Computers & Graphics 77 (2018) 97–107

Contents lists available at ScienceDirect

Computers & Graphics

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

Special Section on SIBGRAPI 2018

Contour-aware 3D reconstruction of side-view sketches

Saulo Ramos[a,*], Diogo Fernando Trevisan[a], Harlen C. Batagelo[a], Mario Costa Sousa[b], João Paulo Gois[a]

[a] Federal University of ABC, Brazil
[b] University of Calgary, Canada

ARTICLE INFO

Article history:
Received 27 April 2018
Revised 9 September 2018
Accepted 10 September 2018
Available online 24 September 2018

Keywords:
Sketch-based modeling
Direct 3D reconstruction
HRBF implicits
Hermitian data

ABSTRACT

The 3D reconstruction from a single 2D side-view sketch faces challenges in capturing details of the curves and inferring the hidden parts. Here, we introduce an approach that ensures the 2D contours are interpolated to a suitable 3D reconstruction while capturing the small details of the sketch. To this end, we propose a novel strategy that combines a proper 3D Hermitian data generation from the vicinity of the sketches with an approach for identification and completion of sketch curves. Feasible 3D models are then generated using Hermitian Radial Basis Functions (HRBF) Implicit Surfaces. Results indicate that our approach provides not only more detailed 3D reconstructed models but also more perceptual agreement from the input sketches in comparison to previous work.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

The growing production of video games and interactive graphics content by hobbyist developers, as well as the popularization of manufacturing devices (*e.g.*, 3D printers) in the maker community, has increased the demand for 3D models by non-professional workers.

Despite the many options for 3D modeling software, the learning curve for content creation is frequently still too steep. 3D reconstruction from 2D drawings has gained attention in the last years as they allow beginner users to create 3D models from a set of 2D sketches quickly. However, when dealing with 3D reconstructions from 2D drawings, significant barriers arise, such as the ambiguity and difficulty in dealing with hidden traits [1].

Recent research shows that the most effective approaches seek to reconstruct a specific family or surfaces [2]. In our case, our method assumes the structural symmetry hypothesis present in various classes of objects where 3D reconstructions from fully drawn foreground parts are replicated in the background.

Different from approaches that impose further limitations by using generalized surfaces of revolution along a 2D skeleton estimated from the sketch, our approach preserves the sketch contours at the 3D outputs as well as estimates relief and handles parts with varying depth.

To obtain plausible 3D reconstructions, we first classify parts from the 2D input contours while inferring Hermitian data (points and normals) to be interpolated independently, then apply a Hermitian Radial Basis Function (HRBF) surface reconstruction for each part [3].

The main contributions of this work are: an automatic classification approach that allows the reconstruction of parts of sketches discarded in previous work; a skeleton-free 3D reconstruction technique based on scattered Hermitian data, which allows capturing details; the guarantee of interpolation of contours present in sketches, improving the fidelity between inputs and results; and the flexibility to flatten or round shapes to create plausible final models.

This paper is organized as follows: In Section 2 we present the related studies that inspired our approach. An overview of the technique is presented in Section 3. In Section 4 we show how contours are identified and separated into classes, and how parts containing open contours are treated. In Section 5 we cover the generation of 3D Hermitian data and the normal propagation method used to estimate shape for each part. We demonstrate how the parts of the models are allocated in Section 6 and how the parts are reconstructed using HRBF Implicits in Section 7. Later, in Section 8, we show results as well as limitations of the proposed technique. Our conclusions are drawn in Section 9.

2. Related work

Sketch-based modeling can be categorized as interactive methods, data-driven methods, multi-view methods, and single view methods.

* Corresponding author.
E-mail address: saulo.ramos@ufabc.edu.br (S. Ramos).

https://doi.org/10.1016/j.cag.2018.09.009
0097-8493/© 2018 Elsevier Ltd. All rights reserved.

Interactive methods are usually employed to draw parts of a complete object and interactively add new features from different viewpoints. After the publication of the Teddy system [4], a range of methods based on user interaction has been proposed, including methods based on the interactive edition of meshes [5], automatic artwork inflation [6], modeling 2D regions in paintings and scenes [7], and structured annotations [8]. There are also systems with natural user interfaces: SmoothSketch [9] infers plausible 3D free-form from visible-contour sketches; NaturaSketch [10] projects the strokes on a 3D canvas and allows its complete edition; FiberMesh [11] guarantees forms based on control curves by optimizing the final mesh in real time.

BendSketch [2], one of the most recent approaches, uses cue strokes to define concave and convex surfaces, rigid and valley shapes, and flat regions. This technique, however, is not able to automatically infer the depth of 3D symmetries from a single side-view drawing. For instance, for creating a pair of symmetrical limbs in a side view, the user must draw the frontmost limb, rotate the model, edit the depth of the 3D model, rotate the model again and then draw the next limb.

Data-driven methods are typically based on 3D shape or image search engines. They usually start from an input image or 2D sketch – possibly with annotations – and search for model parts in databases, or even search for entire models which fit the image or sketch. These methods are similar to shape matching methods that look for 3D models that fit into 2D drawings. Some studies have investigated the relationship between the development of 3D models from a template and an image [12], from a set of parameterized object views [13], or from classification systems using neural networks [14].

Multi-view methods generate 3D models from two or three sketches. Shin and Shin [15] used a multi-view method that required consistent sketches of the object to be reconstructed, and Gestalt principles to eliminate ambiguities. Zhang et al. [16] demonstrated the reconstruction of rotational parts, and Rivers et al. [17] extended the concept of visual hull and silhouette intersection [18]. Artistic drawings can also be reconstructed from different views using multi-view tools, combining 3D navigation with 2D drawing [19]. These interfaces were explored by Shao et al. [20] and Xu et al. [1]. Suzuki and Igarashi [21] proposed a collaborative system so that non-professional workers could reconstruct various objects.

Single view methods seek to interpret and infer the reconstruction of bodies from the silhouette and other feature strokes of sketch or a single image [22], imposing some constraints and mimicking the human ability to see 2D representations of 3D objects [23]. Jayaraman et al. [24] estimated the local geometry by creating a 3D-like soft shading guided by cues, connections and spatial arrangement of wrinkle strokes. Other works present methods to reconstruct 3D models by detecting regularities such as parallelism [25,26]. Regular features are often found in technical drawings and architectural designs where straight lines and orthogonal corners predominate. Shao et al. [20] identified crossed lines and propagation normals that allowed estimating the normals of perspective drawings. Together with some regularities like symmetry, planarity, and parallelism, they were able to complete the reconstruction of general forms [1]. Later, Iarussi et al. [27] extended the regularized curvature lines concept to non-orthogonal crossed fields estimating surface normals and 3D curvature directions from rough 2D concept sketches. Buchanan et al. [28] used circular arcs on 2D contours to create a skeleton that eliminates problems with hidden parts but enforces a range of simplifying assumptions. Bessmeltsev [29] used skeletons to match 2D character poses and recover 3D models from curve networks.

Sýkora et al. [30] presents Ink-and-Ray, a technique that simulates global illumination effects in hand-drawn characters. Instead of reconstructing a 3D model, it generates a bas-relief model that suffices to simulate the global illumination of the input view. However, this reconstruction cannot be reused in other views.

To reconstruct 3D animals, Entem et al. [31] proposed to generate skeletons for limbs from side-view sketches of animals. Estimated by contours and a set of connected curves in a graph, the medial axis is used as a 2D skeleton to create a scale-invariant implicit surface controlled by radius and his length.

However, this approach does not interpolate the contours. As a result, it may simplify or discard details of the input sketch. By leveraging the information provided by sketch, our technique successfully reconstructs models ensuring that the contours define the parts. Besides that, our approach allows the 3D reconstruction of objects with or without symmetry. We adopt a normal propagation method, previously employed for interactive shading [20,32], to propagate the Hermitian data that is further used to estimate the shape of the model.

3. Overview

The input to our method is a digital sketch in a sagittal plane. We consider that the input is an adequately filtered drawing for our purposes. Cases such as over-sketching have already been resolved, and drawings have been cleared as usually required in previous works [1,19,31]. In our case, we consider that the sketches have at least one closed contour and the curves do not present self-intersections or cross-sections.

We first build a half-edge structure from cubic Bézier curves that compose the sketches. These sequences of connected edges represent the structural parts of the model such as the main body and symmetrical parts.

Once the parts are classified, we prepare them for reconstruction using the contours in drawings in a novel method.

First, we perform a 2D sampling of points on the contours and inside the sketches. 3D normals are estimated for each sampled point using a propagation approach once that normals belonging to the contour are orthogonal to the viewer [33]. Meanwhile, we estimate the depth of the sampled points based on the width of the bounding box of the sketches. Symmetries are then computed by replicating points and normals for the symmetric parts. Together, these points and normals compose the set of 3D Hermitian data.

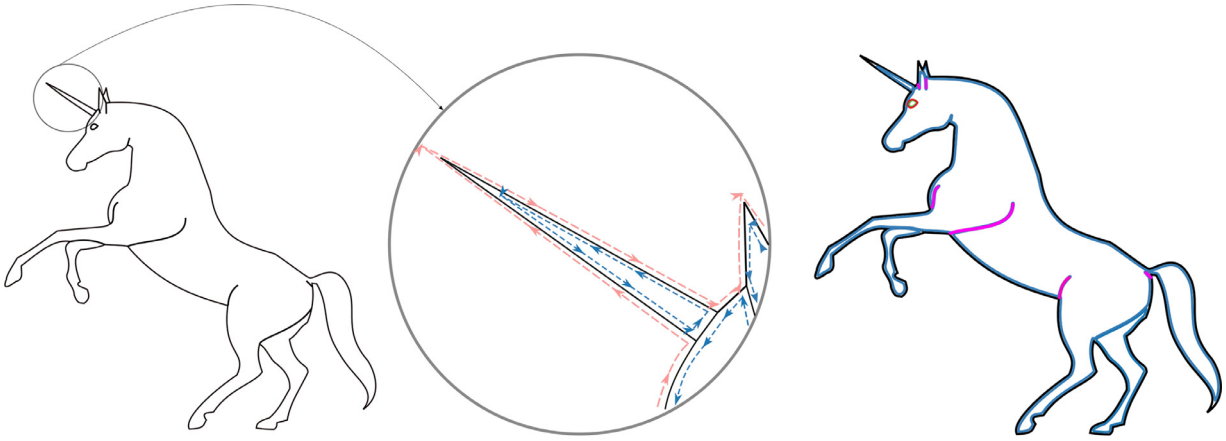
The final stage uses HRBF Implicits to reconstruct the model. The 3D model is obtained by interpolating the 3D Hermitian data sampled for each part of the drawing. Plausible results are produced using all the estimated data providing shape details, while flat or smooth surfaces are reconstructed according to the depth of the points.

4. Identification and completion

The first step in our technique is to create a half-edge data structure from the input drawing. This step automatically creates cycles, *i.e.*, closed sequences of connected half-edges as shown in Fig. 1(a). We assume the correctness of input contours in a clean vectorized drawing to avoid wrong classifications.

We identify and classify the cycles according to the following order [31], limiting the classification of each cycle to one group (Fig. 2):

1. **Outer cycle**- The largest clockwise oriented cycle that defines the external contour of the sketch.
2. **Border cycles** - Cycles with one or more half-edges belonging to the outer contour, but oriented counterclockwise. The cycle sequence is defined by traversing the half-edges of smaller angle with the previous ones.
3. **Island cycles** - Counterclockwise cycles not connected to the contour but which are inside the outer contour. The contours



(a) Input sketch and half-edge structure in detail: outer cycle, border cycle (b) Classified cycles and inner edges.

Fig. 1. Half-edge data structure created from input drawing and cycle classification with inner edges: closed sequences of half-edges define these cycles. The opposing half-edges that belong to the same cycle are named inner edges, the case of the open ends in the legs of the unicorn.

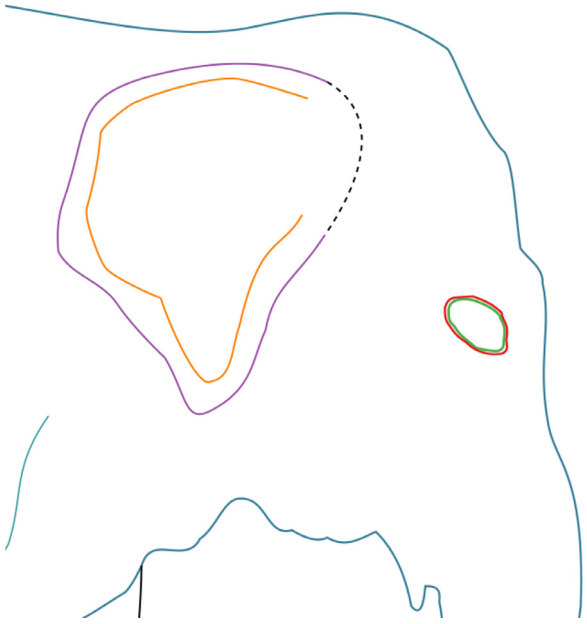


Fig. 2. Example of cycle classification: boarder cycle, island cycle, adjacent barder cycle, adjacent feature cycle and other.

that define an island cycle should be closed, creating two half-edge cycles.

4. **Adjacent border cycles** - Cycles not connected to border cycles, located inside the outer contour, and with all half-edges belonging to the same cycle.
5. **Adjacent feature cycles** - Cycles not connected to the contour, located inside the bounding box of the adjacent border cycles, and with all half-edges belonging to the same cycle.
6. **Others** - Remaining cycles that do not belong to any of the above groups, *e.g.*, the opposite clockwise cycles from island cycles.

Entem et al. [31] considered the outer, border, island and other cycles in their work. We also consider the adjacent border cycle to allow the reconstruction of parts inside border cycle contours. These new parts, classified and discarded in previous work for belonging to the class "others", are attached to the main body through an open contour and must be fully drawn inside a border cycle, as in the elephant's ear in Fig. 2. Besides that, the use of HRBF Implicits and the 3D normal propagation [33] tends to generate rounded surfaces. To reconstruct flat-like surfaces, we added adjacent feature cycles to the classification, determining that all internal points to these contours have the same depth. It ensures that the area delimited by the adjacent feature cycle is smoothed out by interpolating the points in the reconstruction.

The classification of these new cycles is done as follows. Firstly, we classify as adjacent border cycles any remaining cycle that is inside the largest border cycle. Differently from island cycles, in these contours, all half-edges belong to the same cycle. We iteratively check if any cycle is inside another cycle. If so, the smaller cycle is classified as an adjacent feature cycle.

In cycles that have open contours, *i.e.*, border cycles and adjacent border cycles, opposite half-edges (inner edges) can belong to the same cycle defining extremities of limbs of animals as the legs of the unicorn in Fig. 1(b). Different parts are defined by border cycles connected to a T-junction, a node where one contour meets another without crossing it, forming the shape of a "T". This connection indicates that there are contours to be closed in both their extremities, separating symmetric limbs from other bodies.

We close the open contours of the inner edges by creating new Bézier curves that connect their extremities, ensuring C^1 continuity (dashed lines in Figs. 3(a) and 2). Such new closed parts (Fig. 3(a)) belong to the foreground and are symmetrical to those parts drawn in the background according to the structural symmetry hypothesis. It is worth to mention that the bounding box used to classify adjacent border cycles is computed before creating the new curve that closes the contour.

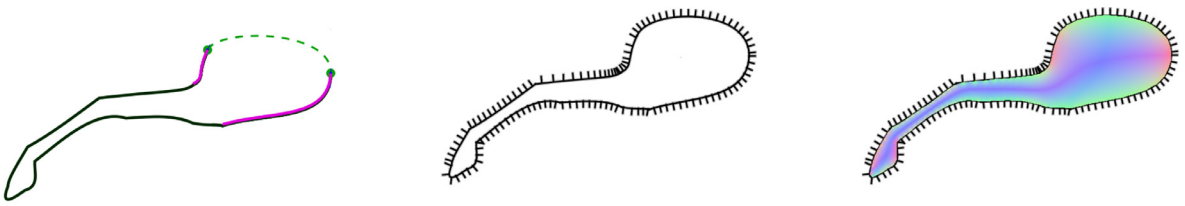
Thus, the original contours drawn in the background, classified previously as border cycles, are identified and discarded according to the proximity for each foreground part. To this end, we pair each foreground limb with a border cycle using the smaller distance between the part's bounding boxes.

For instance, the right leg of the unicorn in Fig. 1(b) is not 3D reconstructed, as we consider the reconstruction of the left leg (foreground).

5. Generating 3D Hermitian data

Once the cycles have been classified, the next step is to generate the 3D generalized Hermitian data, *i.e.*, points and normals used as constraints in the HRBF Implicits (Section 7).

Firstly, we sample points in the contours that define each part and estimate their 3D normals as shown in Fig. 3(b). Then, by propagating the normals to the interior, we can determine the



(a) All open contours containing inner edges must be properly closed with a new line (dashed line).

(b) For each part of the drawing, normals are estimated on the contour.

(c) From the normals on the contour, a normal field is generated in the interior of the contour.

Fig. 3. Normal estimation overview.

shape of each part. Specifically, we aim to sample the interior of the contours. To this end, we define a regular grid in the interior of the contour [34].

With the points sampled in the interior of the contour, our next step is to compute a 3D normal $n(p) = (n_x(p), n_y(p), n_z(p))$ for each vertex $p = (x, y, z)$ of the grid, as proposed by Nascimento et al. [33]. The components of the normal vector read as follows:

$$n_x(p) = \frac{1}{\omega(p)} \int_C \frac{n_x(s)}{|p - C(s)|^2} ds, \tag{1}$$

and

$$n_y(p) = \frac{1}{\omega(p)} \int_C \frac{n_y(s)}{|p - C(s)|^2} ds, \tag{2}$$

where C is the closed contour that defines a part of the sketch, and $\omega(p)$ is

$$\omega(p) = \int_C \frac{ds}{|p - C(s)|^2}. \tag{3}$$

Finally,

$$n_z(p) = \sqrt{1 - n_x(p)^2 - n_y(p)^2}. \tag{4}$$

Fig. 3(c) depicts the normal field computed from the normals distributed along the contour.

It is worth to highlight that we do not need to consider all the data as input to the interpolation. We noticed that better and faster results are achieved when we consider a subset of data closer to the contours. Specifically, for the presented results we only interpolated pairs of points and normals when $|n_z < 0.5|$.

The last step is to define the depth of the points, completing the generation of 3D generalized Hermitian data. For contours classified as border or island cycles, we determine the depth of the points by multiplying n_z of each point by the width of the bounding box of each part of the sketch and the value of λ_p, which adjusts the depth of the points to create rounded or flatten surfaces.

The normal propagation for a contour classified as an adjacent border cycle with an adjacent feature cycle in its interior (*e.g.*, the elephant's ear) demands a distinct procedure. First, we are assuming that an adjacent border cycle only connects to the body by the new Bézier curve we create to close the contour (Fig. 4(a)-**dashed line**). Second, the adjacent feature plays the role of determining a constant depth but maintaining the normal field of the adjacent border cycle. As shown in Fig. 4(c), the points created inside the adjacent feature cycle are mirrored below the plane defined by the adjacent border cycle.

6. Placement and assemble

The parts classified as symmetrical members, such as legs, ears, and eyes, need to be placed according to the depth dictated by the sketch because up to now all the members are located in the same plane in which the sketch is drawn. The goal is to place each part according to the sketch's suggestions, *i.e.*, the members drawn in the foreground should be placed in front of the sketching plane, whereas the background ones should be allocated behind the sketching plane.

As we disregard the original contours in the background, we only need to compute the depth of the members in the foreground and thus replicate it to the other half of the drawn in the background. We separated the sketch into two subsets:

1. Parts in the sketching plane:
   - The parts classified as border cycles and;
   - The parts that were classified as border cycle but did not contain an open contour indicating symmetry, *e.g.*, tails, and horns.
2. Symmetric parts to be copied to the background:
   - The parts classified as border cycles whose ends were closed by containing inner edges;
   - Parts classified as island cycles and;
   - Parts classified as adjacent border cycles.

Therefore, the placement of the symmetric contours is done as follows. First, we find the portion of the contour inside the border cycle (the green region in Fig. 5). Second, we find the point p (the red dot in Fig. 5) with the normal vector that maximizes n_z inside the green region. To this end, we again apply the normal propagation presented in the previous Section. We then use p to estimate the depth of the part and the symmetric contour at p, multiplying n_z of the point at the symmetric part by its respective width of the bounding box.

Then, we define along the z-axis the depth of the symmetrical part regarding the border cycle depth at that point. The depth coordinate is given by:

$$H_z = n_z \cdot \texttt{bbwidth}_b \cdot \lambda_s, \tag{5}$$

where H_z is the depth coordinate of the symmetric part, n_z is the normal value of the point in z-axis, $\texttt{bbwidth}_b$ is the side-view width of the border cycle under the symmetric limb and, λ_s is the parameter that allows to vary the depth of the symmetric limb (see Fig. 6).

For the adjacent internal parts, the placement is done using the newly generated curve (Fig. 4). The middle point of this new curve is placed at the same depth of the point at the border cycle, leaving the adjacent border part outwards to the border cycle.

Once the depth coordinates of all parts have been computed, the symmetric parts are replicated to be reconstructed in the background.

7. Hermitian radial basis functions implicits

Hermitian Radial Basis Functions (HRBF) Implicits can be used to reconstruct implicit surfaces from generalized Hermitian data, *i.e.*, points, normals and even tangents [3]. In this work, we consider only points and normals constraints. The goal of the HRBF

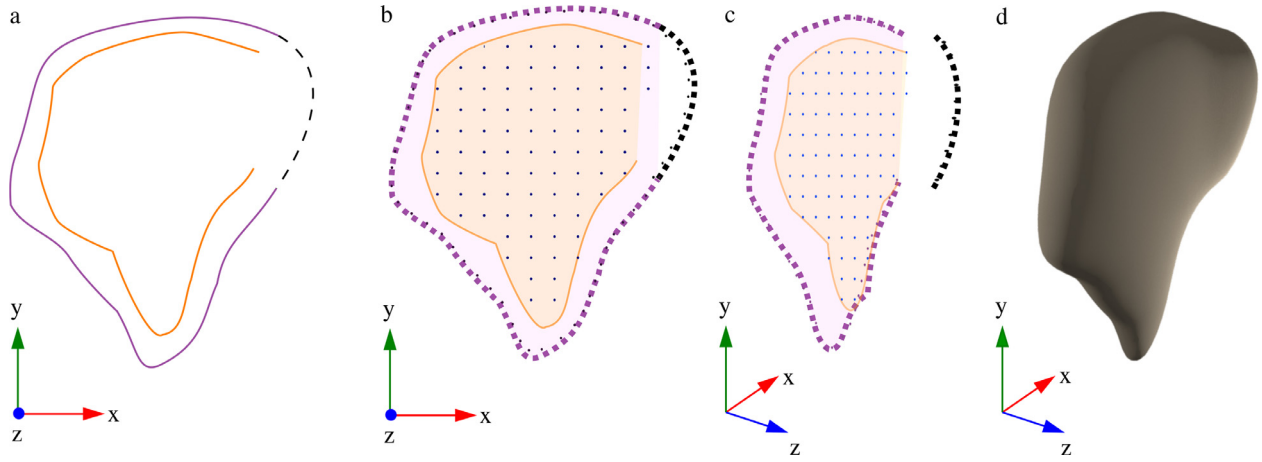


Fig. 4. The process of reconstruction of a flat adjacent internal part: After the classification, the process starts by creating a new contour that closes the adjacent limb (dashed-line in a), then, points are sampled in contours and inside the adjacent feature cycle (b), the depth of the points sampled in adjacent barder cycle remains on the sagittal plane while the depth of the points on the adjacent feature cycle are defined by computing the 3D normals $n(p)$. Therefore, we multiply the n_z by a constant to determine the depth of the new contour that closes the adjacent limb (c). Lastly, the implicit surface is reconstructed interpolating these points and normals (d).

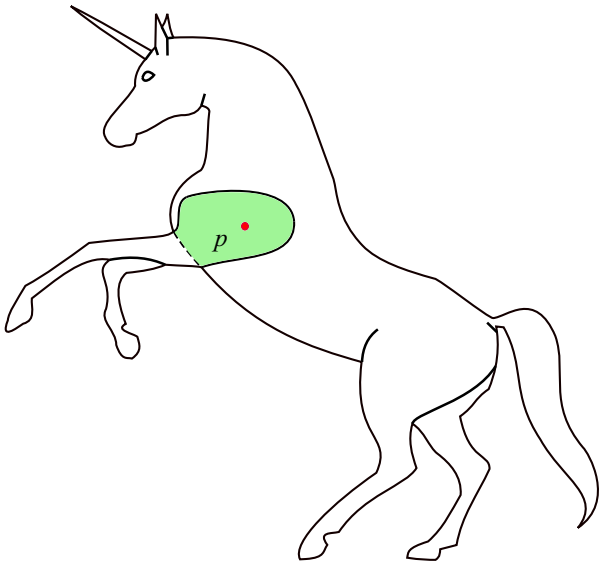


Fig. 5. The point p represents the normal with the highest value within the intersection of adjacent limb and the body classified as border cycle below. This point is used to estimate the depth of the limb at the part classified as border cycle. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Implicits for points and normals is to build a function f that interpolates $\mathbf{V}$, the set of n point constraints $\mathbf{v_j}$, and $\mathbf{C}$, the set of m normal constraints $\mathbf{g_j}$ placed at $\mathbf{c_j}$, ensuring:

$$f(\mathbf{v}_j) = 0, \ \forall \ \mathbf{v}_j \in \mathbf{V}, \tag{6}$$

and

$$\nabla f(\mathbf{c}_j) = \mathbf{g}_j, \ \forall \ \mathbf{c}_j \in \mathbf{C}. \tag{7}$$

The HRBF Implicits is defined as the zero level of the following equation:

$$f(\mathbf{x}) = \sum_{j=1}^{n} \alpha_j \phi(\mathbf{x} - \mathbf{v}_j) - \sum_{j=1}^{m} \langle \beta_j, \nabla \phi(\mathbf{x} - \mathbf{c}_j) \rangle + p(\mathbf{x}), \tag{8}$$

where $\alpha_j \in \mathbb{R}$ and $\beta_j \in \mathbb{R}^3$ are the unknown coefficients, ϕ is defined by the radial function kernel, and $p(\mathbf{x})$ is a low-order polynomial. We used the Polyharmonic Splines kernel and a complete polynomial of degree 2. The HRBF interpolation leads us to a

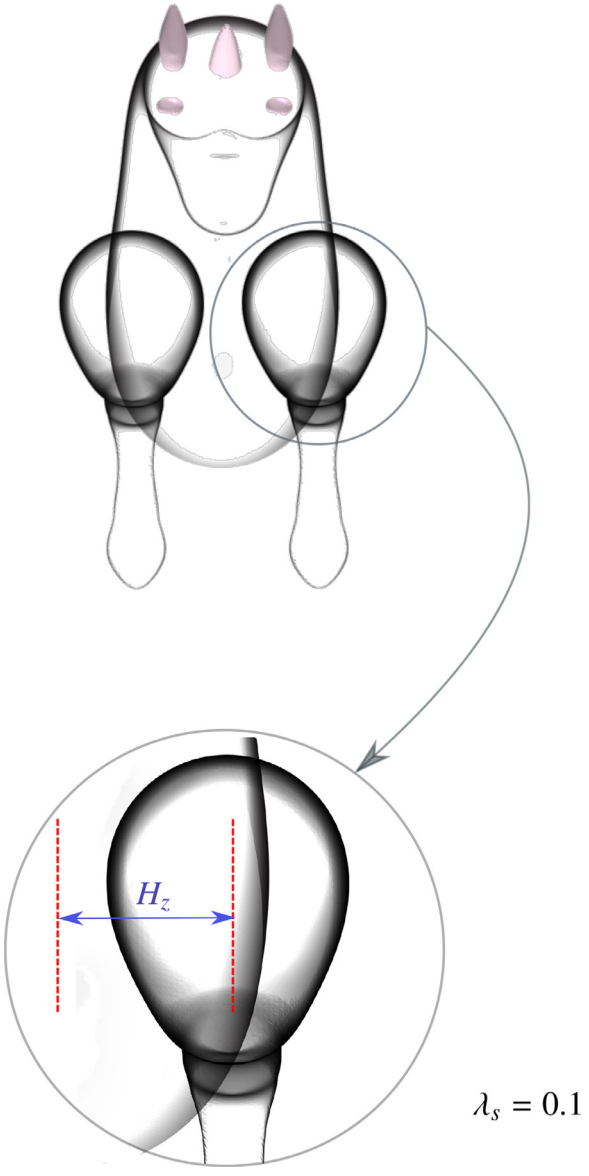


Fig. 6. Processing of the depth coordinate of the symmetric parts. H_z is the final depth coordinate of the limb concerning the sketching plane.

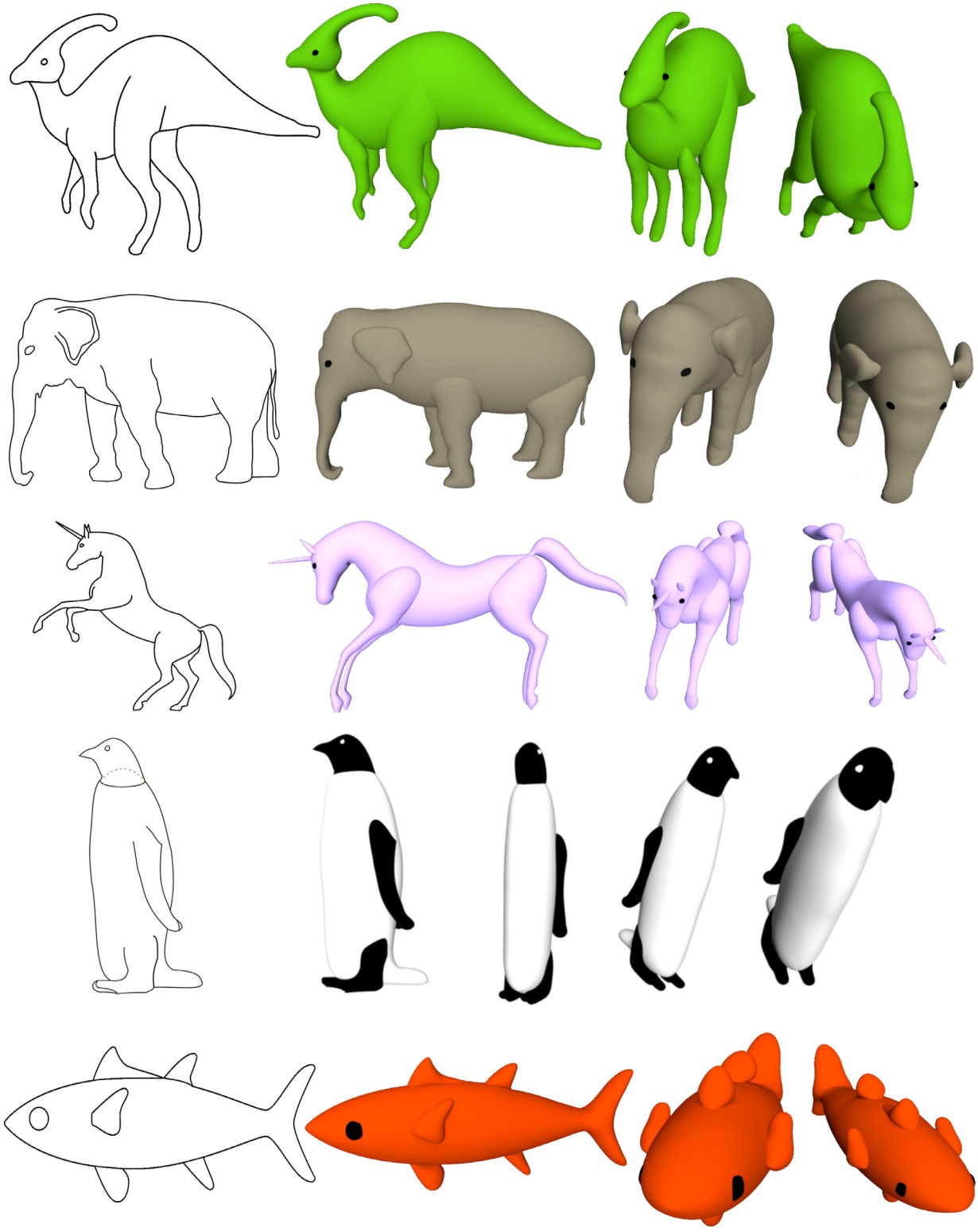


Fig. 7. Input drawings (left) and their 3D reconstructed models.

symmetric linear system. Solving this system, we obtain the coefficients for Eq. (8), thus defining the function f from which we can extract its zero level.

We use HRBF Implicits [3] due to its flexibility, which allows the insertion of point constraints extracted from the sketch contour, and normal constraints generated using the normal propagation method (Section 5), or even normals and points placed at other desired positions. The method also guarantees that the generated mesh will interpolate the input contour described by the user.

8. Discussion and results

We tested our method on a variety of sketches. For comparison, we provide drawings similar to those presented in the works of Entem et al. [31], Sýkoraet al. [30] and Li et al. [2].

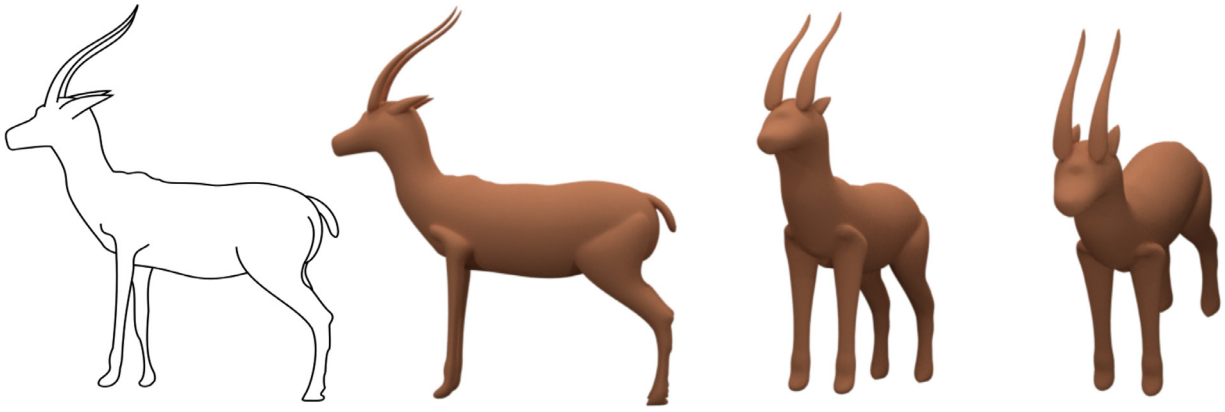


Fig. 8. Input drawing (left) and its 3D reconstructed model. Observe that the horn is close to the ear. In previous works, the propagation method for the background parts did not guarantee the correct identification of the two parts by using a heuristic based on an angular threshold. In our case, we pair limbs in foreground and background using the smaller distance between the part's bounding boxes.

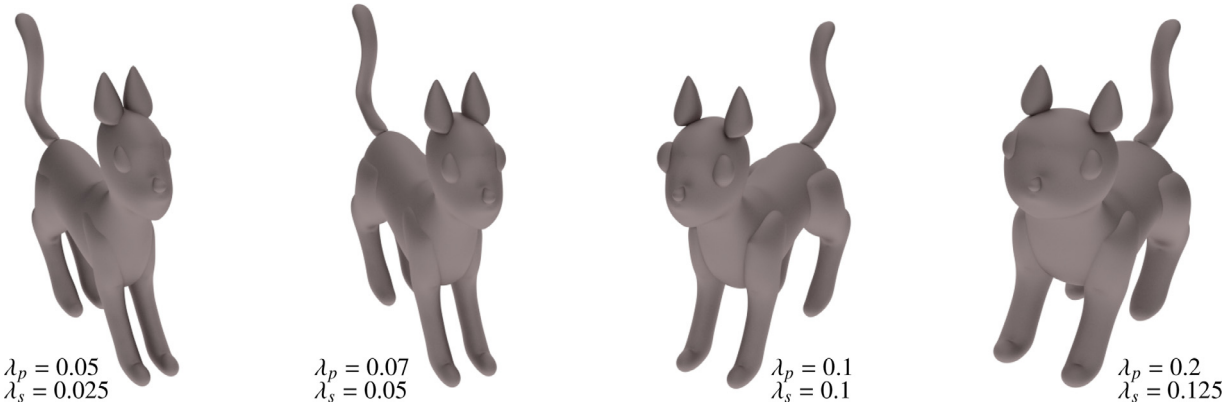


Fig. 9. Adjustment flexibility of the final model based on proportion. We use the side-view width of each part of the sketch multiplying it by the value of λ_p. Our method also provides the flexibility to alter the depth of symmetrical parts by changing the value of λ_s.

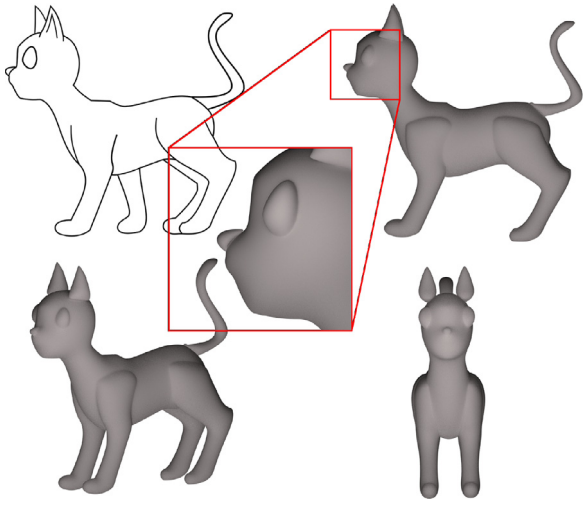


Fig. 10. Contour constraints guaranteed by reconstruction.

It can be observed how the present approach can preserve features ensuring the interpolation of the 2D curves and generate flatten regions. The proposed curve classification achieves these results (see the elephants ear in Fig. 7).

The hidden parts are allocated by using the symmetry hypothesis present in many groups of objects, including animals. This structural condition allows us to suppress the need to infer parts not contemplated by the sketch in only one view and assures us there will be no part reconstructed erroneously because of the hidden traits.

Notice the beak of the penguin in Fig. 7. Previous approaches are not able to achieve this level of fidelity from the input sketch since the final reconstruction relies on generalized cylinders for the skeleton nodes [7,23,31,35]. Still, in the penguin, the body and the head were reconstructed separately, not being necessary dealing with the T-junctions that separate them. Notice that a part of the body is inside the head in the final mesh, softening the encounter between the parts. In the fish model, the difficulty lies in avoiding undesirable fat or skinny results.

The 3D reconstruction of the unicorn's horn (Fig. 7) is successful because a side-view sketch can entirely describe the reconstruction, instead, *e.g.*, a mooses horn, that contains details perceived only in other views. In the dinosaur model, the neck becomes thinner near the head. This thinning is described by the straight contour that closes this part, creating a beveled mesh in the reconstruction.

It is worth to mention that, besides the flexibility to change the aspect ratio used to compute the depth of the points in the models, the grid density of the sample points inside the contours can also be adjusted. Regarding the reconstruction of the rabbit's paw in Fig. 13, observe that it is connected to the main body by two small strokes that delimit the junction of the meshes. The HRBF interpolation ensures that the mesh is smoothed between all points even for different depth levels. The input design of this model does not contain an adjacent feature cycle having only one adjacent

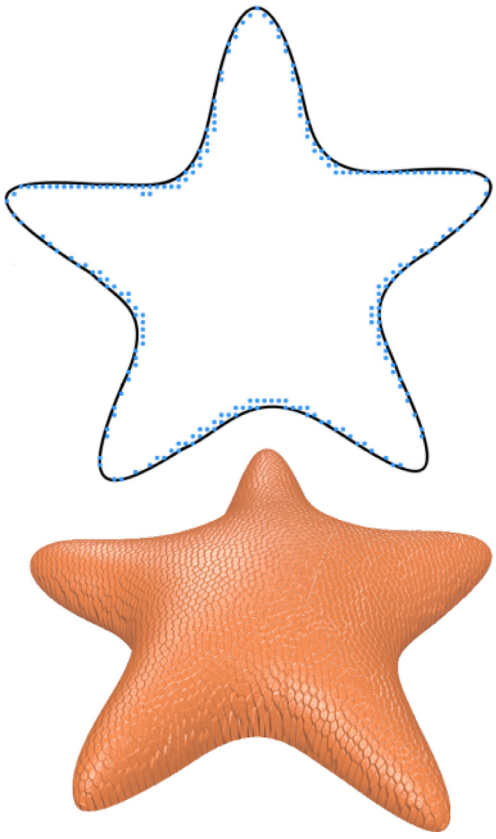


Fig. 11. Starfish reconstructed from a subset of data closer to the contours. The center of the sketch does not contain points or normals to be interpolated.

border cycle, resulting in a rounded mesh differently from the elephant ear shown in Fig. 7.

These values are used to retrieve more details from sketches, varying the number of points to be interpolated. This flexibility, together with the possibility to increase or decrease the number of pairs of points and normals delimited by the value of n_z (Section 5) allows to vary the predictability in reconstruction and consequently for each part in the final model. In our results, the sampled points on contours were spaced at intervals of 10 pixels, whereas the grid spacing was adjusted to 20 pixels.

In Fig. 10, we emphasize the face and the nose of a cat model reconstructed with all features preserved. Notice that all symmetric limbs in the background were reconstructed as the foreground limbs.

Different from other HRBF reconstructions that use local support functions [36], our choice allows us to reconstruct surfaces with fewer points using global support functions. It can be observed in Fig. 11 that the interpolation of points and normals that are closer to the contours ($n_z < 0.5$) could improve the shape reconstructed where no points were allocated. This is observed in the center of the starfish, creating a smooth mesh that covers the surface between the points and normals in the grid.

Besides that, the flexible aspect ratio used to estimate the shape of parts allowed us to obtain flatter or rounder shapes according to the sketch. This allows us to empirically alter the bodies and the depth coordinate of the parts by observing the proportions used by artists. Fig. 9 compares different aspect ratios used for the same drawing, based on the width of each part of the sketches multiplied by the value of λ_p and λ_s.

The attachments on the final model could be smoothed. To this end, we group each part and merge it using a Screened Poisson surface reconstruction [37]. We apply this reconstruction in the goat, rabbit and bird models, shown in Figs. 8, 13 and 14, making them smoother at junctions. Notice that the bird's wing is an island cycle, which is replicated in the background automatically and does not have an attachment created with points under the wing's plane, unlike the rabbit's paw which is an adjacent border cycle.

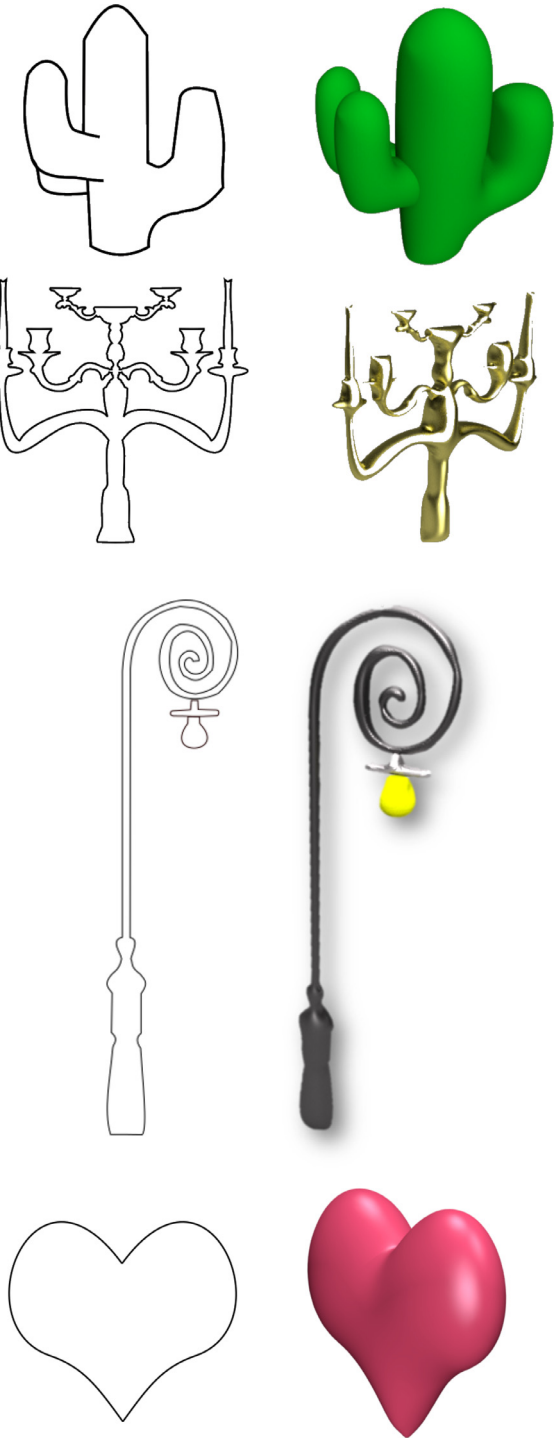


Fig. 12. Our approach has the potential to be applied to the reconstruction of a variety of objects, symmetrical or not.

Our approach has a potential to be applied to the reconstruction of further objects, symmetrical or not. In Fig. 12, we present other objects reconstructed by our method. Our method only

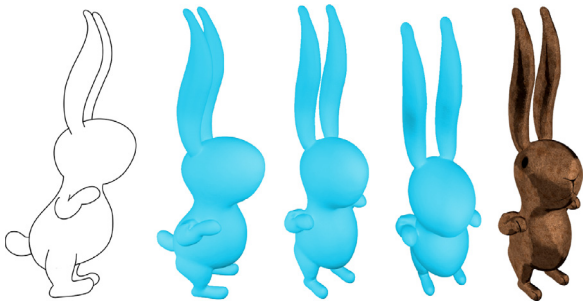


Fig. 13. Rabbit generated by our method. In the rabbit's paw, our classification of adjacent internal parts, when used without a feature cycle, creates rounded limbs like other parts where the depth of the points is estimated based on the width of the part on the sketch.

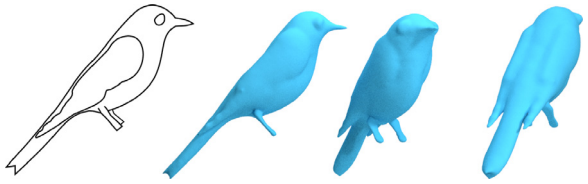


Fig. 14. Result from merging and smoothing processes in a bird model.

Table 1
Time to solve the linear systems for each model. The matrix order varies according to the number of points sampled in the sketches, making the method flexible to allow the user to choose more accurate or faster results.

	Points	Normals	Matrix order	Time(s)
Goat	1024	1024	4100	1.29 s
Cactus	214	214	860	0.12 s
Candle holder	342	342	1372	1.18 s
Rabbit	794	794	3180	0.60 s
Dinosaur	968	968	3876	1.64 s
Elephant	984	984	3940	3.13 s
Cat	1433	1433	5736	2.70 s
Bird	348	348	1396	0.31 s
Fish	658	658	2636	1.27 s
Penguin	927	927	3712	1.81 s
Unicorn	1078	1078	4316	2.13 s

Table 2
Time to solve the linear system for the cat model.

Sampling (Contour/Grid)	Points	Normals	Matrix order	Time (s)
5 px / 10 px	3785	3785	15144	29.32 s
10 px / 20 px	1433	1433	5736	2.70 s
20 px / 40 px	601	601	2408	0.41 s
x				

requires a closed contour that represents an outer cycle and a border cycle obeying the restrictions at the input, which allows using any sketch, as the candlestick in Fig. 12. By combining this with the hypothesis of structural symmetry and the mandatory drawn of suggestive contours, other symmetrical objects can also be reconstructed automatically, as the cactus in Fig. 12.

We implemented and ran our method on a desktop computer with a 3.7Ghz Intel Core i7 processor and 16GB memory. Table 1 shows the timing statistics to solve the linear systems for the generated models. The current bottleneck lies in the cost related to the number of points and normals used to solve the linear system (Table 2). However, we note that our 3D models ensure plausible features from sketches with spacing points and normals every 10 pixels in contours, and every 20 pixels in inside contours, while providing affordable time-consuming.

(a) Points sampled every 5 px in contour and 10 px in grid.

(b) Points sampled every 10 px in contour and 20 px in grid.

(c) Points sampled every 20 px in contour and 40 px in grid.

Fig. 15. Results with different sampling rates shown in Table 2. The sampling distance can be altered in the contour and in the grid points where the normals will be propagated within the contours.

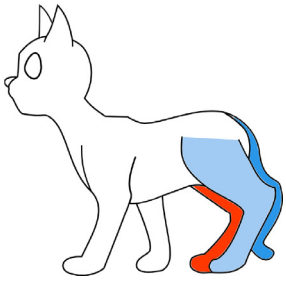


Fig. 16. Failed classification when parts are drawn too close. In this case, the method interprets the tail as a leg.

Fig. 15 shows the difference between the various sampling rates indicated in Table 2. It is possible to observe that the more points sampled in the interior, the more plausible the mesh. A small number of points sampled inside the contour could generate thicker shapes, once that most points used in HRBF interpolation are in contours where the $n_z = 0$. Given the size of the cat's head, the ear generated in the model in Fig. 15(c) is incompatible, unlike the ear in Fig 15(a). This incompatibility occurs because the spacing in the grid is so large that very few points whose $n_z < 0.5$ were sampled inside contours, leading the reconstruction to use mainly the points sampled in the contours whose normals directions are parallel to the sagittal plane.

Furthermore, while decreasing the number of points sampled in the contours, some features from the sketches start to disappear, as can be seen in the cat's nose (Fig. 15(c)). In this case, the HRBF interpolation extrapolates the actual size of this part because it does not have enough points sampled in the contour.

Some issues remain to be investigated in the future:

- The classification of the sketches is far from dealing with all cases, such as sketches containing adjacent parts in island cycles or inside symmetric limbs. Even with trait recognition to better define bodies, such as adjacent border cycles curves, and recognition of internal limbs, it is not possible to identify an adjacent limb on another adjacent limb;
- The method does not handle cases in which the same part overlaps, for example, a snake that overlaps itself, or the overlapping tentacles of an octopus;
- It is mandatory that symmetrical limbs be drawn in the foreground and background to identify the contours that need to be duplicated in the final model. Although the distance between bounding boxes works well for most cases, if three parts are drawn too close, the classification may fail as shown in Fig. 16.;
- The reconstruction of parts that are curved in the normal direction to the surface is not supported. For instance, the depth of horns and wings cannot be identified from a single side-view drawing. Fig. 17 shows an elk's horn in side-view and its

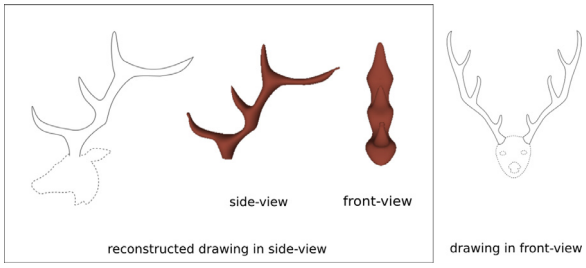


Fig. 17. Curved parts in the normal direction cannot be inferred. From left to right: Elk's horn in side-view, its reconstruction, and its rotation to the front-view. One could expect a curved horn in the reconstructed front-view, as depicted in the front-view drawing (right).

reconstruction. When we rotate the reconstruction to the front-view, we can observe that it is not curved. On the other hand, one could expect that the horn must be curved, as depicted in the front-view drawing on the right.

Another limitation is the addressing of open curves. We need to assume that all curves are closed to classify and interpret the cycles correctly. All curves must be correctly connected, and suggestive contours defining adjacent parts are mandatory. Otherwise, the half-edge structure will contain cycles that do not match the original sketch and will be classified incorrectly. However, there is still a demand in the artistic community to support rough sketches. It is common for artists to sketch objects with hidden features, open contours, strong or weak traits, and overtraced strokes. Overtraced strokes could improve the reconstruction indicating important characteristics of the objects.

9. Conclusions

Techniques for 3D reconstruction from sketches have become popular in the last years. One of the goals is to provide, to both the layman and the professional, design tools to ease the production of 3D models.

In this work, we present a method for 3D reconstruction from sketches. The final model is obtained using points and normals estimated by a method of normal propagation and a depth estimate of the parts of the model, which are estimated from the drawings. This eliminates the need of skeletons to guide the reconstruction of the parts. Besides, some parts that were not identified in previous works were reconstructed from a new classification strategy, as demonstrated.

As far as we know, this is the first work that explores the problem of reconstruction from a side-view sketch with a skeleton-free technique based on scattered Hermitian data. In addition, it is known from previous works that the estimated skeleton produces ambiguous models and does not allow the classification and reconstruction of contours that are not in the sagittal plane.

Our technique is capable of capturing the details of the sketches, thus generating plausible models. However, other methods, such as those based on Poisson Equation [38], Biharmonic Fields [39] or Laplacian Reconstruction [40], could be exploited to achieve smoother surfaces.

A direction for future work is to use drawing annotations to indicate various types of surfaces (Fig. 12) and to explore the method of normal propagation and HRBF Implicits interpolation to improve fidelity and the depth placement of the limbs. Another direction is to proportionally increase the number of points according to the variation of curvature of the strokes, therefore improving the features preserved in the final models. Furthermore, it is possible to investigate and compare functions with global support and compact support, focusing on techniques that present solutions with closed formulas [36].

Also, an image-based reconstruction could use sketches with images to texture the drawn parts in the input. This could generate more realistic or cartooned models even for users who are not artists but would like to create models from simple 2D sketches.

Acknowledgments

The authors thank to São Paulo Research Foundation – FAPESP (proc. 2014/11067-1) and the National Council of Technological and Scientific Development – CNPq (proc. 305241/2013-3) for the financial support of this work. This study was also financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) – Finance Code 001.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cag.2018.09.009.

References

[1] Xu B, Chang W, Sheffer A, Bousseau A, McCrae J, Singh K. True2form: 3d curve networks from 2d sketches via selective regularization. ACM Trans Graph 2014;33(4) 131:1-131:13.
[2] Li C, Pan H, Liu Y, Tong X, Sheffer A, Wang W. Bendsketch: modeling freeform surfaces through 2d sketching. ACM Trans Graph (TOG) 2017;36(4):125.
[3] Gois JP, Trevisan DF, Batagelo HC, Macêdo I. Generalized hermitian radial basis functions implicits from polygonal mesh constraints. Vis Comput 2013;29(6–8):651–61. doi:10.1007/s00371-013-0802-8.
[4] Igarashi T, Matsuoka S, Tanaka H. Teddy: a sketching interface for 3d freeform design. In: Proceedings of the ACM siggraph 2007 courses. ACM; 2007. p. 21.
[5] Nealen A, Sorkine O, Alexa M, Cohen-Or D. A sketch-based interface for detail-preserving mesh editing. In: Proceedings of the ACM SIGGRAPH 2007 courses. ACM; 2007a. p. 42.
[6] Joshi P, Carr NA. Repoussé: automatic inflation of 2d artwork. In: Proceedings of the SBM; 2008. p. 49–55.
[7] Bernhardt A, Pihuit A, Cani M-P, Barthe L. Matisse: painting 2d regions for modeling free-form shapes. In: Proceedings of the SBM'08-Eurographics Workshop on Sketch-Based Interfaces and Modeling. Eurographics Association; 2008. p. 57–64.
[8] Gingold Y, Igarashi T, Zorin D. Structured annotations for 2d-to-3d modeling. ACM Trans Graph (TOG) 2009;28(5):148.
[9] Karpenko OA, Hughes JF. Smoothsketch: 3d free-form shapes from complex sketches. ACM Trans. Graph. (TOG) 2006;25(3):589–98.
[10] Olsen L, Samavati F, Jorge J. Naturasketch: Modeling from images and natural sketches. IEEE Comput. Graph. Appl. 2011;31(6):24–34.
[11] Nealen A, Igarashi T, Sorkine O, Alexa M. Fibermesh: designing freeform surfaces with 3d curves. ACM Trans Graph (TOG) 2007;26(3):41.
[12] Kraevoy V, Sheffer A, van de Panne M. Modeling from contour drawings. In: Proceedings of the 6th eurographics symposium on sketch-based interfaces and modeling. ACM; 2009. p. 37–44.
[13] Yang C, Sharon D, van de Panne M. Sketch-based modeling of parameterized objects.. In: Proceedings of the SIGGRAPH sketches. Citeseer; 2005. p. 89.
[14] Wang F, Kang L, Li Y. Sketch-based 3d shape retrieval using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 1875–83.
[15] Shin B-S, Shin YG. Fast 3D solid model reconstruction from orthographic views. Comput-Aided Des 1998;30(1):63–76. doi:10.1016/S0010-4485(97)00054-7.
[16] Zhang S, Shi Y, Fan H, Huang R, Cao J. Serial 3D model reconstruction for machining evolution of rotational parts by merging semantic and graphic process planning information. CAD Comput Aided Des 2010;42(9):781–94. doi:10.1016/j.cad.2010.04.007.
[17] Rivers A, Durand F, Igarashi T. 3d modeling with silhouettes. ACM Trans Graph 2010;29(4) 109:1–109:8. doi:10.1145/1778765.1778846.
[18] Matusik W, Buehler C, Raskar R, Gortler SJ, McMillan L. Image-based visual hulls. In: Proceedings of the 27th annual conference on computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co.; 2000. p. 369–74.
[19] Bae S-H, Balakrishnan R, Singh K. Ilovesketch: as-natural-as-possible sketching system for creating 3d curve models. In: Proceedings of the 21st annual ACM symposium on user interface software and technology. ACM; 2008. p. 151–60.
[20] Shao C, Bousseau A, Sheffer A, Singh K. Crossshade: shading concept sketches using cross-section curves. ACM Trans Graph 2012;31(4) 45:1-45:11.
[21] Suzuki R, Igarashi T. Collaborative 3d modeling by the crowd. In: Proceedings of the 43rd graphics interface conference. School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada: Canadian Human-Computer Communications Society; 2017. p. 124–31. ISBN 978-0-9947868-2-1. doi:10.20380/GI2017.16.

[22] Yeh C-K, Huang S-Y, Jayaraman PK, Fu C-W, Lee T-Y. Interactive high-relief reconstruction for organic and double-sided objects from a photo. IEEE transactions on visualization and computer graphics 2017;23(7):1796–808.
[23] Cordier F, Seo H, Park J, Noh JY. Sketching of mirror-symmetric shapes. IEEE Trans Vis Comput Graph 2011;17(11):1650–62.
[24] Jayaraman PK, Fu C-W, Zheng J, Liu X, Wong T-T. Globally consistent wrinkle-aware shading of line drawings. IEEE Trans Vis Comput Graph 2017;24(7):2103–17.
[25] Lee S, Feng D, Gooch B. Automatic construction of 3d models from architectural line drawings. In: Proceedings of the 2008 symposium on interactive 3D graphics and games. ACM; 2008. p. 123–30.
[26] Lau M, Saul G, Mitani J, Igarashi T. Modeling-in-context: user design of complementary objects with a single photo. In: Proceedings of the seventh sketch-based interfaces and modeling symposium. Eurographics Association; 2010. p. 17–24.
[27] Iarussi E, Bommes D, Bousseau A. Bendfields: regularized curvature fields from rough concept sketches. ACM Trans Graph (TOG) 2015;34(3):24.
[28] Buchanan P, Mukundan R, Doggett M. Automatic single-view character model reconstruction. In: Proceedings of the international symposium on sketch-based interfaces and modeling. ACM; 2013. p. 5–14.
[29] Bessmeltsev M. Recovering 3d shape from concept and pose drawings. University of British Columbia; 2016.
[30] Sỳkora D, Kavan L, Čadík M, Jamriška O, Jacobson A, Whited B, et al. Ink-and-ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters. ACM Trans Graph (TOG) 2014;33(2):16.
[31] Entem E, Barthe L, Cani MP, Cordier F, Van De Panne M. Modeling 3D animals from a side-view sketch. Comput Graph (Pergamon) 2015;46:221–30. doi:10.1016/j.cag.2014.09.037.
[32] Gois JP, Marques BAD, Batagelo HC. Interactive shading of 2.5 D models. In: Proceedings of the 41st graphics interface conference; 2015a. p. 89–96.
[33] Nascimento R, Queiroz F, Rocha A, Ren TI, Mello V, Peixoto A. Colorization and illumination of 2d animations based on a region-tree representation. In: Proceedings of the 24th SIBGRAPI conference on graphics, patterns and images (Sibgrapi),. IEEE; 2011. p. 9–16.
[34] Gois JP, Marques BAD, Batagelo HC. Interactive shading of 2.5d models. In: Proceedings of the 41st graphics interface conference. Toronto, Ont., Canada, Canada: Canadian Information Processing Society; 2015b. p. 89–96. ISBN 978-0-9947868-0-7.
[35] Bessmeltsev M, Chang W, Vining N, Sheffer A, Singh K. Modeling character canvases from cartoon drawings. ACM Trans Graph (TOG) 2015;34(5):162.
[36] Liu S, Wang CC, Brunnett G, Wang J. A closed-form formulation of hrbf-based surface reconstruction by approximate solution. Comput-Aided Des 2016;78:147–57.
[37] Kazhdan M, Hoppe H. Screened poisson surface reconstruction. ACM Trans Graph (ToG) 2013;32(3):29.
[38] Waechter M, Beljan M, Fuhrmann S, Moehrle N, Kopf J, Goesele M. Virtual rephotography: novel view prediction error for 3d reconstruction. ACM Trans Graph (TOG) 2017;36(1):8.
[39] Argudo O, Brunet P, Chica A, Vinacua À. Biharmonic fields and mesh completion. Graph Models 2015;82:137–48.
[40] Ghiasi G, Fowlkes CC. Laplacian pyramid reconstruction and refinement for semantic segmentation. In: Proceedings of the European conference on computer vision. Springer; 2016. p. 519–34.