

Effective Line Drawing Generation

Optimizing Line Drawings Based on Human 3D Shape Perception

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Visual Computing

eingereicht von

Pascal Plank, BSc

Matrikelnummer 01225804

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Assistant Prof. Dipl.-Ing. Dr. techn. Ivan Viola

Mitwirkung: Univ. Ass. Dr.techn. Manuela Waldner

Wien, 30. April 2018

Pascal Plank

Ivan Viola

Effective Line Drawing Generation

Optimizing Line Drawings Based on Human 3D Shape Perception

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Visual Computing

by

Pascal Plank, BSc

Registration Number 01225804

to the Faculty of Informatics

at the TU Wien

Advisor: Assistant Prof. Dipl.-Ing. Dr. techn. Ivan Viola

Assistance: Univ. Ass. Dr.techn. Manuela Waldner

Vienna, 30th April, 2018

Pascal Plank

Ivan Viola

Erklärung zur Verfassung der Arbeit

Pascal Plank, BSc
Kohlgasse 47 Top 35, 1050 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 30. April 2018

Pascal Plank

Danksagung

Allen voran, möchte ich all denjenigen danken, die mich während der Zeit meines Informatikstudiums seit der Schulzeit unterstützt haben. Ich möchte besonders Alexandra Plank, Patricia Plank und Pawel Miedzinski danken, ohne deren Unterstützung diese Arbeit nicht möglich gewesen wäre. Ich möchte weiters besonders Ivan Viola für seine professionelle Begleitung dieser Arbeit sowie Manuela Waldner für ihren kenntnisreichen Input in den Bereichen Datenanalyse und statistische Tests danken.

Abschließend bedanke ich mich bei allen Teilnehmern meiner User Study, die es mir erlaubten in kurzer Zeit die benötigten Daten zu sammeln. Namentlich sind dies Dominik Bauer, Andreas Deutsch, Johannes Diep, Igor Kotic, Verena Kremsner, Katharina Krösl, Johannes Löffler, Michaela Löffler, Pawel Miedzinski, Alexandra Plank, Patricia Plank, Stefan Puxbaum, Gwendolyn Rippberger, Johannes Unterguggenberger, Ivan Viola, Johannes Warmuth und Manuela Waldner. Danke euch allen.

Acknowledgements

First and foremost, my thanks go to everyone who has supported me during my pursuit of studying computer science since high school. I would like to thank especially Alexandra Plank, Patricia Plank, and Pawel Miedzinski, without your support this thesis would not have been possible. I especially want to thank Ivan Viola for his professional guidance and feedback for this thesis. Likewise, my thanks go to Manuela Waldner for her knowledgeable input on data analysis and statistical testing.

Lastly, I would like to thank all the participants who took part in the user study and allowed me to collect the necessary data quickly, namely Dominik Bauer, Andreas Deutsch, Johannes Diep, Igor Kopic, Verena Kremsner, Katharina Krösl, Johannes Löffler, Michaela Löffler, Pawel Miedzinski, Alexandra Plank, Patricia Plank, Stefan Puxbaum, Gwendolyn Rippberger, Johannes Unterguggenberger, Ivan Viola, Johannes Warmuth and Manuela Waldner. Thank you all.

Kurzfassung

Fortschrittliche Rendering-Algorithmen wie Suggestive Contours sind in der Lage Objekte im Stil von Linienzeichnungen mit vielen Detailgraden darzustellen. Wie allerdings ein ansprechender Detailgrad gewählt werden soll, basiert auf rein visueller Ästhetik und nicht auf fundierten Charakteristika wie beispielsweise der Genauigkeit von menschlicher 3D Gestaltwahrnehmung. Das Ziel dieser Arbeit ist es einen neuen Ansatz zur effektiven Generierung von Linienzeichnungen im Stil von Suggestive Contours zu finden, welcher für menschliche 3D Gestaltwahrnehmung optimiert ist und zugleich den Anteil an Tinte im Bild auf ein Minimum beschränkt. Die dabei entwickelte Meta-Heuristik zur Optimierung von Linienzeichnungen benutzt Schwellwerte basierend auf empirisch gesammelten Daten der menschlichen Gestaltwahrnehmung. Die Heuristik kann neben der Optimierung in Bezug auf 3D Gestaltwahrnehmung auch für die Optimierung anderer Charakteristika wie beispielsweise Cognitive Load oder auch für andere Linienzeichensstile wie Ridges and Valleys genutzt werden.

Die entwickelte Optimierungsroutine basiert auf einer durchgeführten Wahrnehmungsstudie mithilfe des Gauge Figure Tasks wobei mehr als 17.000 Datenpunkte von Oberflächennormalen in Suggestive Contours Bildern erhoben wurden. Durch diese Studie liefert die vorliegende Arbeit neue Erkenntnisse für ein tieferes Verständnis menschlicher Gestaltwahrnehmung. Besonders die Genauigkeit von 3D Formwahrnehmungen sowie Gestaltzweideutigkeiten in Bezug auf wechselnde Detailgrade und in Bezug auf den verwendeten Objekttypus wurde untersucht. Zusätzlich wurden die Datenpunkte genutzt, um zwei Pixel-basierte Wahrnehmungscharakteristika zu berechnen: die optimale Größe einer Nachbarschaftsfläche für die Schätzung von 3D Formen sowie der optimale lokale Tinten-Prozentwert in dieser Nachbarschaft.

In der Analyse konnte eine Nachbarschaftsgröße von 36 Pixel mit einem optimalen Tintenprozentatz von 17.3% identifiziert werden. Diese Schwellwerte werden genutzt, um Suggestive Contours Bilder in einem Nachbearbeitungsschritt mittels eines greedy Nearest-Neighbor-Ansatzes zu optimieren. Die entwickelte Meta-Heuristik liefert visuell überzeugende Ergebnisse, wobei jeder Pixelwert möglichst nahe an den identifizierten Schwellwerten liegt. In der praktischen Anwendung kann dieses Optimierungsschema in Bereichen, in denen 3D Gestaltwahrnehmung wichtig ist, wie beispielsweise für Möbelhandbücher oder Architekturdarstellungen, eingesetzt werden. Sowohl die empirischen Erkenntnisse zur menschlichen Gestaltwahrnehmung als auch die praktische Anwendung

der Ergebnisse formen die Basis zur Optimierung anderer Linienzeichnungsalgorithmen sowie für ein besseres Verständnis wie Menschen 3D Gestalten aus Linienzeichnungen wahrnehmen.

Abstract

Advanced rendering algorithms such as suggestive contours are able to depict objects in the style of line drawings with various levels of detail. How to select an appropriate level of detail is based on visual aesthetics rather than on substantial characteristics like the accuracy of 3D shape perception. The aim of this thesis is to develop a novel approach for effectively generating line drawings in the style of suggestive contours that are optimized for human 3D shape perception while retaining the amount of ink to a minimum. The proposed post-processing meta-heuristic for optimizing line drawings uses empirical thresholds based on probing human shape perception. The heuristic can also be used to optimize line drawings in terms of other visual characteristics, e.g., cognitive load, and for other line drawings styles such as ridges and valleys.

The optimization routine is based on a conducted perceptual user study using the gauge figure task to collect more than 17,000 high-quality user estimates of surface normals from suggestive contours renderings. By analyzing these data points, more in-depth understanding of how humans perceive 3D shape from line drawings is gained. Particularly the accuracy of 3D shape perception and shape ambiguity in regards to changing the level of detail and type of object presented is investigated. In addition, the collected data points are used to calculate two pixel-based perceptual characteristics: the optimal size of a local neighborhood area to estimate 3D shape from and the optimal local ink percentage in this area.

In the analysis, a neighborhood size of 36 pixels with an optimal ink percentage of 17.3% could be identified. These thresholds are used to optimize suggestive contours renderings in a post-processing stage using a greedy nearest neighbor optimization scheme. The proposed meta-heuristic procedure yields visually convincing results where each pixel value is close to the identified thresholds. In terms of practical application, the optimization scheme can be used in areas where high 3D shape understanding is essential such as furniture manuals or architectural renderings. Both the empirical results regarding shape understanding as well as the practical applications of the thesis's results form the basis to optimize other line drawing methods and to understand better how humans perceive shape from lines.

Contents

Kurzfassung	xi
Abstract	xiii
Contents	xv
1 Introduction	1
2 Background and Previous Work: What Lines Convey and What Humans Understand	5
2.1 Lines and Line Drawings	5
2.2 Silhouettes, Creases, Outlines, Contours, and Suggestive Contours . .	8
2.3 Perception of Lines	12
2.4 Perception of 3D Graphics	15
2.5 Depth Perception and Depth Cues	17
2.6 Psychophysical Measuring Techniques	18
3 State of the Art	23
4 Quality and Expressiveness of Lines	27
4.1 Technical Pipeline: From Mesh to Empirical Sample Points	27
4.2 Research Questions and Analysis Process	28
5 Technical Implementation	31
5.1 Approaches to Implementing Contours and Suggestive Contours	31
5.2 Gauge Figure Task Framework	36
6 Shape Perception User Study	39
6.1 Stimuli	39
6.2 Sampling Procedure	41
6.3 Experiment Setup	42
7 Results and Discussion	45
7.1 Results of Analysis and Discussion	45
	xv

7.2 Optimizing Suggestive Contours for Shape Perception	58
8 Conclusion	63
9 Future Work	65
List of Figures	69
List of Tables	73
Glossary	75
Bibliography	77

Introduction

A popular saying goes “a picture is worth a thousand words” and there are studies providing empirical evidence that graphical representations are superior to information provided by merely textual means [WST01]. Visualization as a human form of expression is as old as the first human cave drawings from up to 40.000 years ago as shown in Figure 1.1. Over the last millennia, human skill to convey information via graphical representation has improved beyond simple depictions of animals and hunting scenes in the paleolithic era and has evolved significantly since then - from the work of Leonardo Da Vinci to modern photorealistic renderings and information visualization systems.



Figure 1.1: Example of a 30.000-year-old cave painting of a hyena found in the Chauvet Cave, France. (Image courtesy Wikimedia Commons)

One category of modern non-photorealistic visualization and rendering styles is called “line drawing”, which is inspired by human artists depicting scenes utilizing pencil strokes. This form of rendering allows the artist to realize two key aspects of visualization, namely to be a tool to enable the user to gain insight into the data and a way to form a mental vision, image or picture of something that is not visible, present to the sight, or of an

abstraction. Line drawings are mainly used in visualization for the goal of presenting information and communicating results as opposed to the goals of data exploration or analysis, which modern visualization systems are often capable of. Despite its seniority among modern visualization techniques, line drawings are still prevalent in a wide array of applications ranging from simple sketches and instructions to do-it-yourself manuals and architectural renderings as shown in Figure 1.2.



Figure 1.2: Example of a line drawing in a modern architectural sketch. (Copyright by Braunger Wörtz Architekten)

This thesis' aim is to gain a better understanding of how shape of 3D objects is perceived by humans when depicted in the style the rendering technique of suggestive contours. Furthermore, it is examined how the quality of perception varies with varying degree of detail as well as shape category. In addition, the empirical results found in course of this thesis are used to create a novel rendering approach to suggestive contours and other line drawing algorithms to generate effective line drawings in terms of human 3D shape perception. Effective in this context means to create images that are high in perceived shape information while retaining low levels of ink. The resulting algorithm may be used in architectural renderings or in furniture user handbooks to produce images with high expressiveness in 3D shape understanding while reducing the perceived visual complexity of the drawn object to a minimum.

This is achieved by first evaluating possible implementation approaches for suggestive contours. Then, the error of human shape perception is quantified by using the gauge figure task to obtain samples from several participants from a set of predefined objects. By analyzing this error and by deriving optimal thresholds of human shape perception, insights into how humans perceive shape in line drawings are gained.

The user study for evaluating the error in human shape perception is performed by a group of unpaid participants who have to evaluate four different objects in five levels of line drawing quality of the suggestive contours rendering technique. Each object falls into its own shape category, e.g., organic, abstract, or familiar object with a smooth surface, to consider both the level of detail as well as the shape category as independent variables.

The result of each participant's gauge figure task is a set of 960 sample points on a virtual hemisphere representing all possible angles when seeing a rendered object with a fixed

viewpoint. These sample points are then cleaned, analyzed, and extracted thresholds are used in a novel approach to line drawing rendering for optimizing suggestive contours in regards to human visual perception of 3D shape.

In the following Chapter 2 the background to this thesis is described, specifically the basics of lines and line drawings as well as psychological measuring techniques for the human visual perception of 3D graphics. In Chapter 3 “State of the Art” the latest results in research regarding line drawings is summarized, an overview of previous results of performed gauge figure task studies is given, and the most relevant related work to the topics of this thesis is presented.

In Chapter 4 “Quality and Expressiveness of Lines” the entire technical pipeline on a conceptual level from the definition of stimulus objects for the gauge figure task to optimizing the suggestive contours line drawing method is given, and the set of predefined research questions of interest for this thesis are presented. Following this, in Chapter 5 “Technical Implementation” the details of implementation for this thesis are explained with a focus on the different implementation methods for suggestive contours and the custom-made gauge figure task framework for the user study.

Chapter 6 describes the performed user study in detail beginning with a description of the used stimulus objects, the definition of the user study sampling procedure, the practical experiment setup and ending with the demographics of the participants. In Chapter 7 “Result and Discussion” the performed analysis and its results are explained as well as visualized in regards to all defined research questions. In addition, the proposed algorithm for optimizing suggestive contours using the empirical results from the user study is presented. In Chapter 8 “Conclusion” the meaning, the impact of the thesis’ results, and how these results fit into previous research are discussed. To conclude the thesis, Chapter 9 “Future Work” presents possible extension points to this thesis and improvements of the proposed meta-heuristic for future researchers to investigate.

Background and Previous Work: What Lines Convey and What Humans Understand

To give a better overview of lines and their perception by humans, this chapter provides detailed information on lines and line drawings, on silhouettes, contours and suggestive contours, on human perception of lines as well as 3D graphics, on human depth perception and the involved depth cues, and lastly on psychophysical measuring techniques such as the gauge figure task.

2.1 Lines and Line Drawings

The work on computer-generated line drawings coincides with the beginnings of computer graphics itself since lines are the simplest form of rendering 2D images from 3D scenes. In the early 1960s the Bresenham's line algorithm by IBM, the first commonly used line drawing algorithm for plotters, was introduced [Bre65]. This algorithm determines the necessary points in a pixel grid that should be colored to approximate a straight line between two points. With the introduction of the Scanline-Algorithm for rendering [WREE67], [Bou70], [Wat70], hidden-line algorithms [App67], [Gal69], [Hor82], [Lou70], and eventually Z-buffering for managing occlusions [Str74], the rendering of lines in computer graphics was brought to the basic levels of line rendering we know and use today.

In general, there are two types of rendering when it comes to computer graphics: photorealism and non-photorealism. The last three decades in computer graphics research in developing new rendering techniques both in academia and industry put a strong focus on photorealistic rendering, e.g., for games and animated movies, to the point

2. BACKGROUND AND PREVIOUS WORK: WHAT LINES CONVEY AND WHAT HUMANS UNDERSTAND

of creating the lifelike images we know from movies like *Avatar* by James Cameron from 2009. Aside from this mainstream development of new photorealistic rendering techniques, non-photorealistic methods were on the rise in areas such as visualization, e.g., information visualization, explorational data analysis, molecular rendering, and simulation visualization. Non-photorealistic rendering (NPR) research focuses on how to use the principles of visual information abstraction utilized by artists for many millennia to avoid unnecessary detail and therefore reduce the cognitive load for viewers.

Line renderings are one of the commonly used techniques in non-photorealistic renderings. As such, line drawings follow Tufte’s Visualization Rules and can therefore be regarded as an appropriate tool to convey information visually [Tuf86] [Tuf90]. Tufte’s Visualization Rules are:

- Make all visual distinctions as subtle as possible, but still clear and effective
- Maximize ink for the display of information and minimize non-data ink
- Hide data that does not make a difference in what you are trying to depict
- Minimize clutter
- Separate figure and background

In contrast to high-level methods for visual abstraction that are based on smart visibility, line drawings fall into the category of low-level techniques that are based on a stylized depiction of a scene. High-level methods include view-dependent transparency [DWE02], cutaways and breakaways [DWE03], volume splitting [IDSC04], importance-driven feature enhancement [VKG05], as well as hybrid visibility composing and masking for illustrative rendering [BRV⁺10]. Besides suggestive contours [DFRS03], low-level abstraction techniques include metal and tone shading [GGSC98], hatching [PHWF01], stipple drawings [LME⁺02], curvature-based ridge and valley enhancement [KWTM03], and lit sphere maps [SMGG01].

The method to create lines in line drawings is usually referred to as a “line generator”. A line generator may be a simple mathematical equation describing line pixels while more complex line generators utilize several levels of processing to create the desired results. Regardless of its complexity, the lines created by a specific type of line generator fall into one of two categories: image-space feature lines or object-space feature lines, of which object-space feature lines can either be view-dependent or view-independent. Image-space lines are lines that are extracted from an image, which is usually rendered with Lambertian shading and a single light source. These lines are well-suited for GPU processing but lack the ability to be stylized easily. Image-space lines include for example the Canny edge detector [Can87] or isophotes corresponding to toon shading boundaries [The96].

In contrast to image-space lines, object-space feature lines are computed directly on the 3D surface of an object. If view-independent, it is possible to precompute object-space lines since they are an intrinsic property of shape, e.g., constant-altitude lines on topographic maps or creases. In the view-dependent case, the lines must be recomputed per frame. Examples for view-dependent lines include outlines or suggestive contours.

Previously restricted to offline rendering, real-time stylized line drawings were shown to be possible by Markosian around 1997 [MKG⁺97]. Soon the effect of temporal coherence was used in line and silhouette stylization for real-time rendering [NM00] [MMK⁺00] [KDMF03]. A single line drawing can incorporate several elements from the non-realistic technique toolbox such as cartoon shading, hatching, outlines, detail marks on the surface, stylized strokes, and paper effects. In order of complexity and building on each other, line drawings usually contain silhouettes or outlines, contours, and more complex lines like suggestive contours or apparent ridges. The details of these line types are discussed in Section 2.2.

Line drawings are generally hard to simulate well, i.e., like a real artist, because there are several dimensions in artistic style, e.g., emotional to explanatory, veridical to abstract, and loose drawings to controlled ones. For example, Figure 2.1 shows an illustration of Michelangelo, which is not only hard for a computer to generate but also for humans to understand because the illustration focuses on creating emotions rather than depicting exactly what is happening in a scene. In contrast to this, modern schematics of tools or do-it-yourself furniture are easily generated by computers due to their strict explanatory and veridical depiction of a scene.



Figure 2.1: Example of a line drawing by Michelangelo. It is not quite clear what is happening in the scene except that the illustration shows a battle scene or a fight. This line drawing focuses on conveying emotion rather than exact and veridical information.

In contrast to hard to generate line drawing techniques, some methods used by artists are well understood and can be described by algorithms. For example, simple abstractions may be generated by rounding the shapes and therefore removing lines with too much detail as shown in Figure 2.2 [DS00]. Other well understood techniques include hatching [HZ00] and controlled shading [RBD06].

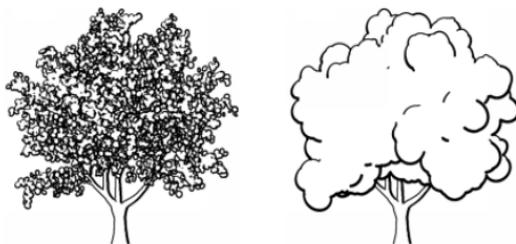


Figure 2.2: Example of simple abstractions created by renderings algorithms [DS00].

Even though some artistic illustration techniques are well understood, there are still many possible ways of conveying the same information with different line drawing renderings as shown in Figure 2.3 [CGL⁺08]. Smart algorithms try to bridge this gap to a real artist’s intuitive illustration technique, e.g.:

- Edge filtering: e.g., by Canny [Can87], and Kang and Wang [KW07]
- Image ridges and valleys: this approach generates lines at image intensity ridges and valleys by using areas of dark and light in a shaded image [LMLH07].
- Geometric ridges and valleys: this method generates lines in areas where the surface normal is changing rapidly indicating smoothed creases, i.e., lines of normal discontinuity [JDA07]. This approach is independent of the lighting conditions in a scene.
- Extension of contours: this technique renders additional contours from nearby views to incorporate occluding contours. It is similar to artists who often include contours that are not visible in a scene to hint an object’s hidden geometric structure. The suggestive contours method is based on this idea of contour extension [DFRS03]. Due to its usage in artists’ illustrations and its general independence on lighting conditions in a scene, suggestive contours is used in this thesis to explore human perception of 3D shape from line drawings.

2.2 Silhouettes, Creases, Outlines, Contours, and Suggestive Contours

Modern line drawings consist of a variety of different line types and include outlines, silhouette edges, silhouettes, contours, creases, and more complex types of lines such as ridges and valleys, or suggestive contours. Sometimes these terms do not have a consistent definition in the literature, e.g., silhouettes vs. silhouette edge vs. contours. To provide clarity for this thesis and the subsequent chapters, the terms are used according to the followings definitions.

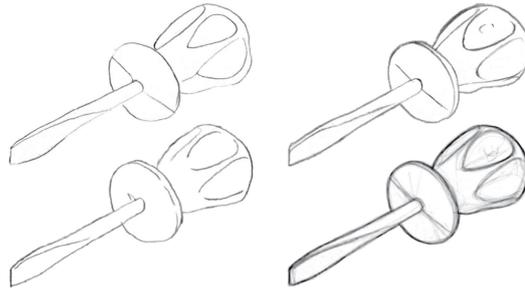


Figure 2.3: Examples of how there are several methods of conveying the same information with different line rendering styles [CGL⁺08].

- **Silhouette:** A silhouette is defined as the shadow of an object created by a directional light source in the direction of the view vector v projected onto a plane, which surface normal n is perpendicular to v , i.e., the set of points occluded by an object in an orthographic projection in the direction of v onto a perpendicular plane.
- **Outline:** The outline is defined as the set of points on the edges of a silhouette, i.e., where sharp discontinuities in depth are present [PB13]. In other words, the boundaries between an object and its background.
- **Contours:** In the context of this thesis a contour is defined as the set of locations of points where the normal vector $n(p)$ and the view vector $v(p)$ for a point p on a smooth and closed surface S are mutually perpendicular, i.e., $\langle n(p) \cdot v(p) \rangle = 0$ [DFRS03] or $\vartheta = 90^\circ$ [LP16]. In the discrete case, edges are highlighted when the dot product of v and the normals of the incident triangles changes its sign, i.e., the edge between a front-facing and a back-facing triangle. Contours are view-dependent and important for gaining the first impression of an object, but they are not capable of depicting salient regions nor are they useful to provide a proper spatial impression of an object.
- **Silhouette edges:** In this thesis, the definition of “silhouette edges” by Markosian et al. is used [MKG⁺97], which is equal to that of the contour lines by DeCarlo et al. from above [DFRS03].
- **Crease Lines:** Crease lines are defined as the set of edges where incident triangles change strongly according to their dihedral angle, i.e., when the angles of the normals of the incident triangles n_i and n_j change strongly [LP16]. The threshold for being a “strong” angle is defined by the user with τ . As the magnitude of the curvature can be estimated by observing the change in normals, the set of points creating crease lines are the points where the underlying absolute value of maximum curvature exceeds a threshold, i.e., $\kappa \geq \tau$ or $\langle n_i \cdot n_j \rangle \geq \tau$. The resulting lines are not view-dependent, are only drawn along edges and depict strong edges

well while having the disadvantage of not being able to depict objects with small features appropriately.

2.2.1 Suggestive Contours

The most commonly used line drawing algorithms, which yield lines of highest quality today, include suggestive contours, ridges and valleys, and apparent ridges, see Figure 2.4 for a direct comparison. Suggestive contours are occluded contours visible when changing the current viewpoint slightly [DFRS03]. Ridges and valley lines are a generalization of creases on smooth surfaces generated by using local extrema of surface curvature along the principal curvature directions [IFP95], [TG96], [PKG03], [OBS04]. Apparent ridges are an extension of ridges and valleys that accounts for foreshortening, i.e., the perspective shortening of objects as they seem to recede into the distance, by using the extrema of view-dependent curvature of a given surface [JDA07]. Other modern types of line drawing algorithms include highlight lines [LMLH07], [DR07] and demarcating curves [KST08].

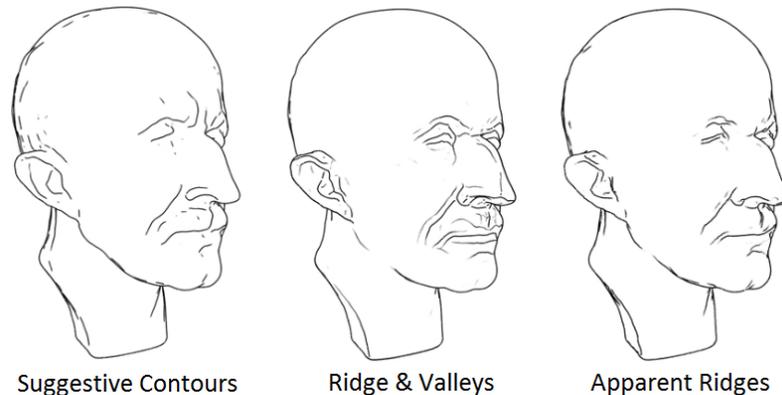


Figure 2.4: Examples of suggestive contours, ridges and valleys, and apparent ridges in a direct comparison [JDA07].

The main focus of this thesis lies in lines generated by the suggestive contours NPR-rendering method. Developed by DeCarlo et al. in 2003 [DFRS03], suggestive contours are a natural extension to contours as defined for this thesis. By this definition, contours are view-dependent, i.e., depending on the point of view the presented set of lines changes. Suggestive contours extend this idea by not only using a single point of view but by rendering the contours from “nearby points of view” as well, i.e., the contours one would see if the point of view changes slightly as depicted in Figure 2.5. In contrast to contours, this inclusion of nearby views makes suggestive contours coherent in time [LP16], meaning that it is possible to create smooth suggestive contours animations or to use them in interactive 3D renderings.

In their paper, DeCarlo et al. give three mathematically equivalent definitions of suggestive contours [DFRS03]:

- Definition I - Nearby views:** Suggestive contours are the set of points on the contour generator of nearby views of radial distance smaller than 90° , i.e., $[-45^\circ; \dots; +45^\circ]$. This situation is illustrated in Figure 2.5. DeCarlo et al. extended this definition by requiring points that are in contours in nearby views but do not have corresponding contours in any radially closer views.

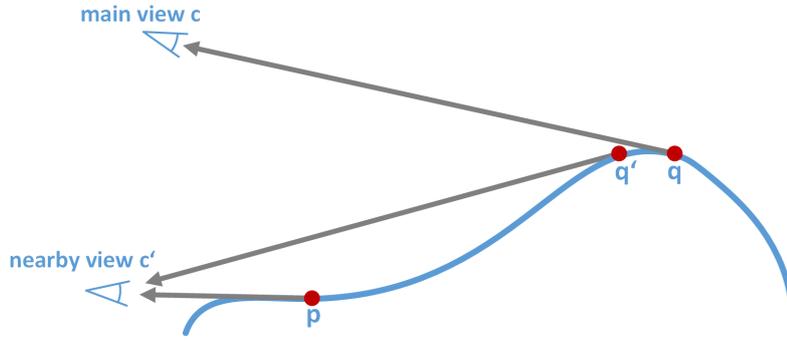


Figure 2.5: A situation showing a surface in the radial plane with both a contour point q and a suggestive contours point p from the main viewpoint c . When changing the viewpoint to c' , a contour is visible in p .

- Definition II - “Almost contours”:** Suggestive contours are the set of points containing the minima of $[n \cdot v]$ in the direction of w as illustrated in Figure 2.6.

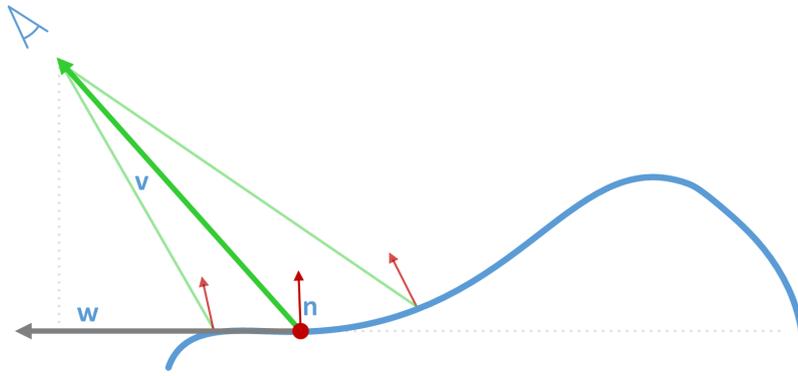


Figure 2.6: Illustration of definition II with suggestive contours being the points where $n \cdot v$ reaches a local minimum.

- Definition III - Zeros of radial curvature:** Suggestive contours are the set of points on the surface of an object at which its radial curvature $\kappa_r = 0$ and the directional derivative of κ_r in the direction of w is positive: $D_w \kappa_r > 0$, i.e., where the surface bends away from the camera. This situation is depicted in Figure 2.7.

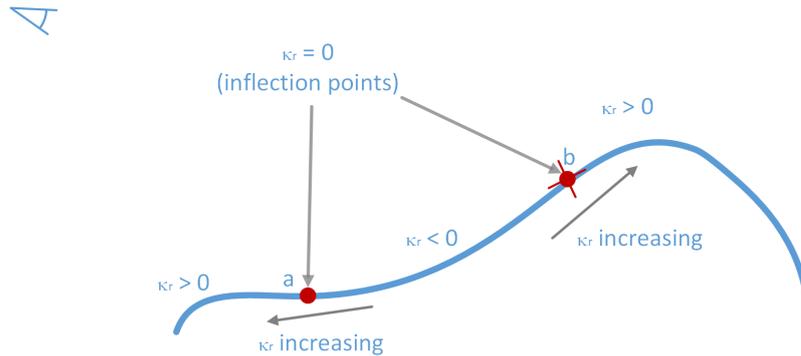


Figure 2.7: A situation with two inflection points satisfying the constraint $\kappa_r = 0$. In contrast to point b , point a also satisfies the second constraint $D_w \kappa_r > 0$.

With p being a point on a smooth surface, $n(p)$ being the unit surface normal at p , $v(p) = c - p$ being the view vector to the viewpoint c , and $w(p)$ being the projection of $v(p)$ onto the tangent plane at p as illustrated in Figure 2.8. $\kappa_r(p)$ is defined as the radial curvature of the radial curve created by intersecting the radial plane with the object’s surface as shown in Figure 2.8. The derivatives in definition **III** underline the fact that suggestive contours are second-order line features. In comparison to higher-order features like apparent ridges or ridges and valleys, suggestive contours are relatively unsusceptible to geometric noise [DFRS03].

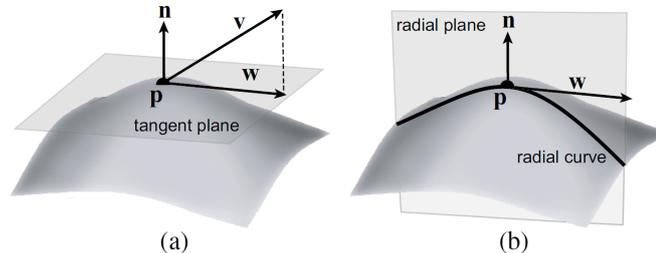


Figure 2.8: The left image (a) depicts the projection of v onto the tangent plane to obtain w . The right image (b) shows how p , n , and w form the radial plane which slices the surface along the radial curve with curvature $\kappa_r(p)$ [DFRS03].

2.3 Perception of Lines

When using lines as a method for visualization, it is usually assumed that the human visual system is able to reconstruct the 3D shape of a given object, but this assumption is only partially correct since there is an inherent ambiguity when projecting 3D curves onto 2D surfaces as shown in Figure 2.9. In this chapter, the most important aspects of this topic are presented to give an overview of how shape perception from lines is not

unambiguous.

Well-known examples like the Penrose triangle in Figure 2.10 and other actually possible structures show how difficult it can be to reconstruct 3D shape from simple line drawings. Impossible figures like the Penrose triangle with only locally possible structures also show that coherence is not a global phenomenon.

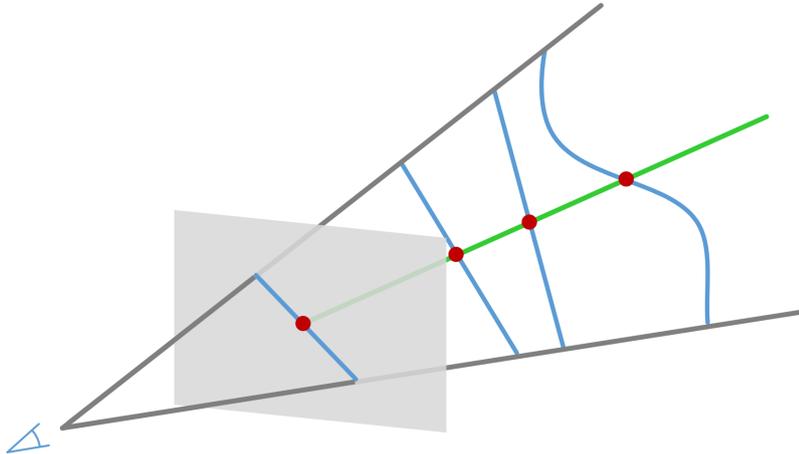


Figure 2.9: An example of ambiguity in lines. There are infinitely many possible lines - all mapping to the same projected line.

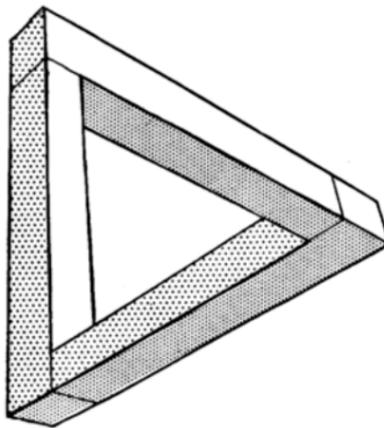


Figure 2.10: The Penrose triangle [PP58].

Besides local visual inference like in the Penrose triangle case, there appears to be also a global aspect of inference of 3D shape from line drawings as shown in Figure 2.11 [BT81]. The two objects differ only in the bottom line, yet the left object appears to be raised in the center, and the right object appears to have a flat top with a deformation along its length. It is still unclear whether this is caused by an internal integration of local information or by non-local interference by the human visual system [RCDF08].

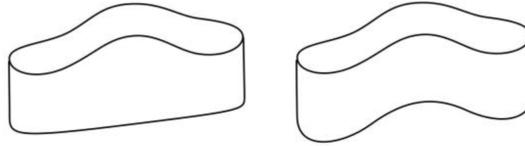


Figure 2.11: An example of how non-local inference changes the perception of 3D shape [BT81].

The process of forming a mental 3D model from line drawings and therefore understanding a scene on a higher level is usually based on two main concepts: recognizing what each line depicts qualitatively and inverting the geometry of line formation [Wal75], [Mal87]. In simple polyhedral scenes, the line segments can have one of three types along its entire length: convex, concave, or occluding edge. For smooth surfaces, the labels of line segments do not necessarily stay the same along the line but at least for convex and concave edges globally consistent line labels can be inferred from local junctions where lines touch.

The presented line drawing algorithms are well understood and used frequently in visualizations, but the exact biological process of how line drawings are understood by the human visual systems are still unknown. Even if the process is not entirely understood, with the help of user feedback is still possible to generate models of objects based on sketch-like user inputs as shown in Figure 2.12 [KC06].

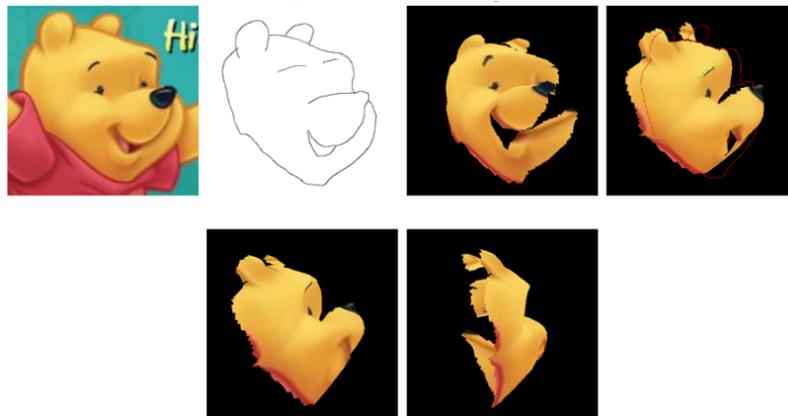


Figure 2.12: Examples of sketch-based modeling of a 3D character with texture-mapped renderings of the resulting mesh [KC06].

Research shows that the interpretation of line drawings on a global scale can at least be modeled and is used in algorithms for labeling of polyhedral scenes by Waltz [Wal75] and even for more general cases like smooth surfaces by Malik [Mal87], but these approaches only label lines with a type and cannot infer geometry [RCDF08].

Even if it is still unknown how humans process line drawings exactly, much is known

about what kind of information might be used. Depending on the type of lines, each line puts a constraint on the depicted shape, and sometimes the type of line can be inferred from the context within the drawing. For example, lines can mark a fixed location on a shape's surface such as sharp folds with creases, ridges and valleys, or texture features, as well as other surface markings with methods like hatching lines. They can also mark view-dependent locations such as contours, suggestive contours, and apparent ridges, and they can mark lighting-dependent locations such as isophotes or simple edges [RCDF08]. Although some level of ambiguity always remains, some interpretations of an object's shape are more reasonable than others, and the human visual system seems to be exceptionally good at finding the most plausible option [RCDF08].

2.4 Perception of 3D Graphics

The process of perceiving 3D objects presented on 2D displays consists of several stages as visualized in Figure 2.13. The first step in most rendering techniques is a 3D triangle mesh representing a 3D object in a virtual scene. This triangle mesh and the surrounding scene is transformed from world space into camera space and then projected onto a virtual image plane as a rendered image according to a rendering algorithm, e.g., the Blinn-Phong Shading model [Bli77] or suggestive contours [DFRS03]. The resulting picture on a 2D display is in so-called screen space.

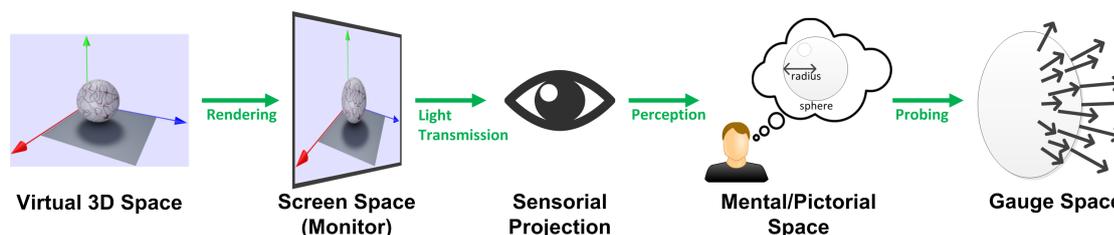


Figure 2.13: The visual perception pipeline from 3D triangle mesh to Gauge Space.

The display presenting the image emits light according to the pixel information contained in the image. This light is partially collected by the eye lenses and projected onto the retina. On the retina, so-called retinal processes occur, which are preattentive and eventually generate electrochemical signals to the brain via the optic nerve. The two optical nerves pass the visual signals to the primary visual cortex for higher level visual processing. At this stage, a mental image of the presented object is formed in so-called “mental space” or “pictorial space” by a combination of bottom-up and top-down mental processes.

In the conventional model of human perception, the senses provide data to the brain, which is then translated into the corresponding mental phenomena, e.g., pictures into a tree or soundwaves into a song. In contrast to this bottom-up process, studies show that a second so-called top-down process exists in parallel, suggesting that human perception

is also shaped by past experiences, the conceptual understanding of the world, and the context in which our senses perceive information [WKG10] [Koe29].

Without this second line of processing, an already existing network of distinct and interrelated ideas and concepts, as well as a general “understanding of the world”, the incoming sensory signal would be perceived as undifferentiated chaos rather than the world be actually apprehend. Because of this dual process, human visual perception is subjective and context-dependent creating practical implications, e.g., when designing visualization systems [STPV12]. This duality of perceptual processes was coined by Danko Nikolic under the term “ideasthesia” [Nik09].

The last stage of the perceptual pipeline pictured in Figure 2.13 is of importance for this thesis because it is the basis for quantifying visual perception. This so-called “gauge space” is an abstract space created by probing the pictorial space to deduce information about the internally created mental image and its general characteristics when changing independent variables such as the level of detail and the type of a presented object. This probing is realized by empirical experiments such as the gauge figure task as described in Section 2.6.

Similar to the ambiguity of lines, the depth information of 3D objects projected onto the 2D retinal surface is partially lost in the process of seeing. To some extent, the 3D information can be reconstructed by a combination of top-down and bottom-up perceptual processes with so-called “depth cues” as described in Section 2.5. Belhumeur et al. named a special form of this loss of structural information the “bas-relief ambiguity” [BKY97]. Their research discusses the case of orthographically rendered objects with Lambertian reflectance. In this special case, the shading and the shadowing on any object with surface $f(x, y)$ stay the same under any bas-relief transformation of the object, $f'(x, y) = \lambda f(x, y) + \mu x + \nu y$, with corresponding adjustment of the albedo. In Figure 2.14 this effect is visualized. Even small movements or changes in the point of view cannot resolve the ambiguity in determining the scaling or flattening in λ . If specular highlights are introduced or the rendering is changed to perspective, more structural information is conveyed, and the effect of bas-relief ambiguity vanishes.

In line drawings, the ambiguity of shape has a stronger effect because line drawings generally do not contain visual nuances such as changing albedo, highlights or lighting that provide further information about an object’s 3D structure. Due to this fact, the change of 3D shape ambiguity in relation to changing the level of detail and changing type object is one of the analyzed questions of this thesis.

This thesis focuses on visual perception and with around 8.75 megabits per second humans perceive by far the most information by sight [KMS⁺06], but perception is not only limited to visual perception - it extends to all our senses. On an abstract level perception is defined as organizing, recognizing and interpreting sensory information of any kind to form a mental image and to make sense of the world around us. When modeling the perception of physical stimuli in the context of all possible human sensations an underlying mathematical model arises referred to as “Steven’s power law” [Mac63]

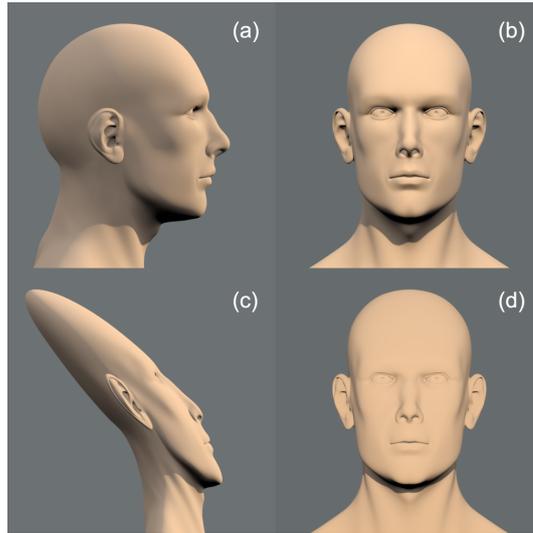


Figure 2.14: Rendering with the bas-relief ambiguity effect. (a) and (b) show the normal undistorted case. If the head is deformed under a bas-relief transformation (c), it still appears to have the same geometry (d). Only the unadjusted albedo and changing lighting conditions indicate a visual change between (b) and (c).

[Sta78]. This law can be characterized as a generalization of the Weber-Fechner law with the advantage of describing a wider range of sensations. It describes the exponential relationship between a physical stimulus and the perceived strength of this stimulus. Equation 2.1 shows this model in its general form.

$$\varphi(I) = kI^a \quad (2.1)$$

I describes the magnitude of the physical stimulus, $\varphi(I)$ is the perceived magnitude created by the physical stimulus, a is an exponent characteristic to the type of physical stimulus, and k is a proportionality constant depending on the used units. For example, the perception of loudness can be modeled by $a = 0.67$ for a sound pressure of a 3000 Hz tone, the perception of brightness for point light sources has $a = 0.5$, taste of salt has $a = 1.3$, thermal pain and cold have $a = 1$, and the perception of time duration with white-noise stimuli can be modeled with $a = 1.1$. The perception of shape might also fall into this group of sensations modeled by Stevens's power law. Therefore, one of the questions this thesis tries to answer is whether the perception of 3D shape from suggestive contours changes according to this law when changing the level of detail.

2.5 Depth Perception and Depth Cues

Since this thesis investigates the perception of shape, which is innately connected to depth perception, this section gives a high-level overview of how human depth perception

is processed.

The human visual system uses several attributes of visual stimuli called depth cues to reconstruct the three-dimensional nature of an object in what is called depth perception. These depth cues can be divided into monocular cues, i.e., using the information available when viewing an object with only one eye, and binocular cues, i.e., using depth information gained by using two eyes. Furthermore, depth cues can be classified as physiological and psychological which are equivalent with binocular and monocular depth cues respectively [How12] [HR12] [Oko12].

Prominent monocular cues include motion parallax and depth from motion, linear perspective, the relative size of objects, aerial perspective, i.e., the phenomenon of objects displaying a bluish distance fog with increased distance, overlapping, texture gradients as well as shades and shadows. Not all monocular cues are readily available in line drawings and still images, e.g., texture gradients, shadows, and motion parallax. These drawbacks are the reason why line drawings are often considered to be less expressive of 3D shape than shaded images, but modern line drawing techniques like suggestive contours or hatching try to compensate for this lack of information by hinting the missing visual information explicitly. Despite this lack of some depth cues, line drawings can convey 3D shape information relatively well as demonstrated by artists through history.

In line drawings and conventional 2D rendering in general, binocular cues like accommodation, i.e., the muscle tension when changing the focal length of the lens of the eye, or convergence, i.e., the difference of direction of slightly inwards pointing eyes when objects are near, are not of importance. This is owed to the fact that the rendered images are presented on a flat 2D display, which prevents the collection of additional information using a second point of view. Stereoscopy rendering and displaying methods can provide additional binocular depth cues but are not in scope of this thesis.

2.6 Psychophysical Measuring Techniques

Since the visual perception of objects and their characteristics is an intrinsically internal process within the human brain, methods for probing this inner process have been developed by researchers and are used in this thesis for probing human shape perception. The first psychophysical experiments were conducted in the 19th century and investigated stereoscopic vision by presenting points of light in otherwise dark rooms [Tod04]. This early research found that human perception can be distorted to perceive curved lines when actually presented with straight ones.

The range of psychophysical methods has expanded and was refined since then, but in general, all methods have in common that the local 3D structure at different sampling points on an object's surface has to be estimated by a group of participants to quantify their visual perception of the object of interest. Todd and Koenderink et al. reviewed the three most commonly used of the psychophysical measuring techniques [KvDKT01] [Tod04].

The first method for the measurement of visual perception is called “relative depth probe task” and it uses shaded objects for the measurement of depth perception [Tod04]. In this technique, two surface points are presented as differently colored dots on the object’s surface. The participants have to decide which of the two points is perceived as closer in depth by pressing a button with the same color as the closer dot. This technique for depth probing is used by Solteszova et al. to quantify and evaluate the quality of a novel shadowing technique in shaded visualizations [SPV11]. In Figure 2.15 an example scene of an experiment for this technique with a stimulus is presented.

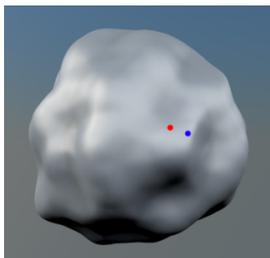


Figure 2.15: Example stimulus for the relative depth probe task. The blue dot should be perceived as nearer to the viewer than the red one.

The second psychophysical measuring technique is called “depth-profile adjustment mask” where the user is presented with a shaded object and an overlay of a linear arrangement of dots [Tod04]. The arrangement of dots is also displayed on a second display with a white background. Moving the dots in the second display with drag and drop rearranges the dots in the first display accordingly. The user’s task is to rearrange the dots to the perceived height profile of the presented object surface. Figure 2.16 shows an example of this measuring technique.

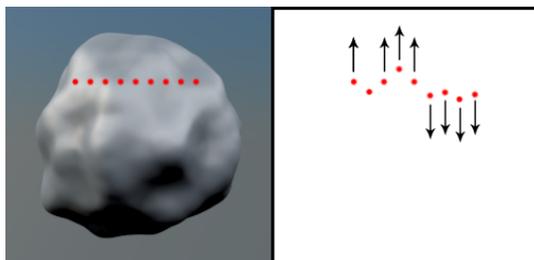


Figure 2.16: An example of the depth-profile adjustment mask technique.

The third most commonly used psychophysical measuring technique is called the “gauge figure task”, which is also used in the context of this thesis for data collection. The gauge figure task was developed by Koenderink et al. and has shown to be an intuitive, fast and easy method for probing a user’s perception of an object’s surface normals. The gauge figure is a projected 3D circle indicating the amount of distortion with a stick perpendicular to the plane defined by the circle. In the literature, this gauge configuration is also referred to as an orthographically projected Tissot’s indicatrix.

2. BACKGROUND AND PREVIOUS WORK: WHAT LINES CONVEY AND WHAT HUMANS UNDERSTAND

Due to the orthographical projection, the circle only appears as a circle when the gauge figure is aligned with a surface perpendicular to the user’s viewing direction. Otherwise, i.e., when the surface is slanted away from the viewing direction, it is seen as an ellipse. In a perfectly adjusted gauge figure position, the stick of the figure is aligned with the surface normal. The length of the stick is equal to the radius of the 3D circle. In Figure 2.17 a well and a poorly positioned gauge figure example are presented.

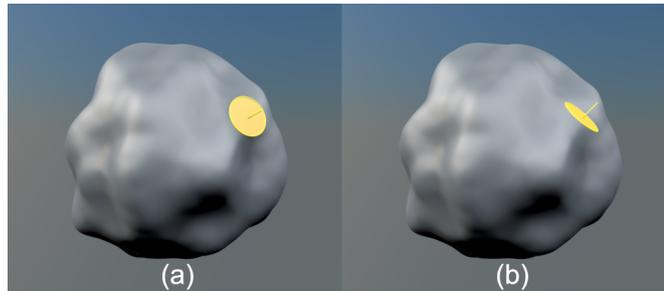


Figure 2.17: Examples of a perfectly aligned gauge figure (a) and a badly adjusted one (b).

To quantify the surface normal of the local tangent space in pictorial space, i.e., the depth gradient ∇Z , the surface normal’s tilt and slant are used. The slant describes the angle between the surface normal the user’s view vector whereas the tilt is the azimuth direction of the normal in eye space, i.e., slant describes the orientation in depth and tilt the orientation in the image space like compass directions. The tilt τ can range from 0 to 2π or 360 degrees and the slant from 0 to $\pi/2$ or 90 degrees.

These ranges of slant and tilt create a hemisphere with its pole at the viewing axis as illustrated in Figure 2.18. The pole has a slant of $\sigma = 0$ and an arbitrary tilt τ [Ste83a] [Ste83b], because the angular deviation from the viewing axis is 0 and any rotation of a view-axis aligned normal around this axis results in an identical normal. Since the length of the gauge figure stick is equal to the circle’s radius, the gauge figure represents a slant of 45 degrees if and only if the indicatrix-circle and the tip of the stick touch [KvDK92].

In the original gauge figure task experiment Koenderink et al. designed a four-stage process as illustrated in Figure 2.19 [Wij12]. First, a suitable image is selected. Then, the outline of the presented object is selected either by hand or by an edge detection algorithm. The sample points are defined by a triangle grid overlay within the previously selected outline. The denser the grid, the more sample points are defined. In the final gauge figure task, the grid overlay is not shown, but the samples are taken at the mid-points of each of the triangles defined by the grid, i.e., the grid serves as a sample point generator for the x and y coordinated in image space.

After defining the sample coordinates, the user study participants have to perform the gauge figure task by adjusting the gauge figure at the sample coordinates to sample their perception of surface normals and therefore the perceived shape of the presented object. To avoid unforeseen influences of varying viewpoints, the participants’ heads rest on a

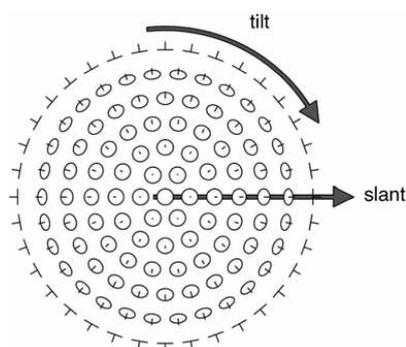


Figure 2.18: Examples of possible slant and tilt combinations of a normal at different points on the hemisphere. Slant and tilt are both continuous [vK04].

chin rest. This assures equal distance from the screen, a centered eye position, and an equal configuration among all participants to collect comparable data points.

The final stage in the original gauge figure task is to reconstruct the 3D surface of the 2D object in regards to the users' estimates. For this, the triangles of the triangle mesh are rearranged to match the estimated surface normals [Wij12]. The resulting mesh represents the perceived 3D shape from empirically sampled 2D user estimates.

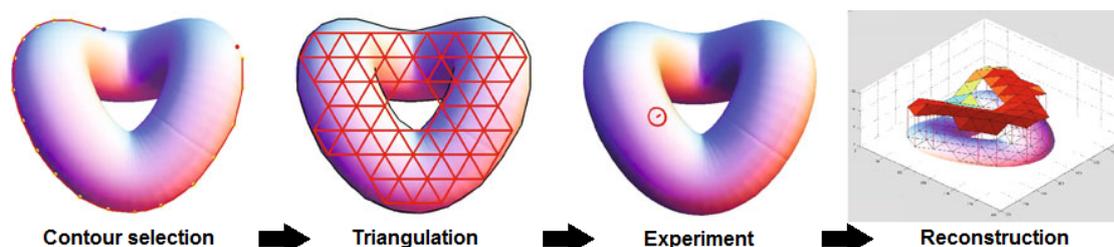


Figure 2.19: Illustration of the original gauge figure task design with its four stages [Wij12].

Originally designed to probe only the surface perception of images and paintings, virtually constructed objects and their renderings allow the comparison of empirical estimates with the ground truth surface normals.

According to Koenderink et al. the gauge figure task involves no overt reasoning and is therefore immediate because the participants do not form complex image abstractions from their perception. In comparison to the other two psychophysical probing techniques, the gauge figure task is the easiest, the most natural to use and, by far the most reliable [KvDKT01] [Wij12]. These advantages are the reason for using the gauge figure task for probing human shape perception in this thesis.

State of the Art

Continuing from Chapter 2 where line types, their perception by humans, and psychophysical measuring techniques are introduced, this chapter gives an overview of state-of-the-art research in line drawings, and previous results of empirical psychophysical experiments are presented.

The first experiments in probing human visual perception were conducted in the 19th century, and since then a vast body of knowledge has developed in this field. As the goal of this thesis is to produce optimal line drawings, a way to assess optimality is needed. For this, best practices from psychophysics in shape perception and previous results are utilized.

The most prominent result of psychophysical probing experiments of human visual perception is the recurring underestimation of surface slants in the gauge figure task. This means that the stimulus objects seem to appear flatter to the viewer than they actually are, even when the real shape of an object is well known.

Zimmerman et al. are one of the teams investigating this phenomenon [ZLC95]. In their experiment, the authors measure the accuracy via slant estimates of 3D test planes by projecting them perspectively onto a 2D screen. The perceived slant is calculated from estimates of the relative length of two orthogonal lines on the plane's surface, and the results show an average underestimated slant of 7 degrees. They conclude that depth estimates are usually not accurate for disconnected surfaces and that there are two internal representations of 3D space, one of which was later named "pictorial space".

Other results of experiments performed by Mamassian and Kersten estimate the underestimation of perceived slant to be at around 30 degrees [MK96]. In their experiments, viewers estimated displayed slants of 60 degrees to be 30 degrees in their perception. Slants above 20 degrees are generally underestimated while slants below this threshold are overestimated. The authors also found that locally egg-shaped surfaces result in less biased surface orientation estimates than locally saddle or cylindrical surfaces and that

in this specific experiment the variance for perceived tilt was higher than for perceived slant, which contradicts previous results of Koenderink et al. [KvDK92].

Another result of Mamassian and Kersten is that objects rendered with Lambertian shading rather than simple silhouettes are estimated with higher accuracy and less bias. They conclude that depth overestimation or a flattening of objects may be the cause of the persistent underestimation of slants in experiments. De Haan et al. [DHEN95] and Solteszova et al. [STPV12] also confirmed these previous findings of systematic underestimation of slant for both abstract and well-known shaded surfaces respectively. Solteszova et al. determined the crossing point of under- and overestimation between 15 and 25 degrees, which is consistent with the findings of Mamassian and Kersten.

Similarly, Bernhard et al. [BWP⁺16] [Pla15] confirmed the general underestimation of slants, and for spherical objects even the wave-like error with a crossing point at around 40 degrees when shaded surfaces are presented on a standard 2D display. Furthermore, the underestimation may be reduced significantly by introducing stereoscopic 3D viewing conditions. When assuming that under close to perfect stereoscopic viewing conditions only the psychophysical measuring technique has a significant influence on the error produced, the results by Bernhard et al. suggest that at least part of the underestimation of slants can be attributed to the method of the gauge figure task itself. The authors conclude that under close to perfect gauge figure task conditions, i.e., a meticulously constrained experimental setup with well-known objects under ideal 3D stereoscopic viewing conditions, the minimal slant error for the gauge figure task is bounded by $\pm 5^\circ$.

Other studies do not explicitly measure perceptual accuracy of renderings styles but rather how much cognitive effort is required to recognize a given object and which areas capture a viewer's interest the most [SD04], [CDF⁺06]. Similarly, researchers evaluated the effectiveness of rendering techniques by measuring how much time is needed to recognize facial expressions [WBC⁺07] and facial caricatures [GRG04]. Recognition speed usually increases when adding stylization techniques. In order of effectiveness shading, contours and texture are the all valid methods to improve a viewer's performance in visual recognition tasks, at least in dynamic environments [WFGS07].

A study by Cole et al. uses a public dataset of artists' drawings of different shapes along with their registered 3D models and viewpoints to evaluate the effectiveness of six different line drawing algorithms to depict shape [CSD⁺09]. Similar to previous experiments, the authors used Koenderink's gauge figure task to probe the participants' perception of shape in comparison to the ground truth surface normals of stimulus objects. For this, they collected a set of 275,000 gauge figure task measurements of 70 different images estimated by 500 participants via Amazon Mechanical Turk on which statistical and anecdotal data was extracted and presented in their results.

Cole et al. conclude that in general people interpret line drawings similarly to shaded images and that line drawings can be as effective as photorealistic images at conveying shape although some line drawing algorithms seem to be more effective than others. In addition, line drawings can also match an artist's line drawings in the effectiveness of

depicting shape. The errors in shape depiction are usually attributed to unique properties of the lines used and are often localized.

The parameters of the different rendering styles to create the rendered stimulus images evaluated by Cole et al. are chosen to produce clean, smooth and continuous lines. Objects with rough surfaces are smoothed to facilitate smoother results. In total, a set of 12 stimulus objects are used to probe the participants' perception of shape; an example of such a model is given in Figure 3.1.

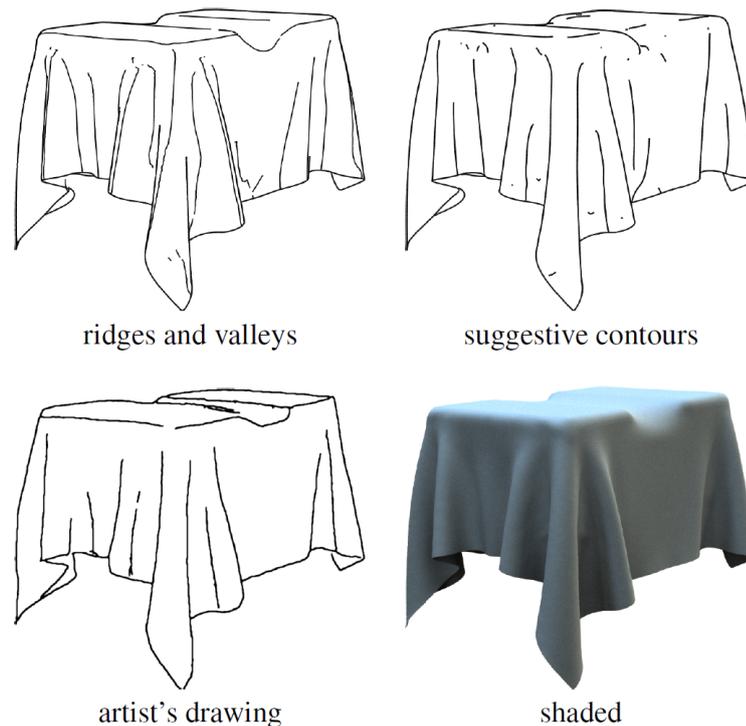


Figure 3.1: Four examples of a table cloth model in different rendering styles used in the experiments conducted by Cole et al. [CSD⁺09].

When examining the results in detail, the authors found that shape from simple contours is estimated considerably worse than other more complex line rendering methods like suggestive contours or apparent ridges. For objects with more complex surfaces like vertebra, the estimates are worse - 35 – 40 degrees error on average for the best models - than for simple objects like cubes with holes in it with an average error 15 degrees. This result confirms the intuition that certain shapes are generally hard or impossible to depict effectively with line renderings methods.

The exact parameters of the different line rendering styles used in the study are unknown. Since different parameters lead to visually different line drawings created by the algorithms, it is valid to assume that they are perceived differently even for a single method of line drawing. Hence, the idea of this thesis is to choose one complex line drawing method -

3. STATE OF THE ART

in this case suggestive contours - and to investigate the change in perceptual error of shape on different levels of detail for different stimulus object types. By evaluating the resulting estimates and use them to optimize suggestive contours an improved shape perception for this method of line drawing may be possible.

Quality and Expressiveness of Lines

This thesis is comprised of both the collection and analysis of empirical user estimated to answer the research questions of interest for this thesis as outlined in this chapter as well as the usage of these results to optimize the suggestive contours line drawing method as discussed in Chapter 7. In the first of the following two subsections, the general technical pipeline from 3D meshes to empirical sample points and optimized line drawings is presented to give an overview of the entire process on a conceptual level. The second subsection presents and discusses the set of defined research questions of interest for this thesis as well as the subsequent analysis process to answer them.

4.1 Technical Pipeline: From Mesh to Empirical Sample Points

The main goal of this thesis is to analyze how the perception of 3D shape varies with a single visually monotonic parameter for the suggestive contours line drawing method and with changing the type of stimulus object used. In addition, the goal is to synthesize perfect lines based on the gained understanding during the analysis of how well they convey shape. For this, the following steps in the technical pipeline are taken and visualized in Figure 4.1.

First, the stimulus object types are chosen, and for each, a visually monotonic sequence from simple silhouettes to highly detailed suggestive contours line drawing is rendered with a 3D modeling and rendering software. These renderings are used to probe the participants' shape perception of the suggestive contours rendering technique. As a second step in the generation of images, the normal maps of the stimulus objects are rendered.

After stimulus image generation, the necessary tilt and slant combinations for proper sampling of the virtual hemisphere of all tilt-slant-combinations are defined. These values are then used to find corresponding ground truth normals in the normal maps. In this process of finding correct ground truth normals a sanity check for the found normals is conducted: if they lie on a line in the most detailed version of a line drawing rendering for a specific object, the search for a proper ground truth normal is continued. The found normals are then exported into a text file for later usage in the gauge figure task. At this stage all files for starting the gauge figure task with volunteering participants are present.

To conduct the gauge figure task, a framework is implemented for the subsequent execution of the gauge figure task on the rendered objects with several levels of detail and at the predefined ground truth normal positions. After all participants have estimated the objects on all levels of detail, the resulting data is aggregated, anonymized, and cleaned in the statistical programming language *R*. Then, the data points are analyzed, model fitting is conducted, and the results are visualized to answer the predefined research questions.

In a subsequent post-processing stage, the stimulus object renderings are optimized according to the analysis' results to create line drawings containing maximal perceived 3D shape information while being low in the amount of ink.

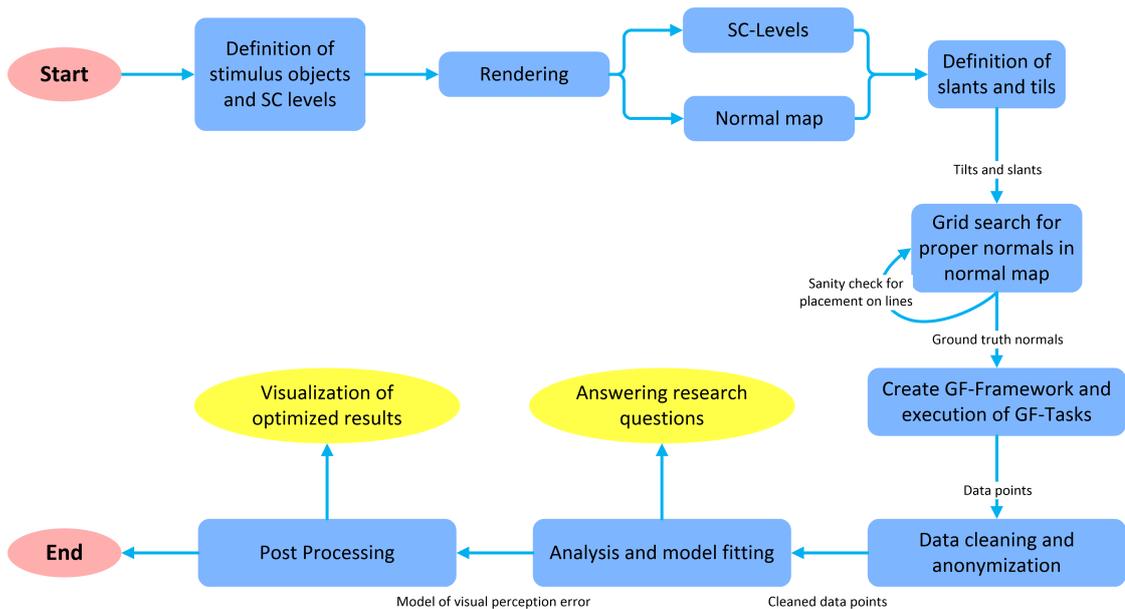


Figure 4.1: The technical pipeline and its intermediate results.

4.2 Research Questions and Analysis Process

There are several key questions to be answered by analyzing the collected empirical user study data, these are:

1. **Perceptual shape ambiguity:** How large is the perceptual 3D shape ambiguity for suggestive contours renderings and how does it change with the level of detail and with the type of object shown? This metric is important for removing inaccurate data points where it can be assumed that the estimated surface normal was placed at random or the perception was entirely off. Proper error calculations would be impaired by including such points. In addition, this metric reveals how strongly the general top-down 3D perception of shape is affected by the level of detail and the type of object. Research shows that in comparison to slant, tilt is usually estimated relatively well [NW07] [MK96]. Therefore strong deviations, e.g., in the opposite direction, are assumed to be good indicators of shape ambiguity. To quantify the amount of shape ambiguity the number of estimated tilts, which deviate more than 90° from the ground truth tilt, is calculated and analyzed.
2. **Error of 3D shape perception:** How is the error of the participants' estimates distributed and how does it change in regards to the level of detail and type of object stimulus? These statistics are essential to get a better understanding of the type of data and how it is affected by the independent variables of object type and level of detail for the suggestive contours rendering technique. Furthermore, if the statistics match the results of previous gauge figure task experiments, it is an indication that the collected data has no overt flaws. Since slant is regarded as the key indicator for the error of 3D shape perception, it is used to calculate the error statistics [LC95] [BWP⁺16].
3. **Steven's Power Law:** Does the perception of 3D shape from suggestive contours renderings follow Steven's power law and how does it change with the level of detail? If it follows Steven's power law, the perception of shape follows the same principle as for example the perception of light intensity. To answer this research question, model fitting is performed on the collected data point estimates, and the parameters of Steven's power law are approximated.
4. **Size of local estimation area:** Which size of the local surface area is used by the participants to estimate surface normals? It is hypothesized that the users use a specific area around a sampling point in the bottom-up part of the estimation of the local 3D structure. If this is the case, the data may show how big this local area is and indicate on how much local information is considered ideal by the human visual cortex. To answer this question, the average ground truth slants of differently sized local neighborhoods are compared to the participant's estimates. First, it is analyzed if there is a preferred neighborhood size and if that is the case its size is determined. Then, it is analyzed how this optimal neighborhood size changes depending on the level of detail and the type of object used.
5. **Optimal amount of local ink:** What is the optimal threshold for the local amount of ink in regards to the type of object and 3D shape perception error over all levels of detail in suggestive contours? This threshold is key information for optimizing suggestive contours renderings in terms of the minimum amount of ink

while retaining maximum shape understanding. To answer this question, the slant estimation error is analyzed in regards to the percentage of local ink in differently sized local neighborhoods for different stimuli and different levels of detail. The minimal error defines the threshold for how low the amount of ink may be while still retaining good perception of 3D shape.

Technical Implementation

Following the conceptual explanation of the technical pipeline, the gauge figure task framework and the subsequent analysis phase in Chapter 4, this chapter presents the implementation steps in detail to allow for reproducibility of this thesis' results. For this, the different methods for implementing suggestive contours including their advantages and disadvantages are explained as they differ in quality of the resulting line drawings. In addition, the details for the line generator used for this thesis as well as the technical aspects of the gauge figure task framework are presented.

5.1 Approaches to Implementing Contours and Suggestive Contours

Line drawing algorithms can be categorized by their implementation either as image space method or as object space method. For suggestive contours as proposed by DeCarlo et al. it is possible to realize both types of implementation [RCDF08] [DFRS03] [Bae10]. This section gives an overview of three possible approaches to implementing suggestive contours to show how they differ and which one to choose for a user study like in this thesis.

5.1.1 Suggestive Contours in Image Space

For regular contour lines from images two simple methods can be implemented:

1. **Diffuse renderings:** For this approach, a diffuse perspective rendering with a point light source at the camera position is created resulting in a scalar field corresponding to $n \cdot v$ at every pixel according to Lambert's cosine law. On these values, a threshold can be applied to obtain contour lines. Problematic with this technique is the non-existent control of line widths.

2. **Sobel filter on depth maps:** By applying edge detection with a Sobel filter on a rendered depth map it is possible to obtain lines with equal width [SF68].

Similar to the first approach from above, suggestive contours can be extracted from diffuse-shaded renderings by using a radial filter as proposed by DeCarlo et al. [DFRS03]. The idea behind this approach lies in the fact that for suggestive contours valleys need to be detected. These valleys present themselves as a set of dark pixels intersecting a neighborhood of mostly bright pixels. Suggestive contours pixels are pixels where only a certain percentage of neighborhood pixels with smaller intensity exist and where the difference between the maximum value in the neighborhood and the current pixel is smaller than a predefined threshold. By enlarging the radius, more suggestive contours are found while processing time increases proportionally. This method can easily be implemented in a vertex shader on a GPU to allow for larger radii [Bae10].

5.1.2 Suggestive Contours in Object Space

Object space rendering methods are characterized by working directly on the original mesh of an object. In case of contour lines, this means that individual line segments are calculated from the original vertex coordinates and this list of segments can then be rendered directly. This approach has the advantage of full control over the width and general stylization of the lines as shown in Figure 5.1 as well as allowing for analysis on the lines' position, length, and interconnectivity. The disadvantage is that object space methods are more complicated and in general slower than their image space counterparts.

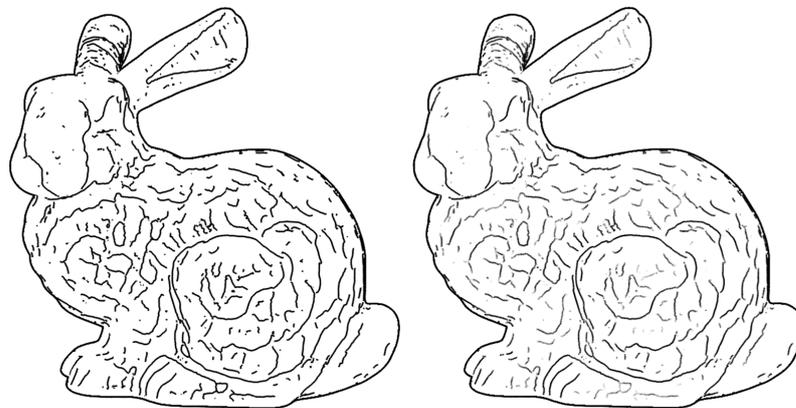


Figure 5.1: On the left side, the Stanford bunny is rendered with a CPU-based suggestive contours method with equal width lines. On the right, the lines are also stylized with a fading effect. Both images are rendered with the “rtsc” suggestive contours viewer by Szymon Rusinkiewicz and Doug DeCarlo.

Two well-known approaches for object space CPU-based contours are:

- **$N \cdot V$ -Algorithm:** in this approach, all faces are visited, and zero crossings of the $n \cdot v$ dot product are searched for. When a zero crossing is identified, a line is drawn over the face using linear interpolation as described by DeCarlo et al. [DFRS03].
- **Apple’s algorithm:** To improve the processing time, one can simply draw the lines between front and back-facing polygons instead of across faces. This approach might lead to small errors like contour loops, but for high polygon meshes, the error diminishes [App67].

In accord with DeCarlo’s definitions in Subsection 2.2.1, suggestive contours require the evaluation of radial curvature κ_r and the directional derivative $D_w \kappa_r$ at every vertex. Since mesh vertices are discrete, several approximation algorithms exist to estimate the surface curvature as described by Rusinkiewicz [Rus04] and Do Carmo [DC16]. For example, the Euler formula may be used: $\kappa_r(p) = \kappa_1(p) \cos^2 \phi + \kappa_2(p) \sin^2 \phi$, with κ_1 and κ_2 being the principal curvatures at point p , and ϕ being the angle between the first principal direction and the projected view vector on the radial plane at p .

The directional derivative in the direction of the projected view vector v on the radial plane at p , i.e., w , can be computed by the formula derived by Rusinkiewicz:

$$D_w \kappa_r = \frac{D_{e_1} \kappa_1 u^3 + 3D_{e_2} \kappa_1 u^2 v + 3D_{e_1} \kappa_2 u v^2 + D_{e_2} \kappa_2 v^3}{\|w\|^2} + 2\kappa_1 \kappa_2 \|w\| \left(1 - \left(\frac{n \cdot v}{\|v\|}\right)^2\right)^{-1} \quad (5.1)$$

By visiting every face and checking for zero crossings of radial curvature, none, one or two suggestive contour lines may be found. Methods for speeding up the calculation mainly focus on not visiting every face and performing fewer tests on each face, e.g., backface culling, using a face normal cache in a pre-pass, or using Gaussian curvature for the suggestive contours rendering pass. Another approach for improving speed is by using a stochastic algorithm where the requirement of finding all contours is loosened to only test a subset as proposed by Markosian et al. [MKG⁺97].

5.1.3 Suggestive Contours in Object Space - Parallel Approach

The idea of developing and using GPUs in the field of computer graphics lies in the nature of graphical problems being usually parallelizable. In the case of suggestive contours the vertex and matrix calculations, which have to be done for every vertex of the mesh in every frame, can be easily be parallelized [Bae10].

Up until 2010 GPUs have only been used partly to their full potential in suggestive contours renderings. GPUs were used in a Texture Mapping variant of suggestive contours rendering where a texture map containing a straight line was indexed by the radial curvature on the x-axis and its derivative on the y-axis. To achieve equal-width lines at different zooming levels, mipmaps were used. The disadvantages of this technique

are that texture map interpolations introduce line artifacts, and the curvature and the derivative have to be computed on the CPU, slowing down the overall rendering speed.

In 2010 Jeroen Baert fully ported the suggestive contours technique to a GPU-Vertex-Shader solving the problem of partly processing the graphics on the CPU [Bae10]. For contour lines, the dot product $n \cdot v$ is calculated and compared to a threshold resulting in thick contours at areas with low curvature. To limit this effect, the value can be divided by the curvature:

$$\frac{(n \cdot v)^2}{\kappa_r} < t_C \quad (5.2)$$

For suggestive contours a similar limit can be formulated using the derivative of radial curvature $D_w \kappa_r$ as the measure of curvature change:

$$\frac{|\kappa_r|}{D_w \kappa_r} < t_{SC} \quad (5.3)$$

This solution enables the usage of object space methods for defining suggestive contours on a per-pixel level, speeding up the processing significantly.

5.1.4 Comparison of the Three Approaches

This section contains a comparison of the three presented approaches for rendering suggestive contours based on three implemented and evaluated prototypes created in course of this thesis.

In general, image-based methods are the simplest to implement. The input is a diffusely shaded image and applying the mentioned radial filter results in a suggestive contours image of the scene. The bigger the filter radius, i.e., the more detailed the suggestive contours rendering, the longer the processing time. The lines of the resulting images in this approach have an uneven width due to being not controllable, and the images contain a rather high amount of pixelated artifacts as shown in Figure 5.2. Even with a post-processing pass combining median filters for salt and pepper artifacts and smart eroding filters on for controlling the line width, the resulting image quality is rather poor.

The object space approach using CPUs creates the highest quality results of the three approaches due to the explicit extraction of a set of lines from the mesh. By rendering these lines one by one, the line width and the general stylization can be controlled easily including fading effects. The disadvantage of this method is the slow processing speed due to CPU-based computation, but with today’s hardware and simple scenes real-time rendering of suggestive contours is possible as demonstrated in the reference implementation of suggestive contours by Szymon Rusinkiewicz and Doug DeCarlo called “rtsc”.

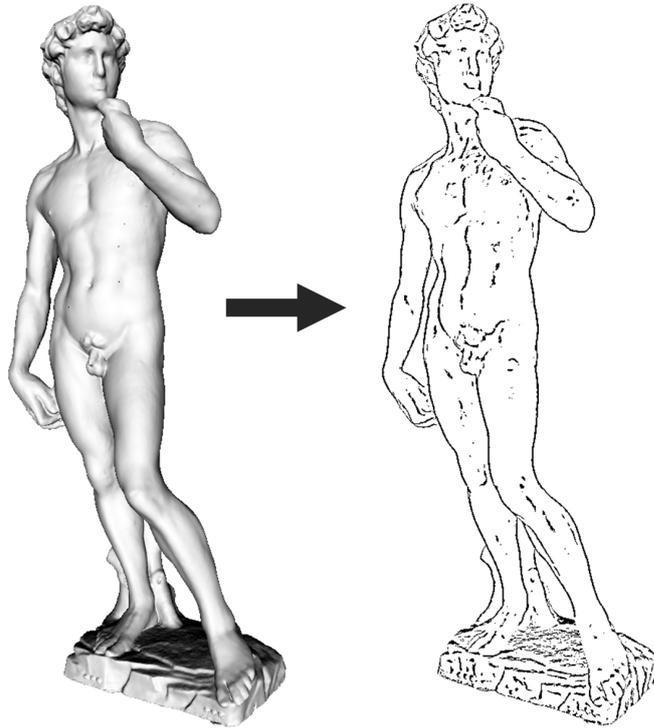


Figure 5.2: On the left, the original diffuse-shaded object is shown, on the right the resulting suggestive contours rendering after applying a radial filter in image space.

The GPU-based object space approach proposed by Jeroen Baert was also implemented and evaluated for this thesis. After porting the shader code from a fixed function pipeline to modern fully shader-based OpenGL 3.3 core profile, the increase in speed could be confirmed. For the objects used for this evaluation, the proposed dynamic threshold did not adequately regulate the uneven line width problem. Due to being a method based on thresholding a function, in this case the curvature, artifacts similar to the image space method can appear depending on the object’s geometry, although to a lesser extent.

Since the experiments in the user study for this thesis do not include real-time scenes and a high-quality rendering is assumed to be of high importance, a CPU-based approach is being utilized for generating the suggestive contours stimulus images. For this, the open-source software Blender and the plugin Freestyle are used. According to Maxime Curioni and Tamito Kajiyama, the suggestive contours renderings generated by Freestyle are comparable to the results created by the reference implementation “rtsc” [CK17]. In addition, Blender is able to export normal maps for the ground truth normals, and as an open-source tool, it is providing easy reproducibility and extendability of the results of this thesis for future researchers.

The Freestyle suggestive contours algorithm uses a so-called “sphere radius” parameter, which affects the calculation of curvatures and in extension the degree of detail for the

suggestive contours similar to the regulation of the threshold in the GPU-based approach proposed by Jeroen Baert. The smaller the sphere radius α , the finer the sampling of mesh surface and therefore the more detailed the extracted curvature is. With sufficient geometric detail in the mesh and a low α , detailed curvature and subsequently more detailed suggestive contours can be generated. If α is too low or the mesh is too rough, the generated suggestive contours contain more noise. In the user study, the used values of α are 50, 20, and 7 creating a visually monotonic sequence as desired by the thesis' assumptions. To achieve lines with equal width, the parameter "line thickness" is set to "absolute" and a value of $1.000px$. To limit the line length to a visually controlled amount the parameter "Min 2D length" is set to 7.0, and the parameter "Kr Derivative Epsilon" is set to 0.0 to achieve suggestive contour lines as described by DeCarlo. This parameter setup is fixed for all suggestive contours stimulus renderings and only the parameter α is changed for creating visually monotonic renderings per object.

5.2 Gauge Figure Task Framework

To perform the user study including the gauge figure task as shown on a conceptual level in Figure 4.1, a framework was implemented in the course of this thesis. The framework is implemented in C++ with OpenGL 3.3 core profile for the 3D rendering of the gauge figure and OpenCV for the image processing components. The core profile is used instead of vendor-specific extensions to allow the code to run GPU-vendor-independent. GLM is used for all mathematical calculations including the computation of transformation matrices in the rendering pipeline as well as transformations for proper tilt and slant display.

The framework serves two main purposes:

1. Extract a suitable set of normals to be estimated in the user study
2. Guide the participants through the user study while saving the users' estimates

Part one is achieved by importing pre-rendered normal maps of the stimulus objects and extracting four sets of normals - one for each stimulus object - as described in Section 6.2. The resulting sets of ground truth normals and their positions for each stimulus object are then saved into a text file.

Part two of the framework handles the user study where the participants are guided through the procedure and their estimates of surface normals in the gauge figure task are saved for later analysis. In this part, a conventional rendering pipeline in OpenGL 3.3 core profile with vertex and fragment shader shaders is set up. After showing an initial explanatory screen for the user study, the framework displays the suggestive contours stimulus objects and the gauge figure at the pre-extracted ground truth normal positions - one after another. The participants use the mouse to orient the gauge figure and press the space-key when it is properly aligned according to their perception of the scene.

Properly aligned in this context means that in the participants' perception the surface normal and gauge figure stick are aligned, or the local surface area and the gauge figure disk superimpose.

Internally the framework saves the following parameters per estimate:

1. Position of the ground truth normal in x and y in image space
2. Level of detail (1 – 5)
3. Type of object displayed (one of four types: a humanoid armadillo, a human brain, the Stanford bunny or an heptoroid)
4. Ground truth normal as tilt and slant
5. Estimate of the normal as tilt and slant
6. Percentage of ink in a local circular neighborhood around the normal's position for 41 different sizes of the neighborhood
7. Average ground truth tilt in a local circular neighborhood around the normal's position for 41 different sizes of the neighborhood
8. Average ground truth slant in a local circular neighborhood around the normal's position for 41 different sizes of the neighborhood

After completing the estimation of all 960 sample points, the framework closes the application window, and the participant has completed his or her part in the user study. Following the completion of the gauge figure tasks, the results of all participants are pooled together. In addition, an anonymous "user" variable for differentiating the participants, a "tilt group" variable for every one of the eight sectors of a circle, a "slant group" variable for every one of the six slant levels, and the genders of the participants are added. The resulting dataset contains 135 parameters per estimate and 17,280 estimates in total - 960 for each of the 18 participants.

Shape Perception User Study

To achieve the goal of identifying how the change in detail for the suggestive contours rendering technique and the type of object presented change the perception of 3D shape, a user study based on the gauge figure task was conducted. In this chapter, the used stimulus objects, the sampling procedure, and the experimental setup are described in detail.

6.1 Stimuli

For the user study, two independent variables are considered: the level of detail for the suggestive contours method and the type of object presented. The four used objects and their corresponding characteristics are presented in the following list and are depicted in Figure 6.1:

1. **Armadillo:** an armadillo displaying easy-to-recognize humanoid characteristics such as an upright stand as well as two arms and legs. In addition, the object has small features on its front side to evaluate their perception with increasing detail in the suggestive contours technique.
2. **Human brain:** this object has repetitive organic features and on a global scale a convex appearance making it useful to evaluate the perception of locally smoothed surfaces.
3. **Stanford bunny:** this object is well known in the computer graphics community, and as such, it is assumed to be highly recognizable. Furthermore, the object does not display strong concave areas or intricate details on most of the shown surface, which should lead to a better perception of its 3D shape.

4. **Heptoroid (heptagonal toroid)**: this object is an artificial mathematical structure with holes and radial symmetry, to be exact the object is a toroidal ring of seven 4th order saddles. For this object, it is assumed that participants are not familiar with it and as such it represents the baseline to evaluate the bottom-up processes involved in 3D shape perception without interference of top-down processes from previous encounters with this shape.

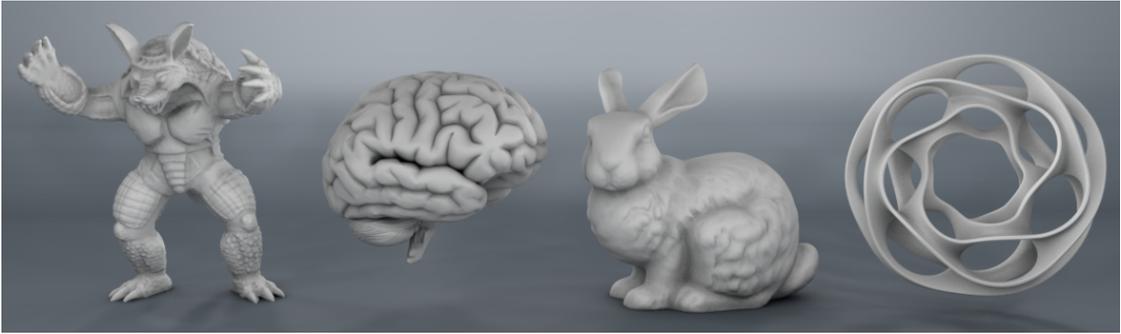


Figure 6.1: A rendered scene presenting the four stimulus objects used in the user study.

It is assumed that the armadillo, the brain, and the bunny cause the utilization of both top-down and bottom-up mental processes since participants are most likely acquainted with these objects on an abstract level (brain), on a familiar level (bunny), or a combination of both (humanoid armadillo). Due to its unfamiliarity, it is also assumed that the heptoroid produces the highest error in 3D shape perception among the four objects. All four objects come from the 3D model collection called the “Suggestive Contours Gallery” by Szymon Rusinkiewicz, Doug DeCarlo, Adam Finkelstein, and Anthony Santella, and were also used in previous line drawing research [DFRS03] [DR07] [GIHL00]. Simple objects like a sphere or torus were deliberately excluded from the user study since they do not display adequate visual differences on the various levels of detail for the suggestive contours method.

The second independent variable in the user study experiment is the level of detail in the suggestive contours NPR-rendering method. For this, five levels of detail with strong visual differences were chosen with each level including all lines of the previous level. In Figure 6.2 the five levels are depicted for four stimulus objects. The levels are:

1. An **outline** as defined for this thesis in Chapter 2.2
2. **Contours**
3. **Suggestive contours with $\alpha = 50$**
4. **Suggestive contours with $\alpha = 20$**
5. **Suggestive contours with $\alpha = 7$**

The goal of the five levels of detail is to create a visually monotonic parameter that describes the suggestive contours technique. The parameter α determines the level of detail for the suggestive contours methods as described in Section 5.1.4 - small values represent a high level of detail and large values a low level of detail. To achieve monotonic levels of detail an additional level - the outline of the stimulus object - is included which is non-parametric in contrast to suggestive contours. Level two represents contours and is included because suggestive contours are a natural extension to contours. Suggestive contours attach directly onto these classical contour lines. Level three to five are direct results of changing the main parameter of the suggestive contours method, i.e., the size of the neighborhood of nearby views to include contours from.

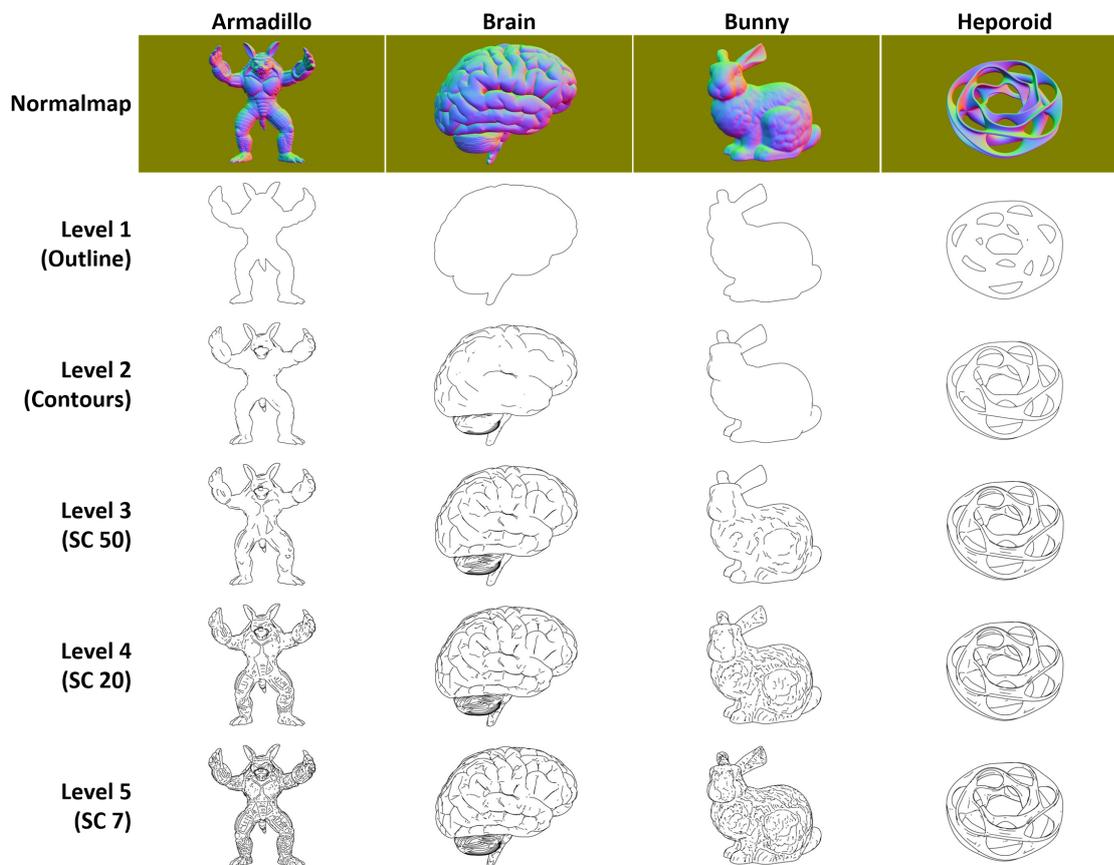


Figure 6.2: The five levels of detail for the four stimulus object.

6.2 Sampling Procedure

For the gauge figure task, the ground truth normals of the four stimulus objects are compared to the participants' estimates, which represents a deviation from the original gauge figure task by Koenderink et al. where ground truth normals were not used

[KvDK92]. Due to the continuous nature of both tilt and slant, it is necessary to define a discrete set of sampling points on the virtual hemisphere created by all possible tilt and slant combinations. The hemisphere is therefore divided into six different slants with a step size of about 15 degrees: 5, 20, 35, 50, 65 and 80 degrees with a tolerance of ± 1 degree. The possible tilts of 0 to 360 degrees are divided into eight disjoint, non-empty bins with a range of 45 degrees with the beginning of the first sector at 45 degrees, i.e., the north direction. Within each bin, a random candidate position is chosen.

Slants with 0 and 90 degrees were excluded due to being a point of singularity and the constrained maximal deviation respectively. At a slant of 0 degrees the gauge figure points directly in the direction of the viewing vector v , i.e., to the participant, and has an undefined tilt, i.e., all tilts are valid, since any rotation of the gauge figure around the axis created by the stick does not change the displayed gauge figure. Any small deviation from this position creates high changes in tilt; hence it is not sensible to include a slant of 0 degrees in the evaluation of the participants' shape perception. A slant of 90 degrees is problematic because the possible rotations of the gauge figure in the experiments are constrained to a maximum of 90 degrees, i.e., the user would just have to rotate the gauge figure until it does not rotate any further.

The discrete combinations of tilt and slant subdivide the hemisphere into 48 sections. With four different stimulus objects and five levels of detail, a total of 960 sampling points have to be estimated by every participant. This number seems high, but research shows that participants are able to perform this many estimates in the gauge figure task well [Pla15] [BWP⁺16].

In preparation for the user study experiment, a normal map is rendered for each stimulus object. On these normal maps, a brute force pixel by pixel grid search is performed to find the required sampling point positions. For this, every normal is extracted that does not lie on the background of the scene or lies on a line in the highest level of detail version of the line drawing for a specific stimulus object. Since every level of detail contains all lines from the previous levels, it is ensured that the participants are not presented with fringe cases along lines. From this pool of valid normal positions, the 48 combinations of tilt and slant are selected according to the constraints mentioned beforehand. The search for valid sampling positions is repeated for every stimulus object, and the resulting four sets of normals containing 48 sampling point positions are then saved into a text file to be used in the experiments on every level of detail for each object.

6.3 Experiment Setup

After finding 18 willing participants to take part in the user study, all participants conducted the same experiment as described in this section. First, the participants are seated, and a laptop is positioned in front of them. Then the general procedure of the user study and goal are explained to them. Initially, the participants are presented with a starting screen explaining the gauge figure task as shown in Figure 6.3 and a few

gauge figures are positioned to test as well as fully understand the task for higher quality estimates in the real gauge figure task.



Figure 6.3: The start screen explaining the gauge figure task to the participants.

After the test run the participants start with the actual gauge figure task. In this task they have to estimate the object’s surface at all predefined gauge figure positions for one object at a time as depicted in Figure 6.4 - initially for level one, then two, etc. until level five with fully detailed suggestive contours. This process took participants between 25 and almost 90 minutes, on average the participants finished after about 45 minutes. The estimated surface normals are saved into a text-file with additional information such as the ground truth, the object, the level of detail, and the local amount of ink in local circles with varying radii around the sampling position for further analysis.

The objects and the gauge figure are both rendered perspective with a focal length of 35.0 millimeters for visual alignment with the pre-rendered stimulus object. The gauge figure is added interactively in the designed gauge figure task framework as an overlay without interaction with the presented object similar to a second independent render pass. When the gauge figure position is further away from the viewing axis perspective distortions occur which may lead to a backward rendered gauge figure from the current viewpoint. The participants were informed about the perspective rendering and its visual impact to ensure proper understanding of the scene’s setup.

To highlight these distorted cases, the gauge figure stick and outer rim of the circle are rendered in red, and the top area of the figure is rendered in transparent green. When flipped due to perspective distortions, the backward facing side of the circle is rendered

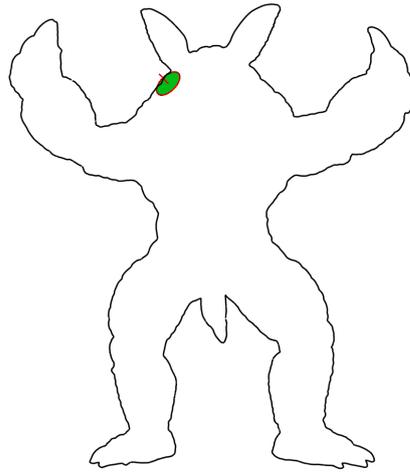


Figure 6.4: Example screen from the gauge figure task for the armadillo object at level one.

completely transparent as shown in Figure 6.5. In addition, for every new estimate of a normal, i.e., a new screen, the gauge figure is positioned with its stick perpendicular with the viewing axis to give the participant a visual clue about the strength of perspective distortion at this point.

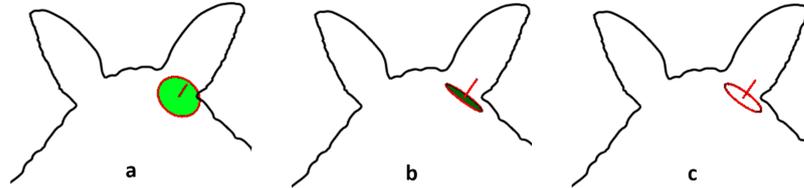


Figure 6.5: Snapshots of a rotating gauge figure from a front facing orientation (a) to a back facing one (c) due to perspective distortions.

The participants are all unpaid and are between 20 and about 45 years of age. The background of the participants is quite diverse ranging from professionals in related fields to this thesis like students and professors of visual computing to teachers of German, students of archeology and biology as well as chefs and entrepreneurs. In general, most participants were not tired during or after the gauge figure task, but two required additional motivation in the form of chocolate and sweets to complete the full experiment.

Results and Discussion

After conducting the user study and collecting estimates from 18 participants, the resulting total of 17,280 sample points is analyzed to answer the research questions of interest for this thesis. This chapter presents the findings of the analysis both in a numerical and visual way as well as discusses the empirical results of this thesis. In addition, this chapter also includes the details of how to use the gathered empirical results to effectively generate line drawings by optimizing the suggestive contours technique by means of an additional post-processing stage.

7.1 Results of Analysis and Discussion

This section explains the analysis and its results in detail for each of the research questions defined in Section 4.2.

7.1.1 Perceptual Shape Ambiguity

The first research question is concerned with how well participants can reconstruct an adequate 3D mental model of an object in the context of shape ambiguity introduced by the suggestive contours rendering technique - both in regards to the type of the object and the level of detail for the method. As tilts are usually estimated well, the percentage of tilts with a deviation of more than 90° is used as an indicator for shape ambiguity in this analysis.

Over all stimulus objects and over all levels of detail the ambiguity lies between 13.89% and 27.78%, the median is at 16.55 and the mean at 18.31% indicating a slightly skewed distribution. In Figure 7.1 the percentage of wrong tilts as defined for this analysis is plotted over all levels for each object; the exact percentages are presented in Table 7.1.

The plot shows that for the unfamiliar heptoroid a strong decrease in perceptual shape ambiguity is present, going down more than 20% from almost 28% to about 22%. For

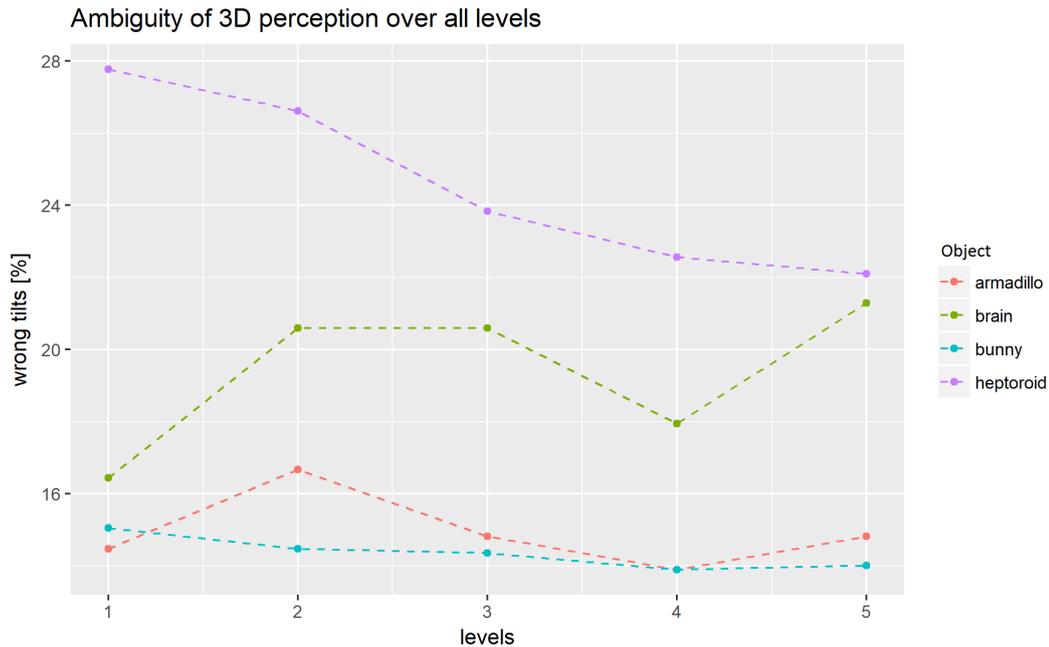


Figure 7.1: The percentage of wrong tilts over all defined levels of detail for each of the four stimulus objects.

familiar objects represented by the bunny and the armadillo, the change is less pronounced. The organic and familiar shape of the brain even has a significant overall upward trend contrary to the decrease in the heptoroid case. This might be explained by the small creases created by the brain’s finer structures. The overall convex and almost sphere-like shape is perceived as such in the lowest level, but with more detail, the creases become visible and affect the participants’ overall impression of a convex object.

In summary, these results indicate that at least for unfamiliar, artificial and complex objects like the Heparoid shape ambiguity is decreasing with the level of detail while familiar object experience almost no change and organic shapes with fine structures tend to become more ambiguous.

7.1.2 Error of 3D Shape Perception

The second research question is concerned with the error when estimating shape from line drawings in regards to the level of detail and the type of stimulus object used. Since depth and shape are mainly conveyed via slant, the slant error is used as key indicator for the quality of shape perception. The slant error for an individual estimate is calculated with $error_{slant} = slant_{GT} - slant_{est}$. A negative value means that the ground truth slant was smaller than the estimate and the participant overestimated the slant meaning that the estimate was too steep, which indicates that the object is perceived as having more depth than it really has. Similarly, a positive value indicates an underestimation, i.e.,

Object	Level of Detail	Percentage
Armadillo	1	14.5
Armadillo	2	16.7
Armadillo	3	14.8
Armadillo	4	13.9
Armadillo	5	14.8
Brain	1	16.4
Brain	2	20.6
Brain	3	20.6
Brain	4	17.9
Brain	5	21.3
Bunny	1	15.0
Bunny	2	14.5
Bunny	3	14.4
Bunny	4	13.9
Bunny	5	14.0
Heptoroid	1	27.8
Heptoroid	2	26.6
Heptoroid	3	23.8
Heptoroid	4	22.6
Heptoroid	5	22.1

Table 7.1: Percentages of wrongly adjusted tilts for all object and level of detail combinations.

the object is perceived flatter than it really is. For this thesis, the median of $error_{slant}$ is used as a robust indicator of perceptual accuracy while the standard deviation is used as an indicator of precision.

The median slant error over all participants, levels, and objects is 8.34° indicating a systematic underestimation, i.e., in general, the objects for suggestive contours are perceived flatter than they really are. In contrast, the tilt median is -0.32 showing no significant systematic rotational error around the viewing axis. The overall slant standard deviation lies at 28.39° while the standard deviation for tilt lies at 126.45° . When accounting for shape ambiguity, the standard deviation for tilt drops down to 72.22° .

The overall accuracy improves with increasing level of detail as depicted in Figure 7.2, which confirms the idea that higher levels of detail are adding more visual information. In contrast to this, precision, as indicated by the standard deviation, shows a more complex behavior as depicted in Figure 7.3. The precision improves for both the bunny and the armadillo but decreases for the brain and heptoroid. Over all objects, the precision

changes from 26.6° at level one to 28.4° at level five, which is a slightly significant change of 6.7%.

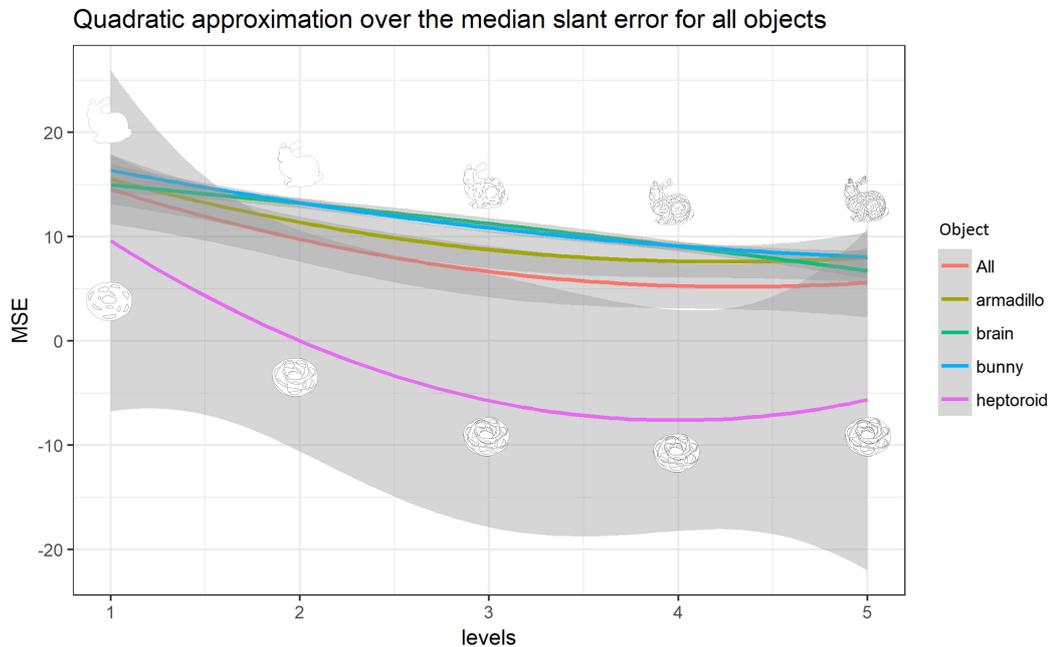


Figure 7.2: Quadratic approximation of slant error for each of the four stimulus objects and their aggregate. All five approximations show a trend of lower median slant error, i.e., higher accuracy, with increasing level of detail. For better understanding, the bunny and the heptoroid stimulus images are depicted for every level.

When differentiating between stimulus objects, the armadillo, the brain, and the bunny show very similar behavior with a strongly significant improvement in accuracy of shape perception with increasing level of detail as shown without approximations in Figure 7.4. All known objects are systematically overestimated between 6.65° and 16.34° indicating a flatter appearance than they actually have. This aligns with the hypothesis that increasing the visually monotonic variable of level of detail increases shape perception.

For the assumed to be unknown heptoroid object, the median slant error shows a different behavior: the accuracy of 3D perception of shape improves with strong significance too, but starting from level two onwards the heptoroid is underestimated up to about 7.53° indicating a perception of more depth than it should actually convey. This behavior might be caused by the unknown and artificial nature of the object where participants may be uncertain and give their estimates “a nudge” to compensate for the uncertainty of the object’s shape.

The distributions of slant error are plotted in Figure 7.5 showing both the discrepancy between known objects (armadillo, brain, bunny) and unknown objects (heptoroid) as well as the similarities between the distributions of all objects.

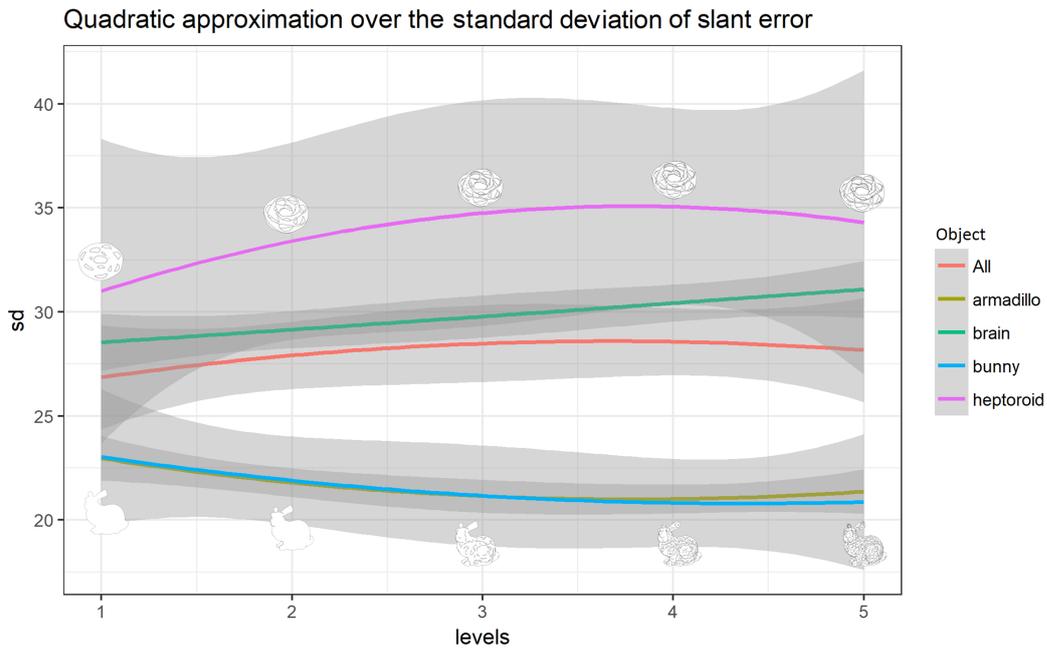


Figure 7.3: Quadratic approximation of slant standard deviation for each of the four stimulus objects and their aggregate. For better understanding, the bunny and the heptoroid stimulus images are depicted for every level.

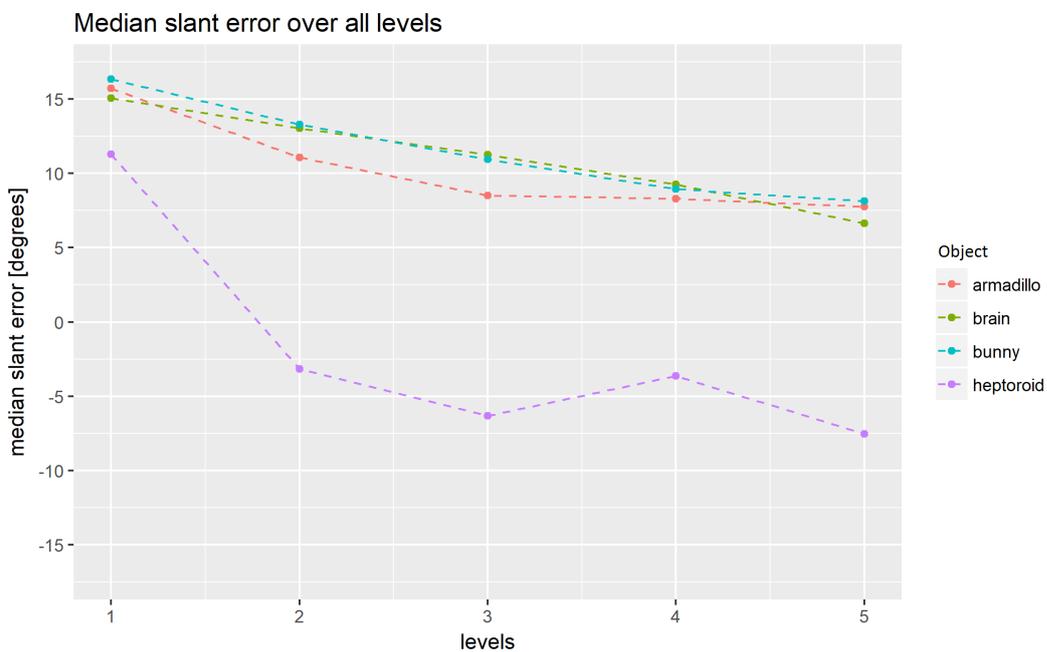


Figure 7.4: Median error of slant estimates over all levels for each stimulus object.

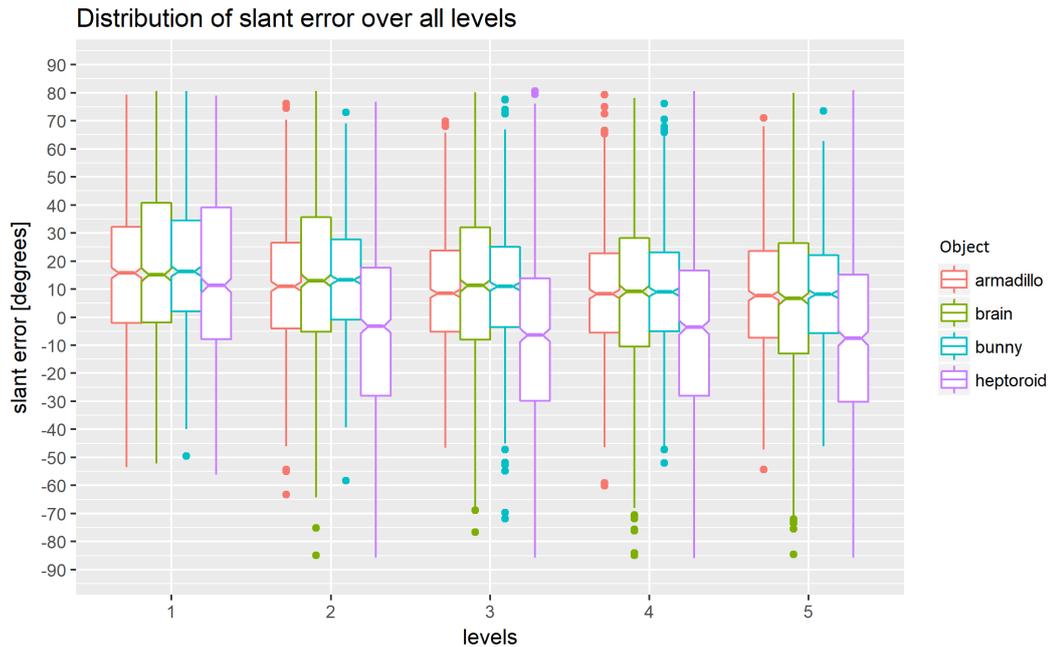


Figure 7.5: Boxplot of slant error distributions.

For the familiar stimulus objects, these findings are consistent with previous research in terms of general underestimation [STPV12], [MK96], [BWP⁺16]. The range of median slant error from the ground truth for known objects of 6.65° to 16.34° even lies below the range stated in literature with 15° to 25° [STPV12], [MK96], suggesting better shape perception is achieved by suggestive contours than with Blinn-Phong shading. This better result also contradicts the findings of Cole et al. who state that the average error for the perception of shape with suggestive contours is about 30° when adjusted for ambiguity.

There are several possible explanations for this discrepancy. For example, Cole et al. rely on Amazon’s Mechanical Turk for collecting the gauge figure samples. Mechanical Turk is a platform for collecting a large number of user estimates for a small amount of money. The users’ intent in joining the study on such platforms may not be attributed with high motivation for precise and accurate estimation in the task at hand. Second, it is unclear at which level of detail the suggestive contours images of Cole et al. are created, which renders a direct comparison of results invalid.

In summary, the suggestive contours renderings used for this user study seem to indicate a better perception of shape than found in previous experiments, but more quality user studies with highly motivated participants are needed for a justified explanation.

When investigating accuracy indicated by slant error on a sectoral basis of tilt and slant combinations, the results show that smaller slants are relatively well estimated as depicted in Figure 7.6. Again, the armadillo and brain show similar distributions of median slant error over all levels while the brain has slightly higher positive errors,

i.e., redder in the plot, indicating a stronger underestimation. The heptoroid displays a contradicting color scheme for lower slants indicating a slight overestimation over all levels with a tint of blue while larger slants are underestimated similarly to but not as strongly as the three familiar stimulus objects. Overall, the median slant errors are not distributed evenly across all tilt and slant combinations which is consistent with previous research [STPV12].

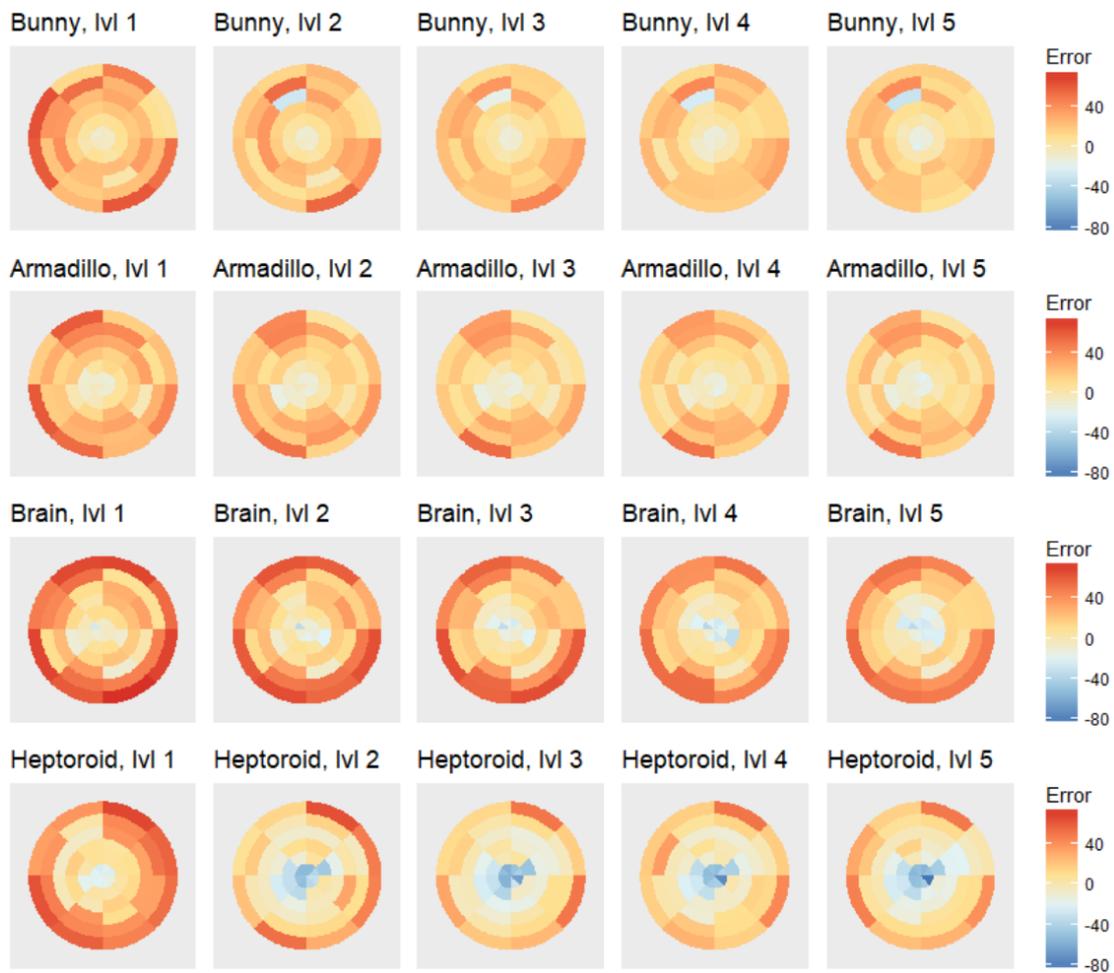


Figure 7.6: Polar plots of the four stimulus objects over all levels showing the median slant error per sector of tilt and slant combination.

7.1.3 Steven's Power Law Regression Fitting

As precision, measured by the standard deviation in gauge figure tasks, is usually quite high across most previous user studies indicating an inherently high variability in participants' estimates, only the accuracy measured by the median slant error is considered for robust model fitting in this thesis. Such a fitted model of median slant

error over the defined levels of detail shows how the systematic error of shape perception changes.

Steven's power law describes the functional dependency of an input stimulus' magnitude - in this case the level of detail - to the perceived magnitude. The perceived magnitude for this user study is the accuracy of shape perception. The slant error describes the exact opposite of this and therefore needs to be inverted. This is achieved by multiplying the data by -1 and adding a constant shifting all data points into a positive range.

Since only the average perception of the participants is of interest for this thesis, the slant errors are not pooled together like in the analysis for the previous research questions, but the median of slant error is calculated per participant resulting in 18 median error points per level of detail as shown in Figure 7.7. This plot shows that the collected data does not properly fit an exponential function but rather a quadratic one. Even when accounting for outliers in the participants by calculating the median per level, the resulting data set only achieves an exponential model fit with a p-value of 0.5465, an adjusted r^2 -value of 0.6777 and an AIC-score of 24.23549. Taking into account that taking the median twice - once per level and participant and once per level again - produces a data set of mere 5, albeit very robust, values, it is relatively safe to say that Stevens's power law is not an adequate model for representing the relationship between a visually monotonic level of detail parameter for the suggestive contours technique and the resulting quality of perception of 3D shape.

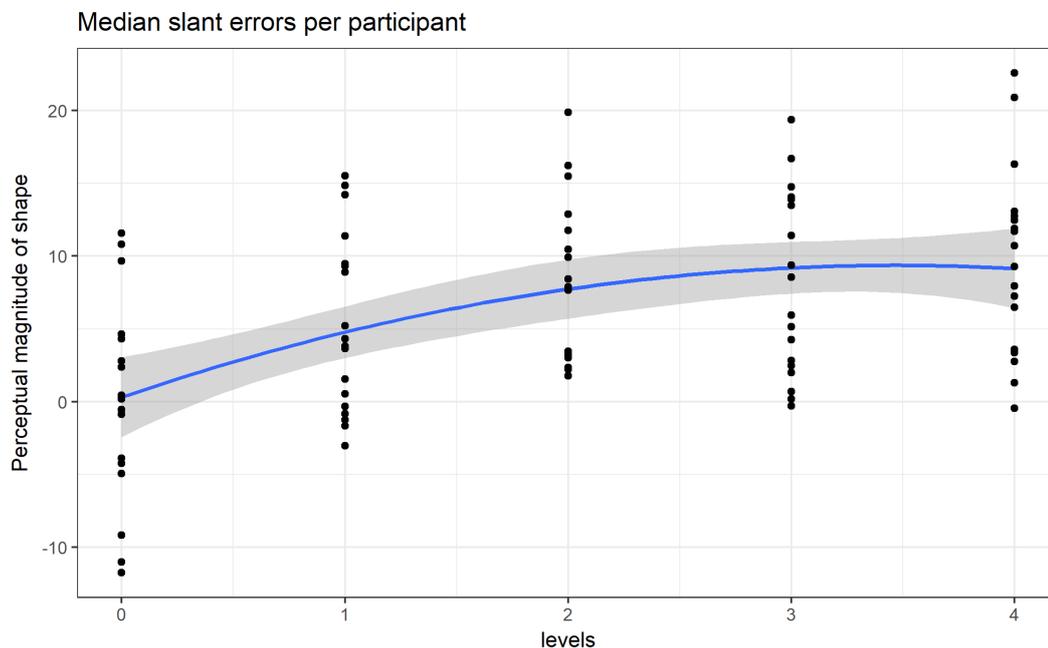


Figure 7.7: Quadratic model fitting the 18 inverse median slant errors of the participants over all levels.

Model	Formula	p-value	Adjusted r^2	AIC	Parameters
Linear	$y = \sum_{i=0}^1 a_i x^i$	0.03665	0.7499	24.38717	$a_0 = 15.0909$ $a_1 = -2.2391$
Quad.	$y = \sum_{i=0}^2 a_i x^i$	0.02179	0.9564	15.62255	$a_0 = 8.3736$ $a_1 = -7.0806$ $a_2 = 3.1988$
Cubic	$y = \sum_{i=0}^3 a_i x^i$	0.04367	0.9953	3.02899	$a_0 = 8.3736$ $a_1 = -7.0806$ $a_2 = 3.1988$ $a_3 = -1.1278$
Exp.	$y = (x + 1)^a + e^b$	0.00257	0.9561	-6.70878	$a = -0.9480$ $b = 3.2969$

Table 7.2: Model characteristics when fitting to the median of slant errors per level.

When fitting a curve to the median of slant errors per level rather than the inverse as assumed above, relatively good fitting models are found as shown in Table 7.2. The four tested models are polynomials of degree one (linear), two (quadratic), and three (cubic), as well as an exponential function as shown in Figure 7.8. All models are significantly better than their null models, e.g., for the linear model the dependent variable being equal to its mean, with p-values below 0.05 and, except the linear model, achieve an adjusted r^2 -score of more than 95%. Adding an additional parameter from the quadratic model to the cubic one reduces the AIC-score from 15.62255 to 3.02899 while also increasing the adjusted r^2 -value to almost an ideal 100%. All four models have relatively low absolute model parameter values indicating no presence of overfitting.

Even though the exponential model has a lower adjusted r^2 -value of 0.9561 instead of 0.9953 for the cubic model, the exponential model is preferable due to having fewer model parameters, which is also visible in the lower AIC-score. Overall, the exponential model $y(x) = (x + 1)^a + e^b = (x + 1)^{-0.9480} + 27.03$ with x being the level of detail and y the median slant error describes the collected data relatively well. This result indicates that even if Steven's power law does not fit the data properly, there seems to be an exponential relationship between the level of detail and the human error of shape perception.

So far the inconclusive analysis of possible connections to Steven's power law is made under the assumption of a visually monotonic selection of reference renderings with the suggestive contours technique mapped to five levels of detail ranging from 0 to 4. By examining this selection of levels of detail, it becomes clear that by the act of selecting visually monotonic images by the author of this thesis perceptual processing already takes place. The parameter adjusted for this monotonic characteristic of the image series is the sphere size of the suggestive contours algorithm. By examining the implicitly chosen values of 7, 20 and 50 and approximating these three values with an exponential function, the resulting equation is $y = 2.6781 \cdot e^{0.9831x}$ with $r^2 = 0.9985$. This approximation fits well and might suggest that the author's selection of visually monotonic images was

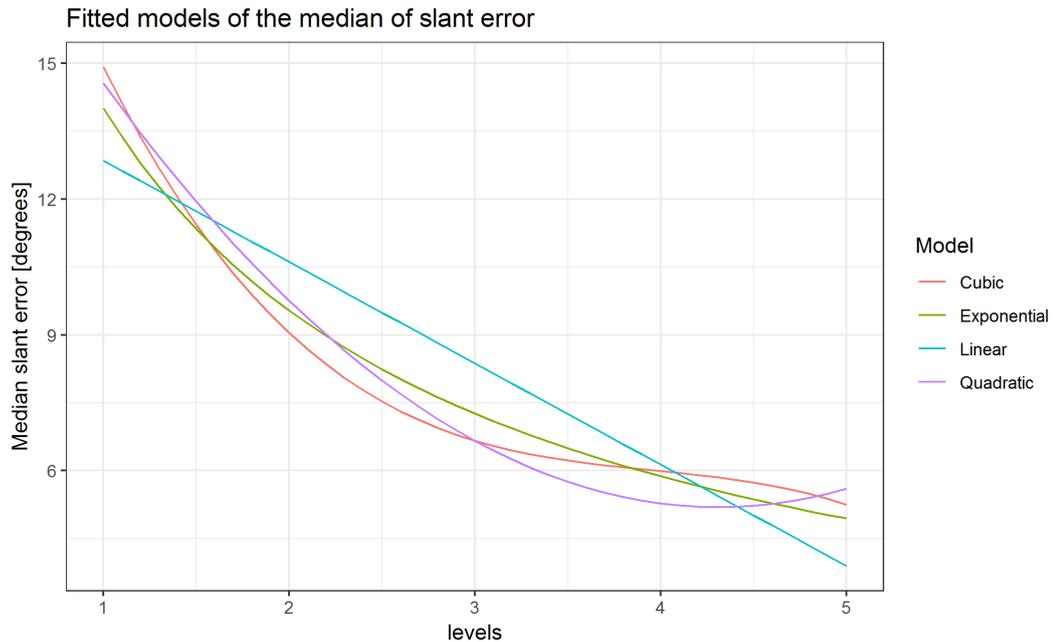


Figure 7.8: Different curve types fitted to the median of slant error.

implicitly guided by an exponential form of perception as described by Steven’s power law. Due to the limited granularity of the 3 suggestive contours levels of detail, further user study must be conducted in order to investigate this new hypothesis.

7.1.4 Size of the Local Estimation Area

When participants perform the gauge figure task, they use both a top-down and a bottom-up approach to visually estimate the presented surfaces. The top-down visual processes are the application of collected knowledge of previously encountered objects in the real world that help to interpret both familiar and unfamiliar objects - in short, making sense of the world around us from previous experiences. Bottom-up processes on the other hand focus on the interpretation of low-level information from lines and other simple visual cues. Usually performed in milliseconds and before any conscious thinking takes place, bottom-up processes are somewhat “hardwired” into the lower levels of human visual cognition.

In the context of bottom-up low-level visual cues, it is interesting to look into the question of how large the area for local surface estimation is, i.e., what area of the surface is used for interpreting 3D shape from lines on a lower cognitive level. For this, the average ground truth slants of differently sized local neighborhoods are compared to the participants’ estimates. For each estimate, the best fitting neighborhood is selected, and histograms of these best fitting neighborhoods are created for every object as well as every level and analyzed. The size of a neighborhood is defined as a radius in pixels ranging from 0 to

40. The average slants are calculated from the projected normals in image space via the pre-rendered normal map. For this, only pixels falling within a circle with the target radius around the sampling position are used for the calculation of the average slant. The resulting histograms on a per object basis, on a per level basis as well as both, are depicted in Figure 7.9.

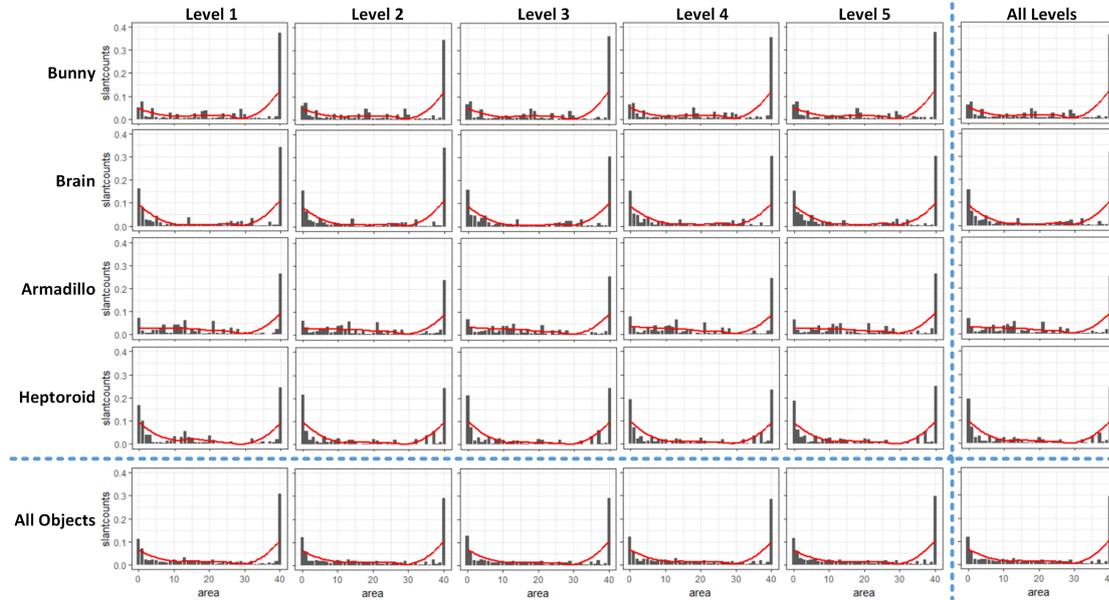


Figure 7.9: This overview show the 20 histograms of best fitting local slant average for all stimulus object and level combination. In addition, the summarized histogram per object and per level are included. The bottom-right histogram shows the distribution of best fitting local area of estimation over all collected data points.

All objects over all levels except the armadillo show a strong tendency of estimating slant mostly in a small 10-pixel radius around the sampling position. In the experiment, a local neighborhood with a radius of 10 pixels corresponds to approximately 1% of the display’s width. This is especially strong in the case of the brain and the heptoroid.

All histograms show a strong peak at a radius of 40 which is probably attributed to the fact that the participants’ intuition is to use the average slant of a larger area than the sampled 40 pixels for estimating the slant. Points where this tendency is strong collect in the largest bin of the histogram as it is the closest value to the area they use to estimate the slant, i.e., the underlying distribution has a long tale to the right. The armadillo object has a similar distribution as the other objects but with a less pronounced density around the smaller radii.

In general, there is a slight tendency to use a smaller neighborhood with increasing level of detail which may be attributed to more information present in the stimulus objects. For example, this trend is visible if one compares “All Object” at level one with level five

where there are more slants estimated from neighborhoods with smaller radii.

In summary, the data shows that there seems to be a systematic tendency to use either rather small radii for estimating the slant or rather large ones. This may be explained by the participants using local features for most of their estimates but when the estimation is not sufficient enough top-down processes become more active, and the object is estimated in regards to its overall structure rather than from local features.

7.1.5 Optimal Amount of Local Ink

The last of the research questions is concerned with optimizing the amount of local ink. The goal is to generate suggestive contours line drawings with a high amount of perceived shape information while retaining the amount of used ink optimally. For this, the absolute slant error is analyzed in regards to the percentage of local ink in differently sized neighborhoods. Similar to the previous research questions, the neighborhood is defined in terms of the radius of a circle around the sampling point in pixels. The minimal error defines the threshold of how low the amount of local ink at which level may be while still retaining good perception of 3D shape. Figure 7.10 gives an overview of the error surfaces created by cubic approximations at differently sized local neighborhoods for measuring the percentage of ink.

The error surfaces show a degrading amount of percentage of ink with increasing size of the local neighborhood. This is expected as in line drawings the main image content is white background of which the percentage increases when calculating it on a larger area. The minima of slant error tend to lie at either level one or level five. By analyzing the slant error over all levels, a more accurate approximation can be derived due to the larger data set size. This approximation for differently sized local neighborhoods creates the graph depicted in Figure 7.11 as a result.

As an approximation for the non-linear relationship between the absolute slant error and the percentage of local ink for a given circular neighborhood a cubic spline regression model is used. The knots are evenly spread through the covariate values and are penalized by the conventional integrated square second derivative cubic spline penalty as described by Wood [Woo06]. The knots in splines reduce the ringing artifacts at the ends of the definition interval one would expect when using conventional polynomial approximations. In addition, by using cubic regression splines with an optimal number of knots found by the minimal AIC-score the risk of overfitting is reduced.

The red line in Figure 7.11 is the cubic spline approximation containing the global minimum, i.e., the optimal amount of ink with the lowest absolute slant error. The value of this optimum is an ink-percentage of 17.3% in a local neighborhood of 36 pixels. These thresholds can be used to create an optimized version of suggestive contours line drawings with a high amount of 3D shape information while being low in the amount of ink used as described in the next Section 7.2.

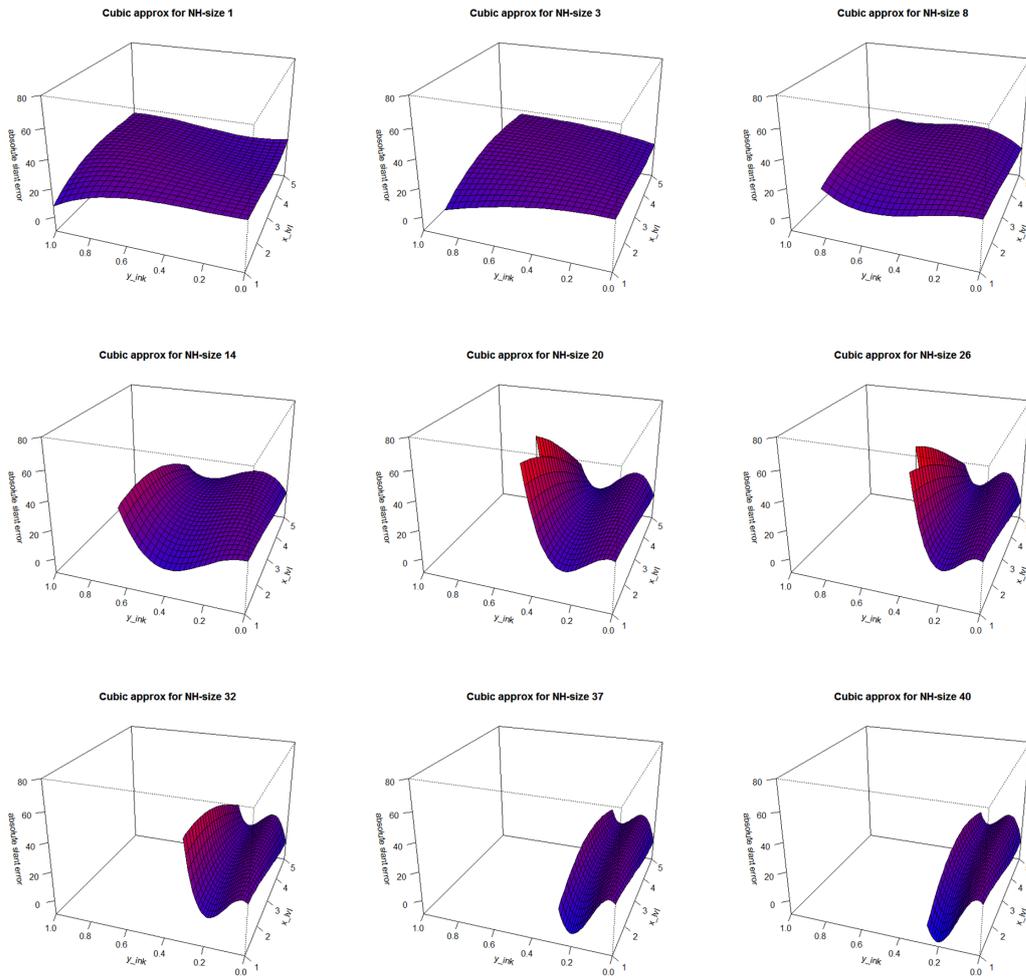


Figure 7.10: Cubic approximations of the error surfaces for differently sized local neighborhoods.

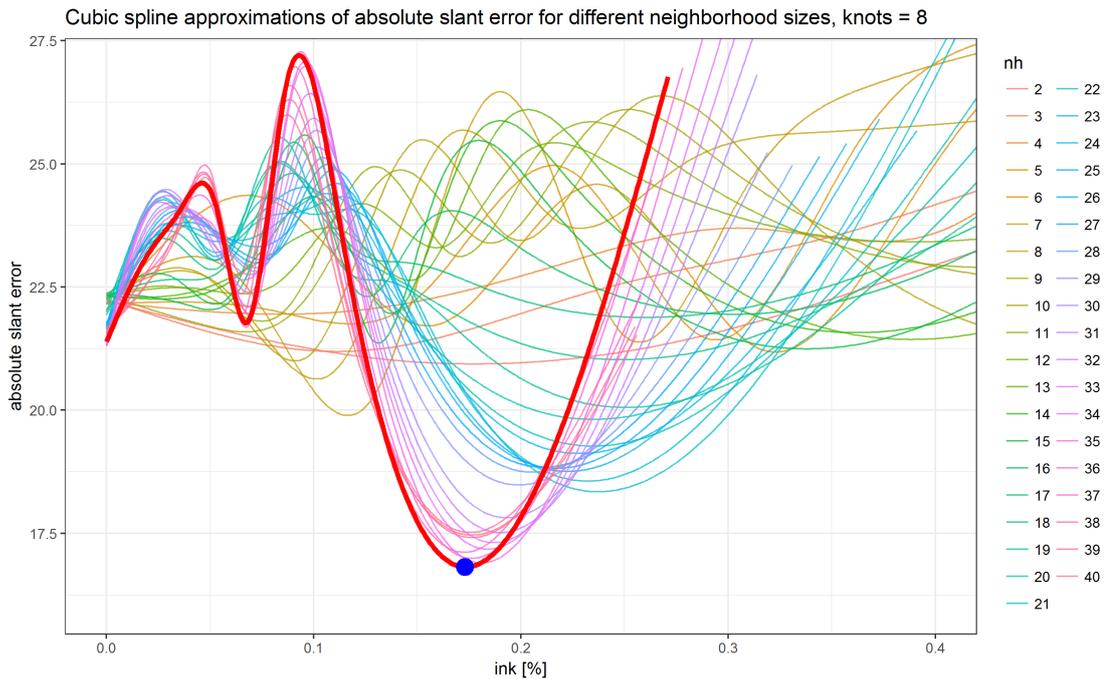


Figure 7.11: Approximations of the slant error in regards to percentage of ink over all levels for differently sized local neighborhoods. The red spline indicates the approximation containing the global minimum.

7.2 Optimizing Suggestive Contours for Shape Perception

The optimal thresholds for local ink percentage and the size of the local neighborhood to estimate it from are found to be a percentage of 17.3% in a circular local neighborhood of 36 pixels. These thresholds are used to create an optimized version of the suggestive contours images from the stimulus renderings used in the user study - one for each of the four objects.

The goal is to create a suggestive contours image where for every pixel the local ink percentage in a circular local neighborhood of 36 pixels is as close as possible to 17.3%. To ensure that the algorithm is bounded and does not create gaps in the boundaries, the silhouette image is used a lower bound, i.e., the algorithm can only add detail to the silhouette image. To create optimized suggestive contours images a greedy local nearest neighbor approach over the five reference images is utilized and implemented in Matlab.

For each pixel, the local percentage of ink in a circular neighborhood of 36 pixels is calculated for the five stimulus images by applying a convolution with a custom circular kernel the size of the neighborhood. The kernel is a black square image with $length = 2r$, where r is the radius of the neighborhood, i.e., 36 pixels. In the kernel, all pixels within a circle with size of the neighborhood are set to 1 corresponding to white. By dividing

each pixel in the kernel by the number of white pixels in the kernel, a non-changing overall brightness is ensured when applying the kernel. Convoluting a black and white line drawing with this kernel is mathematically equivalent to a local ink percentage calculation for each pixel.

After applying this kernel to all five reference images, each pixel of the resulting optimized image, which has the same size as the reference images, is set to the color - black or white - of the corresponding pixel in the five reference images with the closest ink percentage to the threshold of 17.3%, i.e., the nearest neighbor. The result of this procedure is an image where each pixel represents the value of the corresponding reference image pixel with the closest local ink percentage to the empirical threshold of 17.3%. This process is summarized as pseudocode in Algorithm 7.1.

Algorithm 7.1: Generating optimized suggestive contours images

Input: Size of a circular neighborhood ns , empirical ink-threshold t_{ink} , an image S with five channels where each channel corresponds to a level of detail of the suggestive contours method with channel one being the image with lowest level of detail

Output: Optimized suggestive contours image O

```

1 kernel = createCircularKernel(ns);
2 kernel = divideKernelBySumOfValues(kernel);
3 percS = conv(S, kernel);
4 O = zeros(size(S(:, :, 1)));
5 difAbs = abs(percS -  $t_{ink}$ );
6 [rows, columns, levels] = size(percS);
7 for y = 1 : rows do
8   for x = 1 : columns do
9     // get the index of the pixel with the closest value
10    [ , minDifIdx(y, x)] = min(difAbs(y, x, :));
11    O(y, x) = S(y, x, minDifIdx(y, x)); // set resulting pixel to its value
12  end
13 end
14 return O;

```

For the armadillo object, the resulting optimized image is shown in Figure 7.12 along with visualizations of the differences to the original five levels of detail. The visualization of differences shows in blue that for levels one and two lines are only added, which is to be expected since these levels represent the silhouette and simple contours respectively. Beginning from level three the suggestive contours become plentiful adding more lines and therefore getting closer to the optimum as defined by the empirical thresholds. In level five no additional information is added in comparison to the optimized image but only removed as shown in red.

In general, the presented optimization algorithm seems to be able to add lines, i.e.,

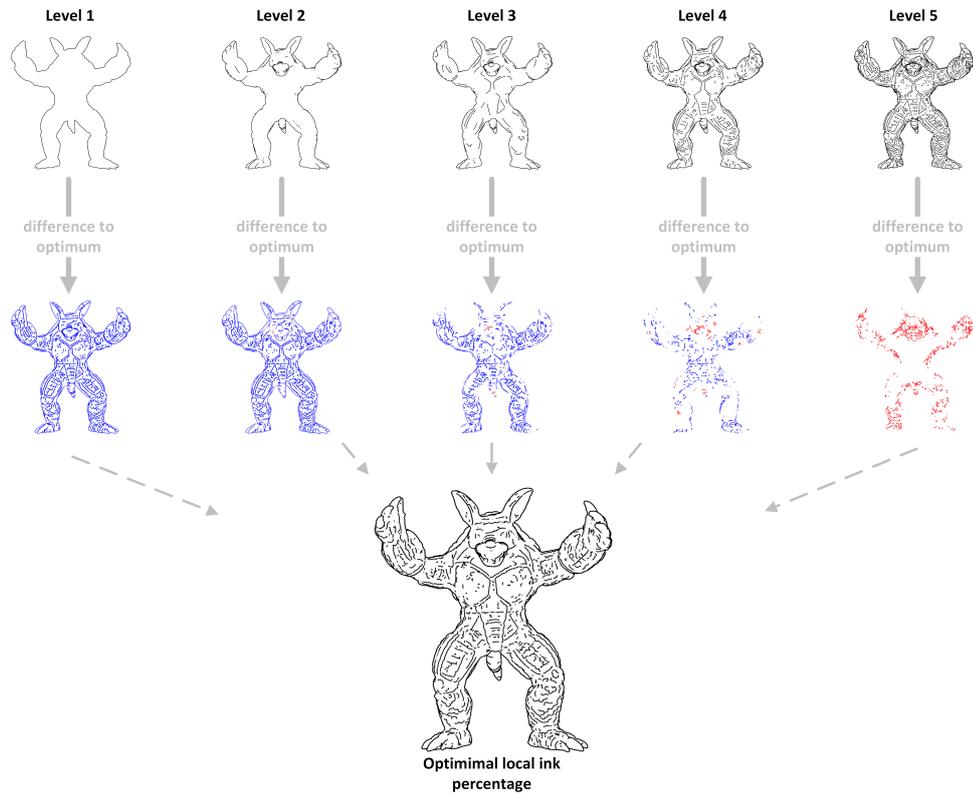


Figure 7.12: Visualization of the optimized armadillo stimulus object and the differences to the five levels of detail. Red indicates a removal of lines by the optimization routine and blue shows additions.

information, from higher levels of detail where necessary and removes unnecessary detail generated by the suggestive contours methods for pixels where the fewer lines suffice. The initial problem of suggestive contours to find a good level of detail is therefore solved by creating an optimized image from a spectrum of several levels of detail. One simply generates suggestive contours images ranging from very few lines to highly detailed ones and uses the presented algorithm to generate a new image optimized for perceptual shape understanding. This removes the challenge of finding optimal suggestive contours parameters for adding an extra post-processing optimization step.

For the other four stimulus objects, the optimized versions and the visualizations of differences are shown in Figure 7.13. Similar to the armadillo object, the optimization for the brain in the first levels encompasses mostly additions of lines. The differences to the last two levels show mostly removals of lines in areas with high mesh frequency like around the brain stem. For the bunny and the heptoroid, this removal of lines in areas of high mesh frequency is less pronounced as these two objects have fewer high-frequency areas and a generally smoother surface. All four cases show how the nearest neighbor

approach described above achieves sensible and well-looking results by either adding or removing lines to produce images close to the empirical local thresholds collected in the user study of this thesis.

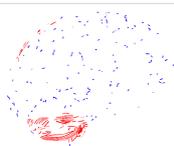
Optimal local ink percentage				
Difference to level 1				
Difference to level 2				
Difference to level 3				
Difference to level 4				
Difference to level 5				

Figure 7.13: Visualization of the optimized suggestive contours images for all four stimulus images and the differences to the five levels of detail. Blue indicates additions of lines and red indicates removals of lines.



Conclusion

The complexity of modern line drawing algorithms allows users to generate images of various levels of detail by simply adjusting input parameters. This freedom comes with the drawback of having to choose an appropriate level of detail based purely on visual aesthetics. In case of suggestive contours, this input parameter is the distance of nearby views from which to incorporate lines into the final rendering. In this thesis, a novel post-processing meta-heuristic for optimizing line drawings from complex algorithms like suggestive contours is proposed. The proposed approach may be used in practical applications such as furniture handbooks or architectural renderings to produce images with high expressiveness in 3D shape understanding while reducing the amount of ink.

The additional post-processing stage is not only able to compensate the shortcoming of having to choose an appropriate level of detail for complex line drawing algorithms but also capable to find a result that optimizes for 3D shape perception while retaining the amount of ink to a minimum by means of empirically found perceptual thresholds - local neighborhood size and local ink percentage. By collecting these thresholds for other line drawings methods such as ridges and valley or apparent ridges and analyzing them according to this thesis, the proposed meta-heuristic may be used for optimizing these algorithms as well or even to optimize for visual characteristics other than 3D shape perception, e.g., cognitive load.

To find the empirical thresholds, a user study guided by the design of previous research experiments and based on the gauge figure task was conducted to collect more than 17.000 high-quality user estimates of surface normals utilizing a custom-made gauge figure task framework. The collected samples were analyzed to answer the defined research questions and to yield the empirical perceptual thresholds for the developed post-processing stage.

The analysis of sample points could confirm the common finding of previous shape perception studies that participants generally underestimate, i.e., perceptually flatten, 3D objects in the gauge figure task. This result validates not only previous research but also

the study design for this thesis. In addition, a possible link of human visual perception of 3D shape to Stephen's power law could not be established, but a new data-driven hypothesis could be formed to further investigate this connection based on the visually monotonic aspect of the chosen levels of detail. Furthermore, the results of this thesis broadened the understanding of how humans perceive 3D shape from line drawings and how shape ambiguity behaves in regards to different levels of detail and type of object presented to participants. Besides this, the additional findings of this thesis, summarized in the research questions, open up new ways to explore our understanding of shape perception to get us closer to the goal of understanding how humans perceive 3D shape from line drawings.

Future Work

Although the analysis of the collected data yielded interesting results to answer the five stated research questions and the results are used successfully to improve the suggestive contours method, new possible routes of improvement and new ideas to be explored opened up during the course of this thesis. In this section, these new ideas and possible improvements are summarized.

The first possible improvement for future work based on the results of this thesis concerns the optimization algorithm. The algorithm designed for this thesis is based on a one-pass greedy nearest neighbor approach, i.e., the interpolation procedure uses the closest value available. A natural extension when using interpolation methods is to use more accurate methods such as linear interpolation or interpolation with a polynomial of higher degree. In the case of optimizing suggestive contours, one would not just use the pixel value of the level with the closest local ink percentage to the threshold but use interpolation between two or more levels to improve results. This improvement would most likely need a more fine-grained spectrum of levels of detail than the five used for this thesis.

In addition to this, another improvement of the algorithm concerns the one-pass greedy characteristic. Greedy algorithms search for global optima by taking the locally optimal choice, but this optimization heuristic does usually not find the globally optimal solution. When optimizing suggestive contours as a post-processing pass like for this thesis, local pixel values are changed to produce perceptually improved images. Changing the value of a pixel also changes the local ink percentages of all other pixels in the defined neighborhood which yields result where the local ink percentage in the optimized image is not as close to the optimal value as possible. An iterative procedure that uses several passes over all pixels to converge the local ink percentages closer to the empirical optimum of 17.3% would probably yield better results. This multi-pass procedure could be repeated until there is no significant change of pixel values anymore, i.e., using nearest neighbor iteratively until all pixels are set to the pixel color of the level of detail closest to the threshold of 17.3%.

An assumption for this thesis is that the processing of lines uses mostly bottom-up cognitive processes, which is also backed by biology when looking at basic line processing occurring already in the retina. The results show that most of the participants' estimates match the average normal of the local neighborhood at the sampling point visible by the peak of matches in the histograms of Figure 7.9. The histograms also show a long-tailed distribution indicating that for some sample positions a larger area is used to estimate the local surface shape. Based on these findings, an interesting path for future investigation would be to look closer into top-down processes by using additional methods like eye-tracking to pinpoint the looking pattern while internally reconstructing 3D shape from line drawings. This could also be combined with a continuation of the research conducted by Cole et al. to take a closer look into line types and their ability to convey shapes [CSD⁺09], i.e., which local lines are not suitable or need additional visual information to convey 3D shape accurately.

Another possible way for future research is the confirmation of findings of this thesis by collecting additional data to facilitate more in-depth analysis of human 3D shape perception. Widely used tools such as Amazon Mechanical Turk could be used to collect more data points. The downside of this tool is that additional information of the participants is scarce and the motivation of participants is somewhat questionable in comparison to the few but high-quality samples used for this thesis. The larger number of data points possible to collect with Amazon Mechanical Turk may make up for these disadvantages and allows for interesting new findings and confirmation of the presented results. Another way to augment the participants' estimates would be to find and use suitable previous data sets of earlier research, which might further diversify the pool of participants' estimates and increase the validity of results.

Already hinted in the discussion of results, further analysis of the connection between human 3D shape perception from line drawings and Stephen's power law would probably yield a deeper understanding of this functional dependency. The results of this thesis suggest that there might be a connection based on the selected visually monotonic series of suggestive contours renderings, but the user study setup and the collected data points do not allow further investigation of this connection. Future researchers could conduct user studies with a broader spectrum of levels of detail to test the validity of this hypothesis.

Overall, the results of this thesis confirm earlier research in this field, tested new hypotheses, and the optimized suggestive contours renderings seem plausible and may be improved with the suggestions provided in this section. As with all new findings, future research needs to be conducted to confirm the findings and test the validity of the results. The general methodological approach of adding a post-processing stage for optimizing an existing line drawing method may also be applied to line drawing algorithms other than suggestive contours to improve their perception in regards to shape understanding. Other complex algorithms like ridges and valley or apparent ridges would be ideal for applying this post-processing step to as they are similar to suggestive contours in terms of line drawing granularity and their ability to depict 3D shape. Another possible route to take in the future is to use this new post-processing stage to not optimize for 3D

shape perception but other characteristics like for example cognitive load measured via the time to estimate surface normals.

In accordance with earlier research in the field of line drawing perception, the findings of this thesis are descriptive. A natural extension of this work would be to conduct prescriptive user studies, e.g., to predict which lines depict a given shape most accurately while minimizing the amount of ink used. Especially the optimized suggestive contours renderings would benefit from being tested in subsequent user studies to confirm the improved 3D shape perception with less ink to help solve the ultimate goal in this field: to understand how and not only how well lines are able to convey 3D shape.

List of Figures

1.1	Example of a 30.000-year-old cave painting of a hyena found in the Chauvet Cave, France. (Image courtesy Wikimedia Commons)	1
1.2	Example of a line drawing in a modern architectural sketch. (Copyright by Braunger Wörtz Architekten)	2
2.1	Example of a line drawing by Michelangelo. It is not quite clear what is happening in the scene except that the illustration shows a battle scene or a fight. This line drawing focuses on conveying emotion rather than exact and veridical information.	7
2.2	Example of simple abstractions created by renderings algorithms [DS00].	8
2.3	Examples of how there are several methods of conveying the same information with different line rendering styles [CGL ⁺ 08].	9
2.4	Examples of suggestive contours, ridges and valleys, and apparent ridges in a direct comparison [JDA07].	10
2.5	A situation showing a surface in the radial plane with both a contour point q and a suggestive contours point p from the main viewpoint c . When changing the viewpoint to c' , a contour is visible in p	11
2.6	Illustration of definition II with suggestive contours being the points where $n \cdot v$ reaches a local minimum.	11
2.7	A situation with two inflection points satisfying the constraint $\kappa_r = 0$. In contrast to point b , point a also satisfies the second constraint $D_w \kappa_r > 0$	12
2.8	The left image (a) depicts the projection of v onto the tangent plane to obtain w . The right image (b) shows how p , n , and w form the radial plane which slices the surface along the radial curve with curvature $\kappa_r(p)$ [DFRS03].	12
2.9	An example of ambiguity in lines. There are infinitely many possible lines - all mapping to the same projected line.	13
2.10	The Penrose triangle [PP58].	13
2.11	An example of how non-local inference changes the perception of 3D shape [BT81].	14
2.12	Examples of sketch-based modeling of a 3D character with texture-mapped renderings of the resulting mesh [KC06].	14
2.13	The visual perception pipeline from 3D triangle mesh to Gauge Space.	15
		69

2.14	Rendering with the bas-relief ambiguity effect. (a) and (b) show the normal undistorted case. If the head is deformed under a bas-relief transformation (c), it still appears to have the same geometry (d). Only the unadjusted albedo and changing lighting conditions indicate a visual change between (b) and (c).	17
2.15	Example stimulus for the relative depth probe task. The blue dot should be perceived as nearer to the viewer than the red one.	19
2.16	An example of the depth-profile adjustment mask technique.	19
2.17	Examples of a perfectly aligned gauge figure (a) and a badly adjusted one (b).	20
2.18	Examples of possible slant and tilt combinations of a normal at different points on the hemisphere. Slant and tilt are both continuous [vK04].	21
2.19	Illustration of the original gauge figure task design with its four stages [Wij12].	21
3.1	Four examples of a table cloth model in different rendering styles used in the experiments conducted by Cole et al. [CSD ⁺ 09].	25
4.1	The technical pipeline and its intermediate results.	28
5.1	On the left side, the Stanford bunny is rendered with a CPU-based suggestive contours method with equal width lines. On the right, the lines are also stylized with a fading effect. Both images are rendered with the “rtsc” suggestive contours viewer by Szymon Rusinkiewicz and Doug DeCarlo.	32
5.2	On the left, the original diffuse-shaded object is shown, on the right the resulting suggestive contours rendering after applying a radial filter in image space.	35
6.1	A rendered scene presenting the four stimulus objects used in the user study.	40
6.2	The five levels of detail for the four stimulus object.	41
6.3	The start screen explaining the gauge figure task to the participants.	43
6.4	Example screen from the gauge figure task for the armadillo object at level one.	44
6.5	Snapshots of a rotating gauge figure from a front facing orientation (a) to a back facing one (c) due to perspective distortions.	44
7.1	The percentage of wrong tilts over all defined levels of detail for each of the four stimulus objects.	46
7.2	Quadratic approximation of slant error for each of the four stimulus objects and their aggregate. All five approximations show a trend of lower median slant error, i.e., higher accuracy, with increasing level of detail. For better understanding, the bunny and the heptoroid stimulus images are depicted for every level.	48
7.3	Quadratic approximation of slant standard deviation for each of the four stimulus objects and their aggregate. For better understanding, the bunny and the heptoroid stimulus images are depicted for every level.	49
7.4	Median error of slant estimates over all levels for each stimulus object.	49
7.5	Boxplot of slant error distributions.	50

7.6	Polar plots of the four stimulus objects over all levels showing the median slant error per sector of tilt and slant combination.	51
7.7	Quadratic model fitting the 18 inverse median slant errors of the participants over all levels.	52
7.8	Different curve types fitted to the median of slant error.	54
7.9	This overview show the 20 histograms of best fitting local slant average for all stimulus object and level combination. In addition, the summarized histogram per object and per level are included. The bottom-right histogram shows the distribution of best fitting local area of estimation over all collected data points.	55
7.10	Cubic approximations of the error surfaces for differently sized local neighborhoods.	57
7.11	Approximations of the slant error in regards to percentage of ink over all levels for differently sized local neighborhoods. The red spline indicates the approximation containing the global minimum.	58
7.12	Visualization of the optimized armadillo stimulus object and the differences to the five levels of detail. Red indicates a removal of lines by the optimization routine and blue shows additions.	60
7.13	Visualization of the optimized suggestive contours images for all four stimulus images and the differences to the five levels of detail. Blue indicates additions of lines and red indicates removals of lines.	61

List of Tables

7.1	Percentages of wrongly adjusted tilts for all object and level of detail combinations.	47
7.2	Model characteristics when fitting to the median of slant errors per level.	53

Glossary

accuracy Accuracy refers to the closeness of a measured value to a standard or known value, i.e., the ground truth. If for example in reality it is 30.0 C outside and a temperature sensor reads 30.1 C, then that sensor is relatively accurate. Accuracy is independent of precision. 23, 24, 47, 48, 50–52, 70

AIC-score The Akaike information criterion (AIC) is an estimator of the relative quality of statistical models and can be used for relative model selection, with lower AIC scores indicating a better model. 52, 53, 56

Amazon Mechanical Turk A crowdsourcing marketplace to coordinate human labor to perform (mini-)tasks. 24

CPU Central Processing Unit. 32, 34, 70

curvature Describes the amount by which a geometric object deviates from being flat. 6, 9–12, 33, 34, 36, 69

fragment shader A piece of code in a programmable GPU-pipeline that works on fragments and creates the final pixel output. 36

GLM OpenGL Mathematics (GLM) is a C++ mathematics library for graphics software. 36

GPU Graphical Processing Unit. 6, 32, 33

ground truth Refers to the absolute truth of something, e.g., the actual surface normal. 21, 24, 28, 29, 35–37, 41, 46, 50, 54

hatching A shading effect created by drawing closely spaced lines. 18

motion parallax A depth cue that is the result of an individual's motion. As we move, closer objects move further across the field of view than more distant objects. 18

OpenCV Open source computer vision library containing functions for real-time computer vision tasks. 36

OpenGL Open Graphics Library: a cross-language, cross-platform API for rendering vector graphics. 35, 36

p-value The probability that, when the null hypothesis is true, the sample mean difference between two groups is greater than or equal to the actual observed results. 52

precision Precision refers to the closeness of two or more measurements to each other. If for example you measure the temperature ten times and get 30.0 C each time, then the measurement is very precise. Precision is independent of accuracy. 47, 51

Stereoscopy A technique for creating an illusion of depth for binocular vision. 18

surface normal In the three-dimensional case, a surface normal to a surface at a point P is a vector that is perpendicular to the tangent plane to that surface at P. 8, 9, 12, 20, 29, 37

vertex shader A piece of code in a programmable GPU-pipeline that works on vertices. 32

Weber-Fechner law A law describing the relation between a change in a physical stimulus and the perceived change. 17

Bibliography

- [App67] Arthur Appel. The notion of quantitative invisibility and the machine rendering of solids. In *Proceedings of the 1967 22nd national conference*, pages 387–393. ACM, 1967.
- [Bae10] Jeroen Baert. Contourlijnextractie en weergave in interactieve 3d-toepassingen. Master’s thesis, Katholieke Universiteit Leuven - KU Leuven, Oude Markt 13 - bus 5005, 3000 Leuven, Belgium, 2010.
- [BKY97] P.N. Belhumeur, D. Kriegman, and A.L. Yuille. The bas-relief ambiguity. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1060–1066, Jun 1997.
- [Bli77] James F Blinn. Models of light reflection for computer synthesized pictures. In *ACM SIGGRAPH Computer Graphics*, volume 11, pages 192–198. ACM, 1977.
- [Bou70] W Jack Bouknight. A procedure for generation of three-dimensional half-toned computer graphics presentations. *Communications of the ACM*, 13(9):527–536, 1970.
- [Bre65] Jack E Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems journal*, 4(1):25–30, 1965.
- [BRV⁺10] Stefan Bruckner, Peter Rautek, Ivan Viola, Mike Roberts, Mario Costa Sousa, and M Eduard Gröller. Hybrid visibility compositing and masking for illustrative rendering. *Computers & Graphics*, 34(4):361–369, 2010.
- [BT81] Harry G Barrow and Jay M Tenenbaum. Interpreting line drawings as three-dimensional surfaces. *Artificial intelligence*, 17(1-3):75–116, 1981.
- [BWP⁺16] Matthias Bernhard, Manuela Waldner, Pascal Plank, Veronika Šoltészová, and Ivan Viola. The accuracy of gauge-figure tasks in monoscopic and stereo displays. *IEEE Computer Graphics and Applications*, 36(4):56–66, July 2016.
- [Can87] John Canny. A computational approach to edge detection. In *Readings in Computer Vision*, pages 184–203. Elsevier, 1987.

- [CDF⁺06] Forrester Cole, Douglas DeCarlo, Adam Finkelstein, Kenrick Kin, R Keith Morley, and Anthony Santella. Directing gaze in 3d models with stylized focus. *Rendering Techniques*, 2006:17th, 2006.
- [CGL⁺08] Forrester Cole, Aleksey Golovinskiy, Alex Limpaecher, Heather Stoddart Barros, Adam Finkelstein, Thomas Funkhouser, and Szymon Rusinkiewicz. Where do people draw lines? In *ACM Transactions on Graphics (TOG)*, volume 27, page 88. ACM, 2008.
- [CK17] Maxime Curioni and Tamito Kajiyama. Benchmark tests of freestyle suggestive contours. online, January 2017. available at: <https://freestyleintegration.wordpress.com/2017/01/22/benchmark-tests-of-freestyle-suggestive-contours/> [accessed on 15.06.2018].
- [CSD⁺09] Forrester Cole, Kevin Sanik, Doug DeCarlo, Adam Finkelstein, Thomas Funkhouser, Szymon Rusinkiewicz, and Manish Singh. How well do line drawings depict shape? In *ACM Transactions on Graphics (ToG)*, volume 28-3, page 28. ACM, 2009.
- [DC16] Manfredo P Do Carmo. *Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition*. Courier Dover Publications, 2016.
- [DFRS03] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. Suggestive contours for conveying shape. *ACM Transactions on Graphics (TOG)*, 22(3):848–855, 2003.
- [DHEN95] Erik De Haan, Roderik GF Erens, and AndréJ Noest. Shape from shaded random surfaces. *Vision research*, 35(21):2985–3001, 1995.
- [DR07] Doug DeCarlo and Szymon Rusinkiewicz. Highlight lines for conveying shape. In *International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, August 2007.
- [DS00] Oliver Deussen and Thomas Strothotte. Computer-generated pen-and-ink illustration of trees. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 13–18. ACM Press/Addison-Wesley Publishing Co., 2000.
- [DWE02] Joachim Diepstraten, Daniel Weiskopf, and Thomas Ertl. Transparency in interactive technical illustrations. In *Computer Graphics Forum*, volume 21, pages 317–325. Wiley Online Library, 2002.
- [DWE03] Joachim Diepstraten, Daniel Weiskopf, and Thomas Ertl. Interactive cutaway illustrations. In *Computer Graphics Forum*, volume 22, pages 523–532. Wiley Online Library, 2003.
- [Gal69] R Galimberti. An algorithm for hidden line elimination. *Communications of the ACM*, 12(4):206–211, 1969.

- [GGSC98] 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*, pages 447–452. ACM, 1998.
- [GIHL00] Ahna Girshick, Victoria Interrante, Steven Haker, and Todd Lemoine. Line direction matters: an argument for the use of principal directions in 3d line drawings. In *Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 43–52. ACM, 2000.
- [GRG04] Bruce Gooch, Erik Reinhard, and Amy Gooch. Human facial illustrations: Creation and psychophysical evaluation. *ACM Transactions on Graphics (TOG)*, 23(1):27–44, 2004.
- [Hor82] Christoph Hornung. An approach to a calculation-minimized hidden line algorithm. *Computers & Graphics*, 6(3):121–126, 1982.
- [How12] Ian P Howard. *Perceiving in depth, volume 1: basic mechanisms*. Oxford University Press, 2012.
- [HR12] Ian P Howard and Brian J Rogers. *Perceiving in depth, Vol. 2: Stereoscopic vision*. Oxford University Press, 2012.
- [HZ00] Aaron Hertzmann and Denis Zorin. Illustrating smooth surfaces. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 517–526. ACM Press/Addison-Wesley Publishing Co., 2000.
- [IDSC04] Shoukat Islam, Swapnil Dipankar, Deborah Silver, and Min Chen. Spatial and temporal splitting of scalar fields in volume graphics. In *Volume Visualization and Graphics, 2004 IEEE Symposium on*, pages 87–94. IEEE, 2004.
- [IFP95] Victoria Interrante, Henry Fuchs, and Stephen Pizer. Enhancing transparent skin surfaces with ridge and valley lines. In *Proceedings of the 6th conference on Visualization'95*, page 52. IEEE Computer Society, 1995.
- [JDA07] Tilke Judd, Frédo Durand, and Edward Adelson. Apparent ridges for line drawing. In *ACM Transactions on Graphics (TOG)*, volume 26, page 19. ACM, 2007.
- [KC06] Matthew Kaplan and Elaine Cohen. Producing models from drawings of curved surfaces. In *SBM*, pages 51–58, 2006.
- [KDMF03] Robert D Kalnins, Philip L Davidson, Lee Markosian, and Adam Finkelstein. Coherent stylized silhouettes. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 856–861. ACM, 2003.

- [KMS⁺06] Kristin Koch, Judith McLean, Ronen Segev, Michael A. Freed, Michael J. II Berry, Vijay Balasubramanian, and Peter Sterling. How much the eye tells the brain. *Current Biology*, 16:1428 – 1434, 2006.
- [Koe29] Wolfgang Koehler. *Gestalt Psychology*. New York: Liveright, 1929.
- [KST08] Michael Kolomenkin, Ilan Shimshoni, and Ayellet Tal. Demarcating curves for shape illustration. In *ACM Transactions on Graphics (TOG)*, volume 27-5, page 157. ACM, 2008.
- [KvDK92] Jan J. Koenderink, Andrea J. van Doorn, and Astrid M. L. Kappers. Surface perception in pictures. *Perception and Psychophysics*, 5(52):487–496, 1992.
- [KvDKT01] Jan J. Koenderink, Andrea J. van Doorn, Astrid M. L. Kappers, and James T. Todd. Ambiguity and the 'mental eye' in pictorial relief. *Perception.*, 30(4):431–448, Jun 2001.
- [KW07] Chung-Chia Kang and Wen-June Wang. A novel edge detection method based on the maximizing objective function. *Pattern Recognition*, 40(2):609–618, 2007.
- [KWTM03] Gordon Kindlmann, Ross Whitaker, Tolga Tasdizen, and Torsten Moller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Visualization, 2003. VIS 2003. IEEE*, pages 513–520. IEEE, 2003.
- [LC95] Gordon E Legge and Patrick Cavanagh. Pictorial depth cues: a new slant. *J. Opt. Soc. Am. A*, 12(14), 1995.
- [LME⁺02] Aidong Lu, Christopher J Morris, David S Ebert, Penny Rheingans, and Charles Hansen. Non-photorealistic volume rendering using stippling techniques. In *Proceedings of the conference on Visualization'02*, pages 211–218. IEEE Computer Society, 2002.
- [LMLH07] Yunjin Lee, Lee Markosian, Seungyong Lee, and John F Hughes. Line drawings via abstracted shading. *ACM Transactions on Graphics (TOG)*, 26(3):18, 2007.
- [Lou70] Philippe Paul Loutrel. A solution to the hidden-line problem for computer-drawn polyhedra. *IEEE Transactions on Computers*, 100(3):205–213, 1970.
- [LP16] Kai Lawonn and Bernhard Preim. Feature lines for illustrating medical surface models: Mathematical background and survey. In *Visualization in Medicine and Life Sciences III*, pages 93–131. Springer, 2016.
- [Mac63] DM MacKay. Psychophysics of perceived intensity: A theoretical basis for fechner's and stevens' laws. *Science*, 139(3560):1213–1216, 1963.

- [Mal87] Jitendra Malik. Interpreting line drawings of curved objects. *International Journal of Computer Vision*, 1(1):73–103, 1987.
- [MK96] Pascal Mamassian and Daniel Kersten. Illumination, shading and the perception of local orientation. *Vision research*, 36(15):2351–2367, 1996.
- [MKG⁺97] Lee Markosian, Michael A Kowalski, Daniel Goldstein, Samuel J Trychin, John F Hughes, and Lubomir D Bourdev. Real-time nonphotorealistic rendering. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 415–420. ACM Press/Addison-Wesley Publishing Co., 1997.
- [MMK⁺00] Lee Markosian, Barbara J Meier, Michael A Kowalski, Loring S Holden, JD Northrup, and John F Hughes. Art-based rendering with continuous levels of detail. In *Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 59–66. ACM, 2000.
- [Nik09] Danko Nikolic. Is synaesthesia actually ideaesthesia? an inquiry into the nature of the phenomenon. In *Proceedings of the Third International Congress on Synaesthesia*. Science and Art, Granada, Spain, 2009.
- [NM00] JD Northrup and Lee Markosian. Artistic silhouettes: A hybrid approach. In *Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 31–37. ACM, 2000.
- [NW07] J Farley Norman and Elizabeth Y Wiesemann. Aging and the perception of local surface orientation from optical patterns of shading and specular highlights. *Perception & Psychophysics*, 69(1):23–31, 2007.
- [OBS04] Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. Ridge-valley lines on meshes via implicit surface fitting. In *ACM transactions on graphics (TOG)*, volume 23-3, pages 609–612. ACM, 2004.
- [Oko12] Takanori Okoshi. *Three-dimensional imaging techniques*. Elsevier, 2012.
- [PB13] Bernhard Preim and Charl P Botha. *Visual computing for medicine: theory, algorithms, and applications*. Newnes, 2013.
- [PHWF01] Emil Praun, Hugues Hoppe, Matthew Webb, and Adam Finkelstein. Real-time hatching. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, page 581. ACM, 2001.
- [PKG03] Mark Pauly, Richard Keiser, and Markus Gross. Multi-scale feature extraction on point-sampled surfaces. In *Computer graphics forum*, volume 22-3, pages 281–289. Wiley Online Library, 2003.

- [Pla15] Pascal Plank. Human visual perception of 3d surfaces. Master’s thesis, Institute of Computer Graphics and Algorithms, Vienna, University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, June 2015.
- [PP58] Lionel S Penrose and Roger Penrose. Impossible objects: A special type of visual illusion. *British Journal of Psychology*, 49(1):31–33, 1958.
- [RBD06] Szymon Rusinkiewicz, Michael Burns, and Doug DeCarlo. Exaggerated shading for depicting shape and detail. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 1199–1205. ACM, 2006.
- [RCDF08] Szymon Rusinkiewicz, Forrester Cole, Doug DeCarlo, and Adam Finkelstein. Siggraph 2008-class: Line drawings from 3d models. online, August 2008. available at: <http://gfx.cs.princeton.edu/proj/sg08lines/> [accessed on 05.06.2018].
- [Rus04] Szymon Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pages 486–493. IEEE, 2004.
- [SD04] Anthony Santella and Doug DeCarlo. Visual interest and npr: an evaluation and manifesto. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, pages 71–150. ACM, 2004.
- [SF68] Irwin Sobel and Gary Feldman. A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in*, pages 271–272, 1968.
- [SMGG01] Peter-Pike J Sloan, William Martin, Amy Gooch, and Bruce Gooch. The lit sphere: A model for capturing npr shading from art. In *Graphics interface*, volume 2001, pages 143–150, 2001.
- [SPV11] Veronika Solteszova, Daniel Patel, and Ivan Viola. Chromatic shadows for improved perception. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering, NPAR ’11*, pages 105–116, New York, NY, USA, 2011. ACM.
- [Sta78] JE Staddon. Theory of behavioral power functions. *Psychological Review*, 85(4):305, 1978.
- [Ste83a] Kent A. Stevens. Slant-tilt: The visual encoding of surface orientation. *Biological Cybernetics*, 46(3):183–195, 1983.
- [Ste83b] Kent A. Stevens. Surface tilt (the direction of slant): A neglected psychophysical variable. *Perception and Psychophysikcs*, 33(3):241 – 250, 1983.

- [STPV12] V. Solteszova, C. Turkay, M.C. Price, and I. Viola. A perceptual-statistics shading model. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2265–2274, Dec 2012.
- [Str74] Wolfgang Straßer. *Schnelle kurven-und flächendarstellung auf grafischen sichtgeräten*. PhD thesis, Technical University Berlin, 1974.
- [TG96] Jean-Philippe Thirion and Alexis Gourdon. The 3d marching lines algorithm. *Graphical Models and Image Processing*, 58(6):503–509, 1996.
- [The96] Holger Theisel. On geometric continuity of isophotes. In *Proceedings of Chamomix*, pages 1–8, 1996.
- [Tod04] James T. Todd. The visual perception of 3d shape. *Trends in Cognitive Sciences*, 8(3):115–121, Mar 2004.
- [Tuf86] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1986.
- [Tuf90] Edward Tufte. *Envisioning Information*. Graphics Press, Cheshire, CT, USA, 1990.
- [vK04] Joost van Kasteren. Retaining form in two dimensions: Depth - the missing dimension - is inside the human head, 2004.
- [VKG05] Ivan Viola, Armin Kanitsar, and M Eduard Groller. Importance-driven feature enhancement in volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):408–418, 2005.
- [Wal75] David Waltz. Understanding line drawings of scenes with shadows." the psychology of computer vision. patrick henry winston, ed, 1975.
- [Wat70] Gary S Watkins. A real time visible surface algorithm. Technical report, UTAH UNIV SALT LAKE CITY SCHOOL OF COMPUTING, 1970.
- [WBC⁺07] Christian Wallraven, Heinrich H Bülthoff, Douglas W Cunningham, Jan Fischer, and Dirk Bartz. Evaluation of real-world and computer-generated stylized facial expressions. *ACM Transactions on Applied Perception (TAP)*, 4(3):16, 2007.
- [WFGS07] Holger Winnemöller, David Feng, Bruce Gooch, and Satoru Suzuki. Using npr to evaluate perceptual shape cues in dynamic environments. In *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, pages 85–92. ACM, 2007.
- [WGK10] Matthew Ward, Georges Grinstein, and Daniel Keim. *Human Perception and Information Processing in Interactive Data Visualization: Foundations, Techniques, and Application*. A K Peters, 2010.

- [Wij12] Maarten W.A. Wijntjes. Probing pictorial relief: from experimental design to surface reconstruction. *Behavior Research Methods*, 44(1):135–143, 2012.
- [Woo06] Simon N Wood. *Generalized additive models: an introduction with R*. Chapman and Hall/CRC, 2006.
- [WREE67] Chris Wylie, Gordon Romney, David Evans, and Alan Erdahl. Half-tone perspective drawings by computer. In *Proceedings of the November 14-16, 1967, fall joint computer conference*, pages 49–58. ACM, 1967.
- [WST01] Joseph B Walther, Celeste L Slovacek, and Lisa C Tidwell. Is a picture worth a thousand words? photographic images in long-term and short-term computer-mediated communication. *Communication Research*, 28(1):105–134, 2001.
- [ZLC95] G Lee Zimmerman, Gordon E Legge, and Patrick Cavanagh. Pictorial depth cues: A new slant. *JOSA A*, 12(1):17–26, 1995.