# Skeleton-based decomposition of simple polygons

**Maike Buchin[1], Axel Mosig[2], and Leonie Selbach[2]**

1 **Technical University Dortmund**
`maike.buchin@tu-dortmund.de`

2 **Ruhr University Bochum**
`Axel.Mosig|Leonie.Selbach@rub.de`

## Abstract

In the application of polygon decomposition for the dissection of tissue samples certain constraints on the size and convexity of the subpolygons are given. We present a decomposition method in which different feasibility criteria can be included. Our method is based on a discrete skeleton of the given polygon and can be modified for different optimization problems.

## 1 Introduction

Polygon decomposition is a common method in algorithmic geometry. Depending on the application different constraints for the shape of the subpolygons are used, for example in triangulations or convex decompositions. We present a skeleton-based decomposition method where cuts are restricted by the skeleton points and various constraints for size or shape can be incorporated. Our work is motivated by a problem that arises in histopathology when dissecting disease-specific subregions from tissue samples using the so-called laser capture microdissection (LCM) [3]. The extraction with LCM is not successful unless the regions fulfill certain conditions based on their size and shape. Hence we develop a method to decompose the regions of interest into smaller parts which all satisfy the given constraints.

### 1.1 Problem Statement and Solution

Let $\mathcal{P}$ be a simple polygon without holes. We want to find a feasible decomposition $Z$ of $\mathcal{P}$ given some feasibility criteria, where a decomposition $Z$ is *feasible* if every polygon in $Z$ is feasible. Our method is based on the medial axis or skeleton of $\mathcal{P}$ and allows only specific cuts. Because discrete data in form of digital images is given and a discrete output is expected we use discrete skeletons, that is skeletons consisting of a finite set of points resp. pixels. This leads naturally to a finite number of possible cuts we have to consider. Let $\mathcal{S}$ be the skeleton of $\mathcal{P}$ consisting of $n$ skeleton points. If the degree of the skeleton points does not exceed three, a feasible decomposition of $\mathcal{P}$ based on $\mathcal{S}$ can be computed in time $\mathcal{O}(n^k)$, where $k$ is the number of skeleton points with degree one. This holds also for the minimum number problem and the minimum edge length problem that is minimizing the number of subpolygons in the decomposition or the total length of inserted cuts.

Here we will disregard the actual computation of the feasibility and assume that the feasibility of a subpolygon can be tested efficiently. In our research we consider criteria such as size or (approximate) convexity. In both cases we can compute those values for all adjacent cuts beforehand in time $\mathcal{O}(m)$ for $m$ being the number of boundary points of the input polygon. Given the information for adjacent cuts we can iteratively generate all information needed during the execution of our algorithm in constant time. For other feasibility criteria this may not be the case, in which case this would need to be included in the overall runtime.

35th European Workshop on Computational Geometry, Utrecht, The Netherlands, March 18–20, 2019.
This is an extended abstract of a presentation given at EuroCG'19. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

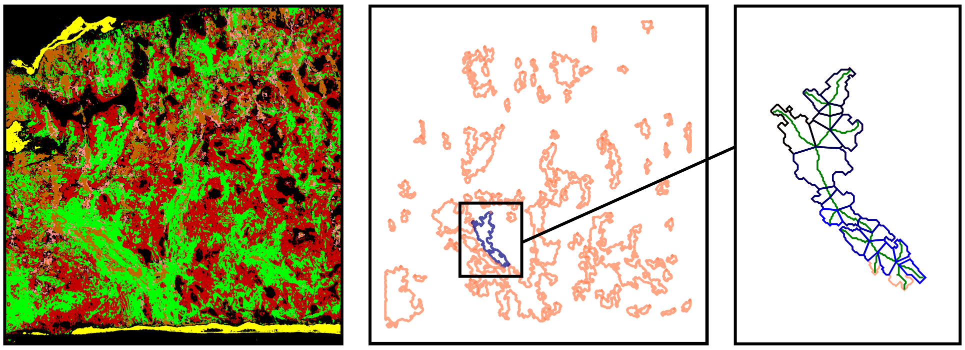


**Figure 1** Decomposition in a histopathological tissue sample given as a classified image (left). This decomposition was generated using the algorithm from [7] using area for feasibility.

## 1.2 Basic Definitions

Let $D \subset \mathbb{R}^2$ be a connected bounded domain. The *medial axis* or *skeleton* $S(D)$ of the set $D$ is the locus of centers of maximal disks in $D$. A *maximal disk* $B$ in $D$ is a closed disk contained in $D$ such that every other disk containing $B$ is not contained in $D$. Let $s$ be the center of a maximal disk $B(s)$, $s$ is called a *skeleton point*. We define the *contact set* of $s$ as $\mathcal{C}(s) = B(s) \cap \partial D$. A connected component of $\mathcal{C}(s)$ is called a *contact component* of $s$ and the elements of $\mathcal{C}(s)$ are called *contact points*. The *degree* of a skeleton point is defined as the number of its contact components. A skeleton $S$ is given as a graph consisting of connected arcs $S_k$ which are called *skeleton branches*. Skeleton branches meet at skeleton points of degree three or higher. We call these points *branching points*.

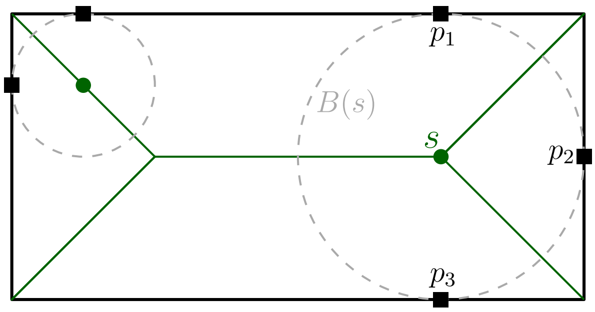



**Figure 2** A skeleton point $s$ with its maximal disk $B(s)$ and contact points $p_1, p_2, p_3 \in \mathcal{C}(s)$.

In our application we consider digital images where we interpret a given object as a polygon by defining each boundary pixel as a corner vertex. We skeletonize the polygon resulting in a simplified discrete skeleton using the method from [2]. The discrete skeleton consists of pixels but fulfills some of the basic properties of the medial axis such that contact components are given. In the skeleton-based decomposition of a polygon $\mathcal{P}$ the cuts are restricted to line segments connecting a skeleton point to a contact point. The cuts induce subpolygons between two or more consecutive skeleton points. $P_k(i, j)$ denotes a polygon generated by two skeleton points $i, j$ on the same skeleton branch $S_k$ as shown in Figure 3.

Since a branching point belongs to more than one branch and has at least three contact points those two points corresponding to the considered branch are chosen, see Figure 4 (a). Notice that a polygon generated by more than two skeleton points can always be represented as a union of subpolygons generated by two skeleton points. See Figure 4 (b) for an example.

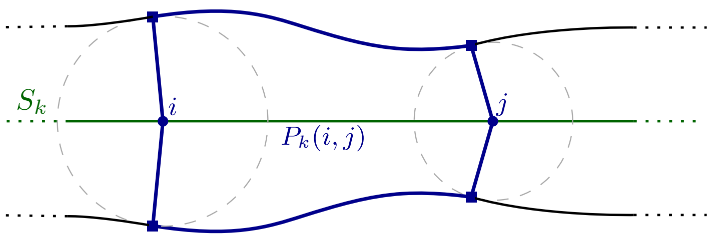
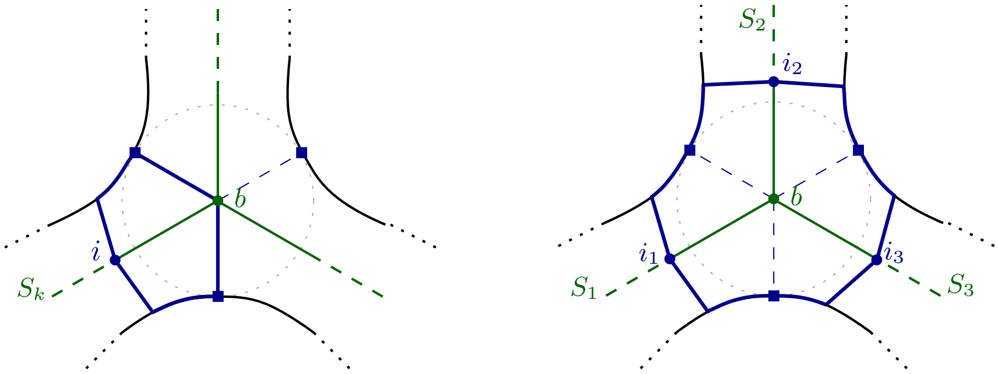



**Figure 3** Polygon $P_k(i,j)$ generated by two skeleton points $i,j$ on the same branch $S_k$.




**(a)** In blue: $P_k(b,i)$.

**(b)** In blue: $P_1(b,i_1) \cup P_2(b,i_2) \cup P_3(b,i_3)$.

**Figure 4** Subpolygons at a branching point (left) and generated by three skeleton points (right).

## 1.3 Related Results from the Literature

Skeletons are used in many applications such as object recognition, medical image analysis and shape decomposition [6]. Leonard et al. [4] use the medial axis for the decomposition of 2D objects to determine a parts hierarchy. Simmons and Séquin [9] compute a hierarchical decomposition of an object using the related axial shape graph. Tănase and Veltkamp [10] use the straight line skeleton to compute decompositions of polygonal shapes into possibly overlapping parts. There are several methods which use a one-dimensional curve skeleton of 3D shapes. Reniers and Telea [5] use the curve skeleton for the segmentation of 3D shapes into meaningful components. Serino et al. [8] propose a method for decomposing a 3D object by using a polygonal approximation of the curve skeleton.

# 2 Decomposition algorithms

For a polygon $\mathcal{P}$ without holes the skeleton $S$ is given as an acyclic graph. We represent $S$ as a tree $T$. For this we pick an arbitrary branching point as root $r$. All other vertices are labeled $v_k$ and correspond to a skeleton branch $S_k$. We define $C(v)$ as the set of children of a vertex $v$. This tree gives us a chronological order of how to work our way through the skeleton. The skeleton points on each branch $S_k$ are labeled from top to bottom – according to the chosen tree representation $T$ – starting with 1 at the top. See Figure 5 for an example.

Before we describe our general decomposition method we consider a special case – which is discussed by Selbach in [7] – where we decompose the polygon by considering each branch of the skeleton on its own. In this case we only have to deal with linear skeletons.

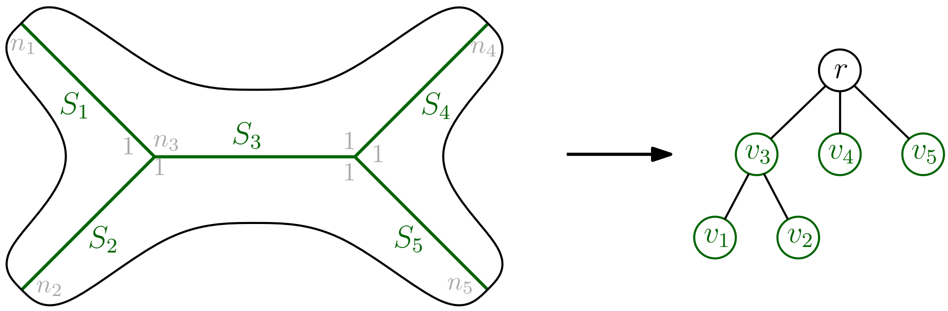



**Figure 5** A representation of a skeleton with two branching points as a tree.

## 2.1 Decomposition based on linear skeletons

Given a polygon $\mathcal{P}_k$ belonging to a skeleton branch $S_k$ with a linear skeleton of size $n_k$, i.e. $\mathcal{P}_k = P_k(1, n_k)$. A feasible decomposition can be found by dynamic programming, using an array $X_k$ such that $X_k(i)$ equals `True` if there exists a feasible decomposition of $P_k(i, n_k)$.

$$X_k(i) = \begin{cases} \texttt{True} & \text{if } \exists j : i < j \leq n_k \text{ s.t. } P_k(i,j) \text{ is feasible and } X_k(j) = \texttt{True}. \\ \texttt{False} & \text{else.} \end{cases}$$

We can adjust the formula of $X_k(i)$ easily to solve different optimization problems. By defining $X_k(i)$ as $\min_{i<j\leq n_k} X_k(j) + 1$ resp. $r(i)$, where $P_k(i,j)$ is feasible, we can solve the minimum number problem resp. the minimum edge length problem.

This results in an $\mathcal{O}(n^2)$ time algorithm [7] for computing a feasible decomposition of a polygon with a linear skeleton. Note that for certain combinations of (simple) feasibility criteria and optimization goals decomposing polygons with linear skeletons is closely related to segmentation and can be done more efficiently, see for example [1].

## 2.2 General decomposition

In the following we restrict ourselves to skeletons where the degree of the skeleton points does not exceed three. If there are $m$ branching points, there will be $m + 2$ end points, namely skeleton points of degree one. We now consider decompositions consisting of subpolygons which can be generated by more than two skeleton points. As stated above those polygons can be represented as a union of subpolygons that are generated by two skeleton points. Notice that the largest number of skeleton points generating a polygon is equal to the number of leaf vertices in the skeleton tree.

The decomposition problem can be solved using a bottom-up approach in the skeleton tree. We will present the method for the minimum number problem, but as before our method can be modified for other optimization problems. The general idea is that we compute for each skeleton point the size of a minimal feasible decomposition of the subpolygon up to this point. For this, we compute entries $X_k(i)$ for all $i \in S_k$ for every vertex $v_k$ working our way up the tree $T$. Here $X_k(i)$ is the number of polygons in the minimal feasible decomposition of the polygon $P_k(i)$ corresponding to skeleton point $i$ on branch $S_k$, see Figure 6. We eventually compute the value $X_r$ for the root vertex, which is defined as the number of polygons in the minimal feasible decomposition of the entire polygon.

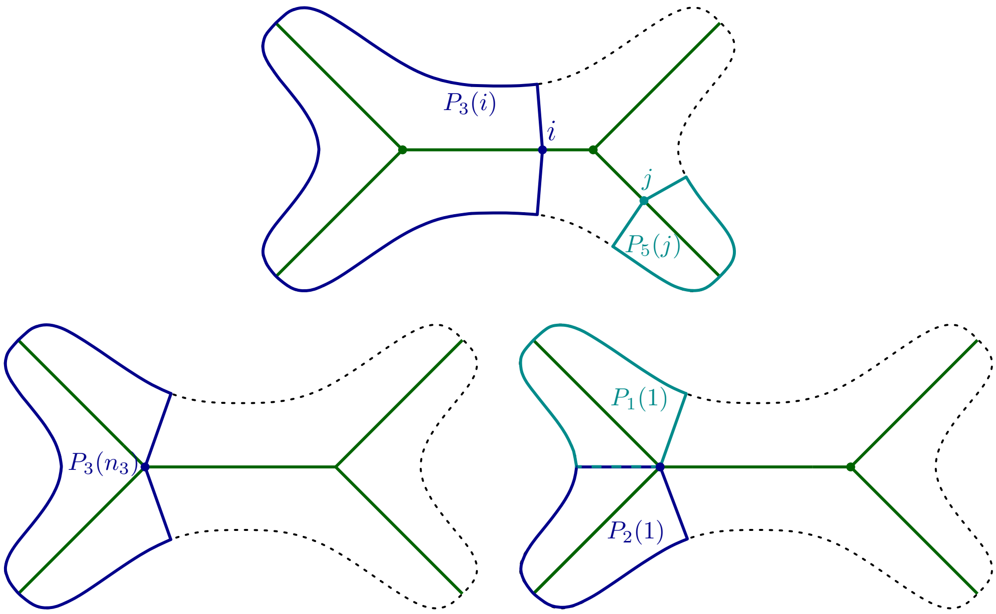



**Figure 6** Different subpolygons according to the tree representation given in Figure 5.

For computing the entry $X_k(i)$ observe that a minimal feasible decomposition of $P_k(i)$ consists of a feasible subpolygon $P$ ending at $i$ and minimal feasible decompositions of the connected components of $P_k(i)\backslash P$. Hence to compute $X_k(i)$ we search over all possible combinations of cuts, i.e. skeleton points, in the corresponding subpolygon $P_k(i)$ that together with $i$ generate a feasible polygon $P$, and return the minimal size at these.

To do this we use a function $Q[s, I, a, P]$, which searches through the subtree rooted at $s$. The parameter $I$ is a set of vertices which corresponds to the currently considered skeleton branches, on which we are searching for cuts – starting at index $s$. In $P$ the generated polygon is stored and updated as the search continues. The parameter $a$ is the size of minimal decomposition where cuts have already been chosen. Hence initially, $P$ is empty, $a = 0$, $s = i$ and $I = \{v_k\}$ for a skeleton point $i$ on branch $S_k$. Now for every vertex $v_k \in I$ we have two choices: Either we cut on the branch $S_k$ and check the possible cuts on the other branches in $I \setminus \{v_k\}$. Or we continue the search in the subtree of $v_k$ by including the children $C(v_k)$ into the set of considered branches – in this case the generated polygon contains the whole subpolygon $P_k(s, n_k)$. For the computation of $Q[s, I, a, P]$ we choose an arbitrary vertex $v_k \in I$ for the first iteration. The function is then defined as follows:

$$Q[s, I, a, P] = \min \begin{cases} \min\limits_{j \geq s \in S_k} Q[1, I \setminus \{v_k\}, a + X_k(j), P \cup P_k(s, j)], & \text{(1a)} \\ Q[1, (I \setminus \{v_k\}) \cup C(v_k), a, P_k(s, n_k)] & \text{(1b)} \end{cases}$$

$$Q[1, \emptyset, a, P] = \begin{cases} a + 1 & \text{if } P \text{ is feasible.} \\ \infty & \text{else.} \end{cases}$$

We define $X_k(i) = Q[i, \{v_k\}, 0, \emptyset]$ and $X_r = Q[1, C(r), 0, \emptyset]$. Notice that when $s = 1$ we search over all $j \in S_k$. This is the case in every iteration except for the initial one.

▶ **Example 2.1** (Two branching points). In case of two branching points we have three different cases of computation to consider, see Figure 7 for an illustration.

1. Leaf vertex: For computation of the entries of $X_k$ for $k = 1, 2, 4, 5$ the equation for $X_k(i)$ equals the formula for the linear skeleton.
2. Inner vertex: A feasible decomposition of the polygon up to the skeleton point $i$ on the branch $S_3$ contains a polygon generated by either $i$ and some $j > i$ on branch $S_3$ or $i$ and two points $(i_1, i_2) \in S_1 \times S_2$. Where the first case is calculated by (1a) as for linear skeletons and the second is calculated by (1b) as follows:

$$\begin{aligned}
&Q[1, \{v_1, v_2\}, 0, P_3(i, n_3)] \\
&= \min_{i_1 \in S_1} Q[1, \{v_2\}, X_1(i_1), P_3(i, n_3) \cup P_1(1, i_1)] \\
&= \min_{(i_1,i_2) \in S_1 \times S_2} Q[1, \emptyset, X_1(i_1) + X_2(i_2), P_3(i, n_3) \cup P_1(1, i_1) \cup P_2(1, i_2)] \\
&= \min_{(i_1,i_2) \in S_1 \times S_2} \{X_1(i_1) + X_2(i_2) + 1 \mid P_3(i, n_3) \cup P_1(1, i_1) \cup P_2(1, i_2) \text{ is feasible}\}
\end{aligned}$$

3. Root vertex: For the root we compute:

$$X_r = Q[1, \{v_3, v_4, v_5\}, 0, \emptyset] \; = \min \begin{cases} \min_{i_3 \in S_3} Q[1, \{v_4, v_5\}, X_3(i_3), P_3(1, i_3)] \\ Q[1, \{v_1, v_2, v_4, v_5\}, 0, P_3(1, n_3)] \end{cases}$$

This results in a calculation of the minimum feasible decomposition size over $(i_3, i_4, i_5) \in S_3 \times S_4 \times S_5$ for the first case and $(i_1, i_2, i_3, i_4) \in S_1 \times S_2 \times S_4 \times S_5$ for the second.

▶ **Theorem 2.2.** *Let $\mathcal{P}$ be a polygon with skeleton $S$. Let $S$ consist of $n$ skeleton points with degree less or equal 3. A feasible decomposition of $\mathcal{P}$ based on $S$ can be computed in time $\mathcal{O}(n^k)$, where $k$ is the number of leaves in the skeleton tree (or skeleton points with degree 1).*

**Proof.** We argue the runtime by assigning weights to the skeleton tree. The weight of a vertex $v$ is $g(v) = |S_v| + \prod_{w \in C(v)} g(w)$, which is the maximal number of skeleton points considered in the computation of one $X_v(i)$. The first part of this sum corresponds to (1a) and the second part to (1b). The computation of $X_v(i)$ for all $i \in S_v$ takes $|S_v| \cdot g(v)$ time. Let $L(v)$ be the number of leaves in the subtree with root $v$. We can show that $g(v) = \mathcal{O}(n^{L(v)})$. The overall runtime is asymptotically dominated by the computation of $X_r$ which takes time

$$|S_r| \cdot g(r) = g(r) = \mathcal{O}(n^{L(r)}) = \mathcal{O}(n^k). \tag{2}$$

Sketch of correctness: As we compute $X_k(i)$: A feasible decomposition of the considered subpolygon up to skeleton point $i$ consists of a feasible polygon $P$ generated by the skeleton point $i$ and feasible decompositions of the remaining polygon (or polygons). The polygon $P$ is either generated by another skeleton point on the branch $S_k$ – as computed by (1a) – or by some other skeleton points in the subtree of $v_k$ – as computed recursively by (1b). ◀

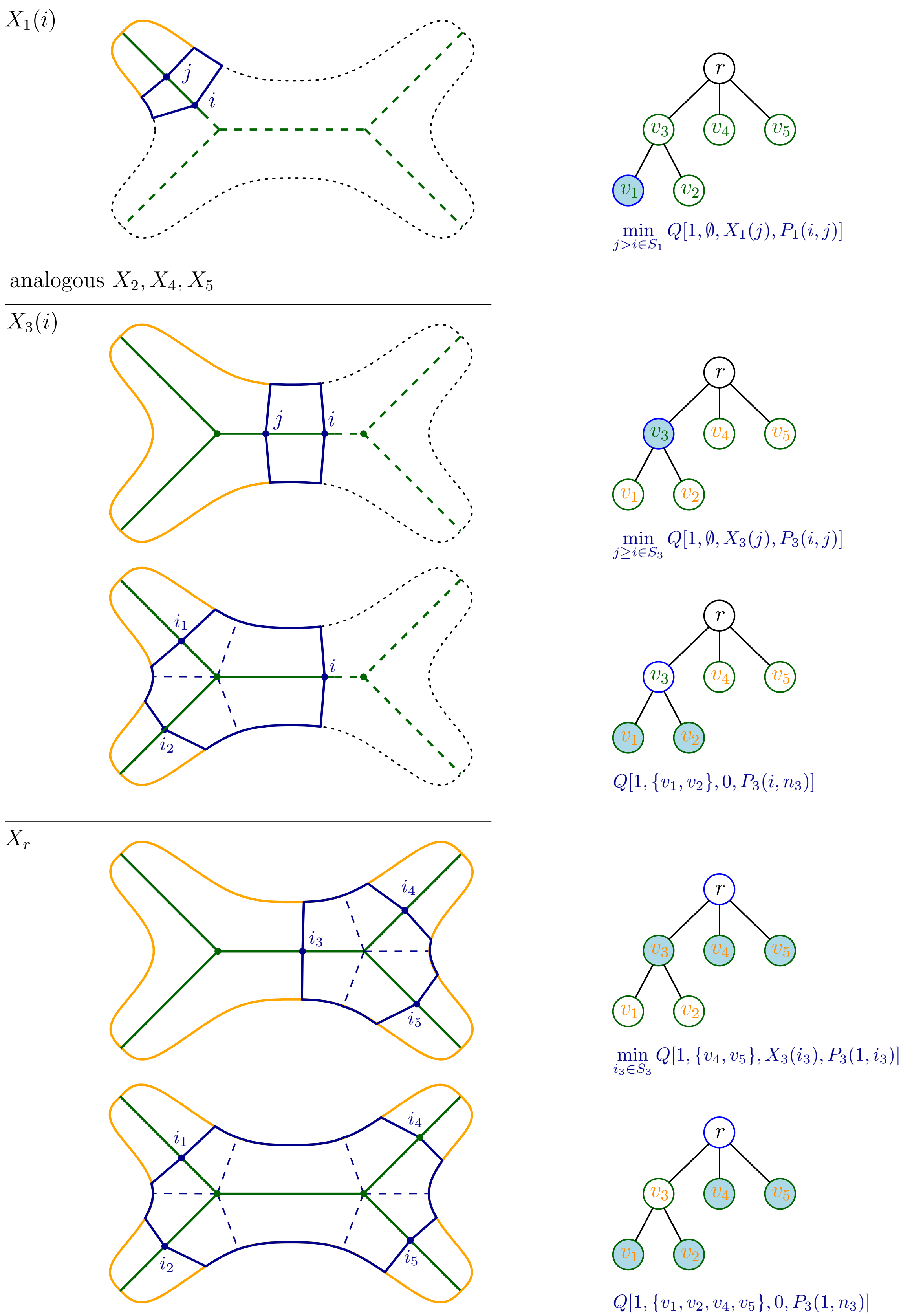



**Figure 7** Decomposition of a polygon with two branching points. The non-dashed polygon is the currently considered subpolygon and the polygon generated in a certain iteration is shown in blue.

## 3 Conclusion

We presented a method to find an optimal feasible decomposition of a simple polygon based on a discrete skeleton, which allows to include different feasibility criteria. The algorithm will be further implemented and analysed in the application. In practice we do not expect the runtime to be $\mathcal{O}(n^k)$ – as we may stop the iteration if $P$ is no longer feasible. Also we use a pruned skeleton which means that have some control over the factor $k$ – for the tissue sample in Figure 1 it was $k < 10$. We are currently working on an algorithm for higher degrees. Also we are looking into the similarities to tree resp. graph decomposition/partition problems. And there is the question if our methods can be adjusted for polygons with holes.

### References


1. Sander P. A. Alewijnse, Kevin Buchin, Maike Buchin, Andrea Kölzsch, Helmut Kruckenberg, and Michel A. Westenberg. A framework for trajectory segmentation by stable criteria. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Dallas/Fort Worth, TX, USA, November 4-7, 2014*, pages 351–360, 2014.
2. Xiang Bai, Longin Jan Latecki, and Wen-Yu Liu. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE transactions on pattern analysis and machine intelligence*, 29(3), 2007.
3. Frederik Großerueschkamp, Thilo Bracht, Hanna C Diehl, Claus Kuepper, Maike Ahrens, Angela Kallenbach-Thieltges, Axel Mosig, Martin Eisenacher, Katrin Marcus, Thomas Behrens, et al. Spatial and molecular resolution of diffuse malignant mesothelioma heterogeneity by integrating label-free ftir imaging, laser capture microdissection and proteomics. *Scientific reports*, 7:44829, 2017.
4. Kathryn Leonard, Geraldine Morin, Stefanie Hahmann, and Axel Carlier. A 2d shape structure for decomposition and part similarity. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 3216–3221. IEEE, 2016.
5. Dennie Reniers and Alexandru Telea. Skeleton-based hierarchical shape segmentation. In *Shape Modeling and Applications, 2007. SMI'07. IEEE International Conference on*, pages 179–188. IEEE, 2007.
6. Punam K Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja. A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters*, 76:3–12, 2016.
7. Leonie Selbach. Algorithmen zur Zerlegung von einfachen Polygonen unter Nebenbedingungen. Master's thesis, Ruhr-Universität Bochum, 2018.
8. Luca Serino, Carlo Arcelli, and Gabriella Sanniti di Baja. Decomposing 3d objects in simple parts characterized by rectilinear spines. *International Journal of Pattern Recognition and Artificial Intelligence*, 28(07):1460010, 2014.
9. Maryann Simmons and Carlo H Séquin. 2d shape decomposition and the automatic generation of hierarchical representations. *International Journal of Shape Modeling*, 4(01n02):63–78, 1998.
10. Mirela Tănase and Remco C Veltkamp. Polygon decomposition based on the straight line skeleton. In *Geometry, Morphology, and Computational Imaging*, pages 247–268. Springer, 2003.