Polygon Decomposition based on the Straight Line Skeleton

Mirela Tănase
Institute of Information & Computing Sciences
P.O. Box 80089
3508 TB Utrecht
The Netherlands
mirela@cs.uu.nl

Remco C. Veltkamp
Institute of Information & Computing Sciences
P.O. Box 80089
3508 TB Utrecht
The Netherlands
Remco.Veltkamp@cs.uu.nl

ABSTRACT

We propose a novel type of decomposition for polygonal shapes. It is thought that, for the task of object recognition, the human visual system uses a part-based representation. Decompositions based on skeletons have been previously proposed in computer vision. Our method is the first one, however, based on the straight line skeleton. Compared to the medial axis, the straight line skeleton has a few advantages: it contains only straight segments and has a lower combinatorial complexity. The skeletal nodes and the way they are generated are the basis for our decomposition, which has two stages that result in a decomposition into (possibly overlapping) parts. First, a number of visually striking parts are identified, then their boundaries are successively simplified, by locally removing detail. Our method runs in time $O((n + r_1 + r_2^2) \log^2 n)$, after the skeleton construction, where n is the number of vertices in the polygon, r_1 the number of split events, and r_2 the number of reflex edge annihilations. The decomposition is invariant to rigid motions and uniform scalings. We present results indicating that it provides plausible decompositions for a variety of shapes. This makes it attractive for partial shape matching in content-based image retrieval.

Categories and Subject Descriptors

I.3.5 [**Computing Methodologies**]: Computer Graphics—*Computational Geometry and Object Modeling*

General Terms

Algorithms

Keywords

Polygonal decomposition, straight line skeleton

1. INTRODUCTION

It is thought that, for the task of object recognition, the human visual system uses a part-based representation. In computer vision, many recent approaches to the decomposition problem draw inspiration from the human perception theories. Existing decomposition methods can be classified into those that are boundary-based, using only contour information for extracting parts, and those that are region-based, using information about the interior of the shape. Probably the most known and influential approach in the first category is that proposed by Hoffman and Richards [1]. Their *minima rule* states that for any silhouette, the negative curvature minima of the contour are boundary points that separate different parts. This rule led to a representation of the shape boundary based on *codons*, or pieces of boundary bounded by curvature minima. Although the minima rule indicates a precise set of points on the shape boundary that mark the demarcation between parts, it does not indicate how to join these points in order to obtain the actual parts.

When Blum introduced the medial axis [2] back in 1967, a decomposition at its branching points was also suggested. The medial axis is region-based and can be defined as the locus of centers of maximally inscribed disks. Variations of it include smoothed local symmetries [3] and, for polygonal shapes, the straight line skeleton [4]. The medial axis is capable of capturing important visual cues of the shape, such as symmetry and complexity (reflected by branching points). It also allows recovery of the original shape and provides rich local support (local boundary modifications affect only a small portion of the skeleton). These properties make it attractive for shape description. However, one disadvantage of it is its sensitivity to noise. Moreover, its computation on raster data also poses problems: it often leads to non-connected axes and it may require extensive preprocessing (smoothing of the boundary) or post-processing (pruning of spurious edges). Some of these problems can be overcome by a computation based on Voronoi diagrams of the boundary points, together with a hierarchical clustering of the skeleton components [6].

Along with the medial axis branching points, local minima of the maximally inscribed disk radii have been suggested as useful for a decomposition into parts. A different approach [7] to using skeletons for shape decomposition, associates to the medial axis a weighted graph, called axial shape graph. The weights are capturing both local and global information about the shape. The decomposition is derived by repeatedly partitioning the axial shape graph into two sets of nodes of roughly equal weights.

Siddiqi and Kimia [8] proposed a way of combining extremal curvature boundary points with region information, since using only boundary information seems insufficient. Their decomposition involves two types of parts, limbs and necks. Limbs are the result of a partitioning through a pair of negative curvature minima with a smooth continuation of their boundary tangents. Necks arise

SoCG'03, June 8-10, 2002, San Diego, USA.

from narrowings in the shape, characterized by local minima of the maximally inscribed disk radii. Limbs and necks are also related to singularities (shocks) in a curve evolution process, based on a reaction-diffusion equation, introduced by Kimia, Tannenbaum and Zucker [9]. A contour evolution under constant deformation (contour points move along the normal with constant speed) is equivalent to Blum's grass-fire interpretation of the medial axis. The locus of shocks formed under constant deformation of the boundary is thus the medial axis. The formation of the shocks however provides additional information to the axis, and this is exploited in an organization of shocks into a shock-graph [10].

A contour evolution and hierarchical decomposition method is proposed by Latecki and Lakämper [11]. They propose a simple contour simplification process: at every step, the two consecutive line segments with the least relevance, as defined by some relevance measure, are substituted with a single line segment joining their endpoints. This process yields a hierarchy of simplifications, which are used next in the decomposition, starting from the highest level in the hierarchy (corresponding to the simplest shape in the evolution) down to the lowest level (corresponding to the original shape). The parts of the contour at a given level of the shape hierarchy are determined by the maximal convex arcs with respect to the object.

Polygon decomposition is a well-established research area in Computational Geometry. There is a substantial body of literature that focuses on developing efficient algorithms for partitioning or covering a polygon with the smallest number of a particular type of subpolygons (triangles, convex, monotone, spiral or star-shaped subpolygons), see [12] for a survey. The emphasis on the optimality of the decomposition comes from the fact that polygon decomposition often serves as a preprocessing step for many geometric algorithms. Many geometric problems have simpler and faster solutions on such a restricted type of polygon, so the strategy of solving these problems for general polygons is to decompose them into simpler parts, solve the problem on each part and combine the partial solutions. For polygon recognition purposes, however, this may not be a desirable property of the decomposition. Moreover, the results of a decomposition into components of a fixed type, convex polygons included, do often correlate poorly with human judgement.

1.1 Our Contribution

The results presented in this paper are: (i) a new decomposition of polygons, (ii) a new taxonomy of events in the wavefront propagation, (iii) a new result on the form of cells swept out during the wavefront propagation that is used to proof a lemma about the number of vertices introduced by the decomposition, (iv) a favourable experimental comparison with other state-of-the-art methods.

The novel type of decomposition for polygonal shapes is based on the straight line skeleton, a variant of the medial axis, introduced by Aichholzer and Aurenhammer [4]. The general advantages of the straight line skeleton, as compared to the medial axis, are its composition of only straight segments and its lower combinatorial complexity. Similar to the grass-fire analogy of the medial axis, the straight line skeleton is defined by a wavefront propagation process. In this process, edges of the polygon move inward at a fixed rate. The counterparts of the medial axis branching points, called nodes, are induced by combinatorial changes of the wavefront (events). The nodes and the way they are generated are central to our decomposition method. The wavefront events associated with the nodes endows them with additional information, in the same way as shocks add a coloring to the medial axis points in shock graphs [10]. We exploit this information in deciding how the node contributes to our decomposition (globally, locally or not at all). This results in a two-step decomposition. Nodes corresponding to splittings of the propagating wavefront (split events) lead to a decomposition into non-overlapping parts. These are usually the most striking parts of the polygon. Nodes corresponding to annihilation of edges (edge events) lead to a local simplification of the boundary of these parts, together with a removal of smaller protrusions. The successive boundary simplifications constitute the second step of our decomposition. The final output is a decomposition into overlapping parts. In both stages, the actual computation of the parts is based on a backward propagation process, similar to the inward propagation that generates the skeleton. The decomposition is invariant to rigid motions and uniform scalings. It can be computed in $O((n + r_1 + r_2^2) \log^2 n)$ time, after the straight line skeleton computation, where n is the number of vertices in the polygon, r_1 the number of split events, and r_2 the number of reflex edge annihilations in the skeleton construction. We present results indicating that it provides intuitive decompositions.

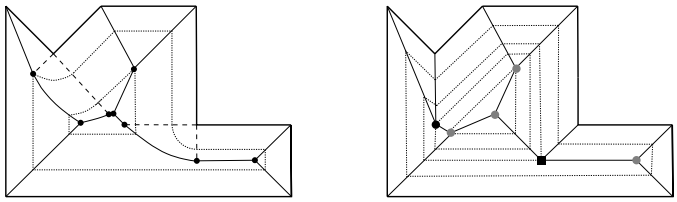

Figure 1: Medial Axis (a) vs. Straight Line Skeleton (b). In (b) the black disk marks a reflex edge annihilation, while gray disks mark convex edge annihilations. An edge-edge collision generates the arc between the black box (vertex-edge collision) and a gray disk (convex edge annihilation).

The rest of this paper is organized as follows. In section 2 we present the straight line skeleton. Specifically, we discuss its relation with the medial axis, give a complete categorization of the events occurring in the wavefront propagation and proof a new result used in one of our lemma's. We detail the proposed decomposition method in section 3 and analyze its computational complexity in section 4. Examples of the implementation results are presented in section 5, which is followed by some concluding remarks in section 6.

2. STRAIGHT LINE SKELETON

Aichholzer and Aurenhammer [4] introduced the straight line skeleton, a new type of skeleton for polygons. It is closely related to the medial axis, being also based on a wavefront propagation. In the grass-fire analogy of the medial axis, a wavefront is defined as the set of points having some fixed distance to the original polygon. The wavefront consists of straight line segments and circular arcs (see figure 1(a)) and, as the distance to the polygon increases (a process also called wavefront propagation), the breakpoints between consecutive line segments and circular arcs trace the Voronoi diagram of the polygon. By removing the segments in the diagram incident to the reflex vertices, we obtain the medial axis, which consist of straight line segments and parabolic arcs.

To construct the straight line skeleton, we also let wavefront edges move parallel to the polygon sides. In contrast to the medial axis, edges incident to a reflex vertex will grow in length. The front remains a polygon, whose vertices during the process trace out the skeleton (see figure 1(b)). As its name suggests, it consists of straight line segments only. It also has a smaller combinatorial complexity than the medial axis.

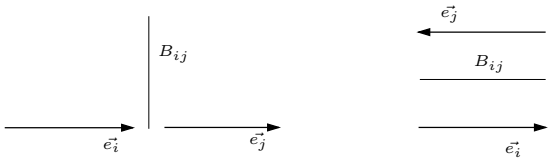

Figure 2: Degenerate bisectors.

In this section we take a closer look at the straight line skeleton definition, identifying all the possible morphological changes in the wavefront, as these events are of prime importance for our decomposition. We also review the known properties of the straight line skeleton and extend a previous result.

2.1 Preliminaries

Let P be a simple polygon, with a counterclockwise order defined on its edges. The orientation of the polygon is of no consequence for our decomposition, but simply a convention. If $\vec{e_i}$ is an edge of P, let $l_{\vec{e_i}}$ denote its supporting line.

The bisector B_{ij} of $\vec{e_i}$ and $\vec{e_j}$ is the half-line angular bisector of $l_{\vec{e_i}}$ and $l_{\vec{e_j}}$ lying to the left of both $\vec{e_i}$ and $\vec{e_j}$. As we do not assume the polygon to be in general position, we need to define two degenerate bisectors as in figure 2. For two disjoint edges, $\vec{e_i}$ and $\vec{e_j}$, having the same orientation and supporting line, B_{ij} is the ray perpendicular to $l_{\vec{e_i}}$, originating in a point of $l_{\vec{e_i}}$, equally distant to $\vec{e_i}$ and $\vec{e_j}$ and to the left of both (see figure 2, left). If $\vec{e_i}$ and $\vec{e_j}$ have opposite orientation and parallel supporting lines, B_{ij} is the line parallel to $l_{\vec{e_i}}$ and $l_{\vec{e_j}}$, and equally distant to them (see figure 2, right).

We call an edge whose endpoints are convex vertices of P, a *convex edge*. An edge with at least one endpoint being a reflex vertex of P, will be called a *reflex edge*.

The following propagation process is defined on P: edges are translated at constant speed remaining self parallel, keeping sharp corners at reflex vertices (see figure 1(b)). Such a shrinked version of P is called a *wavefront*. During propagation, the wavefront can split, after which it is a union of simple, disjoint polygons. Wavefront vertices move, in the propagation, on angular bisectors of wavefront edges, which in turn may increase or decrease in length. If x is a point inside P, we denote by $\mathcal{F}(x)$ the wavefront that has x on its boundary.

DEFINITION 1. *The* straight line skeleton *of P, $S(P)$, is defined as the union of pieces of angular bisectors traced out by the wavefront vertices during the above propagation process.*

These pieces of bisectors are called *arcs*. The endpoints of the arcs in $S(P)$ that are not vertices of P are called *nodes*.

2.2 Event taxonomy

In the propagation, the wavefront repeatedly changes continuously until events occur. The following is a complete categorization of these events:

1. *Edge Events.* A wavefront edge may collapse into a point, its neighbouring edges becoming adjacent. We distinguish:
   (a) *Convex Edge Annihilation.* If the collapsing edge is a convex edge of P.
   (b) *Reflex Edge Annihilation.* If the collapsing edge is a reflex edge of P. In this case, also a reflex vertex disappears from the front.

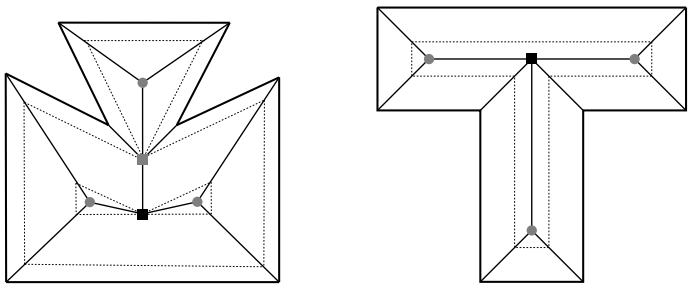

Figure 3: (a) Events in the wavefront propagation: gray disks mark convex edge annihilations, the black box marks a vertex-edge collision, while the gray box is a vertex-vertex collision. (b) Front annihilation: the front collapses down to two line segments.

   See figure 1(b) for examples of convex and reflex edge annihilations. Multiple edge events can occur at the same location.
   (c) *Front Annihilation.* An entire simple polygon of the wavefront may collapse into a point (equivalent with at least three edge annihilations at the same location), into a line segment, or into two or more line segments. (See figure 3(b)).
2. *Split Events.* A simple polygon of the wavefront may be split into a number of polygons following one or a combination of the following events at the same location:
   (a) *Vertex-Edge Collision.* A reflex vertex of the front may collide into an edge, thus splitting the front into two (see figure 3(a)).
   (b) *Vertex-Vertex Collision.* Two vertices (at most one being convex) may meet at the same location, splitting the front into two. See figure 3(a), where the event, marked with a gray box, is generated by two reflex vertices that reach the same location simultaneously. Unlike edge annihilations or vertex-edge collisions, this kind of event can introduce a new reflex vertex to the wavefront.
   (c) *Edge-Edge Collision.* Two parallel front edges may collapse into each other (see figure 3(b)). This may give rise to a front splitting at one or both endpoints of the arc in the $S(P)$ generated by the event. See figure 1(b) where such an event generates a splitting of the front into a simple polygon and a degenerate rectangle.

Note that in [4] a split event is only what we call a vertex-edge collision. In [13] a vertex-vertex collision between two reflex vertices is called a vertex event. Vertex-vertex collisions involving a convex vertex and edge-edge collisions do not appear in their classification. The reason for this is that a small perturbation of the polygon removes these types of events from the skeleton without altering its structure. Vertex-vertex collisions between two reflex vertices, on the other hand, cannot be removed by perturbations of the original polygon without changing the structure of the skeleton.

2.3 Properties of the Straight Line Skeleton

For a simple polygon with n vertices, $S(P)$ has at most $n-2$ nodes and $2n-3$ arcs, and both bounds are smaller than those for

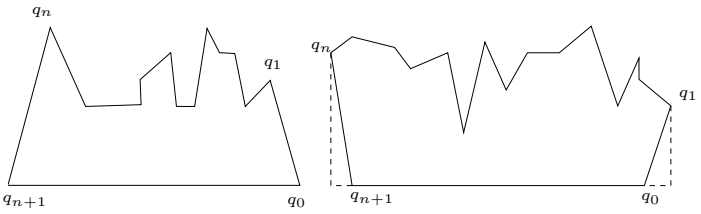

Figure 4: A normal histogram (left) and a pseudo-normal histogram (right)

the medial axis ($n + r - 2$ and $2n + r - 3$, respectively, where r is the number of reflex vertices) [4].

$S(P)$ gives rise to a polygonal partition of the interior of P. Each cell in this partition is the region swept out during the propagation by one of the edges of P. For the rest of this paper, we say that an edge $\vec{e}$ is *involved in an event*, if the node introduced in $S(P)$ by that event is on the boundary of the cell swept out by $\vec{e}$. We denote the cell swept out by $\vec{e_i}$ by $C(\vec{e_i})$. Aichholzer and Aurenhammer [4] observed that the cells of $S(P)$ are monotone polygons. We extend this result by showing that they are quasi-normal histograms.

A *normal histogram* is a polygon that is monotone with respect to one of its edges (see figure 4, left).

A monotone polygon is called *quasi-normal histogram* if one of the two monotone chains is convex. If the convex chain has two or three edges, we call it a *pseudo-normal histogram* (see figure 4, right).

LEMMA 1. *The cells of $S(P)$ are quasi-normal histograms.*

PROOF. Let $\vec{e}$ be an edge of P and $C(\vec{e})$ its corresponding cell. Let $C(\vec{e}) = \{q_0, q_1, ..., q_{n+1}\}$, with $\vec{e} = q_{n+1}q_0$. We can assume without loss of generality that $\vec{e}$ is horizontal and $q_0^x > q_{n+1}^x$. The first observation is that new reflex vertices may appear in the wavefront only following a vertex-vertex collision.

Let q_k (q_p) be the first event that is not a vertex-vertex collision in a counterclockwise (clockwise) traversal of $C(\vec{e})$ starting from q_0. After both these events occur, the wavefront edge(s) originating from $\vec{e}$ are incident only with convex vertices. Thus in the subchain $C_2 = \{q_k, ..., q_p\}$ there is no vertex-vertex collision that gives rise to a new reflex vertex.

Because the subchain $C_1 = \{q_p, ..., q_k\}$ is convex, it suffices to show that C_2 is monotone with respect to the horizontal line $l_{\vec{e}}$. We have thus to show that ${q_k}^x \geq {q_{k+1}}^x \geq \ldots \geq {q_p}^x$. If there exists $i \in \{k, \ldots, p-1\}$ so that ${q_i}^x < {q_{i+1}}^x$, then let $\vec{e'}$ be the edge in P so that $q_i q_{i+1}$ is a piece of the bisector of $\vec{e}$ and $\vec{e'}$. This implies that at the moment that the front reaches q_i, the wavefront edges corresponding to $\vec{e}$ and $\vec{e'}$ become adjacent and form a reflex vertex. This contradicts the fact that there are no such vertex-vertex collisions in C_2, so C_2 is monotone with respect to $l_{\vec{e}}$. □

The following corollary is used in proving the bounds on the number of Steiner points added to the decomposition in the first phase of the proposed method.

COROLLARY 1. *If an edge of P is not involved in any vertex-vertex collision, then its corresponding cell is a pseudo-normal histogram.*

3. PROPOSED DECOMPOSITION

We saw in the previous section that in the propagation process, the wavefront suffers two types of modifications: it can be split into two or more parts and edges of the front can disappear. The first type of modifications are generated by split events. They give information about the number and the configuration of the most striking parts of the polygon. The second type of modifications are generated by edge events. Through these events the boundary of the shape is simplified by removing the edges roughly in an order of increasing significance. A special type of edge event is the reflex edge annihilation, because it corresponds to a protrusion of the shape. All events are related to changes in the shape of the wavefront during propagation. But while split events are induced by drastic modifications of the wavefront, reflex edge annihilations are caused by smaller protrusions of the polygon, that annihilate 'smoothly' in the propagation. The events generated in the propagation process capture thus two different types of information about the shape. It is natural then to use the split events first in order to identify the most striking parts, and then use the reflex edge annihilations to simplify those parts, by removing detail (protrusions). Our decomposition process has these two stages.

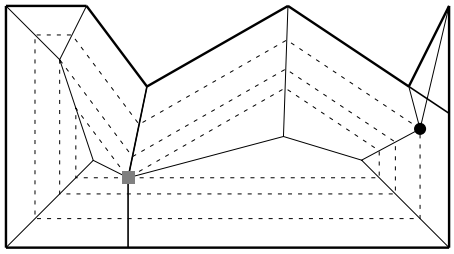

Figure 5: Straight line skeleton and a decomposition induced by its split and reflex edge events.

The basic idea for computing the parts in our decomposition is to reverse the process that caused the events in the first place, by back-propagating the wavefront at the moment that any of these events occur. As explained in detail below, each stage of the decomposition process handles the events sequentially, in the increasing chronological order of their occurrence. In the first stage, each split event induces a division of one of the parts in the current decomposition. In the second stage, reflex edge annihilations induce simplifications of the parts computed in the first stage, removing some of their smaller details (protrusions). In figure 5, we see a very simple example of such a decomposition. The vertex-edge collision, marked with a square box, is a split event that induces a decomposition of the polygon into two parts. Next, the protrusion in the larger right part is removed with the handling of the reflex edge annihilation marked with a disk in the figure.

In the rest of this section we concentrate on the computation of the proposed decomposition.

3.1 Phase 1: Global Splitting

The handling of all split events results in a decomposition of P into non-overlapping parts. The split events are treated in the order that they occur in the inward propagation process. Degenerate cases (simultaneous split events at the same location) can be treated in arbitrary order.

A *split-chain* is a polygonal chain that divides p into two components, with both endpoints on the boundary of a part p in the current decomposition of P and all the other points inside p. Each split event induces such a split-chain that will divide one of the parts in the previously computed decomposition into two. In our decompo-

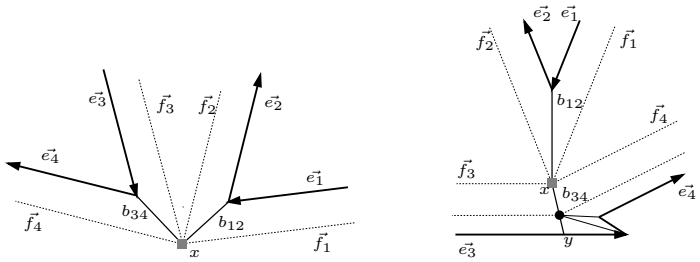

Figure 6: Vertex-vertex collisions and the split-chains they generate. Thick arrows indicate ordered contour edges, dotted line indicate wavefront edges, solid lines denote the split chain.

sition, we construct the split-chain in a back-propagation process, similar to the one that leads to the straight line skeleton. We prove that any split-chain consists of at most three segments, with one or two new Steiner points being added to the decomposition.

Let x be a point inside P so that when the wavefront reaches x it is split. The wavefront at the moment x is reached, $\mathcal{F}(x)$, contains a union of two simple polygons, with disjoint interiors and having x as a common vertex. We propagate backwards these polygons simultaneously and at the same speed until the edges of $\mathcal{F}(x)$ adjacent to x reach their counterparts on the original polygon. The split-chain induced by x is the trace of the intersection of the back-propagating polygons.

Before detailing the computation of the split-chain for each type of split event, we make a few observations. First, the inward propagation process defined in Section 2.1 is not a reversible one. The result of the backward propagation process, a polygon P', is usually not the original polygon P, but a simplified version of it. Edges of P lost in edge events during propagation are not in P'. Secondly, an edge-edge collision is taken care of by handling its endpoints. If an edge-edge collision splits the wavefront at one of its endpoints then this endpoint is handled as a vertex-edge collision, otherwise an edge of the wavefront must also collapse into that endpoint, so the endpoint is treated as an edge event.

3.1.1 Handling a Vertex-Vertex Collision

Let x be a vertex-vertex collision: two vertices of the front and nothing else meet at x, and split the front. In figure 6, at the moment that x is reached, the wavefront $\mathcal{F}(x)$ is split into two polygons. They share a vertex (x) in $\mathcal{F}(x)$, $\vec{f_1}$ and $\vec{f_4}$ are adjacent in one polygon, while $\vec{f_2}$ and $\vec{f_3}$ are adjacent in the other. The split-chain is defined as the trace of the intersection of these two polygons in their propagation backwards to the original polygon. The growth of the edges in the back-propagation is restricted by the interaction with each other, in the same way as in the inward propagation defining the skeleton. Let p be the part in the current decomposition of P that has x in its interior. This is the part that will be divided through a split-chain in the handling of the event. We now prove a bound on the length of the split-chain induced by a vertex-vertex collision. The bound on the length of the part-chain, in this case, is given by the following lemma.

LEMMA 2. *The split-chain induced by a vertex-vertex collision consists of two line segments, introducing one Steiner point (the event itself) to the decomposition.*

PROOF. If x is our vertex-vertex collision, let $\vec{f_i}$, $i = 1 \ldots 4$, be the edges of $\mathcal{F}(x)$ incident to x so that, prior to the event, $\vec{f_1}$ was incident to $\vec{f_2}$ and $\vec{f_3}$ to $\vec{f_4}$, respectively. Let $\vec{e_i}$, $i = 1 \ldots 4$, be their counterparts on P. Shortly before reaching x, the two front vertices that caused the event were moving along arcs of $S(P)$. Let b_{12} be the arc of $S(P)$, piece of the bisector B_{12} of $\vec{e_1}$ and $\vec{e_2}$, and b_{34} be the arc of $S(P)$, piece of the bisector B_{34} (see figure 6). One of these arcs (b_{12}, for example) can have length zero, when the vertex-vertex collision coincides with the moment that the front edges corresponding to $\vec{e_1}$ and $\vec{e_2}$ become incident. In other words, when x is also an edge event.

We distinguish two cases:

- *Reflex-Reflex Collision.* If the vertices that caused the event were both reflex (see figure 6, left), b_{12} has one endpoint at x, and one on P, or at a vertex-vertex collision, wherein the wavefront edges corresponding to $\vec{e_1}$ and $\vec{e_2}$ became adjacent. In either case this endpoint of b_{12} is on the boundary of a part in the current decomposition. The same holds for b_{34}. So the split-chain in this case is included in $b_{12} \cup b_{34}$.

- *Reflex-Convex Collision.* If one of the vertices is convex, assume this is the one that moved previously along b_{34} (see figure 6, right). Based on the above mentioned argument, b_{12} has its endpoint other than x on the boundary of a part in the current decomposition. The other endpoint of b_{34}, though, may or may not lie inside p. The split-chain induced by x in this case is included in $b_{12} \cup B_{34}$. In figure 6 right, the split-chain is given by $b_{12} \cup xy$, where y is the intersection of the bisector B_{34} with the boundary of p.

So in both cases the split-chain consists of two segments. □

3.1.2 Handling a Vertex-Edge Collision

Let x be a vertex-edge collision: at the moment that the wavefront reaches x, x is a reflex vertex of $\mathcal{F}(x)$ that collides with and splits a wavefront edge. As in the previous case, the split-chain is the trace of the intersection of two polygons in $\mathcal{F}(x)$ in their back-propagation towards P.

LEMMA 3. *The split-chain induced by a vertex-edge collision consists of two or three line segments, introducing one or two Steiner points to the decomposition.*

PROOF. If x is a vertex-edge collision, let $\vec{e_1}$ and $\vec{e_2}$ be the edges of P that propagated gave rise to the reflex vertex x in $\mathcal{F}(x)$. If $\vec{e}$ denotes the wavefront edge that is split in the event, let $\vec{e_3}$ be its counterpart on P. Finally, let p be the part in the current decomposition that has x in its interior.

1. Tracing the intersection point of the fronts backwards to the vertex. On this side of x, that intersection point moves along the

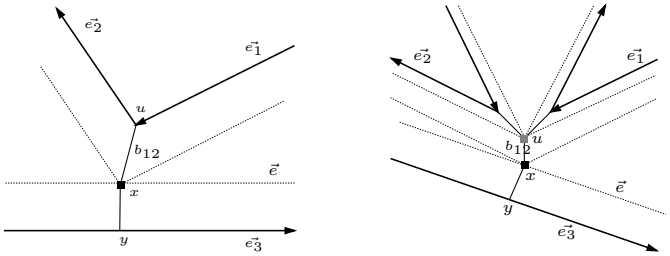

Figure 7: The reflex vertex generating a vertex-edge collision at x can originate in a vertex of P (left) or may have appeared in the propagation process (right). In both cases, the split-chain contains on this side of x only one segment ($xu = b_{12}$). On the other side of x, the split-chain contains one segment (xy) also, but only when y, the projection of x onto the edge that is split, falls inside it.

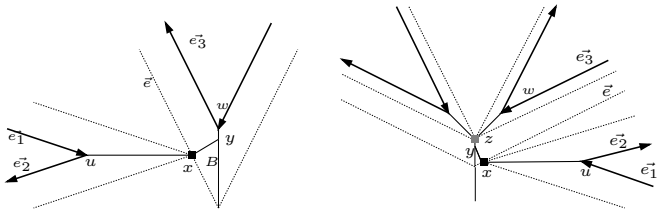

Figure 8: The split-chain may contain, on the side of the edge that is split ($\vec{e_3}$), two segments (xy and yw on the left, xy and yz on the right). This may happen only when the projection of the event x onto $l_{\vec{e_3}}$ falls outside $\vec{e_3}$.

arc b_{12} in $S(P)$, piece of the bisector B_{12} of $\vec{e_1}$ and $\vec{e_2}$. This has one endpoint in x, the other lies on P, or is a vertex-vertex collision prior to x in the propagation (see figure 7). In both cases it is on the boundary of a part in the current decomposition. So the intersection trace contains on this side of x at most one segment.

2. Tracing the intersection of the polygons backwards to the edge. On this side of x, the intersection is traced along the line perpendicular to the supporting line $l_{\vec{e_3}}$ of $\vec{e_3}$. If the projection of x on $l_{\vec{e_3}}$ falls inside $\vec{e_3}$ (see figure 7), there are no morphological changes in the back-propagating front along the intersection trace, so the split-chain on this side contains only one segment. Let's suppose now that the projection of x on $l_{\vec{e_3}}$ falls outside $\vec{e_3}$ (see figure 8). Because $x \in \vec{e}$, which originates in $\vec{e_3}$, x lies inside the cell $C(\vec{e_3})$ in the subdivision of P induced by $S(P)$. It follows that in projecting x on $l_{\vec{e_3}}$ we meet first the boundary of $C(\vec{e_3})$. It also follows that the closest endpoint of $\vec{e_3}$ to the above projection must be a reflex vertex w. Let B denote the arc of $S(P)$ incident to w. If $\vec{e_3}$ is not involved in any vertex-vertex collision previous to x (see figure 8 left), from Corollary 2 we have that in projecting x on $l_{\vec{e_3}}$, we must intersect B. If we denote this intersection by y, it's interpretation is the following: one of the pieces of $\vec{e}$ shrinks to y in the back-propagation, and the other piece becomes adjacent with the other edge of P incident to w. This way, if y is not already on the boundary of p, our intersection is further traced along B towards w. In this case, the trace on this side of x contains two segments sharing y as endpoint. If $\vec{e_3}$ is involved in at least one vertex-vertex collision, which must be previous to x (see figure 8 right), then in projecting x onto $l_{\vec{e_3}}$ we may meet any of the segments on the boundary of $C(\vec{e_3})$ between w and x. The intersection trace again may contain at most two segments, as all the vertex-vertex collisions on the boundary of $C(\vec{e_3})$ between w and the last vertex-vertex collision (z in figure 8 right) prior to x were handled, so all these segments in $C(\vec{e_3})$ between w and z are edges of parts in our decomposition. □

3.2 Phase 2: Locally Removing Detail

A decomposition of P into r_1 non-overlapping parts is the outcome of the first phase of the algorithm, where r_1 is the number of split events in $S(P)$. Let P_i, $i = 1, \ldots, r_1$, denote these parts. We see three examples of such decompositions induced by split events in the left column of figure 11. We can further simplify some of the parts in these decompositions. Reflex edge annihilations are related to protrusions of the shape. In the second step of the algorithm, these events are handled in the increasing chronological order of their occurrence, in order to locally simplify the boundary by removing detail (protrusions). We use for this purpose the events in the straight line skeleton of the original polygon. A recomputation of the skeletons for P_i would induce new splittings that are not significant for the original polygon. In the rest of this section, we give a simple method for locally simplifying the boundary of the parts P_i based on reflex edge annihilations. Each reflex edge annihilation inside P_i induces a simplification of the boundary of P_i. Because we do not know in advance the appropriate level of detail removal needed, we generate the whole sequence of simplifications for each P_i. This leads to a decomposition of P into overlapping parts.

Let x be a reflex edge annihilation: a wavefront edge incident to a reflex vertex collapses down to x, its neighbouring edges becoming adjacent and forming a convex angle in $\mathcal{F}(x)$ (see figure 9). Let $\vec{f_1}$ and $\vec{f_2}$ (in this order along the wavefront) denote the edges in $\mathcal{F}(x)$ incident to x, and $\vec{e_1}$, and $\vec{e_2}$ their counterparts on P. We also denote by $\vec{e}$ the edge in P collapsing in the event. On P, $\vec{e_1}$ and $\vec{e}$ may not be adjacent, but then the edges in between $\vec{e_1}$ and $\vec{e}$ collapse before $\vec{e}$ during the wavefront propagation. The same holds for $\vec{e}$ and $\vec{e_2}$. We call the chain $\mathcal{P}(x) = \{\vec{e_1}, \ldots, \vec{e_2}\}$ a *protrusion-chain.* Our purpose is to find a simplification of this chain that cuts the protrusion off. A *protrusion-cut* is a segment with its endpoints resting on the extremal edges, $\vec{e_1}$ and $\vec{e_2}$, of the protrusion-chain. Each reflex edge annihilation induces, in our decomposition, such a protrusion-cut that locally simplifies the boundary by removing the protrusion. In figure 9, we have a simple example of a boundary simplification. The protrusion-chain here consists of only 3 edges ($\vec{e_1}$, $\vec{e}$ and $\vec{e_2}$), and the dashed line shows how its corresponding protrusion is separated from the part. We now describe a way of computing such a protrusion-cut, which is also based on a back-propagation.

We consider this process only in the close vicinity of the event x, as we are interested in simplifying only a portion of the boundary. It can be shortly described as follows. We back-propagate the two wavefront edges in $\mathcal{F}(x)$ incident to x until we hit *both* their counterparts on P. Among the points of $\vec{e_1}$ and $\vec{e_2}$ that were reached by the back-propagating chain we select the two points, $u \in \vec{e_1}$ and $v \in \vec{e_2}$, that are closest *along* $\mathcal{P}(x)$. They define the endpoints of the protrusion-cut segment induced by x.

The back-propagation of these two edges is done in the same way as the inward propagation that defines the straight line skeleton. Their growing, however, is restricted by the intersection with P. Their intersection point moves along B_{12}, the bisector of $\vec{e_1}$ and $\vec{e_2}$, towards its origin $\{q\} = l_{\vec{e_1}} \cap l_{\vec{e_2}}$. We will show that defined in this way, the computation of the protrusion-cut is very simple, requiring little information, namely only the relative position of $\vec{e_1}$ and $\vec{e_2}$ with respect to q. We first note that the areas swept in the inward propagation by $\vec{e_1}$ and $\vec{e_2}$, before x is reached, are disjoint. As a consequence, only the following configurations can occur (see

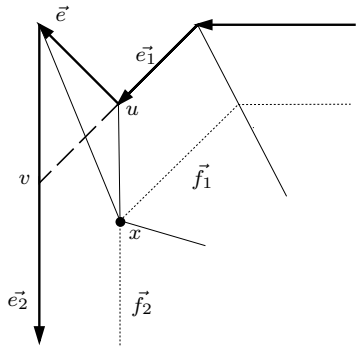

Figure 9: A reflex edge annihilation (x) and the protrusion-cut it generates (the dashed line).

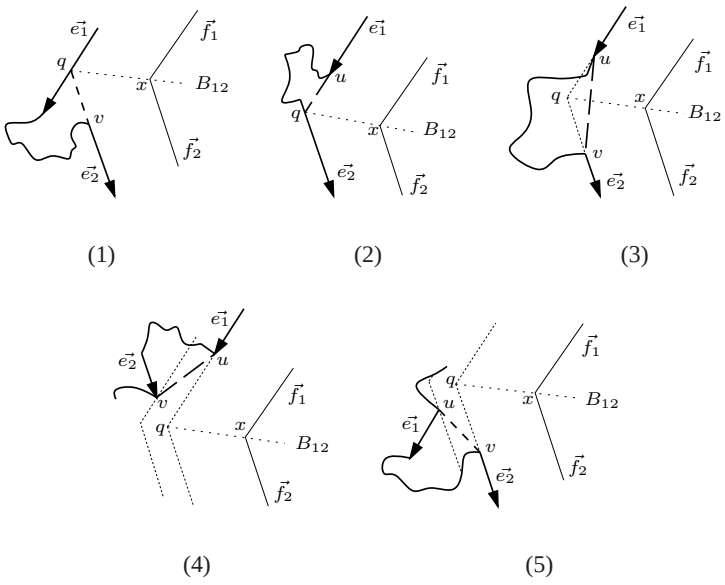

Figure 10: There are only 5 possible configurations of $\vec{e_1}$ and $\vec{e_2}$ with respect to $\{q\} = l_{\vec{e_1}} \cap l_{\vec{e_2}}$, that propagated become adjacent in a reflex edge annihilation x. The protrusion-cut induced by the event, in each case, is the thick dashed segment.

figure 10):

1. $q \in \vec{e_1}$ and $\vec{e_2}$ is to the left of $l_{\vec{e_1}}$ (as defined by the orientation of $\vec{e_1}$);
2. $q \in \vec{e_2}$ and $\vec{e_1}$ is to the left of $l_{\vec{e_2}}$;
3. $q \notin \vec{e_1}$, $q \notin \vec{e_2}$, $\vec{e_1}$ is to the left of $l_{\vec{e_2}}$ and $\vec{e_2}$ is to the left of $l_{\vec{e_1}}$;
4. $q \notin \vec{e_1}$, $q \notin \vec{e_2}$, $\vec{e_1}$ is to the left of $l_{\vec{e_2}}$ and $\vec{e_2}$ is to the right of $l_{\vec{e_1}}$;
5. $q \notin \vec{e_1}$, $q \notin \vec{e_2}$, $\vec{e_1}$ is to the right of $l_{\vec{e_2}}$, and $\vec{e_2}$ is to the left of $l_{\vec{e_1}}$.

When back-propagating $\vec{f_1}$ and $\vec{f_2}$ in the way defined above, one of the following cases can occur:

- *$\vec{e_1}$ and $\vec{e_2}$ are hit simultaneously.* This happens in cases 1-3 from above. If $q \notin \vec{e_1}$ and $q \notin \vec{e_2}$ (case 3), all of their points are reached by the back-propagating chain (see figure 10 (3)). The protrusion-cut is then the segment between the end point of $\vec{e_1}$ and the start point of $\vec{e_2}$, as these are closest along $\mathcal{P}(x)$. In the other two cases, there is a part of the edge containing q not reached by the back-propagating chain, all points on the other edge being reached. The protrusion-cut in these cases will connect q with the closest point along $\mathcal{P}(x)$ on the edge not containing q (see figure 10 (1) and (2)).
- *$\vec{e_1}$ and $\vec{e_2}$ are not hit simultaneously.* This happens in cases 4 and 5 from above. Let $\vec{e_1}$ be the one that is hit first (case 4, see figure 10 (4)), all of its points being reached. We continue the back-propagation, inside P, until the target v of $\vec{e_2}$ is hit. The point on $\vec{e_1}$ closest to v along $\mathcal{P}(x)$, is then the end point u of $\vec{e_1}$ and uv is then the protrusion-cut segment.

A protrusion-cut segment often has both endpoints on the boundary of the P_i that contains the event generating it. It may be that, because of a split event that involves an edge and/or vertex of the protrusion-chain, the endpoints of the protrusion-cut segment lie on different parts. A reflex edge event inducing such a protrusion-cut is ignored in the decomposition process because its handling would not remove detail of P_i.

4. TIME COMPLEXITY

The polygonal decomposition we described in the previous section requires the computation of the straight line skeleton as a preprocessing step. Despite its similarity with the medial axis, which can be computed in linear time [14], the fastest known algorithms for the straight line skeleton are slower. The first sub-quadratic algorithm was proposed by Eppstein and Erickson [13]. It runs in $O(n^{1+\epsilon} + n^{8/11+\epsilon} r^{9/11+\epsilon})$ time with a similar space complexity, where r is the number of reflex vertices and ϵ is an arbitrarily small positive constant. A more recent algorithm by Cheng and Vigneron [15] computes the straight skeleton of a non-degenerate simple polygon in $O(n \log^2 n + r\sqrt{r} \log r)$ expected time. For a degenerate simple polygon, its expected time bound is $O(n \log^2 n + r^{17/11+\epsilon})$. The simpler and more practical algorithms given in [4] and [17] run in $O(n^2 \log n)$ and $O(nr + n \log n)$, respectively.

The basic approach of [13], [4] and [17] is to simulate the sequence of split and edge events that define the skeleton. With no additional computational complexity we can modify these algorithms in order to retain the information required by our decomposition process. So, for the rest of this section, we assume that, after the preprocessing, along with the straight line skeleton, we have a list of split events and a list of reflex edge annihilations, ordered according to their occurrence in time. Also, for each event we have pointers to the edges involved in the event.

In the first phase of our decomposition, split events are handled in chronological order of their occurrence, the outcome being a decomposition into non-overlapping parts. In the handling of a split event, a part in the current decomposition of P is divided by a chain (split-chain) of at most three segments. The first difficulty here is finding the endpoints of this split-chain, as defined in section 3.1, on the current part. For a vertex-edge collision, an endpoint of the split-chain is a vertex of P or lies on an edge involved in the event or on a previously computed split-chain. For a vertex-vertex collision involving one reflex and one convex vertex, one endpoint of the split-chain may also lie on an edge of P not involved in the event, so that this edge is not known. Finding out this edge requires searching in the current part boundary. The naive approach (checking the split-chain for intersection with the current part boundary) leads, over all split events, to $O(r_1(n-r_1))$ time in the worst case, where n is the number of vertices in P and $r_1 \leq r$ is the number of split events. This is a ray-shooting problem in a dynamic planar subdivision. A solution for this problem is found in the dynamic data structure introduced by Goodrich and Tamassia [16]. It uses linear space and supports ray-shooting queries and updates, like vertex and edge insertions and deletions, in $O(\log^2 N)$ time, where N is the current complexity of the subdivision. Point locations can also be performed in $O(\log^2 N)$ time, and this solves the second difficulty of this step, which is identifying the part that gets divided by the event. This is the part that has the event in its interior.

So, the first step of the decomposition takes time $O(r_1 \log^2 n)$, after the construction of the above mentioned ray shooting data structure for P in $O(n \log^2 n)$ time.

The handling of the reflex edge annihilations in the second step leads to successive simplifications of P_i, $i = 1, \ldots, r_1$, the parts resulted from splitting. When handling such an event x, the part P_j that has x in its interior is further simplified, if the protrusion-

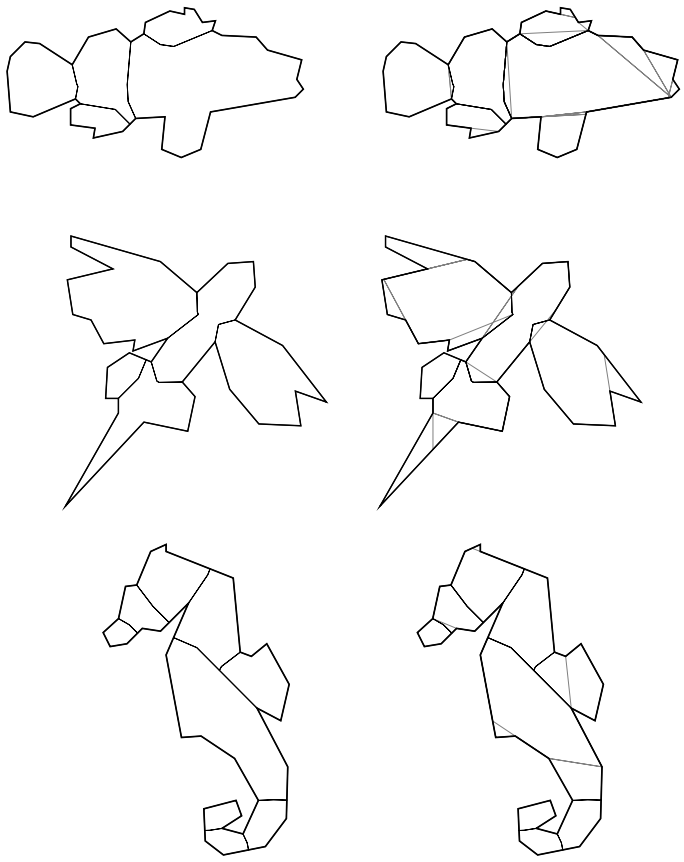

Figure 11: Examples of the proposed decomposition: the parts resulted after the global splitting (left) are further locally simplified (right), by means of protrusion-cut segments. The boundary of the shape is drawn in thick lines, the split-chains in thin solid lines and the protrusions cut segments in dotted lines.

cut segment has both its endpoints on P_j. Identifying the part P_j takes $O(\log^2 n)$ time using the above mentioned data structure for $\cup_{i=1}^{r_1} P_i$. The construction of the protrusion-cut segment takes constant time, but we have to check its intersection with the protrusion-chain. This is again a ray shooting problem that can be solved, over all reflex edge annihilations, in $r_2^2 \log^2 n$ time, where $r_2 = r - r_1$ is the number of such events in $S(P)$.

THEOREM 1. *For a simple polygon with n vertices, r_1 split events and r_2 reflex edge annihilations in its straight line skeleton construction, the proposed decomposition can be computed in $O((n + r_1 + r_2^2) \log^2 n)$ time, once the straight line skeleton is constructed.*

5. RESULTS

We have implemented the method described in section 3. We opted for a simple implementation. For the straight line skeleton computation, a simple, straightforward method [17] was preferred to the faster algorithm in [13]. Ray-shooting in dynamic subdivisions, as introduced by Goodrich and Tamassia [16], requires the maintenance of balanced decompositions of simple polygons via geodesic triangles. Instead, we used the arrangement package of CGAL, the Computational Geometry Algorithms Library [18].

Without strict evaluation criteria, it is difficult to judge how good a decomposition method works. Also, without having available software and test shapes used by others, it is difficult to compare one method with another. We show a few example results of our decomposition method, and a few results presented by others, in order to allow visual comparison.

The examples we present in figure 11 come from the SQUID database [19] which contains 1100 images of contours of marine animals and are part of the MPEG7 test set. After thinning these contours to one pixel width, a polygonal contour is extracted from the image. In this contour, each pixel corresponds to a vertex. To decrease the number of vertices, we applied the Douglas-Peucker [20] polygon approximation algorithm. Figure 11 illustrates the results of the proposed decomposition technique for three such contours. The output of the first stage (global splitting) in the decomposition is on the left. The parts in this non-overlapping decomposition are further locally simplified in the second stage. These successive local simplifications, are shown on the right.

In figure 12, a comparison with the recent method of Latecki and Lakämper [11] is provided. Their decomposition is based on a discrete evolution of the contour. Boundary points with a small relevance measure are iteratively removed from the contour. This induces a hierarchy of shape simplifications, that serves as a basis for the decomposition. Maximally convex arcs of the shapes in the hierarchy define parts on the original contour. These parts are simply cut by joining the endpoints of the maximal convex arcs. It is unclear from [11], however, how the selection of levels in the hierarchy is done. We can't associate each iteration in the contour simplification with a hierarchy level, even for the contours in figure 12, because this would lead to a very fragmented decomposition. The way the hierarchy levels are selected among the large number of iterations in the simplification step influences greatly the resulting decomposition. A second drawback of their method comes from the fact that the proposed contour evolution may lead to self-intersections. Finally, only boundary information is used for decomposition. The contour evolution often fails to capture region information. Parts like the kangaroo foot and the tails of the rabbit and the donkey cannot be properly partitioned in any hierarchy selection, since only contour convexity information is not sufficient here. The counter-intuitive partitioning of the rabbit's head, in their decomposition, has other causes. A different simplification hierarchy could lead here to a different partitioning. Our method is region-based and, as can be seen also from figure 11 and figure 12, the first stage of the decomposition is powerful enough to extract the main visual parts of the shape.

In figure 12, a comparison is also made with the method of Siddiqi and Kimia [8], which aims at identifying parts called necks and limbs. Most striking in their result is that the tail of the kangaroo is not separated from the body, and that the ear of the rabbit is cut into two.

The decomposition resulting from our method appear to be plausible for a variety of shapes. Note that the decomposition need not be anatomically correct in order to be useful for shape recognition, for example. The split of the elephant body in figure 12 is caused by the reflex vertex between the legs.

6. CONCLUDING REMARKS

We presented a novel type of decomposition for polygons. Decompositions based on various types of skeletons have been previously proposed in computer vision. This is the first one based on the straight line skeleton.

The skeletal nodes and the way they are generated are the basis for our method. The wavefront propagation events associated with the skeletal nodes indicate not only where the partitioning should

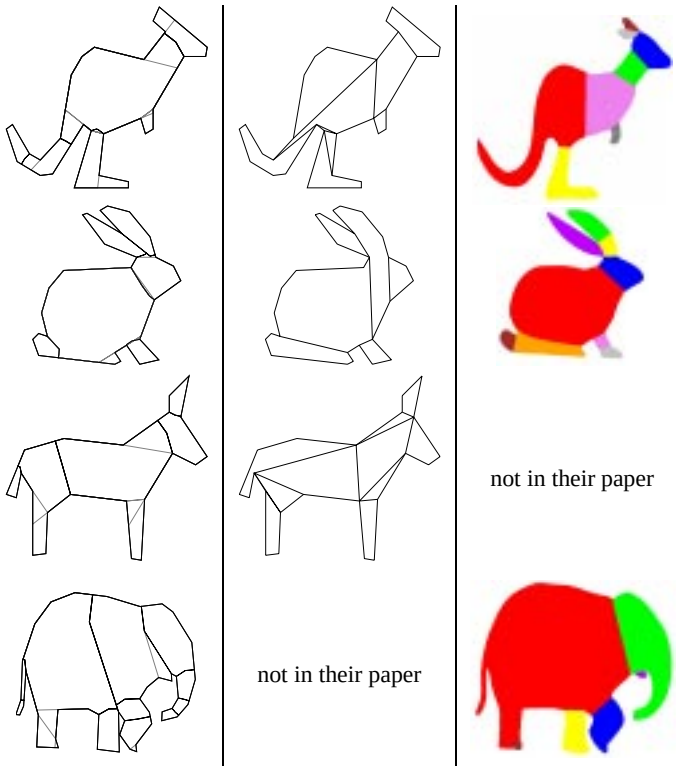

Figure 12: A comparison of our decomposition results (left) with those from Latecki and Lakämper [11] (middle), and Siddiqi and Kimia [8] (right).

be done, but also how it should be done (globally, locally, or not at all). The first step in our decomposition was motivated by the observation that split events are usually related with the most perceptually significant parts of the contour. This is consistent with the results obtained from the implementation.

Sharp reflex angles have a big impact on the form of the straight line skeleton, and a large effect on the resulting decomposition. If the incident edges are large, the decomposition may look natural. However, if the edges are small, the effect on the decomposition can be equally large, while the indent in the contour looks insignificant. Our decomposition is sensitive to this kind of noise.

Our method is invariant to rigid motions and uniform scalings. The decomposition can be computed in $O((n + r_1 + r_2^2) \log^2 n)$ time, after the straight line skeleton computation, where n is the number of vertices in the polygon, r_1 the number of split events, and r_2 the number of reflex edge annihilations in the wavefront propagation. We have presented results that show that it provides natural decompositions for a variety of shapes. This makes it attractive for partial matching in content-based retrieval. See figure 13 for an example, which is part of the Similarity-based Multimedia Retrieval Framework (SMURF), `http://give-lab.cs.uu.nl/Smurf/`. After decomposing all the shapes in the database, it is easy to search for particular shapes based on only a part of the shape, using as similarity measure the function difference of the turning angle functions [5].

Acknowledgements

We thank Petr Felkel for kindly providing the code for the straight line skeleton computation [17], and Geert-Jan Giezeman for programming the retrieval application. One of the anonymous reviewers made a useful remark about the quasi-normal histogram.

This research is supported by the Dutch Science Foundation (NWO) under grant 612.061.006.

7. REFERENCES

[1] Hoffman, D., Richards, W.: Parts of Recognition. Cognition **18** (1984) 65–96.

[2] Blum, H.: A Transformation for Extracting New Descriptors of Shape. Symposium Models for Speech and Visual Form. ed: W. Wathen-Dunn. MIT Press (1967) 362–381.

[3] Brady, M., Asada, H.: Smoothed Local Symmetries and their Implementation. The International Journal of Robotics Research **3(3)** (1984) 36–61.

[4] Aichholzer, O., Aurenhammer, F.: Straight Skeletons for General Polygonal Figures in the Plane. In: Proc. 2nd International Computing and Combinatorics Conference COCOON '96. Lecture Notes in Computer Science, Vol. 1090. Springer-Verlag (1996) 117–126.

[5] Arkin, E., Chew, P., Huttenlocher, D., Kedem, K., Mitchel, J.: An Efficiently Computable Metric for Comparing Polygonal Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence **13(3)** (1991) 209-215.

[6] Ogniewicz, R. L., Kubler, O.: Hierarchic Voronoi Skeletons. Pattern Recognition **28(3)** (1995) 343–359.

[7] Simmons, M., Séquin,C. H.: 2D Shape Decomposition and the Automatic Generation of Hierarchical Representations. International Journal of Shape Modeling **4** (1998) 63–78.

[8] Siddiqi, K., Kimia, B.: Parts of Visual Form: Computational Aspects. IEEE Transactions on Pattern Analysis and Machine Intelligence **17(3)** (1995) 239–251.

[9] Kimia, B.B., Tannenbaum, A.R., Zucker, S.W.: Shapes, Shocks, and Deformations I: The Components of Two-Dimensional Shape and the Reaction-Diffusion Space. International Journal of Computer Vision **15(3)** (1995) 189–224.

[10] Siddiqi, K., Shokoufandeh, A., Dickinson, S.J., Zucker, S.W. Shock Graphs and Shape Matching. International Journal of Computer Vision **35(1)** (1999) 13–32.

[11] Latecki, L.J., Lakämper, R.: Convexity Rule for Shape Decomposition Based on Discrete Contour Evolution. Computer Vision and Image Understanding **73(3)** (1999) 441–454.

[12] Keil, J.M.: Polygon Decomposition In: J.-R. Sack and J. Urrutia, editors, Handbook of Computational Geometry. Elsevier (1999) 491–518.

[13] Eppstein, D., Erickson, J.: Raising Roofs, Crashing Cycles, and playing Pool: Applications of a Data Structure for Finding Pairwise Interactions. Discrete and Computational Geometry **22(4)** (1999) 569–592.

[14] Chin, F., Snoeyink, J., Wang, C.-A.: Finding the Medial Axis of a Simple Polygon in Linear Time. Discrete Computational Geometry **21(3)** (1999) 405–420.

[15] Cheng, S.-W., Vigneron, A: Motorcycle Graphs and Straight Skeletons. In: Proc. 13th ACM-SIAM Symp. Discrete Algorithms (2002) 156–165.

[16] Goodrich, M.T., Tamassia, R.: Dynamic Ray Shooting and Shortest Paths in Planar Subdivisions via Balanced Geodesic Triangulations. Journal of Algorithms **23(1)** (1997) 51–73.

[17] Felkel, P., Obdržálek, Š.: Straight Skeleton Implementation. In: Proc. of Spring Conference on Computer Graphics, Budmerice, Slovakia (1998) 210–218.

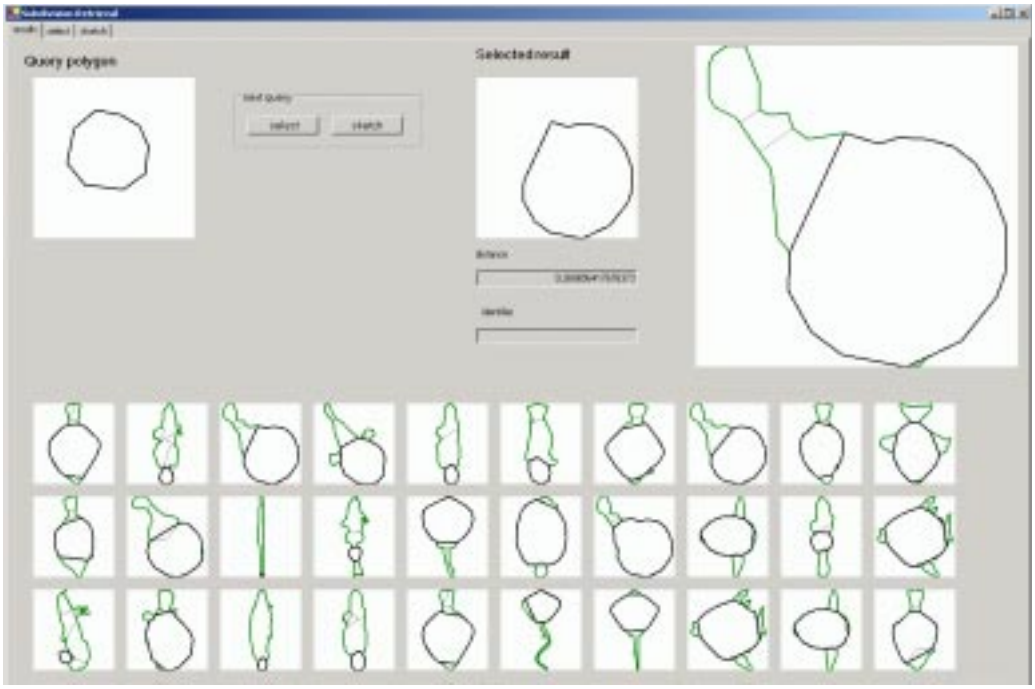

Figure 13: Part-based shape retrieval.

[18] The Computational Geometry Algorithms Library. `http://www.cgal.org/`.

[19] SQUID database. `http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html`.

[20] Douglas, D.H., Peucker, T.K.: Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. The Canadian Cartographer **10(2)** (1973) 112–122.