Structuring and layering contour drawings of organic shapes

Even Entem, Amal Dev Parakkat, Marie-Paule Cani, Loic Barthe

Structuring and Layering Contour Drawings of Organic Shapes

Even Entem
IRIT, Université de Toulouse, LJK, Université de Grenoble-Alpes, CNRS and Inria

Amal Dev Parakkat
AGCL, Indian Institute of Technology

Marie-Paule Cani
LIX, Ecole Polytechnique, CNRS

Loïc Barthe
IRIT, Université de Toulouse, CNRS

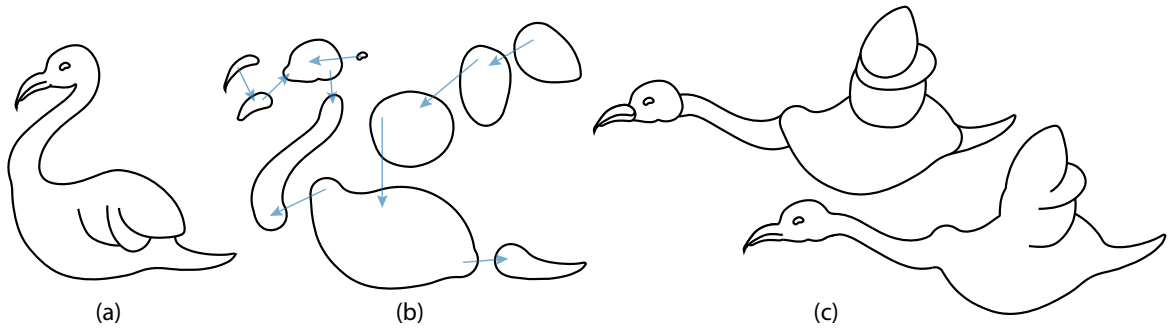

Figure 1: An illustration of the steps of our automatic structuring and layering system. (a) The input drawing. (b) Structure analysis and part-completion estimate the constituent parts and their layering. The arrows represent the partial depth ordering ("over" relation). (c) Manipulation of the depth ordering as desired (top), followed by the union of these elements composed with the original internal contours to produce a new drawing (bottom).

ABSTRACT

Complex vector drawings serve as convenient and expressive visual representations, but they remain difficult to edit or manipulate. For clean-line vector drawings of smooth organic shapes, we describe a method to automatically extract a layered structure for the drawn object from the current or nearby viewpoints. The layers correspond to salient regions of the drawing, which are often naturally associated to 'parts' of the underlying shape. We present a method that automatically extracts salient structure, organized as parts with relative depth orderings, from clean-line vector drawings of smooth organic shapes. Our method handles drawings that contain complex internal contours with T-junctions indicative of occlusions, as well as internal curves that may either be expressive strokes or substructures. To extract the structure, we introduce a new part-aware metric for complex 2D drawings, the *radial variation metric*, which is used to identify salient sub-parts. These sub-parts are then considered in a priority-ordered fashion, which enables us to identify and recursively process new shape parts while keeping track of their relative depth ordering. The output is represented in terms of scalable vector graphics layers, thereby enabling meaningful editing and manipulation. We evaluate the method on multiple input drawings and show that the structure we compute is convenient for subsequent posing and animation from nearby viewpoints.

Expressive '18, August 17–19, 2018, Victoria, BC, Canada

ACM ISBN 978-1-4503-5892-7/18/08…$15.00
https://doi.org/10.1145/3229147.3229155

CCS CONCEPTS

• **Computing methodologies** → **Image manipulation**; *Computational photography*;

KEYWORDS

Vector drawing analysis, Shape segmentation, Contour completion

ACM Reference Format:
Even Entem, Amal Dev Parakkat, Marie-Paule Cani, and Loïc Barthe. 2018. Structuring and Layering Contour Drawings of Organic Shapes. In *Expressive '18: The Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering, August 17–19, 2018, Victoria, BC, Canada.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3229147.3229155

1 INTRODUCTION

Contour drawings are commonly used for shape depiction. They are both easy to create and easy to interpret for a human and thus it makes them a convenient and expressive solution for visual communication. They are found in children's books, advertisements, technical books, and more. In contrast, these drawings are difficult for a computer to interpret. They usually depict silhouette curves

and internal contours as well as expressive strokes, and they may represent fully visible, self-occluding, or locally hidden regions.

Many contour drawings are directly authored in vector graphics applications or are easily converted to a compatible representation using vectorization tools. Automatically decomposing them into distinct and simple structural parts, layered in depth (as in Figure 1 (b)), permits users to edit and manipulate the drawings intuitively by rescaling, moving, rotating, copying, and pasting parts without the need for intricate manual modifications and corrections which would otherwise be required for such operations.

In this paper, we present an automated geometry-based method for extracting apparent structure and depth layers from clean contour line-drawings. We assume that the input drawing is intended to represent an organic shape, i.e., any free-form 3D solid with smooth connections between its 3D structural parts. The extracted depth-ordered structure is similar to the collection of blobs that artists sometimes use to temporarily define the construction lines and volumes of the shape they want to depict (see results of a web image search with the terms "tutorial drawing construction animals").We record additional information, namely, where these volumes blend together and where contours should be erased. This information can then be used for both current and nearby views editing to achieve new poses such as in Figure 1 (c). Although view-dependency may prevent the structure from being complete relative to the actual structure of the 3D depicted shape, we sill claim that it is a useful reference for editing the current drawing to create other postures of the shapes or near-by viewpoints.

The input drawing may be composed of silhouette curves as well as different categories of internal and external curves. They include internal open contours connected to silhouette curves, e.g., the contours of the feather groups in Figure 1 (a). Regions in the drawing that are demarcated by silhouette curves may also include a number of internal regions depicting sub-shapes, possibly lying on top of one another, such as the eye of the swan in Figure 1. Highly ambiguous curves, such as disconnected internal curves and connected external open curves, are considered in our work as decorative curves. We also detect and discard internal elements that fail to define their own silhouettes (see Section 3.2).

Our three contributions towards solving structuring and layering problems for drawings are as follows:

- We describe a simple and efficient method for the aesthetic closing of sub-parts contours. This method provides a consistent solution when open-end points are not explicitly defined in the input drawing (Section 4).
- We introduce the *radial variation metric* (RVM), a novel part-aware metric for complex 2D drawings, inspired by the *volumetric shape image* used for shape segmentation of 3D models [Liu et al. 2009]. Its variation along the medial axis of parts in a drawing enables the identification of salient connections between sub-parts (Section 5).
- We describe a recursive algorithm enabling the successive identification of sub-parts in a complex sketch and their assignment to depth-layers (Section 6). The key insight lies in processing the possible junction zones between the identified sub-parts in a specific order based on the types of contours involved. This enables us to handle cases of multiple connected internal contours forming a tree-like structure, as can be observed for the swan wing in Figure 1.

The structure and layering information we obtain can be used to represent and edit the input sketch in current or nearby views (Section 7).

We also demonstrate the automatic conversion of the sketch into a Vector Graphics Complex (VGC), a structure that eases the computational and manual editing of vector drawings, and that readily allows for manipulation, editing, and animation.

2 RELATED WORK

Recovering the structural parts of 3D objects in 2D vector contour drawings is a long standing and complex problem. This is due to the lack of information required for the unambiguous and automated shape understanding of most drawings. For instance, the understanding of the main features in a drawing is often based on contextualized interpretations.

Given that a drawing is only composed of lines, several approaches have been proposed to identify which visual mechanisms are used to help interpret the drawn lines in terms of self-consistent shapes and contours [Singh 2015].

Several methods address specific aspects of this complex process and techniques have been developed to evaluate them on well-defined data sets. Alternatively, more practical approaches aim at providing actionable interpretation methods by leveraging knowledge about particular object classes and relying on Gestalt principles.

One such Gestalt principle, the principle of closure – how our visual system tends to perceive the missing parts of curves or contours – has been used by algorithms to complete hidden and subjective contours [Nitzberg and Mumford 1990; Ullman 1976; Williams and Hanson 1996]. Many solutions rely on plausible, visually appealing curves such as Minimum Energy Curves (MEC) [Horn 1983] that maximize the curve smoothness, and Minimum Variation Curves (MVC) [Moreton 1992] that generate fair solutions. Our technique uses a variation of the latter with the aim of efficiently generating aesthetic curves.

Our approach is also inspired by the fact that the human visual system tends to segment a complex shape into simpler parts. Shape segmentation problems have been tackled for very long. Pioneering work relied on the branches of the skeletal representation of 2D shapes for inferring segmentation [Blum and Nagel 1978]. Observing that the quality of the correspondence between branches and depicted shape parts drops as the complexity of the object rises, subsequent work rather made use of maxima of negative curvature on the contour to identify part boundaries and focused on trying to disambiguate pairing between such boundary elements [Latecki and Lakamper 1999; Richards et al. 1987]. Shape segmentation then became a classical problem for both 2D and 3D shapes. We refer the reader to [Yang et al. 2008] and [Shamir 2008] for surveys. Fully accepted general solutions do not yet exist, and segmentation remains an area of active research even for the 2D case [Carlier et al. 2016; Larsson et al. 2015; Leonard et al. 2016]. In parallel, interactive sketch segmentation and/or completion methods based on human input were proposed to fill in drawings [Pessoa and Weerd

2003], to select layers for shape manipulation [Igarashi and Mitani 2010], to segment sketchy drawings [Noris et al. 2012] or to simplify them [Liu et al. 2015; Mi et al. 2009].

In this work, we build-on, or draw inspiration from, a number of the processing steps introduced for 2D [Giesen et al. 2009] and 3D [Liu et al. 2009] shape segmentation. However, our goal is to automatically segment complex sketches with not only contours but with internal silhouettes as well. Indeed, internal silhouettes are great perceptual conveyors of shape, as shown by their extensive use for expressive depictions of 3D models and high-reliefs [DeCarlo et al. 2003; Eisemann et al. 2009; Kunsberg and Zucker 2017]. Therefore, taking them into account is essential for being able to segment a larger variety of drawings.

This new goal brings us to prior work in the area of sketch-based modeling, where a number of methods relied on the analysis of complex sketches to infer a 3D shape or a 2.5D high relief from a single sketch, e.g., [Cordier and Seo 2007; Karpenko and Hughes 2006; Xu et al. 2014; Yeh et al. 2017]. Most methods in the area built on *a priori* knowledge (or contextual information) in order to resolve ambiguities - such as requiring exact symmetric shapes [Cordier et al. 2011], being restricted to garments [Turquin et al. 2007] or to side views of animals [Entem et al. 2014] or requesting 3D information such as a 3D skeleton from the user [Bessmeltsev et al. 2015]. Others methods obtained great results by relying on interactive user annotations for helping to infer high-reliefs from photos or drawings [Bui et al. 2015; Rivers et al. 2010; Sýkora et al. 2014; Yeh et al. 2017].

Closer to our goal, interpreting drawings of general organic shapes (ie. smooth 3D solids) depicting all visible silhouettes including cusps was tackled by Karpenko's pioneering work [Karpenko and Hughes 2006] in the context of 3D modeling from a sketch. To this end, drawn lines were represented as networks of *oriented* curves, enabling the authors to identify holes and to propose a reconstruction method for hidden silhouettes indicated by T-junctions. While we build on this work, we chose not to ask the user for the orientation of contours, thus interpreting a two-circle drawing of a torus as two superposed spheres. We focus instead on extending the handling of internal curves beyond cusps, enabling us to handle more complex suggestive contours as well as extra decorative elements.

In summary, we provide the first fully automatic method able to extract structural parts from cartoon drawings of organic shapes. We provide in Section 7 a detailed comparison between our results and those of previous work, by re-using a number of their examples.

Finally, we note that our method outputs a layered structure of sub-parts locally ordered in depth, so as to ease the subsequent editing of the drawing. Different structures and representations have been developed to handle partial depth orderings [Dalstein et al. 2014; Wiley 2006; Wiley and Williams 2006]. In this work we output results in the VGC format [Dalstein et al. 2014], and we choose to define a self-consistent global depth ordering of the shape parts. The extracted structure is view-dependent (there is no structure inferred for parts that are completely occluded, for instance), and the rule that treats internal silhouettes as distinct cases may lead to occasional surprises: the pupil of the cat's eye in Figure 2 is in fact a hole inside the eye, and thus its correct manipulation would require an intersection operator with the eye region to not protrude out of it.

3 OVERVIEW

Our method decomposes an input drawing into a set of 'structural parts', layered in depth, and possibly overlapping in 2D, together with rules for combining these. The decomposition is based on connected internal silhouettes and inner closed contours in the drawings, which provide explicit indicators of part layering; the resulting parts are intended to be meaningful for artists as shape-defining-volume silhouettes, and can be used for editing and posing in nearby views.

This section introduces the terminology we use throughout the paper, defines the assumptions we make on the input drawing, and presents the main features of our algorithm.

3.1 Terminology and assumptions

The input of our method is a vector line-drawing $\mathcal{D}$ defined in the (x, y) plane as a set C of parametric curves that may only intersect at their endpoints. The drawing may be either directly created in this form or obtained from a rasterized drawing using a vectorization algorithm [Noris et al. 2013] and then cutting curves at all intersections. Paramatric curves are also uniformely sampled as polylines for some of the further processing. In the following, we therefore refer to points along the curves C as *samples*.

As in Smoothsketch [Karpenko and Hughes 2006], our algorithm is designed to handle contour-drawings of smooth, closed shapes, which we refer to as *organic shapes*. However, to be able to handle a larger category of drawings, we also allow them to include specific categories of decorative curves such as those often used in cartoon drawing.

Therefore, our algorithm includes a mechanism for the automatic detection of contour curves within C. Since we are not asking for any additional information (such as contour orientations) from the user, we assume that the depicted shapes have no surface-to-surface contact, have genus 0, and are not self-overlapping (e.g., no animal's tail passing under and behind the body and forming a new background region).

We use the following terminology throughout the paper (see Figure 2):

Contour: A contour is a curve in C that corresponds to a silhouette of the depicted 3D shape, i.e. to points where the normal to the shape is orthogonal to the viewpoint. The contour graph is the planar graph structure formed by the contours and their intersections.

Region: A region $\mathcal{R}_\rangle$ is a 2D connected component of $\mathcal{D}$ delimited by a counterclockwise face cycle in the contour graph.

Part: Structural parts (or parts) are the 2D counterparts of the structural elements of the 3D shape represented by the drawing, as depicted in Figure 1(b). Our goal is to extract them.

Suggestive contour: A suggestive contour is an internal curve within a region $\mathcal{R}_\rangle$ of the drawing, connected with tangent continuity to its external contour (and thus forming a T-junction). Such curves are used in drawings to partially depict the visible contour of a structural part.

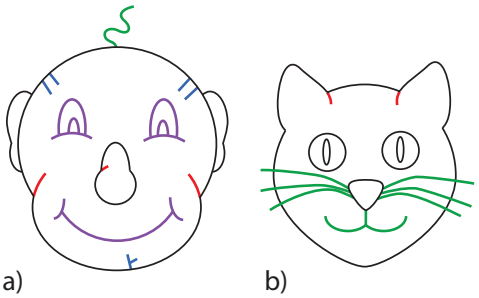

Figure 2: Stroke classification. Red curves are suggestive contours; green curves are hair; blue and purple curves are part of non-hair decorative elements. The remaining contours, in black, are part of the external silhouettes of shape parts.

Note that this definition is slightly different from the one found in the literature, since our term is in between the concepts of suggestive contours [DeCarlo et al. 2003] and of cusps [Karpenko and Hughes 2006], to better match what we observed in typical line drawings.

Decorative elements: These include all curves in $\mathcal{D}$ that are not visible contours of the depicted shape, but can instead represent ornamental details, 1D elements such as hair, or strokes used to represent bas-relief carvings. In our case, a curve is considered as a decorative curve if it falls in any of the following categories:

- Inner isolated subgraphs that do not contain external contours when processed as independent input drawings (in purple in Figure 2). Although they might contain or actually be inner contours, such subgraphs are highly ambiguous. We leave interpreting and processing them for future work.
- Inner trees of curves that are connected to the external contour of a region, but without tangent continuity (blue curves in Figure 2)
- Trees of curves located outside of the region they are connected to (green curves in Figure 2, which we call *hair*).

In our method, the decorative curves are identified and ignored from further contour processing, but are kept in the description of the corresponding to-be-segmented part.

3.2 Processing pipeline

Let $\mathcal{G}$ be the contour graph, the planar half-edge graph defined by the curves C that constitute the input drawing $\mathcal{D}$. As in standard planar graph processing, each half-edge corresponds to a given orientation of a curve. If half-edges are part of a closed contour, they are considered to lie respectively in the interior and in the exterior of the corresponding face cycle of the graph. In the remainder of this section we use both "edge" or "curve" to denote the edges of $\mathcal{G}$, depending on context.

The goal of our method is to process $\mathcal{G}$ in order to extract and progressively refine the set $\mathcal{P}$ of 2D structural parts of $\mathcal{D}$, as well as the associated partial depth ordering $PO_{\mathcal{P}}$ expressing their relative depths. More precisely, $\mathcal{P}$ is defined as: $\mathcal{P} = \{P_i = (e_i, \mathcal{S}_i, O_i)\}$, where e_i is the external silhouette contour (as a list of edges) of P_i and where $\mathcal{S}_i$ (respectively O_i) is the subgraph of $\mathcal{G}$ corresponding to the suggestive contours (respectively, the decorative elements) located within P_i, or attached to it.

Our processing pipeline, detailed below, is depicted in Figure 3. Starting with an initial set of silhouette-complete parts extracted at the initialization stage, our algorithm recursively decomposes each part into simpler sub-parts and updates PO accordingly. This is done until none of the parts in $\mathcal{P}$ has any suggestive contour left (for all i, $\mathcal{S}_i$ is empty).

Initialization:

We aim at initializing $\mathcal{P}$ to a first set of structural parts, such as the heart of the flower versus the part with all its petals in Figure 3 or the head versus the ears and the nose of the head in Figure 2 (a), and to set $PO_{\mathcal{P}}$ to the corresponding partial depth ordering. This involves completing the contours of partially hidden parts, such as the ears. Although this decomposition and figure completion problem was already solved in the past [Williams and Hanson 1996], this was for drawings only depicting silhouette outlines of occluded surfaces. To handle more complex cases of cartoon drawings with decorative lines such as the whiskers in Figure 2 (b), we extend the original method into a four-stage process:

Firstly, if $\mathcal{G}$ contains any vertex v of valence 4 or more (such as vertices where the whiskers of the cat cross the head's contour in Figure 2 (b)), we convert $\mathcal{G}$ into a non-planar graph by dissociating the curves at v, enabling us to further process the whiskers as decorative elements attached to the nose. This is done based on Gestalt's perceptual rule of continuity, as follows: For each vertex v of valence 4 or more with adjacent edges in $\mathcal{G}$ that correspond to pairs of tangent-continuous curves, we join each pair of curves at v into a single curve. This disconnects the pairs of curves from each other, enabling them to be attached to different structural parts. A constraint is also set to prevent more than one pair of merged curves from being interpreted as a contour curve in further processing, since this would violate our hypotheses of organic shape depiction (e.g., two overlapping circles do not correspond to any valid contour of organic shape, and are thus interpreted as a single 2D region crossed by a closed decorative curve). After this stage, any connected component of $\mathcal{G}$ that still contains a vertex of valence larger than 3 is considered in its whole as a decorative element, since a set of curves that connect without any tangent continuity cannot include silhouettes of smooth organic shapes.

Then, we process each remaining connected component CC_j of $\mathcal{G}$ to separate contour curves from curves corresponding to the associated suggestive silhouettes or decorative elements.

Since the input drawing is supposed to contain no self-overlapping part, this can be done by simply moving every edge whose half-edges both belong to the same face cycle (ie. lie in the same 2D region) from CC_j to another subgraph $\mathcal{A}_j$, which gathers candidate edges for either $\mathcal{S}_i$ or O_i. Note that this operation may split CC_j into smaller connected components, since the edges moved to $\mathcal{A}_j$ may include bridges between different subgraphs. In this case, CC_j and the corresponding $\mathcal{A}_j$ are split into smaller subgraphs. We decide to which sub-connected component the bridge sub-graph should be associated to by looking at its tangent continuity with the

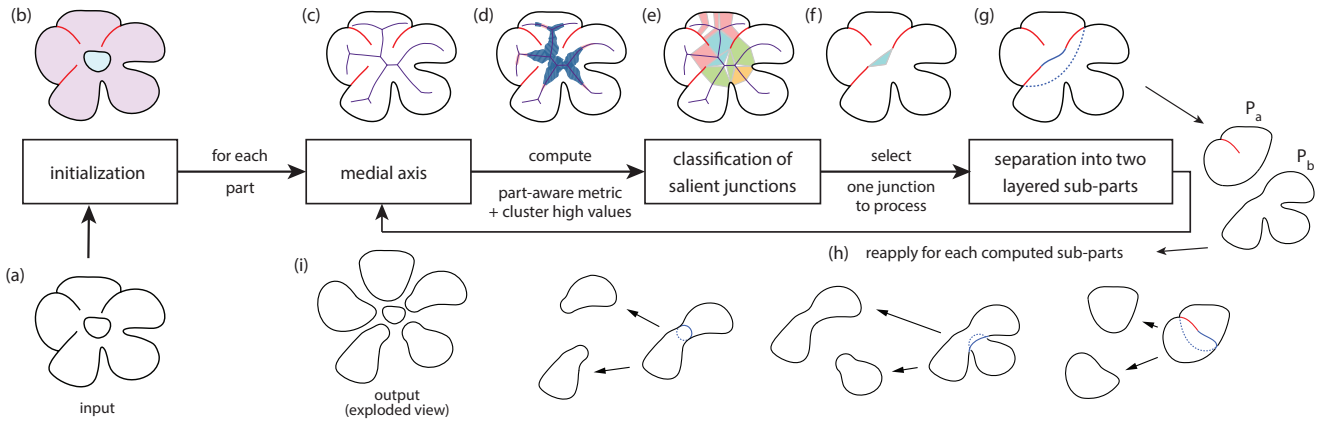

Figure 3: Processing pipeline: The input is (a) and the output is (h) with partial depth ordering (here depicted in exploded view).

neighboring curves: We insert the bridge into the $\mathcal{A}_k$ set associated with the subgraph CC_k of CC_j to which it has tangent continuity at one of its endpoints (e.g., a suggestive curve partially hidden by the contour of an inner part). If there is tangent continuity at both ends, the decision is taken at random. If there is none, the bridge is considered as a decoration and is attached to the subgraph corresponding to the contour of the region where it lies.

At this stage, each CC_j should only contain contour edges (for instance, the drawing in Figure 2 (a) is split into two connected components, namely (1) the nose and (2) the head plus ears, where only black curves remain). This enables us to use the existing algorithm in [Williams and Hanson 1996] to split then into structural parts, by using T-junctions to identify partially hidden parts (such as the ears) and smoothly complete their contours. Failure cases of this algorithm and other limitations are be discussed in Section 7.

In contrast with previous work, we use our own efficient solution, presented in Section 4, to compute closure curves. All resulting parts $\mathcal{P}_i$ are stored in $\mathcal{P}$ together with their contour edges e_i. The corresponding partial depth ordering information is added to $PO_{\mathcal{P}}$. We also add temporary depth relations with the remaining connected components, depending on the number and order of intersections with each other components to reach the background of the drawing with a line (parts with the same number of intersections being considered at the same depth level).

In a final stage, the set $\mathcal{S}_i$ of suggestive contours of each part $\mathcal{P}_i$ is extracted from the set $\mathcal{A}_j$ associated to its former connected component CC_j, by selecting curves with smooth T-junctions with contours e_i and that lie within $\mathcal{P}_i$. The other curves in $\mathcal{A}_j$ connected to e_i are classified as decorative elements and added to O_i. Lastly, every curve in other $\mathcal{A}_k$ sets that were initially crossing any of the current contour curves are stored in the set O_k of the part in which it is located since they no longer can be contours.

The set of parts $\mathcal{P}$ is now ready for further decomposition.

Recursive part decomposition:

The core of the algorithm is a recursive loop that processes each part in $\mathcal{P}$ and recursively decomposes it into sub-parts. This enables us to process complex suggestive contours indicating embedded sub-parts, such as the wing of the swan in Figure 1: The full wing is extracted first and is then recursively split into partially overlapping sub-parts. The recursive loop proceeds as follows:

For each part P in $\mathcal{P}$, we identify the salient potential junction zones between sub-parts, and iterate from best-to-worst until a valid pair (P_a, P_b) of sub-parts is identified. Missing contours are then inferred for P_a and P_b (see Figure 3 (g)) and the depth ordering relation between them is added to $\mathcal{PO}$. P_a and P_b are added to the list of parts $\mathcal{P}$, enabling us to recursively apply this process until no further decomposition is possible (Figure 3 (h)).

This recursive decomposition method raises three challenges, leading to our three key technical contributions. (1) The description of a robust method for completing the contours of the extracted sub-parts in a perceptually valid way. (2) The design of an effective metric for identifying the salient junction zones between the sub-parts of a 2D shape. In contrast with previous work, the 2D shapes we process may include suggestive contours, which are not constrained to come in even numbers, or to be short cusps. (3) The definition of an order in which to process the identified alternative solutions for segmentation into sub-parts. This order is important for extracting consistent sub-parts as it enables us to reuse the same algorithm in a recursive fashion.

Our aesthetic and efficient contour completion is presented in Section 4 and we discuss our new metric for identifying junctions in Section 5. Section 6 details our recursive structuring algorithm. It makes use of our new metric and completion method, with an emphasis on the priority order we set for processing possible junctions.

4 AESTHETIC AND EFFICIENT CONTOUR COMPLETION

In this section, we present the completion method we use for closing contours of both partially occluded structural parts (initialization

step) and of sub-parts extracted during recursive part decomposition. Although not proved to be the best possible perceptual completion method, our solution is simple, efficient, and produces adequate results in practice.

4.1 Scale-Invariant MVC

We first note that perceptually pleasing contour completion is a different problem from that of completing illusory contours, e.g., [Williams and Hanson 1996]. We seek to find aesthetic curves that are appropriate for the editing and subsequent animation of the drawing, during which hidden or omitted contour segments may become visible. In the literature, both curves minimizing the total curvature (i.e. "smooth" curves, MEC's) and curves minimizing the total variation of curvature (i.e., "fair" curves, MVC's) have been proposed as possible solutions to the problem. We choose to use MVC's because they tend to form more circular arcs; these are particularly well suited to later editing operations.

Let A and B be the two end-points of the open contour to be connected, along with corresponding unit tangent directions T_A and T_B as shown in Figure 4 (a). Our goal is to generate a perceptually plausible curve between A and B that best matches an inferred silhouette for the resulting part.

We define the curve connecting these two input points as follows. Let B_{AB} be a Bézier cubic curve connecting A and B, defined by the four control points (A, P_1, P_2, B), and whose tangents are aligned along the unit vectors T_A and T_B, i.e., $P_1 = A + c_1 T_A$, and $P_2 = B + c_2 T_B$. We optimize the free parameters c_1 and c_2 in order to minimize a "fair" curve functional that is a variation of the SIMVC energy originally introduced in [Moreton 1992] as:

$$E_{SIMVC-Moreton} = \left(\int \mathrm{d}s\right)^3 \int \left(\frac{\mathrm{d}K(s)}{\mathrm{d}s}\right)^2 \mathrm{d}s \qquad (1)$$

where $\left(\int \mathrm{d}s\right)^3$ is the product of a regularization term $\left(\left(\int \mathrm{d}s\right)/\|B-A\|\right)^3$ and scale-invariance term $\|B-A\|^3$. This regularization term relies on the cube of the scale-relative arc length of the curve. We increase the regularization term's exponent from 3 to 5 in order to reward slightly shorter curves that avoid cases where closure curves would slightly jut out from the desired boundaries and thus propose to minimize the following energy:

$$E_{SIMVC} = \frac{\left(\int \mathrm{d}s\right)^5}{\|B-A\|^2} \int \left(\frac{\mathrm{d}K(s)}{\mathrm{d}s}\right)^2 \mathrm{d}s \qquad (2)$$

The use of Bézier curves guarantees that the curve lies in the convex hull of its control points and is therefore well suited to interactive sampling and intersection queries. The choice to optimize its parameters with the SIMVC functional produces a scale-invariant result.

In practice, we use the standard Gauss-Kronrod quadrature to numerically integrate the different integral terms. Powell's method is used for minimizing E_{SIMVC}.

4.2 Efficient implementation

Given that the completion curves we compute are invariant to scaling, translation and rotation, there remain only two configuration parameters, namely θ and φ. They are the oriented angles formed by the two tangents with respect to a line between the two points to be connected, as shown in Figure 4 (a).

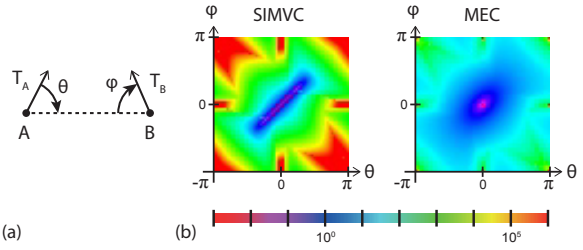

Figure 4: To optimize the computation of our SIMVC curves, we precompute a table of the energies and associated parameters that the curve can take as a function of the two angles θ and φ defined in (a). We illustrate the sampled function SIMCV energy values and compare them with the MEC energy that minimized total curvature.

The SIMVC energy of our curve defined as a function of these two parameters is continuous and smooth in the sub-space of non self-intersecting curves, as shown in Figure 4 (b). This enables us to precompute a table of the different SIMVC Bézier cubic curves, with their SIMVC energy and parameters c_1 and c_2, as a function of θ and φ. Parameters can then be interpolated, with either bilinear or bicubic interpolation, to provide both an approximated curve and good initial parameters for the final curve computation, leading to an important speedup of the gradient descent with two variables.

In practice, Bezier curves enable us to quickly detect invalid contours: indeed, closures for partially occluded or partially occluding parts should not protrude outside of the union of the related parts thus we look for intersections. To avoid unintended intersections we rotate tangents, at the points to be connected, inwards and by a small angle, keeping the alignment of tangents almost imperceptible. During our experiments, we ended with a value of 2 degrees.

5 EXTRACTION OF SALIENT JUNCTIONS WITHIN A PART

While previous work already addressed the segmentation of 2D shapes into perceptually salient sub-parts, these methods do not tackle the segmentation of shapes carrying extra structural information in the form of suggestive contours. This is the problem we are addressing here.

5.1 Salient Junctions

In the remainder of this paper, we define a **salient junction** as a region where the drawing of a part exhibits a perceptual change, enabling it to be divided into two sub-parts in a perceptually consistent way.

We define **junction boundaries** as being the portions of the part contours that delimit such a junction. These portions can either be a point (e.g., the open end of a suggestive contour) or a contour segment, as will be the case when the exact point where the segmentation should occur is unclear.

Among the common approaches to 2D shape segmentation that we review in Section 2, computing junction boundaries as the segments of the contour of maximal negative curvature cannot capture approximate regions such as those indicated by the green and red junction boundary in Figure 5 (a), since the red one is not a curvature maximum and thus would not be detected. Similarly, segmentation methods that directly make use of the branches of a skeletal representation such as the Medial-Axis Transform to identify sub-parts, also fail in a number of cases, as shown in Figure 5 (b,c). Fortunately, such correspondence between a Medial-Axis' branch and a structural part remains generally true for foreground structural parts delimited by suggestive contours, such as the middle sub-part in Figure 5 (b). The solution we describe below therefore builds on skeletal representations, while being based on a new metric.

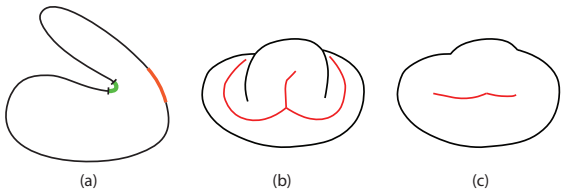

Figure 5: (a) A maximum of negative curvature (in green) defines a junction boundary on the contour but its visual counter-part boundary (in red) shows no such maximum. (b) S-skeleton of a part (in red), i.e. Medial-Axis considering internal silhouettes: the lower structural sub-part is not captured, while the middle sub-part at the top is. (c) Processed Medial-Axis of the external contour of (b), where no branch is associated with the top structural sub-part.

5.2 Radial Variation Metric

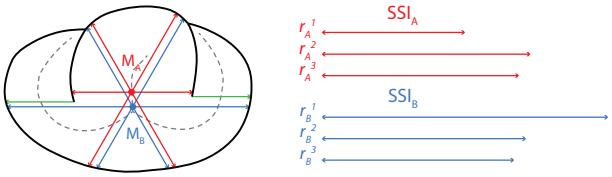

Figure 6: Example of SSI for points M_A (resp. M_B) constructed by measuring the local reaches r_A^k (resp. r_B^k) in the k'th direction of a uniformly sampled set of m directions (here $m = 3$). Some discontinuities of reach are present and shown here as green arrows.

Our new parts-aware metric is inspired by a similar metric [Liu et al. 2009] developed for 3D shape segmentation. In particular, it makes use of a *Surfacic Shape Image*, a modified 2D version of the Volumetric Shape Image (VSI) introduced in [Liu et al. 2009]. However, we tailor this more specifically to our problem, as detailed below.

We define the *Surfacic Shape Image* (SSI) as a signature of silhouette visibility from a given point of view inside a part of the input drawing (see Figure 6). A signature SSI_i is defined as a set of m distances l_i^k between closest points $p0_i^k$ and $p1_i^k$ to the point i in the k^{th} direction of a pre-defined uniformly sampled set of m directions. From 60 to 100 directions per set are used for the different examples in this paper. Both external contours and internal contours are considered at this step, while decorative curves are discarded.

Similar to the 3D case, the local change of SSI (differential of SSI) between two neighboring points can be used to detect junctions between two structural shape parts. We propose to compute this differential between points A and B as follows:

$$\Delta(SSI)_{A,B} = \frac{1}{\sum_k w_{k,A,B}} \sum_{k=1}^{m} w_{k,A,B} \frac{\left| l_A^k - l_B^k \right|}{\|B - A\|} \tag{3}$$

where

$$l_i^k = \left\| p1_i^k - p0_i^k \right\| \tag{4}$$

Discontinuities near open ends of suggestive contours as depicted in Figure 6 generate outliers. To tackle this problem, we fit a Gaussian to the distribution of such values (as in [Liu et al. 2009]) using the weights w_k defined as follows:

$$w_{k,A,B} = \begin{cases} e^{-(d_{k,A,B}-u)^2/(2\sigma^2)}, & \text{if } d_{k,A,B} < u + 2\sigma \\ 0, & \text{if } d_{k,A,B} >= u + 2\sigma \end{cases} \tag{5}$$

where $d_{k,A,B} = \left| l_A^k - l_B^k \right| / \|B - A\|$, u is the mean of the $d_{k,A,B}$'s, and σ the standard deviation.

Note that in contrast to [Liu et al. 2009], $\left| l_A^k - l_B^k \right|$ is not squared in Equation (3), in order to make the equation more linear. In practice, we regularize this term by the distance between neighboring points of view, in order to allow for similar results whatever the sampling resolution at which this measure is used. Thus $\Delta(SSI)$ tends to be scale-invariant for high resolutions when there are no discontinuities of visibility. Salient discontinuities, even pruned, should still contribute to the $\Delta(SSI)$ in salient junctions. However, we note that their contribution is related to the number of rays comprised within the parallax angle of a discontinuity location seen from neighbor points. Thus we propose two sampling methods that correctly balance this contribution. The first uses a uniform spacing that is relative to the drawing size (in practice $0.005 * D_{height}$) and the second uses a dynamic spacing relative to local Medial-Axis disk radii (in practice $0.02 * local\_radius$). While the first is closer to perceptual principles, the second allows for small details to be processed. Our presented results have been generated with the first method for better readability of figures. We emphasize that even if it extends to the continuous case when there are no discontinuities, the resolution of *SSI* sample points should not be higher than that of the input polylines so as not to reflect the lack of curvature of the polylines (otherwise a wave like pattern can emerge).

Let us now define a part-aware metric from the SSI. Note that we cannot reuse the method introduced by Liu et al. [Liu et al. 2009], where the distance used to segment a 3D mesh was defined as the integral of the VSI distance along a geodesic path between vertices. Indeed, in our case, there is no 3D surface to support and define the actual shortest path between two facing vertices on opposite sides of a shape. Therefore, our approach identifies salient sub-parts

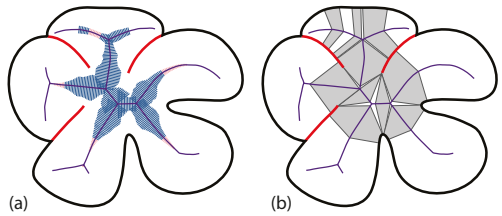

Figure 7: (a) Computed d_{SSI} between vertices of each edge of the S-skeleton of a part, and represented for each skeleton edge as the scale of an orthogonal segment passing through the center of this edge (values are squared for visibility). (b) Salient junctions (in grey).

by defining a 2D Part-Aware metric *along the curves of a specific skeleton*, called the **salient skeleton (S-skeleton)**, as described below.

We initialize the S-skeleton of a shape part as the medial axis of the region bounded by the external contour and the suggestive contours of this part, as shown in Figure 5 (b). Decorative curves are discarded. As usual, it is defined as the locus of the disks of a Medial-Axis Transform (abbreviated *MAT*) which are the maximal disks that do not intersect the set of contours (see Figure 3). For the S-skeleton to only reflect the main shape feature, we proceed in a fashion similar to the Scale-Axis Transform [Giesen et al. 2009]. The goal is to locally remove small disks by considering those that can be covered by others in a version of the *MAT* with larger radii. The final result can be realized by computing the *MAT* of the grown shape defined by the union of scaled up disks from the initial *MAT*, and then scaling down the radii. However for reasons of computational efficiency and simplicity, we choose to iteratively remove disks from branch extremities in the grown *MAT* that are covered by others and then scale down the radii of the remaining disks. This saves the computation of a union and a *MAT* for similar results at the scaling factor used in our algorithm, which is 1.3. Additionally, the associated contour points of these removed disks are given to the nearest remaining neighbor in order to keep a mapping between contours and the S-skeleton.

Given that the drawing represents the silhouettes of a volumetric, organic shape, the S-skeleton is a good candidate for extracting structural information about salient sub-parts. It enables us to recover junction boundaries that either lie along the external contour of the part, or at the open end of a suggestive contour. To correctly identify these boundaries, all the curves in the drawing, as well as the S-skeleton itself, are represented using half-edges. This enables us to assign the duplicate vertices on two sides of a suggestive contour to different branches of the S-skeleton, as illustrated in Figure 7.

Our new 2D part-aware metric, called **radial variation metric** and noted d_{SSI} is then defined over the S-skeleton as the integral of the SSI differential (Equation (3)) along the shortest path joining two skeleton vertices:

$$d_{SSI}(M_A, M_B) = \sum_{e \in E} \Delta(SSI)_e \tag{6}$$

where M_A to M_B are vertices of the S-skeleton and E the set of edges of the shortest path between them.

5.3 Salience junction detection

With the S-skeleton S being computed from a medial axis transform, each edge $e \in S$ is associated with two facing portions of the contours, i.e. the segments or point of the contours that could be generated by drawing discs from this specific edge of the skeleton. This enables us to use the S-skeleton to define both salient junctions (the regions we are looking for) and the junction boundaries that delimit them on the contours:

We initialize salient junctions as the 2D regions corresponding to segments of the S-skeleton with d_{SSI} values over a threshold k (see Figure 7 (b)). Thanks to the scale-independent nature of the metric, a single threshold value k is used regardless of the scale of the input drawing (we use $k = 0.45$ for all our results). These segments are stored using lists of edges of the S-skeleton. Since sharp extremities of structural sub-parts may correspond to large-but-irrelevant d_{SSI} values, we remove them in a second pass: Starting from S-skeleton extremities, we iteratively remove edges while their d_{SSI} values decrease. Increases in d_{SSI} values due to noise can lead to unwanted decomposition of pointy ends, but are mostly avoided by smoothing the d_{SSI} values along the S-skeleton first. Due to the nature of the SSI, sampling points near the middle of a punctually symmetrical transition will yield low d_{SSI} values compared to other transitions as illustrated in Figure 8. However, we note that the negative curvature cues on both sides is implicitly given by the diamond shape formed between the junction zones (Figure 8 (a)). We merge these salient junctions if the distances d_{C0} and d_{C1} between the junctions zones along contours are both: inferior to the distance d_M along the S-skeleton, and inferior to the half of the average radius of the Medial-Axis disks of the junctions zones.

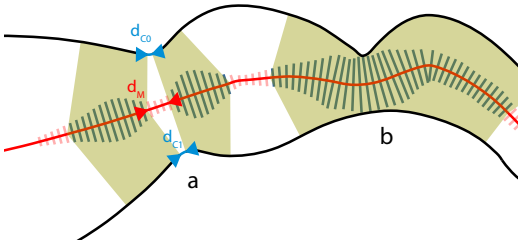

Figure 8: (a) A salient junction misidentified as two junctions due to punctual symmetricity of the shape at the center of the transition (in near radial directions to the Medial-Axis). Both are merged before further processing. (b) A more common case of salient junction with no merging necessary.

6 RECURSIVE PART DECOMPOSITION

Given the general methods that we have described for closing contours and for extracting salient junctions within a structural part, we now detail how a given part is decomposed, i.e., how its

salient junctions are prioritized, and how the corresponding sub-parts are extracted and completed.

6.1 Prioritizing salient junctions

Decomposing a part not only requires extracting possible junction regions between sub-parts, but also assigning them a prioritized order. We achieve this via a classification of salient junctions, depending on the type of contour segments that contributed to this specific part of the S-skeleton.

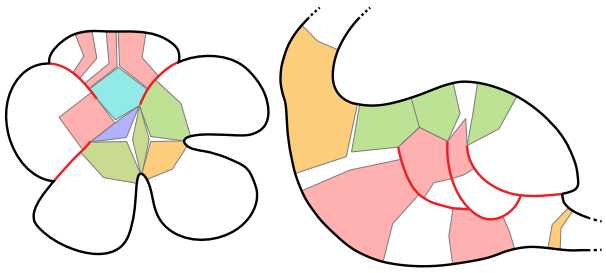

Figure 9: Salient junctions: (SF, SF) in dark blue, (SB, SB) in cyan, (C, SF) in green, (C, C) in orange, discarded regions in pink.

Since they are defined by segments of the S-skeleton, each salient junction comes together with a pair of junction boundaries (the associated parts of the contours, possibly reduced to a point) found on each side of the skeleton. Salient junctions are classified as follows, based on the nature of this pair of junction boundaries (see Figure 9):

(1) **Two segments of suggestive contours, that do not belong to the same tree of internal silhouettes:** The junction is either classified (SF, SF) and (SB, SB), depending on whether the suggestive contour's curves (a set of half edges) correspond to a front (occluding) or to a back (occluded) sub-part of the shape. The occluding side is given by the T-junction properties.
(2) **A pair formed by an external contour segment and a suggestive contour segment:** We only consider the case when the suggestive contour side corresponds to the front of the shape, denoted as (C, SF).
(3) **Two portions of the external contour**: the junction is classified (C, C).

If a curve segment in a pair spans different types of contours, the associated salient junction is subdivided. Regions that do not fit into the categories described above (in pink in Figure 9) are discarded, since they have a bounding contour on one side and an occluding one on the other, and this case is not handled by our decomposition.

This classification is used to select the salient sub-parts to be extracted at each stage of the recursive part decomposition algorithm described in Section 3.2: (SF, SF), (SB, SB), (C, SF) and (C, C) salient junctions are respectively given highest to lowest priority. This enables us to give priority to sub-parts that are unambiguously in front of their neighbors, such as for the bottom-right part of the flower in Figure 3, before processing partially occluded sub-parts and those with weaker depth clues.

6.2 Processing complex suggestive contours

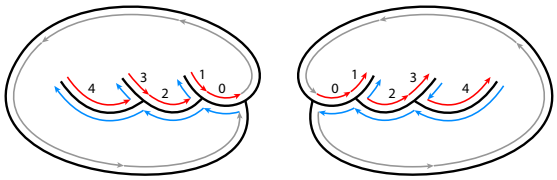

Figure 10: Adding relative depth information along a suggestive contour, illustrated here for the case of a complex curve forming a tree. As shown here, the numbering must be made in forward or backward fashion depending on the root T-junction.

In addition to the sorting order we just defined, sub-parts defined by (C, SF) junctions need to be given a priority order. This enables us to handle cases where the suggestive contour forms a tree of branching curves, such as the swan's wing in Figure 1.

Let us look at the similar shapes on Figure 10: sub-shapes with high label values on the edges need to be extracted first, since they embed the other ones. This enables us to extract the full wing, which can then be progressively decomposed into three consistent sub-parts.

Given that suggestive contours represent internal silhouettes of volumetric sub-parts of a shape that smoothly blend with the parent part, they should have a G^1 continuous junction to the silhouette they are attached to (see Figure 10). The side of this smooth junction indicates which part comes above the rest. Therefore:

(1) If the connection point with the external contour is only C^0, the curve tree is re-classified as decoration.
(2) If G^1 continuity is detected, we use a traversal of the suggestive contour from the connection point to the open end, on the side of the G^1 continuous curve in order to enumerate and prioritize these half edges for decomposition.
(3) During the traversal of the tree, only suggestive contours that demarcate a sub-part that is on the same side as the occluder at the root T-junction are allowed. Other contours are left-out as decorative strokes and are not processed by our algorithm.

Note that even in the case of complex suggestive contours that form a tree as in Figure 10, the suggested sub-shape is always on the same side of the curve, given that the organic shape hypothesis would otherwise be violated.

6.3 Part decomposition method

Decomposing a shape part at a given salient junction always involves generating two contours for closing the two resulting sub-parts. We use the terms *front closure* and *back closure* to refer to the closure curve that closes the sub-parts lying at the front and back, respectively, given the depth clues provided by the suggestive contours. To decompose a part, we first generate the two closure curves for all the junctions having the highest priority, using specific algorithms for each type of salient junction, as detailed below. If either

of the resulting closure curves intersect the shape contour or if the front closure curve intersects a decoration curve, the current junction is discarded. Finally, the junctions that result in most plausible closure, as defined by the minimal sum of their closure curves' energy, are selected for the decomposition. The resulting sub-parts are created and included in the partial depth set $\mathcal{P}$ according to their classification as a front or back closure curve.

Note that the two newly created parts may have overlaps between their respective closure curves, but this is valid since they are each assigned a different depth layer. To assign such depth in the case of (C, C) salient junction without a relative depth cue, e.g., the bottom-right petal of the flower in Figure 3, we use the convention that the largest shape part should be in front, which is often the best choice when the resulting sub-shapes are to be animated.

While inferring the closure of a sub-part given two end-points and the associated tangent vectors is easy (Section 4), and can be done for connecting two suggestive contours (SF, SF) and (SB, SB) salient junctions, the connections in the (C, SF) and (C, C) cases are much more challenging. Indeed, the best pairs of contour points in the salient junction zone should be computed for the front and back closure curves. Our methods for solving these two cases are presented next.

6.4 Contour / Suggestive contour (C,SF) closure

Given that we are in the case where the suggested sub-part is on top, we compute all the possible closure curves that join the tip of the suggestive contour to the sample points on the facing contour segment in order to generate the front closure (Figure 11 (b)). We also generate all the possible closure curves joining the contour segment with the T-junction at the base of the suggestive contour tree (Figure 11 (c)). Keeping only pairs of closure curves whose tips on the contour are not further from each other than the blending radius, we select the most plausible pair of closure curves of minimal energy using the sum of their SIMVC energies (Equation 2). This enables us to efficiently select the best pair of closure curves among the n^2 possible choices.

For this task we must define an adapted sampling to explore the space of possible closure curves. We first compute a blending radius that reflects the size of the transition between the salient sub-part of interest and the remaining part of the current shape. This radius is set as the average of radii at the two ends of the salient junction along the S-skeleton. We then identify *corner segments* within the junction boundary, defined as parts of the curve where curvatures are larger than the blending radius (orange curve segments in Figure 11 (a)). Corners are associated with specific virtual sampling points at two different locations, namely at the beginning and end of the curve-segment that forms the corner, and with two possible tangents depending on which of the two closure curves is being computed (Figure 11 (b) and (c)). Other contour parts are regularly sampled with a distance between consecutive samples equals to the half of the blending radius. For each such sample point we store the incoming (or outgoing) tangent vectors, with a small tilt outwards (or inwards) in order to avoid undesired intersections.

When dealing with suggestive contours that could be considered as big cusps (Figure 12 (a)), the front closure is processed normally (Figure 12 (b)) while the the back closure requires to extend the relevant transition contour (Figure 12 (c)). The transition contour is extended by following its associated S-skeleton branch until either an extra branch is encountered, or the facing contour is a neighbor of the T-junction. This produces the result exposed in Figure 12 (d).

6.5 Contour / Contour (C,C) closure

We sample the boundaries of the salient junction in the same way as for the (C, SF) case. The two closure curves' extremities may be located anywhere on these segments. With n sample points on both contours, a naive method would lead to n^2 possible curves to generate for each of the sub-parts, and thus to n^4 pairs of closure curves to evaluate. To reduce the complexity back to n^2, we only consider the pairs of closure curves between a given pair of points, i.e., the same sample for the two curves.

We use a variation on the energy in this case for selecting the most plausible pair of curves. Rather than selecting short curves, we wish to favor a decomposition close to the middle of the junction zone. Therefore, we use the energy $\hat{E}$ of the two curves to select the best closing pair, defined as:

$$\hat{E} = \left(1 + \frac{(1 - \frac{a_0}{0.5 * l_0})^2 l_0 + (1 - \frac{a_1}{0.5 * l_1})^2 l_1}{l_0 + l_1}\right)(E^0_{SIMVC} + E^1_{SIMVC})$$

with $l_i = a_i + b_i$ and a_i, b_i the junction boundary segments arc lengths shown in Figure 13, E^0_{SIMVC} and E^1_{SIMVC} the energies of the closure curves.

We also wish to reward the use of corners over a solution with two circular arcs forming a circle since it has an energy close to zero. Thus valid closures that use a sample at a corner as an implicit end point are given priority.

7 RESULTS AND DISCUSSION

Figures 1, 3, 20 to 17, and 21 show a variety of shape decomposition and layering results that are automatically computed by our method. Note that for Figure 20, we reused drawings from a recent paper [Bessmeltsev et al. 2015], showing that our method achieves the structuring and layering of such drawings without the need of any extra information, whereas a user-defined 3D skeleton was used in the original paper. Figures 1 and 14 (top-right) show even more challenging cases where some of the suggestive contours form a chain of T-junctions, requiring the labeling method of Section 6.2 in order to be properly processed. Figure 21 shows results computed on cat drawings found on the web using a simple query and vectorized using Adobe Illustrator.

Validation: Segmentation in depth ordered structural parts is a fundamental first step for further applications such as the edition of vector drawings with robust completion of partially hidden parts, 2D animation, or sketch-based 3D modeling. While developing such applications remains out of the scope of this research, we tested our method with two applications in mind: the conversion of the input drawing into a Vector Graphics Complex [Dalstein et al. 2014] that then enables easy and meaningful editing, and 2D vector animation. Posing and animation results are shown respectively in Figure 1 and in the supplemental video. The decomposition of the wing of the swan may not fit the perceived structure for every viewer, but wings are not known to be easily animatable in 2D.

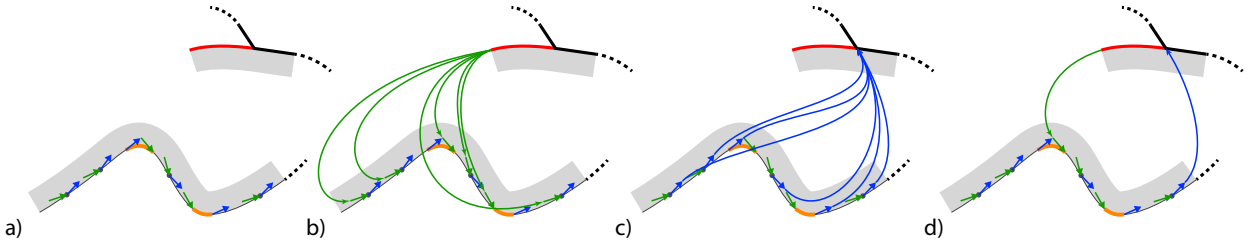

Figure 11: Part decomposition at a salient junction. a) junction boundaries in this case are described by an external contour at the bottom and a red suggestive contour at the top. The external contour is uniformly sampled into a set of points and their associated pair of tangents. Corner segments (orange) samples are given two points instead of one to handle proper connection where curvature is the highest. b) all the closure curves corresponding to pairs of one sample point from the contour and the suggestive contour extremity are computed and their plausibility is evaluated for closing the rightmost sub-part. Curves that intersect contours are eliminated. c) the same closing procedure is applied to the other region, therefore the suggestive contour's T-junction is used instead of its extremity; d) resulting closure curves.

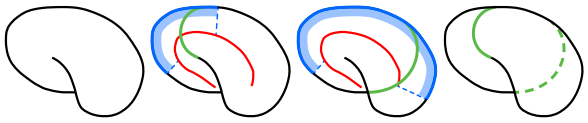

Figure 12: (a) Example of a suggestive contour comparable to a cusp requiring a (C,SF) closure. (b) The front closure unknown end-point is searched along the blue contour part defined by the identified salient junction. (c) The transition is extended for the back closure. (d) Result of the decomposition with sub-parts sharing a wide part of contour.

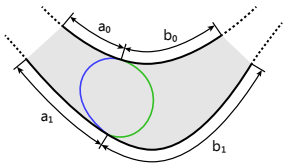

Figure 13: For (C, C) closures we define a new coefficient for $\bar{E}$ based on the samples' positions relative to their respective sampling contours as defined in Section 6.5.

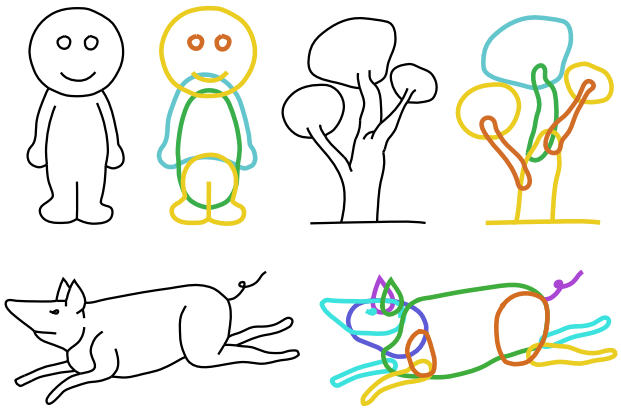

Figure 14: Results on drawings of a cartoonish man, a tree and a pig. The two legs of the man are seen from a special view, thus the surface contact is classified as a decorative element. Warmer colors are in the foreground.

Discussion: Our method for structuring complex drawings performs as expected in most cases. We point out that once a drawing is processed, the union of the extracted parts does not necessarily exactly correspond to the initial outline since the blending between parts is not conserved. However, it is very similar and it would be possible to retrieve a similar outline by representing the parts' contours as iso-contours of 2D scalar fields and blending them together. Scalar fields with skeleton could also allow for easy directional rescaling of structural parts to allow for illusory 3D rotations. For instance it could help for improving the animation of our swan's wings.

In some cases, a valid drawing may be ambiguous and gives rise to several different interpretations. This is the case for the example in Figure 15, where the shape could either be interpreted as a boxing glove (b) or as a snail head protruding out of the shell (c). Our method will output a single result in such cases, the one in (c), because of the way we process complex suggestive contours. Some curves that would be processed naturally by a human as contours are not processed when there is no T-junction such as for the beak of the bird in Figure 1.

Lastly, similarly to the metric in [Liu et al. 2009], our d_{SSI} metric could also be used to define a distance between two points of the contour, which we would define as the d_{SSI} distance between their two corresponding S-skeleton's vertices. In future work we wish to explore the possible applications of this new metric to contour drawings.

Comparison with previous work: The closest work we can compare with is SmoothSketch [Karpenko and Hughes 2006]. While they look for a plausible completion of the hidden contour hinted by cusps, we instead use the facing contour to smoothly wrap a

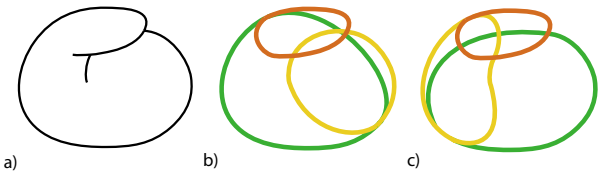

Figure 15: An ambiguity of interpretation. Our algorithm can only produce the result (c) while (b) would also be a perceptually plausible result.

curve around the hypothetical 3D shape as shown in Figure 16. We emphasize that the segment of our hidden closure that is near the T-junction is a plausible hidden contour. Thus, our method could benefit figural completion of cusps in SmoothSketch's failure cases. However, we think that both this algorithm and ours should be used in a complete application since hidden contour completion is more meaningful than our decomposition for the case of cusps that are small relative to the local thickness of the shape, and specific cases such as the swan's wing.

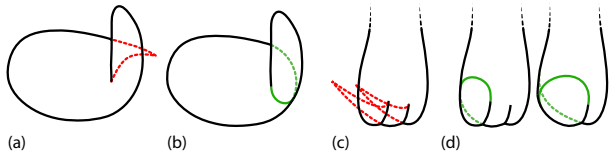

Figure 16: Comparison of our results (b,d) with SmoothSketch's failure cases (a,c).

The recursivity of our decomposition hides some perceptual information such as similarity, grouping, symmetry. In the paw example in Figure 16(c,d), we perceive the similarity and symmetry of the fingers. However our algorithm first decompose the foreground finger, and considers the two others as a whole, thus the background finger is eventually perceived as big as the middle one once the first is extracted. Designing a global method from our recursive one is a non trivial problem since the closures are interdependent in many cases.

We also show results on inputs from [Entem et al. 2014] in Figure 17 (top, middle). Though the segmentation is similar, our initialization step cannot complete complex hidden contours. This limitation would locally require a completion similar to the one used in [Sýkora et al. 2014] but it is non trivial to combine this method with figural completion to be able to completed parts with distinct visible regions in the absence of distinct similarities between these regions such as color or grouping annotations. However our algorithm tackles the case of sub-parts hinted by single suggestive contours as shown in Figure 17 (bottom).

Limitations: Even though our method is giving good results in most of the cases, we have few limitations as well. One main problem is the cyclic arrangement of parts over one another. Figure 18(a) shows an example in which the petals are overlapping to one another in a cyclic fashion. In this case, layering cannot be done

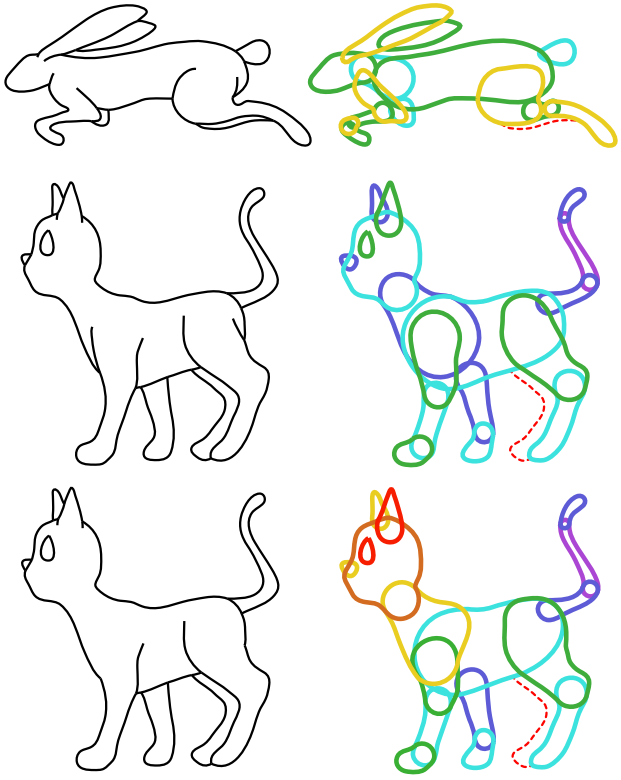

Figure 17: Results on two drawings from [Entem et al. 2014] and the third is the second drawing minus two suggestive contours, a case that their algorithm could not handle. Red dotted curves are contours that make our initialization stage fail.

using the proposed method and makes our part decomposition phase to fail. Since we are assuming that the occlusion results in either T-junctions or cusps, the more complex junctions are not processed in the current system.

Many limitations are found in the initialization step, when drawings carry ambiguities at curve intersections. Notably when T-junctions are not well defined either due to special view or surface contact as in Figure 18 (b,c) and Figure 21 (2,3), or misleading because the hypothetical occluded contour is in fact a texture change as in Figure 18 (e, left). The latter limitation can be manually worked around by erasing a part of the stroke as in Figure 18 (e, right).

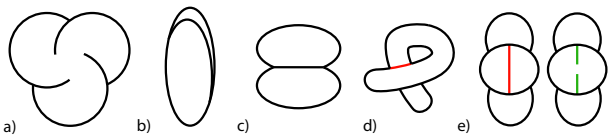

Figure 18: Different limitation cases of the initialization step (b, c, d, e) and recursive decomposition algorithm (a).

Future Work: Our current method can be improved in various directions. One of them is to include user interventions to solve ambiguous cases (recognize whether two concentric circles represent two spheres or a torus), and to do assisted-segmentation as in citeDS14. Another interesting roadway is to use decorative curves as suggestive contours. Figure 19 shows a case in which the decorative curve represents a suggestive curve and our method would ignore it.

Figure 19: An input where a suggestive contour is not connected to the outer contour and thus misclassified as as a decorative element in our initializaton step. We would like to use it as a possible segment of a foreground closure in future works.

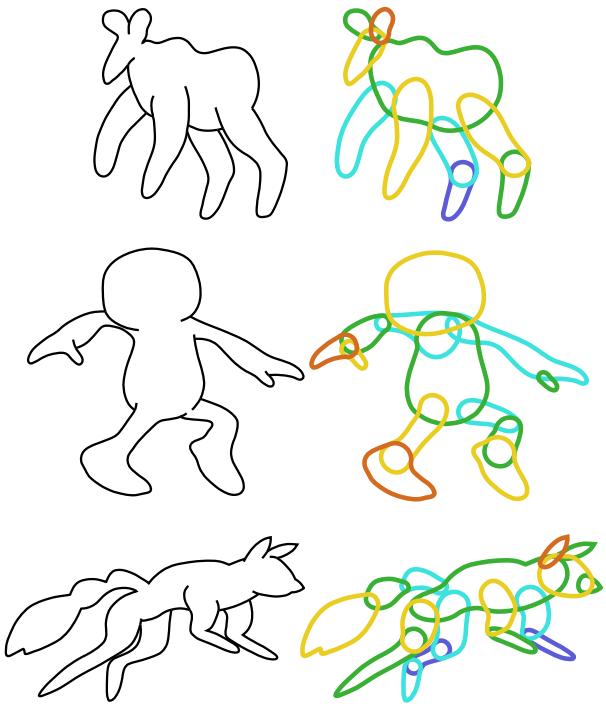

Figure 20: Results on three drawings also used in [Bessmeltsev et al. 2015], except that we removed the hat of the character. Warmer colors are in the foreground.

8 CONCLUSION

We presented the first automatic method able to use complex inner contours in the analysis and recursive decomposition of drawings

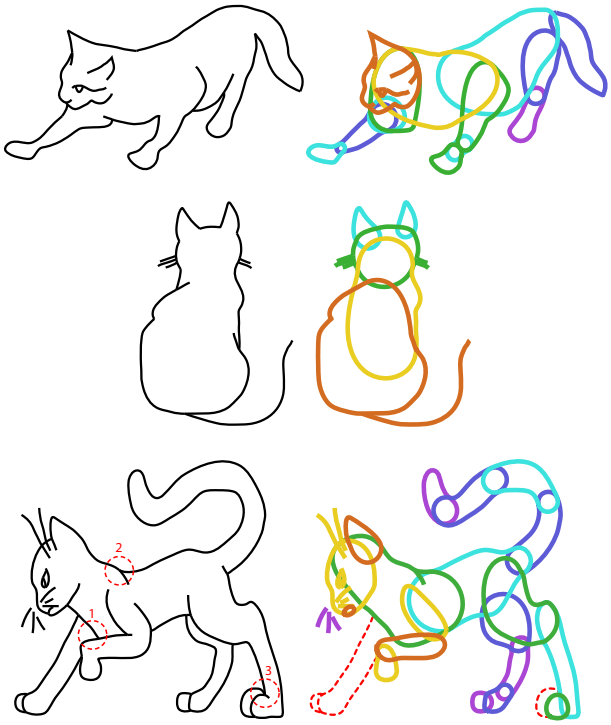

Figure 21: Results for three drawings found on the first results page of Google Images with the following search terms: "cat line drawing". The second result has been produced with salience threshold parameter that is more sensitive than the one used for all the other examples. The third example is subject to three limitations: 1) non trivial hidden part completion; 2) badly defined T-junction; 3) 4-valence vertex.

that represent smooth shapes. Our decomposition method outputs a structure of closed 2D shapes layered in depth. It relies on the inference and progressive refinement of a partial depth tree to store depth information. A new metric computed along a skeleton was proposed to detect salient parts of complex drawings including internal silhouette curves. We introduced a new, perceptual-based criterion for selecting the most salient possible junctions, prioritizing them, and using them to recursively segment a shape into parts. An efficient implementation of curve closures using a variation of Scale-Invariant MVC functional was defined for closing the extracted sub-parts, hidden or salient. Lastly, we managed to keep most parameters scale-invariant, enabling us to achieve structural decomposition of drawings with different resolutions of features.

Many applications can benefit from our method. As we have illustrated, it enables organic sketches to be easily edited in a meaningful way. Subdivision and depth layering makes the model ready for simple 2D animations. As future work, the automatic determination of the implied articulations between overlapping shape parts would make the application to vector drawing animation even more straightforward. We could also locally blend the contours at each animation step in order to enable smooth transitions between silhouette curves where and when needed, as done by the user

in Figure 1 (c). This could be done with the help of 2D implicit contours and specially designed operators. We also plan to investigate 3D shape modeling from our part decomposition method, similarly to what was done in [Bessmeltsev et al. 2015; Entem et al. 2014] for much more constrained input. When applied to vector drawing animations, such a 3D intermediate representation would enable us to apply out-of-plane rotations to limbs and to change the viewpoint, two cases in which the silhouettes and occlusion between shape parts need to be recomputed.

ACKNOWLEDGEMENTS

We would like to thank Michiel van de Panne and John Hughes for their comments on this article and the multiple proof-readings. Thanks to Bessmeltev et al. as well for sharing their input drawings. This work was funded by the advanced grant EXPRESSIVE from the European Research Council (ERC-2011-ADG 20110209).

REFERENCES

Mikhail Bessmeltsev, Will Chang, Nicholas Vining, Alla Sheffer, and Karan Singh. 2015. Modeling Character Canvases from Cartoon Drawings. *ACM Trans. Graph.* 34, 5, Article 162 (Nov. 2015), 16 pages. https://doi.org/10.1145/2801134

H. Blum and R.N. Nagel. 1978. Shape description using weighted symmetric axis features. *PR* 10 (1978), 167–180.

Minh Tuan Bui, Junho Kim, and Yunjin Lee. 2015. 3D-look shading from contours and hatching strokes. *Computers & Graphics* 51 (2015), 167 – 176. https://doi.org/10.1016/j.cag.2015.05.026 International Conference Shape Modeling International.

Axel Carlier, Kathryn Leonard, Stefanie Hahmann, and Geraldine Morin. 2016. The 2D Shape Structure Dataset: A User Annotated Open Access Database. *Computer and Graphics* 58 (2016).

Frederic Cordier and Hyewon Seo. 2007. Free-Form Sketching of Self-Occluding Objects. *IEEE Comput. Graph. Appl.* 27, 1 (Jan. 2007), 50–59. https://doi.org/10.1109/MCG.2007.8

Frederic Cordier, Hyewon Seo, Jinho Park, and Jun Yong Noh. 2011. Sketching of Mirror-Symmetric Shapes. *IEEE Transactions on Visualization and Computer Graphics* 17, 11 (Nov. 2011), 1650–1662.

Boris Dalstein, Remi Ronfard, and Michiel van de Panne. 2014. Vector graphics complexes. *ACM Trans. Graph.* 33, 4 (2014), 133.

Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. 2003. Suggestive Contours for Conveying Shape. In *ACM SIGGRAPH 2003 Papers (SIGGRAPH '03)*. ACM, New York, NY, USA, 848–855. https://doi.org/10.1145/1201775.882354

Elmar Eisemann, Sylvain Paris, and Frédo Durand. 2009. A Visibility Algorithm for Converting 3D Meshes into Editable 2D Vector Graphics. *ACM Trans. Graph.* 28, 3, Article 83 (July 2009), 8 pages. https://doi.org/10.1145/1531326.1531389

Even Entem, Loïc Barthe, Marie-Paule Cani, Frederic Cordier, and Michiel Van De Panne. 2014. Modeling 3D animals from a side-view sketch. *Computers & Graphics* (Oct. 2014). https://doi.org/10.1016/j.cag.2014.09.037

Joachim Giesen, Balint Miklos, Mark Pauly, and Camille Wormser. 2009. The Scale Axis Transform. In *Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry (SCG '09)*. ACM, 106–115.

B. K. P. Horn. 1983. The Curve of Least Energy. *ACM Trans. Math. Software* 9, 4 (Dec. 1983), 441–460. https://doi.org/10.1145/356056.356061

Takeo Igarashi and Jun Mitani. 2010. Apparent Layer Operations for the Manipulation of Deformable Objects. *ACM Trans. Graph.* 29, 4, Article 110 (July 2010), 7 pages. https://doi.org/10.1145/1778765.1778847

Olga A. Karpenko and John F. Hughes. 2006. SmoothSketch: 3D Free-form Shapes from Complex Sketches. *ACM Trans. Graph.* 25, 3 (July 2006), 589–598. https://doi.org/10.1145/1141911.1141928

Benjamin S. Kunsberg and Steven W. Zucker. 2017. Critical Contours: An Invariant Linking Image Flow with Salient Surface Organization. *CoRR* abs/1705.07329 (2017). arXiv:1705.07329 http://arxiv.org/abs/1705.07329

L. Larsson, G. Morin, A. Begault, R. Chaine, J. Abiva, E. Hubert, M. Hurdal, M. Li, B. Paniaga, G. Tran, and M.P. Cani. 2015. Identifying Perceptually Salient Features on 2D Shapes. *Research in Shape Modeling* (2015). https://doi.org/10.1007/978-3-319-16348-2_9

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

Kathryn Leonard, Geraldine Morin, Stefanie Hahmann, and Axel Carlier. 2016. A 2D Shape Structure for Decomposition and Part Similarity. In *Proceedings of ICPR*. Cancun, Mexico.

Rong Liu, Hao Zhang, Ariel Shamir, and Daniel Cohen-Or. 2009. A Part-Aware Surface Metric for Shape Analysis. *Computer Graphics Forum* 28, 2 (2009), 397–406.

Xueting Liu, Tien-Tsin Wong, and Pheng-Ann Heng. 2015. Closure-aware Sketch Simplification. *ACM Trans. Graph.* 34, 6, Article 168 (Oct. 2015), 10 pages. https://doi.org/10.1145/2816795.2818067

Xiaofeng Mi, Doug DeCarlo, and Matthew Stone. 2009. Abstraction of 2D Shapes in Terms of Parts. In *Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering (NPAR '09)*. ACM, New York, NY, USA, 15–24. https://doi.org/10.1145/1572614.1572617

Henry Packard Moreton. 1992. *Minimum Curvature Variation Curves, Networks, and Surfaces for Fair Free-form Shape Design*. Ph.D. Dissertation. Berkeley, CA, USA. UMI Order No. GAX93-30652.

Mark Nitzberg and David Mumford. 1990. The 2.1-D sketch.. In *ICCV*. IEEE, 138–144.

Gioacchino Noris, Alexander Hornung, Robert W. Sumner, Maryann Simmons, and Markus Gross. 2013. Topology-driven Vectorization of Clean Line Drawings. *ACM Trans. Graph.* 32, 1, Article 4 (Feb. 2013), 11 pages.

G. Noris, D. Sykora, A. Shamir, S. Coros, B. Whited, M. Simmons, A. Hornung, M. Gross, and R. Sumner. 2012. Smart Scribbles for Sketch Segmentation. *Computer Graphics Forum* 31, 8 (Dec. 2012), 2516–2527. https://doi.org/10.1111/j.1467-8659.2012.03224.x

Luiz Pessoa and Peter De Weerd. 2003. Filling-In: From Perceptual Completion to Cortical Reorganization.

W. A. Richards, Jan J. Koenderink, and D. D. Hoffman. 1987. Inferring three-dimensional shapes from two-dimensional silhouettes. *Journal of the Optical Society of America A* 4, 7 (Jul 1987), 1168–1175. https://doi.org/10.1364/JOSAA.4.001168

Alec Rivers, Takeo Igarashi, and Frédo Durand. 2010. 2.5D Cartoon Models. *ACM Trans. Graph.* 29, 4, Article 59 (July 2010), 7 pages. https://doi.org/10.1145/1778765.1778796

Ariel Shamir. 2008. A survey on Mesh Segmentation Techniques. *Computer Graphics Forum* 27, 6 (2008), 1539–1556. https://doi.org/10.1111/j.1467-8659.2007.01103.x

M. Singh. 2015. Visual representation of contour and shape. In *Oxford handbook of perceptual organization*. Oxford University Press, Oxford.

Daniel Sýkora, Ladislav Kavan, Martin Čadík, Ondřej Jamriška, Alec Jacobson, Brian Whited, Maryann Simmons, and Olga Sorkine-Hornung. 2014. Ink-and-ray: Bas-relief Meshes for Adding Global Illumination Effects to Hand-drawn Characters. *ACM Trans. Graph.* 33, 2, Article 16 (April 2014), 15 pages. https://doi.org/10.1145/2591011

Emmanuel Turquin, Jamie Wither, Laurence Boissieux, Marie-Paule Cani, and John F. Hughes. 2007. A sketch-based interface for clothing virtual characters. *IEEE Computer Graphics and Applications* 27, 1 (Jan. 2007), 72–81.

S. Ullman. 1976. Filling-in the Gaps: The Shape of Subjective Contours and a Model for Their Generation. *Biological Cybernetics* 25, 1 (March 1976), 1–6. https://doi.org/10.1007/BF00337043

K. B. Wiley. 2006. Druid: Use of Crossing-State Equivalence Classes for Rapid Relabeling of Knot-Diagrams Representing $2^1/_2$D Scenes.

K. B. Wiley and L. R. Williams. 2006. Representation of Interwoven Surfaces in $2^1/_2$D Drawing. *IEEE Computer Graphics and Applications* 27, 4 (2006), 70–83.

L. R. Williams and A. R. Hanson. 1996. Perceptual Completion of Occluded Surfaces. *Computer Vision and Image Understanding* 64, 1 (1996), 1–20.

Baoxuan Xu, William Chang, Alla Sheffer, Adrien Bousseau, James McCrae, and Karan Singh. 2014. True2Form: 3D Curve Networks from 2D Sketches via Selective Regularization. *Transactions on Graphics (Proc. SIGGRAPH 2014)* 33, 4 (2014).

Mingqiang Yang, Idiyo Kpalma K., and Joseph Ronsin. 2008. A Survey of Shape Feature Extraction Techniques. *Pattern Recognition, Peng-Yeng Yin (Ed.) (2008) 43-90* (Nov. 2008), 43–90. http://hal.archives-ouvertes.fr/docs/00/44/60/37/PDF/ARS-Journal-SurveyPatternRecognition.pdf

C. K. Yeh, S. Y. Huang, P. K. Jayaraman, C. W. Fu, and T. Y. Lee. 2017. Interactive High-Relief Reconstruction for Organic and Double-Sided Objects from a Photo. *IEEE Transactions on Visualization and Computer Graphics* 23, 7 (July 2017), 1796–1808. https://doi.org/10.1109/TVCG.2016.2574705