Universidade Federal do ABC

Saulo Ramos de Carvalho Pereira

Sketch-Based Modeling From Single-view Drawings and Applications

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro da CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), Portaria N°206, de 4 de Setembro de 2018 - Brazil (CAPES) – Finance Code 001.

Santo André, 2020

Universidade Federal do ABC

Centro de Matemática, Computação e Cognição

Saulo Ramos de Carvalho Pereira

Sketch-Based Modeling From Single-view Drawings and Applications

Orientador: Prof. Dr. João Paulo Gois

Coorientador: Prof. Dr. Mario Costa Sousa

Tese de doutorado apresentada ao Centro de Matemática, Computação e Cognição para aprovação no exame de Defesa de Doutorado no Programa de Pós-Graduação em Ciência da Computação.

Esta é a versão original da tese, tal como submetida à Comissão Julgadora.

Santo André, 2020

Dados Internacionais de Catalogação-na-Publicação (CIP)

Divisão Biblioteca Central da UFABC

Pereira, Saulo Ramos de Carvalho

Sketch-Based Modeling From Single-view Drawings and Applications/ Saulo Ramos de Carvalho Pereira

Santo André, 2020.

155f.

Tese de Doutorado - Curso de Ciência da Computação - Área de Computação Gráfica - Universidade Federal do ABC, 2020. Orientador: João Paulo Gois. Co-orientador: Mario Costa Sousa.

Modelagem e interfaces baseadas em Sketches (SBIM), HRBF Implícitas, Rendering Não-fotorealístico (NPR), Princípios e práticas da ilustração científica tradicional, Percepção de Profundidade, Sobreposição de Profundidade Universidade Federal do ABC. Divisão de UFABC. II. Título.

REFERÊNCIA BIBLIOGRÁFICA

Ramos, S. **Sketch-Based Modeling From Single-view Drawings and Applications**. 2020. 155f. Tese de Doutorado - Universidade Federal do ABC, Santo André.

CESSÃO DE DIREITOS

NOME DO AUTOR: Saulo Ramos de Carvalho Pereira

TÍTULO DO TRABALHO: Sketch-Based Modeling From Single-view Drawings and Applications.

TIPO DO TRABALHO/ANO: Tese / 2020

0 Saulo Ramos de Carvalho Pereira
Rua Cipriano Funtan, 113
CEP 08320-451 - São Paulo - SP

Declaração de atendimento às Observações

Este exemplar foi revisado e alterado em relação à versão original, de acordo com as observações levantadas pela banca no dia da defesa, sob responsabilidade única do autor e com a anuência de seu orientador.

Santo André, ____ de ____________ de 20____.

Assinatura do Aluno: ____________________________

Assinatura do Orientador: ____________________________

SIGAA - Sistema Integrado de Gestão de Atividades Acadêmicas
UFABC - Fundação Universidade Federal do ABC
Programa de Pós-Graduação em Ciência da Computação
CNPJ n° 07.722.779/0001-06
Av. dos Estados, 5001 - Bairro Santa Terezinha - Santo André - SP - Brasil
poscomp@ufabc.edu.br

UFABC

FOLHA DE ASSINATURAS

Assinaturas dos membros da Banca Examinadora que avaliou e aprovou a Defesa de Tese de Doutorado do candidato SAULO RAMOS DE CARVALHO PEREIRA, realizada em 20 de Outubro de 2020:

Dr. JOAO PAULO GOIS, UFABC

Presidente - Interno ao Programa

Dr. EMILIO ASHTON VITAL BRAZIL, IBM RESEARCH

Membro Titular - Examinador(a) Externo à Instituição

Dr. AFONSO PAIVA NETO, USP

Membro Titular - Examinador(a) Externo à Instituição

Dr. MARIO COSTA SOUSA, UOFC

Membro Titular - Examinador(a) Externo à Instituição

Dr. ANDRE LUIZ BRANDAO, UFABC

Membro Titular - Examinador(a) Interno ao Programa

Dr. CHARLES MORPHY DIAS DOS SANTOS, UFABC

Membro Suplente - Examinador(a) Externo ao Programa

Dr. HARLEN COSTA BATAGELO, UFABC

Membro Suplente - Examinador(a) Externo ao Programa

Dr. THALES MIRANDA DE ALMEIDA VIEIRA, UFAL

Membro Suplente - Examinador(a) Externo à Instituição

SIGAA - Sistema Integrado de Gestão de Atividades Acadêmicas
UFABC - Fundação Universidade Federal do ABC
Programa de Pós-Graduação em Ciência da Computação
CNPJ nº 07.722.779/0001-06
Av. dos Estados, 5001 - Bairro Santa Terezinha - Santo André - SP - Brasil
poscomp@ufabc.edu.br

No dia 20 de Outubro de 2020 às 14:00, https://meet.google.com/wqp-wxkc-yqd, realizou-se a Defesa da Tese de Doutorado, que constou da apresentação do trabalho intitulado "Modelagem Sketch-Based a partir de Desenhos em Vista Única e Aplicações" de autoria do candidato, SAULO RAMOS DE CARVALHO PEREIRA, RA nº 141610016, discente do Programa de Pós-Graduação em CIÊNCIA DA COMPUTAÇÃO da UFABC. Concluídos os trabalhos de apresentação e arguição, o candidato foi considerado aprovado pela Banca Examinadora.

E, para constar, foi lavrada a presente ata, que vai assinada pelos membros da Banca.

Dr. JOAO PAULO GOIS, UFABC

Presidente - Interno ao Programa

Dr. EMILIO ASHTON VITAL BRAZIL, IBM RESEARCH

Membro Titular - Examinador(a) Externo à Instituição

Dr. AFONSO PAIVA NETO, USP

Membro Titular - Examinador(a) Externo à Instituição

Dr. MARIO COSTA SOUSA, UOFC

Membro Titular - Examinador(a) Externo à Instituição

Dr. ANDRE LUIZ BRANDAO, UFABC

Membro Titular - Examinador(a) Interno ao Programa

Dr. CHARLES MORPHY DIAS DOS SANTOS, UFABC

Membro Suplente - Examinador(a) Externo ao Programa

Dr. HARLEN COSTA BATAGELO, UFABC

Membro Suplente - Examinador(a) Externo ao Programa

Dr. THALES MIRANDA DE ALMEIDA VIEIRA, UFAL

Membro Suplente - Examinador(a) Externo à Instituição

Ressalvas e sugestões da Banca examinadora:

Conforme Resolução da CPG nº 05, de 26 de abril de 2016, os membros que participaram de modo remoto, pelos quais o Presidente da Banca assinou no anverso desta Ata, foram:

Emilio A. Vital Brazil; Afonso Paiva Neto; Mario Costa Sousa; André Luiz Brandão; João Paulo Gois

Por sugestão da Banca Examinadora, o novo título passa a ser (em letra de forma e legível):

Assinatura do Presidente da Banca (apenas em caso de alteração)

DEDICATÓRIA

A meu pai, Adalberto pelo exemplo de pai amoroso e respeitável, pelo apoio e suporte incondicional em todos os momentos.

A minha mãe, Aurea pela orientação, cuidado e amor durante toda a jornada da minha vida.

A meu irmão, Henrique, pelo combate lado a lado, por ter seguido meus passos e ter me forçado a trilhar o caminho a seguir.

A meus amores, Thayná e Braian, fontes de inspiração e alegria, que estiveram do meu lado, mesmo em momentos de ausência durante o tempo de desenvolvimento desta tese.

A meus orientadores, João Paulo e Mario, faço esta singela homenagem, pois além de orientadores, são acima de tudo, grandes amigos que desde o primeiro instante, confiaram e me suportaram para a realização deste sonho.

AGRADECIMENTOS

À Deus, por ter me dado vida e me capacitado a cada dia para seguir em frente.

À minha família por todo suporte e apoio.

Aos meus professores *Murilo, Edson, João e Mario* por terem buscado sempre a excelência em inúmeras horas de orientação.

À Universidade Federal do ABC e à University of Calgary por terem me acolhido como aluno e pesquisador.

Agradeço também aos meus amigos pesquisadores *Diogo, Edgard, Allan, Júlio, Clarissa, Roberta e Felipe* pelas infindáveis e valiosas contribuições pessoais e profissionais.

Agradeço com poucas palavras mas com muita relevância, em especial, à *Sicília e Patrícia*, amigas e pesquisadoras, pelo incentivo, amizade e apoio imensurável durante o meu tempo na University of Calgary.

Suba o primeiro degrau com fé.

Não é necessário que você veja toda a escada.

Apenas dê o primeiro passo.

— Martin Luther King

ABSTRACT

To allow non-professionals or beginners to quickly create models from a set of 2D drawings, 3D reconstruction has gained attention in recent years. However, when dealing with drawings, it is necessary to treat important problems such as ambiguity and difficulty in inferring hidden parts. This is especially valid when describing characteristics of plant and animal species as explored by botanists and zoologists, where the traditional illustration approach includes inferring or discarding occluded parts of the specimen being depicted. This thesis aims to reconstruct 3D objects from single-view 2D drawings, assuming that a great class of objects presents structural symmetry or consists of multiples objects drawn as overlaid structures.

In this work, we present two frameworks, an automatic approach that leads the study and development of a robust method for the generation of 3D models using Radial Basis Functions from Hermitian data allowing to interpolate the contours of the 2D drawing with the estimated 3D normals. Despite the promising results, the use of skeletons in previous works discards details present in the drawings, creating generalized cylinders around the medial axis extracted from the drawn parts. Also, a relevant part of the study is the development of new strategies for the segmentation of the drawings allowing the treatment of their different parts to be reconstructed with greater flexibility.

Furthermore, we present an interactive framework of a sketch-based interface for modeling 3D objects with multiple contours as overlaid structures. Inspired by traditional illustration strategies and scientific drawings, our sketching interface allows the user to infer perceivable symmetries and occluded parts of the model before its automatic 3D modeling. We propose a set of 2D visual effects to enhance the visual perception of users while sketching multiple overlaid objects in a single view. Examples with drawings ranging from simple line overlapping to stripes twists illustrations and biological systematics demonstrate the capabilities of our framework.

Keywords: Sketch-based interfaces and modeling (SBIM), HRBF Implicits, Non-photorealistic Rendering (NPR), Traditional scientific illustration principles and practices, Depth perception, Depth overlay

RESUMO

Para permitir que não profissionais ou iniciantes criem modelos rapidamente a partir de um conjunto de desenhos 2D, a reconstrução 3D tem ganhado atenção nos últimos anos. No entanto, ao lidar com desenhos, é necessário tratar problemas importantes como ambigüidade e dificuldade em inferir partes ocultas. Isso é especialmente válido ao descrever características de espécies de plantas e animais exploradas por botânicos e zoólogos, onde a abordagem tradicional da ilustração inclui inferir ou descartar partes oclusas do espécime retratado. Esta tese tem como objetivo reconstruir objetos 3D a partir de desenhos 2D em vista única, assumindo que uma grande classe de objetos apresenta simetria estrutural ou consiste em objetos múltiplos desenhados como estruturas sobrepostas.

Neste trabalho, apresentamos dois frameworks, uma abordagem automática que leva ao estudo e desenvolvimento de um método robusto para a geração de modelos 3D usando Funções de Base Radial a partir de dados Hermitianos, permitindo interpolar os contornos do desenho 2D com as normais 3D estimadas. Apesar dos resultados promissores, o uso de esqueletos em trabalhos anteriores descarta os detalhes presentes nos desenhos, criando cilindros generalizados em torno do eixo medial extraído das peças desenhadas. Além disso, como parte relevante do estudo, está o desenvolvimento de novas estratégias para a segmentação dos desenhos, permitindo tratar diferentes partes com maior flexibilidade.

Além disso, apresentamos um framework interativo de interface baseada em sketches para modelar objetos 3D com vários contornos como estruturas sobrepostas. Inspirado por estratégias tradicionais de ilustração e desenhos científicos, nossa interface de esboço permite ao usuário inferir simetrias perceptíveis e partes ocluídas do modelo antes de sua modelagem 3D automática. Propomos um conjunto de efeitos visuais 2D para melhorar a percepção visual dos usuários enquanto esboçam vários objetos sobrepostos em uma única vista. Exemplos com desenhos que variam de simples sobreposição de linha a ilustrações de torções de fitas e sistemática biológica demonstram as capacidades do nosso sistema.

Palavras-chave: Modelagem e interfaces baseadas em Sketches (SBIM), HRBF Implícitas, Rendering Não-fotorealístico (NPR), Princípios e práticas da ilustração científica tradicional, Percepção de Profundidade, Sobreposição de Profundidade

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

This thesis investigates novel alternatives for Sketch-Based Interfaces & Modeling (SBIM) proposing: *i*) an automatic reconstruction method to improve the fidelity of the 3D models with the 2D drawings as well as to improve the classification and segmentation of the different parts in drawings; *ii*) an interactive framework that includes 2D representation and 3D reconstruction objects in a single-view perspective, as well as, overlaying objects, providing support to layering and real-time feedback for users as illustrators and scientists.

Sketches, drawings, or designs have been used since the earliest days of humanity, usually designed to represent ideas, to materialize thoughts, to demonstrate or exemplify situations, projections, and to document abstract plans or objectives.

Several applications in science, medicine, engineering, and content creation require domain-valid interpretations, modeling and visualization of complex structures. Modeling such structures can be very challenging as they contain geometries and properties that need to be captured in 3D at multiple scales of observations and measurements. The wide range of model conceptualizations and their uncertainties are traditionally depicted and explored in the form of hand-drawn sketches and illustrations. These traditional sketches (Fig. 1) provide rich visual references for helping users formulate and represent conceptualizations based on data acquisition, laboratory, field observations and interpretations.

Thus, development communities began to build systems based on traditional drawing techniques that could be used for modeling, animation, and representation of objects and shapes, creating the concept of sketch-based modeling, exemplified in Fig. 2.

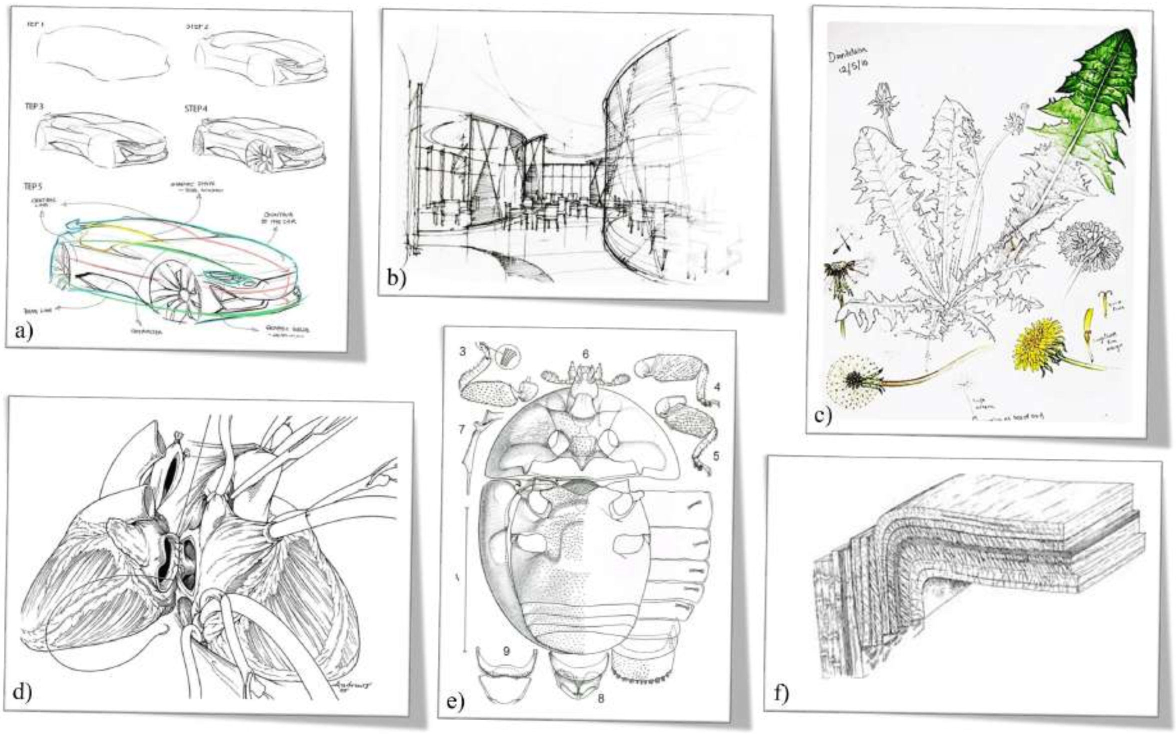


Figure 1: Traditional sketches applied in (a) engineering, (b) architecture, (c) botany, (d) medicine, (e) zoology and (f) geology. Sources: (a) Garnier [37], (b) Hertz [55], (c) Harper [54], (d) Andrews [6], (e) Bowestead and Eccles [15], and (f) Kruhl [76].

Many computational methods proposed to assist the user in this process have their foundations in the field of SBIM [64, 97]. This field introduced a new paradigm that leverages the traditional illustrator's drawing and rendering skills, allowing users to build 3D models more intuitively and rapidly.

Despite the advance in recent years, current modeling workflows lack or have limited number of frameworks enabling users to interactively construct 3D digital models seamlessly and directly from hand-drawn sketches using, for example, a pen and a tablet.

SBIM techniques and systems should 'mimic' the way domain users produce sketches, enforcing algorithmic rules to ensure domain-sound models are generated, and supporting

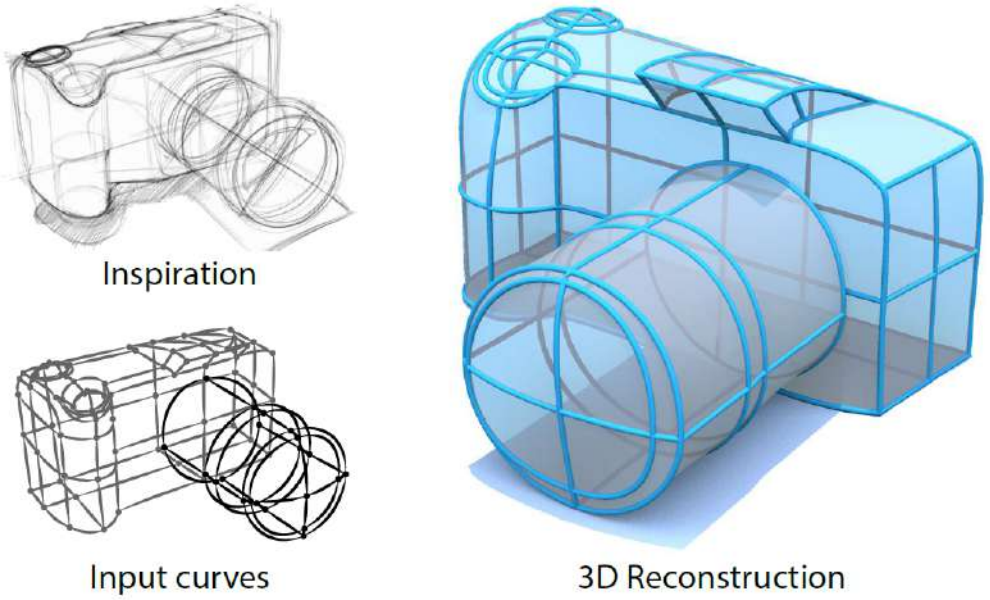


Figure 2: Sketch-Based Modeling. Source: Xu et al. [124].

different scales of geometries being observed and interpreted. The main problem that modeling needs to deal with is that a drawing can contain numerous interpretations [57]. Even if the dots and dashes were correctly designed, the sketch can represent any subset of objects that are drawn in the same way depending on the viewer's perspective [97], as shown in Figure 3.

Besides the interpretation, sketch-based systems present a series of limitations: the acquisition of the drawings, the computational representation, noises, smoothing of the traces, re-sampling, and the interpretation through an algorithm.

Despite efforts to mitigate these issues, recent research has shown that the approaches with the most effective results seek to reconstruct certain groups of objects or drawings [82]. Jayaraman et al. [63] create relief on two-sided objects based on the local geometry of the strokes, Shao et al. [109] creates perspectives using cross-lines to give a 3D appearance to objects, and together with regularities such as symmetry, planarity, and parallelism, Xu et al. [124] reconstruct these drawings creating three-dimensional models. Furthermore, other

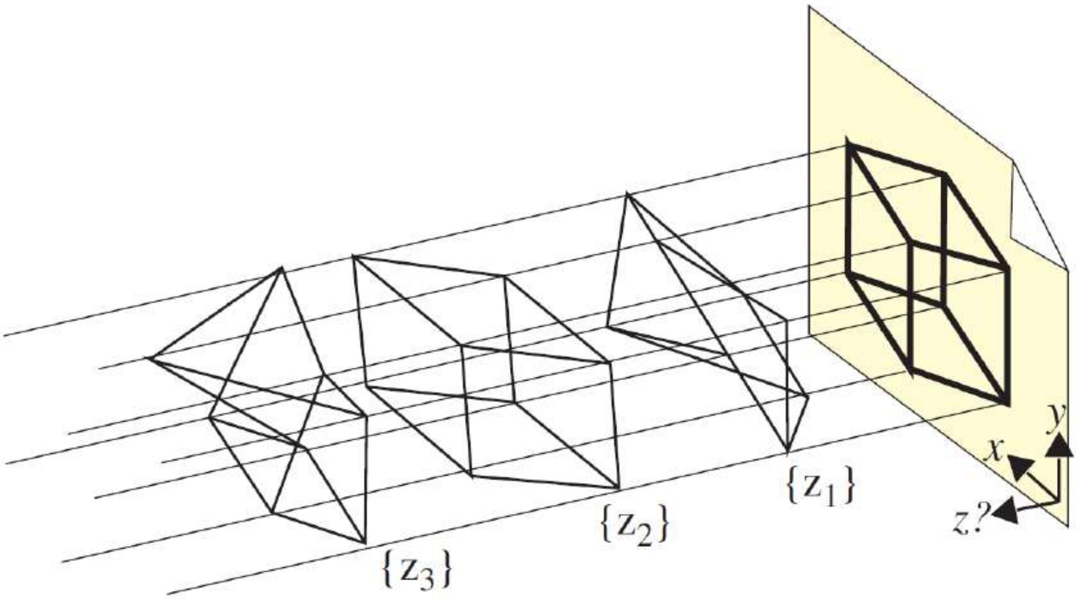


Figure 3: Ambiguous interpretation of an object. Source: Olsen et al. [97]

approaches directly focus on specific objects such as flowers and trees [62, 122], clothes [118], animals [31] and some arbitrary objects using the structural symmetry hypothesis [26, 17].

Even reconstructing specific objects from the structural symmetry hypothesis, the use of skeletons to eliminate problems with hidden parts implies a variety of simplifications in the objects to obtain plausible 3D models, discarding details present in the contours of the drawings.

Other subjects as biological models, such as insects and plants [4], require a more detailed description. Despite the technological evolution of 2D and 3D imaging commonly used in scientific descriptions, the demand for scientific drawings remains, in particular, to describing new specimen [105, 34]. Photographic cameras, scanning electron microscope, and other devices can solve part of the necessary task, but, some essential aspects of interpretive biology work still require a drawing.

Besides that, the aspect of description is particularly true for scientific illustrations as biological systematics, which is the field that studies and describes the living forms on Earth, both present and past [99, 58]. These complex scientific illustrations usually require the exploration of drawing elements, for instance, color palette, shadowing, hatching patterns or, stroke styles, allowing the illustrator to indicate depth, occlusion, textures, inner structures, or surface details.

The first study presented in this thesis proposes a reconstruction technique that considers the contours present in the drawings guaranteeing their shape in the final models. This work also aims to improve the robustness of previous techniques by proposing a new strategy that does not use skeletons. In addition, it proposes a new approach to the identification and classification of the different parts of a drawing allowing a greater number of objects to be reconstructed through the proposed method.

A second study presented in this thesis provides an easy and fast 2D representation and a reliable reconstruction of 3D objects. The presented framework combines 2D and 3D information to enhance the user experience and offers the possibility to complement works that infer depths or discard occluded parts. As an application, the framework was explored to model entomological features often present in species descriptions of the order Diptera, one of the mega-diverse group of insects popularly known as flies and mosquitoes, but the solutions are useful for dealing with any other biological group.

This chapter is organized as follows. Section 1.1 presents the main problems related to modeling based on sketches. Section 1.2 presents the objectives of this work and Section 1.3 defines the hypothesis investigated in this thesis. The contributions and publications of this work are shown in Sections 1.4 and 1.5. Finally, the Section 1.6 presents the organization of this thesis.

1.1 Motivation

Sketches and drawings are usually related to an object and can transform abstract ideas into concrete initial designs or represent our world as used in scientific illustrations.

The creation of designs is widely used in the early stages of a creative process where the overall shape and details are very vague. In contrast, scientific representations describe studied subjects relying on the specific depiction of parts and structures to elaborate documented drawings for species.

In both groups, sketches or drawings are usually related to an object, and, with the advent of the computational era, there is an interest in provide sketching and modeling systems to create feasible 3D models.

In recent years, many approaches have been proposed to overcome the limitations of sketch-based modeling. The work of Olsen et al. [97] presents several gaps to be investigated whereas the work of Cook and Agah [25] focuses in to explore the proposed methods of 3D reconstruction. Kazmi et al. [73] categorize the SBIM pipeline into sketch acquisition, sketch filtering and sketch interpretation while Ding and Liu [30] translate these categories into inputs, knowledge, and approach adding the category of output. However, the research presented by Olsen et al. [97] unravels the limitations into well-defined areas as follows.

GUI: Most of the Graphical User Interfaces (GUI) used are far from natural, many require the user to draw specifically to work properly which reduces immersion and ease of use. Besides that, the depth is not usually addressed properly to define a visual order of the subjects while the user is drawing.

Suggestive Lines: Traditional art employs hatching, doodling, dotting, shading, and suggestive contours to convey or give the appearance of 3D shapes, and when included, increase the potential for drastically improving modeling and reconstructed end objects.

Visual Memory and Visual Rules: Techniques for interpreting and identifying similarities between drawings and 3D shapes are still often ineffective substitutions of human visual memory. This is reflected in the problems of ambiguity in the drawings making it difficult or impossible to define depth and ambiguities in the occlusion of the parts without some form of visual memory, making the systems use the visual rules.

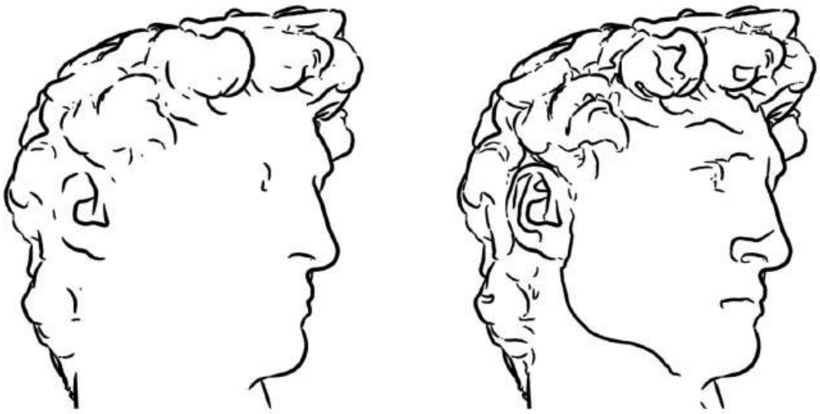
Quality of Models: Parametric or revolution surfaces can create high-quality results, however, the limited topology needs a precise alignment to create complex models. On the other hand, polygonal meshes do not have this limitation, however, the generation process is slow due to the minimization of the energies present in a mesh.

Accuracy: Accuracy is often reduced in favor of simplicity, particularly in free-drawing systems. It is possible to add a higher level of precision by specifying or inferring geometric constraints such as parallelism, perpendicularity, dimension fairness, and vertical or horizontal alignment. However, the greater the precision, the less freedom the user has concerning the drawing.

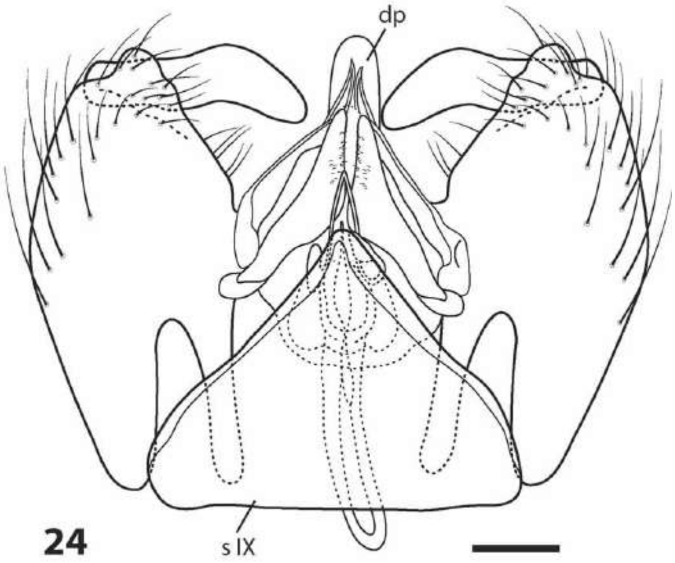
This thesis proposes frameworks to attenuate the limitations present in the Suggestive Lines, Visual Memory and Visual Rules, Accuracy as well as Graphical User Interfaces classes.

The presence of suggestive lines is found in several real situations, such as artistic drawings and scientific drawings (Figs. 4a and 4b).

Generally, these lines, give a 3D appearance to sketches or technical drawings, using painting techniques like shading and hatching or dotted lines and suggestive contours to indicate cuts and hidden parts.



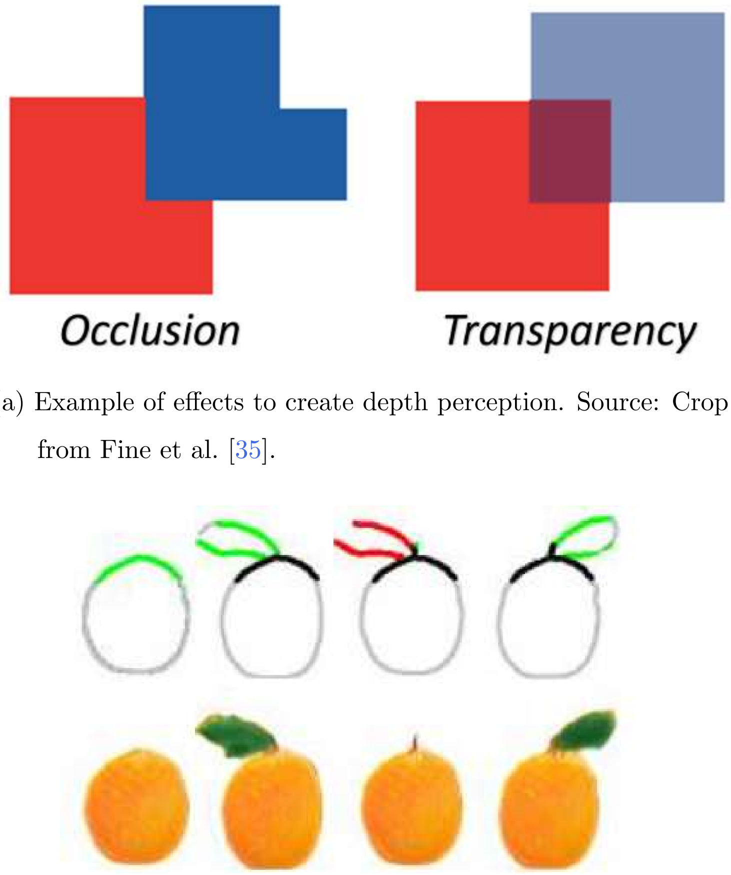
(a) Suggestive contours applied to a 3D model using non-photorealistic rendering (NPR) techniques. The image on left contains only contours while the image on right contains contours and suggestive contours. Source: DeCarlo et al. [29].



(b) Hand-drawn dashed lines as hidden-contours for describing a *Terminalia*. Source: Santos [105].

Figure 4: Examples of suggestive contours employed to artistic and scientific drawings.

When dealing with complex drawings, the interface can aid the user to retrieve information from their visual memory, by having enhanced visual effects that explicitly enhance hidden contours for sketch-based modeling, as shown in Fig. 5, and, can provide actions to be performed on strokes, thus, improving the creative process of drawing and modeling.



(a) Example of effects to create depth perception. Source: Crop from Fine et al. [35].

(b) Example of sketch inference and respective suggested final image. Source: Crop from Ghosh et al. [39].

Figure 5: Examples of real-time user feedback to create complex drawings.

Several studies show that the use of these drawing techniques can improve the classification and segmentation of the sketches [5, 63]. These improvements allow capturing a greater level of detail present in the sketches, avoiding the simplification and generalization of the models [13, 82, 126].

Another limitation arises when dealing with precision problems, *i.e.*, the fidelity of reconstructed 3D models to the traces present in the input sketches leading to the quality and fidelity of the final models compared with the input drawings.

Among the previously proposed approaches, the use of skeletons with revolution surfaces and approximations have been widely used in recent years. However, these methods do not interpolate contours by discarding details in input sketches or simplifying strokes on drawings.

1.2 Objectives

Our research guided this study into the following goals:

1. Investigate automatic 3D reconstruction methods from single-view sketches, proposing segmentation strategies capable of using suggestive contours and indicative lines increasing the number of shapes, bodies, and details that can be reconstructed.

2. Propose a framework for Sketch-Based Systems & Modeling (SBIM) that includes 2D representation of overlaying objects and 3D reconstruction for different contour categories in a single-view perspective, providing support to 3D layering and real-time feedback for users as illustrators and scientists.

3. As an application, explore the proposed framework to model entomological features often present in species descriptions of the order Diptera, one of the mega-diverse group of insects popularly known as flies and mosquitoes.

This thesis aims to obtain results using different interpolation techniques of the contours in the final mesh, combining previous methods and new techniques that enhance the preservation details on the contours into the final 3D models, attenuating the gaps of the techniques that use skeletons with revolution surfaces.

In order to use the feature contours and indicative lines in the drawings – i.e.., describing the shape characteristics of a 3D model (e.g., silhouette, boundaries, folding regions, curvatures, among others). Some techniques previously proposed were improved to include a greater amount of detail in objects and shapes, recognizing reliefs and slopes [82] and organic forms as animals [31].

To deal with the interpolation of the contours in the final mesh, efficient approaches are in the possibility of incorporating the methods to estimate normal 3D based on the transformation of the distance from the contours to the interior of the parts (*Normal Mapping*) [41, 91, 124, 109], as well as sample points in the contours to ensure trait characteristics are preserved, thus generating Hermitian data to be interpolated by Radial Basis Functions [44, 86]. Another efficient approach, the Rotational Blending Surface, was proven to be faster and better on geometric forms that do not contain branches or ramifications [4, 24].

To validate the proposed approaches, the final 3D models obtained as results, contains perceptual characteristics in the drawings preserving details guaranteed by the interpolation of the 2D contours as well as perceptual comparison with results obtained in previous works. Whereas, the framework design proposed through a Sketch-Based System is based on a set of drawing contours that match depth effects explored by the scientific community, and, allow highlighting details of the drawn objects and depicting overlays. As discussed, this is especially relevant for systematists working with species descriptions and biological illustrations of organisms and their parts.

1.3 Hypotheses

Considering the current limitations and the possibility of improvements in the 3D reconstruction of 2D drawings, this work intends to investigate four main hypotheses.

These hypotheses aim to improve the classification and segmentation of the parts, and the precision of the final models when compared to the drawings. Besides that, aims to contribute to attenuate challenges of scientific illustration, preserving the geometrical features of the biological structures when dealing with drawings representing both external and internal morphology.

The hypotheses are:

- **Work on a specific family of objects or surfaces.** Several works present tools for 3D drawings and reconstructions for different types of geometries and different classes of objects and artists, however, recent research shows that sketch-based modeling methods are more effective when dealing with a specific family of objects or surfaces [82].

- **Make use of the estimated proportions present in the sketches.** The proportion refers to a particular relationship between the parties that maintain an order capable of being specified among themselves. When we observe nature, we can perceive clear situations of symmetry, and therefore of proportion. This circumstance has been studied and has given rise to some theories and mathematical relationships that may further highlight the notion of proportion in drawings.

- **Curves and Indicative Lines.** Artists often draw cue lines to indicate smooth surfaces in the drawings. While the sketches are basically created from a sparse set of lines however sufficient to describe 3D shapes as long as the observer mentally extrapolates the curve lines to create a surface, assuming that the geometry of the curve represents the geometry of the surface. In addition, annotations are used to describe concave, convex, or flat surfaces as well as annotations in the contours describing pointed models, such as man-made objects, or soft as organic models.

- **Visual Perception Enhancements.** Layers can be considered as features of sketches. Thus, to enhance the perception of different layers in single-view drawings, effects can be applied for contours drawn in order to offer visual feedback for different parts of the object. These non-photorealistic rendering (NPR) effects are important, especially on single-view drawings, where the depth perception is a desirable feature.

1.4 Contributions

Our research has led to different contributions towards automatic and user-driven sketch-based modeling. These contributions are part of two frameworks, the former, introduce an improvement to automatic 3D modeling of single-view sketches and, the latter, present a framework of Sketch-Based Systems & Modeling for the reconstruction of complex 3D objects drawn as overlaid structures using different types of contours.

We now briefly present the contributions in their order in this thesis.

Contour-Aware 3D Reconstruction of Side-View Sketches (Chapter 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;
- the flexibility to flatten or round shapes to create plausible final models.

Sketch-based modeling supported by 2D visual perception enhancements: application and analysis for illustrations of systematic biology (Chapter 4).

The specific contributions of this work are:

- a novel framework for overlaying contours based on illustration techniques;
- a set of visual effects that aid the user during the creation process of complex and layered objects;
- the assembly of 3D reconstruction methods that support different types of contours;
- a novel method for sketch-based modeling of strings and stripes using few interactions;
- an application and analysis for biological systematic illustrations evaluated by an expert in Diptera species.

1.5 Publications

Ramos, S., Gois, J. P. (2017). Reconstrução 3D de Sketches em Vista Lateral. I Workshop @NUVEM - UFABC, Santo André - Brasil.

Ramos, S., Trevisan, D. F., Batagelo, H. C., Costa Sousa, M. & Gois, J. P. (2018). Contour-aware 3D reconstruction of side-view sketches. Computers & Graphics, 77, 97-107.

1.6 Outline

The organization of this thesis is as follows.

- In Chapter 2, we present the related studies that inspired our approaches in this well-researched area known as Sketch-Based Systems & Modeling (SBIM);

- In Chapter **??**, an overview of the proposed frameworks is presented as well as the research context and the main contributions of the proposed frameworks;

- In Chapter 3, we present an automatic approach for 3D modeling of single-view sketches. We detail the computational characteristics, and present results, comparing with previous approaches;

- In Chapter 4, we present a Sketch-Based System for the reconstruction of complex 3D objects. We describe and discuss the proposed visual enhancements, as well as the different categories of contours supported and present the resulting models created through our system;

- In Chapter 5, our conclusion, as well as, the addressed limitations of the proposed approaches are described. Besides that, future directions are listed in order to illustrate possible future works.

2 | RELATED WORK

Sketch-Based Systems & Modeling (SBIM) can be categorized as data-driven systems, multi-view systems, and single-view systems. This chapter details previous works presenting sketch-based modeling techniques and approaches.

2.1 Data-driven Systems

These 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 that fit the image or sketch. These methods are similar to shape matching methods that look for 3D models that fit into 2D drawings.

In contrast with the proposed frameworks in this thesis, data-driven systems are limited to the availability and amount of data in data-sets. Models are obtained from a 3D database and scaled [125] or deformed [75] to fit the input sketch. More recently, approaches using neural networks to match sketches and models in a retriever system or apply operations to model the final 3D shapes were proposed [84, 120]. Other sketch-based systems propose sketch-to-image results by synthesizing images [23], categorizing drawings [39] or approximating face images to the input contours [22].

Sketch-based Modeling of Parameterized Objects

Yang et al. [125] present a sketch recognition algorithm that matches the points and curves of a set of given 2D templates to the sketch, shown in Fig. 6. This algorithm, instead of combining 3D models with sketches, uses the best-fit template to retrieve objects from a 3D database or performing a procedural construction. Although sketches can be drawn in two views to allow more parts to be recognized and modeled. The models are obtained from a 3D database, and even scaling and allocating the models, the system works only for 3 classes of objects: cups, airplanes, and fishes, limiting the shapes and drawings that the user can reconstruct.

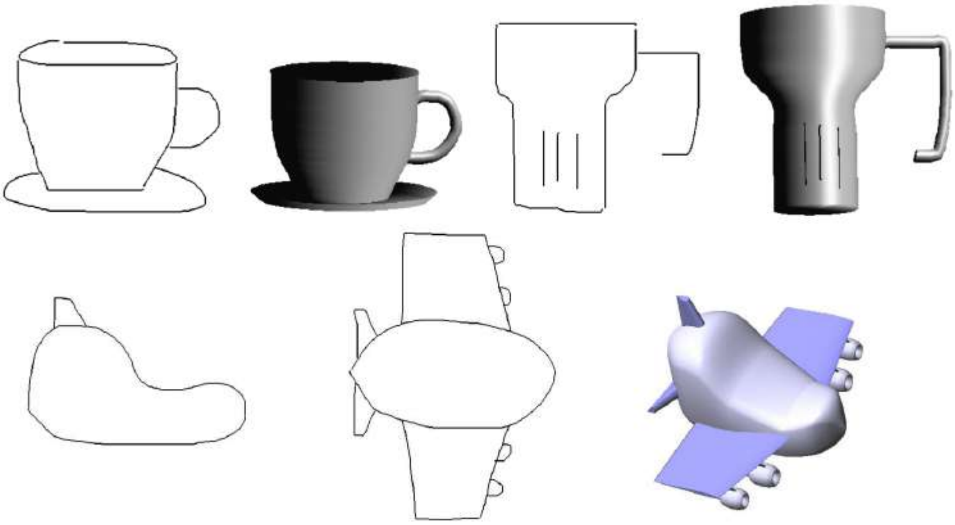


Figure 6: Examples of input sketches and output 3D models. Source: Yang et al. [125].

Modeling from contour drawings

The algorithm introduced by Kraevoy et al. [75] uses a contour drawing and a 3D template as inputs. These inputs are combined and aligned setting a common view direction and scale for the contours and 3D template. Their algorithm uses the contours to deform the template model, as shown in Fig. 7. By using templates as inputs, the symmetrical parts need not be modeled, just as the hidden parts need not be inferred.

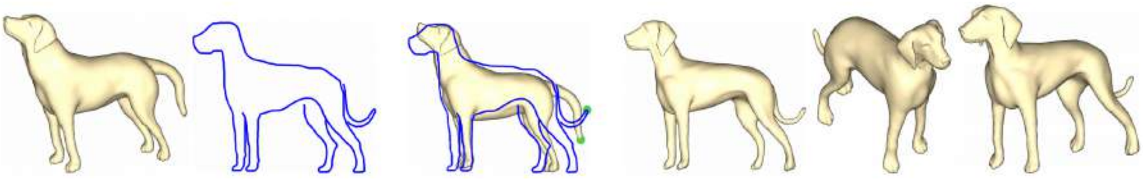


Figure 7: Contour and 3D template used as input. Source: Kraevoy et al. [75]

Sketch-based 3D Shape Retrieval using Convolutional Neural Networks

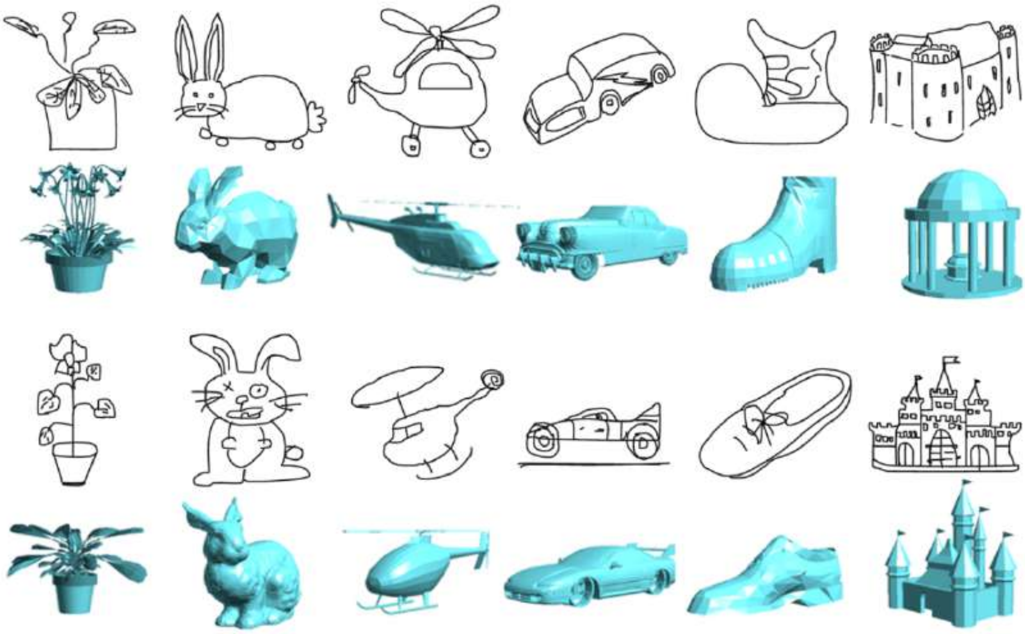


Figure 8: Examples of sketch-based 3D shape retrieval. Source: Wang et al. [120].

Wang et al. [120] propose algorithms and methods to retrieve 3D models in a database corresponding to 2D sketches training two Siamese Convolutional Neural Networks (CNNs) which consists of two identical sub-convolutional networks, one used for two input sketches and the other used for two input views. The models in the database are used to generate

2D line drawings, then the neural network matches corresponding sketches and views to 2D line drawings extracted from 3D models, returning as a final result of the original 3D model of the drawing in the database (Fig. 8).

3D shape reconstruction from sketches via multi-view convolutional networks

Using single or multiple sketches, Lun et al. [84] use multi-view convolutional networks to reconstruct 3D shapes by creating 3D point clouds starting from a trained algorithm to map sketches to 3D meshes. The Figure 9 shows a gallery of results. The blue shapes are final results produced using the input sketches, and the orange shapes are the nearest in the training databases obtained via sketch-based retrieval.

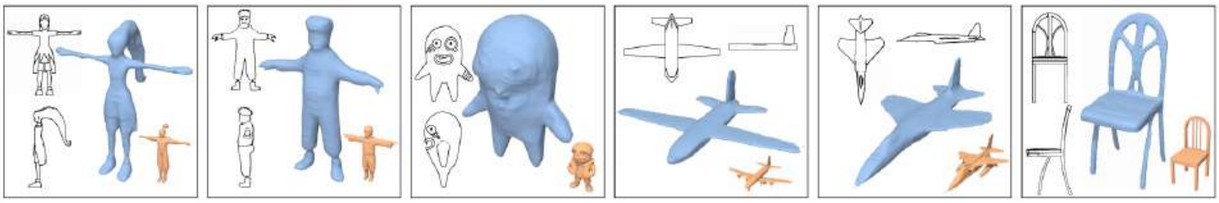


Figure 9: 3D models reconstruction from Sketches via Multi-view CNN Networks. Source: Lun et al. [84].

Sketchygan: Towards diverse and realistic sketch to image synthesis

Chen and Hays [23] present a sketch-based system along with Generative Adversarial Network (GAN) approach that synthesizes plausible images from 50 categories, e.g., motorcycles, horses, and couches. The input to the system is a sketch illustrating an object and the output is a realistic image containing it in a similar position as Fig. 10 illustrate. Some limitations are addressed in this work: a) results generally aren't photorealistic; b) overly faithful sketches can compromise the output images realism; c) *SketchyGAN* fails due to lack of sketch-photo training pairs, mainly due to the difficulty to depict the *human intent* when the input sketch is too literally.

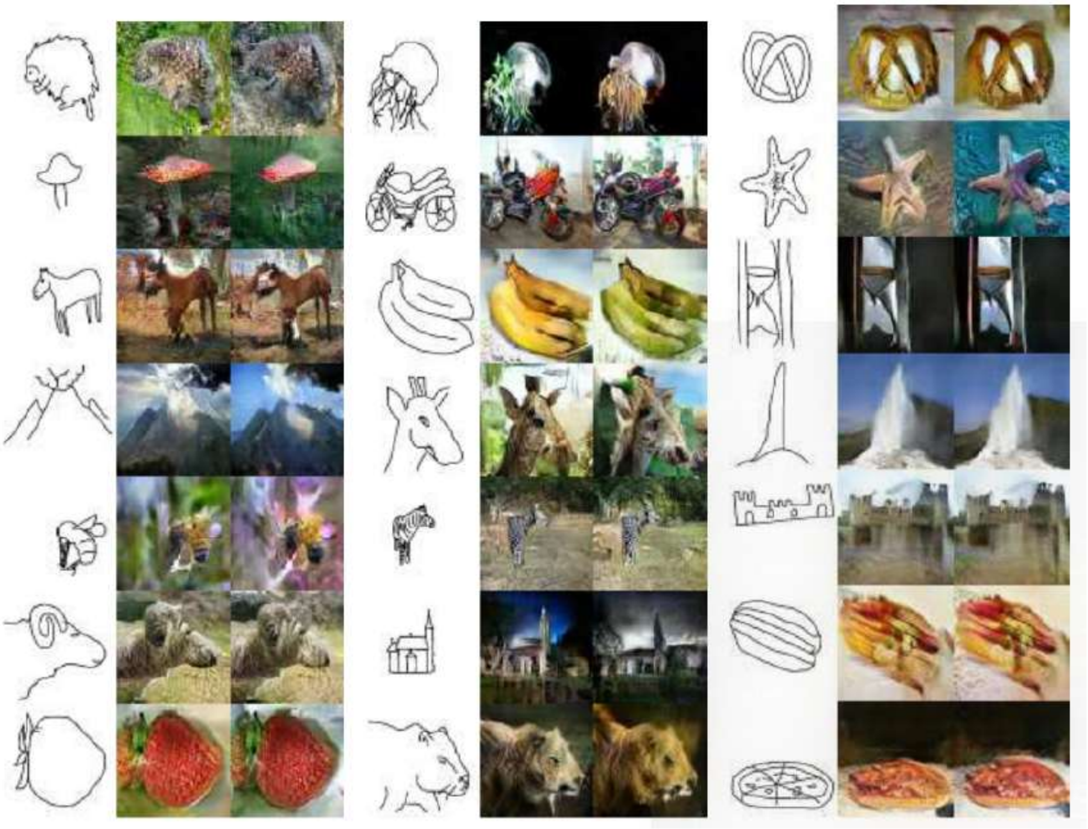


Figure 10: Output images from SketchyGAN system. A pair of output images is shown for each sketch used as input. Source: Chen and Hays [23].

Interactive sketch & fill: Multiclass sketch-to-image translation

Ghosh et al. [39] present a class-conditioned outline-to-image method to retrieve results from a data-set. The process starts with an object category and a sparse sketch, then, plausible completions are recommended by the system while a corresponding synthesized image is suggested. The proposed system is based on a feedback loop where the GAN-based network synthesizes the images while the user edits the sketch interactively, using the network recommendation as shown in Fig. 11.

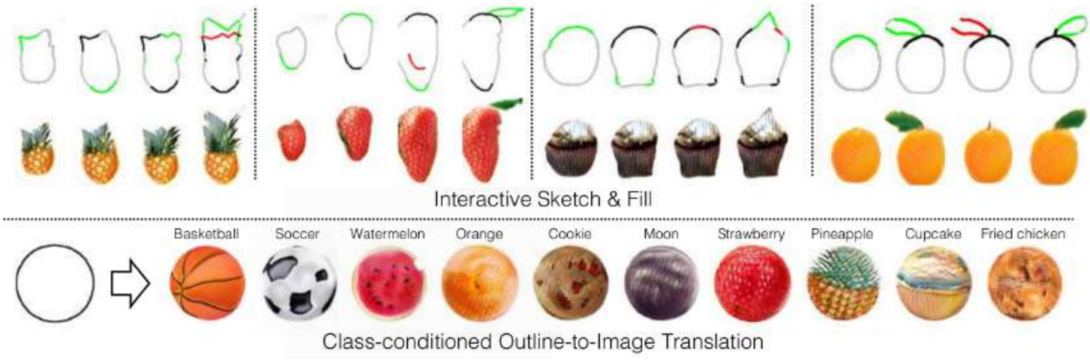


Figure 11: The proposed system estimates the completions for sketched strokes according to the selected category. The system estimates the green lines while removing red strokes over time. The final generated image is shown as feedback to the user. Source: Ghosh et al. [39].

However, the proposed Generative Adversarial Network returns a set of images based on a previously acquired database. The concern around the databases is that, for specific areas such as biological and medical areas, the access sketches and images to train the networks is limited, or are nonexistent.

Deep generation of face images from sketches

More recently, the community have made efforts to present novel sketch-based approaches taking advantages by in recent advances of the machine learning techniques. Chen et al. [22] synthesize plausible face images by approximating input sketches on a local-to-global approach. In their system, named *DeepFaceDrawing*, uses a conditional Generative Adversarial Network [49] to retrieve and then, directly interpolate the face representations of the nearest neighbors of the query feeding the interpolation results to the subsequent generation process. While the approach outperforms existing sketch-to-image synthesis works, with improved flexibility and usability, the final result is limited to images generated by the sketching system.

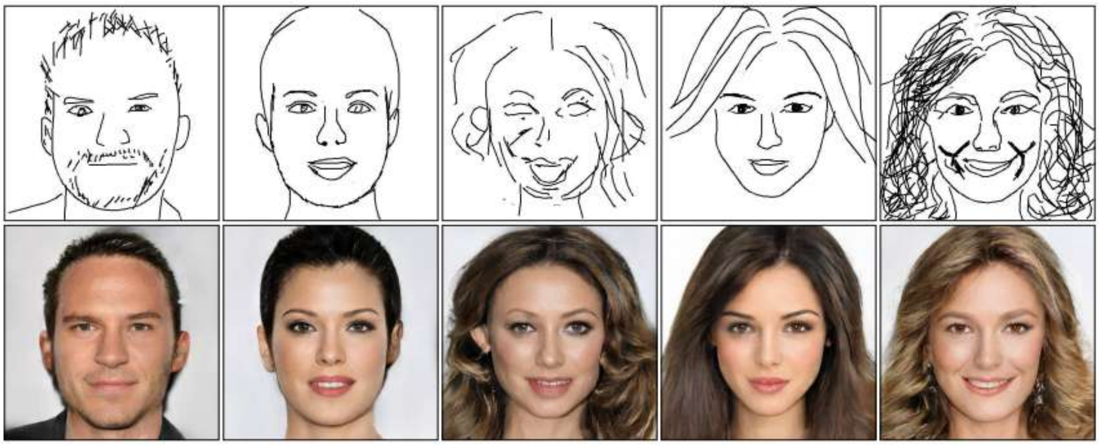


Figure 12: DeepFaceDrawing system allows users with little training in drawing to produce high-quality face images (Bottom) from rough or even incomplete freehand sketches (Top). Source: Chen et al. [22].

2.2 Multi-view Systems

These systems are generally employed to generate 3D models from two or three different views represented by sketches. Multi-view approaches can be categorized as automatic or interactive approaches. Automatic approaches uses orthographic views together with annotations or Gestalt principles [110, 40, 66, 130] or combines skeletons and avatars with drawings to reconstruct 3D models [81, 28]. Whereas interactive approaches require user intervention and expertise to progressively create 3D models [61, 82]. In contrast to this system category, our frameworks enable users and illustrators to create 3D models from single-view inputs, reducing the efforts to properly combine multiple views or sequentially create and refine parts through sketch interactions.

Fast 3D solid model reconstruction from orthographic views

Shin and Shin [110] used a multi-view method that required consistent sketches of the object to be reconstructed, and Gestalt principles to eliminate ambiguities. This approach reconstructs drawings from three orthographic views splitting inputs in vertex, edges, and faces and classifies then using properties of technical drawings as shown in Fig. 13. However, this approach is limited to objects composed of arbitrary planar and limited quadric faces such as cylinders that are parallel to one of the principal axes.

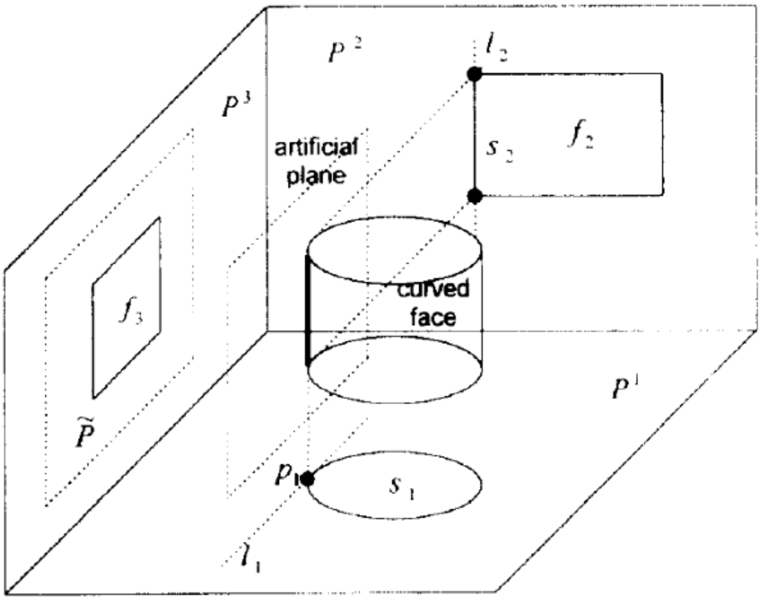
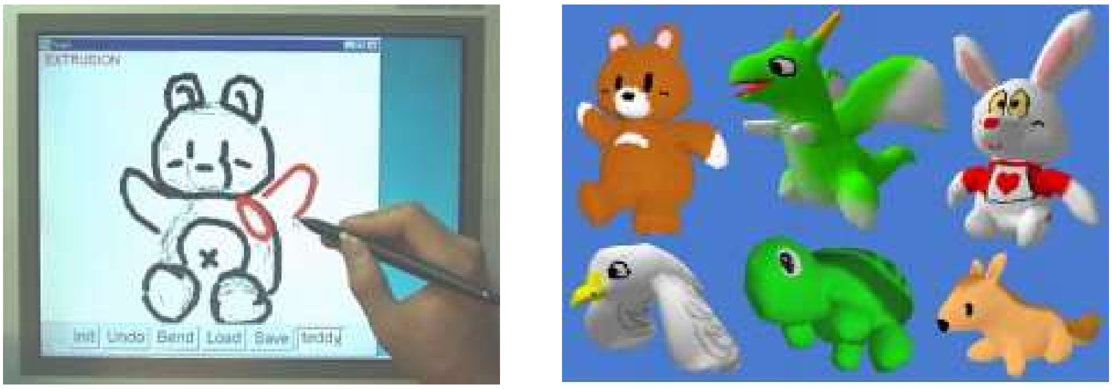


Figure 13: Properties of technical drawings. Source: Shin and Shin [110].

Teddy: a sketching interface for 3D freeform design

After the publication of the Teddy system [61], a range of methods based on user interaction has been proposed. Teddy, shown in Fig. 14, is one of the pioneering modeling interfaces based on sketches for free modeling plush animals, and other types of rounded objects. The user draws 2D dashes interactively on the screen and the system automatically builds 3D polygonal surfaces with clouds surrounded by silhouettes. Being interactive, Teddy also supports object insertion, cutting, extruding, and smoothing operations on the generated polygonal meshes. However, the interface may fail or generate failed results if the user draws unexpected traits. Also, it does not provide for more complex operations such as creating creases, folds, or kinks in the mesh generated.



(a) User Interface

(b) Final models generated by Teddy System.

Figure 14: Teddy System. Source: Igarashi et al. [61].

A Sketch-Based Interface for Detail-Preserving Mesh Editing

Nealen et al. [93] present a method for interactive editing of surface meshes by sketches. The system, exemplified in Fig. 15, allows the user to edit meshes by selecting and cropping surfaces allowing to smooth features and use suggestive contours to influence vertices and normals. Although this framework is an extension of Laplace/Poisson mesh modeling it can accommodate constraints on the normals and the curvatures, place constraints on virtual vertices to edit the mesh, and implements a tangential mesh regularization ensuring well-shaped triangles.

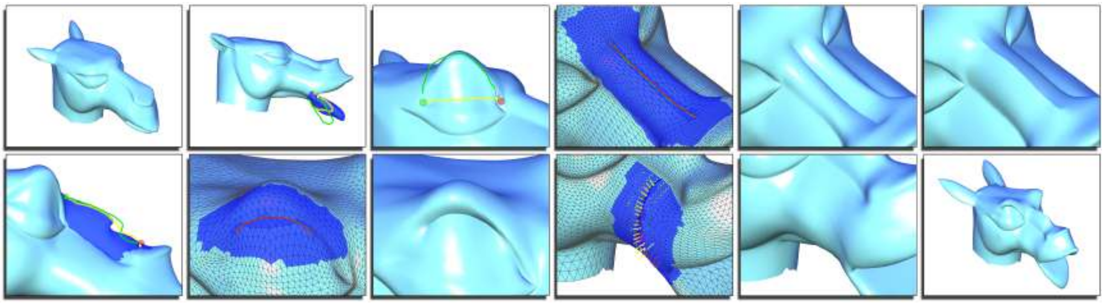


Figure 15: Sketch-Based Interface for Mesh-Editing. Source: Nealen et al. [93]

FiberMesh: Designing Freeform Surfaces with 3D Curves

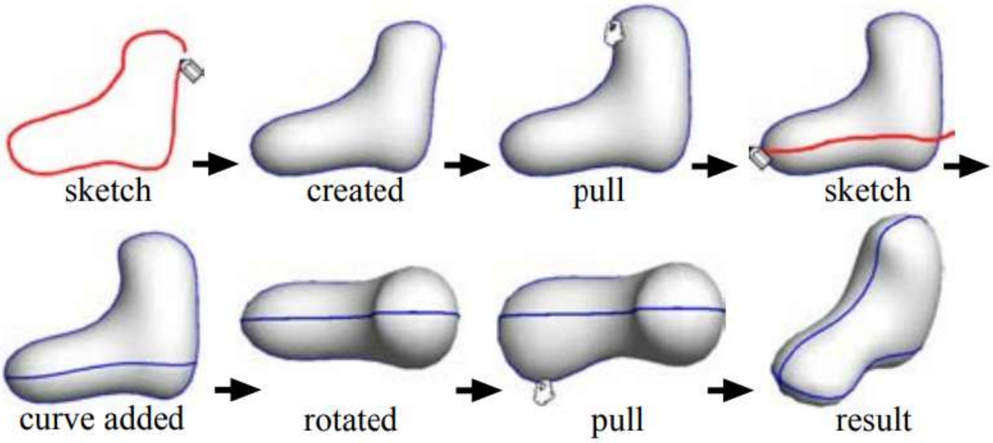


Figure 16: An example of modeling in FiberMesh system. Source: Nealen *et al.* [92].

Nealen et al. [92] extend the Teddy system shown by Igarashi et al. [61] from the user's point of view. The user interactively draws the silhouette of desired models and the system automatically constructs a surface in real-time (Fig. 16). The drawn lines remain on the models allowing the user to alter them to find the desired surface interactively, as well as allowing rows to be added or removed to create details on the final model. Thus, to create models with symmetrical or hidden parts, the user needs to rotate the models several times while drawing new parts not seen in a single view.

ILoveSketch: as-natural-as-possible sketching system for creating 3D curve models

Bae et al. [9] present ILoveSketch, shown in Fig. 17. A drawing system that captures the traits of a stylus and paper for professional designers that allows interaction directly with traces of 3D models. Thus, the system integrates interaction techniques based on sketches with several improvements and properties. Subsequently, their system was used for the elaboration of sketch-based modeling frameworks presented by Shao et al. [109] and Xu et al. [124].

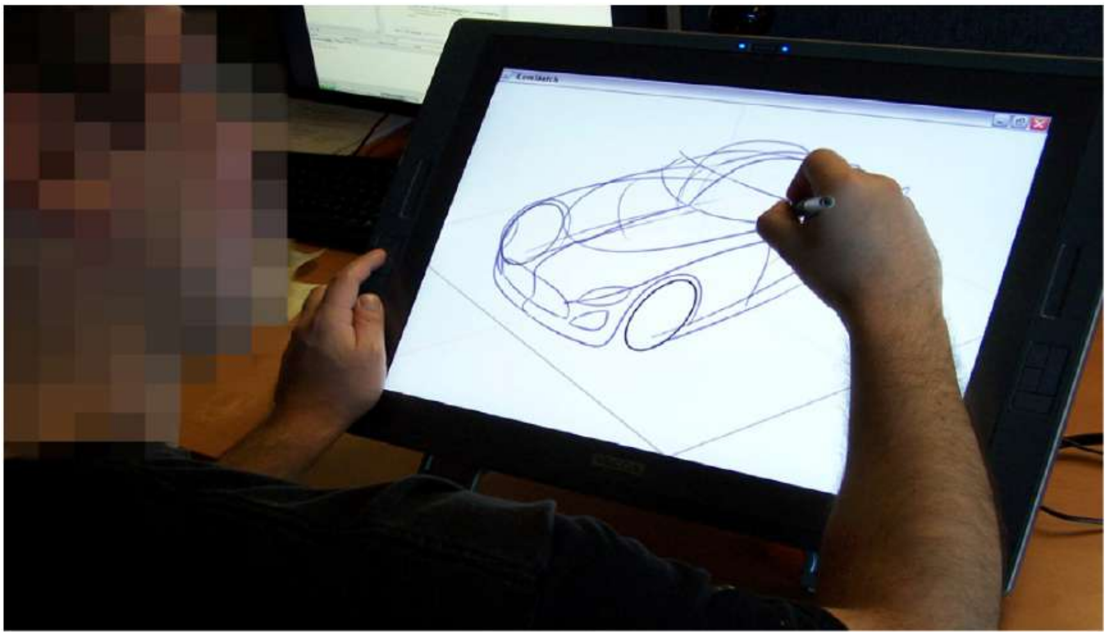


Figure 17: Artist using iLoveSketch on a touch-sensitive device. Source: [9].

Structured Annotations for 2D-to-3D Modeling

Gingold et al. [40] present a system for 3D modeling of free-form surfaces from 2D sketches. In their system, the user places primitives annotations on an image in a single view. Figure 18 illustrates the process. The modeling starts from a generalized cylinder defined by a 2D spine curve and attributes associated with cylinder cross-sections that define a shape, a part out-of-image-plane, scale, and symmetry director. The annotations used to combine primitives placed on an inconsistent 2D image involve connection curves, mirror annotations, alignment, tilts, and scales to attach bodies and different or symmetrical parts. However, the system does not support surfaces with edges or relatively flat surfaces discarding fine-details. To obtain plausible bodies the spines must be drawn in side-view in order to describe the bodies along with it. Besides that, does not allow cycles of connection curves failing in cases where one part overlaps others.

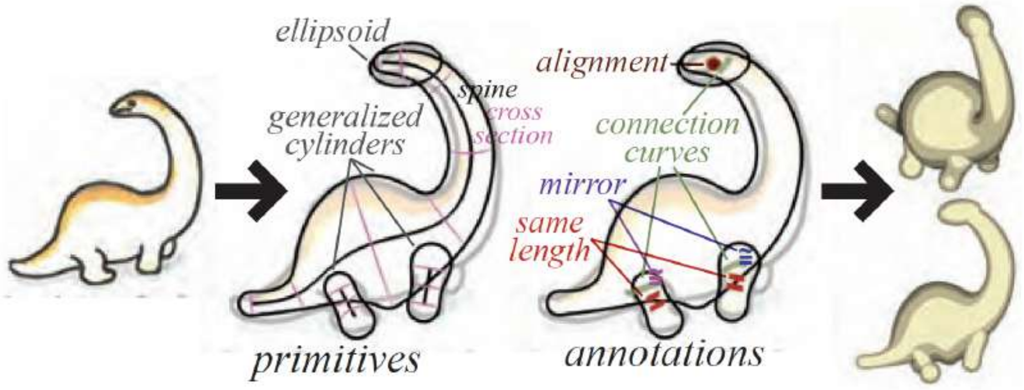
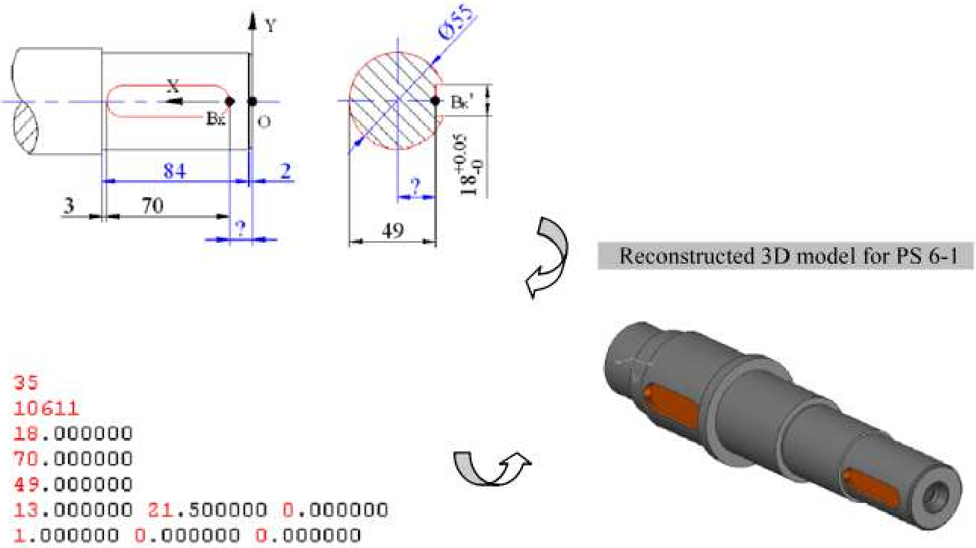


Figure 18: Process of 3D Modeling based in 2D Structured Annotations. Source: Gingold *et al.* [40].

Serial 3D model reconstruction for machining evolution of rotational parts by merging semantic and graphic process planning information

Zhang et al. [130] demonstrated the reconstruction of rotational parts based on technical drawings, this method follows a strategy driven by process planning and machining semantics to reconstruct parts. Using an incremental and knowledge-based process, the proposed method divides the reconstruction into several sub-stages and merges machining semantics like dimensions, projections, and orientation to interpret the 2D drawings, an example is shown in Fig. 19. This system requires a machining operation sheet and cannot be used on the design stage, unlike conventional reconstruction methods.



```
35
10611
18.000000
70.000000
49.000000
13.000000 21.500000 0.000000
1.000000 0.000000 0.000000
```

Figure 19: Reconstructed object using process planning and machining semantics. Source: Zhang et al. [130].

3D Modeling with Silhouettes

Rivers et al. [102] extended the concept of the visual hull and silhouette intersection to be combined using CSG (constructive solid geometry) operations. The system needs two or three views to reconstruct part's silhouettes once the silhouette in one view imposes constraints on the part on other views, shown in Fig. 20. For each silhouette, the algorithm calculates intersections extruding parts from the plane that were drawn, these intersections create facets that when combined assembles the final CSG solid's polyhedron. The main limitation of this system is with solids where facets do not share common axes, as is the case with organic shapes like clothes, sculptures, and animals.

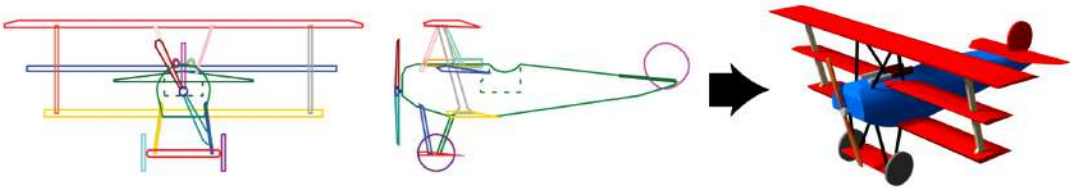


Figure 20: Silhouettes and final 3D model. Source:Rivers *et al.* [102].

ArtiSketch: A system for articulated sketch modeling

The work presented by Levi and Gotsman [81] allows the conversion of 2D content into 3D content by people who do not have artistic skills. ArtiSketch, whose process performed by the algorithm is presented in Fig. 21, automatically converts sketches into articulated 3D objects from different perspectives using interactive tools that drive the user to move the 3D skeleton in different poses, refining it for each *frame*.

According to the authors, the main limitation of the ArtiSketch is that the entry sketches must have a more rigid deformation unlike a soft body and the sketches should preserve the proportion as a normal skeleton. The ArtiSketch does not manipulate exaggerated deformations with unnatural movements as in cartoon characters that contract in stretch bodies without respecting physics.

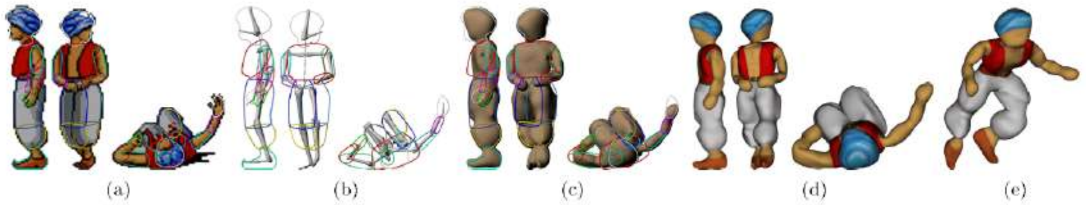


Figure 21: 2D to 3D conversion process from ArtiSketch. Source: Levi and Gotsman [81]

SecondSkin: sketch-based construction of layered 3D models

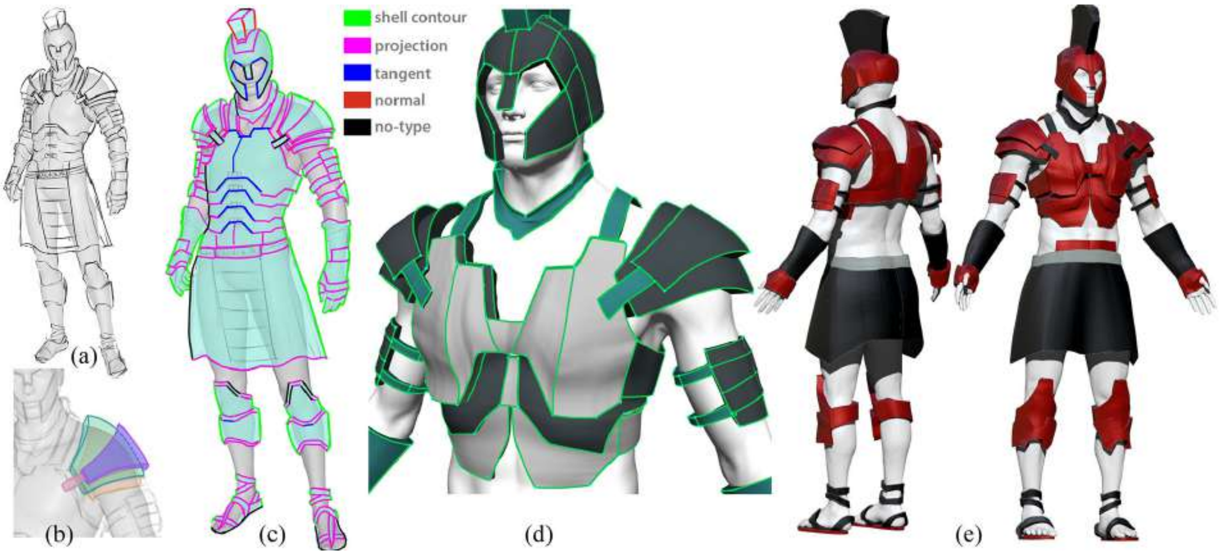


Figure 22: 2D strokes sketched on and around 3D geometry form the as input to SecondSkin (a-c). Classification of strokes based on the relationship between 2D strokes and underlying 3D geometry, generating 3D curves, surface patches, and volumes (d), final 3D model suitable for conceptual design (e). Source: De Paoli and Singh [28].

De Paoli and Singh [28] proposed a sketch-based modeling system for designing layered 3D solid models using existing geometry to interactively structure artistic 2D strokes. Then, the algorithm processes the strokes contextualizing the underlying 3D geometry to create layers over the objects as shown in Fig. 22.

Collaborative 3D Modeling by the Crowd

Suzuki and Igarashi [115] proposed a collaborative system so that non-professional workers could reconstruct various objects, an overview is given in Fig. 23. In their system, each worker draws a simple sketch that represents an orthographic view of the object. Then, the system classifies the sketches according to their dissimilarity, separating them in views. The 3D model is reconstructed using combined sketches from a primitive consensus algorithm for each view.

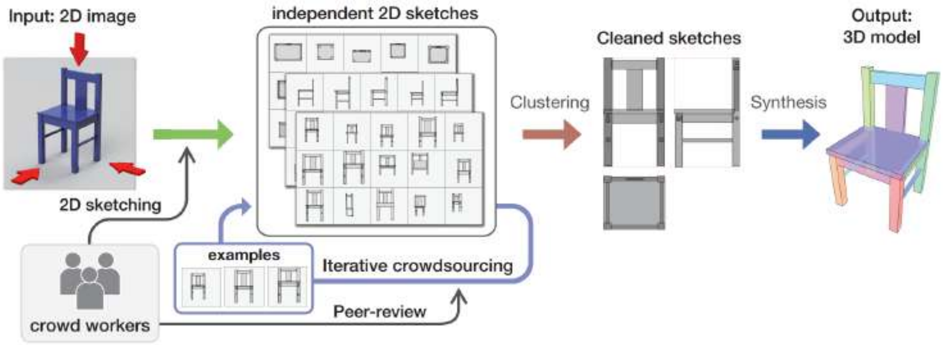


Figure 23: Overview of a collaborative 3D modeling system. Source: Suzuki and Igarashi [115].

BendSketch: modeling freeform surfaces through 2D sketching

BendSketch (Fig. 24), one of the most recent approaches shown by Li et al. [82], 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.

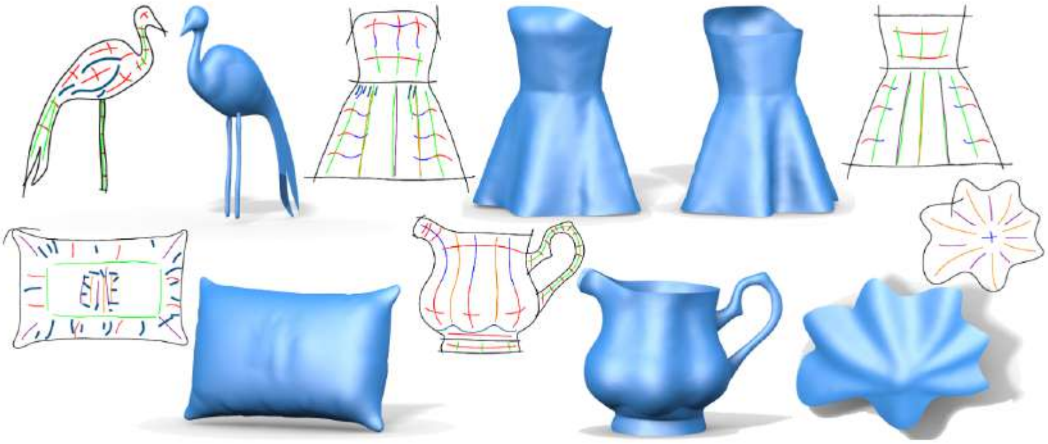


Figure 24: Freeform shapes created by BendSketch system. Source: Li *et al.* [82]

A 3D modeling methodology based on a concavity-aware geometric test to create 3D textured coarse models from concept art and orthographic projections

Junior et al. [66] create textured 3D meshes from the orthographic view of objects, generating a diffuse color map for texturing models together with a coarse 3D mesh. The system uses the Box Modeling technique to accelerate the 3D model production replacing the initial primitive mesh (Fig. 25). Although the consistency of geometry work properly for two depth levels, it fails when inferring depths when the subject consists of multiple levels, for instance, a step of a staircase that is occluded from a side-view perspective. Another addressed limitation is that the voxel-based carving lacks in precision to the corresponding pixels on input views creating some pixel misalignment.

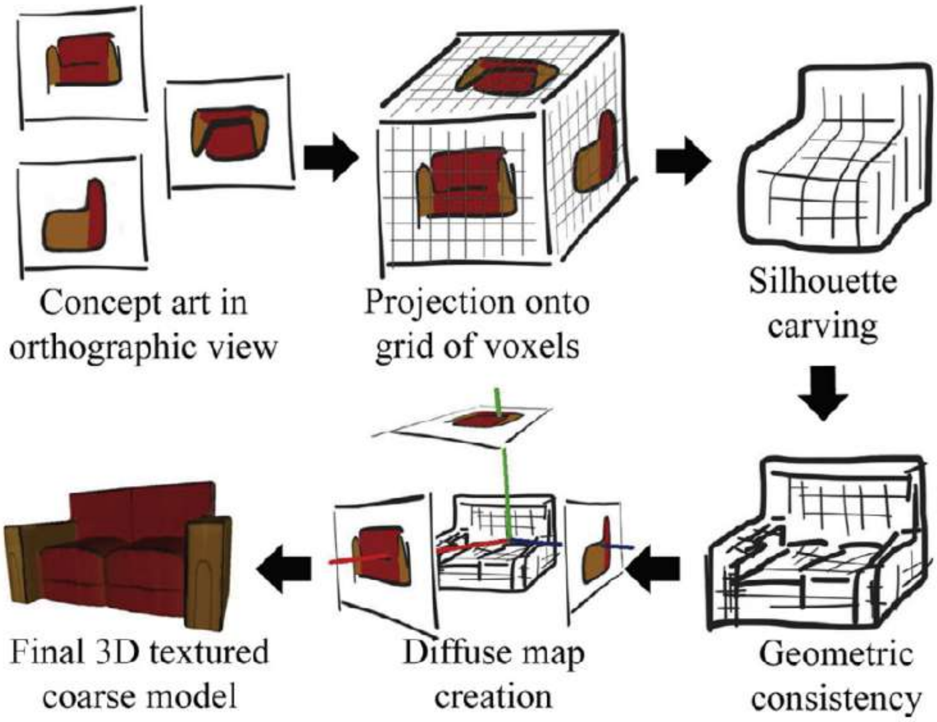

Concept art in orthographic view
Projection onto grid of voxels
Silhouette carving
Geometric consistency
Diffuse map creation
Final 3D textured coarse model

Figure 25: 3D modeling methodology based on a concavity-aware geometric test. Source: Junior et al. [66].

2.3 Single-view Systems

Single-view systems seek to interpret and infer the reconstruction of bodies from the silhouette and other feature strokes of sketch or a single image, imposing some constraints and mimicking the human ability to see 2D representations of 3D objects. To interpret and infer the reconstruction of shapes from silhouettes and other feature strokes, single-view 3D modeling methods have been developed creating feasible 3D models in inspired in scientific and artistic illustrations [24].

These works inspired this thesis approaches by presenting algorithms that estimate normal fields to given 3D appearance [109], classify and reconstruct objects with structural symmetry [31, 26], and the use of traditional illustration methods to reconstruct objects [24].

Sketch-based modeling with few strokes

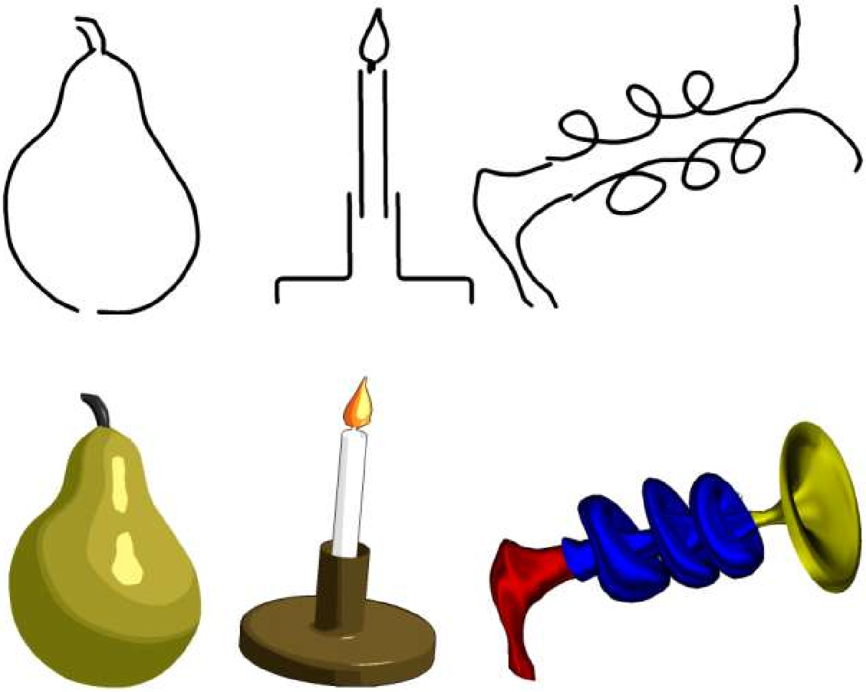


Figure 26: Shapes reconstructed using few strokes by using rotational blending surfaces. Source: Cherlin et al. [24].

Cherlin et al. [24] proposed a sketch-based system for free-form 3D modeling inspired on traditional illustration where sketches are progressively refined with strokes. Also, the reconstruction is performed while parametric surfaces are generated allowing orthogonal deformation and cross-sectional over sketching on 3D models. Still, the proposed sketch-based system lacks support for branching structures and complex objects such as biological models. Another limitation is the multiple operations as translation and rotation that could be improved by creating a rule-based solution for a single operation.

SmoothSketch: 3D free-form shapes from complex sketches

Karpenko and Hughes [68] proposed a system to infer 3D free-form shapes from sketches. This system uses cusps and T-junctions to infer solid shapes while completing hidden contours topologically and smooths embedding of shapes. The system allows to create 3D shapes for various classes of drawings by estimating the probability that they are connected through a hidden contour, thus completing contours for each pair of endpoints of the visible contour, a topological embedding is created to attach the meshes of individual panels into a single mesh. The probability is estimated by a local approach, *i.e.*, the completion of shape depends on the starting and ending points, ignoring the remainder of the input shape. Fig. 27 presents drawings and their respective reconstructed 3D models.

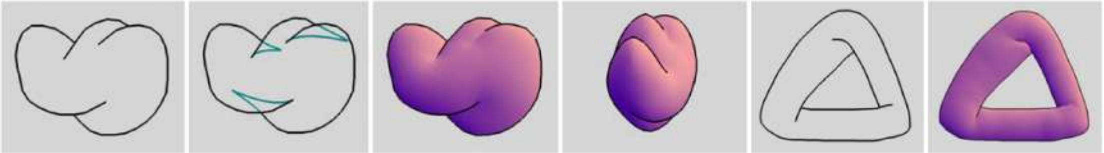


Figure 27: Models reconstructed by SmoothSketch system. Source: Karpenko and Hughes [68].

Automatic construction of 3D models from architectural line drawings

Lee et al. [79] introduce a framework that constructs curved and polygonal 3D models from 2D line drawings, such as architectural or mechanical drawings. The algorithm detects lines and shapes with junctions of types "L, A, Y, T" and curvatures that together with regularities like symmetry, parallelism, collinearity, and orthogonality determinate the shape of the 3D models of polyhedral objects and curved surfaces bounded by polygons, an example of junctions and curvatures are shown in Fig. 28.

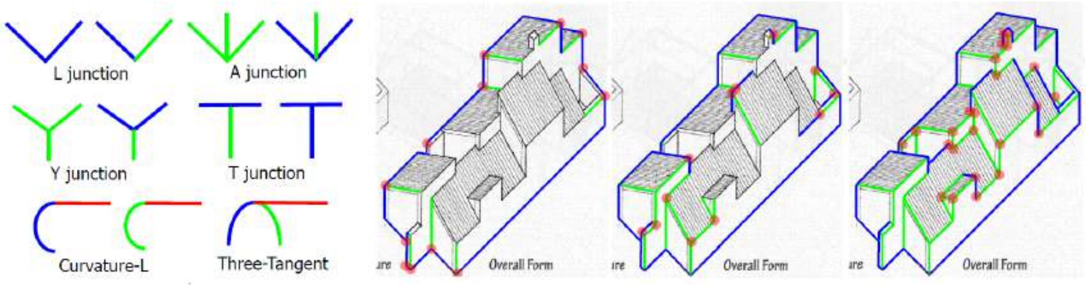


Figure 28: Examples of identified junctions and curvatures from 2D drawings. Source: Lee et al. [79].

Repoussé: Automatic Inflation of 2D Artwork

To automatically inflate artworks Joshi and Carr [65] presents the Repoussé, a system to create 3D shapes by inflating the surface that interpolates the input curves, handling smooth and sharp constraints by interpolation curves and local mean curvature control. Their framework makes a simple classification of the curves determining that traces away from the edges are holes and that traces near the edges of the letters create beveled effects in artworks. An example is shown in Fig. 29.

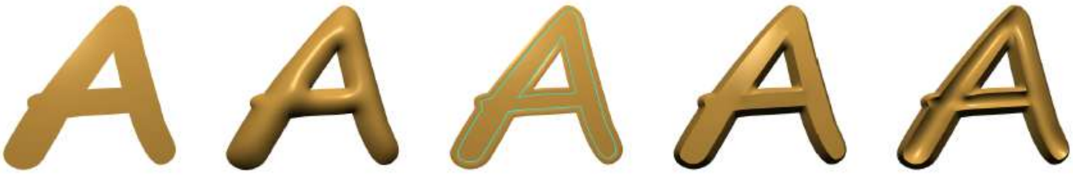


Figure 29: Example of Automatic Inflation of 2D Artwork. Source: Joshi and Carr [65].

Matisse: Painting 2D regions for Modeling Free-Form Shapes

Matisse converts 2D painted regions into 3D shapes using a variant of implicit modeling that fits convolution surfaces to regions [12]. Instead of using sketches, the user interactively paints these 2D regions with different sizes of a brush while fill desired surfaces with a paint bucket and erases regions in paintings to creates holes like mouths and eyes. Once the painting was created, the second step is to extract the medial axis of regions and convert it into a graph used as the skeleton for a convolution surface. These surfaces are blended where the meshes intersect. However, by estimating skeletons to create convolution surfaces, details are lost by expanding the models from the medial axis obtained by the erosion process. Fig. 30 details the skeletons and convolution surfaces.

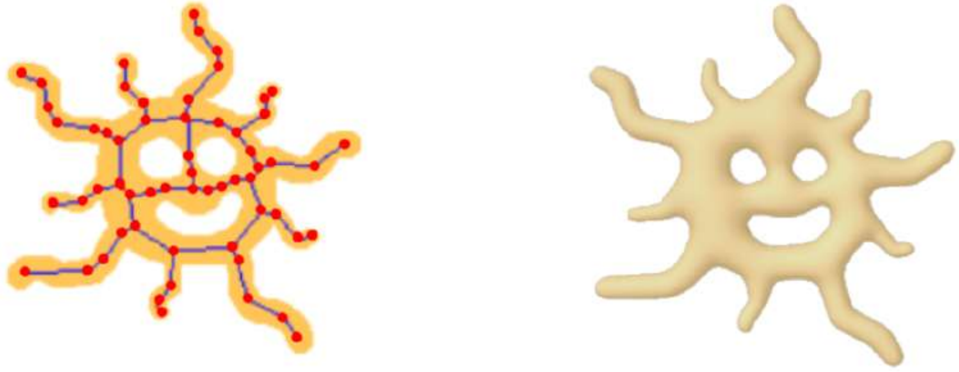


Figure 30: Example of painting-based modeling from Matisse. The convolution surface discards some details in the final model. Source: Bernhardt *et al.* [12].

Apparent layer operations for the manipulation of deformable object

Igarashi and Mitani [60] introduced layer operations for single-view object manipulation with two interaction techniques that include explicit swapping in 3D layers and an automatic depth adjustment through dragging the stripes over or under another object as shown in

Fig. 31. Their system supports local layering, self-occlusions, and folds, this is possible by representing a true 3D environment through a 2.5D interface. Dynamically computing the apparent layer structure, the system makes the depths adjustments to obtain valid results. For layer swap, the system alters the order of the layers under the mouse cursor and nearby areas. Although the proposed algorithm support layer operations in stripes, the framework does not reconstruct 3D models, providing operations only for 3D models built previously.

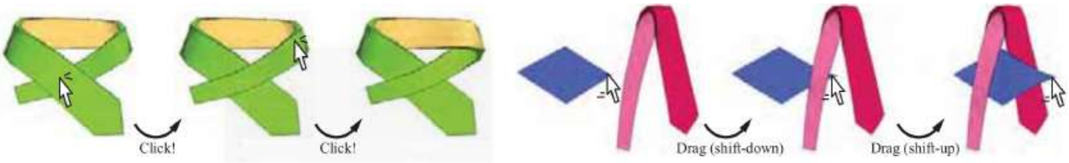


Figure 31: Apparent layer operations. Left: Layer swap allows the user to swap local layers under the clicked point. Right: Layer-aware dragging allows the user to drag over (shift-up) or under (shift-down) a colliding object in the screen space. Source: Igarashi and Mitani [60].

Modeling-in-Context: User Design of Complementary Objects with a Single Photo

Lau et al. [78] present a framework that allows the user to participate in the process of designing their own objects. The presented system, exemplified in Fig. 32, provides a 2D interface to user sketch the new object above a single photo while annotates geometric properties to guide the reconstruction. The system does not use the information on a photo to guide the reconstruction depending on the sketch drawn by the user, requiring that the user define the order of the depths for contours, so the system can rebuild the object correctly.

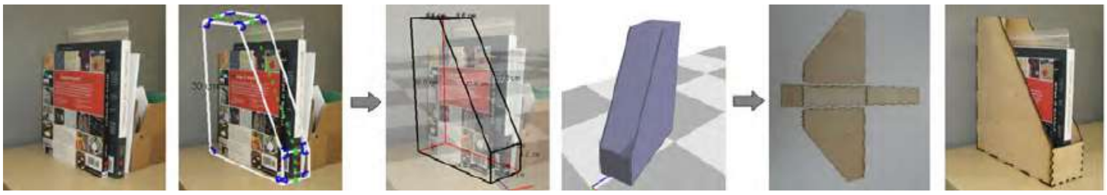


Figure 32: Example of Reconstruction of an Object. Source: Lau *et al.* [78]

NaturaSketch: Modeling from images and natural sketches

NaturaSketch focuses on natural interfaces that allow more real interaction [98]. The algorithm performs a classification of lines drawn to triangulate the points sampled after converting the traces into a polyline approximation, shown in Fig. 33. This classification allows users to draw multiple lines, cross lines, and connect them. The system also allows the user to make annotations indicating holes, bumps, extrusions, embosses, and cross-sections (Fig. 34). The final model is obtained by inflating bodies based on a distance transform from the edges. However, to create 3D models of greater complexity, the user needs to draw each part separately and then join them to complete the models.

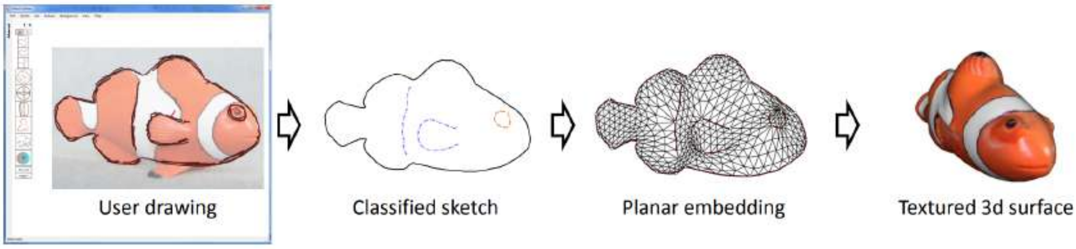


Figure 33: NaturaSketch system overview. Source: Olsen et al. [98].

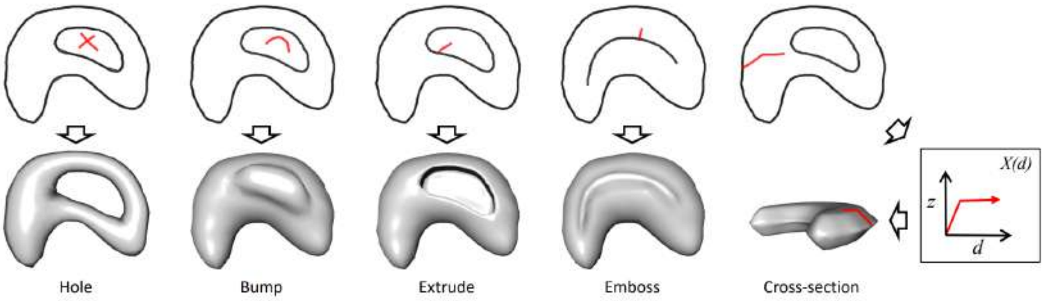


Figure 34: User annotations in NaturaSketch System. Source: Olsen et al. [98].

Sketching of Mirror-Symmetric Shapes

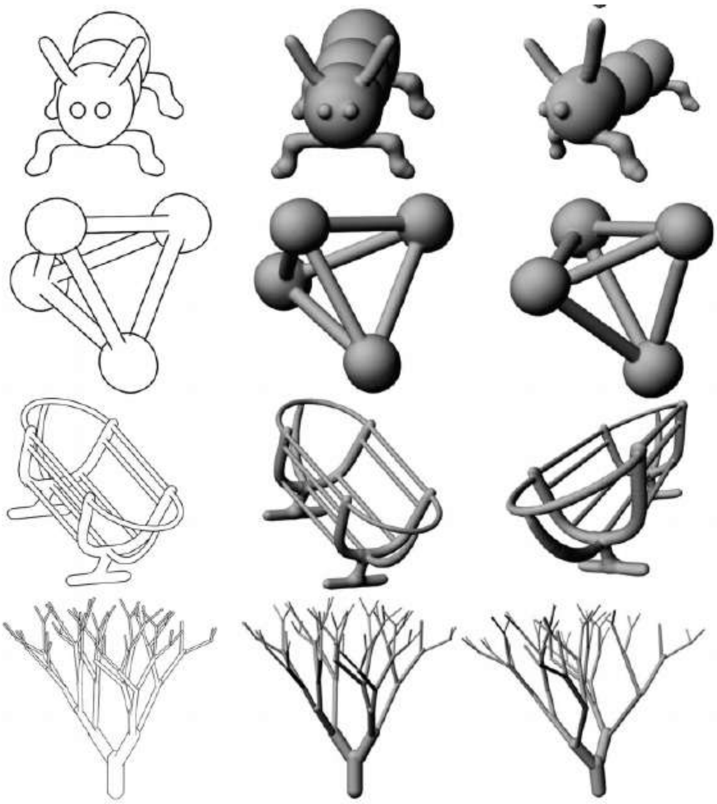
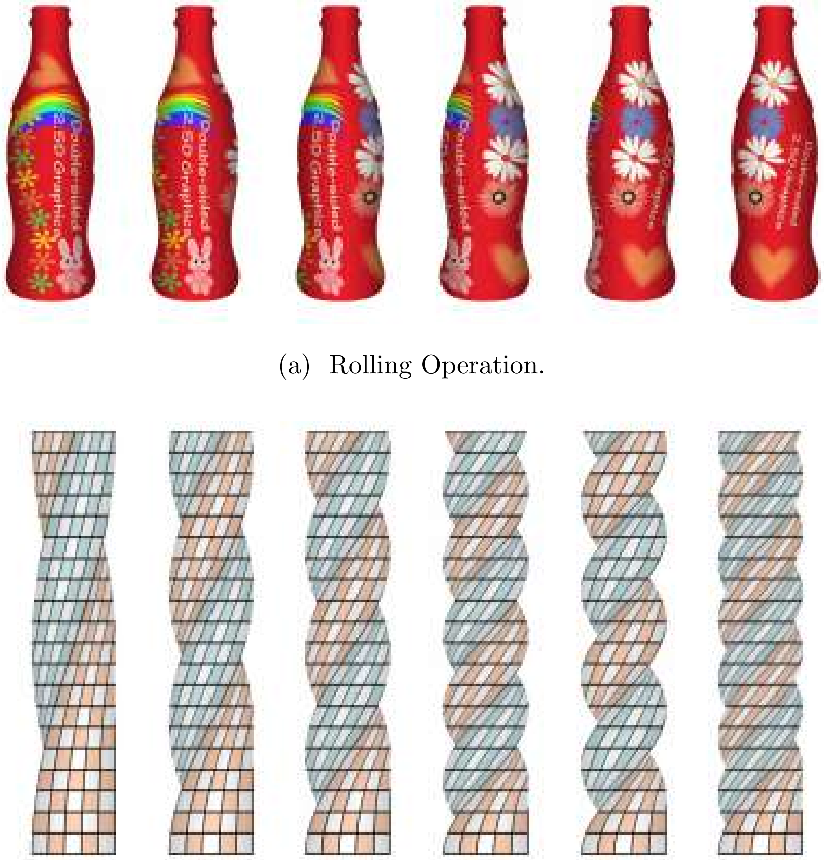


Figure 35: Examples of sketch inputs and final 3D models. Source: Cordier et al. [26]

Cordier et al. [26] presents a system to create mirror-symmetric surfaces from free-form sketches. The system takes the user's 2D silhouettes curves drawn on the sagittal-plane and classify them in two sets of symmetric and nonsymmetric curves. Then, the relative depth order is computed analyzing T-junctions and cusps of the input sketch. Finally, a closed surface is generated for each curve and the final model is obtained as the union of all surfaces, shown in Fig. 35. The inflation of shapes from skeletons and unions can cause a mismatching between the sketch and final models.

Double-sided 2.5D graphics



(a) Rolling Operation.

(b) Twisting Operation.

Figure 36: Operations of double-sided 2.5D graphics. Source: Yeh et al. [127].

To enrich the visual appearance while manipulating 2D graphical objects, Yeh et al. [127] presented a family of novel interactions on 2.5D graphics, that allows users to create visual effects for rolling, twisting, and folding operations on double-sided objects. Although this work presents promising results, the current system has certain limitations such as, rolling or twisting (Figs. 36a and 36b) operations can display visual artifacts as distortions. Another limitation is that the shapes are not associated with surfaces of revolution, and the rolling operation can cause distortions to specific drawings when applied.

CrossShade: shading concept sketches using cross-section curves

Shao et al. [109] facilitate the production of shaders to create 3D perspective effects of conceptual sketches. The work developed deals with the class of construction curves, known as cross-sections, presented in Figure 37, which help to create sketches as well as understanding the represented 3D shape. Then, the developed algorithm estimates the normal fields of these networks of curves to define the appearance of the object using *shaders*. However, the work presented by Shao et al. [109] has some limitations. The projections adopted are orthographic, disregarding other types of projections that can cause ambiguities in the output presented by the algorithm. In addition, the normal interpolations presented do not allow open regions and, the sketch needs to be drawn in separated layers.

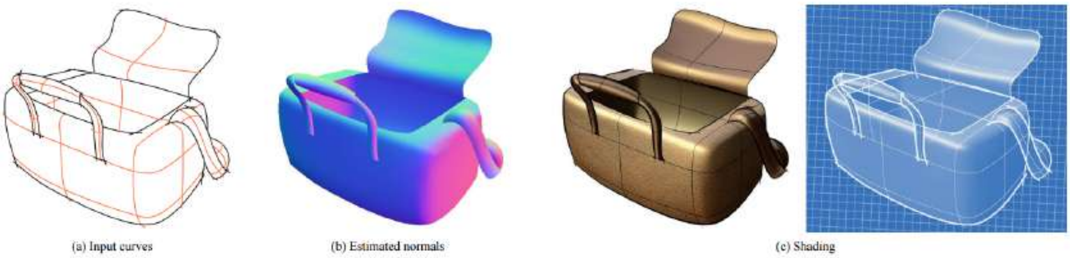


Figure 37: Cross-section shading. Source: Shao et al. [109]

Soft folding

Zhu et al. [131] introduced an interactive method for deforming thin-plate forms. Working in a 3D shape, the user sketches 2D curves specifying magnitude and sharpness for folding these thin-plate objects. Then, the proposed algorithm approximates the 3D surfaces computing the folded 3D geometry based on a folding map represented by local fields. In the end, a globally folded 3D shape is retrieved assembling the folded patches as shown in Fig. 38. The system has a sketching interface that allows to the user specify the planar and folded areas with hints on it, controlling interactively the magnitude and sharpness using slider bars. However, contours drawn in this Sketch-Based system are used to alter 3D surfaces used as inputs, thus, lacks the ability to reconstruct 3D surfaces using the strokes sketched by the user. However, this system infers 3D depths and layers using clues found on drawings or 3D models, failing in drawings with complex overlapping contours. Besides that, the system only creates shell-like layered structures, where is mandatory a previously created 3D model to use as a base for the addictive operations.

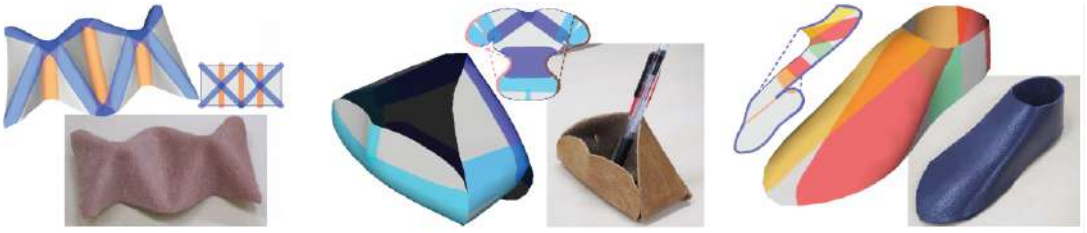


Figure 38: Thin-plate forms modeled by soft folding. Source: Zhu et al. [131].

Automatic single-view character model reconstruction

Buchanan et al. [17] create skeletons from 2D outlines in concept artworks that eliminates problems with hidden parts simplifying assumptions, then using these skeletons, circular arcs are placed to estimate the 3D surface of the characters. This process is exemplified in Fig. 39. In the end, bone rigging and skinning for animation purposes are performed on the mesh, and the original concept image is modified and used as a texture.

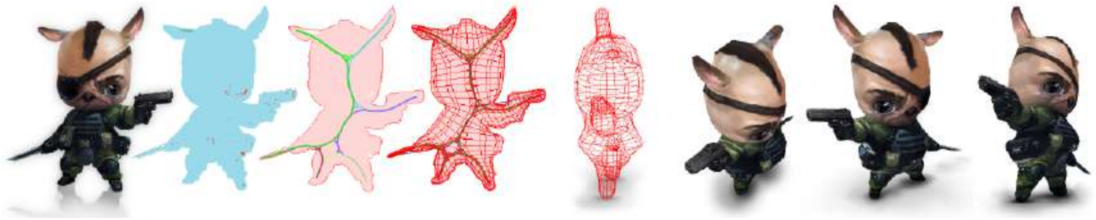


Figure 39: Automatic construction of 3D meshes from a single piece of concept artwork. Source: Buchanan et al. [17].

True2Form: 3D curve networks from 2D sketches via selective regularization

Xu et al. [124] proposed a modeling framework based on sketches that reconstruct the curves of typical drawings by inferring 3D shapes from 2D drawings, *True2Form*. Their work demonstrates that designers favor perspectives that reveal to the maximum information about the 3D shape of the object and strategically describe curves that have information and properties such as curvature, symmetry, parallelism, and others. Figure 40 shows examples of characteristics or regularities used to reconstruct the model. However, this approach has some limitations such as (a) the difficulty to represent an object in a single perspective can affect the result of the proposed algorithm, (b) the framework requires sketches well described and interconnected because it is a purely geometric approach, and (c) the regular properties exploited are few, allowing improvements in the fidelity of the result presented by the algorithm.

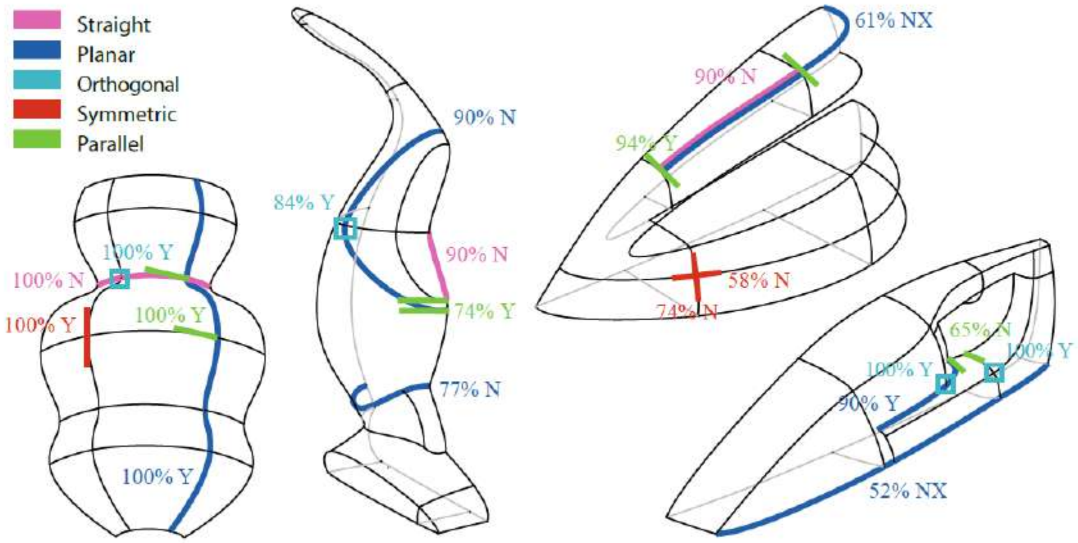


Figure 40: Example of exploited properties on framework True2Form. Source: Xu et al. [124]

Ink-and-ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters

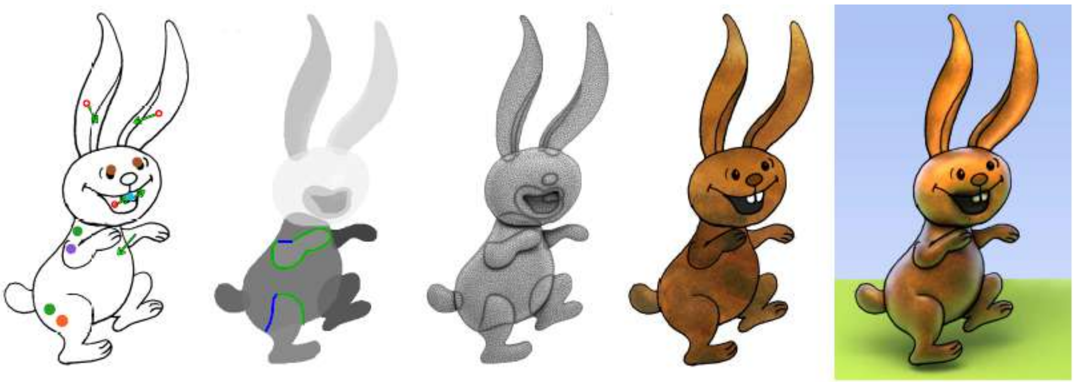


Figure 41: Example of the approach and bas-relief model generated by Ink-and-Ray. Source: Sỳkora et al. [116].

Sỳkora et al. [116] present Ink-and-Ray, a technique that simulates global illumination effects in hand-drawn characters. The user annotates dots and arrows to indicate depth order and segmentation of parts. Instead of reconstructing a 3D model, it generates a bas-relief model that suffices to simulate the global illumination of the input view. The process is exemplified in the Fig 41. However, this reconstruction cannot be used in other views because it deforms and loses coherence.

Bendfields: Regularized curvature fields from rough concept sketches

Iarussi et al. [59] extended the regularized curvature lines concept to non-orthogonal crossed fields estimating surface normals and 3D curvature directions from rough 2D concept sketches. Extrapolating these curvature lines commonly drawn by designers, this approach can interpret contours as running over a convex or a concave surface patch. This process could lead to an inconsistent patch, *i.e.*, an estimated convex surface where it should be a concave surface. Thus, to deal with these cases, the user can annotate surfaces with points through an interface flipping their orientation (red dots in Fig. 42).

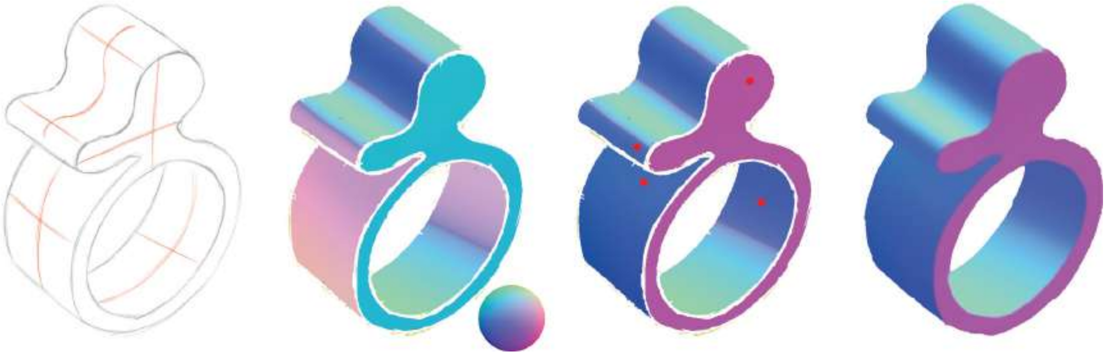


Figure 42: Estimated surface normals and user interaction to correct inconsistent patch. Source: Iarussi et al. [59]

.

Modeling Character Canvases from Cartoon Drawings

The work presented by Bessmeltsev et al. [13] demonstrate a technique to obtaining 3D models of *cartoons* from a drawing and the pose of a skeleton, shown in Fig. 43. Thus, some common limitations in 2D characters such as the ambiguity of 3D interpretations and the creation of faithful models to those of input are eliminated. But the limitations reported in this paper describe the need for well-defined skeletons and non-hidden features, as well as a good match between the design and its skeleton to avoid unexpected results. The fundamental premise for the development of the algorithm is the generalization of surfaces of revolution defined by the input skeleton, failing in cases of hidden contours, reliefs, or deformations in the forms. Still, elements such as eyes, ears, and others that change according to the observer's perspective have not been studied.

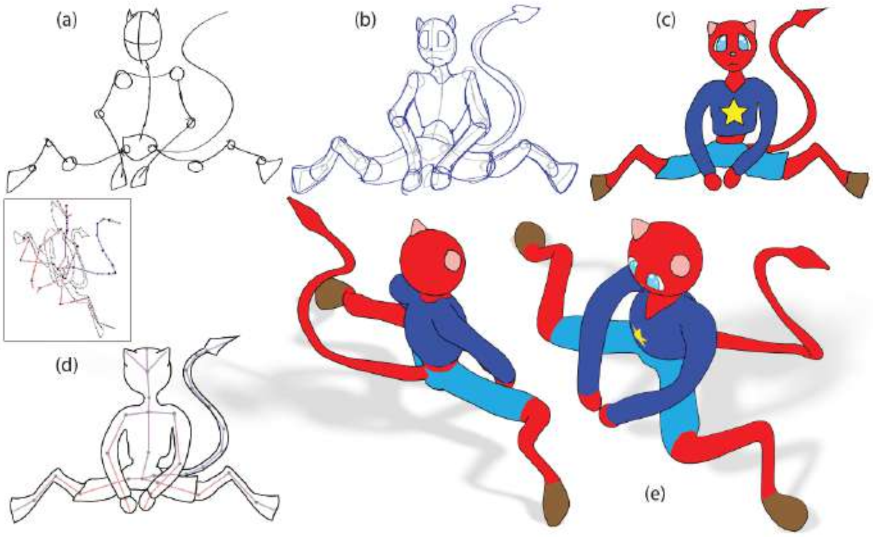


Figure 43: Overview of Modeling Character Canvases from Cartoon Drawings. Example of input (a, b), colorized cartoon (c), processing and matching (d) and, model generated by algorithm (e). Source: Bessmeltsev et al. [13]

Modeling 3D animals from a side-view sketch

To reconstruct 3D animals, Entem et al. [31] proposed to generate skeletons for limbs from side-view sketches of animals, shown in Fig. 44. 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 contours and discard unclassified contours. As a result, it may simplify or ignore the details of the input sketch.

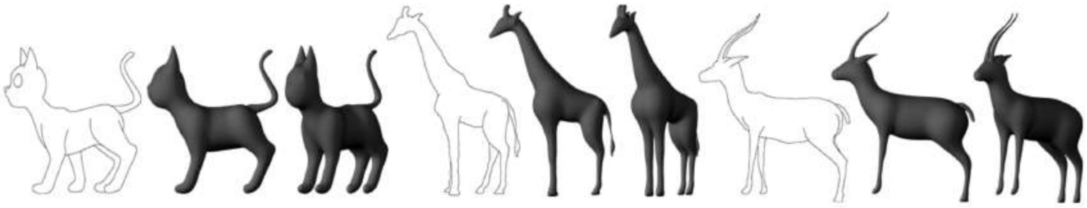


Figure 44: Reconstructed animals from a side-view sketch. Source: Entem et al. [31].

Globally Consistent Wrinkle-Aware Shading of Line Drawings

Jayaraman et al. [63] estimated the local geometry using a set of Gestalt principles to create a 3D-like soft shading guided by cues, connections, and spatial arrangement of wrinkle strokes. The overview of the approach is shown in Fig. 45. However their approach does not create 3D models, even with a consistent stroke interpretation, the system only uses shades to given a 3D appearance to sketches.

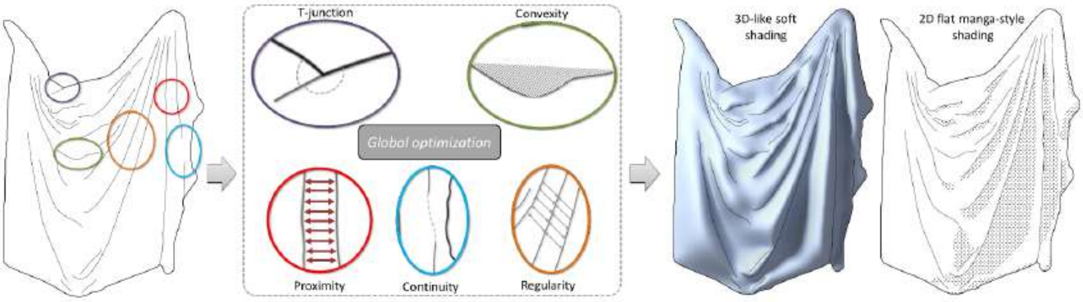


Figure 45: Overview of the process to create a 3D-like soft shading. Source: Jayaraman et al. [63].

Interactive high-relief reconstruction for organic and double-sided objects from a photo

Yeh et al. [128] introduce an interactive user-driven method to reconstruct high-relief 3D geometry from a single photo. The proposed algorithm, shown in Fig. 46, constructs a 2.5D model using the segmented image, with layering and region completion. Then, the user hint slopes and curvature cues on the image in order to inflate and lift up layers. At the end, an optimization and stitching process is performed to produce high-relief 3D models. However, instead of sketches, it uses annotated photos as input to generate high relief reconstructions. Besides that, it presents similar issues to Ink-and-Ray [116], resulting in high relief outputs instead of a 3D reconstruction.

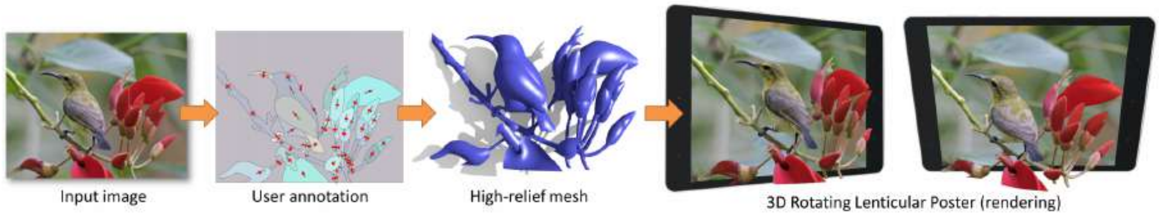


Figure 46: Interactive user-driven method to reconstruct high-relief 3D geometry from a single photo. Source: Yeh et al. [128].

Automatic structuring of organic shapes from a single drawing.

Entem et al. [32] proposed an approach to assign depth-layers to complex sketches using region-based structuring as shown in Fig. 47. Silhouette curves delimit regions and may include internal parts that possibly rely on top, creating layers. These layers are then detected by decomposing the input drawing which allows posing and animation of the models. The limitations addressed to this approach are mainly related to ambiguities on drawings, for instance, the algorithm assumes that the occlusions occur in T-junctions or cusps and must be well defined, otherwise, the algorithm fails.

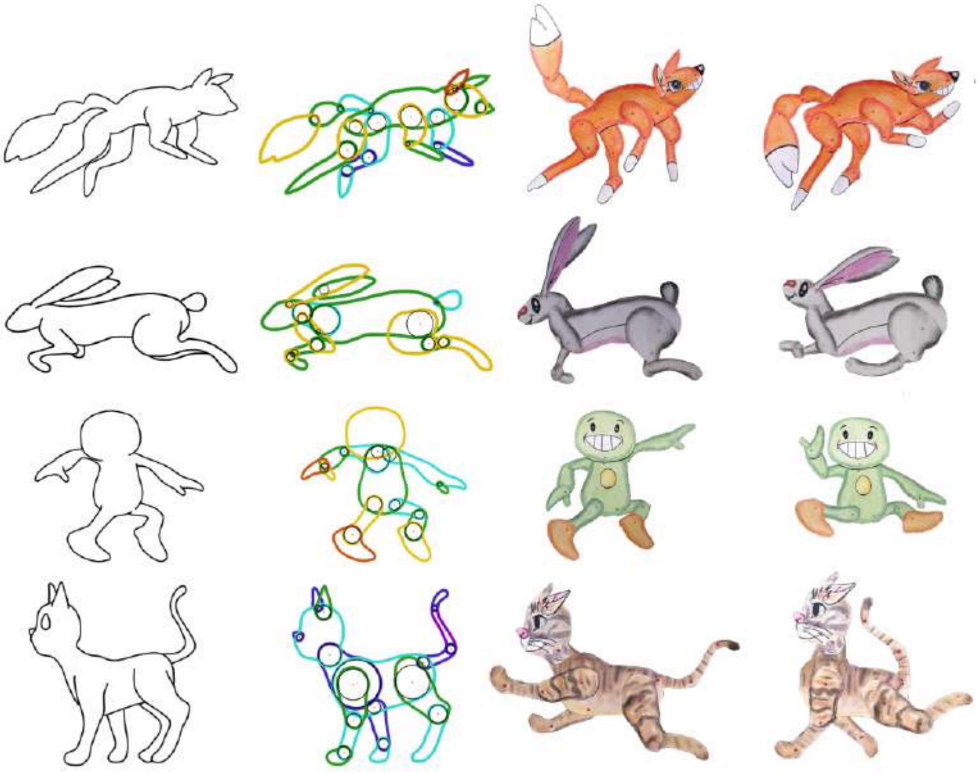


Figure 47: Automatic structuring of organic shapes from a single drawing. Source: Entem et al. [32].

2.4 Thesis Overview

Our research defined trough our goals and hypotheses led the studies of this work to improve and contribute to single-view methods. These methods often interpret and infer the reconstruction of the 3D models from single-view silhouettes and other strokes of sketches or images. The approaches can be categorized into automatic or interactive frameworks as illustrated in Fig. 48.

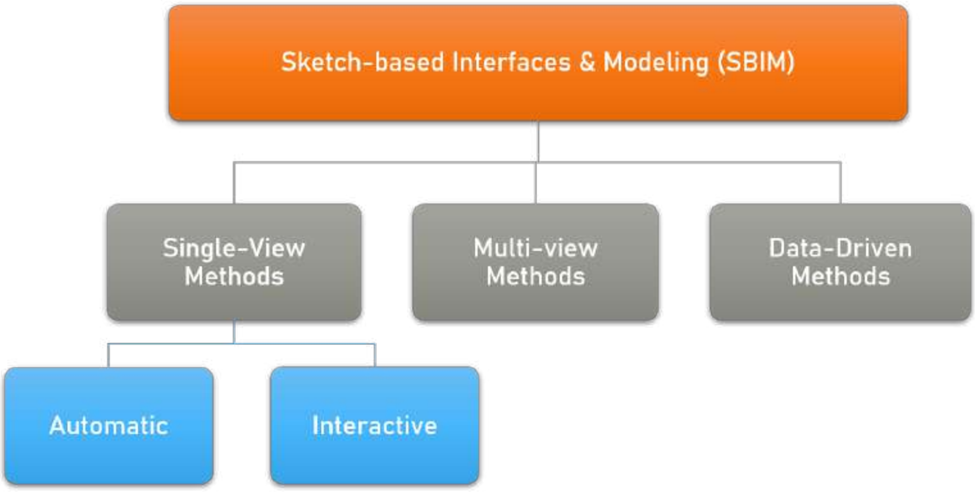


Figure 48: Research context.

Automatic frameworks analyze ambiguities generated by occlusion decomposing sketches or drawings into parts to define a depth order and a 3D shape for reconstruction for each part. In this work, we propose contributions to the state-of-art in automatic algorithms (Chapter 3). However imposing some constraints and using structured 3D drawings, automatic algorithms reconstruct 3D models from a small variety of drawings, limiting their use. Although automatic algorithms decompose sketches and creates meshes for different parts, the user expertise to define layers and infer occluded parts is usually absent, restringing the final models to the reconstruction pattern proposed by the method.

It is worth mentioning that although approaches using machine learning were proposed in recent years when dealing with specific areas, the main concern is the lack of datasets composed by scientific drawings to train the networks.

Different from faces [22] and common models, *i.e.*, cars, houses, and toys [23, 66, 84], in taxonomic research, structured images, and 3D model databases, in which there are no SBIM systems for experts to create their 3D models, are commonly lacking. A further issue is to reconstruct extinct specimens, where there is no photo or reasonable fossil evidence to be used as parameters for generative models. These limitations led our research to a further investigation, the creation and evaluation of an interactive framework (Chapter 4).

Interactive frameworks are another approach for generating information to retrieve as a result, 3D geometry. These methods can be used with or without a reference image as input, where the user can interactively drawn or hint clues over an input sketch or image. Generally, sketch-based systems use this approach for complex objects that contain multiple contours, to overcome the limitations of identification and segmentation through an automatic or semi-automatic algorithm.

The contributions of the proposed algorithms are described as follows.

1. Contour-Aware 3D Reconstruction of Side-View Sketches

In contrast to previous works, in the first study, we propose a skeleton-free approach, exemplified in Fig. 49, that ensures that the 2D contours are interpolated into the 3D reconstruction, thus capturing the small details of the sketch. Feasible 3D models are generated using Hermitian Radial Basis Functions (HRBF) Implicits to interpolate the 2D sketch curves with 3D normal estimated.

This research was developed focusing in:

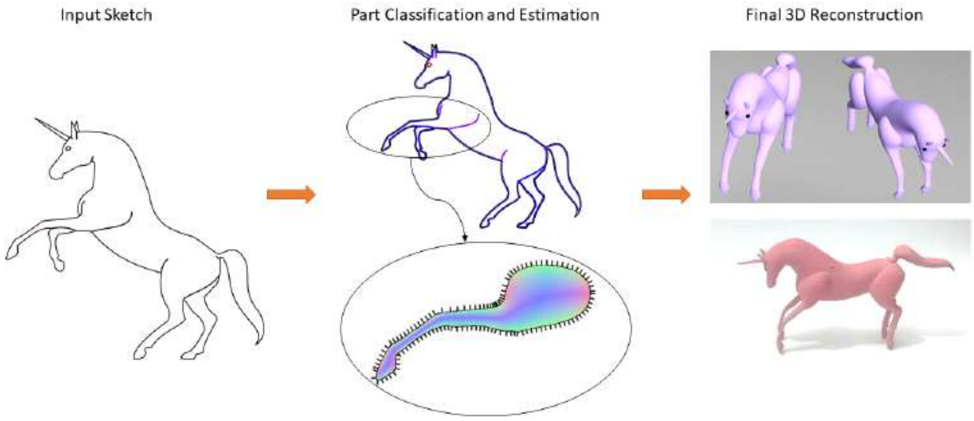


Figure 49: Example of 3D reconstruction for a unicorn drawing used as input sketch.

- A skeleton-free approach for 3D reconstruction from side-view sketches;
- A normal-driven estimation of depth in shapes;
- A 3D reconstruction by contour constraints in drawings based in Hermitian Radial Basis Functions (HRBF);
- A new approach of part classification in sketches.

Previous works generate skeletons for sketches to create models, discarding important features in drawn contours. Besides that, the reconstruction of surfaces with relief, depth, or outside the sagittal plane is not supported. This automatic framework is directly correlated to the works of Cordier et al. [26], Entem et al. [31], Karpenko and Hughes [68]and Sỳkora et al. [116].

By leveraging the information provided by sketch, our technique (Fig. 49), 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 [41, 109], to propagate the Hermitian data that is further used to estimate the shape of the model.

2. Sketch-based modeling supported by 2D visual perception enhancements: application and analysis for illustrations of systematic biology

Inspired by traditional illustration strategies and scientific drawings, in the second study, is proposed a framework of sketch-based interface for modeling 3D objects with multiple contours as overlaid structures, as shown in Fig. 50.

Our sketching interface allows the user to infer perceivable symmetries and occluded parts of the model prior to its automatic 3D modeling employing visual perceptions for drawn parts.

This research was developed focusing in:

- A framework for overlaying contours based on illustration techniques;
- A set of visual effects tailored for complex and layered objects;
- An application and analysis for biological systematic illustrations.

In contrast to previous single-view sketch-based systems, the framework is tailored for sketch-based modeling on overlay drawings, taking advantage of user aiding and also allowing to explicitly describe occluded parts represented by knots, stripes, and overlaid parts. This interactive framework is directly correlated to the works of Anastacio et al. [4], Cherlin et al. [24], Entem et al. [32], Olsen et al. [98] and Ramos et al. [101]

We defined a set of computational tools and visual effects and sketching techniques to propose a sketch-based framework that enables 3D modeling with overlay support detailing a set of procedures to allow not only regular users but scientific illustrators to create and reconstruct 3D models. These tools and visual effects enable the user to define depths for layers in single-view drawings while obtaining real-time feedback for sketches and 3D models. It is worth to mention that our framework can complement previous non-interactive methods in order to provide support to inferred or discarded parts.

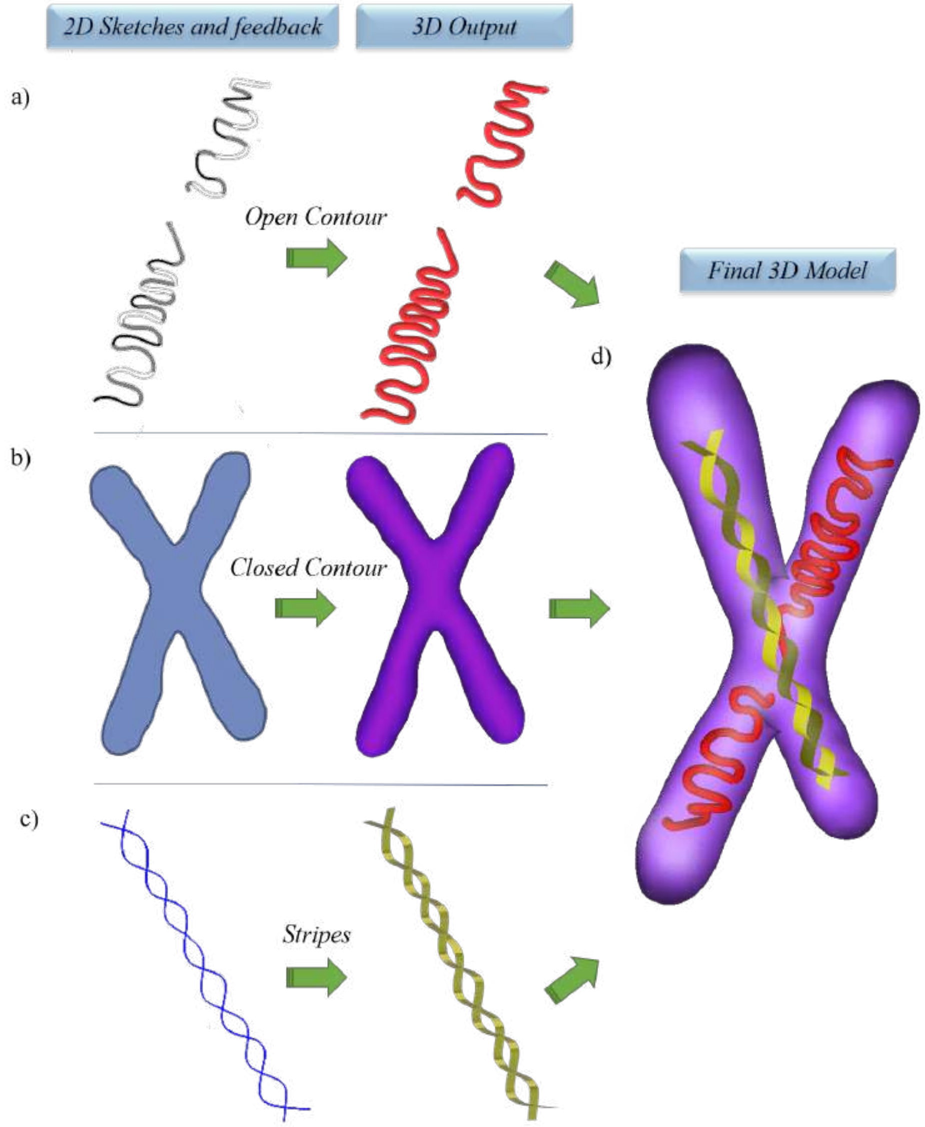


Figure 50: Overview of our system and contour categories supported as inputs for the reconstruction of an illustrative chromosome.

3 CONTOUR-AWARE 3D RECONSTRUCTION OF SIDE-VIEW SKETCHES

3.1 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 [123].

Recent research shows that the most effective approaches seek to reconstruct a specific family of surfaces [82]. 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 [44].

3.2 Overview

The input to our method is a digital sketch in a sagittal plane. Our system allows to use Scalable Vectorial Files (SVG) as inputs. 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 [9, 31, 123]. 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 [91]. 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.

3.3 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. 51. We assume the correctness of input contours in a clean vectorized drawing to avoid wrong classifications.

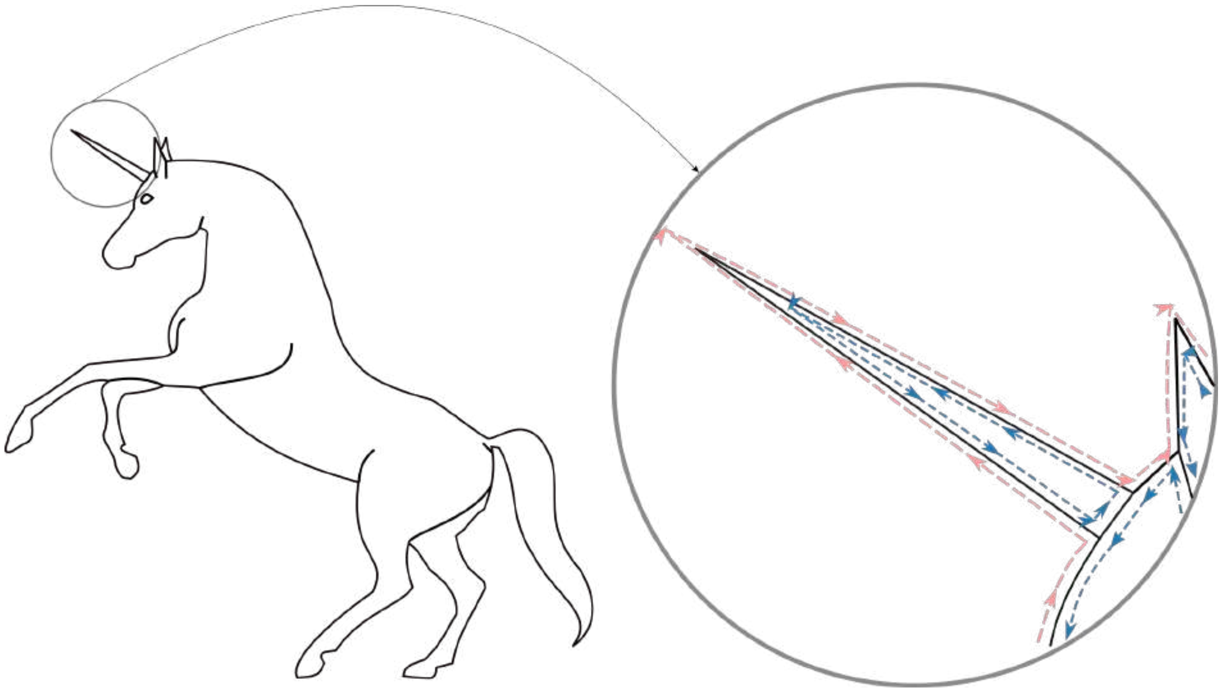


Figure 51: Half-edge data structure created from input drawing and cycle classification with inner edges: closed sequences of half-edges define these cycles. Input sketch and half-edge structure in detail: outer cycle, border cycle.

We identify and classify the cycles according to the following order [31], limiting the classification of each cycle to one group (Fig. 53):

1. **Outer cycle** - The largest clockwise oriented cycle that defines the external contour of the sketch.

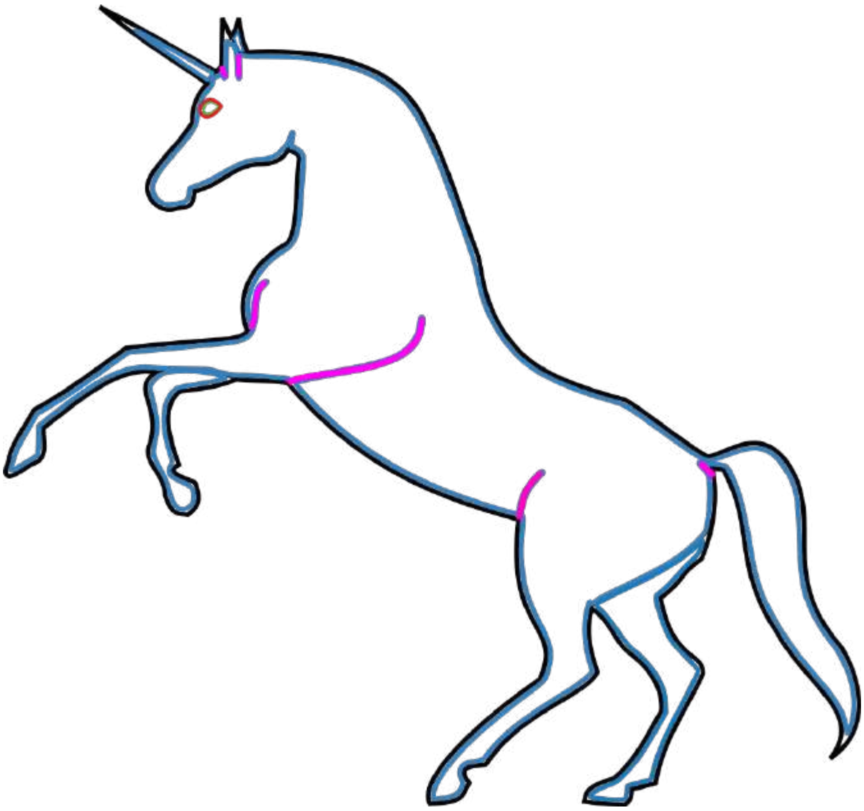


Figure 52: Classified cycles and inner edges. The opposing half-edges that belong to the same cycle are named inner edges, the case of the open ends in the legs of the unicorn.

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 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.

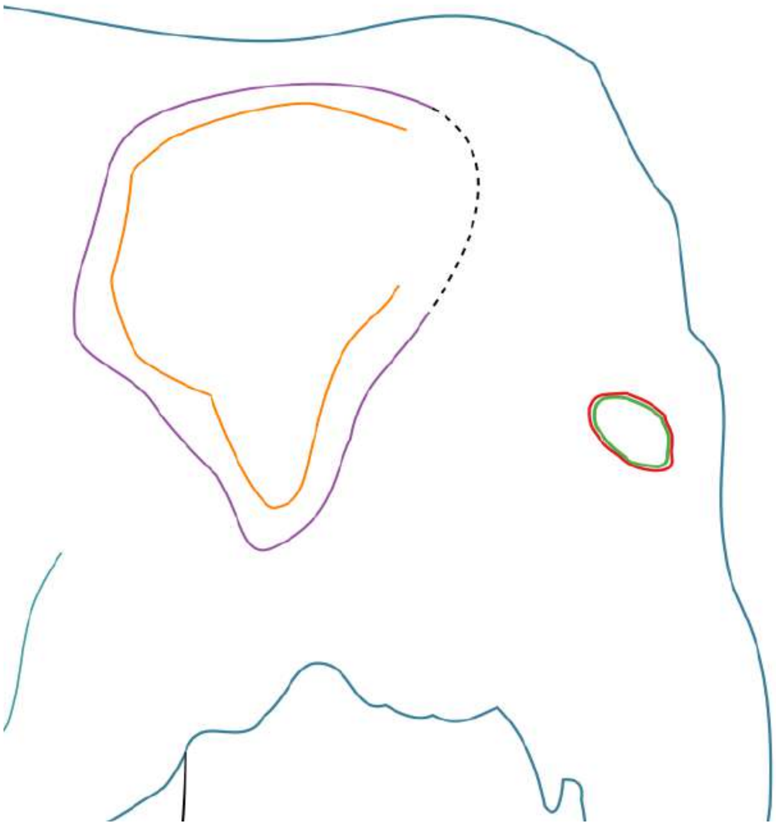


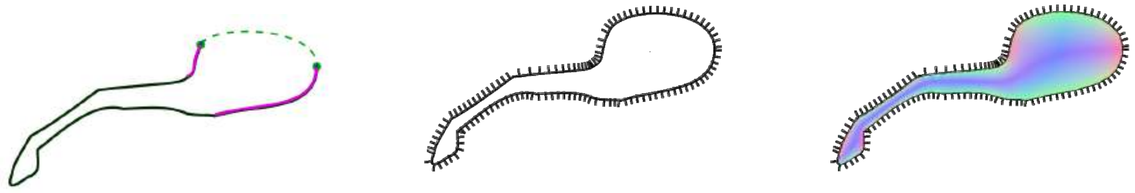
Figure 53: Example of cycle classification: border cycle, island cycle, adjacent border cycle, adjacent feature cycle and other.

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. 53.

Besides that, the use of HRBF Implicits and the 3D normal propagation [91] 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. 52. 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. 53 and 54a).

Such new closed parts (Fig. 54b) 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. 52 is not 3D reconstructed, as we consider the reconstruction of the left leg (foreground).



(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.

Figure 54: Normal estimation overview.

3.4 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 3.6).

Firstly, we sample points in the contours that define each part and estimate their 3D normals as shown in Fig. 54b. Then, by propagating the normals to the interior, we can determine the 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 [45].

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. [91]. 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. 54c depicts the normal field computed from the normals distributed along the contour.

It is worth highlighting 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 process is detailed as follows.

Algorithm 1: Generating Hermitian RBF Data

```
Input: $contourPath, \lambda_s$
Output: 3D Dataset composed of Points and Normals

$contourPoints \longleftarrow$ SamplePointsOnContour ($contourPath$)
$contourNormals \longleftarrow$ EstimateNormalsOnPoints ($contourPoints$)
AddPointsToDataset ($contourPoints$)
AddNormalsToDataset ($contourNormals$)
$gridPoints \longleftarrow$ CreateGridPoints ($contourPath$)
foreach $gridPoint \in gridPoints$ do
    if $gridPoint$ inside $contourPath$ then
        foreach $contourPoint \in contourPoints$ do
            $\omega \longleftarrow \omega + \frac{1}{||gridPoint - contourPoint||^2}$
            $n \longleftarrow n + \frac{normal\ (contourPoint)}{||gridPoint - contourPoint||^2}$
        $n \longleftarrow \frac{n}{\omega}$
        $n_z \longleftarrow \sqrt{1 - n_x^2 - n_y^2}$
        if $n_z < 0.5$ then
            $gridPoint_z \longleftarrow n_z * contourPathWidth * \lambda_s$
            AddPointsToDataset ($gridPoint$)
            AddNormalsToDataset ($n$)
            $gridPoint_z \longleftarrow gridPoint_z * (-1)$
            $n_z \longleftarrow n_z * (-1)$
            AddPointsToDataset ($gridPoint$)
            AddNormalsToDataset ($n$)
```

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.

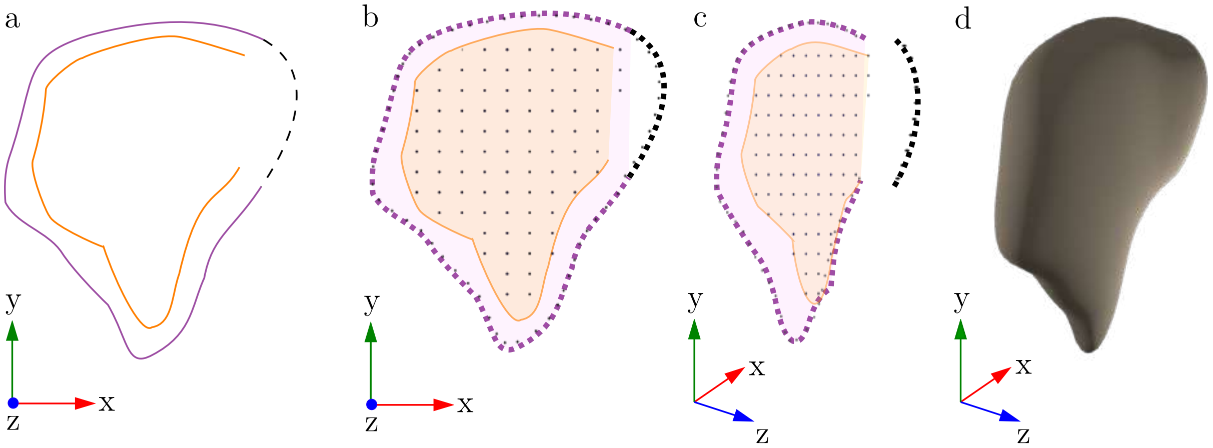


Figure 55: The process of reconstruction of a flat adjacent internal part: After the classification, the process starts by creating a new contour that closes the adjacent limb (dashed-line in a), then, points are sampled in contours and inside the adjacent feature cycle (b), the depth of the points sampled in adjacent border cycle remains on the sagittal plane while the depth of the points on the adjacent feature cycle is 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).

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. 55(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. 55(c), the points created inside the adjacent feature cycle are mirrored below the plane defined by the adjacent border cycle.

3.5 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. 56). Second, we find the point p (the red dot in Fig. 56) 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.

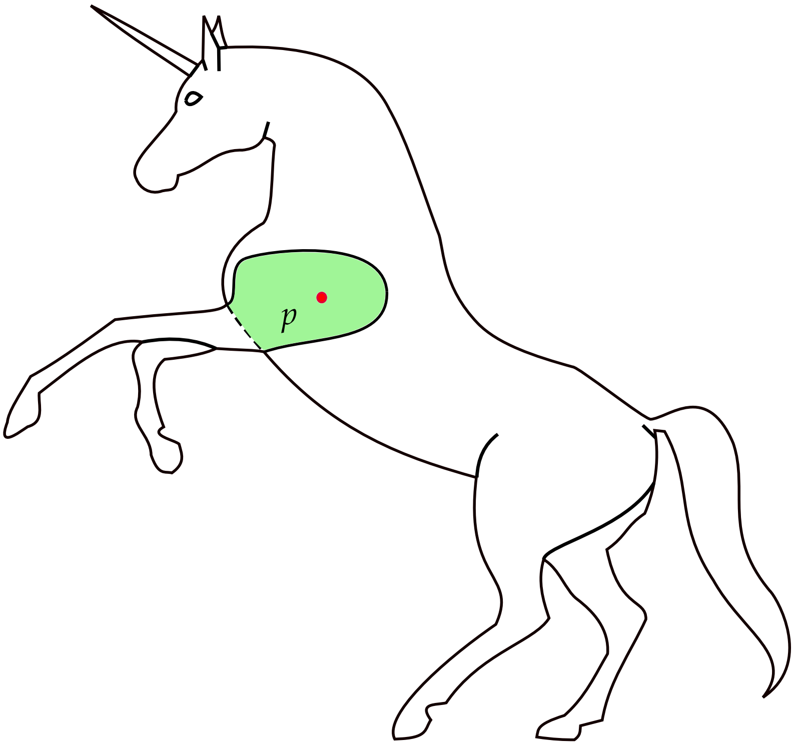


Figure 56: Depth estimation of the limb at the part classified as border cycle. The point p represents the normal with the highest value within the intersection of the adjacent limb and the body classified as border cycle below. This point is used to estimate the depth of the limb at the part classified as border cycle.

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. 57).

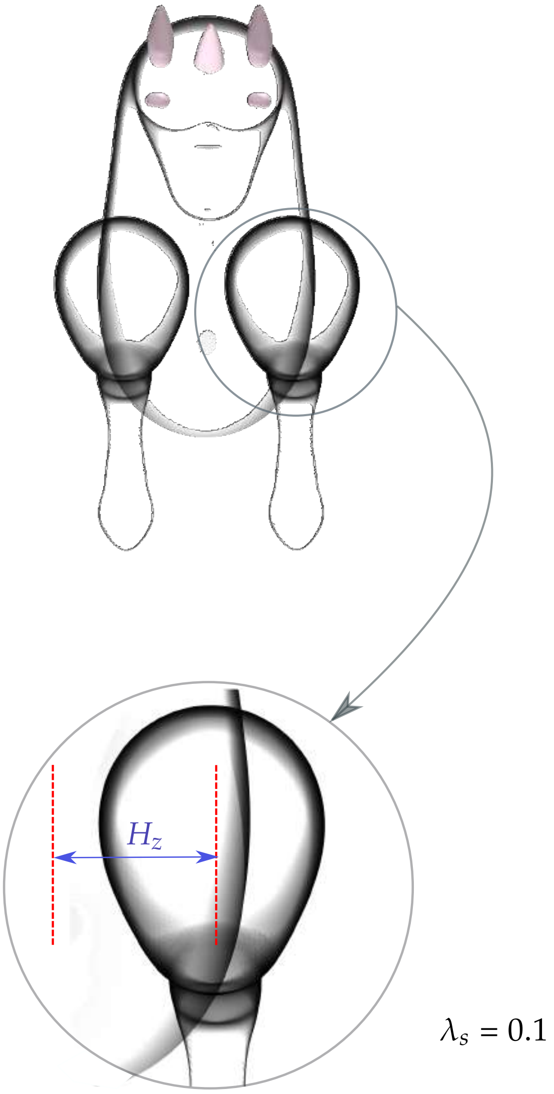


Figure 57: Processing of the depth coordinate of the symmetric parts. H_z is the final depth coordinate of the limb concerning the sketching plane.

For the adjacent internal parts, the placement is done using the newly generated curve (Fig. 55). 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.

3.6 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 [44]. In this work, we consider only points and normals constraints. The goal of the HRBF 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:

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

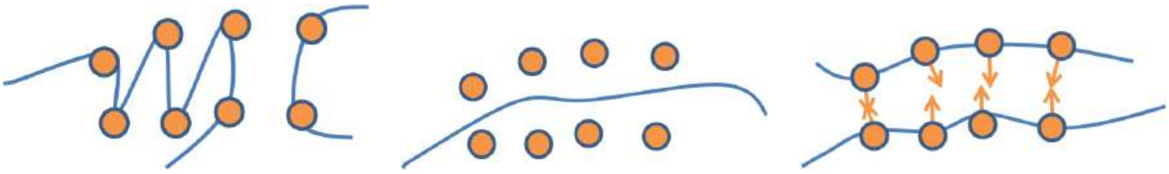
where $\alpha_j \in \mathbb{R}$ and $\beta_j \in \mathbb{R}^3$ are the unknown coefficients, ϕ is defined by the radial function kernel, and $p(\mathbf{x})$ is a low-order polynomial. We used the Polyharmonic Splines kernel and a complete polynomial of degree 2.

The HRBF interpolation leads us to a 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 [44] 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 3.4), 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.

Macêdo et al. [87] describe the main properties and contributions of HRBF Implicits that led this study to their use, between them:

- *Global implicit interpolant surface of Hermite data*: computing a zero-level implicit function that interpolates points and normal vectors.
- *Irregularly-spaced data handling*: HRBF implicits can efficiently interpolate data even in irregular distributions.
- *Handling of close sheets*: The results shown by Macêdo et al. [87], indicate that the Hermite-interpolatory method is superior to previous solutions. A comparison of interpolation and approximation methods in the case of close sheets is shown in Fig. 58, in our method, as the points are sampled near the contours, similar examples can be generated when creating the hermitian data.



(a) Robust Moving Least Squares (RAMLS) [42].

(b) Algebraic Point Set Surfaces (APSS) [51].

(c) Hermitian Radial Basis Functions Interpolation (HRBF) [87].

Figure 58: The issue of *close sheets*. Interpolation and approximation techniques can behave poorly. Source: Macêdo et al. [87] .

3.7 Results and Discussion

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], Sỳkora et al. [116], Li et al. [82]. In Fig. 59, 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.

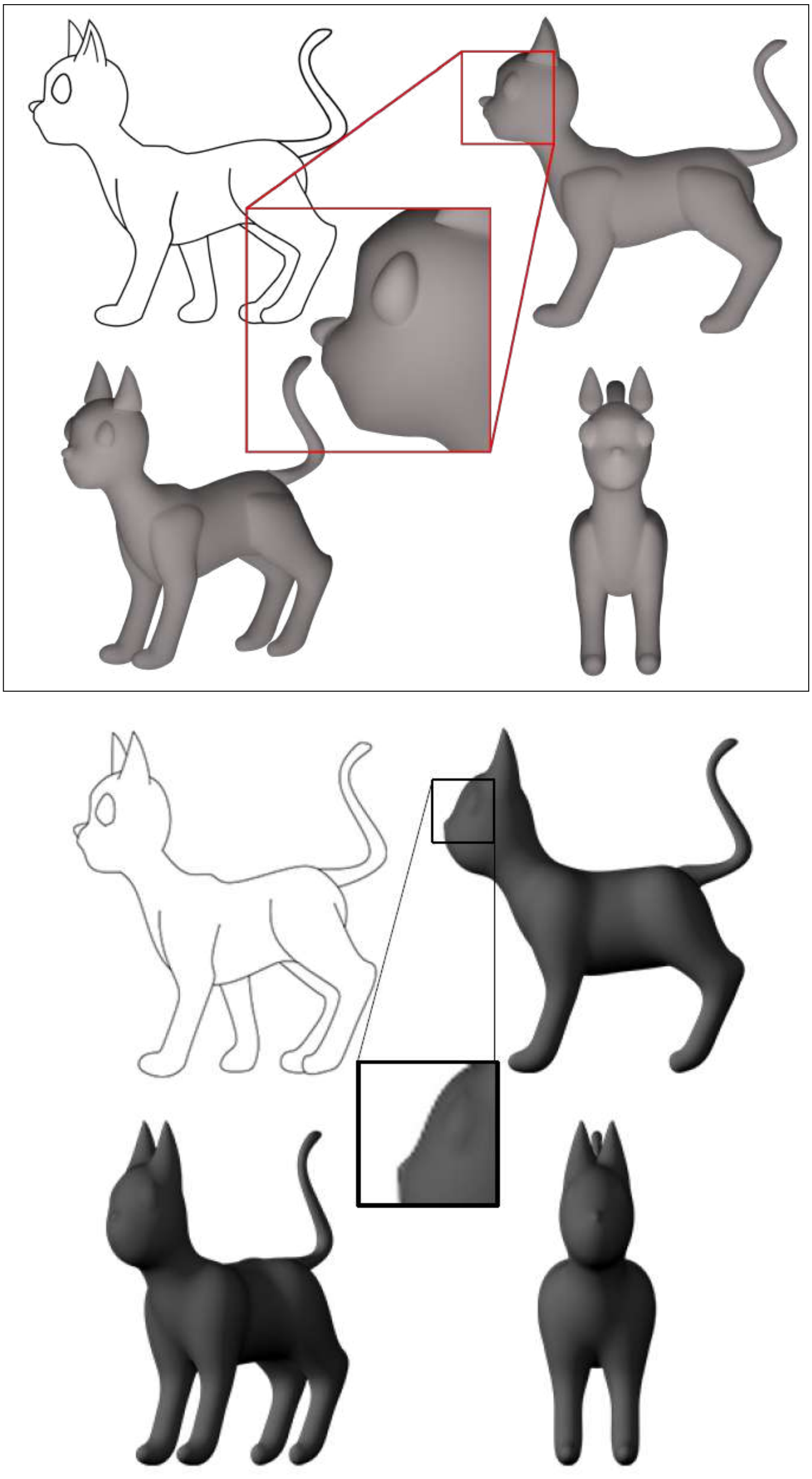


Figure 59: Contour constraints guaranteed by reconstruction. Our approach (top / highlighted). Results from the same drawing reconstructed by Entem et al. [31] (bottom).

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 elephant's ear in Fig. 60).

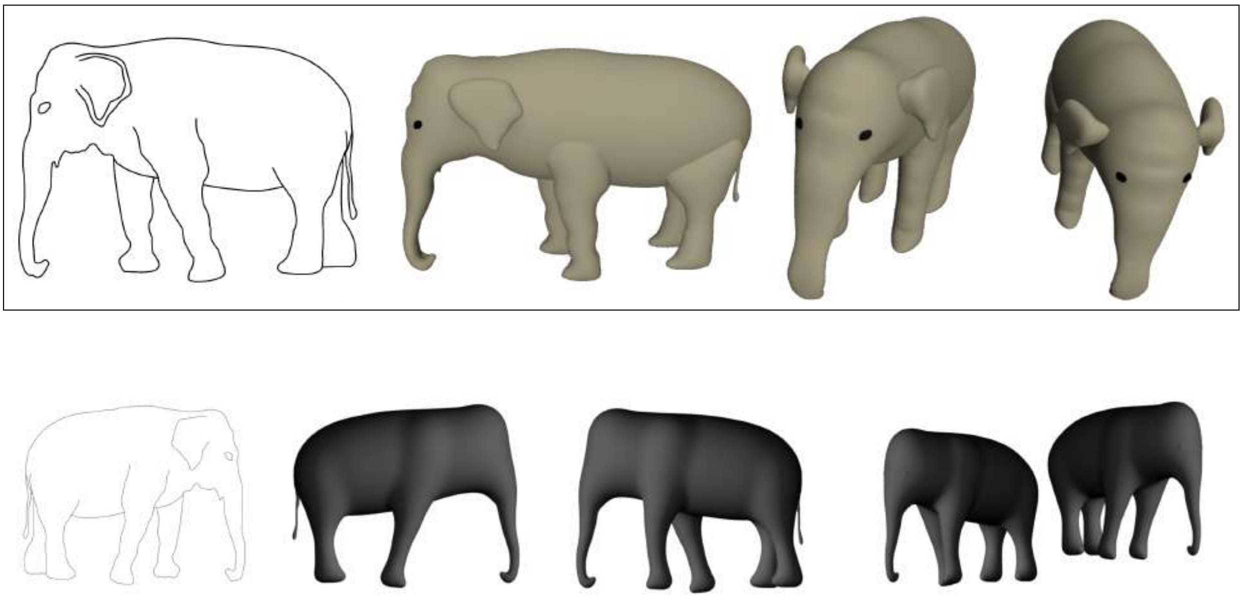


Figure 60: Elephant drawing and respective 3D reconstruction of contours discarded in previous approaches. Our approach (top / highlighted). Results from the same drawing reconstructed by Entem et al. [31] (bottom).

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. 61. 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 [12, 13, 26, 31].

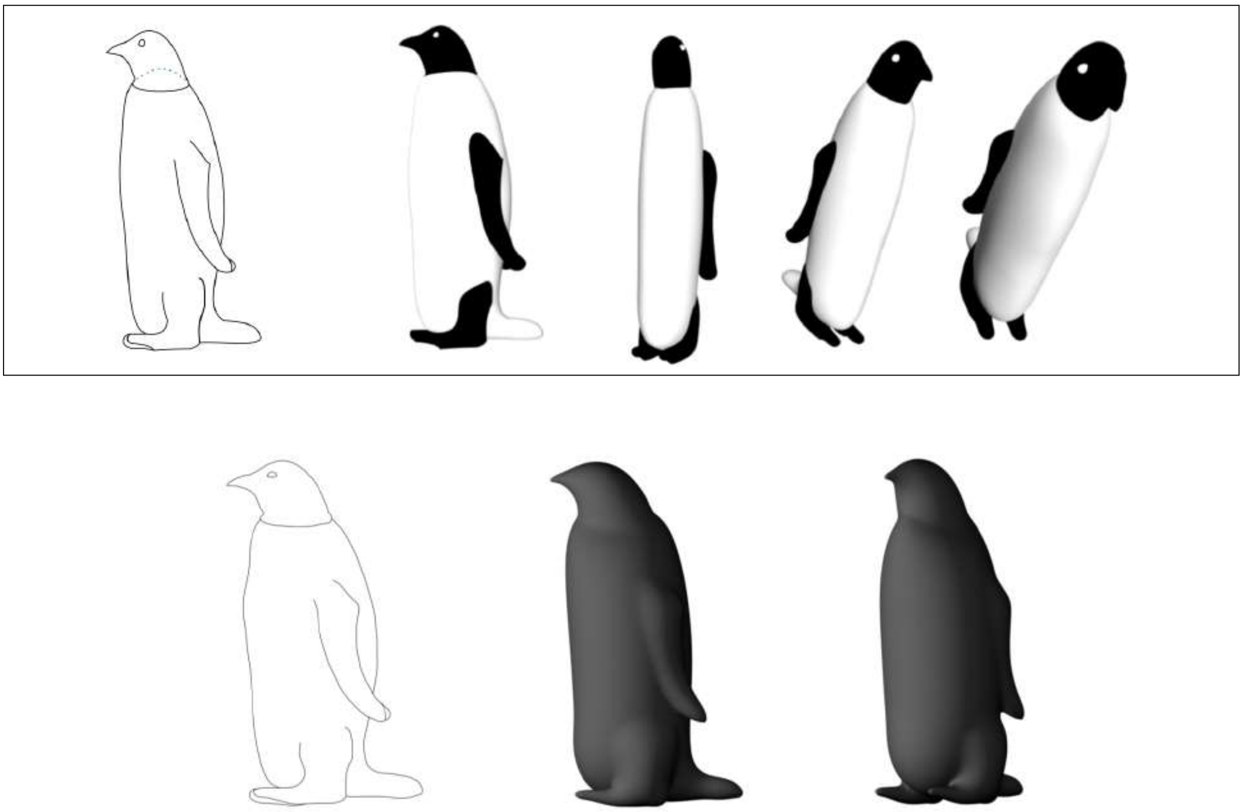


Figure 61: Penguin reconstructed by using body and head input drawings separately. Our approach (top / highlighted). Results from the same drawing reconstructed by Entem et al. [31] (bottom).

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, shown in Fig. 62, the difficulty lies in avoiding undesirable fat or skinny results.

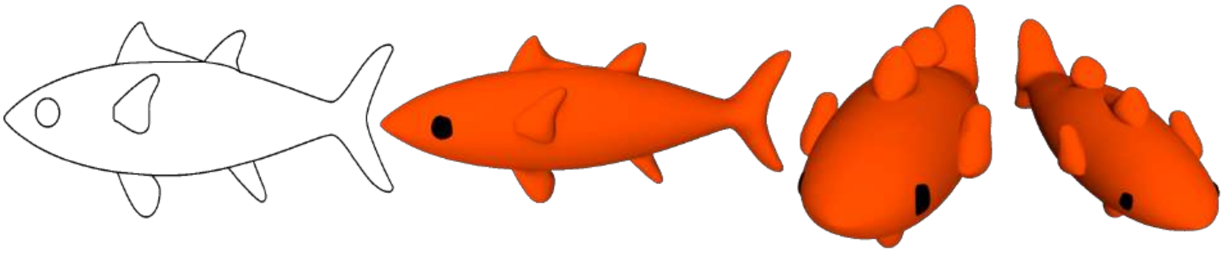


Figure 62: Input fish drawing (left) and its 3D reconstructed model.

The 3D reconstruction of the unicorn's horn (Fig. 63) is successful because a side-view sketch can entirely describe the reconstruction, instead, *e.g.*, a moose's horn, that contains details perceived only in other views.

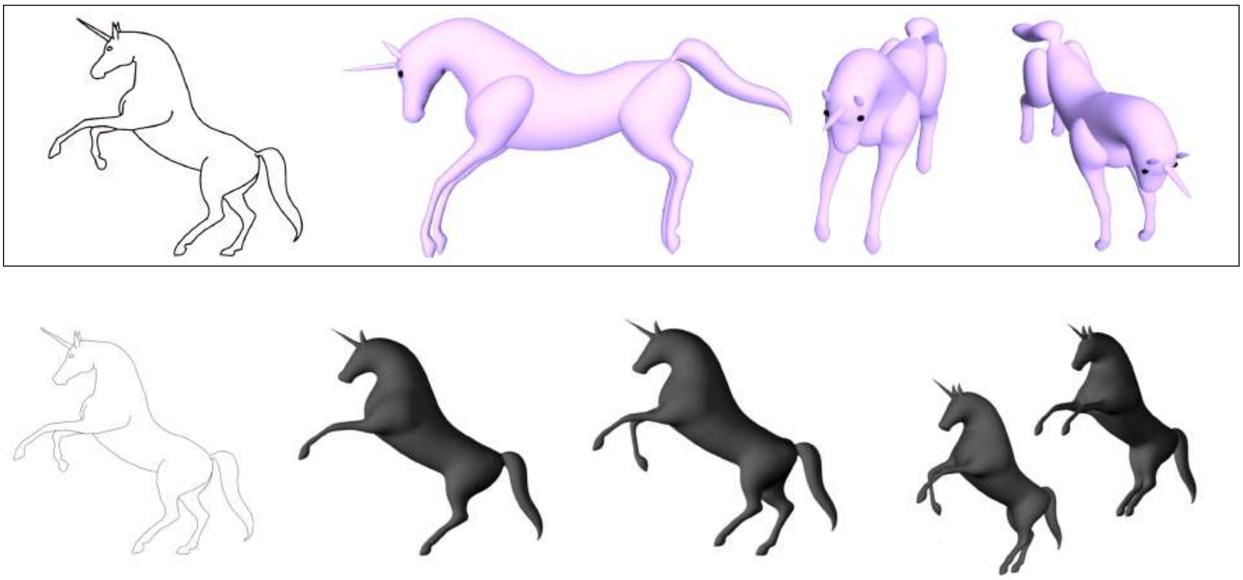


Figure 63: Unicorn reconstructed using our approach (top / highlighted). Results from the same drawing reconstructed by Entem et al. [31] (bottom).

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 mentioning 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.

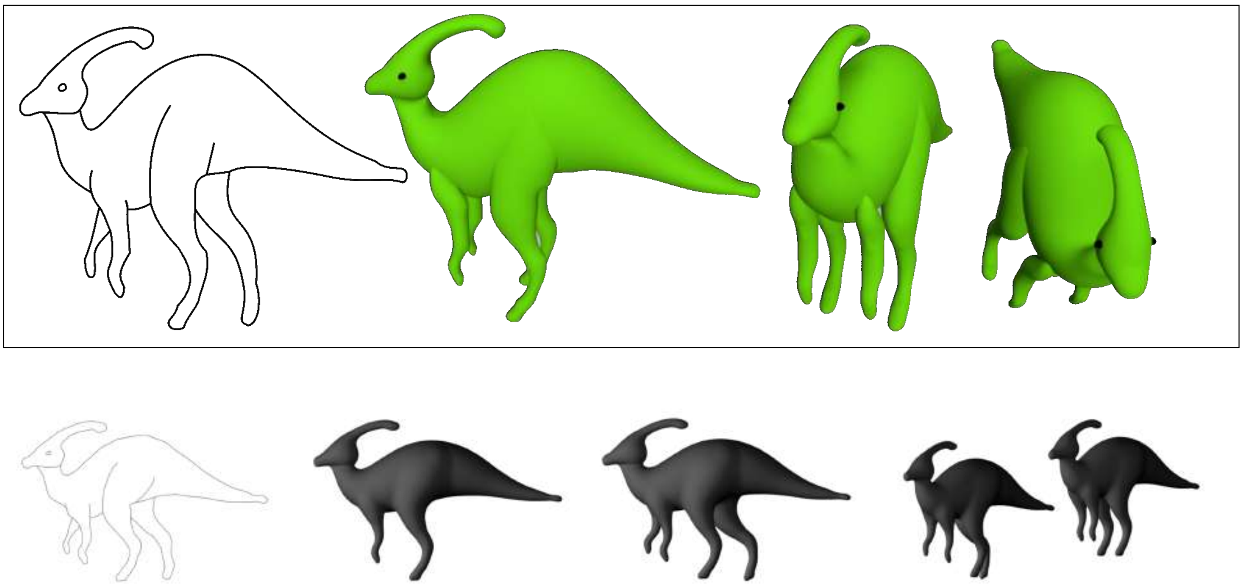


Figure 64: Input drawing of an dinosaur and its 3D reconstructed model. Results from our approach (top / highlighted) and from the same drawing reconstructed by Entem et al. [31] (bottom).

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. 65 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. 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 3.4) 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.

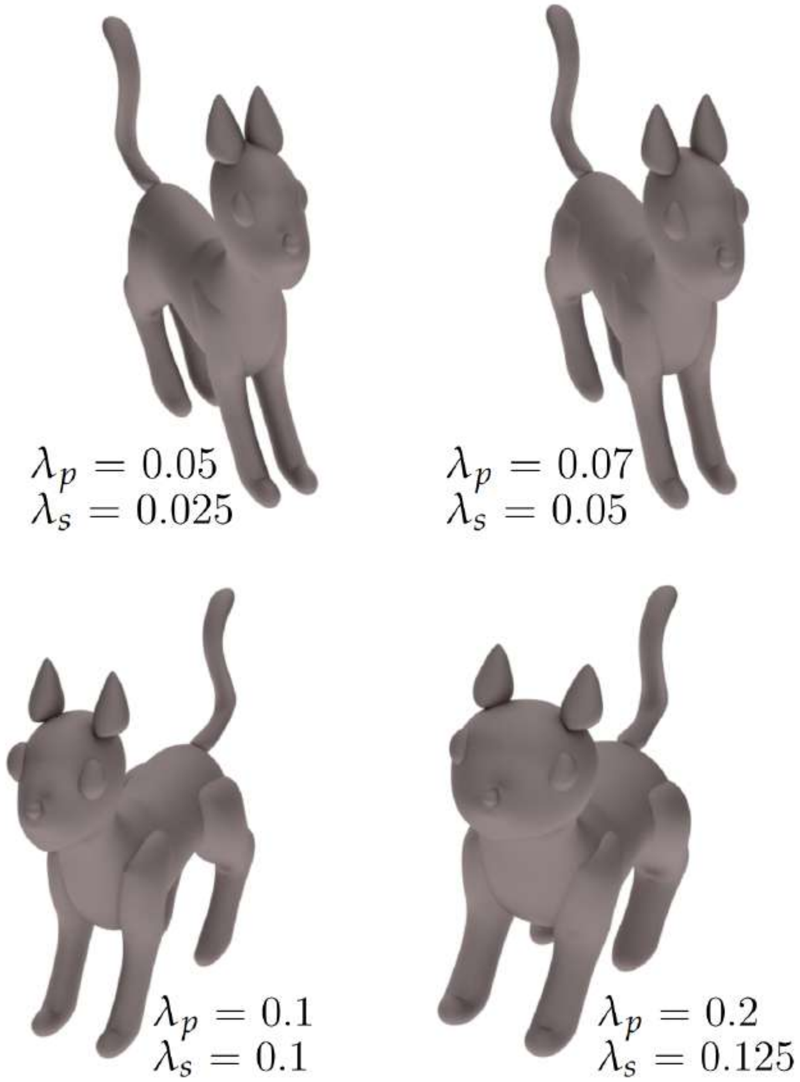


Figure 65: Adjustment flexibility of the final model based on proportion. We use the side-view width of each part of the sketch multiplying it by the value of λ_p. Our method also provides the flexibility to alter the depth of symmetrical parts by changing the value of λ_s.

Different from other HRBF reconstructions that use local support functions [83], our choice allows us to reconstruct surfaces with fewer points using global support functions as the points shown in Fig. 66.

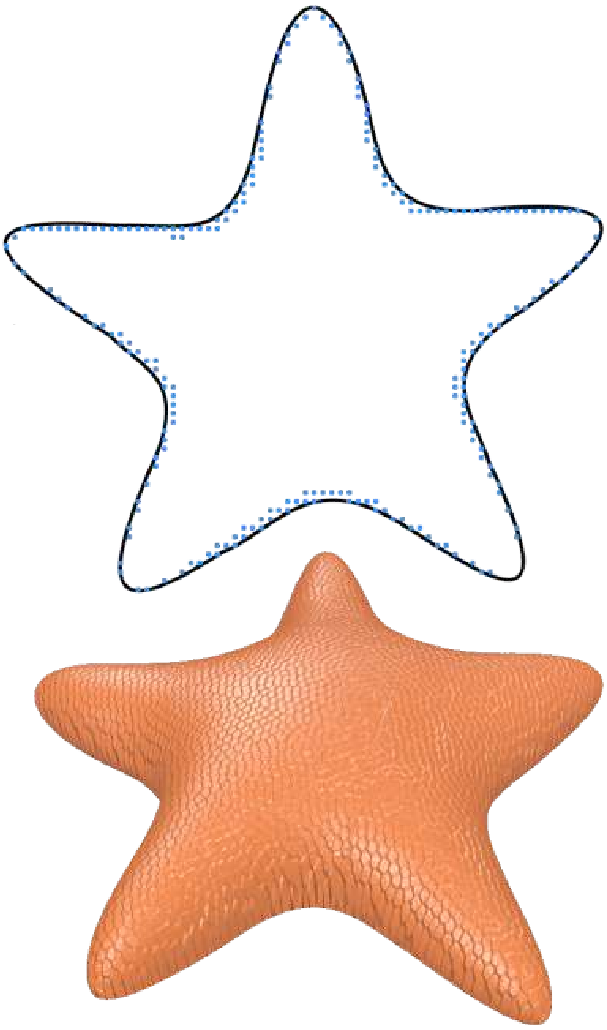


Figure 66: Starfish reconstructed from a subset of data closer to the contours. The center of the sketch does not contain points or normals to be interpolated.

It can be observed in Fig. 66 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.

The attachments on the final model can be smoothed. To this end, we group each part and merge it using a Screened Poisson surface reconstruction [69]. We apply this reconstruction in the goat, rabbit and bird models, shown in Figs. 67, 68 and 69, making them smoother at junctions.

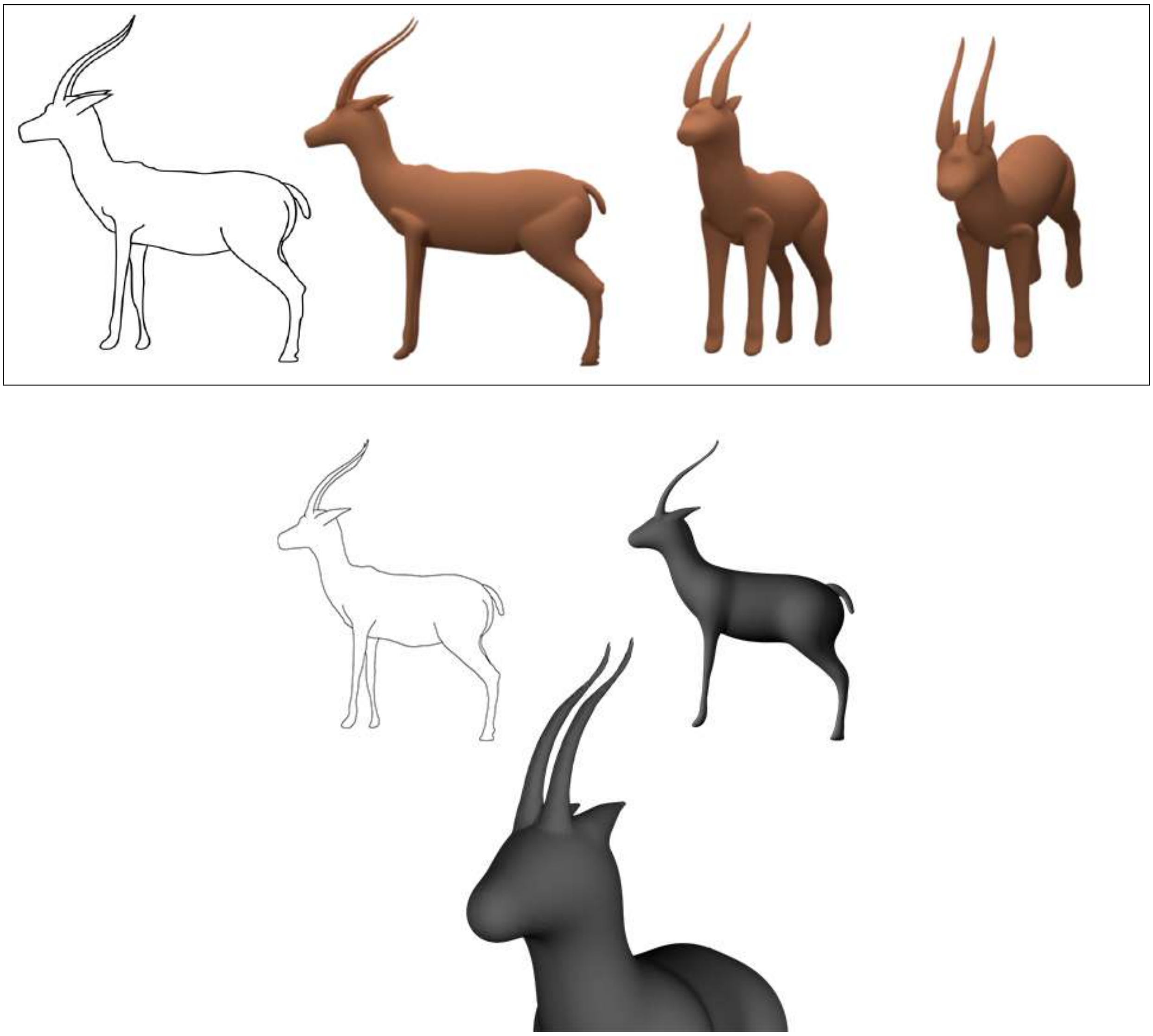


Figure 67: Goat input drawing and its 3D reconstructed model. Our approach (top / highlighted). Results from the same drawing reconstructed by Entem et al. [31] (bottom).

The Figure 67 show a goat 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.

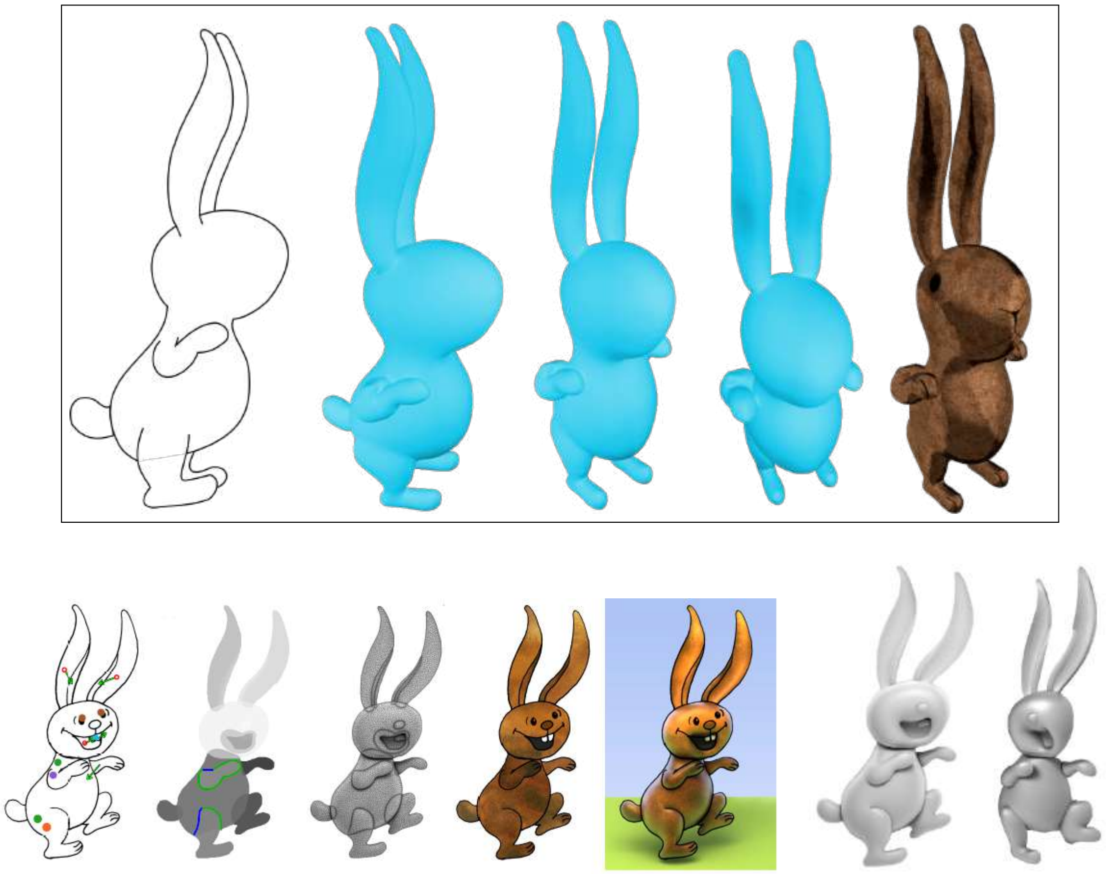


Figure 68: Rabbit generated by our method. Results from approach presented by Sỳkora et al. [116] (bottom) compared with our result (top / highlighted).

Regarding the reconstruction of the rabbit's paw in Fig. 68, 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. In the rabbit's paw (Fig. 68), 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. Notice that the proposed approach differs from previous works that create 3D-like reconstructions or high-relief reconstructions, instead, our final result is a complete 3D model.

The input design of this model does not contain an adjacent feature cycle having only one adjacent border cycle, resulting in a rounded mesh differently from the elephant ear shown in Fig. 60. Notice that the bird's wing is an island cycle (Fig. 69), 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.

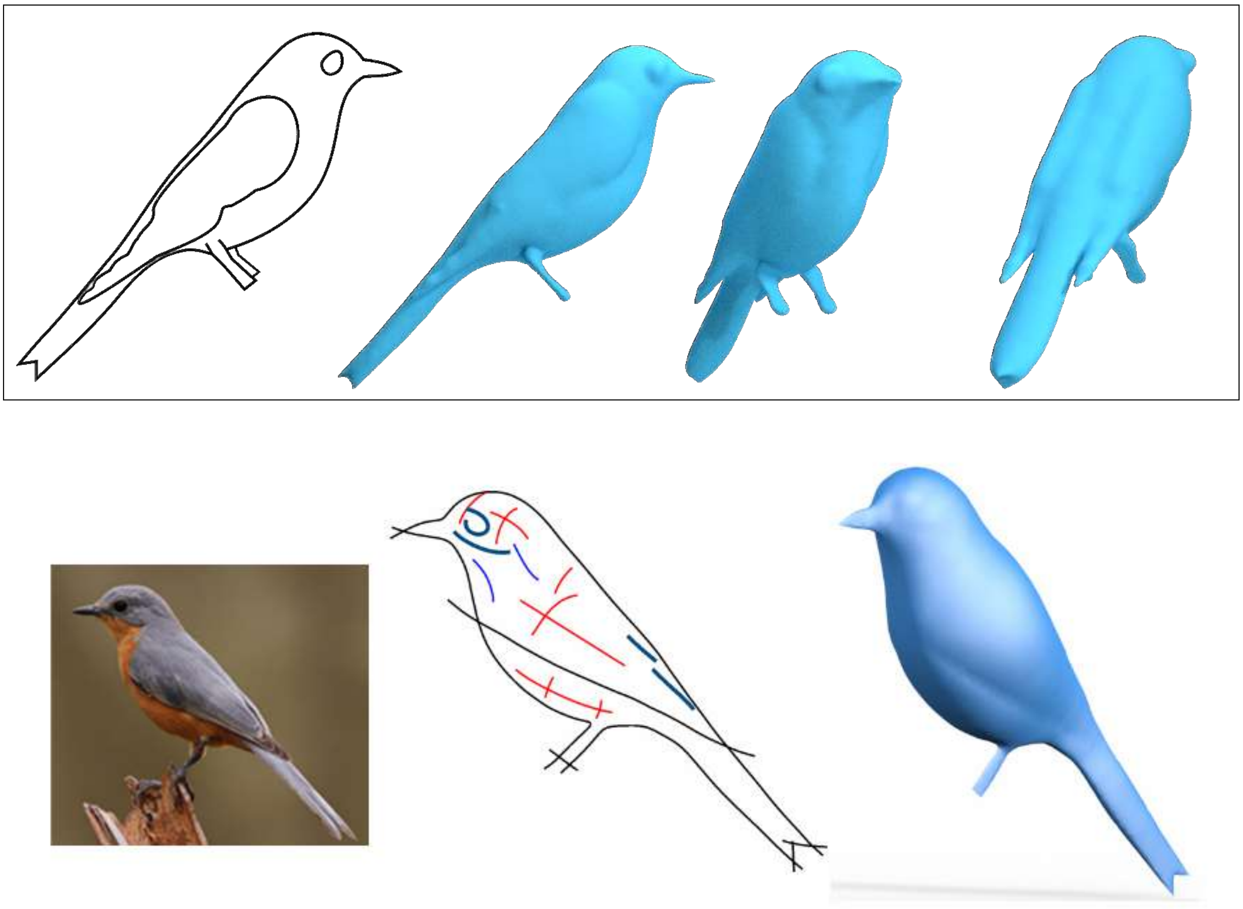


Figure 69: Result from merging and smoothing processes in a bird model. Our approach (top / highlighted). Results from the same drawing reconstructed by Li et al. [82] (bottom).

Our approach has the potential to be applied to the reconstruction of further objects, symmetrical or not. In Fig. 70, we present other objects reconstructed by our method.

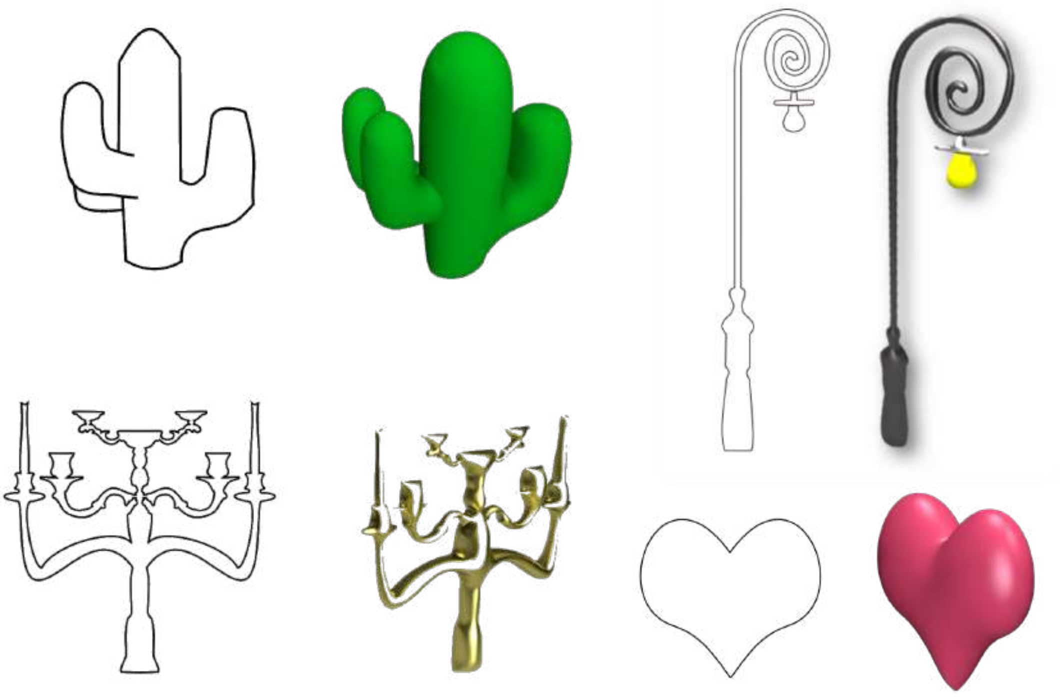


Figure 70: Our approach has the potential to be applied to the reconstruction of a variety of objects, symmetrical or not.

Our method only 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. 70. 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. 70.

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.

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.

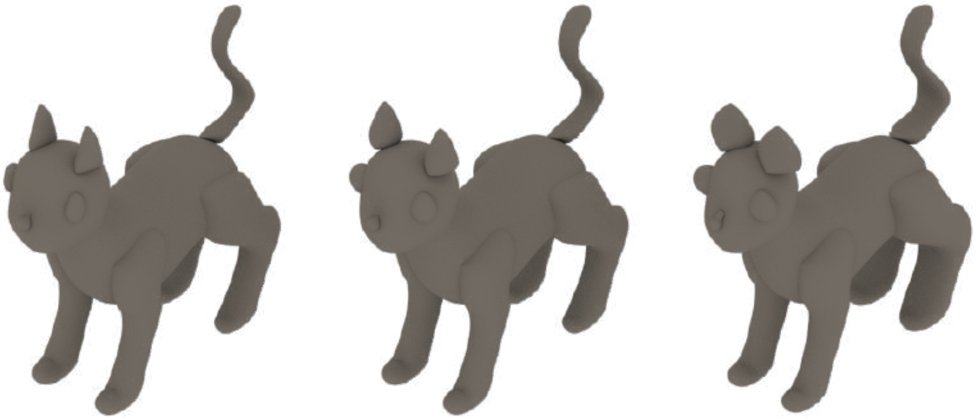
	Points	Normals	Matrix Order	Time(s)
Goat	1024	1024	4,100	1.29s
Cactus	214	214	860	0.12s
Candle Holder	342	342	1,372	1.18s
Rabbit	794	794	3,180	0.60s
Dinosaur	968	968	3,876	1.64s
Elephant	984	984	3,940	3.13s
Cat	1433	1433	5,736	2.70s
Bird	348	348	1,396	0.31s
Fish	658	658	2,636	1.27s
Penguin	927	927	3,712	1.81s
Unicorn	1078	1078	4,316	2.13s

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. 71c). In this case, the HRBF interpolation extrapolates the actual size of this part because it does not have enough points sampled in the contour.

Table 2: Time to solve the linear system for different sampling rates in the cat model.

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

Fig. 71 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) 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.

Figure 71: 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.

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. 71c is incompatible, unlike the ear in Fig 71a. 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.

Despite the promising results, some issues remain to be investigated in the future:

- Symmetrical limbs must 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. 72 in contrast to the cactus, shown in Fig. 73, in this case, the foreground part (lighter blue) is paired with the closest border cycle (darker blue);

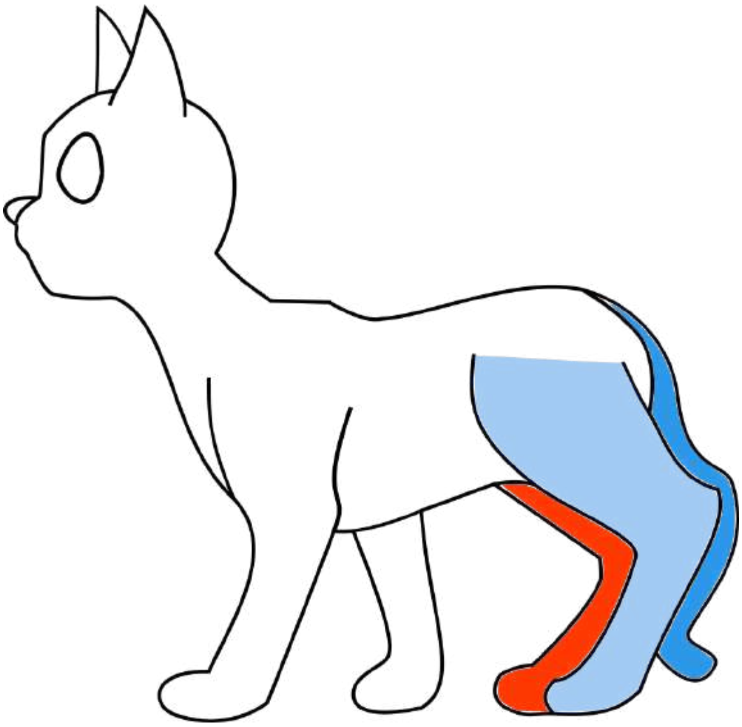


Figure 72: Failed classification when parts are drawn too close. In this case, the method interprets the tail as a leg.

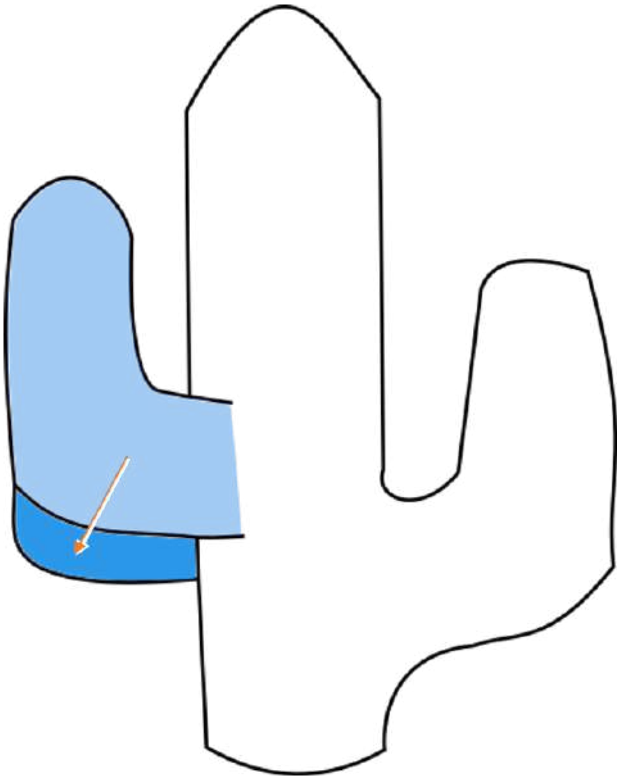


Figure 73: Classification success for cactus drawing, the symmetrical limb is paired with the closest border cycle.

- 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;

- 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. 74 shows an elk's horn in side-view and its 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.

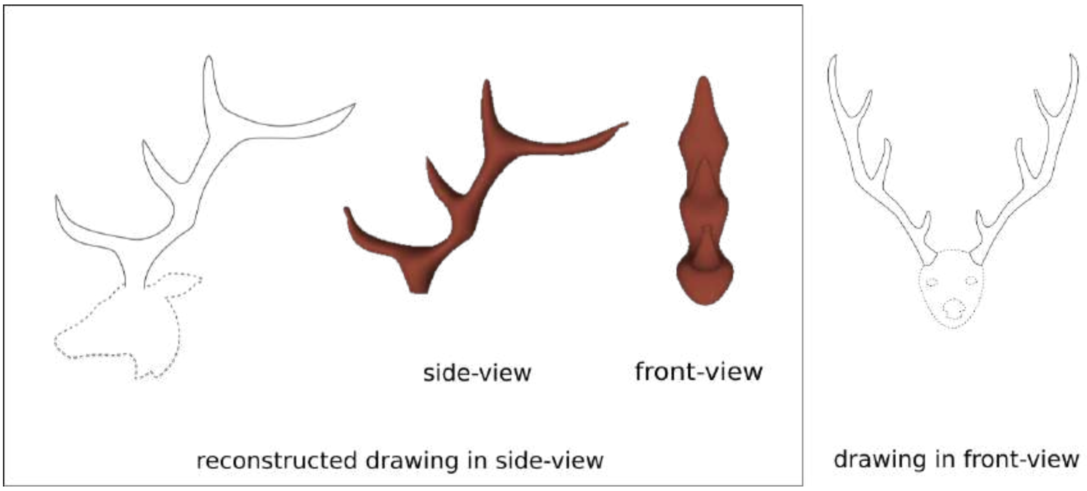


Figure 74: Curved parts in the normal direction cannot be inferred. From left to right: Elk's horn in side-view, its reconstruction, and its rotation to the front-view. One could expect a curved horn in the reconstructed front-view, as depicted in the front-view drawing (right).

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.

3.8 Chapter Remarks

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 chapter, 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 for 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, the work presented in this chapter is the first work to explore the problem of reconstruction from a side-view sketch with a skeleton-free technique based on scattered Hermitian data. Besides, 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 [119], Biharmonic Fields [8] or Laplacian Reconstruction [38], could be exploited to achieve smoother surfaces.

A direction for future work is to use drawing annotations to indicate various types of surfaces (Fig. 70) 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 [83]. 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.

4 SKETCH-BASED MODELING SUPPORTED BY 2D VISUAL PERCEPTION ENHANCEMENTS

The framework presented in Chapter 3, decompose sketches into parts to define depth and 3D shape for reconstruction for each part. We proposed contributions to the state-of-art in automatic algorithms. However, imposing some constraints and using structured 3D drawings, limits its use by reconstructing 3D models from a small variety of drawings. Thus, the user expertise plays a key role to infer absent occluded parts and depths. Although approaches using machine learning were proposed, the main concern is the lack of datasets composed of scientific drawings to train the networks.

These limitations led our research to a further investigation, the creation, and evaluation of an interactive framework presented in the following. This framework enables the user to draw overlaid contours based on principles and practices of traditional illustration and Non-photorealistic rendering, allowing to perceive symmetries and occluded parts to the 3D reconstruction. The system was then applied and analyzed to the 3D reconstruction of biological systematic illustrations.

4.1 Introduction

Images are fundamental to every scientific domain that deals with descriptions. This aspect is particularly true for biological systematics, which is the field that studies and describes the living forms on Earth, both present and past [58, 99]. Besides the taxonomic practice itself, the educational perspective is an underlying reason for using scientific illustrations [2, 56, 77, 89]. Landin [77] discussed the benefits of drawing in science class. More recently, Merkle [89] revisited this topic arguing that professors do not need to be real artists to provide useful drawing experiences for their students.

Despite the technological evolution of 2D and 3D imaging approaches, the demand for scientific drawings remains, in particular, to describing new specimens [34, 106]. For instance, scientific illustrators are increasingly using 3D modeling approaches as part of their production pipeline [56]. In general, one of the challenges of scientific illustration is to preserve the geometrical features of the biological structures when dealing with drawings representing both external and internal morphology.

Even though photographic cameras, scanning electron microscope, and other devices solve part of the necessary task, there are some essential aspects of interpretive biology work that still require actual drawing. The decision to emphasize specific parts of the studied object depends on the researcher's goal, e.g., focusing the drawing on insect abdominal structures, mouthparts, or wings. These choices lead to explore drawing elements that indicate depth, occlusion, textures, inner structures, or surface details. Consequently, relying on the use of drawing elements, e.g., color palette, depth simulation, shadowing, hatching patterns, or stroke styles. When entering in computer-based sketch-based modeling, a very common and powerful ingredient is added: the 3D reconstruction of the sketch, leading to real-time interaction.

Recent automatic and semi-automatic approaches for 3D reconstruction infer depths or discard occluded parts in single-view drawings [31, 101], leading to the use of multi-view systems to reproduce complex overlay drawings [61, 82]. Other approaches interactively create overlays applying twisting and folding operations to existing models [60, 127].

4.2 Overview

In our approach, input strokes are sketched using a mouse or pen-tablet (Section 4.3). These strokes can be part of three categories: open contours, closed contours, and stripes (Fig(s). 75a, 75b and 75c).

Our framework (Fig. 75) considers layers as features of sketches. Thus, to enhance the perception of different layers in single-view drawings, effects are applied for contours drawn to offer visual feedback for different parts of the object. These non-photorealistic rendering (NPR) effects [48, 94, 114] are important, especially on single-view drawings, where the depth perception is a desirable feature (Section 4.4).

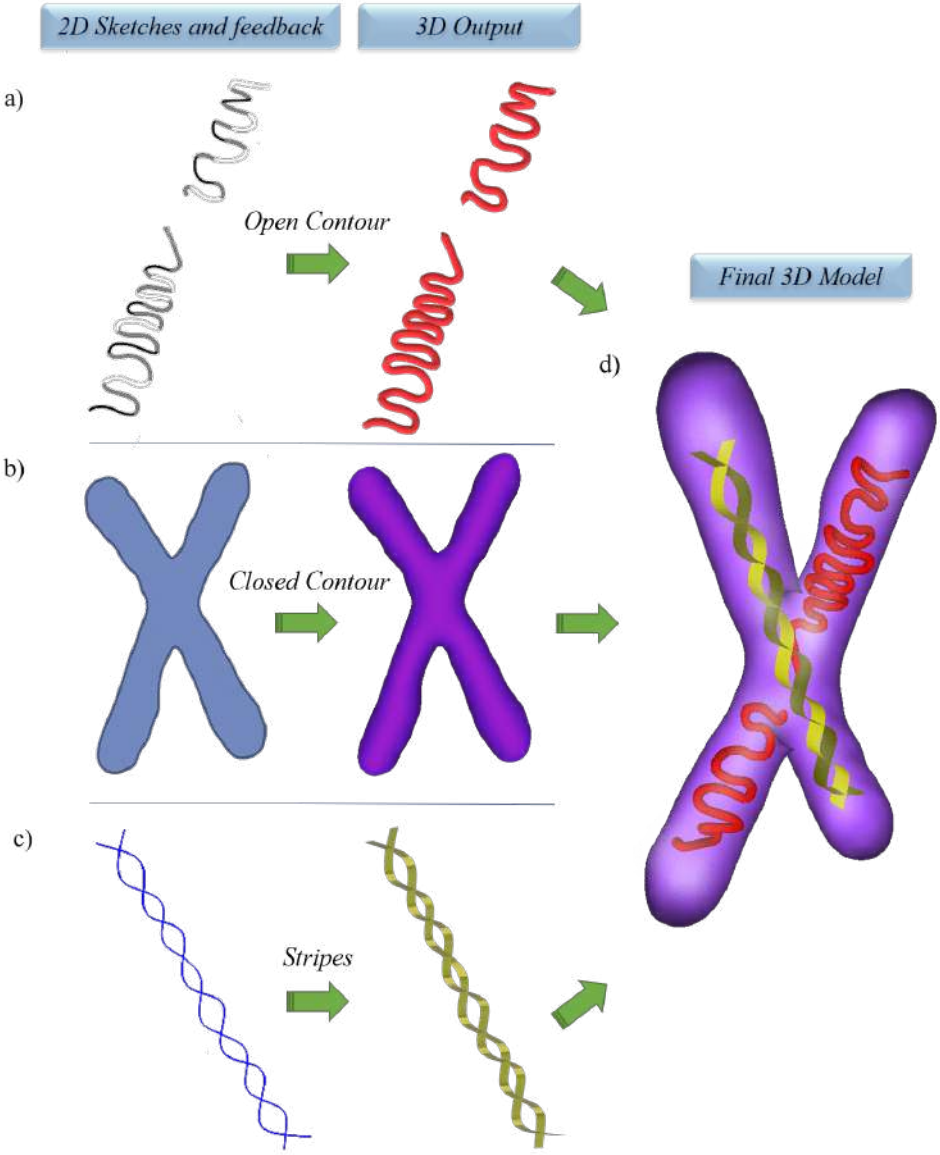


Figure 75: Overview of our system and contour categories supported as inputs for the reconstruction of an illustrative chromosome. Input strokes are acquired through the user interface (with a mouse, for instance) belonging to three categories that differ in their 2D visual perception and 3D reconstruction method: (a) open contours and their respective 3D reconstruction; (b) closed contours and 3D surface reconstruction; (c) Stripes sketched using our system and interactions and its 3D reconstruction; (d) final 3D model combining the three categories.

In order to combine 2D strokes with 3D depth information (Section 4.5), open contours are modeled creating cylinders for each pair of points sampled on the stroke while smoothing their connection between layers (Fig. 75a). Closed contours are reconstructed as closed shapes, creating their 3D form using a Rotational Blending Surface [24] or normal-driven estimation with Hermite-RBF Surfaces [43, 101] method (Fig. 75b), and, stripes are build by creating a 3D mesh points sampled throughout their described shape (Fig. 75c).

To provide an easy and fast 2D representation and a reliable reconstruction of 3D objects, our framework combines 2D and 3D information to enhance the user experience and offers the possibility to complement previous works that infer depths or discard occluded parts.

As a demonstration application, our framework was explored to model entomological features often present in species descriptions of the order Diptera, one of the mega-diverse group of insects popularly known as flies and mosquitoes, but the solutions are useful for dealing with any other biological group.

4.3 Creation Phase

This is the first stage of interaction with our system (Fig. 75). In this first stage, the user sketches three groups of construction lines according to the desirable 3D final shape: open contours for knots and strings, closed contours for 3D surfaces, and stripes for twisting tapes or bands.

Our system captures the input pixels in the screen while the user sketches each line using a mouse or pen tablet. Each of these points is sequentially added to a stroke object assigned to each separate construction line group.

To enhance the contours for reconstruction, the user can select, edit, and refine the strokes using a smoothing or an oversketching operation. The design of these SBIM interactions was inspired by previous works [16, 24] as follows.

The **smoothing** operation is necessary due to the noise on the acquisition of strokes (Fig. 76a), a Reverse Chaikin Subdivision is then employed to create a noiseless B-spline for each stroke (Fig. 76b). This subdivision is based on a quadratic B-spline, and the coarse information can be interpreted as control points of a quadratic B-spline curve. Denoting the fine points by $p_0, p_1, \ldots, p_n$ and the coarse points by $q_0, q_1, \ldots, q_m$ then the general case of the reverse Chaikin scheme, as denoted by Samavati and Bartels [104], is as follows:

$$q_j = -\frac{1}{4}p_{i-1} + \frac{3}{4}p_i + \frac{3}{4}p_{i+1} - \frac{1}{4}p_{i+2} \tag{9}$$

where the step size of i is one. Then, is applied a mean filter denoted by:

$$q = \frac{p_{i-1} + 3p_i + p_{i+1}}{5}, \tag{10}$$

where q is the point in the new curve. Notice that the reverse Chaikin subdivision contains very simple operations, which has been proven to be efficient for real-time applications [10, 24, 104] and minimizes the noise problem as shown in Figs. 76a and 76b.

This reverse subdivision scheme is applied whenever the user desires to enhance strokes reducing noise and complexity. Although it deviates further from the original stroke each time the reverse subdivision is applied, the resulting curve becomes smoother. Figure 76b, and Fig. 76c show a contour smoothed three and nine times respectively.

Another operation is the **oversketching**. This operation allows the user to redefine the boundary of shapes as well as correct or augment shapes. As proposed by [16], first, the user selects the desired contour in which the operation is going to be applied. Then, the user draws a new curve S near this existing shape contour C (Fig. 76d). The old curve

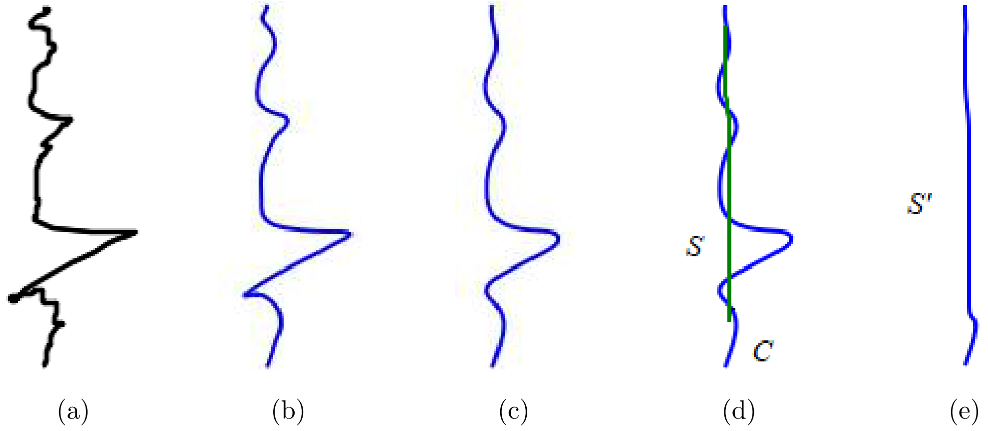


Figure 76: (a) Raw stroke acquired through the user interface (with a mouse, ou a pen-tablet for instance); (b) default smoothing operation applied three times over the raw stroke; (c) smoothing operation applied nine times over the raw stroke; (d) oversketching operation performed by drawing a new curve (green curve) nearby the contour to be modified on canvas; (e) curve after replacing the current sketch with the new curve with points connected on oversketching process.

C is then replaced by S' connecting the extremities to their closest points on the original shape. The ambiguity on which piece will be replaced is solved enforcing coherence on the orientation of the final curve as shown in Fig. 76e.

Besides the hand-drawn contours directly in our system using a mouse or pen tablet, in order to reconstruct closed contours, our framework supports as inputs scalable vectorial graphics (SVG) files composed of Bézier curves, allowing the user to select contours to be reconstructed through sketch operators.

Our framework provides two sketch operators, used to select contours to be reconstructed. This selection is performed by a cross line interaction or a brush line interaction. A **cross selection** is done by stroking another line over the Bézier curve as shown in Fig. 77a.

The second sketch selection operator is the **brush selection**. This operator seeks to create closed contours with a set of line segments, commonly used in drawings to depict an

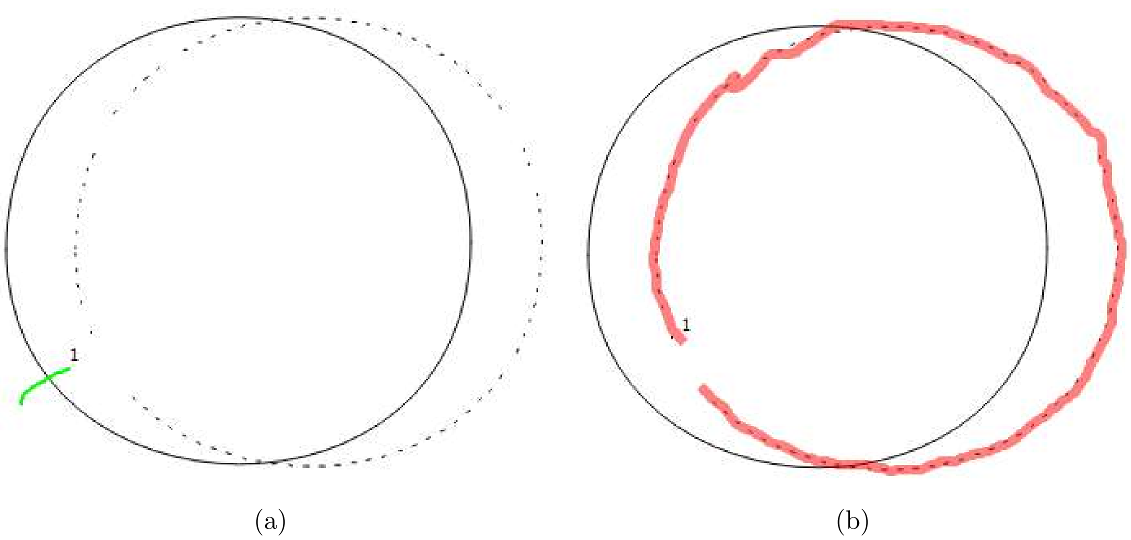


Figure 77: Selection operators implemented in our framework: (a) cross selection operation (green stroke) for a closed contour; (b) brush selection (red stroke) by painting in order all segments belonging to the same contour.

occluded part or contrast important parts. To this end, the user draws a new line over the segments to connect them orderly following the shape silhouette as shown in Fig. 77b.

We ensure that the strokes selected through these operations describe closed loops by connecting the first point of the contours to its last point by creating a new Bézier curve that connects their extremities, guaranteeing C^1 continuity.

Still, our system provides an **erase operator**, used to delete non-necessary curves. This operator works similarly to the cross selection operator, but instead of selecting contours to be reconstructed, this operator removes the curves from the canvas.

The last category of contours is stripes. Stripes have the same width throughout its length, alternating in color or texture at either side of its surface. Our framework allows the user to sketch 3D stripes trough 2D sketches using hotkeys and mouse interactions to draw 3D stripes in different layers. Stripes are created as follows.

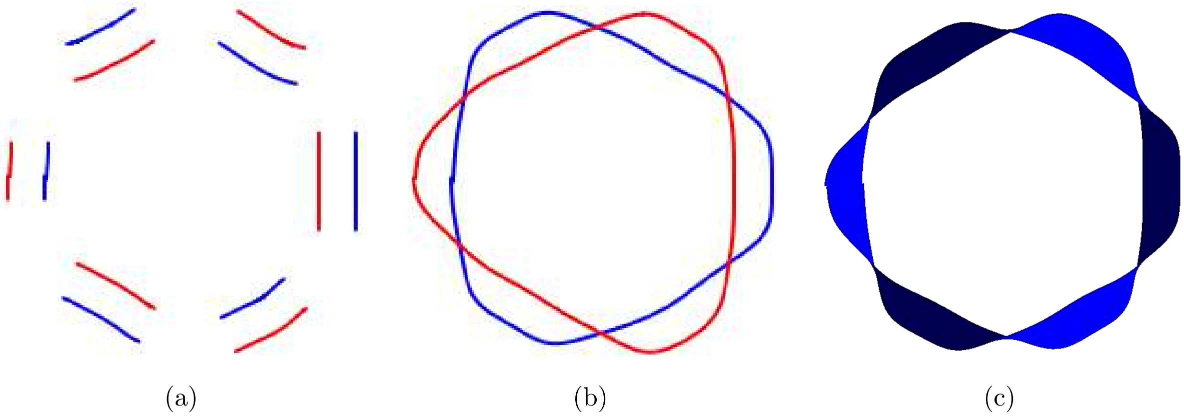


Figure 78: Reconstruction process of a 3D stripe: (a) lines drawn with twist annotation using inverted colors for stripe sides; (b) automatic creation of stripe twists by connecting the contours; (c) 3D parametric reconstruction of the stripe.

The user starts drawing a contour and while drawing, the framework show guidelines following the stripe silhouette in two colors, blue and red, as shown in Fig. 78a. A twist or stripe inversion is created by releasing the mouse button and starting again the drawing interaction, automatically our framework inverts the lines (red and blue) and connects the new lines to the previous strokes when the user finishes the sketch by clicking with the mouse right button (Fig. 78b).

4.4 Visual Enhancements: Layer and Depth Effects

In the second interactive session of our system, the user can modify and include depth effects on drawings. Depiction of overlays and depth is usually achieved by a set of drawing techniques guided by principles of human visual perception found in traditional illustration [24, 33, 67, 113, 129].

Visual perception is a broadly explored area. Fine et al. [35] define three categories of 2D perception: form, depth and motion. Usually, algorithmic approaches implementing visual perception principles employs patterns of image shading, patterns of lines that depict occlusion contours and edges, gradients of textures as dots or parallel lines in surface silhouettes [117].

Our visual perception system is wired in a way we can easily distinguishing partially occluded shapes or their respective depth. However, devising an automatic SBIM approach to select one specific solution for occluded parts of a model can be challenging [97]. This problem is due to the visual ambiguity caused by different occlusion and depth configurations in different parts of a model. We addressed this challenge by employing a hybrid approach with a set of automatic techniques driven by sketch-based user input and intent. This set of techniques are inspired by traditional illustration practices related to the principles of form and depth perception, in particular, object occlusion, transparency and sketch inference in overlaid contours.

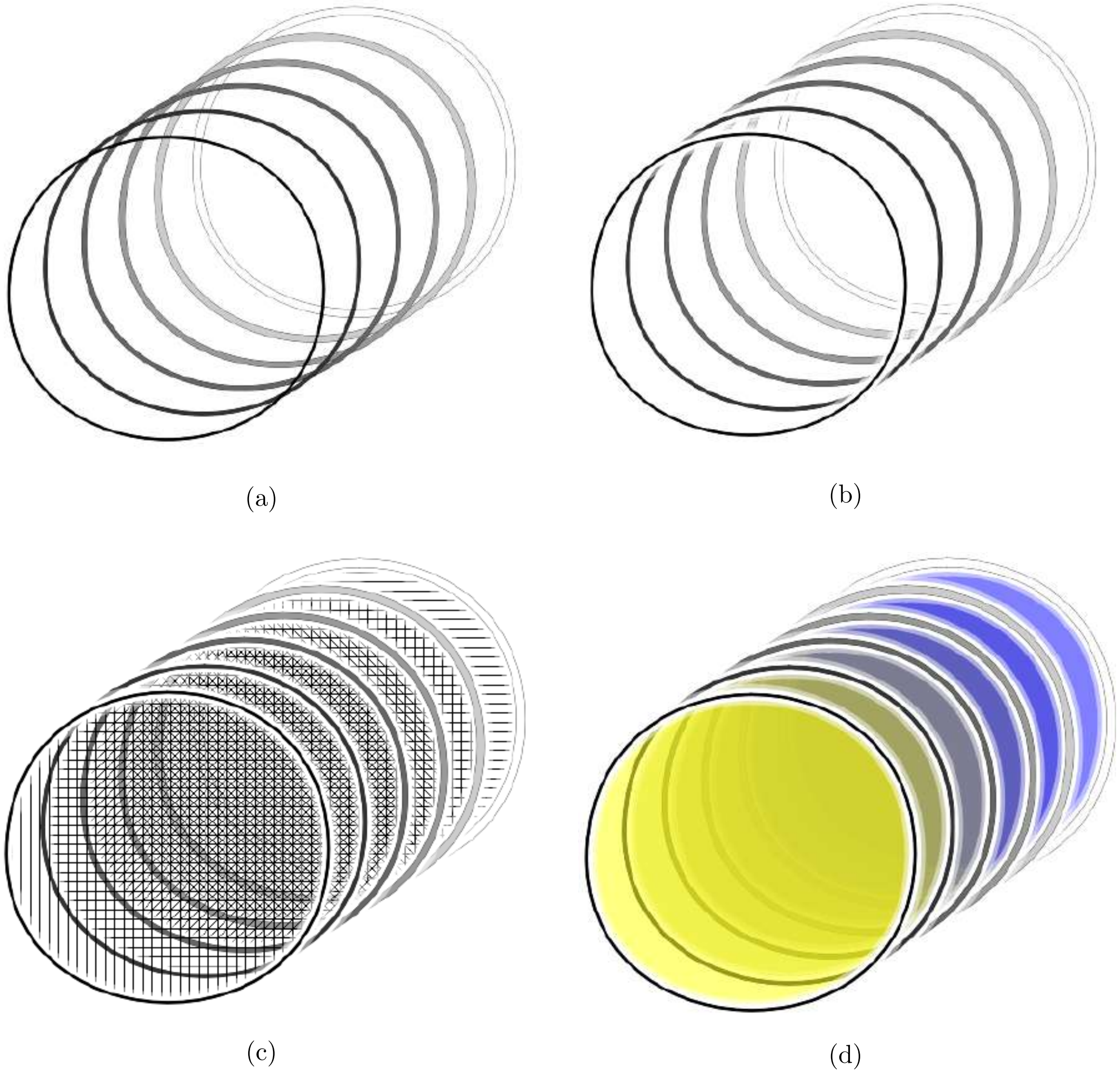
Our system allows the user to define the depth of layers providing a flexible tool layered drawing as shown in Fig. 79.

To increase the depth perception and aid the users, the following visual effects, exemplified in Fig. 79, are provided by our framework as real-time feedback:

1. Generation of a color map to improve the depth perception of shapes drawn in different layers, conveying transparency alongside a color tone shading that avoids the occlusion of layers further to the user and, provides a balanced visual painting for layers closer to the user [11, 121]. Our color map employs color temperature, a quality of color used by artists (e.g., [47, 90]). Refer to Section 4.4.1 for more details.

2. Enhanced depiction of areas using hatching lines (Section 4.4.2), commonly used in arts for illustrations usually conveying texture, tone, and parallel lines to create shading effects or used to indicate different shapes [129].

3. Sketch inference for closed contours to indicate overlays for contours drawn in the same layer and aid the user while sketching new strokes (Section 4.4.5).



(a)

(b)

(c)

(d)

Figure 79: An example of different depth effects after selection and layering. (a) Contours painted using a gray-scale to depict different layers. (b) Example of Halo Effect to emphasize the layers. (c) A hatching effect is applied to different layers, automatically creating a cross-hatching visual effect on overlapping layers. (d) Layer coloring varying from pure blue to yellow, based on traditional artistic principles and practice of using color temperature to indicate different depths in a scene (Sec. 4.4.1).

4. Creation of depth-dependent halos to increase the depth perception of overlapping lines (Section 4.4.3). Further, illustration examples include electric or magnetic field lines, protein structures, and meteorology [33].

5. Tone-based emphasizing of contours varying from darker and lighter tones on gray-scale to enhance the depth perception for layers [18], as well as, weight-based thickening of contours to improve the quality of the final stroke (Section 4.4.4). It is a common effect used by illustrators for creating perceptions based on depth [88, 95, 113].

4.4.1 Layer Coloring

As described above, in our framework, layer coloring is based on a color temperature ramp using a scale of cool colors varying from pure-blue to yellow. Color temperature is a quality of colors employed by artists and it is defined as a color ramp ranging from warm colors (e.g., red, orange, and yellow), to cool colors (e.g., blue, violet, and green). Our visual system is wired in a way that we naturally perceive warm colors advancing and cool colors receding. Artists rely on the juxtaposition of warm and cool colors to provide a depth cue (e.g., [47, 90]). We create a color map for layers starting in pure-blue for the bottom layer, that is, the layer further to the viewer's perspective, to yellow for the top layer as shown in Fig. 79d.

$$Color_{\{r,g,b\}}(\ell) = Start_{\{r,g,b\}} * r(\ell) + End_{\{r,g,b\}} * (1 - r(\ell))\,, \tag{11}$$

where

$$r(\ell) = \begin{cases} 1, & \text{if } n = 1 \\ 1 - \frac{\ell}{n-1}, & \text{otherwise} \end{cases}$$

and $\ell = (0, \ldots, n)$ n corresponds to the number of layers, $Start_{(r,g,b)} = (255, 255, 0)$, $End_{(r,g,b)} = (0, 0, 255)$, and r defines the ratio between layers and tones. Still, in our tests, a transparency of 50% is sufficient to avoid the occlusion of layers further from the user.

4.4.2 Hatching Lines

Hatching Lines fill areas of contours, usually conveying texture and tone commonly used in traditional illustration (e.g., [53, 129]). We use hatching lines to indicate different contours and layers, varying their angle according to the order that the closed contours are drawn, creating automatically a cross-hatching effect on overlapping closed contours, as shown in Fig. 79c.

Our framework paints hatching patterns in four patterns: vertical, horizontal, and two diagonals of opposite angles. We start from vertical hatching on the top layer, horizontal in second layer, and diagonals in the third and fourth layer, repeating this order thereafter.

4.4.3 Halo Effect

Previous works have employed shaded lines, tubes, or ribbons whose color and shape can be changed depending on data properties, *i.e.*, flow, speed, or direction [100]. A halo is created by showing gaps, white spaces that obstruct nearby lines [33]. This halo effect was first proposed by Appel et al. [7].

The halo effect is painted when white spaces are shown to depict gaps while obstructing nearby lines. In our system, these white spaces are created by drawing two thicker white lines for each contour. The former is a solid white line painted by adding $9px$ to their layer number and, the latter is a line painted with 50% of transparency and $16px$ added to its layer number.

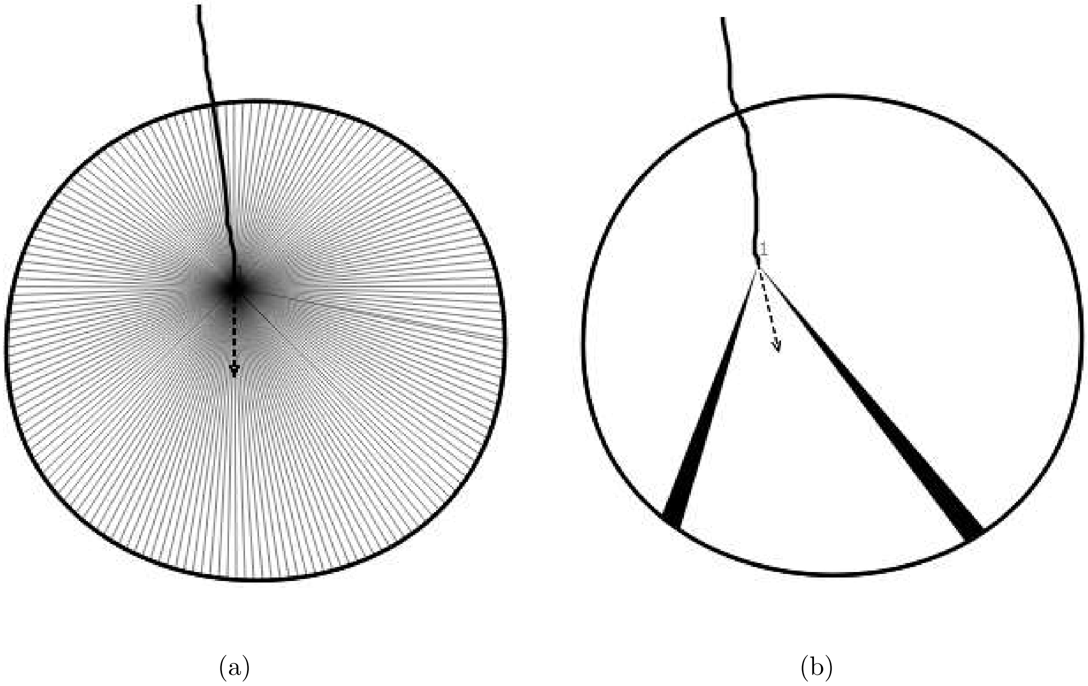
4.4.4 Contour Shading and Thickening

To create the visual perception of layered contours, we developed a tone-based emphasizing varying from black, in the top layer, to white in the bottom layer and a weight-based thickening approach for its contours. These contours were drawn by the user on the creation phase (Section 4.3), respectively, closer and further from the viewer's perspective following the artistic illustration proposed by Burton [18]. In our prototype system, the user paints these contours on canvas in back-order, according to their layer level starting from white on the bottom layer to the top layer as shown in Figure 79a. The Equation 11 defines the color map created using a gray-scale in which each layer has a different tone employing values $Start_{(r,g,b)} = (0,0,0)$, $End_{(r,g,b)} = (255,255,255)$. The user also paints contours on canvas with different thicknesses according to their layer order. The line width is defined by adding $1px$ to the layer number, that is, for layer $n = 1$ the line width is $2px$, while for layer $n = 2$ the line width is $3px$ and so on. Besides that, we draw a thin black line around all contours delimiting them as the effect used by illustrators. This effect is created by painting a thicker black line under the layer line. The thickness of the line is defined by adding $1.5px$ to the layer number.

4.4.5 Sketch Inference

Sketch inference is shown by our system while drawing overlaying contours at the same layer. This inference is performed using the last points drawn, estimating the next points of the curve and their possible sketch area as shown by the headlight beam effect in Fig. 80b, aiding the user on possible next sketch location given the ongoing trajectory of the current sketch. The effect is created by sampling two angles in the drawing curve, the angle in the last point and the angle in the point sampled with $5px$ of spacing starting from the last point. The average of these angles projects a line expressed as an arrow shown in

Fig. 80b. Using the calculated angle, we paint a ”headlight beam” seeking to enhance the sketch inference. The ”headlight beam” effect is created as follows.



(a)

(b)

Figure 80: Sketch inference provided by our framework: (a) triangles created combining a pair of points samples on contour and the last point drawn; (b) a ”headlight beam” effect created by painting triangles according to their angle in relation to the last points of the current stroke.

First, we compose a set of triangles by sampling pairs of points on the contour that is being overlaid and connecting these points to the last point of the contour that the user is sketching Fig. 80a. Then, we calculate the difference between the angle of the starting point for each triangle and the angle calculated previously to create the arrow. Lastly, we paint triangles in which the angle difference in absolute values fall into 28° and 32°, as shown in Fig. 80b.

4.5 3D Reconstruction Phase

At this stage, our system creates a 3D representation directly from the 2D contours sketched by the user as shown in Fig. 75. Three main features of sketched contours are considered: (1) category, (2) silhouette, and, (3) addressed layer.

The first category, composed of open contours, describes knots and strings whereas the second category, named closed contours represents 3D surfaces and, the third category describes stripes, a flat shape that can be twisted. The reconstruction steps for each of these three categories are described in the following.

4.5.1 Open Contours

Open contours are used to describe strings or knots, therefore, discarding the need to reconstruct them as 3D surfaces. Thus, these contours are reconstructed by creating 3D cylinders based on points sampled throughout the sketched strokes.

The smoothing between layers is done as follows. First, the is applied for each contour. As the reverse Chaikin subdivision scheme was performed on creation phase considering 2D points (x, y) (Section 4.3), now for 3D reconstruction, only points on $z - axis$ are considered.

Applying this filter returns a smoother 3D line between layers in our results, as there are no points between the layers at the first smoothing operation. The real-time 3D view with lines addressed to different layers is given by generating cylinders for each pair of subsequent 3D points.

Figure 81 shows the smoothing operation using the z-axis of the contours for one, ten, fifty, one hundred, and two hundred times. The system enables users to define the desired number of operations as there are no points between lines in different layers on the first operation and the expected result may vary considering the depth of the layers.

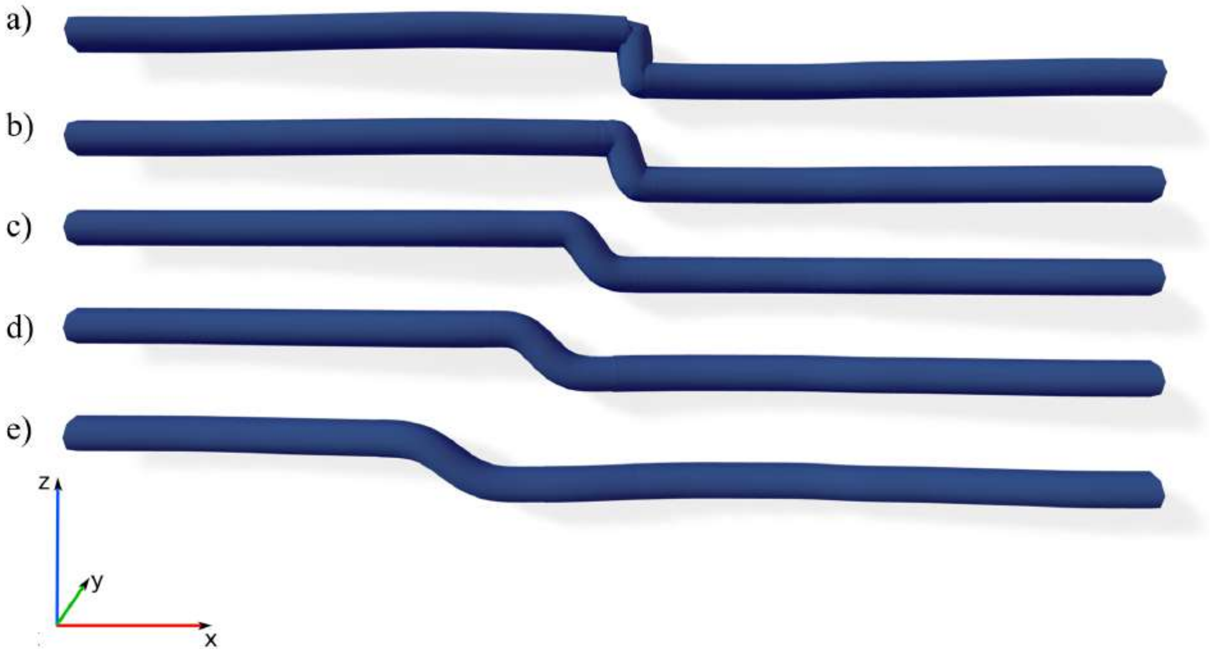


Figure 81: Comparison of smoothing operations applied to $z - axis$ of an open contour with two layers. (a) one time, (b) ten times, (c) fifty times, (d) one hundred times and, (e) two hundred times.

4.5.2 Closed Contours

Closed contours describe 3D surfaces, it means that these contours forms closed loops and do not intersect themselves. Thus, a shape estimation process is performed in preparation for the 3D reconstruction. Models are reconstructed as desired through the Rotational Blending Surface proposed by [24] or Hermite-RBF Surfaces reconstruction proposed by [43].

The second goal in this phase is to prepare and visualize data for 3D model reconstruction. For instance, our framework allows the user to define a desirable medial axis for shapes in which the Rotational Blending Surface Reconstruction is going to be performed or visualize 3D Hermitian data for shapes in which the Hermite-RBF Reconstruction is used. Our framework allows the user to visualize the expected 3D model using a set of 3D points generated by parametric equations or normal-driven estimation.

Parametric reconstructions are proven to be faster and better on geometric forms that do not contain branches or ramifications as organic structures. We use a Rotational Blending Surface for parametric reconstruction as this type of surface requires only two contours representing the silhouette edges of a 3D form [24]. These silhouettes are combined with the a surface of revolution to find the parametric their description.

The creation of a rotational blending surface is done as follows. First, we sample equally spaced points on closed contours $q(u)$, these points are then split in two sets $q_\ell(u)$ and $q_r(u)$. We use these sets of points as constructive contours of the rotational blending surface.

Let $\wp$ denote the plane and $c(u)$ be the medial axis formed by midpoints between $q_\ell(u)$ and $q_r(u)$ at each u. Then we parametrize the circle perpendicularly to $\wp$ with center $c(u)$ containing the points $q_\ell(u)$ and $q_r(u)$ at each u, as proposed by Cherlin et al. [24]:

$$t_u(0) = q_\ell(u),$$

$$t_u(\pi) = q_r(u),$$

$$t_u(2\pi) = q_\ell(u).$$

We interactively change the medial axis defined by midpoints from $q_\ell(u)$ and $q_r(u)$ rotating the vectors in which the points u are stored with a fixed v that represents the rotation angle (Fig. 82).

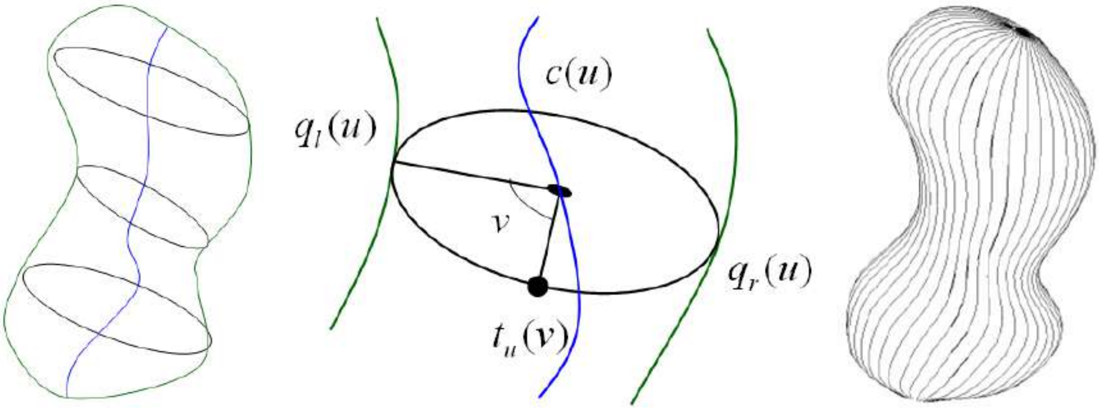


Figure 82: Rotational Blending Surface Reconstruction. The $q_\ell(u)$ and $q_r(u)$ curves are constructive curves for the medial axis $c(u)$ creating a circular slice of the surface, denoted by $t_u(v)$. Source: Cherlin et al. [24].

Algorithm 2: Rotational Blending Surface

Input: q_ℓ, q_r

Output: 3D Mesh Points

```
$n \longleftarrow$ FindPlaneNormal $(q_\ell, q_r)$
$u \longleftarrow 0$
while $u \leq 1$ do
    $v \longleftarrow 0$
    while $v \leq 2\pi$ do
        $s_0 \longleftarrow \frac{q_\ell(u)+q_r(u)}{2}$
        $s_1 \longleftarrow || q_r(u) - q_\ell(u) || \times n$
        $p_\ell \longleftarrow$ RotatePointAboutLine $(q_\ell(u), s_0, s_1, v)$
        $p_r \longleftarrow$ RotatePointAboutLine $(q_r(u), s_0, s_1, -v)$
        $t \longleftarrow (1-t) * p_\ell + t * p_r$
        AddPointToMesh $(p)$
        $v = v + v_{STEP}$
    $u = u + u_{STEP}$
```

The rotational blending surface can create a variety of models [24]. However, previous studies [16] showed that RBF Surface Reconstructions presents good results with objects in which the final model has an uncertain shape or contains organic structures as branches or ramifications or models that contain a reduced number of input points [87], these organic structures cannot be handle by the Rotational Blending Surface for not owning a medial axis that can describe the whole object without segmentation.

First, we use the same sampled points in the contours for each part and estimate their 3D normals. Then, by propagating the normals to the interior, we determine their shape. Specifically, we aim to sample the interior of the contours. To this end, we define a regular grid sampling points in the interior of the contour on each vertex [128]. 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. The last step is to define the depth of the points, generating a set of 3D generalized Hermitian data. We determine the depth of the points by multiplying n_z of each point by the width of the bounding box of each closed contour, adjusting the depth of the points as proposed by [101].

For interactive feedback, our framework generates a 3D view of generated points from both approaches, allowing the user to confirm the expected desirable shape of the drawn object. The 3D model is then reconstructed using the previewed 3D points. To this end, we use the 3D dataset generated as input for *i*) a quadrilateral mesh to build upon the 3D points generated by Rotational Blending Surface or *ii*) points and normals constraints for Hermitian Radial Basis Functions (HRBF) Surface reconstruction. The generated dataset in both approaches contains points to create closed surface meshes, thus, concave and convex surfaces are created to generate closed shapes.

4.5.3 Stripes

Similarly to open contours (Section 4.5.1), sets of stripes drawn in different layers are joined smoothing the lines that connect them, applying the Equation 10.

The reconstruction of 3D stripes is performed by sampling pairs of points along the lines that delimit the stripe (red and blue in Fig. 78b) creating rectangles in order to render stripes as triangle meshes. To create 3D twists and rotation effects, our system estimates the depth of vertices z in rectangles by using the distance between lines and the stripe thickness (red and blue in Fig 78b), applying the following equation:

$$z = h * \sqrt{1 - \frac{c^2}{h^2}}, \tag{12}$$

where c is the distance from the center of the rectangle to its border, and $h = 10$, the default thickness of the stripe.

The value of z is then added to one side of the rectangles and subtracted on the other side, consequently rotating the stripe on points in which the distance between sides is less than the value of h. We then define the face orientation by calculating the normal direction for each triangle of the rectangle. This process automatically alternates its appearance applying color as a front and back face distinguish in their normal direction (Fig. 78c).

4.6 Results and Discussion

We created drawings representing artistic styles (Fig(s). 83, 85, 86, 88, 91) and zoological illustrations (Fig(s). 93, 94, 95, 96).

We specifically evaluated designs containing overlapping structures for our experiments. Figure 83a shows the Haken's Gordian Knot. This complex knot has numerous overlays and represents an unknot, *i.e.*, a closed-loop that is not knotted and can be unfolded [103]. Our framework allows the user to draw it by changing the layer of the open contours as shown in Fig. 83b, and the final render as shown in Fig. 83c.

Some interactive approaches use sketching operators with multiple annotations to create curvature fields, resulting in several adjustments and refinement of meshes. For instance, when creating a revolution surface as shown in Fig. 84. It is worth to mention, that by allowing the user to specifically drawn contours in different categories, the proposed system handles the 3D reconstruction with greater flexibility for each contour stroked by the user.

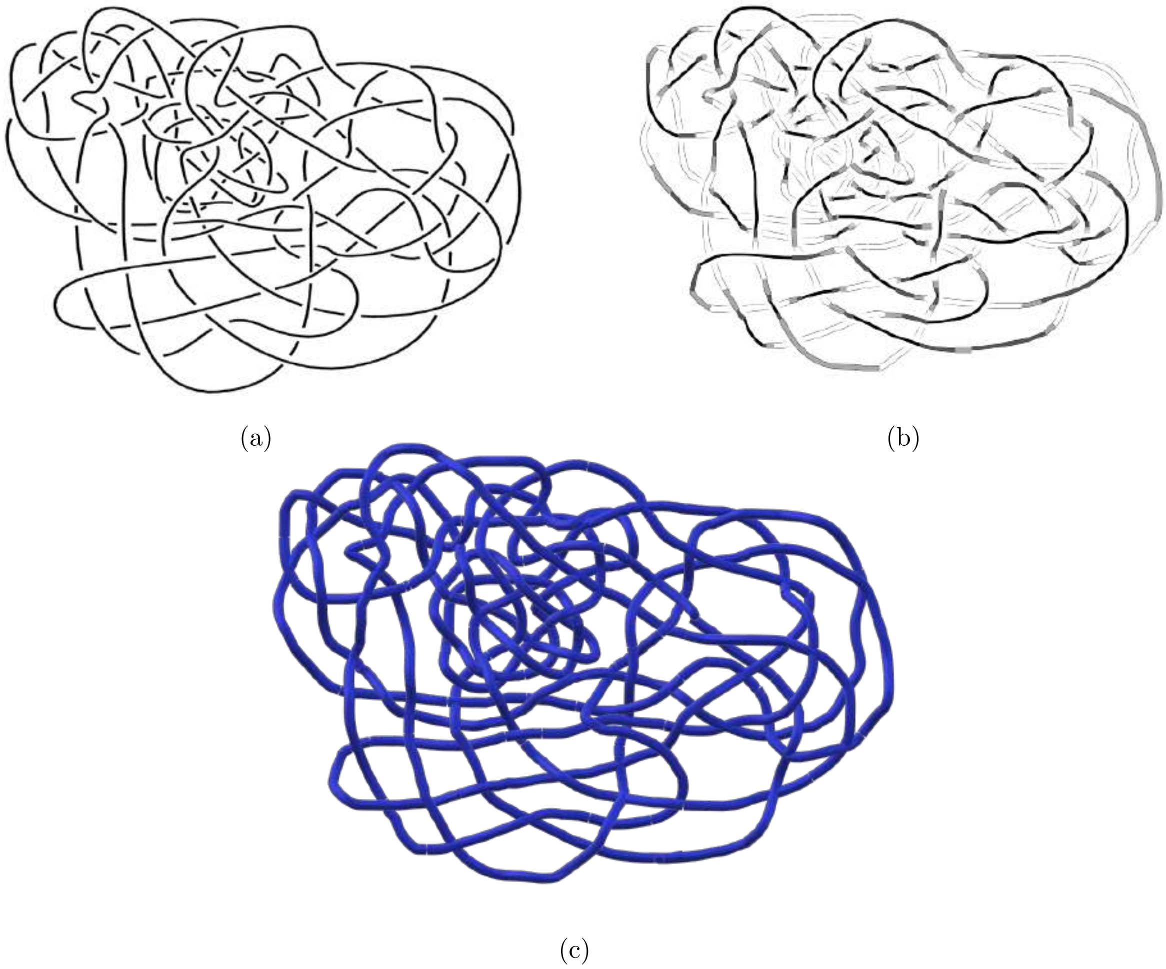


Figure 83: Sketching, visualization and reconstruction of an open contour. (a) Haken's Gordian Knot adapted from Fish and Lisitsa [36], (b) Our sketch traced over the original drawing based on Mick Burton's interpretation [18]. (c) The resulting 3D model is rendered by creating cylinders between three-dimensional points.

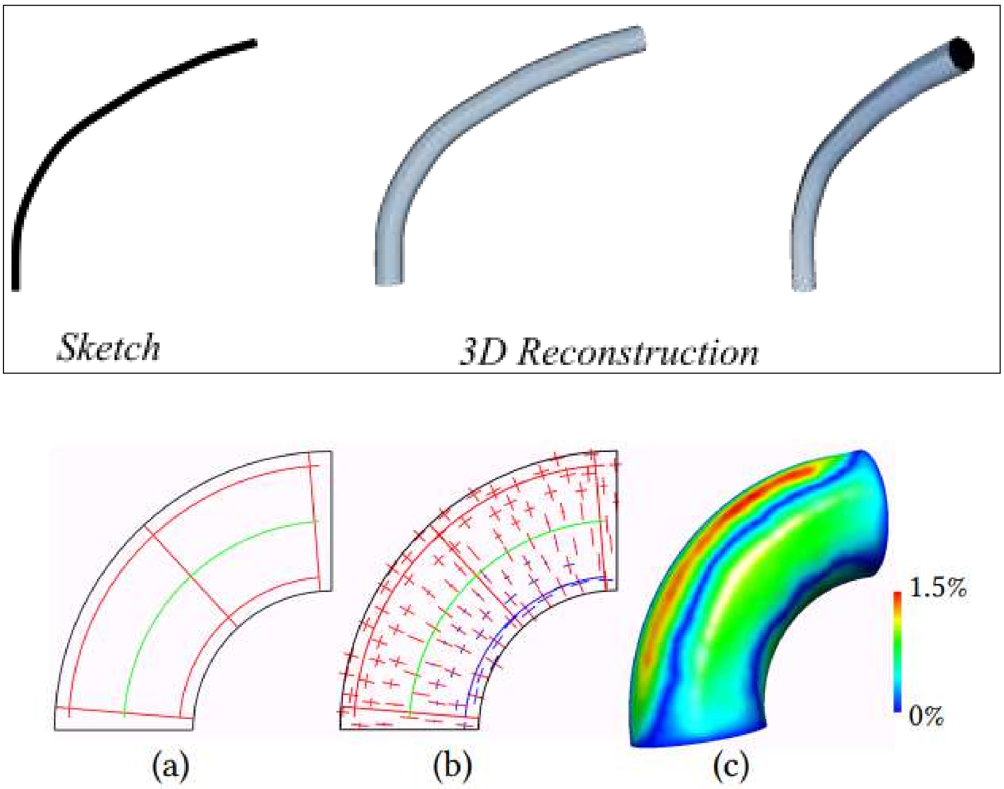


Figure 84: Reconstruction of an Open Contour by our framework (top/highlighted) and result from a similar drawing in comparison to the work by Li et al. [82] (bottom).

The heart in Fig. 85 was reconstructed combining two drawing approaches, stripes, and closed contours for 3D shape reconstruction. First, we sketch the heart silhouette, described by the gray line in Fig. 85b. We then sketch the stripe that defines the blood flow of the heart. We use the Rotational Blending Surface approach (i.e., Sec. 4.5.2) to define a parametric shape for the heart while for the stripe we interactively change its layer while drawing, so the stripes also overlap in a three-dimensional space as shown in Fig. 85c.

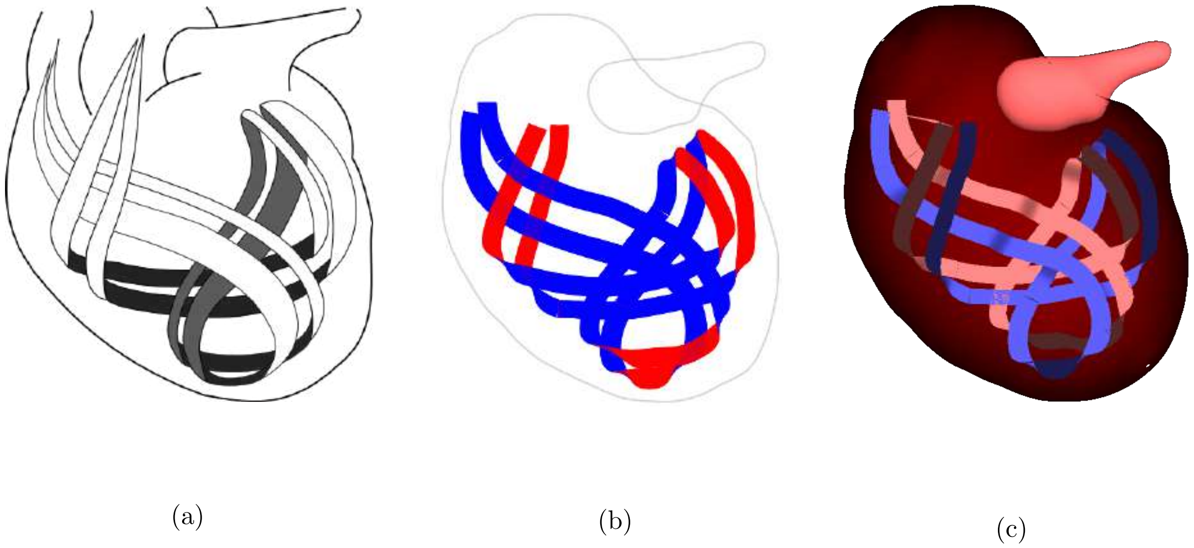


Figure 85: (a) Adapted drawing from heart artistically illustrated by Laura Maaske [85]. (b) Our sketch traced over the original illustration with closed contour and stripes painted in red and blue. (c) The resulting 3D model showing 3D stripes inside the heart's mesh.

Figure 86a was designed using three layers: the first used for the bamboo at the front, and the others used for the parts of the panda and bamboo in different layers. The effects shown are the lines painted in grayscale (Fig. 86b) and coloring for different layers with a hatching effect in Fig. 86c.

Another possible combination allows using different approaches to 3D reconstruction for a single-view input, instead of exhaustively adjust and alter meshes in multi-view approaches as [61, 82].

Although we used closed contours to draw all parts of the illustration, we used Rotational Blending Surface to reconstruct the bamboos and its leaves, and we used HRBF to reconstruct all parts of the panda body. The bamboo has a cylindrical body, which can be parametrically described, while the panda's body has flattened parts due to the flexibility to alter the shapes in HRBF.

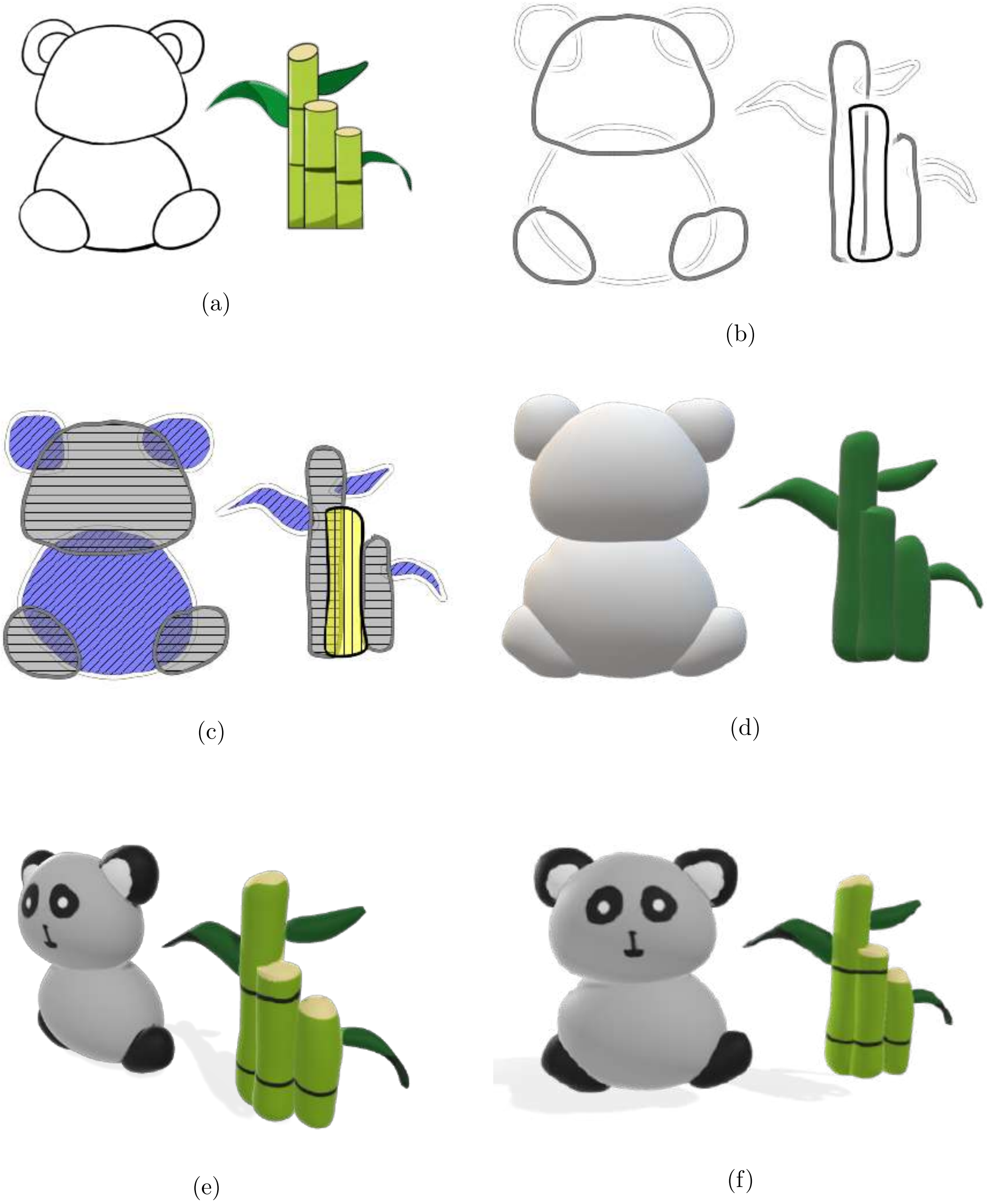
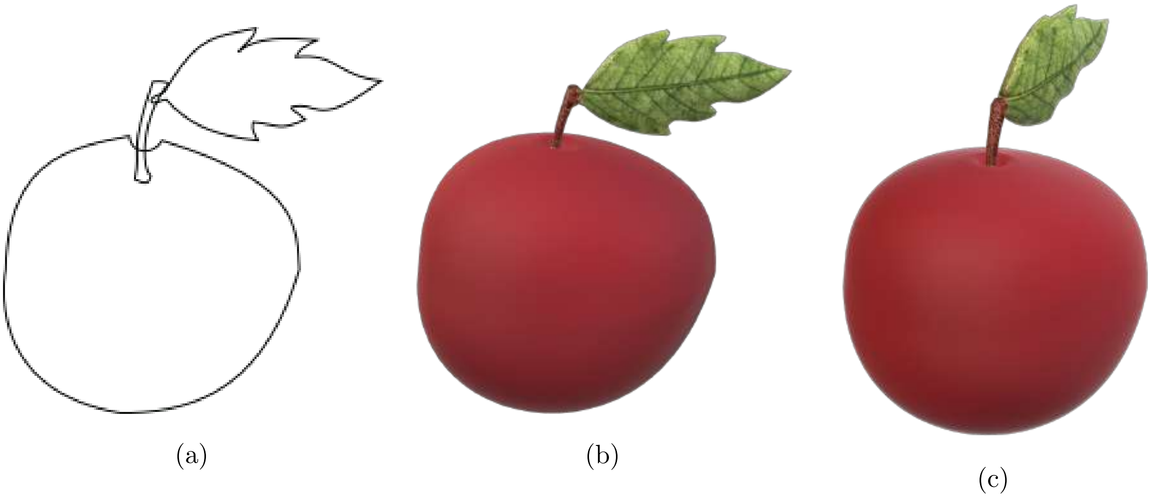


Figure 86: (a) Illustration drawing of a panda and bamboos. (b) Our sketch traced over the original illustration with the contour gray-scale and halo effects applied, (c) Color Layering and Hatching effect. (d)-(f) The resulting 3D model from different perspectives.

Another example is combining the two reconstruction approaches into one object, as shown in Fig. 87. The leaf was reconstructed using the HRBF method and then textured while the apple and the stalk were reconstructed using the Rotational Blending Surface and then painted.



(a) (b) (c)

Figure 87: Illustration of an apple combining the two reconstruction methods in the final 3D model.

Figure 88 exemplifies scalable vector graphics (SVG) input for our framework. Closed contours were selected through a cross selection operation, changing the layer in which each contour should belong before being selected.

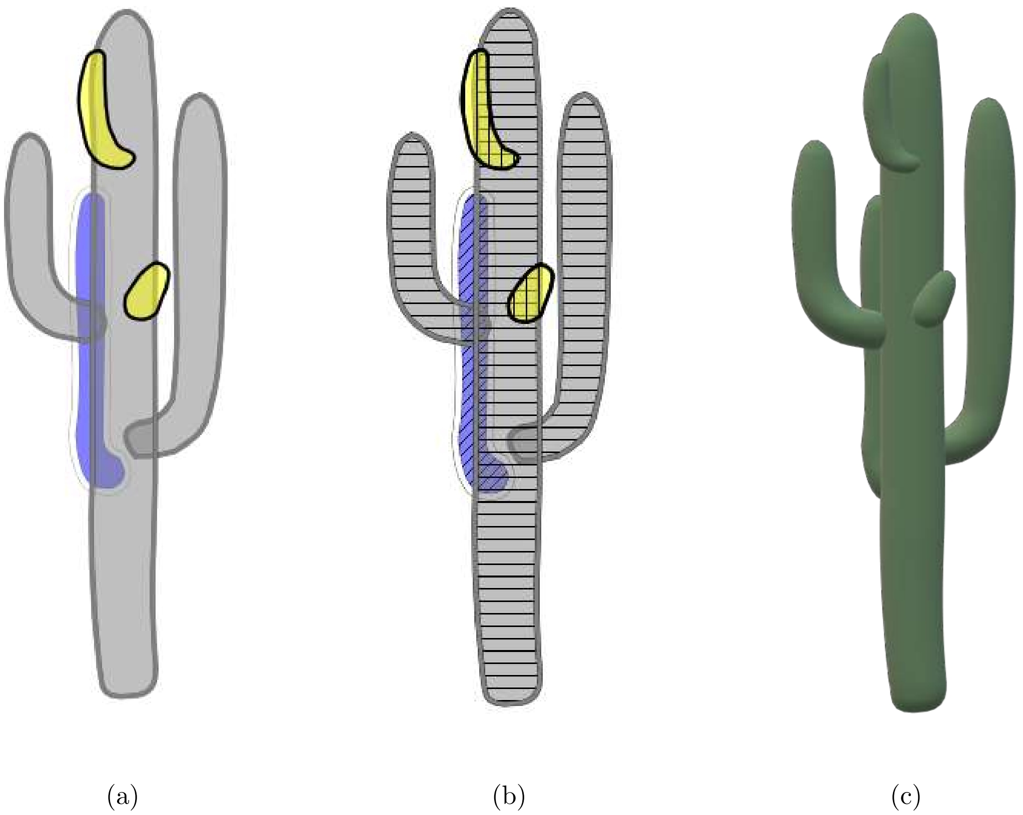


Figure 88: Cactus model with coloring effect (a), and hatching effect (b). The resulting 3D model reconstructed using Rotational Blending Surfaces (c).

In some cases, using several layering effects is not necessary. For instance, in Fig. 88a, using only the coloring by layers, it is possible to identify the bodies, whereas in Fig. 88b, the bottom part of the cactus, in blue, is less noticeable to the user because the contours above have the effect of hatching, creating a partial occlusion. Notice that, for the cactus drawing, previous works [31, 101] do not support line crossing on contours and ramified parts do not contain a corresponding symmetrical part drawn, being addressed to the same layer and depth of the main stalk as shown in Fig 89. In our work, the final model is reconstructed using the Rotational Blending Surface approach (i.e., Sec. 4.5.2), and the texture was applied to the final model.

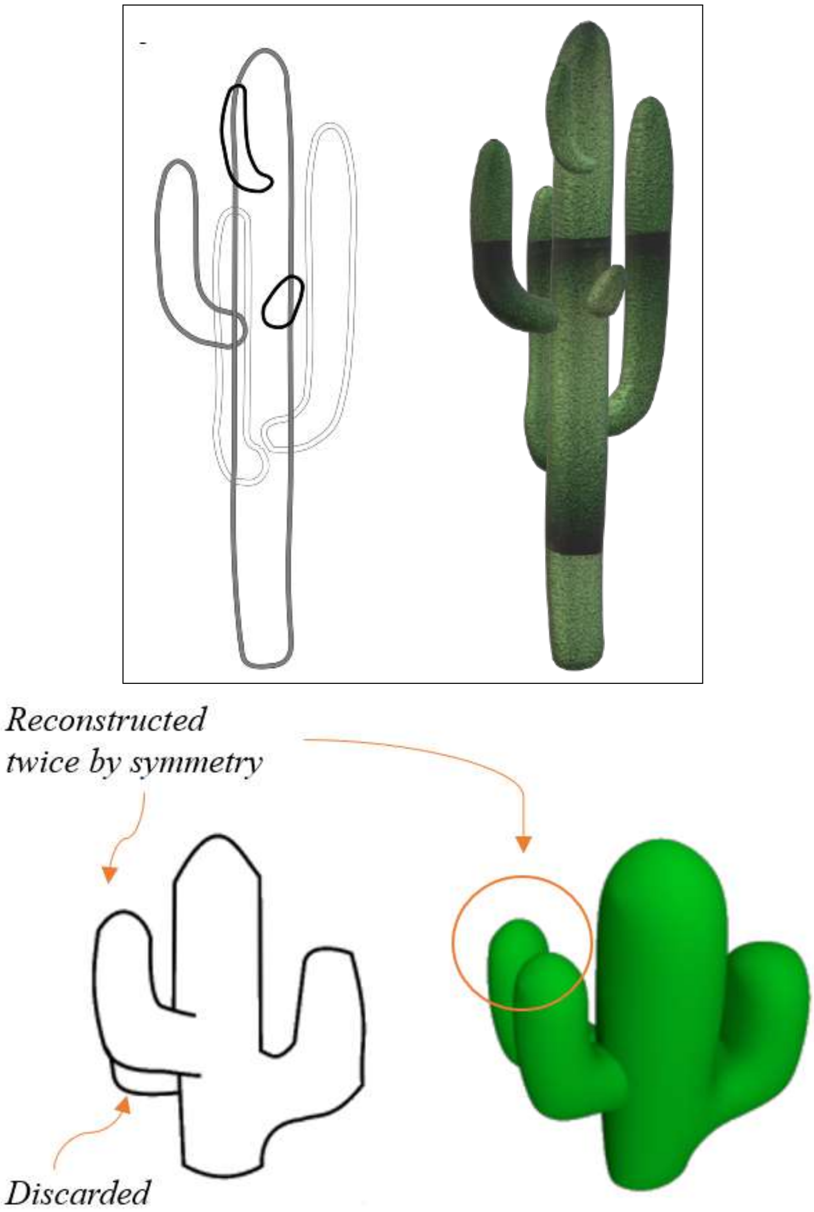


Figure 89: Our approach (top/highlighted). Example of cactus result reconstructed by Ramos et al. [101] (bottom). Different from approaches that infer or make simple assumptions, such as, structural symmetry, enabling the user to specifically draw all different parts of the object in multiple layers, avoid to discard occluded parts.

The support of layers plays a key role in our work, enabling the user to create occluded parts with different depths in single-view drawings. This is specially valid when comparing our results with the work of Li et al. [82], which limits the 3D reconstruction of the model to a single subject, as is notice in Fig. 90, where the final model of the bird lacks one leg. Another example is the comparison with the automatic segmentation proposed by Ramos et al. [101], which does not support multiple overlaid structures, limiting the reconstruction to one pair of the symmetrical overlaid limb as it is possible to notice on the wing of the bird (Fig. 90).

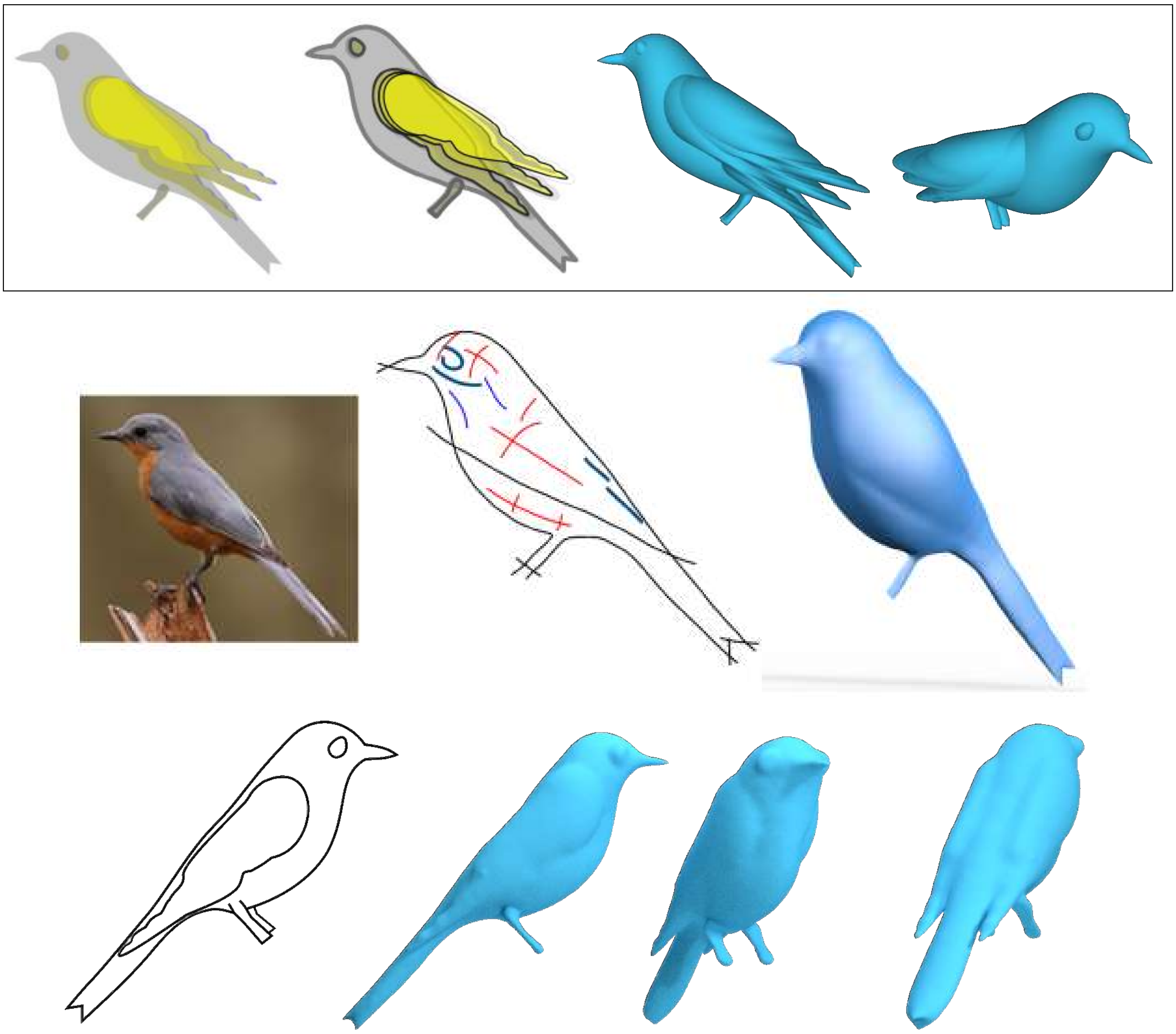
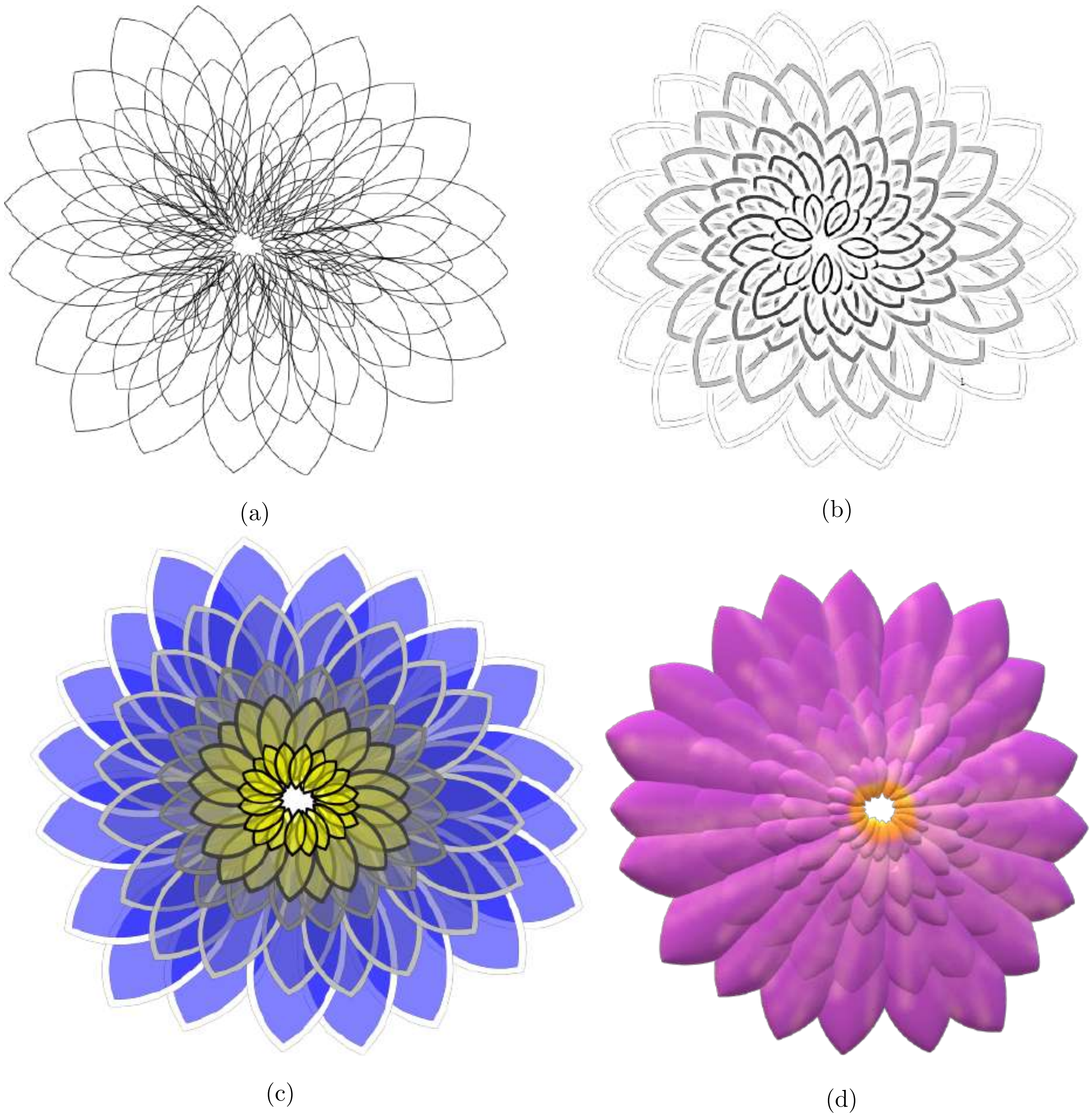


Figure 90: Results comparing a bird model reconstructed by our approach and previous works. Our approach (top/highlighted). Results from a similar drawing reconstructed by Li et al. [82] (middle) and Ramos et al. [101] (bottom).

Our framework supports numerous contours and layers. For instance, the flower in Fig. 91 contains 100 petals, expressed by closed contours divided into five layers. Automatic approaches infer occluded parts using drawing properties [31, 32, 101].



(a)

(b)

(c)

(d)

Figure 91: (a) Illustration of a flower drawn with overlapping contours describing its 100 petals. (b) Our sketch was done by selecting the contours in the original illustration with the Halo Effect, (c) Layer coloring effect. (d) The resulting 3D model rendered with different tones of pink.

In the flower drawing (Fig. 91), the depiction of layers is not an easy task for the human eye due to the number of petals drawn as shown in Fig. 91a. The depth effects implemented by our framework show that a visually confusing or ambiguous set of contours can be better displayed by using different depth effects as shown in Fig(s). 91b and 91c where the identification of layers is easier for the user by visualizing different colors according to contour's layer. The resulting model for the flower is shown in Fig. 91d. Although the layer order starts from the top, closer to the user, to the bottom, further to the user, a better 3D rendering was obtained by flipping the final mesh showing the layers in inverse order to enhance the appearance of a flower.

4.6.1 Application and Analysis for Biological Systematic Illustrations

Taxonomy is the scientific activity of recognizing biodiversity and describing its basic units, the biological species [105]. Zoologists and botanists base their descriptions in external and internal observable attributes. For instance, there are different taxonomic needs when one is dealing with mammals or birds compared to mollusks or arthropods; however, it is quite clear that the product of any taxonomic study is a species (or group) description that is actually a scientific hypothesis [1].

A taxonomic study often begins with fieldwork through collecting biological material, followed by sorting it in the laboratory, and identification of known species (when possible) using identification keys as stated by Santos et al. [107] and Amorim and Donoso [3]. The unrecognizable specimens may be described as new species. To achieve such goals, detailed analyses of morphology, both external and internal, are necessary. The evidence used to support the description of new species is mostly represented through photographs, scanning electron micrographs, and illustrations.

The overall quality of species description, and its further uses in phylogenetic, biogeographical or ecological analyses, profoundly depends on the photographed evidence presented in taxonomic studies. In a nutshell: the more significant the details, the better the description. Below, we discuss aspects of the current practice in entomology (the study of insects), explaining through examples of taxonomic studies and illustrations of species of Diptera.

Insects are widely known by their metameric segmentation. A general update review of insect external morphology is presented by Gullan and Cranston [52] and is the basis of the description below. A general insect consists of three *tagmas* (i.e., a grouping of multiple segments into a coherently functional morphological unit [74]): a six-segmented head, a three-segmented thorax and an 11-segmented abdomen, with varying degrees of fusion. The insect **segments** are heavily sclerotized - sclerotization gives rises to plates called sclerites. The dorsal plate of each segment is the tergum (tergite), the ventral plate is the sternum (sternite) and the side plates are the pleura.

Here, we are mainly concerned with the head and the Terminalia of Diptera, one of the largest orders of *Insecta* [14]. In general, along with the wing, the head capsule and the terminalia are the most important structures for diagnosing dipteran species. That is the motive for their use as the diagnosis of new species, and the main reason why we focus our framework on such features. The terminalia is the anal-genital part of the abdomen, consisting generally of segments 8 or 9 until the abdominal apex. In fact, we may say the terminalia relates to the last three segments of the abdomen (8, 9 and 10) plus a pair of appendages, the cerci, with variations among the insect orders.

Even a slight variation in terminalia may lead to sexual isolation between otherwise morphological similar species. In Diptera, terminalia are widely variable, demanding distinct processes to analyze and draw, mainly due to its internals requiring a clearing process [105, 106]. Another difficulty to represent genital parts in dipteran drawings is compounding both external and internal morphology in the same illustration (layers), keeping the three-dimensional perspective as well as precise measures.

Hence, a careful drawing process must be performed using stereo and optical microscopes attached to camera lucida (Fig. 92). Drawings using camera lucida are necessary for all of the non-sclerotized internal parts, but inking is also crucial for clarity and morphological accuracy. It is worth to mention that, under the microscope, external and internal details can be observed, photographed, and illustrated. One of the challenges concerning sketches of genital parts is maintaining the three-dimensionality. Any computational tool should take such difficulty for granted.

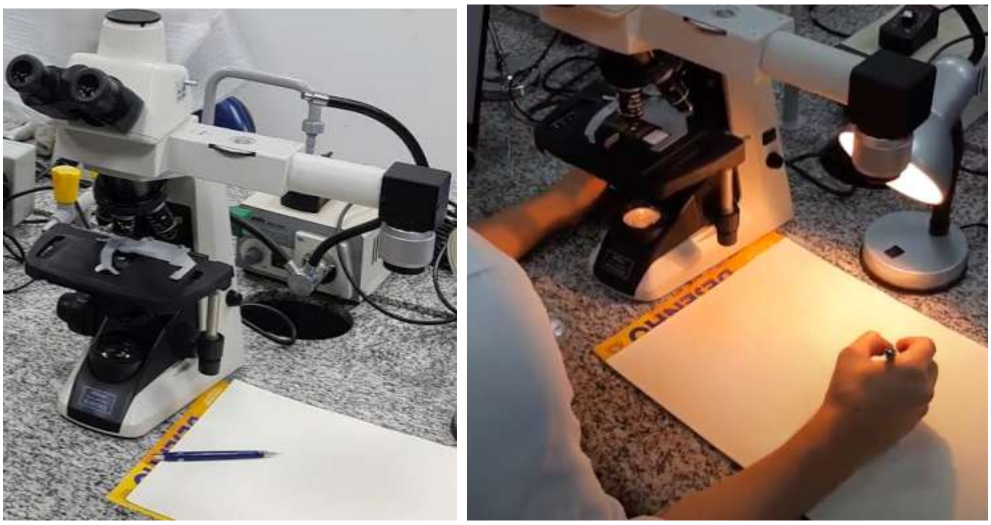
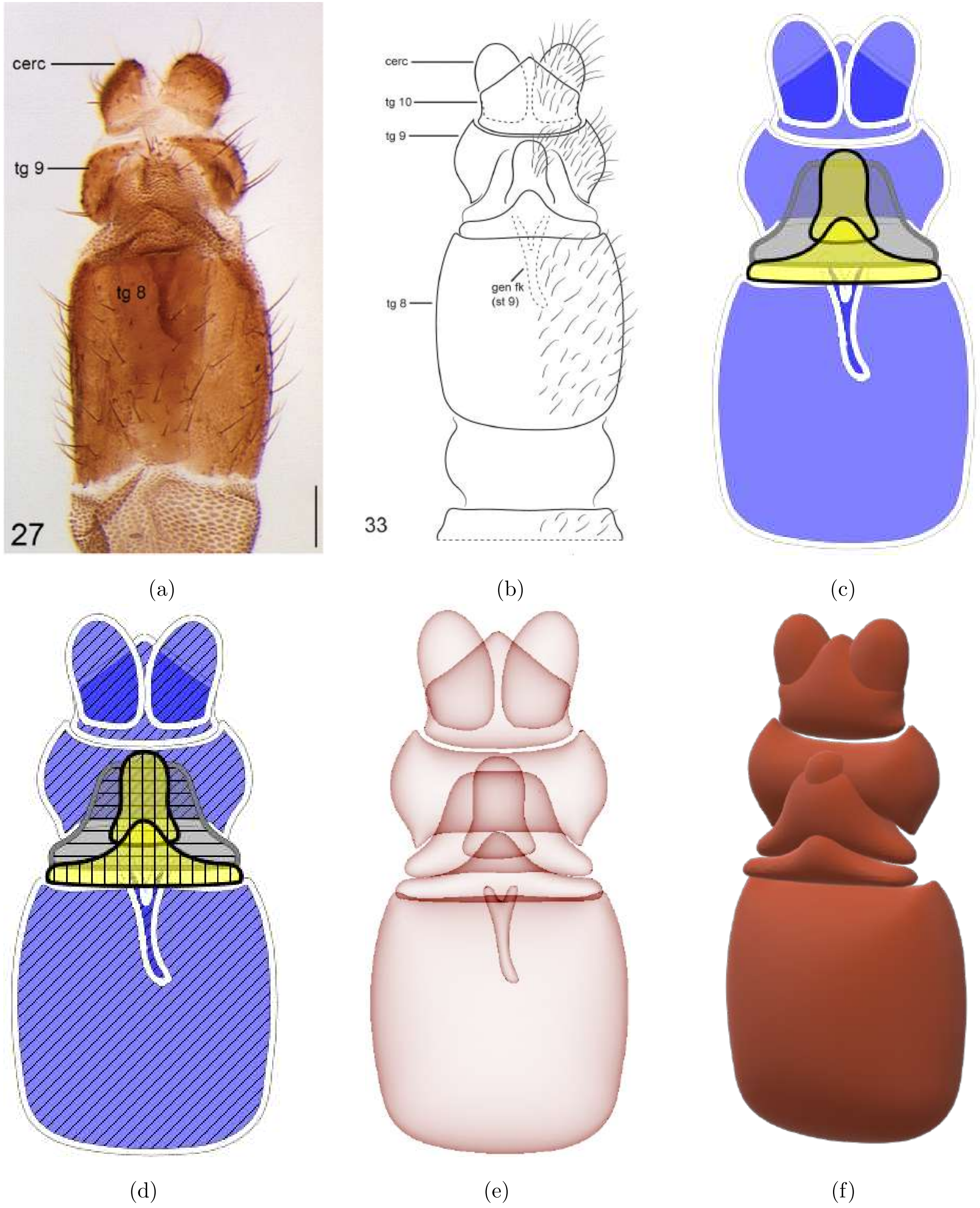


Figure 92: Camera lucida: the illustrator sees with the right eye the insect whereas the left eye see his hand and paper. This stereoview is then overlapping, allowing us to draw onto the insect structure. After that, traditionally, the sketches inked on paper are ”transferred” to an illustration software such as Adobe Illustrator.

For instance, the female terminalia exemplified in Fig. 93 was described containing three layers. Although the illustration describes occluded parts (Fig. 93a), it does not include depth information or layer addressing as it is possible to visually identify parts in different layers on Fig(s). 93b and 93c.

The insect terminalia, generally consisting of segments 8-10 of the abdomen, is composed of hardened external parts called tergites (tergum or dorsal plates) and sternites (sternum or ventral plates), as well as the internal reproductive system containing ovaries, accessory glands, spermathecae, and spermathecal ducts. Such a three-dimensional structure, hardly noticed in 2D pictures (Fig. 93a) or drawings (Fig. 93b) can be seen in Fig. 93f.



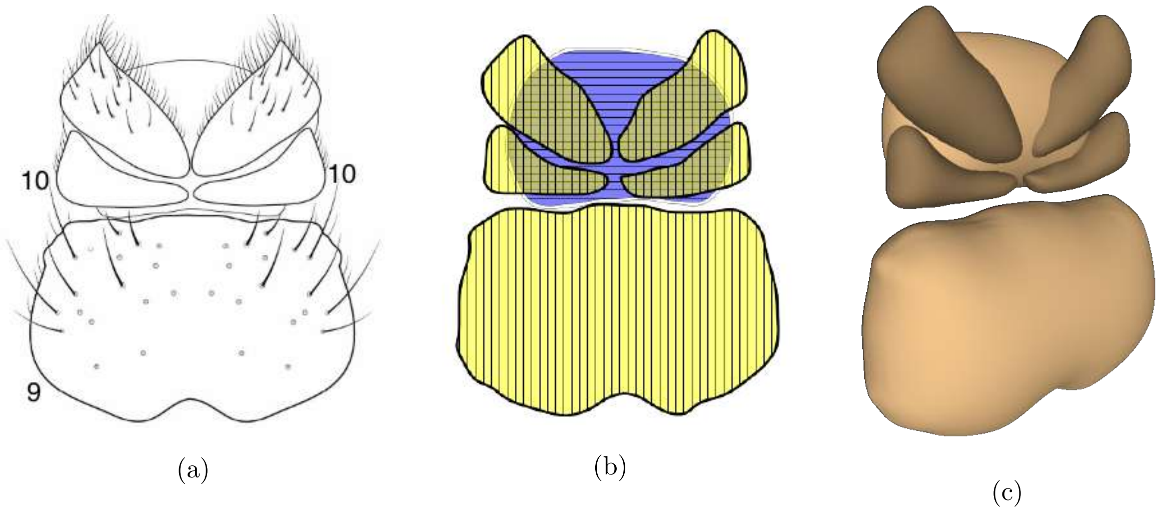
(a) (b) (c)

(d) (e) (f)

Figure 93: Photograph of the female terminalia of an Austroleptis (a), and its adapted drawing without bristles used as input (b) (both from Fachin et al. [34]). Our closed contour selection traced from the adapted illustration (c)-(d). The resulting 3D model (e)-(f).

The 3D model was reconstructed through the HRBF Surface approach, as the model contains non-parametric parts describing an organic shape with ramifications as shown by dashed lines in Fig. 93b and its respective reconstruction using points and normals shown in Fig. 93e. The 3D reconstruction shows that, although different parts can overlap (Fig. 93f), occluded shapes are visually possible to identify when transparency or a proper texture is applied as exemplified in Fig. 93e.

In general, the male insect terminalia consists of the final abdominal segments 8-10, with a dorsal epandrium and a ventral hypopygium. Figure 94 represents the epandrium, composed of the dorsal portion of segments 9 and 10, and the cercus at the apex. Here, it contains two layers. It is possible to notice that, the effects applied (coloring and hatching) can describe both layers as shown in Fig. 94b. As aforementioned, the hatching effect applied for both layers with different directions, horizontal and vertical, automatically creates a cross-hatching pattern where the layers overlap. The 3D model is shown in Fig. 94c and reproduces the hand-drawn sketch performed over the original scientific illustration.



(a)

(b)

(c)

Figure 94: (a) Terminalia, dorsal view, of a *Chrysopilus phaeopterus sp* from Santos et al. [106]. Our sketch traced over the original illustration (b). The 3D resulting model (c).

The ventral portion of the male holotype, called the hypopygium, of a *Chrysopilus balbii* is drawn and reconstructed in Fig. 95 using our framework. The hypopygium has two main structures: the gonocoxite, here composed of the fusion of segments 9 and 10, and a pair of gonostyles, as well as internal structures (ejaculatory apodemes, sperm sac, and aedeagus).

The transparency applied over the layer coloring effect plays a key role in occluded parts. See the layer painted in blue in Fig. 95, four contours belong to the same layer, and still, our prototype system and framework allows the user to visualize all contours by combining the transparency effect with the layer coloring. Even containing occluded parts reconstructed by our framework, enabling to interactively define depths for layers, it is possible to visually identify that all parts were reconstructed with different depths addressed to layers as shown in Fig. 95f.

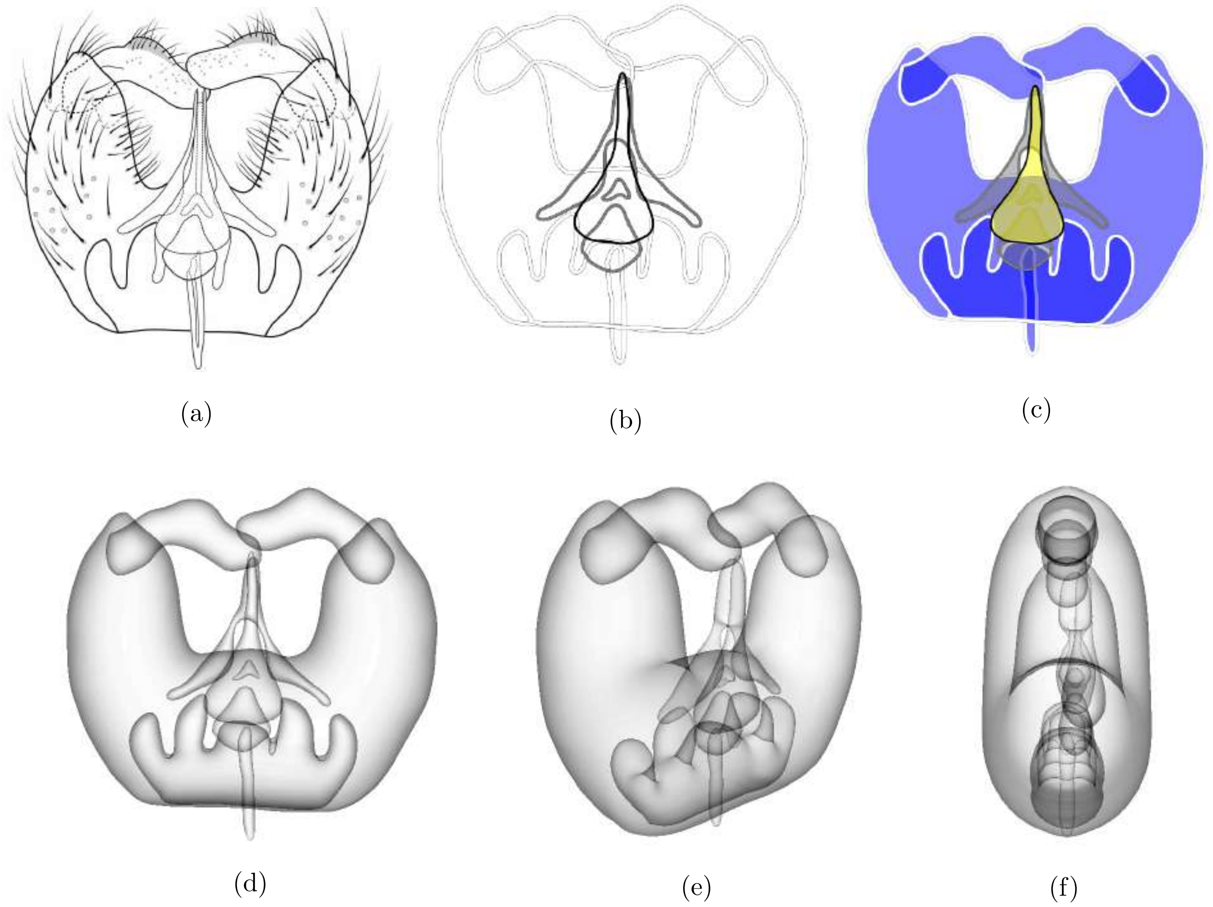
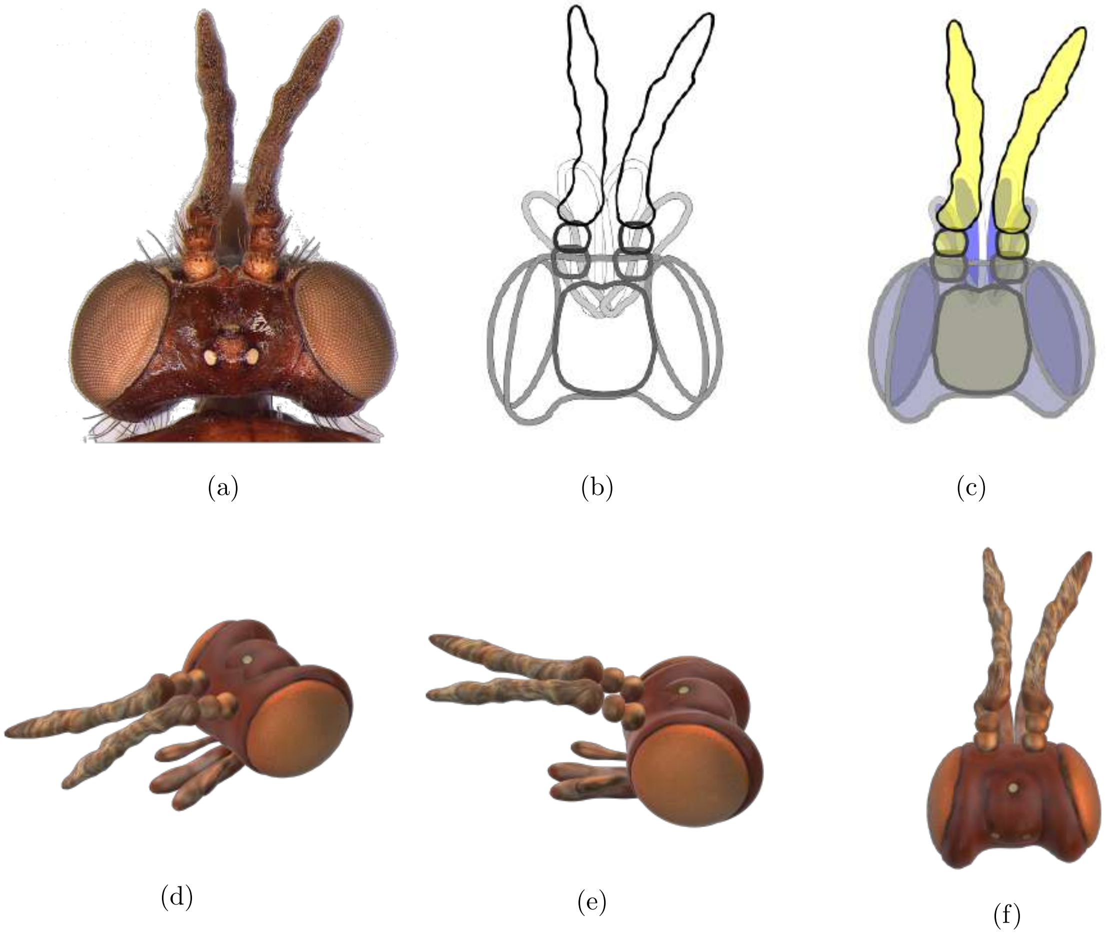


Figure 95: Male holotype of a *Chrysopilus balbii* (a) from Santos et al. [106]. Our sketch traced over the original illustration (b)-(c). The resulting 3D model (d)-(f).

Although apparently unsegmented, an insect head capsule consists of six fused segments, each with a pair of appendages, a great sort of sensory organs, including the compound eyes and antennae, and the external mouthparts, such as mandibles. The head of a specimem from the genus *Austroleptis* was reconstructed through the Rotational Blending Surface, as its parts can be translated to a parametric reconstruction refining the final mesh by rotating the medial axis that is used as the basis for the final model.


(a) (b) (c)

(d) (e) (f)

Figure 96: Picture of the head of an Austroleptis (a) (source [34]). Our closed contour selection traced from the picture (b) and with layer coloring (c). The resulting 3D model (d)-(f).

Defining depths for the layers enables the user to reproduce partial occlusion of different parts of the model as can be noticed in Fig. 96e, the head is composed of two parts, one containing other. It is important to notice that, in this case, a proper interpretation of photographed evidence is fundamental. For instance, the head captured in Fig. 96a presents a distortion due to the angle of the camera, thus the drawing was stretched in Fig. 96b.

4.7 Chapter Remarks

As far as we know, this is the first SBIM system tailored for single-view overlaying sketches that supports different categories of contours to describe 3D shapes: open contours for knots, closed contours for 3D surfaces and a novel method for modeling 3D stripes as bands with twists.

We provided steps towards sketch-based modeling based on domain-oriented features in single-view, including a set of visual effects for 2D drawings based on traditional illustration techniques and different 3D reconstruction methods for each characteristic contour, allowing the user to create complex and layered objects.

Our system supports layering and real-time feedback through depth effects for users and scientific illustrators of published species of the mega-diverse order Diptera. We addressed the problem of dealing with shape inferences for occluded parts presenting a sketch-based interface where the user explicitly defines hidden contours for models enabling the detailing of different contours as open contours, closed contours, and, stripes as well as methods to their respective 3D reconstruction.

Our 3D reconstruction methods allow the user to create a wide variety of 3D surfaces. However, the reconstruction is limited to the predefined methods, mesh edition is an important task to be incorporated, being useful to easily split, scale, stretch, or rotate different parts of the objects for the final reconstruction. It would enhance the reconstruction of single-view sketches, easing the manipulation of concave or convex surfaces, such as wings and shells, or, the reconstruction of specific parts better described in other 2D views.

Other sketching interactions are fundamental to add details to the overall 3D shape of the subject, *i.e.*, our system does not support cross-sectional blending surfaces and orthogonal deformation as proposed by [4, 24]. Many parts of species and plants, e.g., shells, wings and leaves, demands these types of surfaces and interactions for their 3D reconstruction.

Another limitation is the application of textures to meshes, the inclusion of textures based on images and photographs, is necessary to create a more realistic appearance of the 3D models. Figure 96a shows the importance of further work to investigate the proper application of textures to 3D models, as in the compound eyes, composed of many facets, and the antennae and its setae [80].

Although the results show promising uses for our framework, relevant future works were identified.

We are also interested in the 3D illustrations presented in specific areas of Mathematics, for instance, Knots Theory, from the fields of Geometry and Topology [27, 103]. The drawings of Knots frequently rely on hand-draw sketches which low artistic quality due to the lack of suitable tools. Thus, to better handle knots as well as open contours, we aim to approach specifically this class of drawings.

An ongoing investigation is to support 3D labeling of parts on reconstructed species. This process can be an initial step to build an online repository of 3D models from species encountered and reconstructed by many biologists around the world. Small features related to specific characteristics of the specimens are discarded, for instance, the need to simulate bristles as shown in Fig(s). 95 and 96.

In biological systematics, it is worth mentioning the difficulties of producing plausible images from fossils. Extinct insects are usually preserved as flat films on the surface of rocks (impression and compression) or as amber inclusions when the living specimen is trapped in tree resin. For instance, for Diptera, most fossils are incomplete wing compression or amber inclusion. Hence, we consider as future work to investigate, develop, and integrate into our framework novel 3D reconstruction approaches capable of modeling insects from fossils in different states of preservation.

A possible development of the framework herein presented is a full segment-oriented system for sketch-based modeling according to the metameric segmentation *bauplan* (as presented in Sec. 4.6), which would be useful for taxonomists and zoologists of any of the 30 insects orders known to date. Yet, an ongoing investigation, is conduct a user study and evaluation with biological scientists of different levels of expertise focusing on scientific illustrations and models.

Finally, another demand for future work is the development of a web portal to share species descriptions as well as their drawings and 3D models.

5 CONCLUSION AND FUTURE DIRECTIONS

Sketches and drawings allowed humanity to evolve through engineering, art, and design. In the last years, techniques and methods for sketch-based reconstruction have become a well-explored area allowing to layman and professionals to create their own 3D models.

In this thesis, we provided steps towards the 3D reconstruction of single-view sketches presenting two frameworks: *i*) an automatic skeleton-free approach for 3D reconstruction, and *ii*) a Sketch-Based system tailored for single-view overlaying sketches that support different categories of contours.

5.1 Conclusions

In Chapter 1 we listed the sketch-based modeling gaps and limitations related to Graphical User Interfaces, Visual Memory and Visual Rules, Suggestive Lines, and Accuracy. To address these limitations, this thesis focused in:

1. Investigate automatic 3D reconstruction methods from single-view sketches, proposing segmentation strategies capable of using suggestive contours and indicative lines increasing the number of shapes, bodies, and details that can be reconstructed.

2. Propose a framework for Sketch-Based Systems & Modeling (SBIM) that includes 2D representation of overlaying objects and 3D reconstruction for different contour categories in a single-view perspective, providing support to 3D layering and real-time feedback for users as illustrators and scientists.

3. As an application, explore the proposed framework to model entomological features often present in species descriptions of the order Diptera, one of the mega-diverse group of insects popularly known as flies and mosquitoes.

Throughout this work, we presented an automatic method for 3D reconstruction from sketches (Chapter 3). So far, this is the first study that explores the problem of reconstruction from a single-view sketch with a skeleton-free technique based on scattered Hermitian data. Eliminating the use of skeletons to reconstruct the parts, we estimate the depth and 3D models using a normal propagation method, adding to the reconstruction parts not identified in previous works trough a segmentation and classification improvement. We investigated automatic 3D reconstruction methods from single-view sketches proposing advances on segmentation strategies to include suggestive contours and indicative lines discarded in previous approaches.

Furthermore, we presented an SBIM system tailored for single-view overlaying sketches that include a set of visual enhancements to improve the depiction of complex and layered objects (Chapter 4). Our system is able to create 3D models based on different categories: open contours for knots, closed contours for 3D surfaces, and modeling 3D stripes as bands with twists. This framework provides support to 3D layering and real-time feedback for users as illustrators and scientists. Thus, as an application, the proposed approach was used to model entomological features of a mega-diverse group of insects, known as Diptera, that represents the group of flies and mosquitoes.

There has been a growing effort on bringing concepts from visual perception to sketch-based modeling. For instance, the subset of representations that a subject can represent [97] and the development of strategies capable of efficiently classify and use simple information as contours to define parts [32] and generate 3D information using regularities [109, 124]. I have also contributed to these efforts by emphasizing the importance of the preservation of the details present on contours and the use of previously discarded strokes. Besides, introducing to SBIM visual enhancements inspired by illustrators regarding the importance of applying the efforts to a scientific area.

However, many more techniques that implement effective visual perceptions, drawing patterns, suggestive contours, and reconstruction methods are still either incipient or not available in the SBIM community. I argue that exploring these ideas born in traditional illustration and other areas of expertise can provide a pathway of research and creative algorithms in the years to come. I hope that this thesis can provide solid examples that can be used as inspiration for researchers that intend to follow this path in the future.

5.2 Future Works

There are many avenues for future research. Mesh edition would improve the reconstructed 3D model from single-view sketches, easing the manipulation of concave or convex surfaces, such as wings and shells, or the edition of mesh to create models better described with 2 or more views. A sketch-based mesh edition can also be developed in coherence with previously proposed approaches as explored by Nealen et al. [93]. Besides, a direction is the implementation and improvement of operators to create cross-sectional blending surfaces and orthogonal deformation as proposed by [4, 24].

Another direction is to explore other reconstruction algorithms that generally are applied to skeleton-based reconstruction such as the Laplacian interpolation [21], mesh refinement as in the case of solving Poisson problems [119] and mesh completion as the problem explored with bi-harmonic fields [8]. In spite of 3D reconstruction, a future investigation is to explore the method of normal propagation and HRBF Implicits interpolation to improve fidelity, for instance, modify the methodology of sampling points to improve the features preserved in the final models.

Many reconstruction methods were proposed in last years, however, our research showed that there is a lack of works including benchmarks of different approaches for reconstruction in Sketch-based systems. A future investigation is to perform and evaluate different reconstruction approaches proposed in recent years.

Real-world sketches, for instance, engineering, architectural and zoological drawings raise numerous challenges due to inaccuracies, use of overdrawn strokes, and construction lines. Although construction contours present significant 3D information, multiple 2D overdrawn and accidental strokes limit the creation of 3D connectivity through an algorithm. A further step is to develop frameworks that can address the connectivity problem using designer observations and 3D intersection detection as recently proposed by Gryaditskaya et al. [50].

Although many approaches have been proposed, drawings of knots frequently rely on hand-drawn sketches which low artistic quality due to the lack of suitable tools [103]. Thus, a direction is to approach specifically this class of drawings, creating interactions and 3D reconstruction methods to create complex nodes based on works that explore the mathematical description and detail of knots [27].

A further investigation is the inclusion of textures based on images and photographs, commonly used in scientific illustrations to improve the final 3D model's texture, reconstruction, and appearance [20, 108, 112]. Yet, recent works have proposed generative models that can segment and parameterize texture representation of the 3D object estimating a two-dimensional texture map of the model surface [96, 111].

Still, aiming to develop machine learning 3D model retrieval and reconstruction from datasets, another demand for future work is the creation of a web portal to share species descriptions as well as their drawings and 3D models.

REFERENCES

[1] Ingi Agnarsson and Matjaž Kuntner. Taxonomy in a Changing World: Seeking Solutions for a Science in Crisis. *Systematic Biology*, 56(3):531–539, 06 2007. ISSN 1063-5157. doi: 10.1080/10635150701424546.

[2] Shaaron Ainsworth, Vaughan Prain, and Russell Tytler. Drawing to learn in science. *Science*, 333(6046):1096–1097, 2011.

[3] Dalton S Amorim and David A Donoso. Timeless standards for species delimitation. 2016.

[4] Fabricio Anastacio, Mario Costa Sousa, Faramarz Samavati, and Joaquim A Jorge. Modeling plant structures using concept sketches. In *Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, pages 105–113, 2006.

[5] Alexis Andre and Suguru Saito. Single-view sketch based modeling. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, pages 133–140. ACM, 2011.

[6] Bill Andrews. Bill andrews 2018 lifetime achievement award - association of medical illustrators. https://ami.org/press/press-releases/2018/392-bill-andrews-2018-lifetime-achievement-award, 2018. (Accessed on 10/04/2020).

[7] Arthur Appel, F. James Rohlf, and Arthur J. Stein. The haloed line effect for hidden line elimination. In *Proceedings of the 6th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '79)*, page 151–157, 1979.

[8] Oscar Argudo, Pere Brunet, Antoni Chica, and Àlvar Vinacua. Biharmonic fields and mesh completion. *Graphical Models*, 82:137–148, 2015.

[9] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. 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*, pages 151–160. ACM, 2008.

[10] Richard H Bartels and Faramarz F Samavati. Reversing subdivision rules: Local linear conditions and observations on inner products. *Journal of Computational and Applied Mathematics*, 119(1-2):29–67, 2000.

[11] Daniel Berio, Paul Asente, Jose Echevarria, and F Fol Leymarie. Sketching and layering graffiti primitives. In *Proceedings of the 8th ACM/Eurographics Expressive Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*, pages 51–59. Eurographics Association, 2019.

[12] Adrien Bernhardt, Adeline Pihuit, Marie-Paule Cani, and Loïc Barthe. Matisse: Painting 2d regions for modeling free-form shapes. In *SBM'08-Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pages 57–64. Eurographics Association, 2008.

[13] Mikhail Bessmeltsev, Will Chang, Nicholas Vining, Alla Sheffer, and Karan Singh. Modeling character canvases from cartoon drawings. *ACM Transactions on Graphics (TOG)*, 34(5):162, 2015.

[14] Art Borkent and Brian V Brown. How to inventory tropical flies (diptera)-one of the megadiverse orders of insects. *Zootaxa*, 3949(3):301–322, 2015.

[15] Stanley Bowestead and Thomas M Eccles. *Drawing Techniques for Publication.* Flexipress Printing Limited, 2012.

[16] E Vital Brazil, Ives Macedo, M Costa Sousa, Luiz Henrique de Figueiredo, and Luiz Velho. Sketching variational hermite-rbf implicits. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, pages 1–8, 2010.

[17] Philip Buchanan, Ramakrishnan Mukundan, and Michael Doggett. Automatic single-view character model reconstruction. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, pages 5–14. ACM, 2013.

[18] Mick Burton. Continuous line drawings with overlapping contours, 2015. URL https://mickburton.co.uk/tag/gordian-knot/.

[19] Fatih Calakli and Gabriel Taubin. Ssd-c: Smooth signed distance colored surface reconstruction. In *Expanding the Frontiers of Visual Analytics and Visualization*, pages 323–338. Springer, 2012.

[20] Marco Callieri, Paolo Cignoni, Massimiliano Corsini, and Roberto Scopigno. Masked photo blending: Mapping dense photographic data set on high-resolution sampled 3d models. *Computers & Graphics*, 32(4):464–473, 2008.

[21] Junjie Cao, Andrea Tagliasacchi, Matt Olson, Hao Zhang, and Zhinxun Su. Point cloud skeletons via laplacian based contraction. In *2010 Shape Modeling International Conference*, pages 187–197. IEEE, 2010.

[22] Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. Deep generation of face images from sketches. *arXiv preprint arXiv:2006.01047*, 2020.

[23] Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9416–9425, 2018.

[24] Joseph Jacob Cherlin, Faramarz Samavati, Mario Costa Sousa, and Joaquim A Jorge. Sketch-based modeling with few strokes. In *Proceedings of the 21st spring conference on Computer graphics*, pages 137–145. ACM, 2005.

[25] Matthew T Cook and Arvin Agah. A survey of sketch-based 3-d modeling techniques. *Interacting with computers*, 21(3):201–211, 2009.

[26] Frederic Cordier, Hyewon Seo, Jinho Park, and Jun Yong Noh. Sketching of mirror-symmetric shapes. *IEEE Transactions on Visualization and Computer Graphics*, 17 (11):1650–1662, 2011.

[27] Peter R Cromwell et al. *Knots and links*. Cambridge university press, 2004.

[28] Chris De Paoli and Karan Singh. Secondskin: sketch-based construction of layered 3d models. *ACM Transactions on Graphics (TOG)*, 34(4):126, 2015.

[29] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. Suggestive contours for conveying shape. In *ACM SIGGRAPH 2003 Papers*, pages 848–855. ACM SIGGRAPH, 2003.

[30] Chao Ding and Ligang Liu. A survey of sketch based modeling systems. *Frontiers of Computer Science*, 10(6):985–999, 2016.

[31] Even Entem, Loic Barthe, Marie Paule Cani, Frederic Cordier, and Michiel Van De Panne. Modeling 3D animals from a side-view sketch. *Computers and Graphics (Pergamon)*, 46:221–230, 2015. ISSN 00978493. doi: 10.1016/j.cag.2014.09.037. URL http://dx.doi.org/10.1016/j.cag.2014.09.037.

[32] Even Entem, Amal Dev Parakkat, Loïc Barthe, Ramanathan Muthuganapathy, and Marie-Paule Cani. Automatic structuring of organic shapes from a single drawing. *Computers & Graphics*, 81:125–139, 2019.

[33] Maarten H. Everts, Henk Bekker, Jos B.T.M. Roerdink, and Tobias Isenberg. Depth-dependentf halos: Illustrative rendering of dense line data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1299–1306, 2009. ISSN 10772626. doi: 10.1109/TVCG.2009.138.

[34] Diego Aguilar Fachin, Charles Morphy D Santos, DS Amorim, et al. First two species of austroleptis hardy (diptera: Brachycera: Austroleptidae) from brazil. *Zootaxa*, 4369(4):557–574, 2018.

[35] Ione Fine, Alex R Wade, Alyssa A Brewer, Michael G May, Daniel F Goodman, Geoffrey M Boynton, Brian A Wandell, and Donald IA MacLeod. Long-term deprivation affects visual perception and cortex. *Nature neuroscience*, 6(9):915–916, 2003.

[36] Andrew Fish and Alexei Lisitsa. Detecting unknots via equational reasoning, i: Exploration. In *Intelligent Computer Mathematics*, pages 76–91. Springer, 2014.

[37] Freddy Garnier. Car design pro — car design lessons - tip | 01 - by dino visit:... https://cardesignpro.tumblr.com/post/119777337068/car-design-lessons-tip-01-by-dino-visit. (Accessed on 10/04/2020).

[38] Golnaz Ghiasi and Charless C Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *European Conference on Computer Vision*, pages 519–534. Springer, 2016.

[39] Arnab Ghosh, Richard Zhang, Puneet K Dokania, Oliver Wang, Alexei A Efros, Philip HS Torr, and Eli Shechtman. Interactive sketch & fill: Multiclass sketch-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 1171–1180, 2019.

[40] Yotam Gingold, Takeo Igarashi, and Denis Zorin. Structured annotations for 2d-to-3d modeling. *ACM Transactions on Graphics (TOG)*, 28(5):148, 2009.

[41] J. P. Gois, B. A. D. Marques, and H. C. Batagelo. Interactive shading of 2.5 D models. *Proceedings of the 41st Graphics Interface Conference*, pages 89–96, 2015.

[42] Joao Paulo Gois, Anderson Nakano, Luis Gustavo Nonato, and Gustavo C Buscaglia. Front tracking with moving-least-squares surfaces. *Journal of Computational Physics*, 227(22):9643–9669, 2008.

[43] João Paulo Gois, Diogo Fernando Trevisan, Harlen Costa Batagelo, and Ives Macêdo. Generalized hermitian radial basis functions implicits from polygonal mesh constraints. *The Visual Computer*, 29(6):651–661, Jun 2013. ISSN 1432-2315. doi: 10.1007/s00371-013-0802-8. URL https://doi.org/10.1007/s00371-013-0802-8.

[44] Joao Paulo Gois, Diogo Fernando Trevisan, Harlen Costa Batagelo, and Ives Macêdo. Generalized hermitian radial basis functions implicits from polygonal mesh constraints. *Vis. Comput.*, 29(6-8):651–661, June 2013. ISSN 0178-2789. doi: 10.1007/s00371-013-0802-8. URL http://dx.doi.org/10.1007/s00371-013-0802-8.

[45] Joao Paulo Gois, Bruno A. D. Marques, and Harlen C. Batagelo. Interactive shading of 2.5d models. In *Proceedings of the 41st Graphics Interface Conference*, GI '15, pages 89–96, Toronto, Ont., Canada, Canada, 2015. Canadian Information Processing Society. ISBN 978-0-9947868-0-7. URL http://dl.acm.org/citation.cfm?id=2788890.2788907.

[46] Joao Paulo Gois, Bruno AD Marques, and Harlen C Batagelo. Interactive shading of 2.5 d models. In *Proceedings of the 41st Graphics Interface Conference*, pages 89–96. Canadian Information Processing Society, 2015.

[47] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A non-photorealistic lighting model for automatic technical illustration. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*, page 447–452, 1998.

[48] Bruce Gooch and Amy Gooch. *Non-photorealistic rendering*. CRC Press, 2001.

[49] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[50] Yulia Gryaditskaya, Felix Hähnlein, Chenxi Liu, Alla Sheffer, and Adrien Bousseau. Lifting freehand concept sketches into 3d. *ACM Transactions on Graphics*, 2020.

[51] Gaël Guennebaud and Markus Gross. Algebraic point set surfaces. In *ACM SIGGRAPH 2007 papers*, pages 23–es. 2007.

[52] Penny J Gullan and Peter S Cranston. *The insects: an outline of entomology*. John Wiley & Sons, 2014.

[53] Arthur L Guptill. *Rendering in Pen and Ink: The Classic Book On Pen and Ink Techniques for Artists, Illustrators, Architects, and Designers*. Watson-Guptill, 2014.

[54] Lizzie Harper. Botanical illustration - tips on painting composite flowers - lizzie harper. https://lizzieharper.co.uk/2013/04/tips-on-painting-composite-flowers/, 04 2013. (Accessed on 10/04/2020).

[55] Valentina Hertz. Hospitality/ retail_the wing: Uae, abu dhabi mixed use on behance. https://www.behance.net/gallery/2002335/Hospitality-Retail_The-Wing-UAE-Abu-Dhabi-Mixed-Use, 08 2011. (Accessed on 10/04/2020).

[56] Elaine RS Hodges. *The guild handbook of scientific illustration*. John Wiley & Sons, 2003.

[57] Donald D Hoffman. *Visual intelligence: How we create what we see*. WW Norton & Company, 2000.

[58] David Hull. Science as a process: An evolutionary account of the social and conceptual development of science, 1988.

[59] Emmanuel Iarussi, David Bommes, and Adrien Bousseau. Bendfields: Regularized curvature fields from rough concept sketches. *ACM Transactions on Graphics (TOG)*, 34(3):24, 2015.

[60] Takeo Igarashi and Jun Mitani. Apparent layer operations for the manipulation of deformable objects. *ACM Transactions on Graphics (TOG)*, 29(4):110, 2010.

[61] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3d freeform design. In *Acm siggraph 2007 courses*, page 21. ACM, 2007.

[62] Takashi Ijiri, Shigeru Owada, and Takeo Igarashi. Seamless integration of initial sketching and subsequent detail editing in flower modeling. In *Computer Graphics Forum*, volume 25, pages 617–624. Wiley Online Library, 2006.

[63] Pradeep Kumar Jayaraman, Chi-Wing Fu, Jianmin Zheng, Xueting Liu, and Tien-Tsin Wong. Globally consistent wrinkle-aware shading of line drawings. *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[64] Joaquim Jorge and Faramarz Samavati. *Sketch-based interfaces and modeling.* Springer Science & Business Media, 2010.

[65] Pushkar Joshi and Nathan A Carr. Repoussé: Automatic inflation of 2d artwork. In *SBM*, pages 49–55, 2008.

[66] Sergio N Silva Junior, Felipe C Chamone, Renato C Ferreira, and Erickson R Nascimento. A 3d modeling methodology based on a concavity-aware geometric test to create 3d textured coarse models from concept art and orthographic projections. *Computers & Graphics*, 76:73–83, 2018.

[67] Matthew Kaplan, Bruce Gooch, and Elaine Cohen. Interactive artistic rendering. *Proceedings first International Symposium on Non Photorealistic Animation and Rendering*, pages 67–74, 2000.

[68] Olga A Karpenko and John F Hughes. Smoothsketch: 3d free-form shapes from complex sketches. *ACM Transactions on Graphics (TOG)*, 25(3):589–598, 2006.

[69] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):29, 2013.

[70] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

[71] Misha Kazhdan and Hugues Hoppe. An adaptive multi-grid solver for applications in computer graphics. In *Computer Graphics Forum*, volume 38, pages 138–150. Wiley Online Library, 2019.

[72] Misha Kazhdan, Ming Chuang, Szymon Rusinkiewicz, and Hugues Hoppe. Poisson surface reconstruction with envelope constraints. In *Computer Graphics Forum*, volume 39, pages 173–182. Wiley Online Library, 2020.

[73] Ismail Khalid Kazmi, Lihua You, and Jian Jun Zhang. A survey of sketch based modeling systems. In *Computer Graphics, Imaging and Visualization (CGIV), 2014 11th International Conference*, pages 27–36. IEEE, 2014.

[74] D.R. Khanna. *Biology of Arthropoda.* Discovery Publishing House, 2004. ISBN 9788171418978. URL https://books.google.com.br/books?id=Hd40EDo4gbwC.

[75] Vladislav Kraevoy, Alla Sheffer, and Michiel van de Panne. Modeling from contour drawings. In *Proceedings of the 6th Eurographics Symposium on Sketch-Based interfaces and Modeling*, pages 37–44. ACM, 2009.

[76] Jörn H Kruhl. *Drawing geological structures.* John Wiley & Sons, 2017.

[77] J Landin. Rediscovering the forgotten benefits of drawing. *Scientific American*, 4, 2015.

[78] Manfred Lau, Greg Saul, Jun Mitani, and Takeo Igarashi. Modeling-in-context: user design of complementary objects with a single photo. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, pages 17–24. Eurographics Association, 2010.

[79] Sangwon Lee, David Feng, and Bruce Gooch. Automatic construction of 3d models from architectural line drawings. In *Proceedings of the 2008 symposium on Interactive 3D graphics and games*, pages 123–130. ACM, 2008.

[80] Victor Lempitsky and Denis Ivanov. Seamless mosaicing of image-based texture maps. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6. IEEE, 2007.

[81] Zohar Levi and Craig Gotsman. Artisketch: A system for articulated sketch modeling. In *Computer Graphics Forum*, volume 32, pages 235–244. Wiley Online Library, 2013.

[82] Changjian Li, Hao Pan, Yang Liu, Xin Tong, Alla Sheffer, and Wenping Wang. Bendsketch: modeling freeform surfaces through 2d sketching. *ACM Transactions on Graphics (TOG)*, 36(4):125, 2017.

[83] Shengjun Liu, Charlie CL Wang, Guido Brunnett, and Jun Wang. A closed-form formulation of hrbf-based surface reconstruction by approximate solution. *Computer-Aided Design*, 78:147–157, 2016.

[84] Zhaoliang Lun, Matheus Gadelha, Evangelos Kalogerakis, Subhransu Maji, and Rui Wang. 3d shape reconstruction from sketches via multi-view convolutional networks. In *2017 International Conference on 3D Vision (3DV)*, pages 67–77. IEEE, 2017.

[85] Laura Maaske. Unfolding the heart, 2013. URL https://medimagery.com/untying-the-knot-your-heart-is-actually-a-spiral/.

[86] I. Macedo, J. P. Gois, and L. Velho. Hermite Radial Basis Functions Implicits. *Computer Graphics Forum*, 2011. ISSN 1467-8659. doi: 10.1111/j.1467-8659.2010.01785.x.

[87] Ives Macêdo, João Paulo Gois, and Luiz Velho. Hermite interpolation of implicit surfaces with radial basis functions. In *2009 XXII Brazilian Symposium on Computer Graphics and Image Processing*, pages 1–8. IEEE, 2009.

[88] Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Loring S. Holden, J. D. Northrup, and John F. Hughes. Art-based rendering with continuous levels of detail. *Proceedings first International Symposium on Non Photorealistic Animation and Rendering*, pages 59–66, 2000.

[89] Bethann Garramon Merkle. Drawn to science. *Nature*, 562(7725):S8–S8, 2018.

[90] Patti Mollica. *Color Theory: An essential guide to color-from basic principles to practical applications*, volume 53. Walter Foster, 2013.

[91] Renata Nascimento, Fabiane Queiroz, Allan Rocha, Tsang Ing Ren, Vinicius Mello, and Adelailson Peixoto. Colorization and illumination of 2d animations based on a region-tree representation. In *Graphics, Patterns and Images (Sibgrapi), 2011 24th SIBGRAPI Conference on*, pages 9–16. IEEE, 2011.

[92] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Fibermesh: designing freeform surfaces with 3d curves. *ACM transactions on graphics (TOG)*, 26(3):41, 2007.

[93] Andrew Nealen, Olga Sorkine, Marc Alexa, and Daniel Cohen-Or. A sketch-based interface for detail-preserving mesh editing. In *ACM SIGGRAPH 2007 courses*, page 42. ACM, 2007.

[94] Marc Nienhaus and Jürgen Döllner. Blueprints: illustrating architecture and technical parts using hardware-accelerated non-photorealistic rendering. In *Proceedings of Graphics Interface 2004*, pages 49–56. Canadian Human-Computer Communications Society, 2004.

[95] J. D. Northrup and Lee Markosian. Artistic silhouettes: A hybrid approach. *Proceedings first International Symposium on Non Photorealistic Animation and Rendering*, pages 31–37, 2000.

[96] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4531–4540, 2019.

[97] Luke Olsen, Faramarz F Samavati, Mario Costa Sousa, and Joaquim A Jorge. Sketch-based modeling: A survey. *Computers & Graphics*, 33(1):85–103, 2009.

[98] Luke Olsen, Faramarz Samavati, and Joaquim Jorge. Naturasketch: Modeling from images and natural sketches. *IEEE Computer Graphics and Applications*, 31(6):24–34, 2011.

[99] N Platnick. Systematics and biogeography: Cladistics and vicariance, 1981.

[100] Frits H Post, Benjamin Vrolijk, Helwig Hauser, Robert S Laramee, and Helmut Doleisch. Feature extraction and visualisation of flow fields. In *Eurographics (STARs)*, 2002.

[101] Saulo Ramos, Diogo Fernando Trevisan, Harlen C Batagelo, Mario Costa Sousa, and Joao Paulo Gois. Contour-aware 3d reconstruction of side-view sketches. *Computers & Graphics*, 77:97–107, 2018.

[102] Alec Rivers, Frédo Durand, and Takeo Igarashi. 3d modeling with silhouettes. *ACM Trans. Graph.*, 29(4):109:1–109:8, July 2010. ISSN 0730-0301. doi: 10.1145/1778765.1778846. URL http://doi.acm.org/10.1145/1778765.1778846.

[103] Dale Rolfsen. *Knots and links*, volume 346. American Mathematical Soc., 2003.

[104] Faramarz F Samavati and Richard H Bartels. Local filters of b-spline wavelets. In *Proc. of Intl. Workshop on Biometric Technologies*, pages 105–110, 2004.

[105] CHARLES MD Santos. Description of two new species of neorhagio (diptera, tabanomorpha, rhagionidae), and remarks on a controversial female character. *Zootaxa*, 1174(1):49–62, 2006.

[106] Charles Morphy D Santos, Dalton De Souza Amorim, et al. Chrysopilus (diptera: Rhagionidae) from brazil: redescription of chrysopilus fascipennis bromley and description of eleven new species. *Zootaxa*, 1510(1):1–33, 2007.

[107] Charles Morphy D. Santos, Dalton S. Amorim, Bruna Klassa, Diego A. Fachin, Silvio S. Nihein, CLAUDIO J. B. DE CARVALHO, RAFAELA L. FALASCHI, CÁTIA A. MELLO-PATIU, MÁRCIA S. COURI, SARAH S. OLIVEIRA, VERA C. SILVA, GUILHERME C. RIBEIRO, RENATO S. CAPELLARI, and CARLOS JOSÉ E.

LAMAS. On typeless species and the perils of fast taxonomy. *Systematic Entomology*, 41(3):511–515, 2016. doi: 10.1111/syen.12180.

[108] Stephen Se and Piotr Jasiobedzki. Photo-realistic 3d model reconstruction. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 3076–3082. IEEE, 2006.

[109] Cloud Shao, Adrien Bousseau, Alla Sheffer, and Karan Singh. Crossshade: shading concept sketches using cross-section curves. *ACM Transactions on Graphics*, 31(4), 2012.

[110] Byeong-Seok Shin and Yeong Gil Shin. Fast 3D solid model reconstruction from orthographic views. *Computer-Aided Design*, 30(1):63–76, 1998. ISSN 00104485. doi: 10.1016/S0010-4485(97)00054-7. URL http://www.sciencedirect.com/science/article/pii/S0010448597000547.

[111] Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, et al. Textured neural avatars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2397, 2019.

[112] Sudipta N Sinha, Drew Steedly, Richard Szeliski, Maneesh Agrawala, and Marc Pollefeys. Interactive 3d architectural modeling from unordered photo collections. *ACM Transactions on Graphics (TOG)*, 27(5):1–10, 2008.

[113] Mario Costa Sousa, Kevin Foster, Brian Wyvill, and Faramarz Samavati. Precise Ink Drawing of 3D Models. *Computer Graphics Forum*, 22(3):369–379, 2003. ISSN 01677055. doi: 10.1111/1467-8659.00684.

[114] Thomas Strothotte and Stefan Schlechtweg. *Non-photorealistic computer graphics: modeling, rendering, and animation.* Morgan Kaufmann, 2002.

[115] Ryohei Suzuki and Takeo Igarashi. Collaborative 3d modeling by the crowd. In *Graphics Interface*, pages 124–131, 2017.

[116] Daniel Sỳkora, Ladislav Kavan, Martin Čadík, Ondřej Jamriška, Alec Jacobson, Brian Whited, Maryann Simmons, and Olga Sorkine-Hornung. Ink-and-ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters. *ACM Transactions on Graphics (TOG)*, 33(2):16, 2014.

[117] James T Todd. The visual perception of 3d shape. *Trends in cognitive sciences*, 8(3): 115–121, 2004.

[118] Emmanuel Turquin, Jamie Wither, Laurence Boissieux, Marie-Paule Cani, and John F Hughes. A sketch-based interface for clothing virtual characters. *IEEE Computer graphics and applications*, 27(1), 2007.

[119] Michael Waechter, Mate Beljan, Simon Fuhrmann, Nils Moehrle, Johannes Kopf, and Michael Goesele. Virtual rephotography: Novel view prediction error for 3d reconstruction. *ACM Transactions on Graphics (TOG)*, 36(1):8, 2017.

[120] Fang Wang, Le Kang, and Yi Li. Sketch-based 3d shape retrieval using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1875–1883, 2015.

[121] Keith Wiley and Lance R Williams. Representation of interwoven surfaces in 2 1/2 d drawing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 65–74, 2006.

[122] Jamie Wither, Frédéric Boudon, M-P Cani, and Christophe Godin. Structure from silhouettes: a new paradigm for fast sketch-based design of trees. In *Computer Graphics Forum*, volume 28, pages 541–550. Wiley Online Library, 2009.

[123] Baoxuan Xu, William Chang, Alla Sheffer, Adrien Bousseau, James McCrae, and Karan Singh. True2form: 3d curve networks from 2d sketches via selective regularization. *ACM Trans. Graph.*, 33(4):131:1–131:13, July 2014. ISSN 0730-0301. doi: 10.1145/2601097.2601128. URL `http://doi.acm.org/10.1145/2601097.2601128`.

[124] Baoxuan Xu, William Chang, Alla Sheffer, Adrien Bousseau, James McCrae, and Karan Singh. True2form: 3d curve networks from 2d sketches via selective regularization. *ACM Transactions on Graphics*, 33(4), 2014.

[125] Chen Yang, Dana Sharon, and Michiel van de Panne. Sketch-based modeling of parameterized objects. In *SIGGRAPH Sketches*, page 89. Citeseer, 2005.

[126] Linjie Yang, Jianzhuang Liu, and Xiaoou Tang. Complex 3d general object reconstruction from line drawings. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1433–1440. IEEE, 2013.

[127] Chih-Kuo Yeh, Peng Song, Peng-Yen Lin, Chi-Wing Fu, Chao-Hung Lin, and Tong-Yee Lee. Double-sided 2.5 d graphics. *IEEE transactions on visualization and computer graphics*, 19(2):225–235, 2012.

[128] Chih-Kuo Yeh, Shi-Yang Huang, Pradeep Kumar Jayaraman, Chi-Wing Fu, and Tong-Yee Lee. Interactive high-relief reconstruction for organic and double-sided objects from a photo. *IEEE transactions on visualization and computer graphics*, 23 (7):1796–1808, 2017.

[129] Johannes Zander, Tobias Isenberg, Stefan Schlechtweg, and Thomas Strothotte. High quality hatching. *Computer Graphics Forum*, 23(3):421–430, 2004.

[130] Shusheng Zhang, Yunfei Shi, Haitao Fan, Rui Huang, and Julu Cao. Serial 3D model reconstruction for machining evolution of rotational parts by merging semantic and graphic process planning information. *CAD Computer Aided Design*, 42(9):781–794, 2010. ISSN 00104485. doi: 10.1016/j.cad.2010.04.007. URL `http://dx.doi.org/10.1016/j.cad.2010.04.007`.

[131] Lifeng Zhu, Takeo Igarashi, and Jun Mitani. Soft folding. *Computer Graphics Forum*, 32(7):167–176, 2013.

APPENDIX

Reconstruction Methods

- 3D Input Data:
  - ◇ Gois et al. [46], Nascimento et al. [91], Ramos et al. [101]
- Poisson Surface Reconstruction:
  - ◇ Kazhdan and Hoppe [69], Kazhdan et al. [70], Kazhdan and Hoppe [71], Kazhdan et al. [72]
- Smooth Signed Distance Surface Reconstruction (SSD Reconstruction):
  - ◇ Calakli and Taubin [19]
- Hermite-RBF Reconstruction:
  - ◇ Gois et al. [44], Macêdo et al. [87]
- Rotational Blending Surface:
  - ◇ Anastacio et al. [4], Cherlin et al. [24]

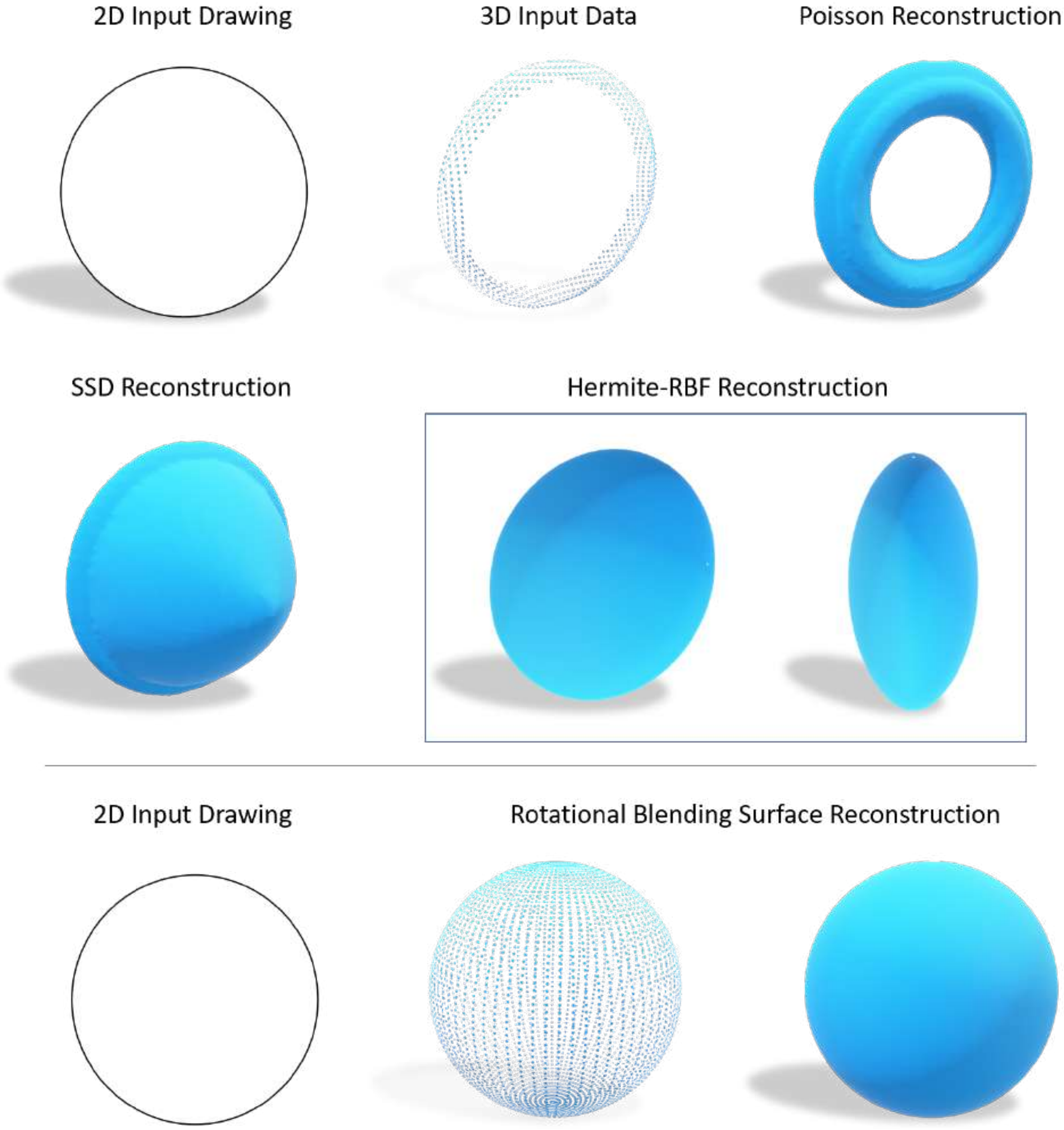


Figure 97: Comparison of Reconstruction Methods for circle drawing.

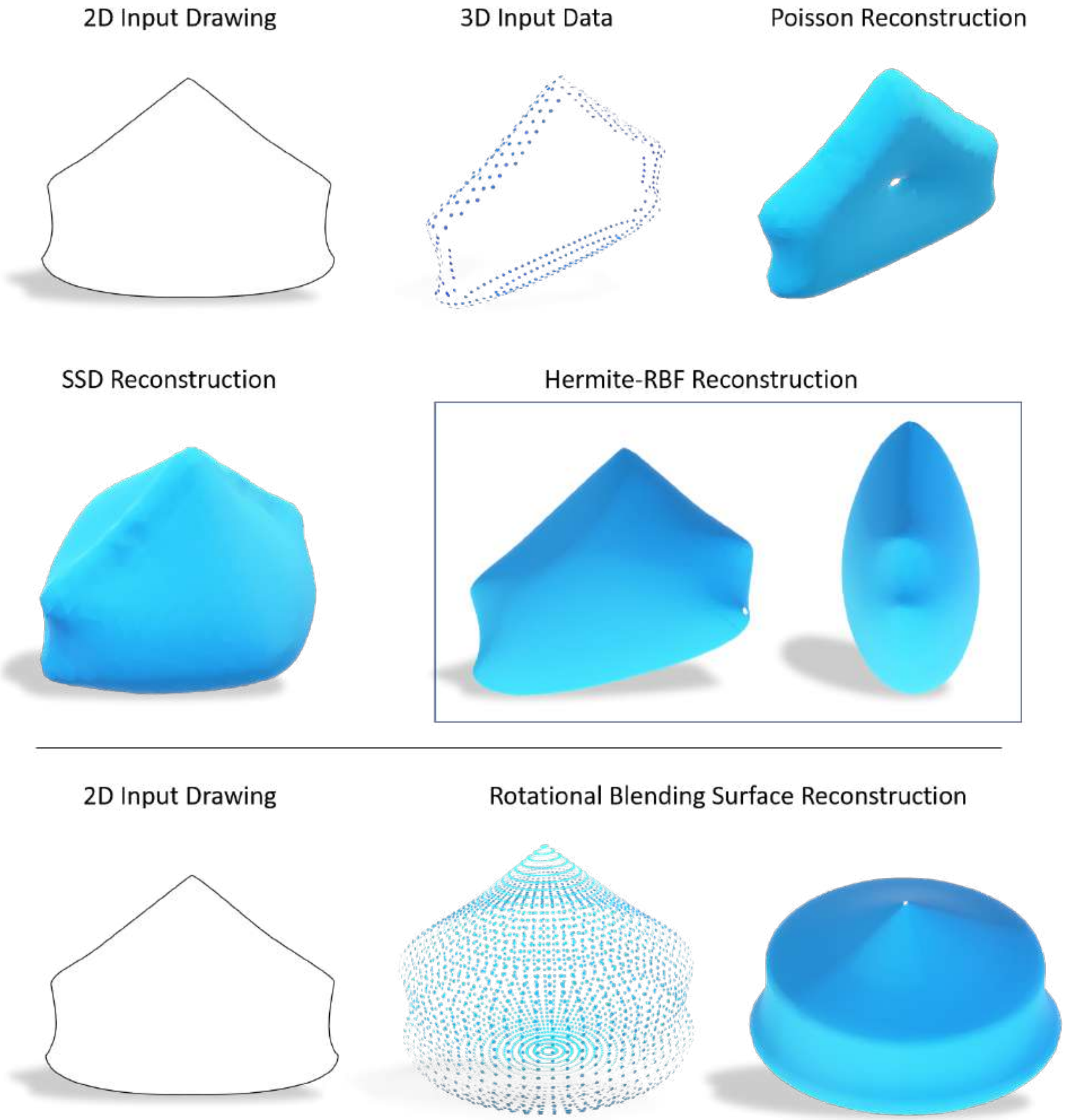


Figure 98: Comparison of Reconstruction Methods for *tagma 10* from Fig. 93b.