a connected component

Majid Banaeyan *, Walter G. Kropatsch

Virtual and Augmented Reality, Institute of Visual Computing and Human-Centered Technology, Favoritenstraße 9/E193-03, 1040, Vienna, Austria

ARTICLE INFO

Editor: Maria De Marsico

MSC:
41A05
41A10
65D05
65D17

Keywords:
Irregular graph pyramid
Image pyramid
Selecting the contraction kernels
Pattern recognition
Spanning tree
Total order
Eccentricity transform
Distance transform

ABSTRACT

Irregular graph pyramids are frequently used as powerful tools in pattern recognition and image processing. They are built by merging specific vertices and edges, known as contraction kernels, at each level. Traditional methods often randomly select these kernels, leading to an unpredictable vertex at the top of the pyramid. This paper presents innovative methods to control the selection of contraction kernels, enabling the intentional preservation of a vertex with desired properties at the pyramid's top. Specifically, we focus on maintaining the center of a connected component (CC) at the pyramid's apex. For calculating the center of a region, we utilize the eccentricity transform, which is robust against noise. Our approach begins by establishing a total vertex order and then devises solutions for continuous spaces in both 1D and 2D. Subsequently, we adapt these continuous space solutions to discrete spaces, again in both 1D and 2D dimensions. The experimental results demonstrate the efficacy and validity of our proposed methods.

1. Introduction

Irregular image pyramids are crucial in pattern recognition, allowing for complexity reduction in image processing tasks like connected component labeling and distance transform [1–5]. These pyramids, formed by layered reduced images or graphs, select contraction kernels at each level to merge vertices and edges.

Historically, contraction kernels were chosen randomly, leading to unpredictability in the pyramid's top vertex [6–8]. Recent studies, however, adopt a predefined vertex order, allowing for more controlled vertex selection, though often fixing it at a grid's bottom-right [2,9].

Contemporary literature [2,9] introduces an approach that adopts a predefined total order of vertices, allowing enhanced discretion in choosing the surviving vertex. Though effective, these techniques often place the surviving vertex at a fixed position, typically the bottom-right of a connected component in a grid structure. This placement suggests that the vertex selection does not inherently align with the characteristics of the connected component.

This paper endeavors to refine the process of selecting contraction kernels according to new input data characteristics. By exerting *control* over kernel selection, a specific segment of the receptive field or connected component can be preserved at the top level of the pyramid. Given the frequent applications in image segmentation and object detection for centering a connected component at the top level, this work offers methodologies that retain the center of a connected component at the apex. In some scenarios, this center may consist of multiple pixels or vertices and needs to be approximated.

The subsequent sections provide a thorough exploration of the topic: Section 2 introduces foundational definitions and background information; Section 3 delves into defining a valid total vertex order; Section 4 proposes methodologies for adapting the valid total vertex order in continuous spaces, encompassing both 1D and 2D scenarios; Section 5 bridges these concepts from continuous to discrete spaces; Section 6 brings the experimental results; Section 7 discusses an interesting open problem and Section 8 concludes the paper.

2. Background and definitions

Irregular pyramids are hierarchical structures composed of progressively smaller graphs, each graph representing a different level of the pyramid. The base graph, corresponding to the lowest level, aligns with the input image. In this graph, each pixel is a vertex, connected to its immediate neighbors to form a 4-connected neighborhood graph, $G(V, E)$. This model avoids intersecting edges and maintains the

* Corresponding author.
E-mail address: majid@prip.tuwien.ac.at (M. Banaeyan).

https://doi.org/10.1016/j.patrec.2025.01.030
Received 12 January 2024; Received in revised form 23 December 2024; Accepted 24 January 2025
Available online 7 February 2025

planarity of the graph, unlike the 8-connected model where diagonal connections can disrupt planarity [10].

The construction of irregular pyramids relies on two fundamental operations: edge contraction and edge removal. Edge contraction merges two connected vertices into one, while edge removal simply eliminates an edge without altering the vertex count or the incidence relationships of the remaining edges. In this process, certain vertices or edges that do not continue to the next level are termed non-surviving, while those that do are known as surviving vertices or edges.

Definition 1 (*Contraction Kernel [11]*)**.** A contraction kernel $K \subset E$ in a graph $G(V, E)$ is a subset of edges forming a spanning forest. Each tree in this forest includes one vertex $s \in V_s \subset V$ surviving to the next higher level of the pyramid. In an extreme case, the tree consists simply of s.

The **Eccentricity Transform (ECC)** is a function crucial in graph-based image analysis [12], particularly for shapes within digital images [13,14]. It calculates the eccentricity for every point in a graph, defined as the longest of the shortest distances from a given vertex to any other vertex [14]. This transform is notable for its invariance to articulated motion, robustness to salt & pepper noise [15], and its utility in boundary determination. The *diameter* of a shape is the maximum value of the eccentricity transform, representing the longest path within the shape. Conversely, the *center* of a graph is determined by the vertices that yield the minimum value in the eccentricity transform, signifying the shortest path to the most eccentric vertices.

The **Reeb graph** is a topological construct used to capture the essence of a shape by analyzing the level sets of a real-valued function defined on a manifold, typically the height function on a surface [16]. It simplifies and abstracts the shape's structure by creating a graph where nodes represent critical points (like peaks, valleys, or saddle points) and edges represent the connectivity or continuity of the surface between these points [17]. This graph effectively reduces complex 3D shapes to more manageable 2D representations, making it valuable in various applications such as shape analysis, computer graphics, and data visualization [18]. The Reeb graph excels in identifying and classifying topological features of shapes, aiding in tasks like feature extraction, morphological analysis, and in understanding the overall geometry and connectivity of the underlying manifold [19].

3. Total order of the vertices

Let $G = (V, E)$ be a connected plane graph with n vertices and the following neighborhood definition.

Definition 2 (*Neighborhood of a Vertex v, $\mathcal{N}(v)$*)**.** In a graph $G = (V, E)$, the neighborhood of a vertex v is the union of vertex v itself and all vertices incident to v:

$$\mathcal{N}(v) = v \cup \{w \in V | (v, w) \in E\} \tag{1}$$

Definition 3 (*Equivalent Contraction Kernel*)**.** An Equivalent Contraction Kernel (ECK) refers to a spanning tree that combines all contraction kernels into a single contraction kernel which generates the same result in one single contraction.

A comprehensive definition of ECK is available in Refs. Kropatsch [20] and Haxhimusa et al. [21].

Definition 4 (*Root of the ECK*)**.** The root of the Equivalent Contraction Kernel of the receptive field is its surviving vertex at a higher level of the pyramid.

Definition 5 (*Parent Link*)**.** A parent link (PL) associates every non-surviving vertex with its respective surviving vertex within an incidence relationship such that combining all the CKs forms the spanning tree of the receptive field.

$$\begin{aligned} &\Pi : V \mapsto V_s \subset V \\ &\forall v \in V \quad \exists v_s \in V_s \ \Pi(v) = v_s \end{aligned} \tag{2}$$

Having the defined the basic structure of the pyramid we now assign the vertices a rank of a strict total order in order to resolve ambiguities in the construction process.

Definition 6 (*Total Vertex Order*)**.** A Total Vertex Order is a **bijective function** that assigns a unique *rank* from 1 to n to each vertex of a connected plane graph composed of n vertices.

$$\begin{aligned} &TO : \{v_1, v_2, v_3, \dots, v_n\} \mapsto \{1, 2, 3, \dots, n\} \\ &TO(v_i) = j \quad i, j \in \{1, 2, 3, \dots, n\} \\ &TO(v_i) = TO(v_k) \iff i = k \end{aligned} \tag{3}$$

Using the principles of a strict total order of vertices as outlined in [22], a parent link (PL) of an equivalent contraction kernel can be derived from the TO of the vertices of the receptive field with the aim of aggregating all contraction kernels in the spanning tree of the receptive field.

Following definition of a *max-link* generates for all non-maximal vertices v of the TO a spanning forest of G:

Definition 7 (*Max-Link*)**.** The *max-link* Up(v) links every vertex v to the **maximum** TO rank of its neighbors:

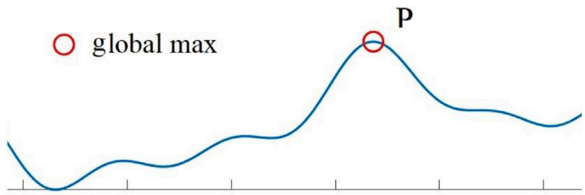
$$\forall v \in V \setminus V_s \quad \text{Up}(v) = \text{argmax}\{TO(w) | w \in \mathcal{N}(v)\} \tag{4}$$

The resulting spanning forest $G(V, \{(v, \text{Up}(v))\}$ with $(v, \text{Up}(v)) \subset E)$ contains for every local maximum of the TO one tree. If the global maximum is the only local maximum of TO, $G(V, \{(v, \text{Up}(v)) | v \in V\})$ is a tree spanning $G(V, E)$. Hence, the max-link $Up(v)$ might be aptly suited as parent-link for an ECK. Since all vertices can locate the maximal TO-neighbor in parallel the parallel complexity depends only on the maximal degrees of the involved vertices, not on their number.

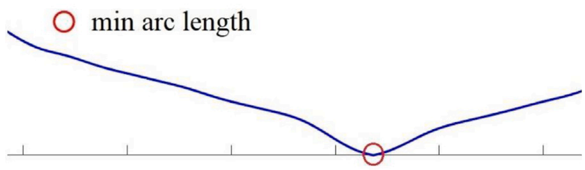
Proposition 1. *If the max-link is considered for generating a potential PL from a total order, the total order is valid if it has a single global maximum, if it lacks any other local maxima, and if it allocates the highest rank, n, to the root.*

Proof. Let us take a connected plane graph, $G = (V, E)$, comprising $|V| = n$ vertices. In accordance with the properties of total order, every vertex receives a distinct number. This setup ensures each vertex links to a sole neighboring vertex with a superior rank. Given only a single global maximum and no other local maxima, all vertices — bar the root — can identify one neighbor vertex distinct from themselves using max-link (=PL). Since the graph is connected every vertex except the global maximum have selected one neighbor by PL, no vertex remains isolated. The uniqueness of TO ranks ensures that each vertex selects only one neighbor, thus preventing the formation of loops in the resulting connected graph. As a consequence, this graph embodies a spanning tree of the initial graph, validating the designated total order. □

Proposition 1 ensures that if there is only one global maximum and no other local maxima, then the total order created by the max-link is a valid total order. The primary question that arises is: How can a valid total order be established? To address this, we first tackle the problem in a continuous space and subsequently adapt the solution to a discrete space. Furthermore, we begin by solving the problem in 1D and then extend our approach to 2D space.



(a) The 1D curve showing multiple local extrema and the global maximum.



(b) The arc length measurement to the global maximum.

Fig. 1. Analysis of a 1D curve with extrema, including local and global features.

4. Solving the problem in continuous space

4.1. Total order in a 1D curve

Let S be a $1D$ curve with one global maximum at point p. The curve may have many local extrema. In order to overcome the problem of having the local extrema, for each point of the curve its geodesic distance to the global maximum is calculated. Therefore the resulting curve has no local extrema.

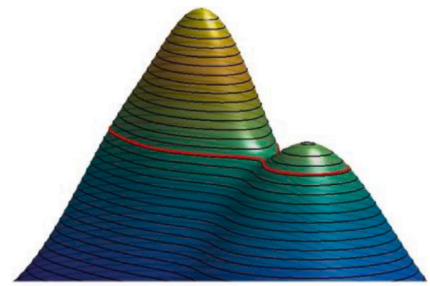
Proposition 2. *Given a point P on a 1D continuous, open curve S, the computation of the arc length from each point on S to P yields a function with a single global minimum, devoid of any other local extrema.*

Proof. Consider a point P on the curve. This point divides the curve into two segments: those points to the left of P and those to the right. For any point situated on the left, the corresponding arc length increases as one moves further away from P. Hence, there exists a monotonically descending path from each point on the left to the point P. Similarly, this reasoning can be applied to points on the right side of P. □

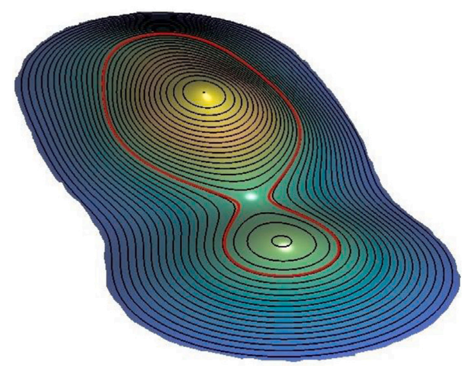
It is important to notice that the curve, whether smooth or non-smooth, should not have any discontinuities. Fig. 1-a shows a curve with point P highlighting the global maximum by a red circle. Its associated arc length (or geodesic distance) relative to point P is illustrated in Fig. 1-b, revealing a curve characterized by a single global minimum at P and devoid of any local extrema.

4.2. Total order on a 2D surface

Let S be a continuous open surface, where P represents its unique global maximum. While the surface may contain multiple local maxima, the objective is to create a monotonic path from every point on the surface to the global maximum by eliminating these local maxima. In essence, the intent is to transform the local maxima into plateau regions. This transformation is achieved by utilizing the surface's level curves. Notably, the level curves surrounding both local and global extrema, whether minima or maxima, are closed curves that intersect the boundary. According to Kropatsch and Banaeyan [11], saddle points on the surface are unique in that their level curves can intersect. Given knowledge of the global maximum, saddle points can be sorted based on height to determine the level curves related to a local maximum. The height of these enclosed level curves around a local maximum gradually decreases until they align with the level curve of the adjacent lower saddle point. To flatten a local maximum, all inner level curves within its saddle point's curves are adjusted to match the value of that saddle point's level curve. This adjustment effectively eliminates the local maximum. By applying this approach to all extrema except the global maximum, the original surface is transformed into one devoid of all other local maxima. Fig. 2 depicts a surface with a single global maximum and an additional local maximum. The level curves, represented by closed curves, intersect the surface's boundary. The level curve corresponding to the saddle point is accentuated in red.



(a) A surface with one global and one local maximum



(b) Top view of the surface

Fig. 2. A surface with its level curves.

5. Total order in discrete space

In discrete spaces, addressing the issue of eliminating local maxima presents a more formidable challenge. This complexity arises from the discretization of continuous spaces, where the sampling resolution might not be sufficiently fine to capture all critical points of a continuous surface within the sampling grid structure. Nonetheless, in subsequent sections, novel methods are introduced to mitigate this challenge and adapt the concepts from continuous spaces for use in discrete domains.

5.1. 1D total order

Let $G = (V, E)$ be a *string* composed of n vertices, defined as:

Definition 8. A string is a graph with $n \geq 2$ vertices where only two vertices are leaves with degree 1. If $n > 2$ all the other $n - 2$ vertices possess a degree of 2.

Using the diameter of this string, the vertex (or vertices) exhibiting the minimum eccentricity transform (ECC) is determined as per [13].

Proposition 3. *A string has a single vertex with the minimum eccentricity transform value if and only if it comprises an odd number of vertices.*

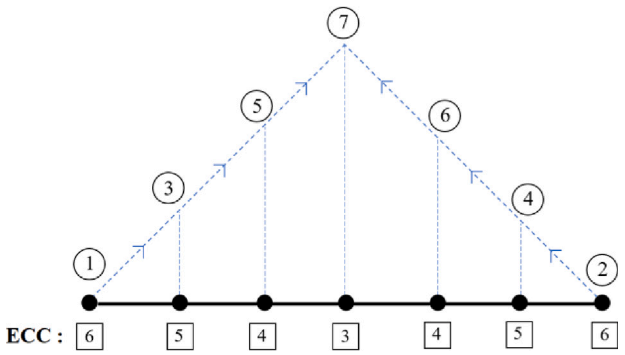


Fig. 3. Depiction of a TO for a linear tree. The numbers within the circles represent the TO.

Proof. Given a string with an odd number of vertices, the ECC is computed beginning from its two endpoints. These endpoints are the eccentric points with identical maximum ECC values. Moving inwards, vertex by vertex from both sides, the ECC value decreases at each step. Due to the string having an odd number of vertices, the central vertex ultimately obtains the minimum ECC value. □

Proposition 4. *A string has two vertices with the minimum eccentricity transform value if and only if it consists of an even number of vertices.*

Proof. Utilizing the approach from the previous proof, the ECC is computed for a string with an even number of vertices. At the conclusion of this process, two central vertices emerge with identical minimum ECC values. □

To establish a valid total order for a string, begin with the case where the number of vertices in the linear tree is odd, represented as $n = 2k + 1$, $k \in \mathbb{N}$. In this configuration, the maximum number, n, is allocated to the vertex with the minimum ECC value. For the remaining vertices, two distinct paths are formulated, both originating from the global maximum and concluding at the two leaves. One path encompasses all even numbers less than n, that is, $\{n-1, n-3, \dots, 2\}$, and it decreases monotonically until it culminates at the leaf assigned the number 2. Conversely, the alternate path incorporates odd numbers less than n, specified as $\{n-2, n-4, \dots, 1\}$, decreasing monotonically until it terminates at the leaf designated with the number 1.

For the scenario where two vertices exhibit the minimum ECC, one vertex is assigned the value n while the other is allocated $n-1$. Beginning from the vertex marked n and terminating at a leaf, the path is assigned a series of even numbers that decrease monotonically, represented by $\{n-2, n-4, \dots, 2\}$. Conversely, the path originating from the vertex designated $n-1$ and culminating at another leaf is allotted the set of odd numbers, specifically $\{n-3, n-5, \dots, 1\}$.

Fig. 3 displays a string with an odd number of vertices, specifically $|v| = 7$. The diagram reveals that the two endpoints of the string possess the highest ECC value of 6, while the center vertex attains the lowest ECC value, 3. This center vertex is assigned a value of $n = 7$. The two monotonically ascending paths from the string's endpoints to its center are depicted by dotted lines.

5.2. 2D total order

Consider $G = (V, E)$, a connected plane graph comprising n vertices. When a total order is assigned to the graph's vertices, one global maximum emerges, and potentially several local extrema may arise. The objective is to eliminate all local maxima, ensuring a monotonic path exists from every vertex in the graph to the designated global maximum, which is treated as the graph's *root*. Given that a plane graph uniquely partitions the 2D continuous space, it can be superimposed onto a 2D continuous surface. Consequently, the derived graph becomes a *geometric graph* [23,24], where each vertex possesses a height value corresponding to its point on the 2D surface.

To establish a valid total order in the geometric graph, we introduce two distinct methodologies. The first approach leverages the Reeb graph and incorporates the total order derived from previous processes. In contrast, the second approach bypasses the previously established total order and operates directly on the data by utilizing the eccentricity transform. Detailed explanations of both methodologies are provided in the subsequent sub-sections.

5.2.1. Employing the Reeb graph

For a given geometric graph $G = (V, E)$, the associated Reeb graph, denoted $G_R = (V_R, E_R)$, is constructed. Notably, this Reeb graph comprises significantly fewer vertices and edges than its original counterpart, as it retains only the critical points of the surface and their adjacency essential to the topology of the surface. Within the Reeb graph, individual vertices correspond to connected components (CCs), and the interrelations among these CCs are encapsulated by the graph's edges. The initial goal is to assign a valid total order to the vertices of the Reeb graph, subsequently extending this order to the primary geometric graph.

Assuming that the Reeb graph of the original geometric graph contains k vertices, each leaf represents a local maximum, while branching points signify saddle points, as detailed in [11]. The vertex with the highest height is designated as the Reeb graph's root, obtaining the numeric value k. The challenge then pivots to allocating the remaining numbers, ranging from 1 to $k-1$, to the other vertices. This is achieved by sorting all branching points based on their heights, assigning numbers from $k-1$ in a descending sequence. Subsequently, the remaining vertices, being the tree's leaves, are also ordered in a descending fashion based on their heights. This ordering ensures that the vertex with the lowest elevation is attributed the number 1. This method ensures a unique monotonic path from each vertex to the Reeb graph's root, culminating in a valid total order across the Reeb graph's vertices.

Each vertex in the Reeb graph, representing a connected component (CC) of the original geometric graph, can encompass numerous other vertices or pixels. During the Reeb graph's construction, elements within a CC are retained. These preserved elements might reflect the CC's size or a preceding total order used in its formation, and are referred to as an increment within the Reeb graph.

To transfer the Reeb graph's total order to the original geometric graph, intervals corresponding to the sizes of the CCs are considered. The objective is to allocate a unique number to every pixel within each CC, ensuring consistency with the order established in the Reeb graph. To distribute these numbers, we utilize the order previously established for a CC in earlier processes, aligning it with intervals derived from the Reeb graph. This approach is grounded in the principle that every subset of a total order inherently retains its total order characteristics.

Fig. 4 depicts a cross-section of a surface with one global maximum, two local maxima, and two saddle points. The corresponding Reeb graph overlays the surface profile. The total vertex order of the Reeb graph is represented by the numbers, while the directed edges indicate the monotonic paths.

Note that to align the total order with the geometry of a CC, the maximum value k is assigned to the vertex of the CC that exhibits the minimum eccentricity transform value.

5.2.2. Employing the eccentricity and distance transforms

Let $G = (V, E)$ be a connected plane graph containing n vertices. Two distinct approaches are introduced to produce the spanning tree of the graph G. The first approach employs an irregular graph pyramid as a hierarchical structure to derive the spanning tree. The second approach uses the distance transform metric to generate the spanning tree.

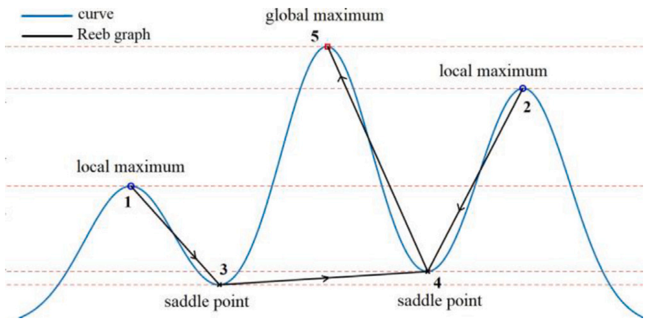


Fig. 4. A vertical cross-section of a continuous surface and the associated Reeb graph.

Computing the Spanning Tree Using a Hierarchical Structure: The graph G is considered the base level of the pyramid. An arbitrary strict total order (TO) assigns a unique number from 1 to n to the vertices. To ensure that the root of the resulting spanning tree is centered, the vertex with the minimum eccentricity (ECC) is assigned the highest rank, n. Applying the max-link generates local maxima. Note that if only one local maximum is produced, the assigned random ranks become the TO.

The contraction kernels of the spanning forest, derived from the max-link, are then contracted to produce the next level of the irregular pyramid. This higher level is a smaller graph containing only the surviving vertices, which are the local maxima from the previous level. A simplification process [25] is applied to remove empty self-loops and parallel edges that do not carry topological information. By continuing the construction of the pyramid in the same manner, the process eventually reaches the top of the pyramid, which contains only one vertex: the global maximum.

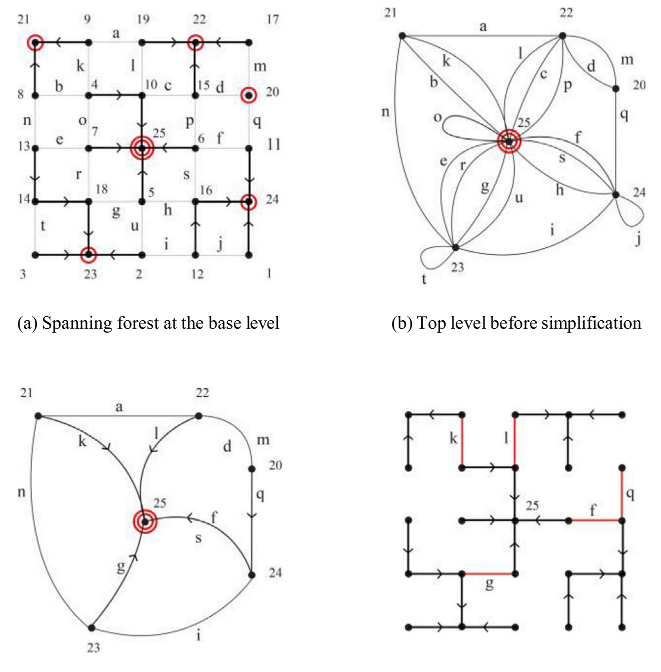
By traversing from the top to the bottom of the pyramid and combining all the selected contraction kernels through the hierarchy, the spanning tree of the graph G at the base level is obtained.

Fig. 5 illustrates an example of constructing the spanning tree using a hierarchical structure with two levels. Vertices identified as local maxima are highlighted with a red circle, and the global maximum is marked with a double red circle. Note that the red edges in Fig. 5-d represent the contraction kernel at the top of the pyramid.

Total Order Using Distance Transform (DT): Consider $G = (V, E)$ as a connected plane graph, and let v_r be the root of the spanning tree to be constructed. This root can either be chosen as the vertex with the minimum ECC of the graph or derived from any desired property based on prior processes. Let v_r serve as the seed vertex, and assume a distance metric, such as the norm-1 (city-block) distance, is used. The distances of all vertices $\forall v \in V \setminus V_r$ from the seed vertex v_r are computed.

The ranks of the strict TO are then assigned based on the distances to the root, following the histogram of these distances. Specifically, the root is assigned the highest rank, n, and vertices at distance 1 are assigned the next highest ranks. For example, if the root has degree m, the vertices at distance 1 receive ranks ranging from $n - m$ to $n - 1$. Similarly, k vertices at distance 2 are assigned ranks ranging from $n - m - k$ to $n - m - 1$. This process continues until all vertices are assigned distinct ranks, thereby producing the strict TO. Once the TO is established, applying the max-link creates the spanning tree of G.

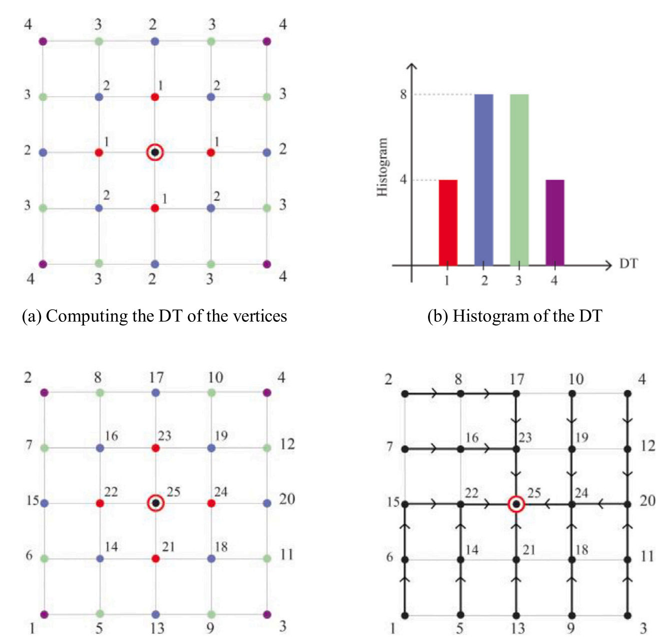
Fig. 6 provides an example of constructing the spanning tree using the city-block distance metric. The distances of the vertices to the root are computed as shown in Fig. 6-a. In this example, the root is the vertex with the minimum ECC, highlighted with a red circle. The histogram of distances, containing four distinct distance values ranging from 1 to 4, is illustrated in Fig. 6-b. Based on this histogram, ranks from 24 to 1 are assigned to the vertices of G as shown in Fig. 6-c. Finally, by applying the max-link, the unique spanning tree corresponding to this TO is generated, as illustrated in Fig. 6-d.



(a) Spanning forest at the base level (b) Top level before simplification

(c) Simplified top of the pyramid (d) Resulting spanning tree at the base

Fig. 5. Hierarchical structure (pyramid) for constructing the spanning tree from a total order.



(a) Computing the DT of the vertices (b) Histogram of the DT

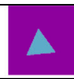
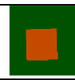
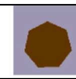
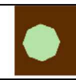
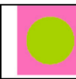
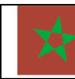
(c) Assigning the TO based on the DT (d) The resulting spanning tree

Fig. 6. Constructing the spanning tree by using the DT.

6. Experiments

To demonstrate the advantages of the proposed methods, we simulated the creation of spanning trees using TO in the 2D digital space,

Table 1
Diameter and root distance ratios of proposed method vs. random spanning trees.

Shapes in the Dataset									
Diameter Ratio	0.66	0.65	0.60	0.58	0.62	0.61	0.60	0.59	0.76
Root Distance Ratio	0.35	0.37	0.33	0.34	0.36	0.38	0.32	0.35	0.50

as explained in Section 5.2.2. The tested dataset[1] comprises nine geometric 2D shapes. Each shape is randomly drawn on a 200 × 200 RGB image. The published dataset consists of nine data classes, with each class representing a geometric shape type (Triangle, Square, Pentagon, Hexagon, Heptagon, Octagon, Nonagon, Circle, and Star). Each class contains 10,000 generated images.

Two metrics were used to compare the results of the spanning trees generated by the proposed method with those obtained using the minimum spanning tree (MST) derived from Prim's algorithm [26]:

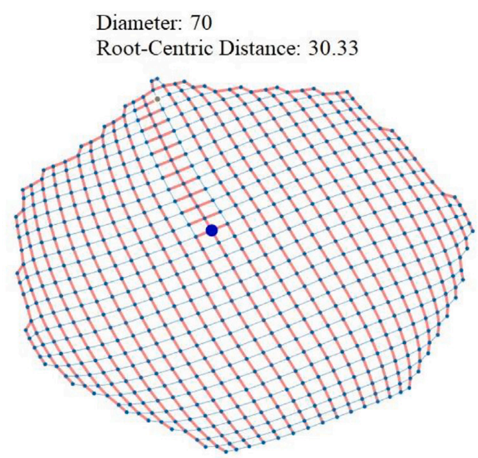
1. Maximum Path Length (Diameter): This metric measures the longest path in the spanning tree from the root to any node. A smaller diameter indicates a more centralized tree structure.

2. Root-Centric Distance Metric: This computes either the average or maximum distance from the root to all nodes. Smaller distances indicate better proximity and compactness.

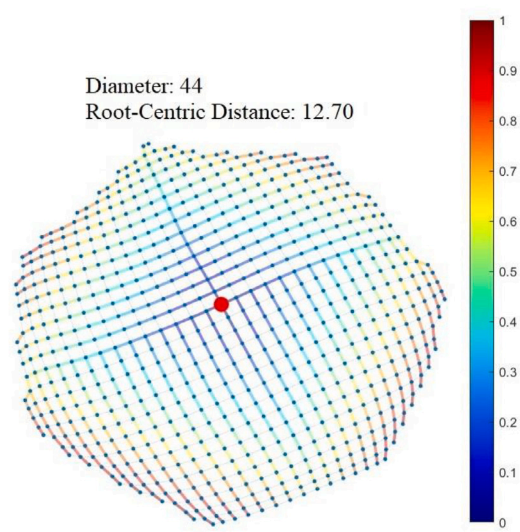
Both metrics are critical in assessing the complexity of the proposed methods. Using the method in [3], the parallel complexity of propagating the distance transform (DT) is $\mathcal{O}(\delta(T))$, where $\delta(T)$ is the diameter of the spanning tree of the foreground connected plane graph. Similarly, the parallel complexity of computing the ECC using the method in [13] is also $\mathcal{O}(\delta(T))$.

Fig. 7a illustrates an MST computed using Prim's algorithm, where the diameter is 70.80, and the root-centric distance is 34.02. In contrast, the proposed method, which selects the vertex with minimum ECC as the root and constructs the spanning tree based on TO, yields a diameter of 44 and a root-centric distance of 12.70. We computed spanning trees over 900 images, with 100 images for each of the nine shape categories. For each shape, 100 spanning trees were generated using Prim's algorithm. The minimum metrics among these 100 spanning trees were averaged to represent the MST metrics for that shape. We then applied the proposed method and computed the corresponding metrics. The results are shown in Table 1. In all shapes except the star, the diameter of the spanning tree produced by the proposed method is approximately 0.6 times the average diameter of the MST computed by Prim's algorithm. Additionally, for all shapes except the star, the root-centric distance is about one-third of that in the MST. This demonstrates that the proposed method produces highly compact spanning trees.

For the star shape, the ECC does not align with the center of the shape due to the hole in its middle, causing it to be located in the peripheral part of the image. Consequently, the ratios of the metrics for the star shape are slightly higher compared to the other shapes.



(a) A random spanning tree



(b) Spanning tree from the TO

Fig. 7. Spanning tree computation for a heptagon shape.

7. Discussion

This study proposed two methods for generating the spanning tree of a given connected plane graph. The first method uses a hierarchical structure to derive the spanning tree. However, it is not always possible to determine a total order (TO) corresponding to the resulting spanning tree. In other words, when considering the max-link as a Parent-link, there exist spanning trees for which generating a TO through the max-link is impossible.

An interesting example is a spiral spanning tree in a compact shape, starting at the boundary and spiraling inward to the center of the shape, where it ends at the root. Fig. 8 illustrates such a spiral with the root R at the center. It is evident that the edge $e = (A, B)$ cannot be produced by max-link because both R and B are neighbors of vertex A, and R has the highest rank. Consequently, applying max-link would force vertex A to select the root R as its parent, making it impossible to produce the edge e. Therefore, finding a TO that generates a given arbitrary spanning tree through max-link remains an **open problem**. Future work may explore alternative functions or methods to derive a TO from any arbitrary spanning tree. The second method derives the TO based on the distance of vertices to the root. When m vertices have the same distance to the root, there are $m!$ possible permutations of their ranks. Our method selects one of these permutations. For example, in Fig. 6, there are 9.364×10^{11} distinct spanning trees with the same

[1] Anas, EL KORCHI (2020), "2D geometric shapes dataset", Mendeley Data, V1, doi: 10.17632/wzr2yv7r53.1.

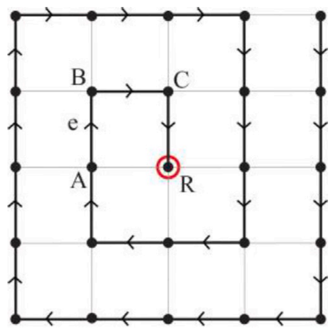


Fig. 8. A spiral spanning tree with R as its root.

diameter, calculated as:

Total Spanning Trees $= 4! \times 8! \times 8! \times 4!$.

This highlights the vast number of spanning trees that can be generated even for relatively small graphs, emphasizing the flexibility and potential of the proposed method.

8. Conclusion

This study introduces a foundational theory for adapting a total vertex order, anchored in the geometrical attributes of a connected component. Moving away from traditional methods of constructing irregular pyramids, where contraction kernels are selected randomly, this research presents methods that enable controlled selection. This control ensures the persistence of a specified vertex from a connected component at the pyramid's apex. Solutions are proposed for both continuous and discrete spaces, offering a structured approach to 1D and 2D scenarios. Moreover, the use of the Reeb graph in 2D discrete cases indicates the method's potential for extension to higher n-dimensional situations.

Having control over the selection of contraction kernels based on desired properties such as color, attention, textural features, etc., opens new applications where these properties can be preserved at the top level. This approach may bring us closer to understanding the human vision system's processing mechanisms.

CRediT authorship contribution statement

Majid Banaeyan: Writing – review & editing, Writing – original draft. **Walter G. Kropatsch:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] M. Banaeyan, D. Batavia, W.G. Kropatsch, Removing redundancies in binary images, in: International Conference on Intelligent Systems and Patterns Recognition (ISPR), Hammamet, Tunisia, March 24-25, 2022, Springer, 2022, pp. 221–233.
[2] M. Banaeyan, W.G. Kropatsch, Parallel $\mathcal{O}(log(n))$ computation of the adjacency of connected components, in: International Conference on Pattern Recognition and Artificial Intelligence (ICPRAI), Paris, France, June 1-3, 2022, Springer, 2022, pp. 102–113.
[3] M. Banaeyan, C. Carratù, W.G. Kropatsch, J. Hladůvka, Fast distance transforms in graphs and in gmaps, in: IAPR Joint International Workshops on Statistical Techniques in Pattern Recognition (SPR 2022) and Structural and Syntactic Pattern Recognition (SSPR 2022), Montreal, Canada, August 26-27, 2022, in: Lecture Notes in Computer Science, Springer, 2022, pp. 193–202.
[4] M. Banaeyan, W.G. Kropatsch, Distance transform in parallel logarithmic complexity, in: Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods - ICPRAM, SciTePress, 2023, pp. 115–123.
[5] M. Banaeyan, W.G. Kropatsch, Distance transform in images and connected plane graphs, in: International Conference on Pattern Recognition Applications and Methods, ICPRAM, Springer, 2024, pp. 19–32.
[6] P. Meer, Stochastic image pyramids, Comput. Vis. Graph. Image Process. Vol. 45 (No. 3) (1989) 269–294.
[7] W.G. Kropatsch, Y. Haxhimusa, Z. Pizlo, G. Langs, Vision pyramids that do not grow too high, Pattern Recognit. Lett. Vol. 26 (3) (2005) 319–337.
[8] Y. Haxhimusa, R. Glantz, W.G. Kropatsch, Constructing stochastic pyramids by MIDES - maximal independent directed edge set, in: E. Hancock, M. Vento (Eds.), 4th IAPR-TC15 Workshop on Graph-Based Representation in Pattern Recognition, vol. 2726, Springer, Berlin Heidelberg, New York, 2003, pp. 24–34.
[9] M. Banaeyan, W.G. Kropatsch, Fast labeled spanning tree in binary irregular graph pyramids, J. Eng. Res. Sci. 1 (10) (2022) 69–78.
[10] R. Klette, Concise Computer Vision, 233, Springer, 2014, pp. 90–94.
[11] W.G. Kropatsch, M. Banaeyan, Emerging topics in Pattern Recognition and Artificial Intelligence, Chapter 2: Controlling Topology Preserving Graph Pyramids, Controlling Topology Preserving Graph Pyramids, in: Pattern Recognition and Intelligent Systems, World Scientific book, 2024, pp. 27–75.
[12] D. Aouada, D.W. Dreisigmeyer, H. Krim, Geometric modeling of rigid and non-rigid 3D shapes using the global geodesic function, in: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008, pp. 1–8.
[13] M. Banaeyan, W.G. Kropatsch, Reducing the computational complexity of the eccentricity transform of a tree, in: 13th IAPR-TC15 International Workshop on Graph-Based Representations in Pattern Recognition (GbRPR), Salerno, Italy, September 6-8, 2023, 2023, pp. 160–171.
[14] W.G. Kropatsch, A. Ion, Y. Haxhimusa, T. Flanitzer, The eccentricity transform (of a digital shape), in: A. Kuba, L. Nyúl, K. Palágyi (Eds.), Discrete Geometry for Computer Imagery, vol. 4245, Springer, Berlin Heidelberg, 2006, pp. 437–448.
[15] A. Ion, S. Peltier, S. Alayranges, W.G. Kropatsch, Eccentricity based topological feature extraction, in: S. Alayranges, G. Damiand, L. Fuchs, P. Lienhardt (Eds.), Workshop Computational Topology in Image Context, CTIC 2008, Universit'e de Poitiers, 2008.
[16] Edelsbrunner, Harer, Zomorodian, Hierarchical morse—smale complexes for piecewise linear 2-manifolds, Discrete Comput. Geom. 30 (2003) 87–107.
[17] Y. Shinagawa, T.L. Kunii, Y.L. Kergosien, Surface coding based on morse theory, IEEE Comput. Graph. Appl. 11 (05) (1991) 66–78.
[18] J. Tierny, J.-P. Vandeborre, M. Daoudi, Partial 3D shape retrieval by reeb pattern unfolding, in: Computer Graphics Forum, vol. 28, Wiley Online Library, 2009, pp. 41–55, 1.
[19] S. Biasotti, B. Falcidieno, M. Spagnuolo, Extended reeb graphs for surface understanding and description, in: Discrete Geometry for Computer Imagery: 9th InternationalConference, DGCI 2000 Uppsala, Sweden, December 13–15, 2000 Proceedings 9, Springer, 2000, pp. 185–197.
[20] W.G. Kropatsch, Equivalent weighting functions to equivalent contraction kernels, in: Sixth International Workshop on Digital Image Processing and Computer Graphics: Applications in Humanities and Natural Sciences, vol. 3346, International Society for Optics and Photonics, 1998, pp. 310–320.
[21] Y. Haxhimusa, M. Saib, R. Glantz, W.G. Kropatsch, Equivalent contraction kernels using dynamic trees, in: B.s. Likar (Ed.), Computer Vision - 6th CVWW, Computer Vision Winter Workshop, Slovenian Pattern Recognition Society, 2001, pp. 267–275.
[22] P.R. Halmos, Naive set theory, van Nostrand, 1960, pp. 225–272.
[23] H. Bunke, K. Shearer, A graph distance metric based on the maximal common subgraph, Pattern Recognit. Lett. 19 (3) (1998) 255–259.
[24] S.P. Dwivedi, R.S. Singh, Error-tolerant geometric graph similarity and matching, Pattern Recognit. Lett. 125 (2019) 625–631.
[25] M. Banaeyan, Redundant Edges and Parallel Algorithms in Binary Irregular Graph Pyramids (Ph.D. thesis), Technische Universität Wien, Vienna, Austria, 2024.
[26] R.C. Prim, Shortest connection networks and some generalizations, Bell Syst. Tech. J. 36 (6) (1957) 1389–1401.