Die approbierte Originalversion dieser Dissertation ist an der Hauptbibliothek der Technischen Universität Wien aufgestellt (http://www.ub.tuwien.ac.at).

The approved original version of this thesis is available at the main library of the Vienna University of Technology (http://www.ub.tuwien.ac.at/englweb/).

Dissertation

Image Stylization from Stereo Views of Natural Scenes

ausgeführt
zum Zwecke der Erlangung des akademischen Grades eines
Doktors der technischen Wissenschaften

unter der Leitung von
Ao. Univ. Prof. Dipl.-Ing. Mag. rer. nat. Dr. techn. Margrit Gelautz
Institut 188 für Software Technologie und Interaktive Systeme

eingereicht
an der Technischen Universität Wien
Fakultät für Informatik

von
Dipl.-Ing. Danijela Marković
Arndtstrasse 56/2/8
1120 Wien
Matr.-Nr. 0227186

Wien, im Februar 2007

Abstract

As media technologies have developed over time, generating novel methods of creating, presenting and interacting with visual information, artists began experimenting with new concepts and possibilities by producing artworks which moved conventional borders of art expression. New media technologies have introduced new opportunities for artistic creation and have brought the computer created design and artwork to a wide spread instrument of visual communication.

These recent trends in artistic production and presentation of visual material consequently stimulated the development of novel techniques in the field of computer graphics. Particularly, the (semi-)automatic creation of output that resembles conventional art forms, such as various styles of drawings or paintings produced by hand, has attracted a lot of attention in the computer graphics community. To generate results with a handcrafted-like appearance, a number of systems use a 3D modeled scene and thus benefit from full control over the scene structure. On the other hand, using images of natural scenes in order to compute their stylized representation is a very challenging task. The process of extracting information that is useful for the subsequent stylization from the real input scene can be quite complex and demanding, mostly because of the presence of noise and other artifacts. In spite of this, natural scenes offer a variety of ready-made shapes, objects and textures, which sets them to a powerful and very attractive alternative to 3D modeled scenes for simulating the traditional and exploring new art forms.

The focus of this thesis is the stylization of images from real scenes captured using a stereo camera set-up. An essential feature of the depiction process is a novel silhouette extraction approach that we introduce here. This so-called Edge Combination algorithm utilizes stereo-derived disparity maps in order to find silhouettes, which delineate the dominant structure of the scene. Our research results demonstrate that silhouettes extracted using the Edge Combination algorithm are suitable and convenient for image depiction and stylization in various simulations of the traditional artwork, such as sketching and drawing. We also illustrate the benefit of utilizing disparity maps to orient strokes to facilitate the "depth" perception in drawn-like image representations. Finally, we show that silhouettes extracted by the Edge Combination algorithm can serve for motion delineation in imitation of comics produced by hand and also to portray the impression of change in a moving object's speed.

Kurzfassung

Ausgelöst durch die kontinuierliche Weiterentwicklung von Medientechnologien in Richtung neuer Methoden zur Generierung und Präsentation visueller Information, einschließlich zugehöriger Möglichkeiten zur Interaktion, haben Künstler begonnen, mit neuen Konzepten und Möglichkeiten zu experimentieren. Dabei wurden Kunstwerke geschaffen, welche die konventionellen Grenzen des künstlerischen Ausdrucks verschieben. Moderne Medientechnologien haben damit neuartige Möglichkeiten zur Generierung künstlerischen Inhalts eröffnet und computergenerierte Kunstwerke und Darstellungen zu einem weit verbreiteten Instrument der visuellen Kommunikation gemacht.

Diese gegenwärtigen Trends in der künstlerischen Erzeugung und Präsentation visueller Inhalte stimulierten zugehörige Entwicklungen in der Computergrafik. Ein besonderes Augenmerk liegt dabei auf der möglichst automatischen Erzeugung von Bildern, die traditionellen Kunstwerken wie händischen Zeichnungen und Malereien nachempfunden sind. Um solche Ergebnisse zu generieren, verwenden derzeitige Systeme üblicherweise ein vorhandenes 3D-Modell der Szene, welches dem Bearbeiter vollständige Kontrolle über die Szenenstruktur ermöglicht. Im Gegensatz dazu stellt die Verwendung von echten Bildaufnahmen natürlicher Szenen für die nachfolgende Stilisierung eine große Herausforderung dar, insbesondere auch wegen Störfaktoren wie Bildrauschen und anderer Artefakte. Natürliche Szenen bieten jedoch ein breites Spektrum an vorhandenen Formen, Objekten und Texturen, was sie zu einer mächtigen und äußerst attraktiven Alternative zur traditionellen 3D-Modellierung als Ausgangspunkt für die Simulation konventioneller und die Entwicklung neuer Kunstformen macht.

Der Fokus dieser Arbeit liegt auf der Stilisierung von Bildern realer Szenen, welche mit einem Stereo-Kamera-Setup aufgenommen wurden. Eine wesentliche Neuheit unseres Darstellungsprozesses ist ein Ansatz zur Silhouettenextraktion, den wir in dieser Arbeit beschreiben. Der so genannte Edge-Combination-Algorithmus verwendet dabei Disparitätskarten, welche durch Stereoauswertung berechnet wurden, um Silhouetten zu finden. Die extrahierten Silhouetten stellen die dominante Struktur der Szene dar. Unsere Forschungsergebnisse zeigen, dass mittels des Edge-Combination-Algorithmus berechnete Silhouetten gut geeignet sind, um eine Bilddarstellung und Stilisierung zu erzeugen, welche verschiedene traditionelle Kunstformen wie künstlerische Skizzen und Gemälde simulieren. Des Weiteren demonstrieren wir, wie Disparitätskarten dazu verwendet werden können, um Striche zu orientieren, was die Tiefenwahrnehmung in Bildrepräsentationen, welche an verschiedene Zeichen- und Malstile angelehnt sind, unterstützt. Schließlich zeigen wir, dass Silhouetten, welche durch den Edge-Combination-Algorithmus extrahiert wurden, dazu dienen können, eine Bewegungsdarstellung im Stile handgezeichneter Comics zu erzeugen, wobei dem Betrachter auch ein Eindruck von der Geschwindigkeitsänderung eines sich bewegenden Objektes vermittelt wird.

Acknowledgements

I consider myself a fortunate person because somehow, throughout my life, I have always been surrounded by wonderful, extraordinary people. Each of whom have been a star that has enlightened my small universe. This document would not at all exist without them.

First and foremost, I am indebted to my advisor Margrit Gelautz for her patience, guidance and for all the freedom that I had in my research. She led me through the most important steps of research, writing and publishing.

I would like to extend deepest appreciation to Christian Breiteneder for all his support and for providing the opportunity to work in such an excellent research environment as the IMS group.

I am thankful to Gerti Kappel for financially supporting part of my work and to Ulrike Pastner for all the help and amicable chats.

Particularly, I would like to thank Felix Breitenecker for accepting to be the second reviewer of this thesis and for providing valuable feedback that helped to shape its final form.

A great deal of thanks goes to my dear colleagues, Michael Bleyer and Efstathios Stavrakis, for endless talks, discussions and chats. The work described in the Chapter 5 has been devised in collaboration with Efstathios Stavrakis. A preliminary version of it was published under the title "Parameterized Sketches from Stereo Images". For more publication details see Section 1.6.

I would like to also thank other members of the IMS group, especially Karyn Laudisi for all the time she spent reading my papers.

I am grateful to my family, my parents Dragica and Ljubiša Marković; my grandparents Jagoda and Stojan Ilić; and my little brother Milan Marković, for enormous love, endless support, for believing in me and encouraging me.

I would also like to thank Dejan Ristić who was with me all these years, giving me strength to go for my dreams.

Finally, I wish to thank all my friends for helping and for being a friend.

This research has been funded by the Austrian Science Fund (FWF) (project no. P15663) and by the Austrian Federal Ministry for Education, Science, and Culture, as well as the European Social Fund (ESF) under grant 31.963/46-VII/9/2002.

Posvećeno mojim dragim roditeljima za svu podarenu podršku, ljubav i pažnju.

Contents

Abstract **i**

Kurzfassung **iii**

Acknowledgements **v**

Contents **viii**

1 Introduction **1**
- 1.1 Art: The Visual Ensemble 1
- 1.2 Arts and Technology 3
- 1.3 Digital Art: Man vs. Machine 4
- 1.4 Centerpieces of Real World Delineation 7
  - 1.4.1 Silhouettes 7
  - 1.4.2 Shape Depiction by Shading 10
- 1.5 Contributions and Organization 13
- 1.6 Publications 16

2 Related Work **18**
- 2.1 Introduction 18
  - 2.1.1 Photorealistic Rendering 18
  - 2.1.2 Non-Photorealistic Rendering 20
- 2.2 Silhouettes 23
  - 2.2.1 Modeled Scene 23
  - 2.2.2 Natural Scene 26

2.3 Drawn-like Depiction 28

3 Background **33**

3.1 Stereo Vision 33

3.1.1 Human Vision and Stereoscopy 33

3.1.2 Machine Stereo Vision 34

3.2 Active Contours 41

3.2.1 Introduction 41

3.2.2 Fundamentals - Traditional Snake 43

3.2.3 Aspects of Snake Behavior 46

3.2.4 Gradient Vector Flow Snake 47

3.3 Bézier Curves 50

4 Edge Combination Algorithm **54**

4.1 Outline of the Algorithm 55

4.2 Edge Detection 55

4.3 Edge Search 56

4.4 Edge Linking 57

4.5 Cleaning (Maze Solving Technique) 58

4.6 Results 58

4.7 Application to Object Segmentation 60

4.7.1 Motivation 60

4.7.2 Experiment 64

4.7.3 Object Segmentation Results 65

4.8 Summary 65

5 Sketching by Parameterized Silhouettes **69**

5.1 Introduction 69

5.1.1 Motivation and Overview of Algorithm 69

5.1.2 Raster-to-Vector Conversion 71

5.2 Converting Raster Lines to Bézier Curves 72

5.2.1 Initialization 73

5.2.2 Creating Sketchy Line Segments 73
5.2.3 Initial Control Points 75
5.2.4 Curve Approximation 76
5.3 Line Stylization 77
5.4 Experimental Results 78
5.5 Summary 79

6 Drawn-like Image Representation 84
6.1 Introduction 84
6.2 Algorithm 85
6.2.1 Image Segmentation 86
6.2.2 Stroke Arrangement 86
6.2.3 Stroke Orientation 88
6.2.4 Outlining 89
6.3 Results 90
6.4 Summary 92

7 Comics-like Motion Depiction 99
7.1 Introduction 99
7.2 Algorithm 101
7.2.1 Drawn-like Image Representation 101
7.2.2 Tracking Motion 103
7.2.3 Depicting Motion 104
7.3 Experimental Results 107
7.4 Summary and Outlook 108

8 Summary and Future Work 113

Bibliography 117

List of Figures 132

Chapter 1

Introduction

1.1 Art: The Visual Ensemble

I never saw an ugly thing in my life: for let the form of an object be what it may - light, shade, and perspective will always make it beautiful.
- John Constable (1776-1837).

Art - the magical form of expression that is inwrought in our life, revealing its power by a delicate presence. It spreads out from book shelves, theaters, museums, concert halls, boutiques, radio and television, cinemas, restaurants and perfumeries. It enriches our life, brings beauty, new ideas and a variety of emotions. An artwork, as a “creative impulse”, has the power to calm or incite to action. It is a force that can refine, persuade, influence or mystify. Art, overall, has an incredible potential to heal, unite, inspire and provoke.

By going deeper to the visual aspect of the thrilling force of art, we find images that surround us in everyday life, coming from a variety of media: drawings, paintings, prints, photos, newspapers and magazines, billboards, movies, videos, games, etc. But, what is visual art? Or, maybe, the better question is - what is visual art for you? Some basic descriptions that can be found in encyclopedias are:

Art, also called visual art: a visual object or experience consciously created through an expression of skill or imagination. The term *art* encompasses diverse media such as painting, sculpture, printmaking, drawing, decorative arts, photography, and installation [5].

Art: the product of creative human activity in which materials are shaped or selected to convey an idea, emotion, or visually interesting form. The word *art* can refer to the visual arts, including painting, sculpture, architecture, photography, decorative arts, crafts, and other visual works that combine materials or forms [6].

More sophisticated and advanced defining and categorizing of the art forms falls within the domain of the philosophy of the visual arts. Although various definitions have been proposed, visual art remains to be an ongoing source of debate with a lot of open questions and arguments. As it is a rapidly growing and constantly transforming area, defining and classifying is extremely hard.

Visual art is, generally, a class of artworks which are principally visual in nature, classified usually into fine arts on one side and craftwork on the other side. The term of fine art or high art encompasses a wide variety of human creativity expressions, usually created by utilizing developed techniques and skills with a lot of care, devotion and imagination to provide a high level of longevity. *"Traditionally, high art consists of the meticulous expression in fine materials of refined or noble sentiment, appreciation of the former depending on such things as intelligence, social standing, educated taste, and a willingness to be challenged"* [35]. On the other side, the craftwork is considered as handmade creation that has utility in everyday life use such as ceramics, furniture, glass, jewellery, metal, textiles, wood or puppetry. A more detailed classification list of visual arts can be found in "On the Nature of the Visual Arts" by Thomas Munro [4].

In art as a constantly changing, growing and modifying area, which is tightly connected to the human creativity and knowledge, borders between classes became blurred. The perseverance of artists in experimenting, breaking rules, defying conventions and pushing borders forward with new possibilities of new media and new methods of creation is incessantly forming novel techniques and new styles of expression.

1.2 Arts and Technology

Worldwide communication for collecting and exchanging ideas, methods and information is rapidly transforming from conventionally text based to a rich, interactive and image based one. This change, along with the development of technology, promoted an explosion of visual material that emphasized the term "visual culture" [43]. As a result, the visual culture that portrays human development, information distribution and communication has become an important interdisciplinary field of studies and research.

Today, art, information and educational resources are much more available to wider communities and not only accessible by a handful of privileged people as it was the case through history. Novel systems for image creation have increased our means of expression, understanding and imagination. Interactive and visual demonstrations in presenting instructional, educational or business material, also used for distant collaboration and learning, have introduced novel, faster and more comprehensible means of interaction, education and information, which are also easier to modify.

Through the constant change in human "visual culture", creations of the modern technologies and media such as television, motion pictures, advertisements, comic books and computer games endeavor to find a place next to the traditional means of the artistic expression like drawing, painting and sculpture. A new type of creative expression, digital art, which combines the artistic and technical inquiry, has become one of the most exploited forms of artistic creation. Although it has developed into a vital part of modern global culture, it is still not generally considered as a highly valuable art form. Often ignored by art critics, digital art has attracted the attention of designers, educators and the entertainment industry artists.

Major support for the digital Avant-Garde and interactions between Art and Technology are online web resources and communities, digital art museums and galleries that promote the digital artwork. Below we list some of the pioneers who have undertaken the mission to engage the public, educate the community to understand and appreciate digital art and to support artists in the creation, exhibition and promotion of their work.

- “Digital Art Museum” (DAM), the online resource of digital fine art [38],
- “Ars Electronica Center - Museum of the Future” in Linz, Austria,
- “Museum of the Moving Image” in Astoria, New York, USA,
- “Center for Art and Media” in Karlsruhe, Germany,
- “The Austin Museum of Digital Art” (AMODA), Texas, USA.

In Figure 1.1 we give an illustration that shows “Gulliver’s World”, one of the most popular attractions of the “Ars Electronica Center - Museum of the Future” in Linz, Austria. This set-up is a multi-user mixed reality system that offers to users the ability to design their own environment and to modify characters of the artificial world.

1.3 Digital Art: Man vs. Machine

Digital art as an advantageous form for artists to evanesce borders, fusing art and new technologies, has introduced a range and variety of possibilities to create and experiment; particularly, by combining it with various art expressions like sound or dance, as shown in Figure 1.2. In a sense of quality, creativity and artistic expression, “computerization” of the artwork creation

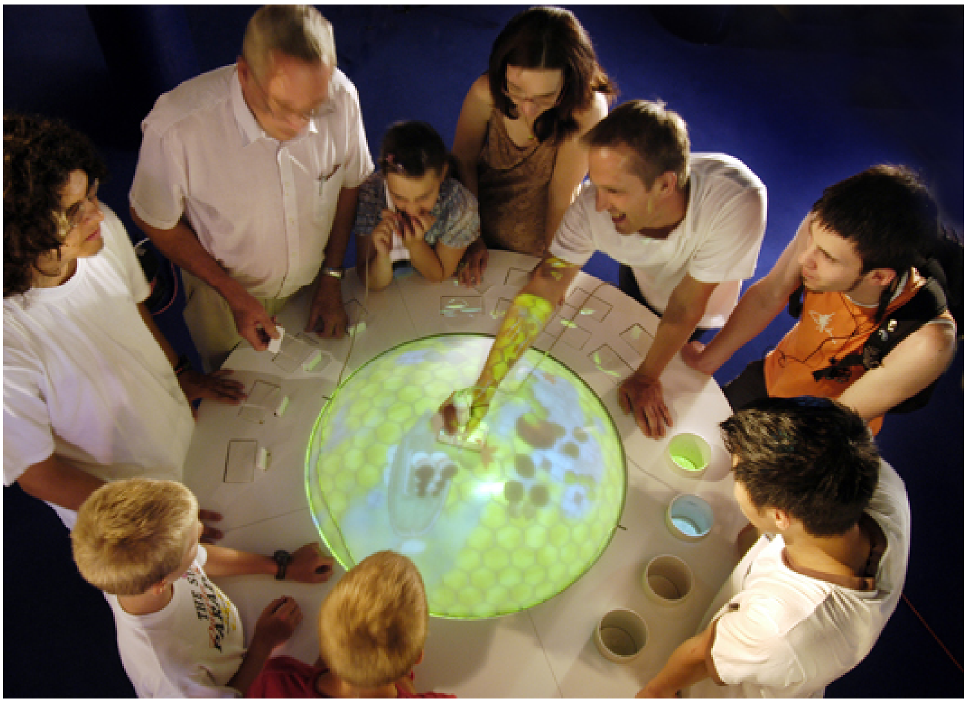


Figure 1.1: Ars Electronica Center Exhibition - “Gullivers World”. © Pilo

process did not change much. New technologies brought novel methods and tools, but the human power to create, imagine and feel still plays a main role. However, novel aspects of creation, such as image creation software, sensors, cameras or screens introduced innovative tools, approaches and methods of the artistic expression beyond the traditional ink, paint, brush and canvas.

For most of the traditional branches of visual art it is necessary to study and develop technical and manual skills. Different to this, computer software offers flexibility in creation and provides even to unexperienced users the opportunity to create interesting image results. Additionally, the necessity of manual dexterity in traditional art creation is suppressed, leaving space for other required elements such as imagination, creativity or image composition. This is particularly obvious in a type of visual art called algorithmic art, in which computer generated art forms such as fractals, genetic or mathematical art are produced completely as output of an algorithm, written by the artist. In this case, the usual manual dexterity required for the art creation is

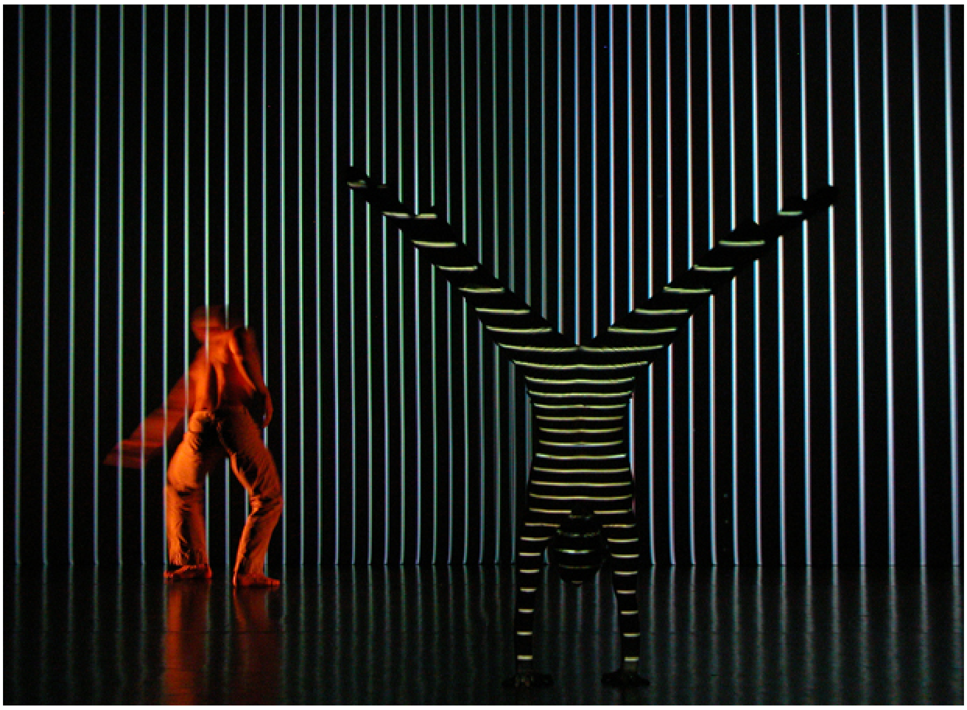


Figure 1.2: Apparition - a dance and media performance for the stage where the dancer interacts with and controls the image and the musical environment. A collaboration between Klaus Obermaier and the Ars Electronica Futurelab © Pascal Maresch

completely replaced by the necessity of good programming and mathematical knowledge. A concise description along with some examples of algorithmic art is presented in [21].

In this thesis we focus on algorithms to stylize images of natural scenes from stereo input. Our goal is to automate the creation and simulate the results of traditional artistic styles such as sketches and drawings. The idea behind the usage of these algorithms is to produce interesting and effective results, similar to handcrafted graphics, from real images of a natural scene by simply tuning parameters. In our research, we concentrate on silhouettes and shading as centerpieces of the human depiction process, as well as on comics-like motion depiction.

1.4 Centerpieces of Real World Delineation

1.4.1 Silhouettes

Silhouette is an essential form to effectively convey a great amount of information in just few strokes. Silhouettes are perceptually and influentially salient features that are part of our everyday life. Exploited by the human vision as the first index into its memory of shapes, silhouettes help us to recognize objects even without necessity of using color, shading or texture information, but only by their contours [61]. This remarkable human understanding of shapes devised only by lines has always been used for artistic work and scientific visualization to efficiently represent the form of objects. Figure 1.3 shows an example of a simple, easily recognizable, delineated form, which is depicted purely through the use of lines.

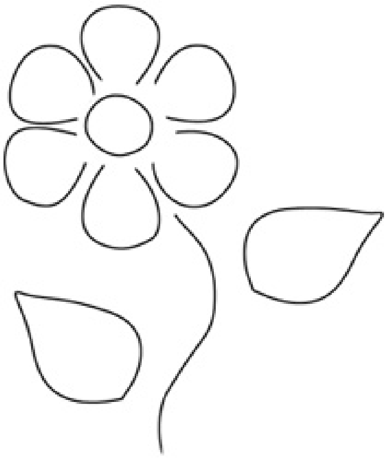


Figure 1.3: Example of form delineation.

The silhouette as a drawing feature has been playing a crucial role in artistic expression from early cave paintings to today. Artists often use silhouettes for depicting basic shapes for further art creation, and as a standalone pure line form in different styles of artwork like sketches or silhouette portraits. This type of portraiture, silhouette portrait, was named by the French finance minister Étienne de Silhouette, whose hobby was the cutting of paper shadow portraits. An example of such silhouette portraiture is shown in Figure 1.4. Referred to as the "poor man's portrait," because of an inexpensive alternative to the more extravagant custom-made oil portraits, silhouette portraits became very popular in the mid-18th-century for all classes of society. Silhouette artists and amateurs would create portraits by using scissors and black paper or by tracing their subject's image with a machine called the camera obscura [42]. Through time methods changed, and the fashion of depicting shapes of friends, family, prominent people or beloved ones established the term silhouette to depict the extracted boundary of an object.

Figure 1.4: Example of a silhouette portrait.

Even today, it is possible to find artists, particularly on streets of large or tourist cities, that instantly cut silhouette portraits using nothing but black paper, scissors and several minutes of the model's time. For example, the photo shown in Figure 1.5 was taken in Paris in 2006.

Playing a very important role in computer science, silhouettes have been used extensively for object segmentation, recognition, tracking or shape reconstruction, as well as for human action recognition or pose estimation, masking and compositing. In computer graphics, silhouette finding and ren-

Figure 1.5: Montmartre, Paris: An artist cuts the silhouette portrait of a tourist.

dering has efficient use to represent the form of objects in technical or medical illustrations, architectural design and in the area of non-photorealistic rendering (NPR).

Mostly, silhouettes are drawn to abstract and denote the distinctive features, or to clearly delineate objects. Since they represent discontinuities in the visibility of an object and those are tightly connected to the depth discontinuities, one of the most valuable source of information for silhouette finding and extraction is depth information. Many techniques use depth information to portray the important features of models in a scene [107, 34, 30, 86, 59]. These approaches, as well as most of the previous work, are focused on the silhouette extraction from a modeled 3D scene, but with thinning and softening the differences between computer vision and graphics methods, a door to exploit the biggest source of 3D models, the real world, has opened. In this sense, we have developed methods to extract silhouettes and to stylize recordings of natural scenes from a stereo image pair in order to depict and explore profound shapes of the real world.

1.4.2 Shape Depiction by Shading

Another extremely expressive form that robustly influences the final result of an art creation is shading. This magical interplay of light and shade gives the objects a live appearance and a sense of solidity. For most pictorial artworks, shading is used to reinforce the feeling of three-dimensionality, to convey roundness, depth, and mass of a depicted subject.

In some styles of depiction - for instance, drawing and print making - the final mixing of colors and tones is visual, the tonal variations of color are produced when the eye "mixes" them. This is different to painting where desired colors are commonly created by physically mixing them. One group of these techniques uses a variety of colors to produce artworks. In this case, the desired colors and tones are usually produced by superimposing layers of a different color shade or different color.

Another group, that we will put more emphasis on, is the group of monochromatic techniques, which essentially use different tones of one color for depiction. The mood of such an artwork and its overall dimensional feel is achieved using different levels of shading tones. Shading tones are formed by applying various shading techniques such as hatching, crosshatching or stippling. Shading techniques use basic strokes organized in textures. Varying tonal values of shading are created from the different stroke densities. This way, shading goes from very light areas with almost no strokes to very dark areas where strokes more or less overlap, forming highlights, midtones and shadows. An example of the shading tone scale produced using the hatching technique is given in Figure 1.6.

A further, very important feature of shading strokes is their orientation. In order to produce the illusion of three dimensions, the feeling of "depth," and to separate objects from each other, shading strokes are typically drawn to follow the shape of a depicted object's surface.

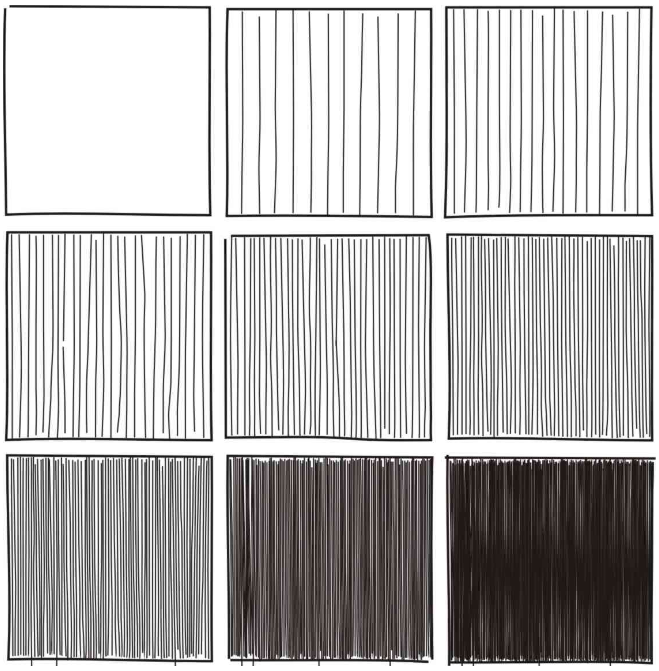


Figure 1.6: Example of a shading tone scale (hatching technique).

Figure 1.7 shows an artwork by Giovanni Battista Piranesi from the mid-18th century. A closer view, which shows a detail of the arch, is given in Figure 1.8. Note the delicate shading strokes, their position, shape, orientation and density that form the depth of the depicted space. Indubitably, the creation of such an artwork requires manual dexterity and artistic skillfulness.

In order to produce a pictorial artwork, it is important to understand how shading is used to depict light variations, to model three-dimensionality, or to add mood and feeling. The artist interacts with the lines on a surface through meticulous consideration of the light variation aspects to delineate delicate shading tone variations in order to convey shapes and forms of the artwork.

In Chapter 6 we present an algorithm for depiction of images of a natural scene in a drawn-like form. In order to attach to the final image result a feeling of "depth", we employed the additional information obtained from the disparity map. We orient shadowing strokes to follow the orientation of

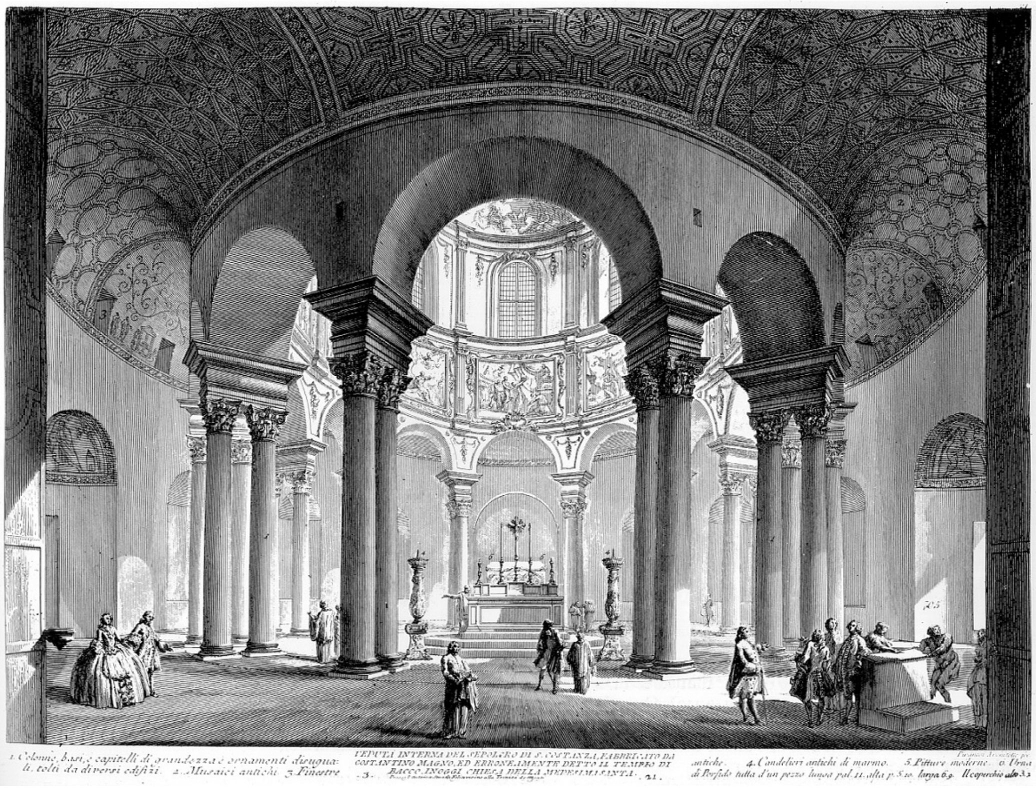

1. Colonne, basi, e capitelli di grandezza e ornamenti disugua: li, tolti da diversi edifizi. 2. Musaici antichi. 3. Finestre
VEDUTA INTERNA DEL SEPOLCRO DI S. COSTANZA, FABBRICATO DA COSTANTINO MAGNO, ED ERRONEAMENTE DETTO IL TEMPIO DI BACCO, IN OGGI CHIESA DELLA MEDESIMA SANTA.
antiche. 4. Candelieri antichi di marmo. 5. Pitture moderne. 6. Urna di Porfido tutta d'un pezzo lunga pal. 11. alta p. 5.10. larga 6.9. Il coperchio alto 3.2

Figure 1.7: Giovanni Battista Piranesi - *Veduta interna del sepolcro di S. Costanza*, etching, 1756.

the disparity map layers. We show that utilization of such an approach for the automatized creation of a stylized image representation can be useful to convey shapes and produce a handcrafted-like result of images of a natural scene. We also show that this approach is applicable to produce stylized image results in one as well as in more colors. The goal of this approach is to artistically depict images of a real scene and in this way to assist the artist, to support the art creation learning process or simply to help common users to produce interesting artistic image results.

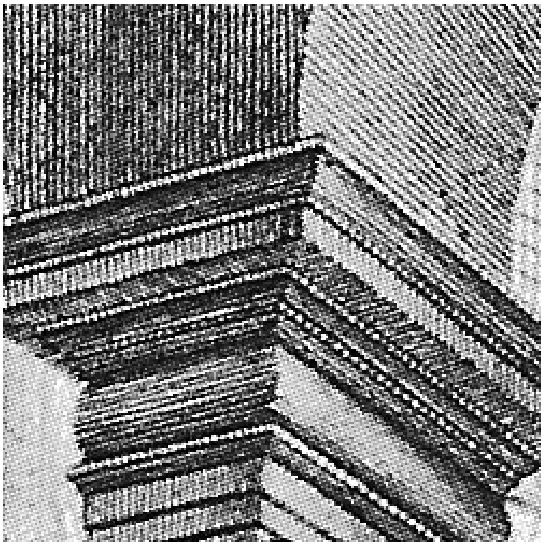


Figure 1.8: A detail of the arch shown in Figure 1.7.

1.5 Contributions and Organization

The foremost contributions of this dissertation are novel techniques to stylize images of natural scenes, recorded using a stereo set-up, in a form similar to creations of traditional artistic styles. The general concept of the work we present here is to create a stylized representation of the real world with the attention on depiction concepts that preserve the natural visual aspect of a scene and its content appearance.

Natural scenes, as opposed to modeled 3D scenes, provide a large variety of ready-made objects, shapes and structures. This characterizes them as an excellent source for further stylized representation and also as a big challenge to accurately extract the information needed for the depiction process. Textured regions, variations in illumination such as shadows or reflections and noise set the precise feature extraction to a complex task.

In this thesis we offer some solutions to existing problems in image based natural scene depiction. As a remedy to the problem of the lack of depth information, which arises in image based rendering of real scenes, we propose using disparity maps computed from stereo image pairs. The information obtained from disparity maps helps us to portray the dominant scene structure, to preserve the "depth" feeling of a scene or to depict motion. We will show the usefulness and benefit of employing such information, through obtained results, particularly for the image stylization of natural scenes. The major goal of our work is to make the stylization algorithms robust to errors present in the stereo-derived disparity maps, which make the extracted depth information of a natural scene to be significantly lower quality than the one obtained from a modeled scene.

The research topics of the thesis are summarized and structured as follows:

In Chapters 2 and 3, we provide an overview of related literature and background information relevant to our work. In Chapter 4, we introduce the Edge Combination algorithm, a silhouette extraction method, which imparts the dominant structure of the scene based on information from the original image and the computed disparity map. In this chapter, we also show that a clean and distinct Edge Combination image, which is delivered by the algorithm, can be used in synergism with Active Contours to significantly improve the results of the object extraction process.

Afterwards in, Chapters 5, 6 and 7, we demonstrate the usefulness of the Edge Combination approach in the main or supporting role for various image stylization procedures. In this manner, in Chapter 5, we present a method for representing images of a real scene in a sketched like form. The apparent dominant structure of the scene, defined through silhouettes of the Edge Combination image, provides the basis for the imitation of a human sketching style. We represent edges of the Edge Combination image in a parametric form by fitting higher degree Bézier curves and further stylize them to produce a handcrafted like appearance.

We explicitly use the term sketch to describe the output of our algorithm, instead of drawing, because of a fundamental difference between these two styles. Sketching loosely captures the likeness of shapes, roughly pointing out important features without adjunct details, while drawing, as we will show in the following Chapter 6, concentrates more on precise depiction and object's shape description, using often shading to faithfully portray objects in a form of an artwork. An example of the difference in those two styles is given in Figure 1.9, where a leaf is handcrafted in a sketched and in a drawn form.

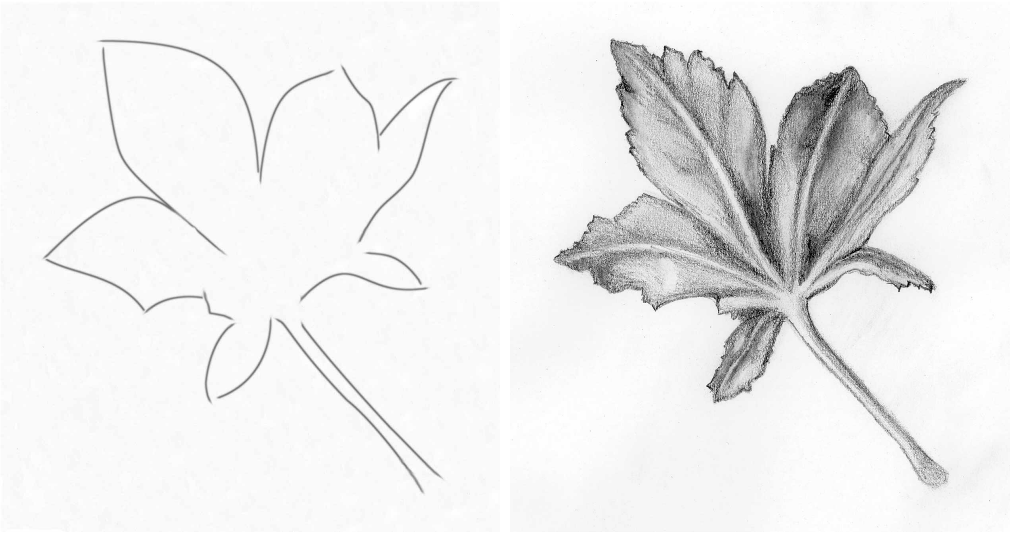


Figure 1.9: The example of an object handcrafted in sketched and in drawn form.

In Chapter 6 we focus on depicting images of a real scene in a drawn like form. The novelties we introduce include an image enhancement in order to better convey shapes of objects and an improved shading style, with strokes following the orientation of the disparity map layers to preserve the "depth feeling" of the scene.

In Chapter 7 we show the advantage of employing silhouettes of the Edge Combination image sequence to depict motion in a comics like style by using motion lines and multiple contours.

Finally point, in Chapter 8, we conclude with a summary discussion and suggestions for future research.

1.6 Publications

Through the process of composing this thesis document, parts of the work have been published in several scientific articles:

- D. Markovic and M. Gelautz. Comics-Like Motion Depiction from Stereo. In *Proceedings of the 14-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG '06)*, pages 155–160, Plzen, Czech Republic, February 2006.
- D. Markovic and M. Gelautz. Drawing the Real. In *Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia (GRAPHITE '05)*, pages 237-243, Dunedin, New Zealand, November 2005.
- E. Stavrakis, M. Bleyer, D. Markovic and M. Gelautz. Image-Based Stereoscopic Stylization. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 5–8, Genova, Italy, September 2005.
- D. Markovic, E. Stavrakis, and M. Gelautz. Parameterized Sketches from Stereo Images. In *Proceedings of the 17th Annual Symposium on Electronic Imaging: Science and Technology*, volume 5685, pages 783-791, San Jose, California, USA, January 2005.
- D. Markovic and M. Gelautz. Experimental Combination of Intensity and Stereo Edges for Improved Snake Segmentation. *International journal "Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications"*, 5(1):243–247, 2005.
- D. Markovic and M. Gelautz. Experimental Combination of Intensity and Stereo Edges for Improved Snake Segmentation. In *Proceedings of the 7th International Conference on Pattern Recognition and Image Analysis: New Information Technologies*, pages 310–313, St. Petersburg, Russian Federation, October 2004.

- M. Gelautz and D. Markovic. Recognition of Object Contours from Stereo Images: an Edge Combination Approach. In *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'04)*, pages 774–780, Thessaloniki, Greece, September 2004.
- D. Markovic and M. Gelautz. Video Object Segmentation Using Stereo-Derived Depth Maps. In *Proceedings of the 27th Workshop of the AAPR/ÖAGM*, pages 197–204, Laxenburg, Austria, June 2003.

Chapter 2

Related Work

2.1 Introduction

Throughout history, image creations have always been used as a form of notation, communication, message, and observation, or as means to record a moment, place or likeness of a subject. Creators of such carriers of expression, important for any culture or time period, have been developing their manual dexterity in order to convincingly depict objects of the real, imaginary or religious scenery. With the invention and expansion of photography, high realism in object depiction lost some of its importance, which left more space for the creation of new styles and techniques of visual art expression.

Major innovations in technology have been always influencing and bringing innovations in the art creation, new tools and/or styles. The wide usage of computer software was not an exception. From the pioneering work till today, computer graphics has become an extremely powerful, effective and usable domain of expression, which complements the traditional ways of presenting, visualizing, conveying and portraying the visual information.

2.1.1 Photorealistic Rendering

In a segment of computer graphics known as photorealism or photorealistic rendering, the typical goal is to generate images that are impossible to differentiate from a photograph of a scene. Over the past few decades great progress has been made toward the development of algorithms that achieve a convincingly realistic appearance of the modeled scene. As we live in a world that is remarkably rich in objects of various sophisticated shapes and materials, whose appearance is affected also by the interplay of light, photorealistic rendering faces an extremely challenging task in producing the illusion of real. Due to such a demanding and complex goal, it is very hard to achieve the general photorealism. Instead, the attention of photorealistic research is more distinctly structured through areas that focus on particular problems in simulation of elements and phenomena of the physical world. Figure 2.1 shows an example of realistic scenery rendering, a result of the work presented in [36], which is focused on realistic simulation of natural plant ecosystems.

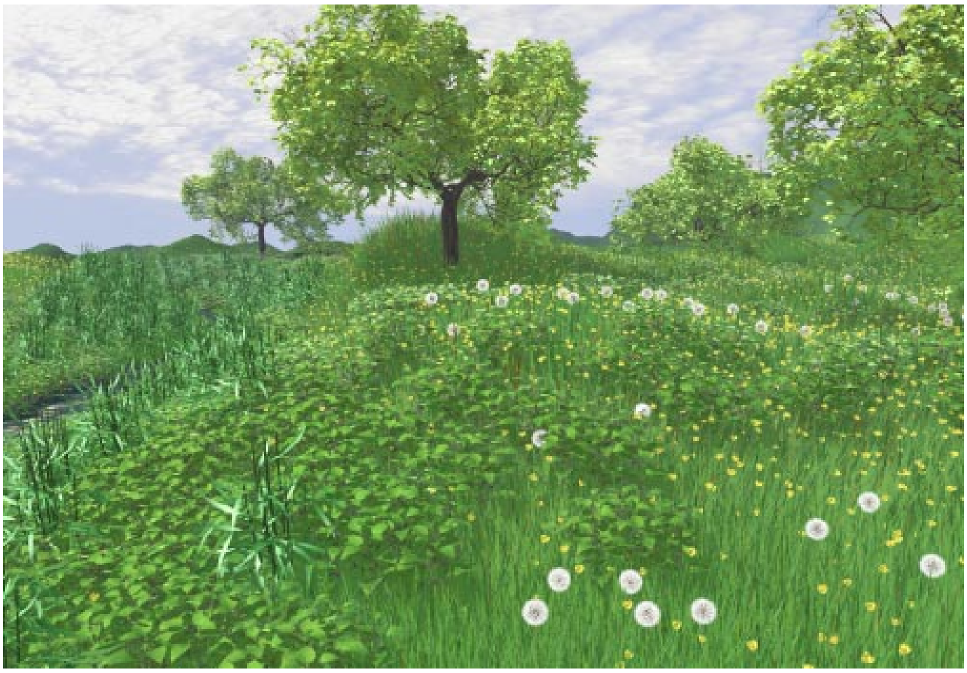


Figure 2.1: Realistic landscape presented in [36].

2.1.2 Non-Photorealistic Rendering

On the other side, the fascinating human power to create fine pieces of artwork has always been attracting scientists' attention. Over the years, a thread of research devoted to the simulation of the human ability to depict real and fantasy worlds has been progressing steadily toward the development of sophisticated methods and approaches that give an output which resembles the result of traditional stylization techniques of the art creation. Known as non-photorealistic rendering (NPR), which essentially means that the rendered output is not being photorealistic, this area of research has advanced to a wide and powerful field of not only traditional art forms emulation but also to a field of novel art styles and forms development [47].

The intention behind this NPR research is not to replace artists but to set a flexible and practical environment for non-highly experienced, educated or skilled people to produce interesting and appealing art forms. Also, because of the high accuracy of the produced results, systems that simulate traditional art techniques can be very useful in training and educational programs. A rich set of novel tools that NPR researchers have developed generated new possibilities and new ideas and also stimulated the creation of some new art forms, which would be impractical or too demanding for handcrafting.

The fact that NPR provides a rich field of opportunities to experiment, depending on the human creativity, perception and expression, can bring artists or viewers to interact more closely with the artwork. The artwork, for instance, could change constantly or from time to time according to the viewers attention, mood or will. An example of such an idea is the approach presented in [115]. In this work, an interactive painterly rendering approach, the "empathic painting", is presented, in which the appearance of the output adjusts itself in real-time to reflect the perceived emotional state of the viewer. In order to identify the emotional state of the viewer, through recognition of the user's facial expression, a set of computer vision algorithms is employed. An illustration of such an interactive approach of stylization through different emotional states of the viewer is shown in Figure

2.2. Another potential form of interaction, in which the artwork is not just reflecting, but stimulating the emotional state of the viewer, could be very inspiring, motivating and pleasurable for the user.

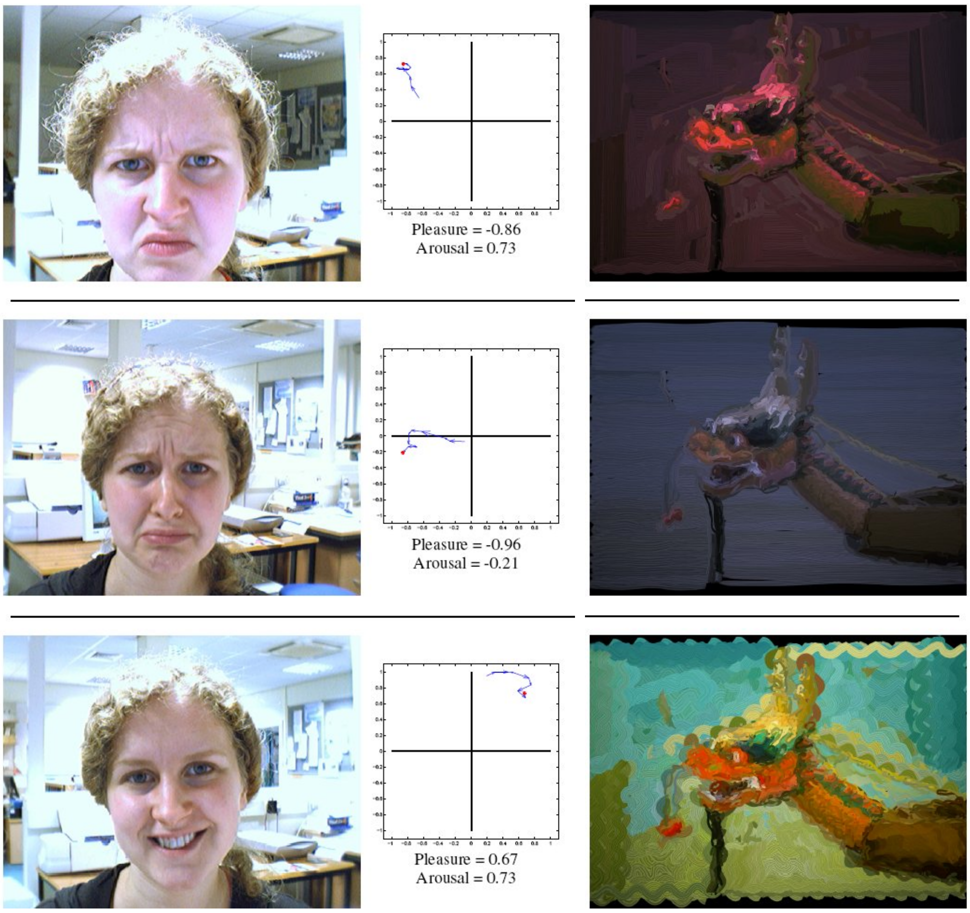


Figure 2.2: Empathic Painting. An example of adjusting stylization output according to the emotional state of the viewer, presented in [115].

Classification

In order to roughly categorize NPR work, various approaches can be used [47]. For instance, one type of classification could be made according to the degree of the user intervention required to produce the expected output. In this sense, NPR systems take a range of classes from automatized to systems that very much depend on the user assistance in the creation process.

A further classification can be made according to the type of pictorial art simulated in the output of the NPR system. In this case, by following the classification of pictorial art given in [4], NPR algorithms can be grouped into algorithms that produce results which resemble:

- paintings [31, 55, 114, 46, 27, 53, 134, 118, 74, 15, 115],
- drawings [126, 109, 108, 116, 56, 96, 39, 76, 103, 72, 135, 80],
- mosaics [41, 52, 69, 13, 44],
- prints [87, 93],
- collages [62, 102].

An additional advantage of computer assisted types of the art creation can be seen through some of the NPR algorithms, which are able to simulate different styles of a particular technique [55, 53]. In this case, various styles of one technique can be simulated by simply changing input parameters. Such a flexible and adjustable approach can be very inspiring for the user by providing a possibility to experiment with different painterly effects. An illustration of such an approach is given in Figure 2.3.

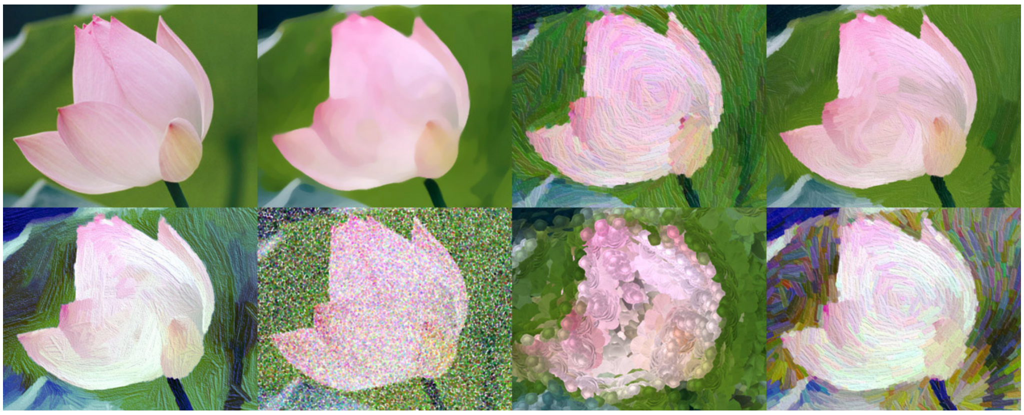


Figure 2.3: Different styles of painterly rendering of a flower (top left), presented in [53].

Another group of NPR systems, described in [94, 128, 66], simulate the creation of artworks whose construction is not directly associated to a specific type of pictorial art. Generally, this type of artworks, such as maze or ornament creations, can be produced by using different techniques such as painting, engraving, drawing or mosaicking. Equivalently to this, NPR systems which simulate the creation of such artworks produce an output that can be adapted to imitate the effect of different techniques.

Finally, we can also classify NPR algorithms according to the type of the input used for depiction. A great deal of developed algorithms, as we will show in following, takes as input a modeled 3D scene. They benefit from the precise information that can be extracted from a 3D polygon mesh. In this way, remarkable results can be produced but, naturally, they are strongly dependent on the modeling process.

Another group of algorithms, that we will put more emphasis on, takes images as input. Due to not fully available information needed for the depiction process, these algorithms usually rely on various image features and on information that can be extracted from them. Particularly, we will focus on algorithms that take as input images of a natural scene. This is an extremely challenging type of input because of the difficulties that arise in extracting accurate information, as desirable for the image stylization process.

2.2 Silhouettes

2.2.1 Modeled Scene

Many techniques use silhouettes to stylize or abstract scene objects, however most of them rely on available 3D models and the precise information that can be extracted from a modeled 3D scene. The most important of these methods are presented in a survey of silhouette algorithms for polygonal models [64]. Generally, these algorithms can be classified into image space, object space and hybrid approaches.

Image Space Algorithms

Image space algorithms find silhouettes of a 3D modeled scene by detecting the silhouette edges in the image buffers and extract them using image processing techniques. The principal difference to our method is that we use images of a real scene as an input to reconstruct and to extract the necessary scene information, a task that poses a very significant challenge. In the following discussion of modeled scene approaches, for the sake of conciseness, we emphasize those image space techniques that employ depth information to find silhouettes, since they are more related to our work.

The incorporation of depth information in order to enhance the visual perception of an object's shape in illustrations was pioneered by Saito and Takahashi [107]. In this work, the concept of G-buffers is introduced. G-buffers are image buffers used to store geometrical properties of the 3D scene. In particular, image processing operations on a G-buffer, that stores depth values, are used to extract silhouettes. The extracted silhouettes are further composed with the result of the scene's rendering in order to enhance the visual comprehensibility of the scene. The idea of using depth information is further explored by Decaudin [34] to produce "cartoon looking" computer generated images from data representing a 3D scene. Deussen and Strothotte [37] presented a method to extract important silhouette lines using depth discontinuities determined by a depth difference threshold to create pen-and-ink illustration of trees in different levels of detail. Curtis suggested a loose and sketchy rendering framework [30], see Figure 2.4, where line plays an important role in the communication of an animation's mood. In this technique a depth map is used to extract two intermediate images, one describing the amount of ink required to be deposited on specially selected image locations, such as depth discontinuities, while the second one is used to guide a particle simulation that crafts the final sketchy lines. This approach is further extended to a real-time rendering algorithm by Ho and Komiya [59].

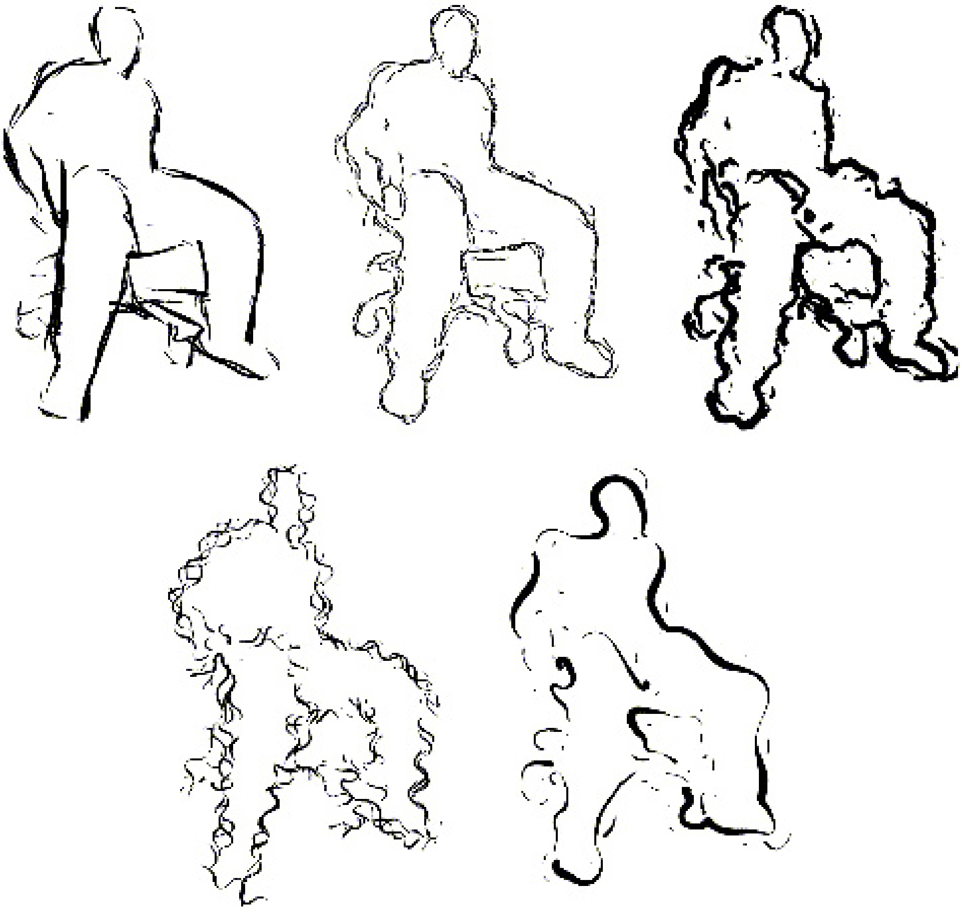


Figure 2.4: Loose and Sketchy Animation. The figure shows various styles that can be obtained by varying parameters as presented in [30].

Object Space Algorithms

Another class of algorithms function as the object space approaches. The basic idea behind object space approaches to silhouette extraction is to find edges that connect front-facing to back-facing polygons. These algorithms operate directly on 3D polygonal meshes, avoiding the representation of silhouettes in a pixel matrix. The problem that they have to solve is the hidden line removal, which means to eliminate all the detected edge lines in the silhouette finding process that are occluded and should not be visible in the final result.

Hybrid Algorithms

A group of algorithms that aim to use positive sides of both, image and object space techniques, are hybrid algorithms. In this case, object space operations and rendering steps are performed that yield silhouette edges in image space. This hybrid type of approach usually requires multiple pass rendering and the result is a silhouette represented in a pixel matrix.

2.2.2 Natural Scene

On the other side, silhouette extraction has been a significant area of research also for algorithms that operate on images of real scenes. These approaches typically consider a silhouette edge to be the object's outline, without taking into consideration additional silhouette edges that may arise at depth discontinuities inside the object, which can enhance the image. From that simplified point of view, the problem of finding silhouette edges is mainly transformed to a problem of effectively separating foreground and background in real images. This poses one of the fundamental and crucial problems in Computer Vision, since the input data may contain various artifacts (e.g. noise) and therefore provide unreliable information for many image processing algorithms. In addition, the lack of a geometric description of the imaged scene makes object-space algorithms ineffective in this case.

One considerable research topic dealing with silhouette extraction and utilization is Shape-From-Silhouette (SFS), a 3D reconstruction method for estimating the shape of an object from multiple silhouette images to construct the object's Visual Hull (VH). First introduced in the PhD thesis of Baumgart 1974 [9], SFS became a very interesting method for further research. Initially performed on static objects, some recent SFS developments are targeted toward the shape estimation of dynamic objects [24]. Computing silhouettes for SFS is usually simple object separation from background using either background of a single uniform color (chromakeying) or image differencing. The latter approach uses a built statistical background model from recorded frames of the scene without presence of the object. This information is further used to extract silhouettes by finding a difference in new frames when an object moves into the scene [19].

Apart from the stylization application that we focus on in our work, the silhouette approach has also shown its advantage in many methods for estimating body posture and motion [99, 17, 1], gait-based human identification [122], video surveillance [70], content-based image retrieval [63], etc. To extract objects of interest, these systems typically use basic background subtraction techniques or specially created equipment.

Further advances of media technologies have generated the need of segmenting the foreground from a complex background. This initiated the development of a number of approaches with different levels of user interaction, such as Intelligent Scissors [88], Image Snapping [45], Active Contours [67], image matting [104, 25, 57, 120], graph-cut based methods [16, 12, 101] and also a group of algorithms that utilize disparity maps [33, 65, 129] in order to select the object of interest from the rest of the scene. While some of these methods require a lot of user interaction, others can only approximately find an object's outline. Similar to our idea, in the work presented in [98], depth discontinuities are used for the stylization of images of natural scenes. Depth discontinuities are detected by utilizing a camera with multiple flashes. The camera set-up is designed in such a way that flashes, placed on the camera, are positioned to cast shadows along depth discontinuities in the scene. Due to the strategy of finding depth discontinuities by exploiting actively produced shadows, this approach is not able to handle open space or distant scenes. An example of the result of this technique is shown in Figure 2.5.

In contrast, we propose a new automatic method that extracts accurately silhouettes of foreground objects by using disparity maps that are computed from stereo image pairs of natural scenes.

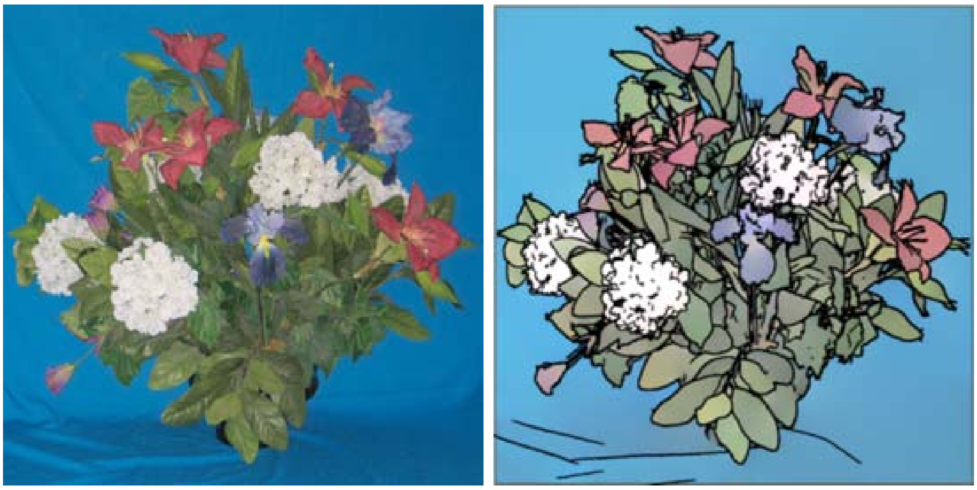

(a) (b)

Figure 2.5: Illustration of (a) the input image of a flower plant and (b) a stylized texture de-emphasized rendering, presented in [98].

2.3 Drawn-like Depiction

A lot of attention of the computer graphics community has been attracted to simulate expressive traditional media in the form of artistic creation. Consequently, considerable research has been devoted to produce renderings with a handcrafted appearance by simulating traditionally created drawings.

One part of related studies takes an available 3D scene as input, such as [126, 116, 56, 96]. These geometry-based approaches benefit from complete access to the 3D geometry. An example shown in Figure 2.6 presents a result of such a drawn-like depiction that takes a 3D model for input. This particular real-time approach, described in [96], aims to give an output that resembles those handcrafted drawings in which shading is produced using the hatching technique.

On the other side are image-based systems, which use available images of a modeled or natural scene to produce illustrations. These approaches have no a-priori comprehension of the scene geometry, and to convey shapes by orienting strokes, they include either user interaction [108, 39, 109] or information obtained from image feature analysis [76, 72, 80].

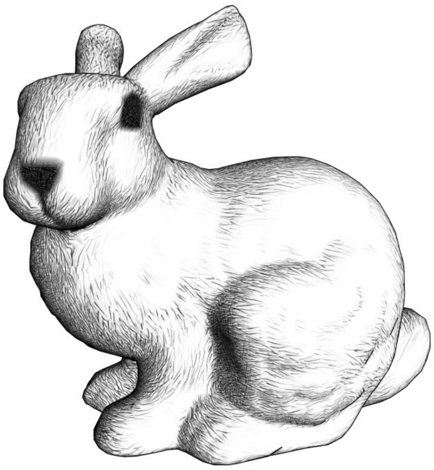


Figure 2.6: Example of a result of the real-time hatching algorithm result presented in [96].

An interactive system for creating pen-and-ink illustrations is presented in [109], where the user creates the image by “painting” with different stroke textures to create the desired result. The used stroke textures are formed from collections of strokes arranged in different patterns. In [72] a method for generating pencil drawings is described. Stroke directions are obtained from texture analysis of each region identified in the color segmentation process. Another technique for the automatic generation of pencil drawings from images as input is presented in [76]. This method uses line integral convolution, a texture based vector field visualization technique, to simulate pencil-like drawings. In order to convey shapes and textures of objects, strokes directions are assigned by employing the information obtained from the analysis of the directionality of textures. This work is further extended in [135] to a colored pencil drawing filter. In [80], see Figure 2.7, another method for creating colored pencil style images is proposed. To resemble handcrafted appearance, shading strokes are placed in a way that their directions align along the boundaries of segmented regions.

An earlier work by Saito and Takahashi [107] demonstrated the benefit of using depth maps to create a variety of styles to visualize the 3D scene.

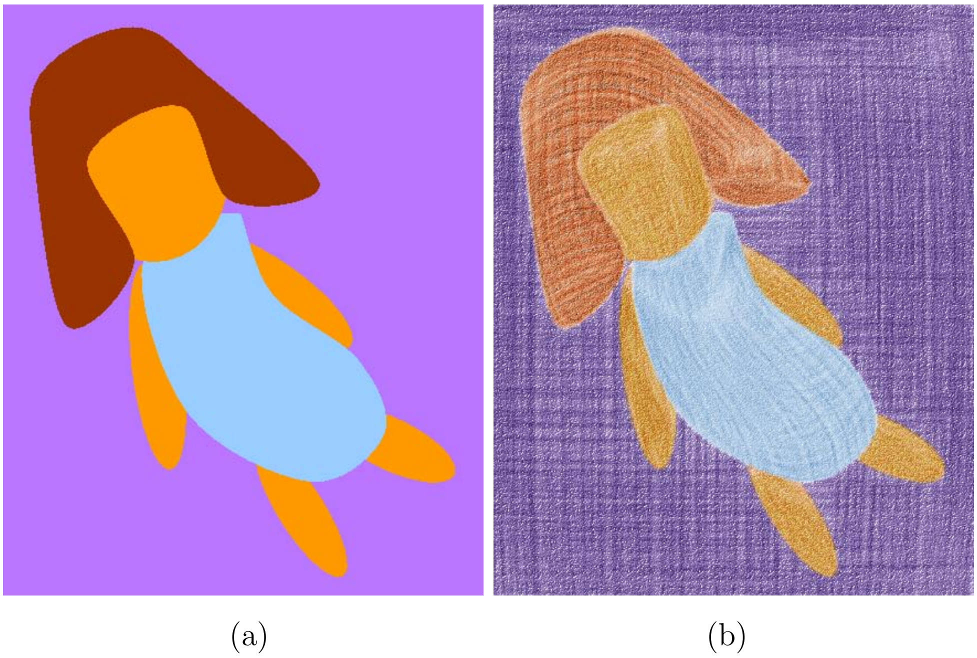

(a) (b)

Figure 2.7: Illustration of (a) the input doll image and (b) a drawn-like representation in colored pencil style, presented in [80].

In this work, besides the image enhancement method that we already mentioned in Section 2.2.1, an approach for drawn-like image representation of 3D models is presented.

Recent research has shown the advantage of using combined techniques. The approach in [112] is image-oriented and additional 3D information, if available, is used to create a larger variety of styles. For the stroke orientation, in this work, various schemes can be employed. Strokes can be oriented by simply following a constant direction, by using sources such as color or intensity gradient information or by utilizing their combination. Figure 2.8 shows an example result of this approach. In this case, for the stroke orientation, the color information of the original image is used.

In [133], an interactive non-photorealistic rendering system is presented, which stylizes and renders outdoor scenes captured by 3D laser scanning. In this work, due to the large size, complexity and incompleteness of the obtained data, a point-based representation of laser scans is used, along with

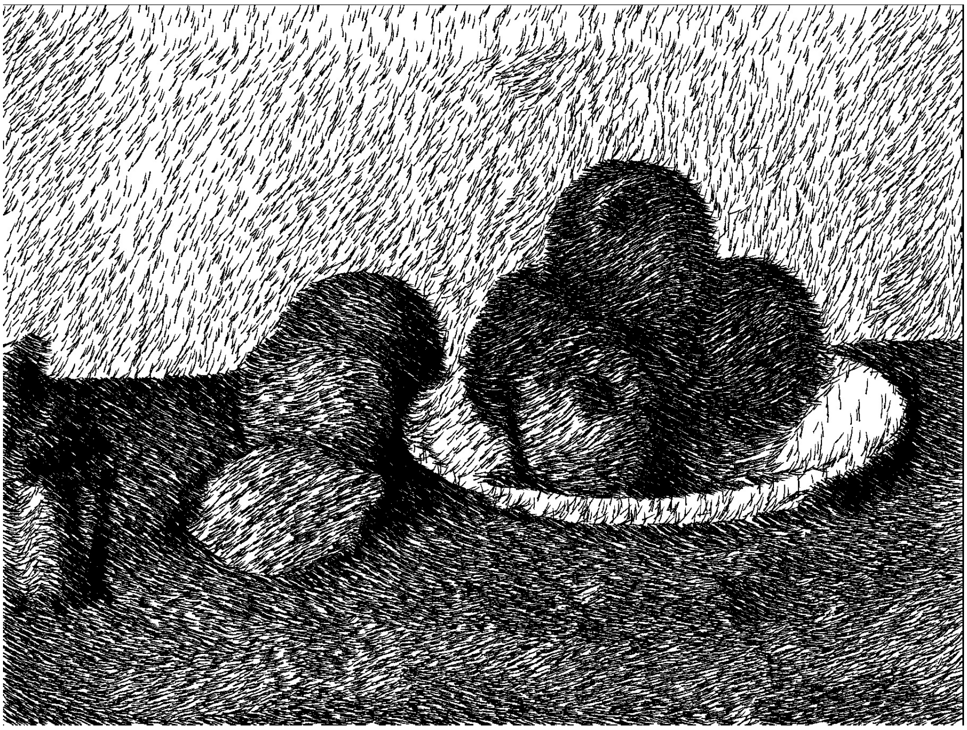


Figure 2.8: A result of the drawn-like depiction approach presented in [112]. The output is a hatched rendering of Cezanne's *Still Life With Apples* used as input.

the geometric information estimated at every point. This information and/or some other criteria such as data accuracy are further used for the "feature degree" computation for each point. By applying a certain threshold, feature points, a subset of points with high feature degree, are extracted. For the final depiction, a two-pass rendering pipeline is used, in which the first pass generates a depth buffer, called visibility mask, and the second pass renders the points in stylized form by utilizing different primitives such as point sprites, line segments and textured strokes. Through this system a natural scene can be represented in a variety of drawing styles and in a form of coherent animation.

As opposed to the previously discussed image-based systems, this thesis is concentrated on recorded stereo image pairs of real scenes. We include additional information by computing the disparity map, which is inversely related to the scene depth, to attach to the illustration a feeling of "depth". The result of our approach resembles the form of the traditionally handcrafted drawings as we will show in more detail in Chapter 6.

Chapter 3

Background

3.1 Stereo Vision

3.1.1 Human Vision and Stereoscopy

How do we see? How do we exploit all the visual information in our everyday life without even noticing that a complex process is happening behind it? Researchers and scientists in many fields have been trying to give answers to questions like these for a long time.

The human visual system is a binocular visual system. This essentially means vision by using two eyes. A slightly different view of the same area, with overlapping fields, captured by each eye is forwarded to the brain for further processing. The fusion of these views in the brain sets up disparities that give us the perception of depth and relative distance of objects. This process in visual perception is called *stereopsis*. The word stereopsis was formed from the Greek words *στερεός* (stereos) meaning solid and *όψις* (opsis) for look.

Although the roots of stereoscopic creative work go far back to the binocular drawings made by Giovanni Battista della Porta (1538-1615) and Jacopo Chimenti da Empoli (1554-1640), pioneer steps to explain binocular vision and to construct a viewing device, the stereoscope, were made by Sir Charles

Wheatstone. In June 1838, he published the paper on stereopsis entitled "On some remarkable and hitherto unobserved, phenomena of binocular vision" [124], where he pointed out that the differences in the two eyes' images, due to the horizontal separation of the eyes, produce the impression of solidity in the mind. A new device, called stereoscope, used two pictures of the same subject from two slightly different viewpoints, displaying them in such a way that each eye sees only one of the images, allowing stereopsis to be stimulated to create a depth illusion. The first stereoscope constructed by Sir Charles Wheatstone, a simple two-mirrored device called "Reflecting Mirror Stereoscope", was improved by Sir David Brewster. He created in 1849 a more practical stereoscope using lenses, a "Lenticular Stereoscope". An illustration showing both devices is given in Figure 3.1.

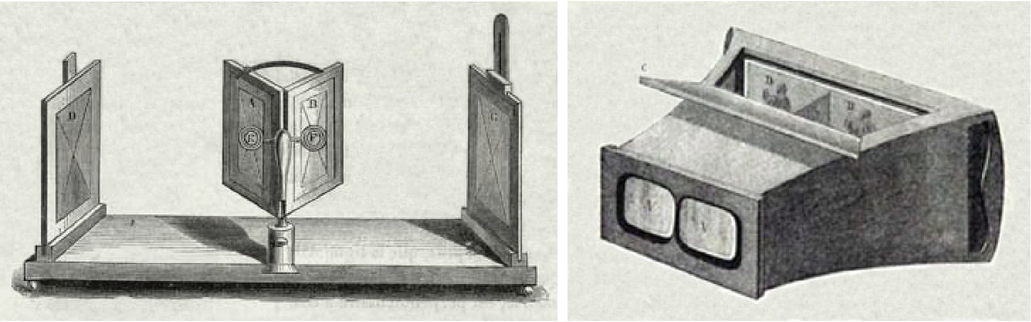


Figure 3.1: Wheatstone's (left) and Brewster's (right) stereoscope.

Over time stereoscopic devices were changing the size, shape and structure. They were mostly used for entertainment, allowing a viewer to see the third dimension of stereoscopic images. Entertaining photos of family, friends, and landscapes viewable in 3D evolved the idea to extract extremely valuable depth information automatically for a further usage.

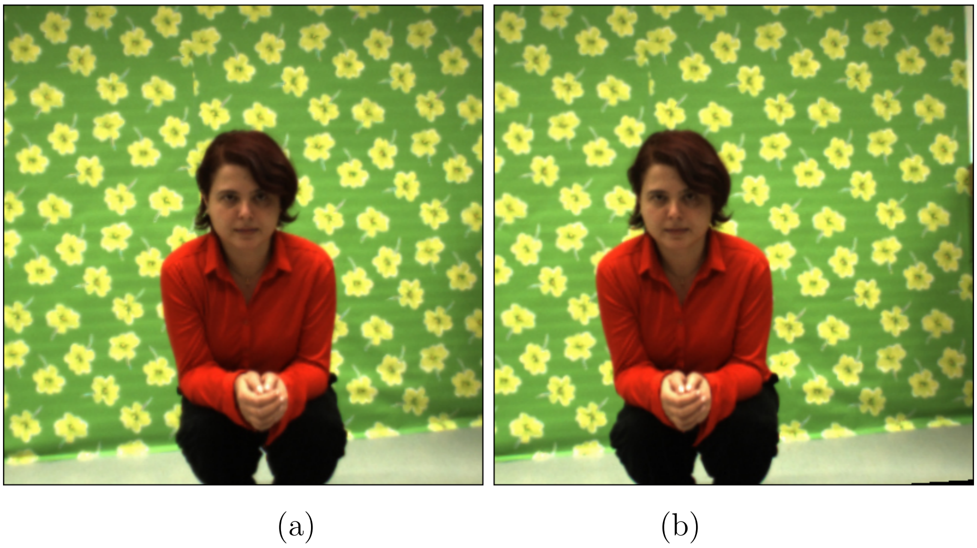
3.1.2 Machine Stereo Vision

As pointed out before, the human visual system is a sophisticated system, which uses stereopsis to extract the necessary depth information that we use in our everyday life. In our mind, the distance of objects is determined using the displacements in the two images that we receive. If the object is further away from the eye, the difference in the two images will be smaller and, otherwise, if the subject is closer, the difference is larger.

A deficiency of the human visual system is that we actually cannot use these image dissimilarities to measure precisely the distance in our mind. For example, if someone throws a ball we will be able to use the information perceived from our eyes to catch the ball, but we will not be able to determine exactly the length of the ball's path or the speed of the moving ball.

The idea of measuring differences in images recorded from slightly different viewpoints to extract depth information initiated a lot of research and has an important purpose in many applications. A considerable amount of research has been focused on depth-from-stereo algorithms. These algorithms use a similar principle as our mind to extract depth information. Figure 3.2 shows an example of a stereo image pair recorded from cameras positioned in parallel. The images are preprocessed, undistorted and rectified, which essentially means that a pixel in one image of the stereo pair should have a corresponding pixel in the other image at the same processing scanline. This procedure makes computing stereo correspondences simpler, allowing a search for corresponding points to be performed along the horizontal lines of the rectified images, which basically simplifies the problem of the two dimensional search to one dimension. Object positions in these images appear slightly shifted. Dissimilarities of the stereo image pair are used further by a stereo algorithm to extract a depth map of the scene. Basically, the output of the stereo algorithm is a disparity map that is computed according to the determined correspondences in two images. It is stored often as a gray value image, called disparity image.

As given in [40], in the case of parallel cameras, disparity and its relationship to depth can be defined as follows: Given a scene with a physical 3D point M and its two projection points m_1 of coordinates (u_1,v_1) in the first image plane and m_2 of coordinates (u_2,v_2) in the second image plane



(a) (b)

Figure 3.2: Example of the stereo image pair: (a) Left camera image, (b) right camera image.

(see Figure 3.3), the *disparity value* d is defined as:

$$d = v_2 - v_1 \tag{3.1}$$

The relationship between disparity d and the distance z of the 3D point M is given by:

$$d = v_2 - v_1 = d_{12} f / z \tag{3.2}$$

with f being the focal length and d_{12} denoting the distance between the two optical centers C_1 and C_2 as shown in Figure 3.3.

The problem of determining correspondences between pixels in the left and right image becomes more complex if we take a closer look at open problems in depth-from-stereo computation. Stereo matching algorithms attempt to reduce the number of false matches by employing different strategies, but several problems are still unsolved. In addition to the complexity of the scene, which strongly influences the correspondence problem, the stereo matching process encounters difficulties in dealing with occlusions. Due to different camera viewpoints, not all the pixels in the left image have a corresponding pixel in the right image. This generates the problem of assigning a disparity

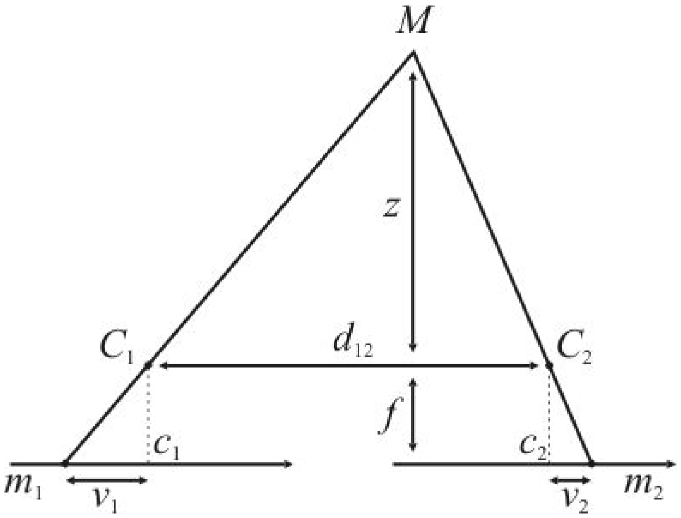


Figure 3.3: Disparity and depth relation for parallel cameras. [40]

value to occluded pixels when computing a final dense disparity map. Difficulties that also encumber the stereo matching process, causing erroneous correspondences, are inducted by light variations, noise or by texture related problems, such as untextured areas or regions with strongly regular and/or repetitive texture patterns.

The challenge to solve these problems stimulated machine stereo vision to become a very intensive and dynamic research field. The importance and usefulness of a reliable depth map computation initiated the proposal of numerous algorithms. For further study of stereo techniques we refer the reader to the survey of Scharstein and Szeliski [111], which accompanied by the Middlebury Stereo Vision Page presents a good overview of stereo algorithms and current developments in the field.

In order to produce smooth and detailed disparity maps, stereo approaches often adopt certain assumptions, usually by enforcing uniqueness and continuity constraints, originally proposed by Marr and Poggio [77]. These constraints can be described as follows:

- Uniqueness: Each point may be assigned at most one disparity value.
- Continuity: Disparity varies smoothly almost everywhere.

According to the employed strategies, most of the algorithms can be broadly classified into local and global methods. A major difference between these two approaches is the different continuity (smoothness) assumption for the disparity map.

Local methods employ a window-based approach in order to solve the correspondence problem. Their main problem is the selection of the appropriate window size. Assuming *implicit* smoothness assumptions and constant disparities within a search window, local methods encounter problems at depth discontinuities, which results in the foreground fattening effect [58].

In contrast to this, global methods apply *explicit* smoothness assumptions. These smoothness assumptions are expressed in a global cost function, which is then optimized. Usually slower than local approaches, global methods generally produce more accurate and coherent depth maps. Local methods still remain to be one very important area of stereo vision research because of their simplicity and efficiency, especially valuable for real-time applications.

In this thesis we aim to show the usefulness of disparity maps for the stylization of images of natural scenes. In order to produce results that resemble traditionally created artwork and their handcrafted appearance, we employ disparity maps computed from stereo image pairs. Therefore, to obtain dense disparity maps, we particularly put attention on two approaches because of the characteristics of results that they produce. We provide a more detailed description of these approaches in the following. First we describe a fast but less accurate stereo matching algorithm developed by Birchfield and Tomasi [11] and right after we present the approach suggested by Bleyer and Gelautz [14] that produces more reliable and less noisy disparity maps.

Pixel-to-Pixel stereo [11]

This method aims to extract reliable depth discontinuities from a stereo pair of images by obtaining a rough disparity map. Stereo correspondences are computed by matching scanlines independently using dynamic programming. An assumption made in this approach is that a depth discontinuity is accompanied by an intensity variation. The ordering constraint is used to locate occlusions in scanlines. This constraint is utilized to suppress noise that affects the disparity map computation especially in untextured regions. In order to handle sampling artifacts and false matches, an image sampling insensitive dissimilarity measure is devised that compares each pixel in the reference image with the linearly interpolated intensity function surrounding its matching pixel in the other image. To refine the obtained disparity map, a postprocessing step that propagates reliable disparities into their neighboring regions of unreliable disparity, by processing the disparity map in vertical and horizontal direction, is employed until an intensity variation is found. An example of a disparity map obtained by using this approach for the input stereo pair shown in Figure 3.2 is given in 3.4.

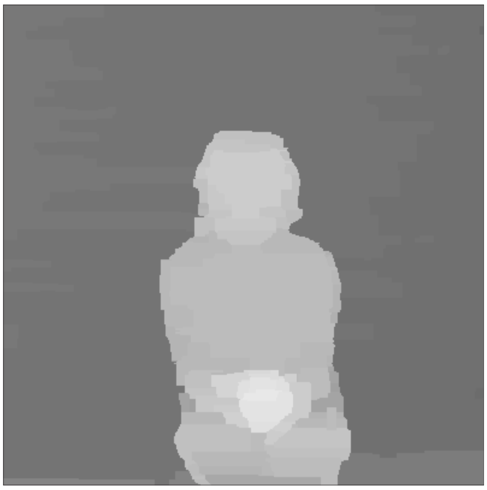


Figure 3.4: Pixel-to-Pixel stereo: Resulting disparity map for the input stereo pair shown in Figure 3.2.

Graph-cut based stereo matching algorithm [14]

To get a more accurate and less noisy disparity map, this method employs color segmentation as a valuable information in determining stereo correspondences. According to this, two assumptions are made: disparity inside each color segment varies smoothly and depth discontinuities coincide with segment borders. In the first step, an initial disparity map is computed by using a window-based method to initialize each segment's planar disparity model. From these initial disparity models, a set of disparity layers, which represent the dominant depth planes occurring in the scene, are extracted in the layer extraction step of the algorithm. In the final, iterative layer assignment step, each segment is assigned to exactly one of those planes by employing a global cost function to measure the quality of such an assignment. A local minimum of the cost function is determined by using a robust graph-based optimization technique. The computed final disparity map, represented in a set of planar layers, can as well be considered as disparity segmentation. An example of a resulting disparity map obtained by using this approach for the input stereo pair shown in Figure 3.2 is given in 3.5.

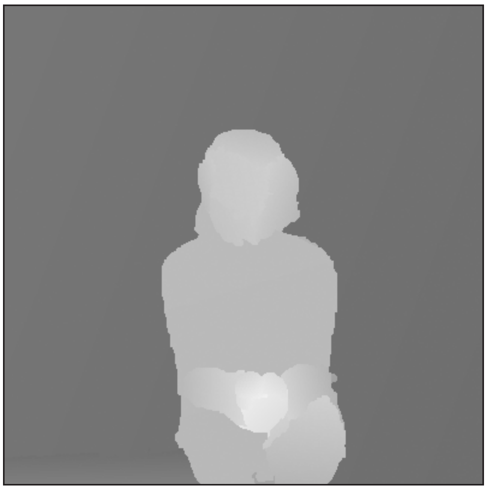


Figure 3.5: Graph-cut based stereo: Resulting disparity map for the input stereo pair shown in Figure 3.2.

3.2 Active Contours

3.2.1 Introduction

In the seminal work titled “Snakes: Active Contour Models”, in 1988, Kass et al. [67] introduced active contours, also called snakes. The snake is presented as the evolving curve that serves to select the boundary of a salient image feature. This work initiated numerous studies on further development as well as investigations on possible applications that employ snakes in the main or an additional role. Originally, active contours were proposed as an interactive method for detecting edges, lines and subjective contours as well as an effective approach for motion tracking and stereo matching. Over time, snakes gained a lot of attention, especially in research in medical imaging, computer graphics and, generally, in very important computer vision topics such as motion analysis, recognition, reconstruction and object segmentation.

In medical imaging research, active contours are widely used to collect important information about anatomical structures. Understanding, localizing, segmenting and tracking anatomical structures or changes in those structures, caused by a particular disorder, can help to identify the source of a problem, monitor the development of a disease and potentially lead to a better treatment. In this sense, active contours are effectively used in the diagnosis and in medical image analysis. Generally, they are employed in order to obtain accurate shape information, some indication of a normal variation or anomaly of anatomical structures, which is valuable information for the further medical analysis or treatment procedure. This promoted the development of active contour models and their variations in 2D, and in 3D as well, to a very important and challenging research topic. Consequently, since their proposal in the late eighties, active contours have become a powerful approach in medical image analysis to segment, match, quantify and track anatomical structures. [82, 130]

In computer graphics, particularly in modeling and animation, deformable curves and surfaces play an essential role. One of the important goals in computer graphics research is to create sophisticated shapes and objects that are modeled or animated in the context of complex environments and interactions with the environment, in order to resemble the natural appearance.

In research that focuses on image stylization and animation, active contours also found an appealing application. Generally this work is based or inspired by the original approach of active contours. A number of approaches employ snakes to select and track the subject, in order to create an animation from an image sequence [2, 3]. These are, typically, interactive approaches where the user plays an active role by adjusting and refining the snake curve. An example of such a technique, described in [2], is illustrated in Figure 3.6. In this work, an animation system, called SnakeToonz, is presented to generate 2D cartoons from a video sequence. The SnakeToonz system is designed as an interactive approach that provides to inexperienced people, particularly children, a tool to create the cel animation.

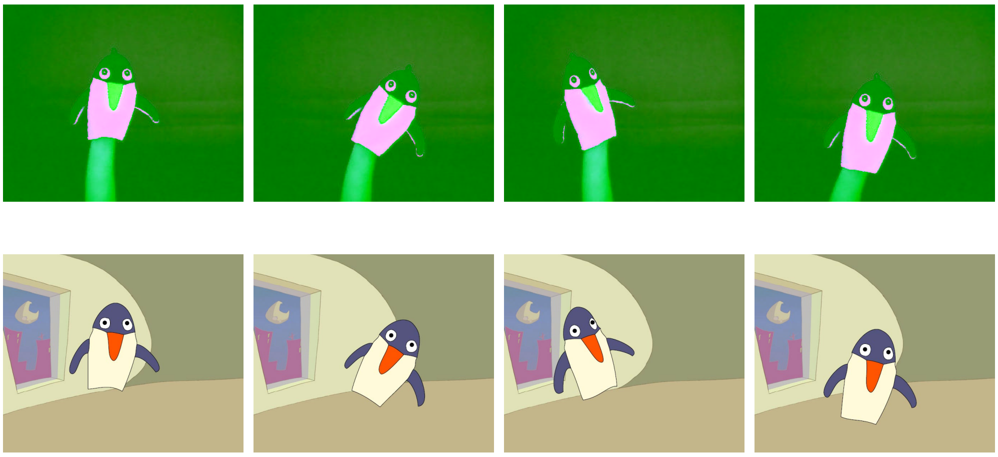


Figure 3.6: Example of a SnakeToonz animation along with its corresponding video frames as presented in [2].

Another useful application of active contours, in which snakes are employed for object selection and motion tracking, comes initially from the work of Kass et al. [67]. Originally, an example of tracking the movement of facial features, such as the speaker's lips, is presented. This inspired the development of systems for actor-driven facial animation [121, 60]. Again, snakes are used to select and track the facial features, in order to create the animation. This way recorded facial expressions of an actor are further used to animate computer created characters.

In this section, so far, we have illustrated the snakes' usability and importance in some of the research fields. In following, we will give some basics of the original approach. We will also shortly describe the Gradient Vector Flow (GVF) method, which we employ in synergism with the Edge Combination image, explained in Chapter 4, in order to accurately segment objects in images. A more detailed description of such an object segmentation approach can be found in Section 4.7.

3.2.2 Fundamentals - Traditional Snake

In the following we will give a concise description of the original approach, also known as the traditional snake approach, derived from the comprehensive and detailed explanations given in [67, 132, 91].

Fundamentally, active contours or snakes are deforming contours defined within an image domain, which as energy minimizing curves move under influence of internal forces, image forces and external constraint forces to user desired salient image features. Active contours were named also snakes because when they move toward image features, their motion type resembles the genuine snake movement.

In active contours, a contour is initiated on the image by specifying an approximate shape and a starting position close to the desired feature's boundary. In subsequent iterations the snake deforms itself under the influence of image constraints, designed to attract the snake toward features of interest. While moving, the snake maintains a certain degree of smoothness and continuity in the contour, specified by internal continuity constraints that force it to remain smooth. This can be expressed as an energy minimization process where the curve is deforming to minimize its energy functional and finally localize the minimum by sticking to the target feature.

In this sense, representing the position of a snake parametrically by $\mathbf{x}(s) = [x(s), y(s)]$, $s \in [0, 1]$, its energy functional E^*_{snake} can be written as:

$$\begin{aligned} E^*_{snake} &= \int_0^1 E_{snake}(\mathbf{x}(s))ds \\ &= \int_0^1 E_{int}(\mathbf{x}(s)) + E_{image}(\mathbf{x}(s)) + E_{con}(\mathbf{x}(s))ds \qquad (3.3) \end{aligned}$$

where E_{int} is the internal energy of the contour, E_{image} denotes the image energy and E_{con} represents the external constraint energy. The external constraint forces are employed in order to put the snake somewhere near the desired contour and can be introduced from different sources, like for instance from a user interface. In this case, the user could push or pull the snake to the desired position and internal and image forces would stick it closer to the target feature. A more detailed explanation of the internal energy E_{int} and image energy E_{image} is given in the following.

Internal energy

The internal energy of the snake, that regulates its characteristics of deformation, is defined as a weighted sum of the first- and second-order derivatives and can be written as:

$$E_{int} = (\alpha(s) \left|\mathbf{x}'(s)\right|^2 + \beta(s) \left|\mathbf{x}''(s)\right|^2)/2 \qquad (3.4)$$

The first-order term, controlled by $\alpha(s)$, characterizes the elasticity of the contour, making the snake to act like a membrane. The second-order term, controlled by $\beta(s)$, characterizes the bending flexibility of the contour, making the snake to act like a tin plate. The choice of the values of the weights α and β influences the tension and stiffness of the contour. High values for α produce large contraction, increasing the tension, whereas low values allow great change in point spacing. High values for β influence the snake to become smooth and less flexible while low values allow the curve to develop a corner.

Image forces

Image forces serve to drive the snake toward salient image features. The total image energy can be expressed as a weighted combination of three terms:

$$E_{image} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term} \tag{3.5}$$

where E_{line}, E_{edge} and E_{term} are line, edge and terminations energy functionals, respectively, controlled by weighting coefficients w_{line}, w_{edge} and w_{term}, respectively. Different snake behavior, usually needed for different applications, can be produced by adjusting the weights. The function of the energy functionals is to attract the snake to intensity extremes, edges and other image features of interest.

Minimizing the energy

In general, the behavior of the snake is determined by internal and external forces. The external force that serves to push the snake close to the salient image features is influenced by image forces and external constraint forces. In this sense, the external energy E_{ext} can be formulated as the addition of the image energy E_{image} and the external constraint energy E_{con}:

$$E_{ext} = E_{image} + E_{con}. \tag{3.6}$$

If we consider $\alpha(s) = \alpha$ and $\beta(s) = \beta$ to be constants, see 3.4, the curve that minimizes the given energy functional, must satisfy the Euler equation:

$$\alpha\mathbf{x}''(s) - \beta\mathbf{x}''''(s) - \nabla E_{ext} = 0 \tag{3.7}$$

In order to find a solution, the snake is transformed to the dynamic form, considering $\mathbf{x}$ to be a function of time t. The expression of the snake as a function of time is then:

$$\mathbf{x}_t(s,t) = \alpha\mathbf{x}''(s,t) - \beta\mathbf{x}''''(s,t) - \nabla E_{ext} \tag{3.8}$$

When the solution $\mathbf{x}(s,t)$ stabilizes, the term $\mathbf{x}_t(s,t)$ disappears, and the solution of 3.7 is obtained.

3.2.3 Aspects of Snake Behavior

There are several well known problems and limitations to the majority of active contour approaches. First of all, sensitivity to initialization - the initial contour should be placed close to the desired boundary, otherwise it can be attracted by other image features and a wrong result can be obtained. Another difficulty is the snake's sensitivity to parameters, which makes its convergence unpredictable. Sometimes it is even necessary to test the snake's behavior under different initialization and different parameters to obtain a satisfying result. Snakes are very susceptible to noise and the strength of the attracting feature, that also affect the quality of the result. Another limitation is the snake's inability of advancing into boundary concavities.

Many approaches reported in the very rich active contour literature aim to solve various snake limitations and to improve its behavior. To accomplish these challenging tasks, different strategies have been employed and numerous snake variations have been proposed such as a greedy algorithm [125], balloon model [26], B-snakes [84], dual active contour concept [49], Fourier snakes [117] and T-snakes [83]. Rather than finding an ideal, standard solution to solve all the snake limitations and difficulties, novel active contour methods focus on specific requirements of different applications. Here we put more attention on and shortly describe in following an approach, presented in [131, 132], which particularly deals with the snake's limitations in advancing into boundary concavities and initialization.

3.2.4 Gradient Vector Flow Snake

The Gradient Vector Flow (GVF) snake, presented in [131, 132], brings several advantages over the traditional snake, a particularly important one being the ability to move into boundary concavities. This is the effect of the external force field, which is derived from the image and designed to have a large "capture range", pointing into concave boundary regions. The increased capture range of the GVF snake is achieved through a diffusion process, which diffuses the gradient map further away from the edges and into homogeneous regions. The external force is also defined statically, which means that, computed at initiation, it remains unchanged in time and does not depend on the position of the contour.

The dynamic snake equation to be solved by the GVF snake, contains a new static external force field $\mathbf{v}(x, y)$, called the Gradient Vector Flow field, and can be written as:

$$x_t(s,t) = \alpha x''(s,t) - \beta x''''(s,t) + \mathbf{v} \tag{3.9}$$

This equation is obtained by replacing the $-\nabla E_{ext}$ in 3.8 with $\mathbf{v}(x, y)$.

Gradient Vector Flow

The energy functional ε to be minimized by the Gradient Vector Flow field, that is defined to be the vector field $\mathbf{v}(x, y) = [u(x, y), v(x, y)]$, is given as:

$$\varepsilon = \int\int \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 \, |\mathbf{v} - \nabla f|^2 \, dxdy \tag{3.10}$$

where the parameter μ is a regularization parameter, whose value should be set according to the amount of noise present in the image, and ∇f is the gradient of the edge map $f(x, y)$, which is derived from the image $I(x, y)$.

Using the calculus of variations, the GVF field is determined by solving the following Euler equations:

$$\mu\nabla^2 u - (u - f_x)(f_x^2 + f_y^2) = 0 \tag{3.11}$$

$$\mu\nabla^2 v - (v - f_y)(f_x^2 + f_y^2) = 0 \tag{3.12}$$

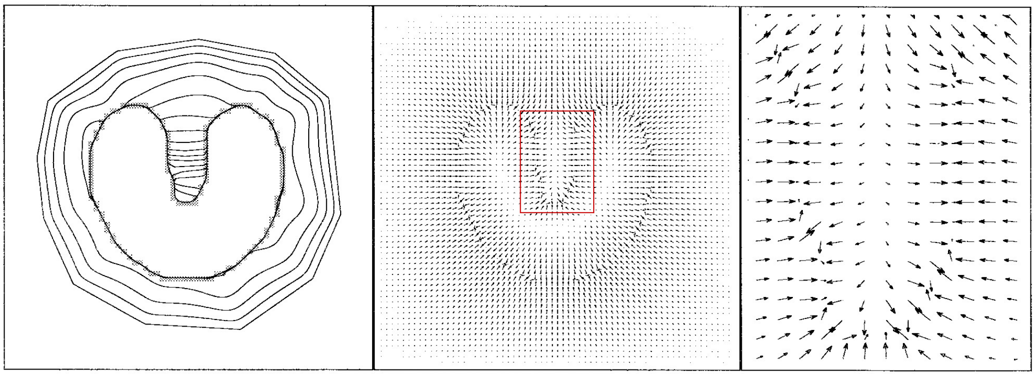
where ∇^2 is the Laplacian operator. By treating u and v as functions of time, equations 3.11 and 3.12 can be solved, yielding the following generalized diffusion equations:

$$u_t(x,y,t) = \mu\nabla^2 u(x,y,t) - [u(x,y,t) - f_x(x,y)]\left[f_x(x,y)^2 + f_y(x,y)^2\right] \tag{3.13}$$

$$v_t(x,y,t) = \mu\nabla^2 v(x,y,t) - [v(x,y,t) - f_y(x,y)]\left[f_x(x,y)^2 + f_y(x,y)^2\right] \tag{3.14}$$

A more detailed description of the numerical implementation along with the iterative solution to GVF is given in [132].

Figure 3.7 shows an example of a snake guided by the Gradient Vector Flow external forces. This simple example is designed to demonstrate an accurate active contour convergence to the boundary, particularly within the concave boundary region.



(a) (b) (c)

Figure 3.7: (a) Convergence of a snake using the GVF external forces shown in (b). A close view of forces within the concave boundary region is given in (c). (After [132].)

All the limitations mentioned in the previous section become even more intensive when the snake is applied on images of a real world scene. Such image sources usually contain noise, textured regions, light variations and low contrast, which set the snake's convergence to an even more complex problem. In order to obtain accurate snake results we suggest the Edge Combination approach in synergism with active contours. In our tests we experiment with the Gradient Vector Flow snake because of its advantages in converging to the desired boundary. The core of the object segmentation process is the Edge Combination approach, which employs disparity maps as an additional source of information in order to extract the dominant structure of the scene. A more detailed description of the Edge Combination algorithm can be found in Chapter 4, with object segmentation results being presented in Section 4.7.

3.3 Bézier Curves

Bézier curves were named after the French mathematician Pierre Bézier, who initiated their broad and practical usage in the wide community. Bézier curves and surfaces became extensively used in computer graphics, particularly in areas such as geometric modeling as well as animation and for stylized depiction.

The Bézier curve is a parametric curve whose degree depends on the number of control points used to form the curve. A polygon created by orderly connecting control points of the Bézier curve is called the control polygon of the Bézier curve. The position and the number of control points influence the curve, its shape and behavior. These characteristics make it easy to further manipulate the curve which means that the shape of the smooth Bézier curve can be adjusted simply through repositioning, adding or removing control points. In this way, with some graphical support, the designer can easily create and modify the curve by defining or adapting the control polygon without knowing the computation process behind. Figure 3.8 shows some examples of two-dimensional Bézier curves constructed using (a) four and (b) six control points. Besides the illustrated appearance of the curves, the figure also shows the control points used to construct these curves and their control polygons.

Given $n+1$ control-point positions $p_k = (x_k, y_k, z_k)$, the corresponding Bézier curve defined by these control points, as presented in [54], is:

$$P(u) = \sum_{k=0}^{n} p_k B_{k,n}(u), 0 \leq u \leq 1 \tag{3.15}$$

where the $B_{k,n}(u)$ are the Bernstein polynomials defined as:

$$B_{k,n}(u) = C(n,k) u^k (1-u)^{n-k} \tag{3.16}$$

In the above equation, $C(n,k)$ are the binomial coefficients given by:

$$C(n,k) = \frac{n!}{k!(n-k)!} \tag{3.17}$$

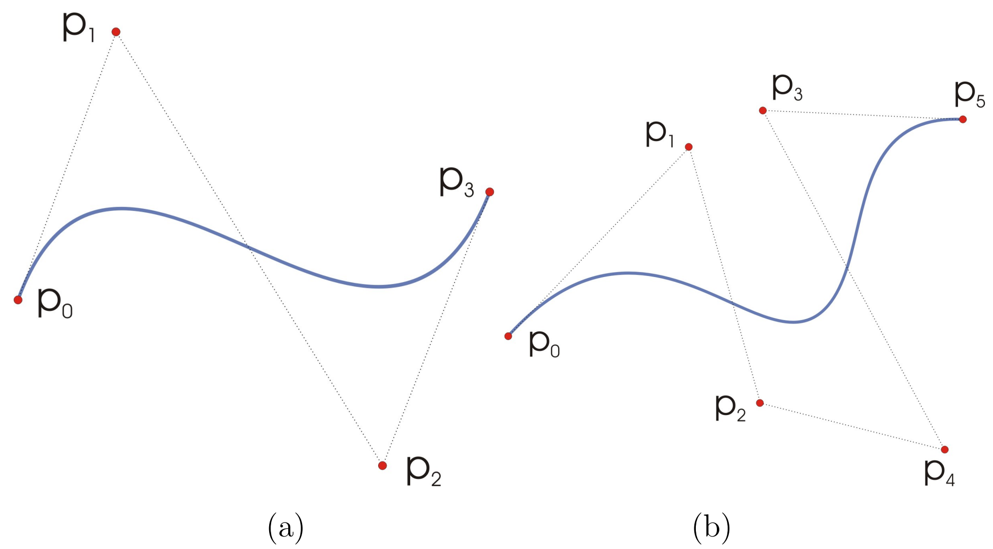


Figure 3.8: Bézier curve examples constructed using (a) four and (b) six control points.

Several Bézier curve properties make them very useful for many computer graphics areas. Some of the most important Bézier curve properties, as described in [54, 7], are:

- *Endpoint interpolation*

  The Bézier curve passes through the first and the last control point of its defining polygon, which can be written as:

$$P(0) = p_0 \tag{3.18}$$

$$P(1) = p_n \tag{3.19}$$

- *Endpoint derivatives*

  The first derivatives of a Bézier curve at the end points are:

$$P'(0) = -np_0 + np_1 \tag{3.20}$$

$$P'(1) = -np_{n-1} + np_n \tag{3.21}$$

  The second derivatives of a Bézier curve at the end points are:

$$P''(0) = n(n-1)[(p_2 - p_1) - (p_1 - p_0)] \tag{3.22}$$

$$P''(1) = n(n-1)[(p_{n-2} - p_{n-1}) - (p_{n-1} - p_n)] \tag{3.23}$$

- *Convex hull*
  The Bézier curve is contained in the convex hull defined by a polygon boundary that is created from the control points.

- *Variation diminishing*
  The number of intersections of a straight line with the Bézier curve is less or equal to the number of intersections of this line with the control polygon of the curve. This along with the previously described convex hull property is illustrated in Figure 3.9.

- *Affine invariance:*
  Bézier curves are invariant under affine transformations.

- *Global control*
  By modifying any of control points the entire curve is affected.

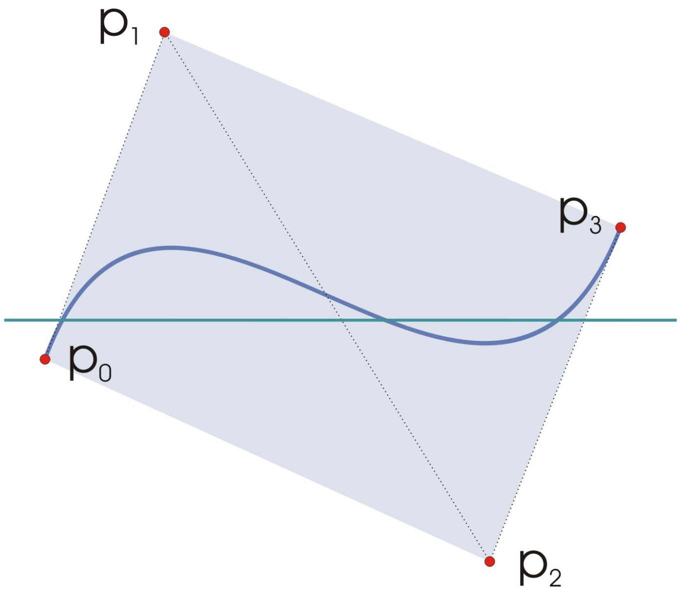


Figure 3.9: A cubic Bézier curve from Figure 3.8, shown with its convex hull and variation diminishing property.

Bézier curves of different degrees can be utilized in order to create shapes of various levels of detail. More detailed shape representation or creation requires a higher degree Bézier curve. From the Bézier curve definition, a higher degree of the curve can be achieved by employing a greater number of control points. There are many advantages of such a Bézier curve design. This way, for instance, a variety of shapes or a sketched-like stroke appearance can be produced, as we will show in Chapter 5. On the other side, higher degree Bézier curves are harder to control due to the global control property.

Another approach of creating more complex shapes is by using the piecewise Bézier curves form. Such a structure can be obtained by attaching several lower degree Bézier curves to each other. In order to ensure that two Bézier curves are continuous, different degrees of continuity can be utilized at the point where they join. This approach provides more control over the curve and it is particularly useful for the Bézier curve design and construction. On the other hand, the case of fitting a piecewise Bézier curve to a contour raises a problem of obtaining the optimal knots' location and their optimal number, as well as the problem of determining the continuity degree between the curve pieces at each knot.

In our application in Chapter 5, we present an algorithm to determine an optimal number and location of control points for higher-degree Bézier curves to represent significant contour edges in a stylized form. In particular, we show how the approximation by Bézier curves can be used to vary the thickness of the sketched-like contours in imitation of line drawings produced by hand.

Chapter 4

Edge Combination Algorithm

In this chapter, we will describe the main structure of the Edge Combination algorithm that we devised in order to find silhouettes, which delineate the principal structure of a natural scene. Images of real scenes usually contain texture, noise, or other features such as shadowing that set the accurate extraction of silhouettes to a complex task. A simple approach of using only the output of an edge detector for most real scenes' images induces the problem of distinguishing object edges from other features' edges. To overcome this problem and to extract only dominant edges by suppressing unnecessary lines detected by edge detection, we use depth information as an additional source of information. To obtain this information from images of a real scene, we use the depth-from-stereo approach. The output of the stereo matching process delivers a disparity map, which is inversely related to the depth. Whereas a perfect disparity map might be used to separate the object of interest from the background in a relatively straightforward way, in practical stereo analysis we have to cope with stereo matching errors that lead to erroneous disparity values, which are present particularly in the absence of texture and along object discontinuities. In order to suppress the matching-induced errors along object contours, we suggest an Edge Combination algorithm that combines the edges derived from the disparity image with the original edges for more accurate localization of contour edges.

4.1 Outline of the Algorithm

A summary of the involved processing steps can be seen in Figure 4.1. A stereo image pair is processed by a stereo matching algorithm, which delivers as output a disparity map in the geometry of one of the two input images. Next, an edge detector (e.g. Canny) is applied to the original image and the disparity image. The core of the processing pipeline of the Edge Combination algorithm is to determine which edges in the original edge image are also present in the disparity edge image. The idea is to combine the edges from the disparity edge image, which suggest the location of scene object discontinuities, with the higher positional accuracy of the edges from the original edge image. This combination allows more accurate recognition of significant contour edges.

In this context, a pixel-by-pixel search is performed to determine corresponding pixels in the original and disparity edge image. To reduce the possibility of mismatches, the search takes into account the orientation of the edge. The result of this step is further refined during an edge linking process that bridges minor gaps in the computed combined edges based on a labeling of the connected edge components in the original edge image. The result of the Edge Combination step is a set of edges that are located along depth discontinuities, but with the positional accuracy of edges in the original edge image. A detailed description of each distinct step of this procedure follows.

4.2 Edge Detection

The first step is to detect edges in both the original image and its corresponding disparity image, by using the Canny edge detector [20]. Apart from the pixel coordinates describing an edge pixel, we gather information about the edge orientation in the range [-$\pi/2$, $\pi/2$], estimated using the gradient in the x and y directions, for both the original and disparity edge image.

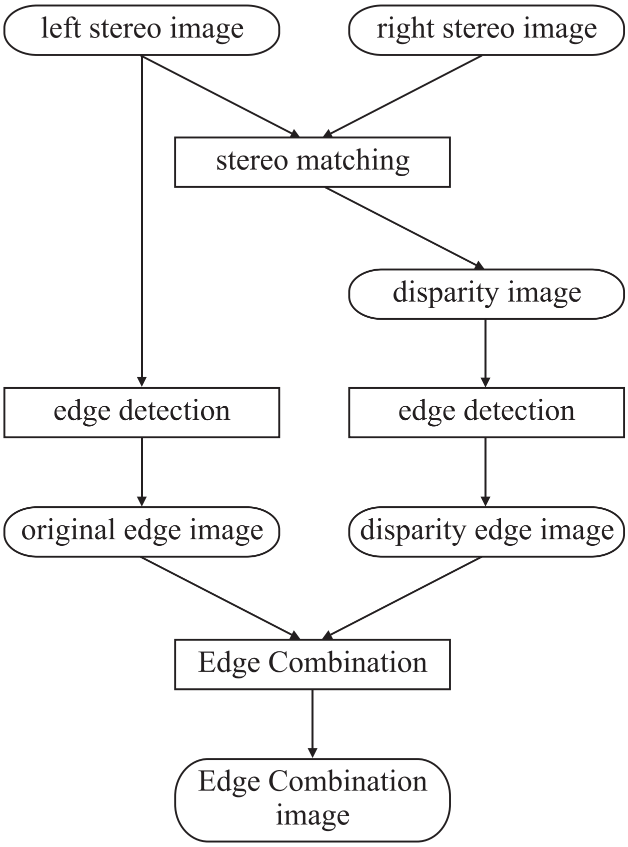
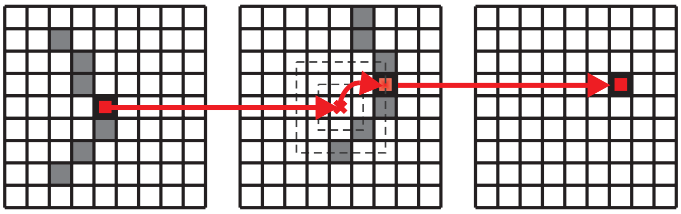


Figure 4.1: Overview of processing chain.

4.3 Edge Search

The original edge image (I_E) and the disparity edge image (I_{DE}) are input to the Edge Combination procedure, shown in Figure 4.2. For each edge pixel in I_{DE}, we determine whether a corresponding edge pixel with a similar orientation can be found in I_E. We use a search window of $N \times N$, with $N = 2k + 1$ pixels, where $k = 0, \ldots, k_{max}$ and k_{max} is an integer parameter. Note that k should be progressively increased from 0 to a user-defined maximum k_{max} until a matching pixel is found. To estimate the similarity in edge orientation we employ a tolerance angle, usually between $5°$ and $20°$. We record every edge pixel in I_E (Figure 4.2(b)) that was found to have a corresponding edge pixel in I_{DE} (Figure 4.2(a)), in order to include it in the

Edge Combination image (Figure 4.2(c)). In this way, we build up a "basic" Edge Combination image (I_{EC}).



(a) I_{DE} (b) I_E (c) I_{EC}

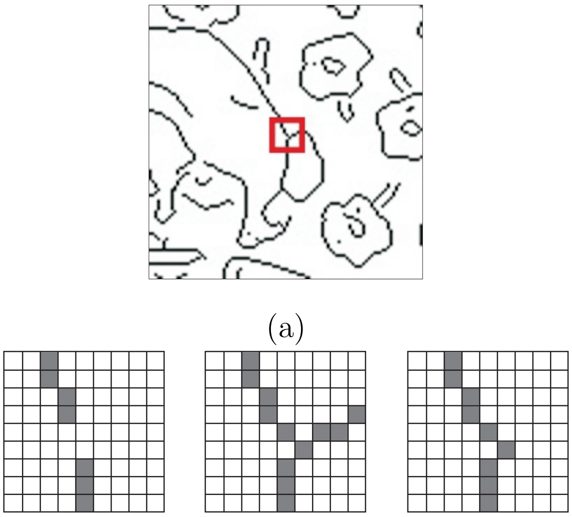
Figure 4.2: Searching process in constructing the "basic" Edge Combination image.

4.4 Edge Linking

Imperfect disparity information along continuous edges, usually sourcing from artifacts in the stereo matching process, will prevent some pixels in the comparing process to match, leaving a gap in the reconstructed contour line. An edge linking procedure is used to repair "broken" edges in I_{EC}, if a continuous edge in I_E indicates that edge segments should be connected. First, we use a labeling algorithm to determine the connected edge components in I_E. For each end point of an edge in I_{EC}, we search within a neighborhood of $M \times M$, with $M = 2k_L + 1$ pixels, where $k_L = 0, \ldots, k_{L_{max}}$, to find another end pixel in I_{EC}. If both of them belong to the same edge in I_E, as determined by the previous labeling, we connect the two end points in I_{EC} by inserting the corresponding edge segment from I_E. In practice, we copy an appropriate window of size $M_{max} \times M_{max}$, with $M_{max} = 2k_{L_{max}} + 1$ pixels, from I_E and insert it into I_{EC}. Before insertion, we clean the window by pruning superfluous parts of the copied edge pattern using the cleaning technique described in the following subsection. The edge linking procedure terminates, if there are no more open end points found in I_{EC} that could be connected.

4.5 Cleaning (Maze Solving Technique)

To remove redundant line parts from the window we identified as the one to be copied in the previous step, Figure 4.3(a), we use the Maze solving technique described in [89]. For every end pixel of the window as shown in 4.4(b), we check if it is one of the end pixels that we initially wanted to connect, marked in 4.4(a). If this test fails we remove that pixel from the window and continue testing the remaining end pixels, as shown stepwise in 4.4(b) through 4.4(h). This procedure terminates when we only find end pixels that have the same position as the pixels that we want to connect, as marked in Figure 4.4(h).

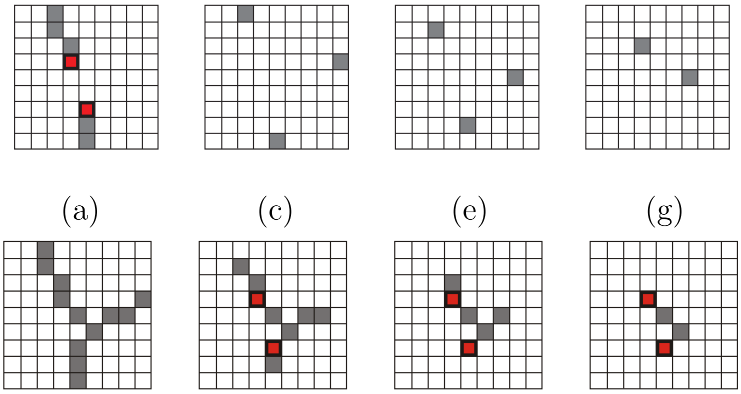

(a)

(b) (c) (d)

Figure 4.3: (a) Part of the original edge image, (b) Edge Combination image window before linking, (c) Edge Combination image window after linking and before cleaning, (d) Edge Combination image window after linking and cleaning.

4.6 Results

For our Edge Combination tests, we utilized the Pixel-to-Pixel [11] and the Graph-cut based [14] stereo matching algorithm in order to compute dense



(a) (c) (e) (g)

(b) (d) (f) (h)

Figure 4.4: Window cleaning.

disparity maps. In both cases, for the output of the Edge Combination algorithm - the Edge Combination image - we obtained similar results. For the sake of conciseness, we present here silhouettes extracted from the Edge Combination approach by using the result of the Pixel-to-Pixel stereo matching algorithm [11]. Additional results, obtained using also disparity maps of the Graph-cut based stereo matching approach [14], will be shown in subsequent sections of the thesis, when we present techniques for object segmentation and image stylization, which utilize the Edge Combination image as an important feature.

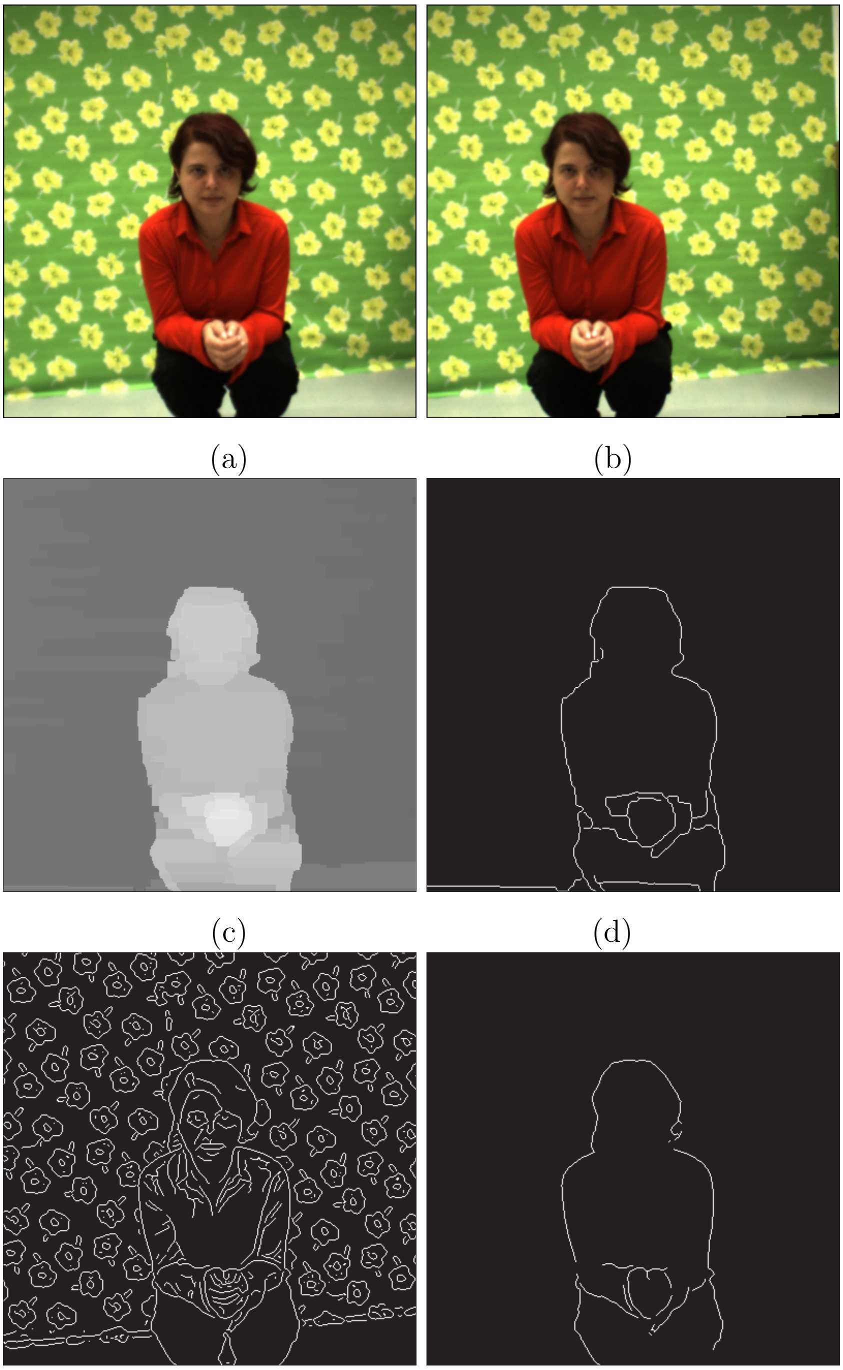
The results of the Edge Combination approach along with our test data are illustrated in Figures 4.5 through 4.7. Figures (a) and (b) show a pair of stereo video frames in epipolar geometry. The resulting disparity map is given in Figure (c). Figures (d) and (e) show the edges derived from the disparity image (Figure (c)) and the original image (Figure (a)), respectively. Figure (f) contains the silhouettes computed by the Edge Combination approach. Figure 4.5 particularly demonstrates the advantage of such an approach when dealing with a textured background. In this case, we show that the dominant edges were successfully located and selected among the other edges in the original edge image. In this, as well as in subsequent Figures 4.6 and 4.7, one can recognize the smoother appearance and the more accurate location of the Edge Combination edges in (f) when they are compared to the edges of the disparity edge image in (d).

4.7 Application to Object Segmentation

4.7.1 Motivation

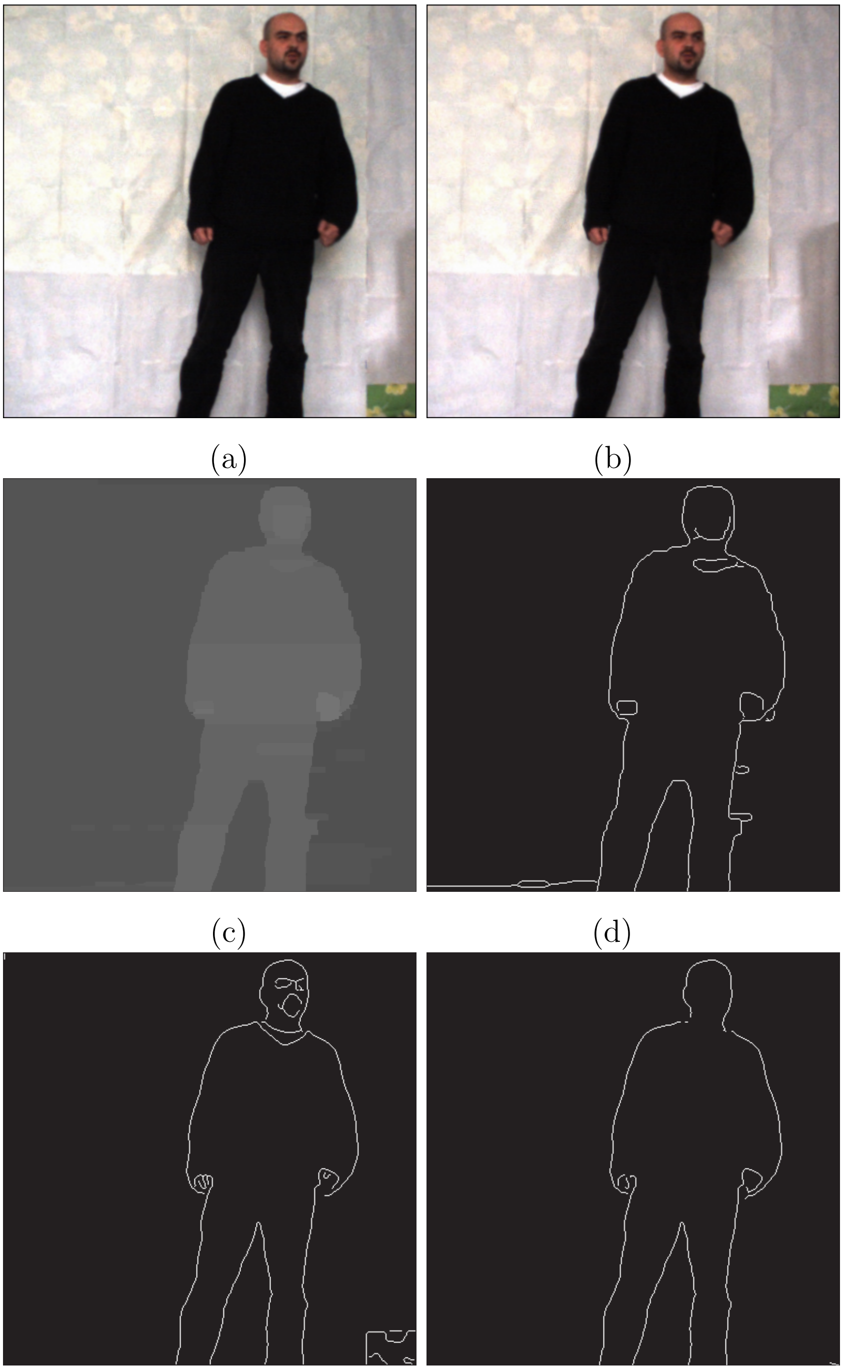
Information about objects present in a scene is crucial for many applications such as content-based multimedia retrieval, video editing and compositing, or mixed reality compositions. A necessary first step for obtaining this information is object segmentation. The significance of extracting the precise information about the object's location and outline in the scene has initiated the development of a considerable number of approaches. The extraction process, usually trivial for the user, can be extremely complicated for a machine to achieve in a robust and accurate fashion. This fact resulted in the development of several systems that utilize user interaction. An often used approach for object segmentation from the image is based on active contours, also known as snakes. The main principles of active contours were already described in Section 3.2.

The task of active contours, moving under the influence of internal and external forces, is to stick to the object in order to extract the desired information about the boundary between the object and the background. However, despite years of research, this remains a very challenging task. Contour-based approaches often have difficulties when dealing with images of a natural scene, mostly because of highly textured regions or low contrast. Recordings of a natural scene usually contain noise, variations in illumination such as shadows and reflections or textured regions that can prevent the active contour from converging to the object boundary. To overcome these problems, the data should be simplified by removing the irrelevant information. Ideally, the simplification process should clearout the complex textured areas or remove unnecessary details without damaging the object boundary information.



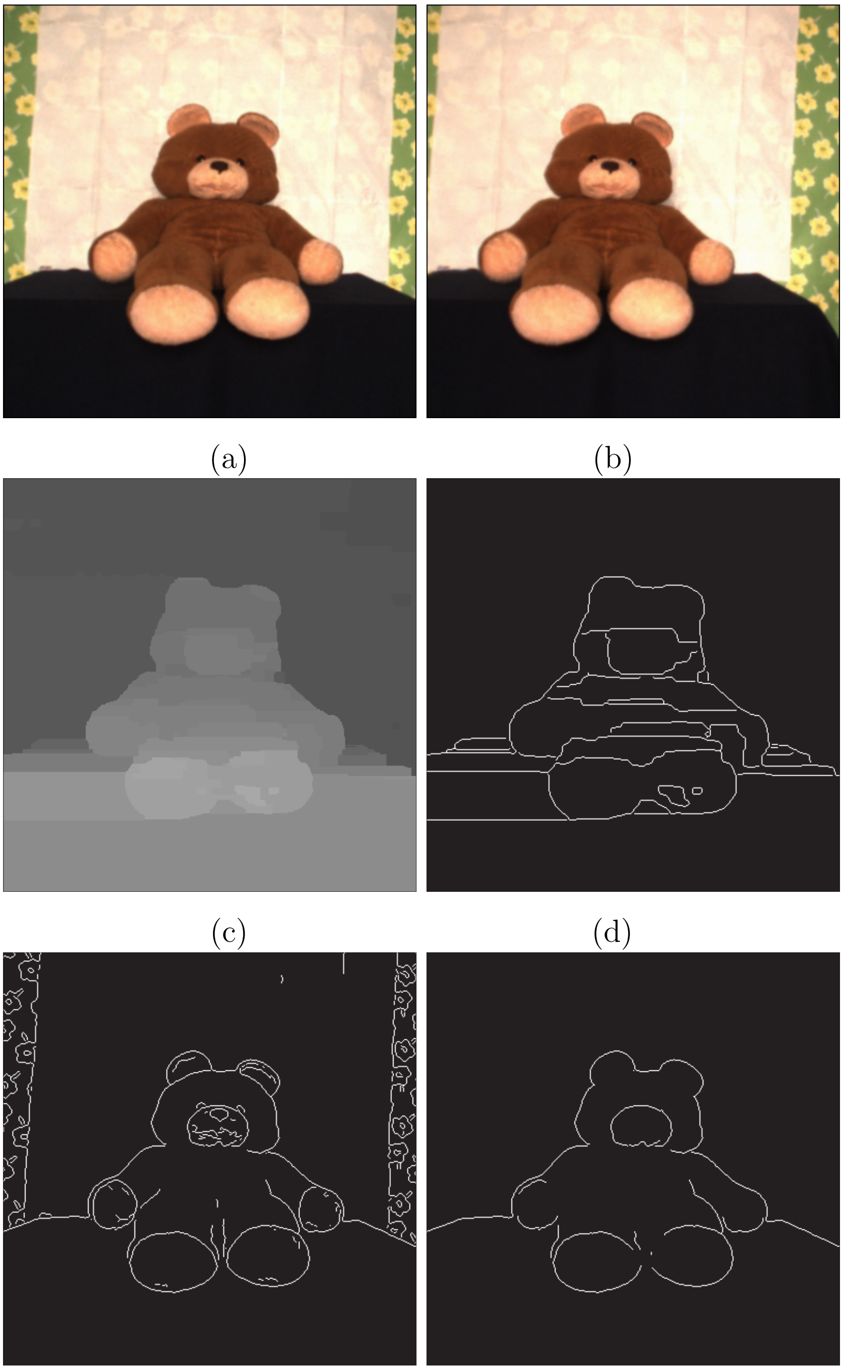
(a) (b) (c) (d) (e) (f)

Figure 4.5: Edge Combination example 1: (a) Left camera image, (b) right camera image, (c) disparity image, (d) disparity edge image, (e) original edge image, (f) Edge Combination image.



(a) (b) (c) (d) (e) (f)

Figure 4.6: Edge Combination example 2: (a) Left camera image, (b) right camera image, (c) disparity image, (d) disparity edge image, (e) original edge image, (f) Edge Combination image.



(a) (b)

(c) (d)

(e) (f)

Figure 4.7: Edge Combination example 3: (a) Left camera image, (b) right camera image, (c) disparity image, (d) disparity edge image, (e) original edge image, (f) Edge Combination image.

To filter out “problematic” information, one of the most valuable sources of information about the objects present in the scene is the depth information. Employing disparity information, which is inversely related to depth and provides a simpler representation of the original image, gives a robust additional cue for object segmentation and analysis. The idea to take advantage of disparity information to perform object segmentation initiated the development of several methods.

For example in [65], Izquierdo presented a stereo-matching algorithm based on block matching with a local adaptive window. The contour of the initial object segmentation, based on disparity information, is further matched to the edges extracted from the original image in order to obtain a better positioned object contour. Woo [129] suggested an object segmentation framework where the initial object segmentation map is obtained by using a background subtraction process and further refined by combining several cues such as intensity, intensity edge, disparity and disparity edge. The idea that we present here is to extract the silhouette information first using the Edge Combination algorithm and to employ it in synergism with active contours in order to precisely segment the object of interest from the rest of the scene.

4.7.2 Experiment

In our experiments, we used the obtained Edge Combination images, computed by utilizing the previously explained Edge Combination algorithm. In order to segment objects from the background, we employed the Gradient Vector Flow (GVF) snake. The GVF snake, as we described in more detail in Chapter 3.2.4, brings particular advantages over the traditional snake such is the ability to move into boundary concavities. We applied the GVF snake on the original intensity image from the stereo image pair, on the computed disparity image and on the corresponding Edge Combination image. In the following section we give the results obtained in our experiments.

4.7.3 Object Segmentation Results

The results of our experiments demonstrate that the proposed Edge Combination approach used in conjunction with active contours can significantly improve the segmentation results, especially in textured regions, where snakes often fail to produce satisfactory results.

Figures 4.8 and 4.9 illustrate the results produced by applying the GVF snake to the original, disparity, and Edge Combination image. The snake initialization shown in (a) and the parameters used for the snake computation were the same in all three cases. The results can be compared in (b), (d), and (f). The active contour computed on the original image in (b) shows the obvious errors caused by the background texture. Clearly, the background pattern pulls away the snake from the object of interest at several locations, which leads to poor segmentation results. More snake iterations resulted in even larger deviations between the computed and actual shape in (b). These errors are no longer present in the snake result from the disparity image in (d). However, because of imperfectness of the stereo matching results, the final position of the GVF snake in (d) does not coincide exactly with the boundaries of the object. The errors in (b) and (d) are largely suppressed by the Edge Combination approach, as demonstrated by the almost perfect fit of the snake in (f).

We carried out more experiments with other test data and obtained similar results. In all cases, the Edge Combination image produced a better snake segmentation than the intensity image or disparity image alone, which demonstrates the usefulness of the combined approach.

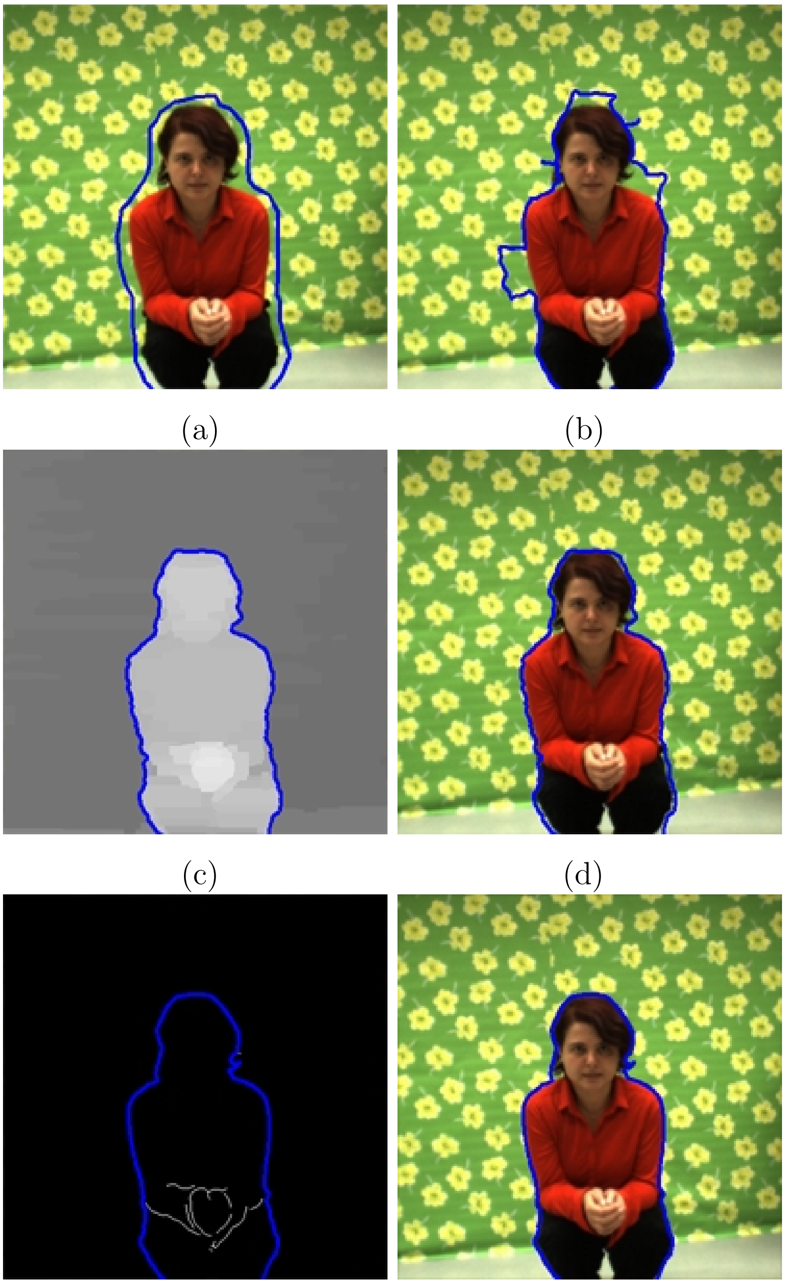
4.8 Summary

In this chapter, we have described the main steps of the Edge Combination algorithm. This approach aims to take the advantage of positive sides of both, the original and disparity edge image.

The edges from the disparity edge image suggest the location of scene object discontinuities. However, due to disparity artifacts induced by stereo matching errors, depth discontinuities may be displaced. In order to suppress the matching-induced errors along object contours by utilizing the higher positional accuracy of edges from the original edge image, we suggest the Edge Combination algorithm. In this way, final silhouettes extracted using the Edge Combination approach can be described as depth discontinuities "shifted" to the proper position. Using the edge detection to identify edges in both, the original and disparity image, gives the ability to the user, by tuning parameters, to specify the desired level of detail in conveying the final dominant scene structure.

In this chapter, we also described an object segmentation approach that utilizes the synergism of the Edge Combination algorithm with the Gradient Vector Flow snake. The presented results demonstrate the usefulness of such a joint approach to perform accurate object segmentation.

Through the following chapters, we will show the utility of the Edge Combination approach for image stylization, in which object contours play an important role. This approach gives the opportunity of utilizing natural scenes of the real world in simulating the traditional and in creating new art forms as an alternative to traditional 3D modeled scenes.

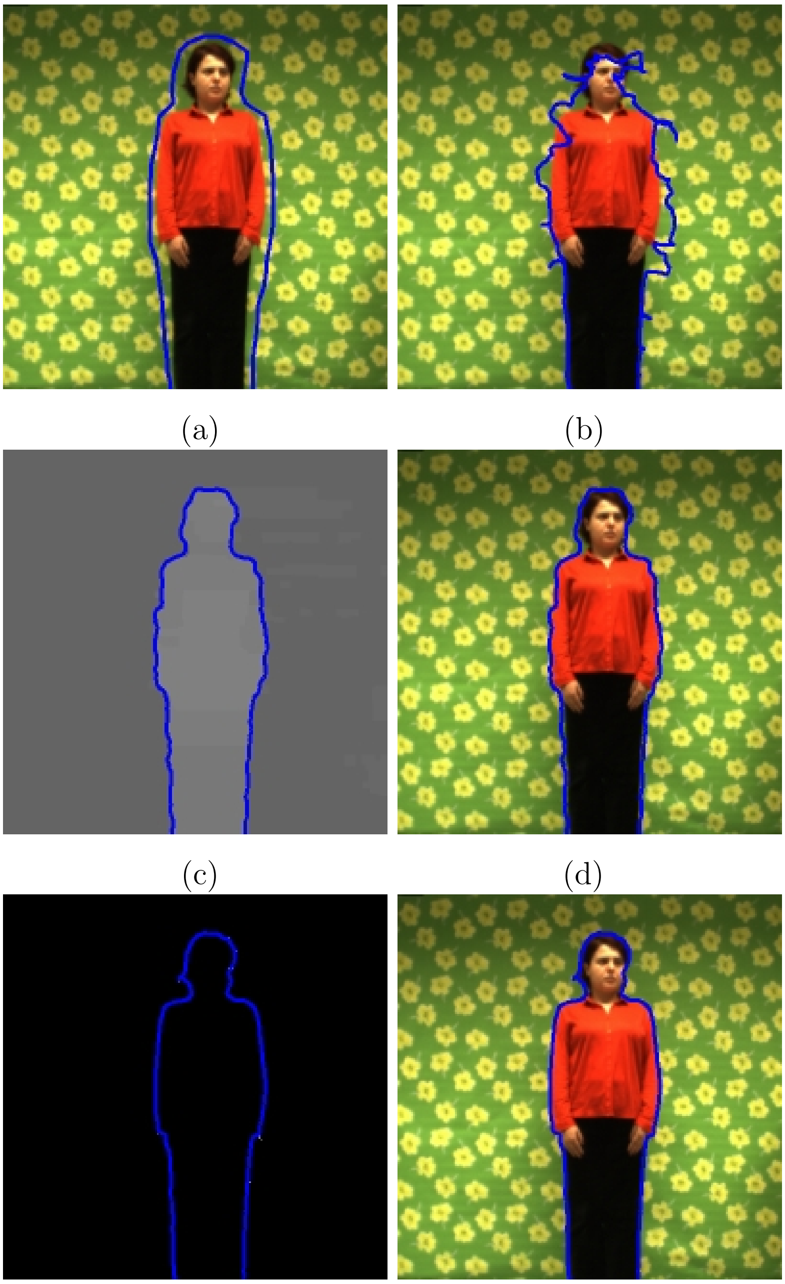


(a) (b)

(c) (d)

(e) (f)

Figure 4.8: Object segmentation example 1: (a) Original image with snake initialization, (b) final snake on original image, (c) final snake on disparity image, (d) original image with snake from (c) overlaid, (e) final snake on Edge Combination image, (f) original image with snake from (e) overlaid.



(a) (b) (c) (d) (e) (f)

Figure 4.9: Object segmentation example 2: (a) Original image with snake initialization, (b) final snake on original image, (c) final snake on disparity image, (d) original image with snake from (c) overlaid, (e) final snake on Edge Combination image, (f) original image with snake from (e) overlaid.

Chapter 5

Sketching by Parameterized Silhouettes

5.1 Introduction

5.1.1 Motivation and Overview of Algorithm

Sketching is an important form of human expression and idea communication. Defined as a quick drawing that loosely captures the appearance of important scene features, it is often used as a study in preparation for larger, more detailed works of art. Sketching has also been the primary tool in the learning process of visual fine arts. Furthermore, its importance becomes apparent in most activities of collaborative work where planning, recording, developing, visualizing and sharing ideas graphically can enhance human communication. More than a labor intensive drawing, sketching makes the information intimate and gives dominance to line over mass.

Although sketching skills can vary from one person to another, almost everybody is able to sketch and understand a sketch drawn by someone else. Simple self-made drafts in our everyday life serve as aids for problem analysis, solution presentation, memory support and communication. Despite its expressiveness and intuitiveness, sketching has not yet become a primary means of interaction with computer systems. Furthermore computers cannot readily communicate information back to humans in a sketched form, and usually they need to abstract complex information that is present in digitized images.

To enable computers to generate graphical sketches of complex data, a lot of research in computer graphics has been devoted to the simulation of the human sketching ability [30, 127, 85, 50, 59, 23, 90, 123]. In this area of research, very natural and often used features are silhouettes because of their important role in object depiction [107, 34, 30, 92, 78, 98]. They illustrate objects enough to stimulate imagination or allow further stylization.

Our focus is to present real scenes as stylized sketches through silhouettes by employing disparity maps to select only significant contours that play the most important role in object description. For this we utilize the Edge Combination algorithm to highlight only dominant edges in an image and to suppress unnecessary features.

The dominant structure of an object's appearance in a scene is depicted by using higher degree Bézier curves to achieve smoothness and simplify the representation of usually complex shapes. As a starting point to our conversion algorithm we use ideas from Bhuiyan and Hama [10] to transform the rasterized contours into a vector graphic representation. Our algorithm strives for economy in data representation by selecting the degree of the Bézier curves that best suits the approximation of each edge line, and also improves the performance by using adaptive parameters. The input to the sketching algorithm is a natural scene, along with a user-defined set of parameters that affect the quality of the image to be produced. The output is a vector representation of the scene with objects emphasized by their dominant edges.

In order to create such a smooth, abstract and descriptive representation of a natural scene that can be stylized to have a handcrafted-like appearance, we have to transform the captured input images to a more appropriate form for further processing and stylization. This procedure can be regarded as a raster-to-vector conversion.

5.1.2 Raster-to-Vector Conversion

Vectorization, i.e. raster-to-vector conversion, is the process of transforming raster images to a more suitable vector form for further viewing, editing or processing. Widely used in document analysis and recognition, this problem has been an important and challenging research topic for many years, which has resulted in a number of techniques proposed to solve it.

The input to this process, a raster image, is usually a recorded or scanned picture of a certain amount of detail, dependent on the image size and resolution. In order to produce a more appropriate form for further manipulation, depending on the image information different strategies are employed.

In case of an input line image (e.g. scanned engineering drawings), the usual vectorization procedure is applied through steps of skeletonization, tracing and approximation [22]. The task of skeletonization is to convert lines of the line image to a one pixel wide skeleton form without affecting the general shape of the line. To obtain such a representation, thinning [71] or non-thinning [73] based techniques can be employed. After the tracing step, which is a line tracking procedure that links the skeleton pixels to chains, the obtained information is used for the approximation. The result of the approximation is a simplified, smooth representation, appropriate for further processing.

Another case is the raster image which contains objects or regions with descriptive outline information that should be vectorized (e.g. geographical images, calligraphic text characters). For this image category, the important contour information is obtained by finding the outline of the object/region.

Recent research shows strong interest in the development of algorithms for vectorization of complex color and intensity raster images (e.g. photographs) [95, 8]. The aim of this research is to transform the inflexible raster photographic imagery to the scalable and resolution independent vector representation.

In our work, due to requirements of the stylized abstract scene representation, the vectorization process consists of several steps. A stereo image pair of a natural scene, transformed into epipolar geometry, is processed by a stereo matching algorithm in order to compute a disparity map. The obtained disparity map is further utilized to find silhouettes through the Edge Combination image computation. The dominant scene structure, maintained in the Edge Combination image, is used to impart an abstract scene representation.

After the extraction of edges using the Edge Combination algorithm, we approximate them by employing Bézier curves. This parametric representation of the contours allows flexible manipulation of the extracted sketching lines; it also acts as a line stylization mechanism, since the data is fitted smoothly, providing a distinct and artistic character to the otherwise rigid edges. The different steps of the algorithm are explained in more detail in the following sections.

5.2 Converting Raster Lines to Bézier Curves

The edges in the final Edge Combination image I_{EC} (see Figures 4.5 - 4.7) are properly localized to present significant edges. However, a bitmap representation of these lines is usually not sufficiently flexible for stylization. Therefore we convert the rasterized contours into a parametric representation by fitting higher degree Bézier curves. Our algorithm transforms complex to much simpler contours by selecting the degree of the Bézier curves that best suits the approximation of each edge line, while efficient performance is achieved by using adaptive parameters. The concept of our algorithm is to initiate a Bézier curve and by adjusting its shape to minimize the distance between the original edge and the approximating curve. To adjust the shape we move the control points and calculate the error, accepting iteratively the new location of the control points if they generate a curve where the estimated error is smaller than the previous one.

5.2.1 Initialization

To position the initial control points of an approximating curve (A), we use a smooth representation (B) of the respective edge by considering all its pixels as Bézier curve control points, with a and b points of the sets A and B respectively. This representation is used to determine the direction and curvature at each point, as explained in [75] and [48]. The direction β_d of the curve at the nth point is determined as:

$$\beta_d(n) = \arcsin\left(\frac{\Delta b_y(n)}{\sqrt{\Delta b_x(n)^2 + \Delta b_y(n)^2}}\right) \tag{5.1}$$

where

$$\Delta b_x(n) = b_x(n+f) - b_x(n-f) \tag{5.2}$$

$$\Delta b_y(n) = b_y(n+f) - b_y(n-f) \tag{5.3}$$

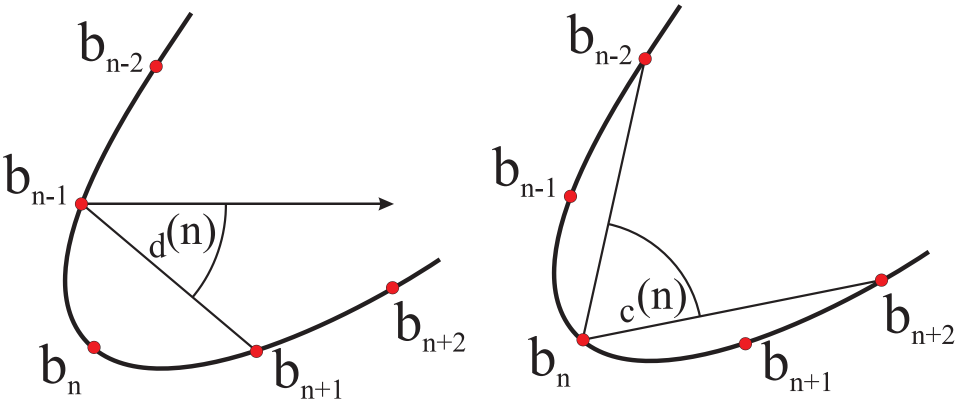
and f is a small integer used to define a neighborhood size around a point. The curvature β_c is then calculated by the angle between two elementary segments:

$$\beta_c(n) = \beta_d(n+f) - \beta_d(n-f) \tag{5.4}$$

as presented in Figure 5.1. This is further used for the detection of extreme points (P_d) by selecting only the regional extrema within each region above a threshold value. By computing the mean value of β_c we determine this threshold value, which can be additionally adjusted by using a scale factor. An example rasterized curve is presented in Figure 5.2(a). The same curve and its smooth representation are shown in Figure 5.2(b), together with the detected dominant points (P_d).

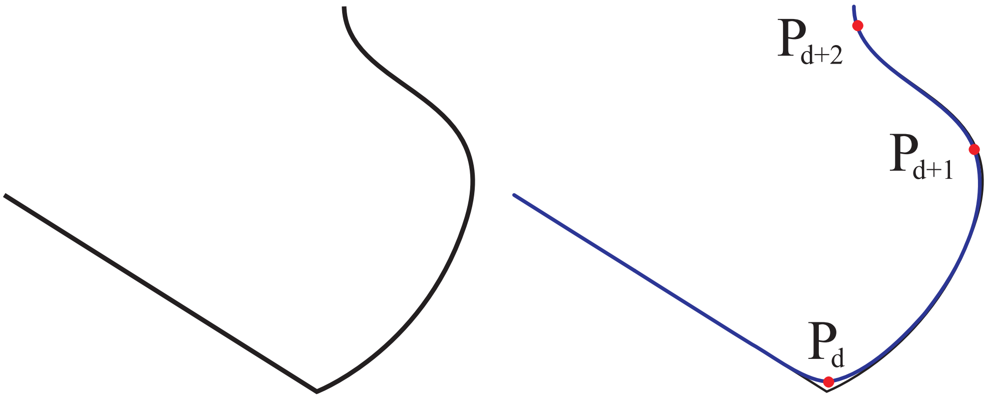
5.2.2 Creating Sketchy Line Segments

The Bézier curves produced by using the above initialization strategy may require further treatment in order to achieve the handcrafted appearance for



(a) (b)

Figure 5.1: Direction and curvature estimation where $f = 1$.



(a) (b)

Figure 5.2: Example of (a) a curve with (b) its Bézier curve representation and dominant points.

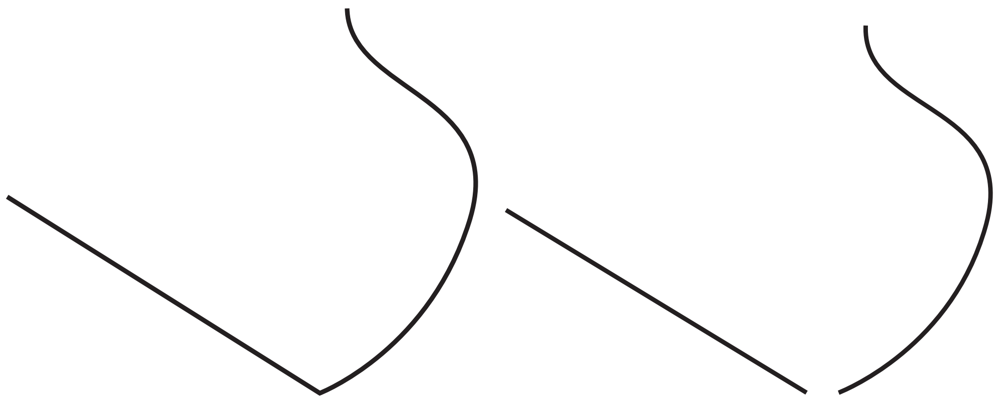

(a) (b)

Figure 5.3: (a) The curve from Figure 5.2, (b) result of the simplification.

which we are aiming. When sketching by hand, it is common to represent long lines by smaller segments that perceptually approximate the desired shape, instead of crafting a long continuous contour. Furthermore, we observe that it is more natural for humans to subdivide long lines where local extrema (e.g. corners) are present. To fulfill these requirements we establish a criterion that can be user-adjusted to define the sensitivity of the algorithm to subdivide the initial curves. For this, we use the surrounding points of each dominant point, detected previously, and while operating on the original rasterized curve we repeat the previous procedure of the direction and curvature determination. Dominant points with a change in curvature greater than a user specified threshold value are considered as points of the curve where it should be divided. The result of such a curve splitting by using breakpoints is illustrated in Figure 5.3. In this example, the original continuous curve was split up at the location of the dominant point P_d from Figure 5.2.

5.2.3 Initial Control Points

We place an initial control point $P_d^{'}$ at a distance of sH_t, in the direction of the height vector H_t of a triangle formed by the dominant points P_{d-1}, P_d and P_{d+1}, as shown in Figure 5.4. We scale the height vector H_t by the factor

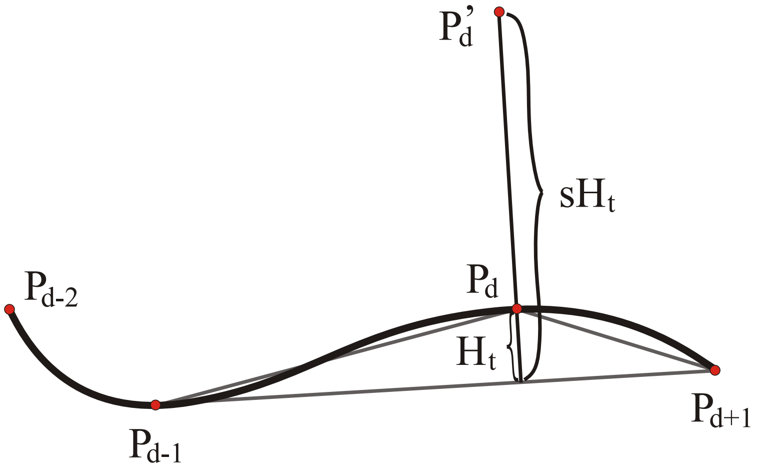


Figure 5.4: Control points initialization.

s, in our experiments usually 4, to shape the curve similarly to curve (B) in order to decrease the computation of the subsequent searching process.

5.2.4 Curve Approximation

To approximate the edge representation with the initialized Bézier curve, we move all the intermediate control points to new locations to minimize the area between the curves A and B. For each intermediate control point $P'_d(x, y)$, we translate it to four possible new locations:

(a) $P'_d(x + e, y)$,

(b) $P'_d(x - e, y)$,

(c) $P'_d(x, y + e)$,

(d) $P'_d(x, y - e)$,

where e is the moving distance. This adaptive parameter is associated with an error metric H according to:

$$e = m \cdot H(A, B) \tag{5.5}$$

where m takes values in each iteration from m_{max} (usually 2) to 0 with the parameter step defined by the user (usually 1). $H(A, B)$ is the error metric which represents the largest distance between the two curves calculated by using the general definition of the Hausdorff distance described by Rote [100], given as:

$$H(A, B) = \max\left(h(A, B), h(B, A)\right) \tag{5.6}$$

where $h(A, B)$ is the Hausdorff distance from A to B presented as:

$$h(A, B) = \max_{a \in A} \min_{b \in B} d(a, b) \tag{5.7}$$

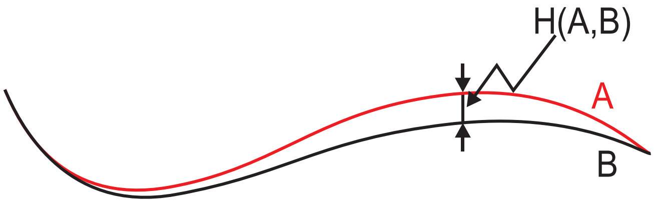
where we take $d(a, b)$ as the Euclidean distance between points a and b of sets A and B, respectively.

Whenever the area between the two curves is decreased by a movement, we accept the new control point location only if the computed value of $H(A, B)$ is reduced. We reiterate this procedure until a maximum number of user-defined iterations has been reached, or the error has become sufficiently small, under a user-defined threshold.

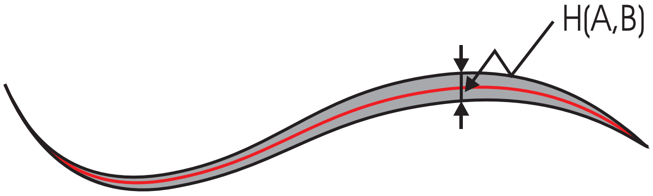
5.3 Line Stylization

We have so far extracted a set of dominant edges from the original scene and converted them into a parametric form suitable for further stylization. A number of algorithms have been proposed to help a user-artist breathe life into simple lines that represent a piece of artwork. In this manner Pudet [97] employs stylization according to the input of a pressure-sensitive stylus on a digitizing tablet, and Curtis [30] produces a sketchy stroke utilizing the particles generated from the template image, guided by a force field. Northrup and Markosian [92] draw each stroke in a style defined by the user, randomly perturbed and with texture attributes. Su et al. [119] use interval splines, Saito et al. [106] vary the thickness of the line according to its curvature and length.

In our work, we stylize smooth curves using available data obtained in the previous step, which we utilize to imitate stroke pressure. For this we use the maximum estimated error, which is the Hausdorff distance as described in the previous section, at the point of the approximated curve at which we estimated this error, as illustrated in Figure 5.5(a). We then interpolate this pressure (i.e. stroke thickness) along the stroke path, setting the pressure to zero at the starting and ending points of the curve. This procedure produces a stroke tapering effect, as can be seen in Figure 5.5(b). Using the mechanism described we obtain the results shown in the following section.



(a)



(b)

Figure 5.5: (a) Final maximum error measured by the Hausdorff distance between the approximated curve A and curve B, (b) final style of the line.

5.4 Experimental Results

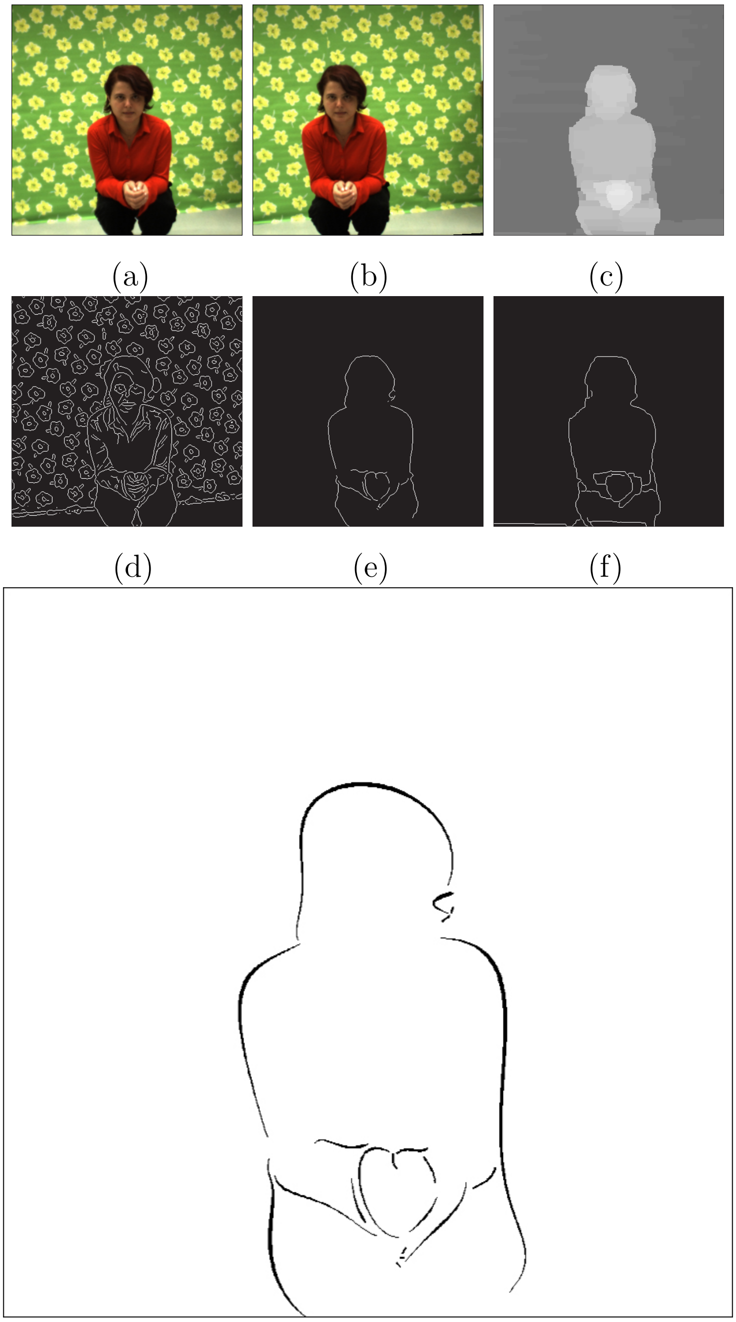
The application of the method to our test data and the results obtained are illustrated in Figures 5.6 through 5.8. Throughout these examples, Figures (a) and (b) show a pair of stereo video frames in epipolar geometry, which we use as input to our algorithm.

For our tests, we utilized the Pixel-to-Pixel stereo matching algorithm described by Birchfield and Tomasi [11]. The resulting disparity image is given in Figure (c). Figures (d) and (f) show the edges derived from the original image (a) and the disparity image (c), respectively. Figure (e) contains the contour edges computed by the Edge Combination approach. Our silhouette-extraction Edge Combination algorithm filters out only those edges from (d) that are needed to describe the dominant scene structure, while preserving the position, proportion and shape of the objects. In this way it provides enough information for the final sketches creation using a small number of simple expressive strokes. It abstracts the scene by omitting unnecessary details.

Figure (g) shows the final sketch result with strokes of dominant edges of the scene computed by the Edge Combination algorithm, represented by the previously explained approximation procedure and stylized to imitate artistic stroke pressure. We achieved a hand-drawn appearance by allowing the line thickness vary similarly to real life pens as explained in Section 5.3. Lines with small curvature along their paths tend to produce thin lines, whereas more curved lines are usually represented by strokes that vary in thickness, comparable to the mark a human directed pen would leave while slowing down to follow the curvature. The final results in Figures 5.6, 5.7 and 5.8 illustrate a real scene by using the dominance of significant lines artistically highlighted as a smooth hand-drawn effect.

5.5 Summary

In this chapter we presented a curve approximation technique based on higher degree Bézier curves to convert these rasterized edges of the Edge Combination image into a parametric form. The parametric edges outlining meaningful objects in the scene are then stylized by using strokes to produce sketches that have a handcrafted appearance. The smoothness of the stylized output

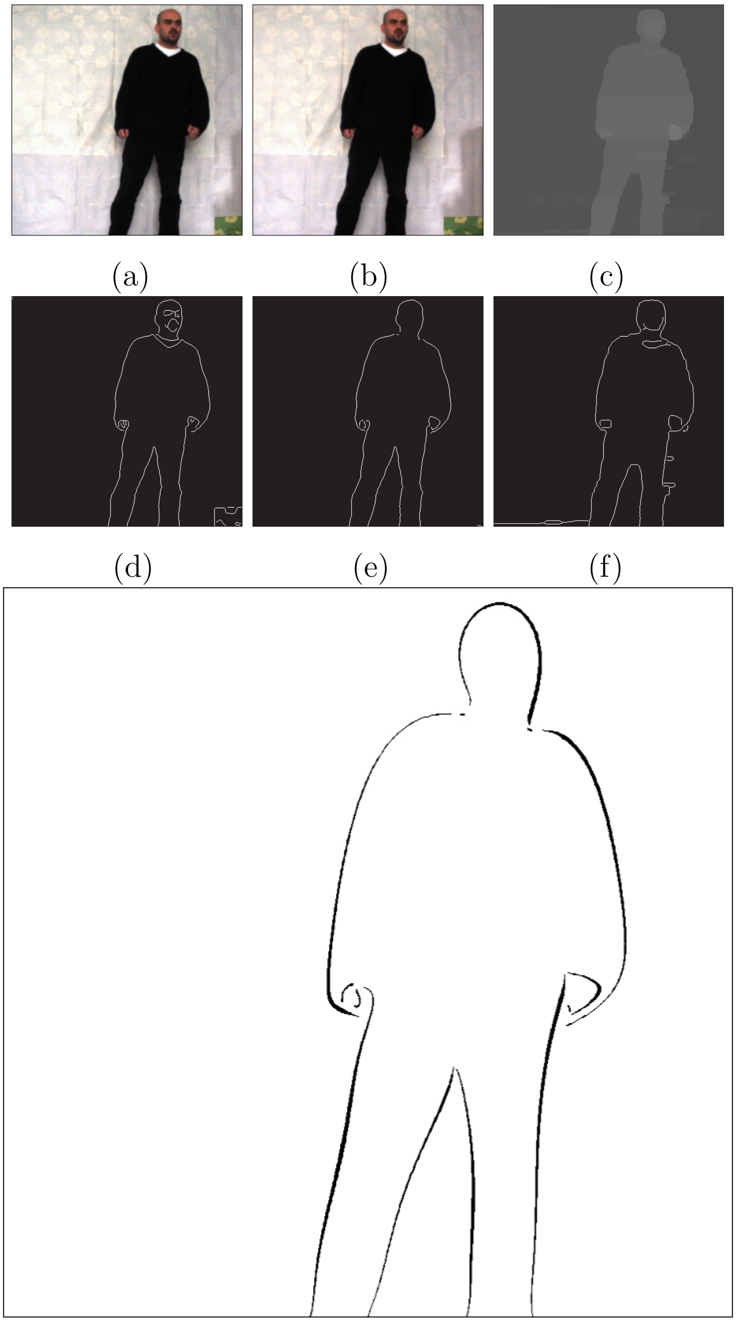


(a) (b) (c)

(d) (e) (f)

(g)

Figure 5.6: Example 1 of the sketched-like image representation: (a) Left camera image, (b) right camera image, (c) disparity image, (d) original edge image, (e) Edge Combination image, (f) disparity edge image, (g) sketched image.

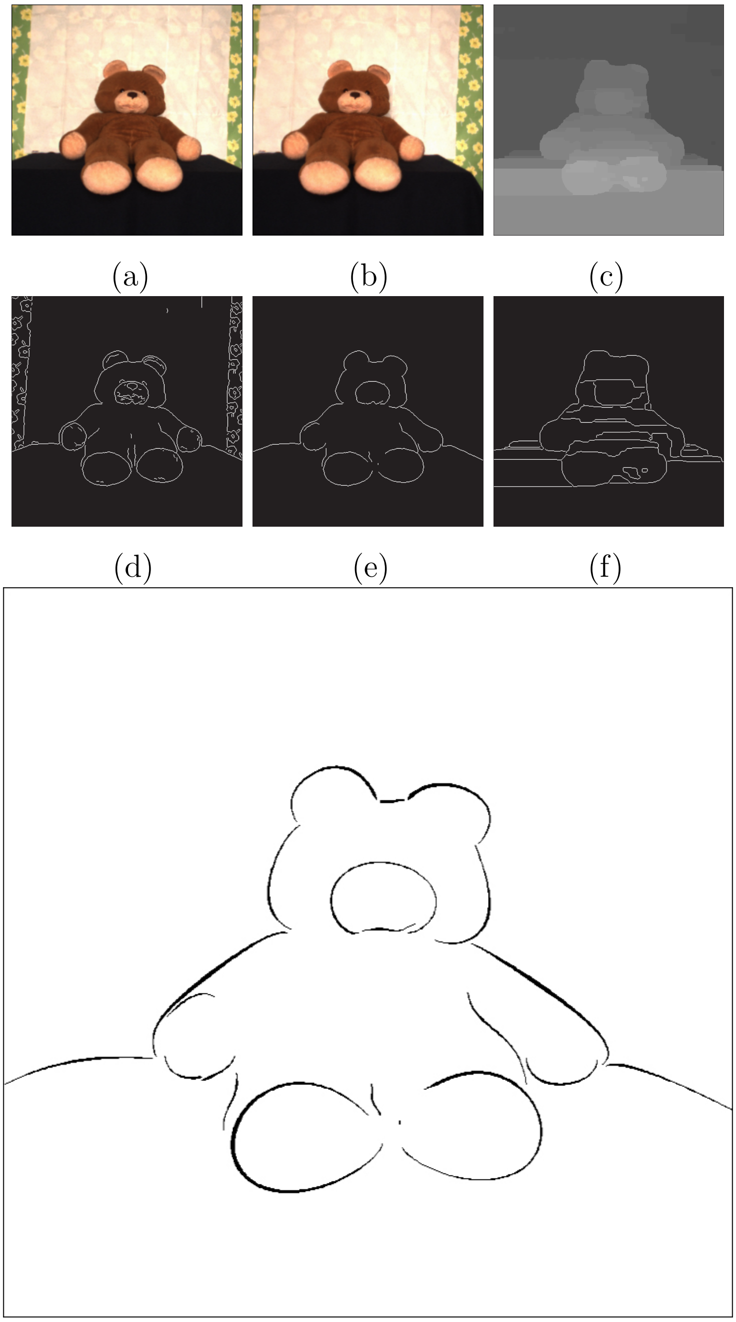


(a) (b) (c)

(d) (e) (f)

(g)

Figure 5.7: Example 2 of the sketched-like image representation: (a) Left camera image, (b) right camera image, (c) disparity image, (d) original edge image, (e) Edge Combination image, (f) disparity edge image, (g) sketched image.



(a) (b) (c)

(d) (e) (f)

(g)

Figure 5.8: Example 3 of the sketched-like image representation: (a) Left camera image, (b) right camera image, (c) disparity image, (d) original edge image, (e) Edge Combination image, (f) disparity edge image, (g) sketched image.

is similar to that found in many concept drawings and sketches. The algorithm produces aesthetically pleasing results with artistically highlighted dominant lines. It can further assist an artist in creating the basis for a fine art work or it can help a beginner artist as a guiding tool on important features of a scene when learning visual fine arts.

Chapter 6

Drawn-like Image Representation

6.1 Introduction

In this chapter we present a novel algorithm for depicting images of a natural scene in a drawn-like form, similar to traditionally created drawings. Input to the algorithm is a stereo image pair of a real scene. This stereo image pair is used for a disparity map computation. The obtained disparity map serves to provide additional information which is used to attach to the final image result a feeling of "depth". In order to better convey shapes of objects we developed an image enhancement step. The result of the image enhancement along with the information obtained from the disparity image is used to arrange strokes and to facilitate the comprehensible depiction of a natural scene. Furthermore, silhouettes extracted using the Edge Combination approach are utilized to outline and highlight the dominant scene structure. The output of such an approach resembles the form of handcrafted drawings, as we will show in following.

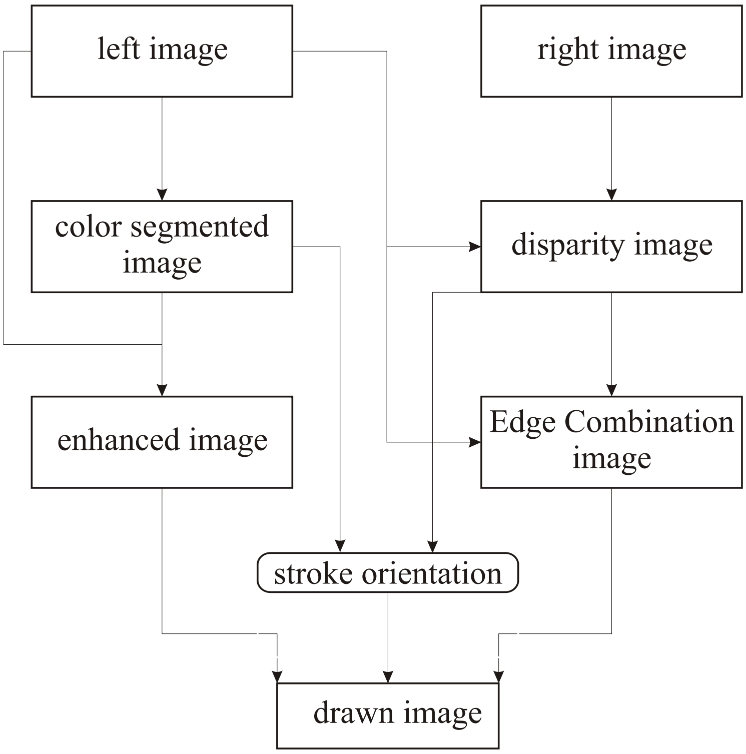


Figure 6.1: Overview of processing steps.

6.2 Algorithm

Figure 6.1 gives a short summary of the involved processing steps. A stereo image pair consisting of a left and right stereo image is processed by a stereo matching algorithm, which delivers a disparity map in the geometry of one of the two input images as output. The disparity map is described by a set of disparity layers, as a result of modeling the 3D scene by a set of planar facets [14]. For further processing, we enhance the reference image by using its color segmentation result. The color segmented image along with the orientation of the disparity layers gives the information to orient strokes. To highlight the dominant structure of the scene we use the Edge Combination algorithm for more accurate recognition of significant contour edges. The various processing steps are described in more detail in Subsections 6.2.1 through 6.2.4.

6.2.1 Image Segmentation

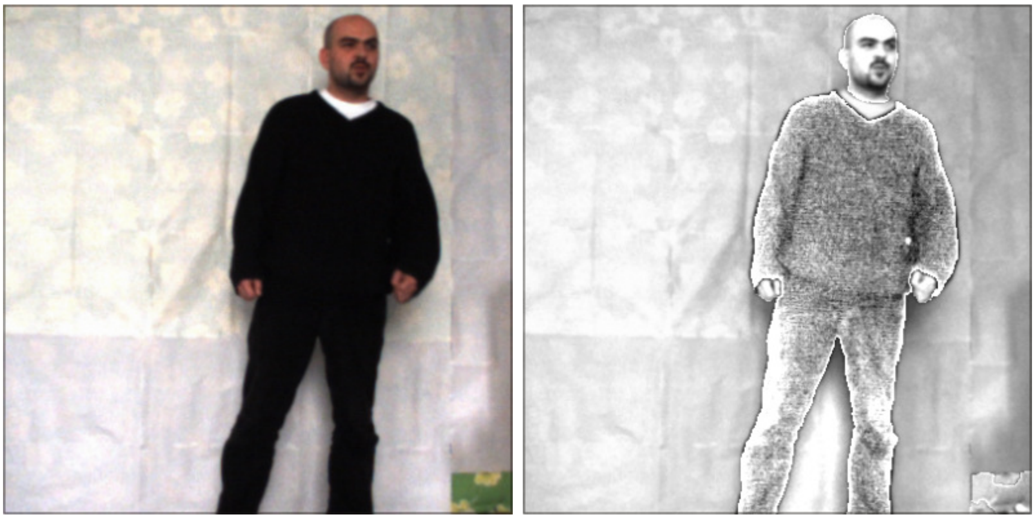
In this step we color segment the reference image employing the Mean Shift Based Image Segmenter (EDISON System) developed by [29]. The image is segmented to group regions of similar color into meaningful image segments. Regions with a size smaller than a user specified threshold are merged to neighboring large regions. The output of the color segmentation step serves to locally adjust the image contrast in order to achieve a more balanced distribution of stroke density across the whole image, as described in the following.

6.2.2 Stroke Arrangement

Our goal is to generate drawn-like illustrations derived from a gray scale version of the recorded images. The stroke density varies according to the tone of the image region, from very dense strokes in dark, to almost no strokes in very light areas, depending on the parameters specified by the user. In images recorded from a real scene we often find very dark areas usually with poor contrast. Inevitably, it is necessary to locally enhance the contrast of such images before further processing in order to better convey local shape variations and provide a well-balanced overall impression. In the following, we will describe the local contrast enhancement step and how the enhanced image is used to calculate the position and local density of the strokes.

Image Enhancement

The principal idea of our contrast-enhancement step is to “remove” color from the image. As a natural solution of this problem we use color segmentation to enhance contrast for each segment. The user selects a scaling factor (constant for the whole image), which is applied to each segment: Values greater than the segment’s mean value, modified by the scaling factor, turn to white and lower values are stretched between a minimum value and white. An example of this image enhancement step is shown in Figure 6.2. Some newly revealed details, especially in extremely dark areas of the image (see, for example, the trousers of the recorded person), are obvious in Figure 6.2 (b).



(a) (b)

Figure 6.2: (a) Left camera image, (b) enhanced image.

Stroke Centers Generation

To control the density of strokes according to the image intensity values, we first generate random points using the Poisson distribution that we further distribute according to the Poisson disc distribution method with variable disc radii, as explained in [110]. The position of the center point of each stroke is specified by calculating the disc radius, which can be considered as a limit that none of the other points can fall inside. The disc radius R of each center point is determined using the intensity value $I_{i,j}$ at that location as:

$$R = R_{df} + 2R_w \left(\tan^{-1} A_I \left(\frac{I_{i,j}}{I_{av}} - 1 \right) / \pi \right), \qquad 0 \leq R_w \leq R_{df} \qquad (6.1)$$

where I_{av} is the average intensity value and A_I, R_w and R_{df} are user defined constants used to control the density of the strokes. When $I_{i,j}$ decreases (i.e., in darker parts of the image), R decreases to produce more dense strokes and otherwise, when $I_{i,j}$ increases R also increases, which makes the strokes more sparsely distributed. This results in a convenient tuning of the image tone by adjusting the density and distribution of the strokes according to the intensity values, with darker areas represented by a denser stroke pattern than bright regions.

6.2.3 Stroke Orientation

To depict shapes and to give the image a "depth" impression, which is a very important factor, the stroke direction should be considered. To produce such effects, each stroke is drawn perpendicularly to the angle θ, which is specified by the orientation of the disparity layers in the disparity image.

In order to obtain a disparity map, we employed the graph-cut based stereo matching algorithm described in [14], as explained in Section 3.1.2. The method is of particular interest, since it represents disparity by a set of planar layers.

In the next step, for each layer, we utilize a square window of a maximum size previously specified by the user, as presented in Figure 6.3 (a). In order to enhance the contrast, the values of the window are rescaled from black to white as in Figure 6.3 (c). To identify the orientation of the projected disparity values, we use image moments as given in [114].

The image moment of lth degree about the x-axis and mth degree about the y-axis is defined as:

$$M_{lm} = \sum_x \sum_y x^l y^m I(x, y). \tag{6.2}$$

M_{lm} is called the image moment of nth degree for $n = l + m$. The image moment of zeroth degree, M_{00}, is simply the sum of all the intensity values of the gray-scale image. The angle θ is calculated as follows:

$$\theta = \frac{1}{2} \tan^{-1} (\frac{b}{a - c}), \tag{6.3}$$

where a, b, and c are defined as:

$$a = \frac{M_{20}}{M_{00}} - x_c^2, \tag{6.4}$$

$$b = 2(\frac{M_{11}}{M_{00}} - x_c y_c), \tag{6.5}$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2, \tag{6.6}$$

where x_c and y_c are:

$$x_c = \frac{M_{10}}{M_{00}}, \tag{6.7}$$

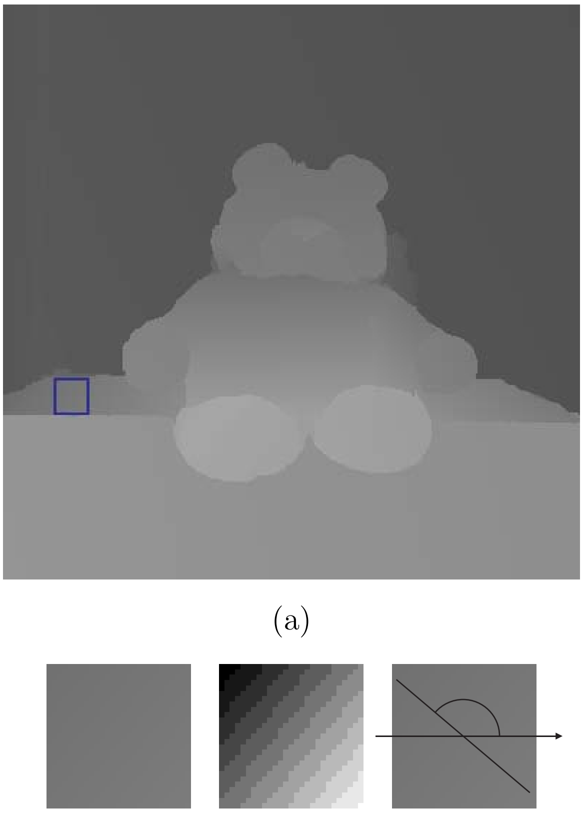
$$y_c = \frac{M_{01}}{M_{00}}. \tag{6.8}$$

As an option, the stroke orientation derived from image moments can be combined with additional user-defined preferences. In our application, we decided to replace horizontally and vertically aligned strokes, which we found to produce less appealing effects, by adding/subtracting a value of 45° whenever θ is close to 90° or 0°.

6.2.4 Outlining

A very important element often used by artists are outlines or silhouettes, which define the edges of objects, describe their shape and indicate their volume. In order to emphasize the dominant structure of the scene we employ silhouette edges extracted using the Edge Combination approach. Outlining strokes placed in this way, will properly delineate objects' shapes.

For strokes resulting from the Edge Combination image, we derive the orientation using image moments applied to the color segmented image. This distinguishes the contour strokes from the rest of the image, where the stroke orientation was computed from the disparity image. As a result, outlines are emphasized in the drawn image.



(a)

(b) (c) (d)

Figure 6.3: Process of disparity layer orientation estimation: (a) disparity image, (b) square window of the disparity layer, (c) rescaled values of the window in (b) from black to white, (d) estimated angle θ of the disparity layer.

6.3 Results

The obtained results of our algorithm applied to images of real scenes are illustrated in Figures 6.5 through 6.8. The size of the images is 400 x 400 pixels for all four test scenes in Figures 6.5 through 6.8. Figures (a) and (b) show a pair of stereo video frames in epipolar geometry. Their resulting disparity image is given in Figure (c), and Figure (e) shows the labeled disparity layers. Figure (d) shows the Edge Combination image, which combines the edges derived from the original image (a) and the disparity image (c), to outline the dominant structure of the scene. The result of the image color segmentation is given in Figure (f). Finally, Figure (g) shows the final drawing result: The real scene is illustrated using strokes oriented according to the disparity layers with the strokes' density reflecting local brightness variations encountered in the contrast-enhanced original image. In addition, dominant edges in the image (e.g., object contours) are highlighted to produce an artistic hand-drawn effect.

To illustrate the advantage of using a disparity map, Figure 6.4 presents the same scene as Figure 6.5, but generated without disparity information. One can recognize that the front and the top side of the Teddy's desk cannot be separated from each other in Figure 6.4, due to the uniform stroke orientation caused by the homogeneous appearance of the whole desk in the original image from Figure 6.5 (a). This results in an obvious absence of the impression of "depth" in Figure 6.4. Contrarily, in Figure 6.5 (g) the computer-generated strokes follow the orientation of the disparity layers and the geometric structure of the desk is clearly discernible.

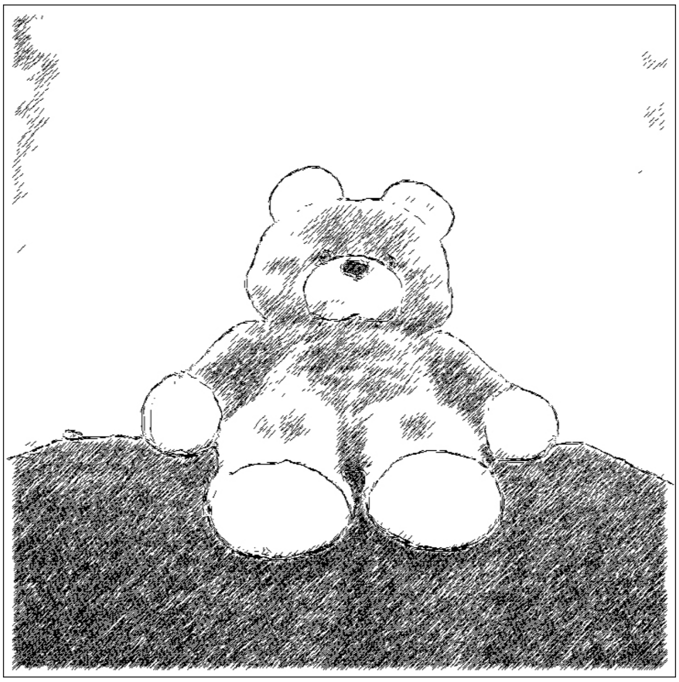


Figure 6.4: Example of the drawn-like result without disparity information.

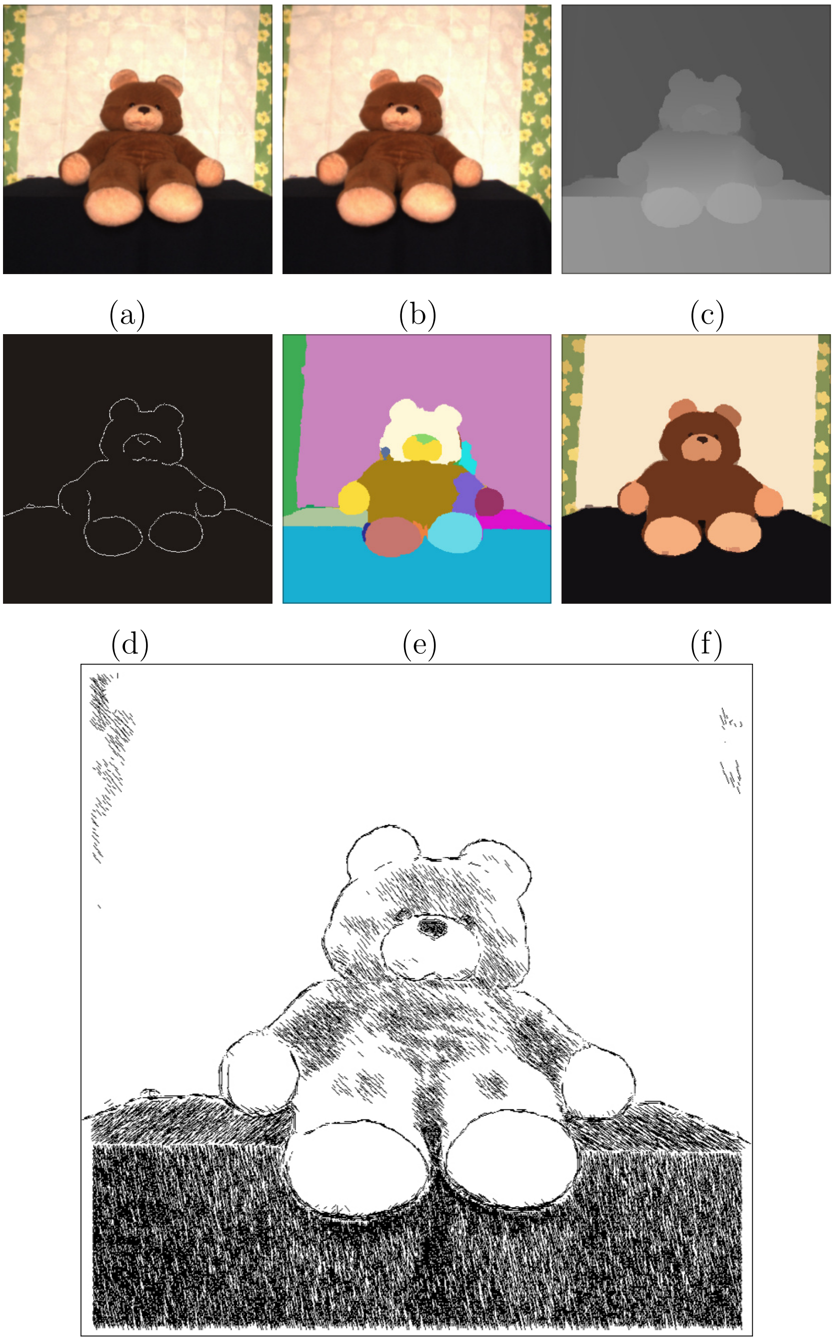
The benefit of the image enhancement step described in Section 6.2.2 can be recognized by comparing the result in Figure 6.6 (g) with the corresponding original and contrast-enhanced images in Figure 6.2. According to the disparity layers in Figure 6.6 (e), most parts of the person's body (except head, neck and hands) are represented by a single layer and thus result in a uniform stroke orientation. However, a sense of depth is conveyed by variations in the stroke density that follow the brightness pattern of the enhanced image in Figure 6.2 (b). A similar result is presented in Figure 6.7. In this case, the person's body is composed from several layers and the contrast enhancement step helps to better visually describe the subject's body shape. This is particularly noticeable in Figure 6.8, in which the contrast enhancement step along with the outlining step facilitates the distinction of the subject from the textured background.

Furthermore, our system can be easily extended to accommodate a variety of different styles of hand-drawn art. Besides the results obtained by employing the approach that we presented in this chapter, Figure 6.9 illustrates a few supplementary results, created using the initial set 6.8. In Figure 6.9 (a) each stroke is rendered by simply replacing the black color with colors sampled from the equivalent location in the original source image. Figure 6.9 (b) presents another style of additionally depicting the lightest areas of the enhanced image along with the original result of the described approach. For this example the usual white color of the background is changed to another color in order to facilitate the visibility of white strokes.

6.4 Summary

This chapter describes a method to generate stylized images derived from real scenes. Our approach employs the output of a stereo matching algorithm and utilizes tone stylization and object outlining techniques to synthesize images with a drawn-like appearance. The layered output of the stereo matcher is used to orient the strokes and in this way preserve the perspective impression of the image. In addition, we generate outlines using the Edge Combination approach that further enhance the image outlook and help to better apprehend the discrete scene structure.

This work comprises a potential of a number of additional steps for future investigation with the goal to obtain a variety of artistic styles for diverse types of illustrations. The output of such an image-based approach provides aesthetic flexibility, controlled by the user, to capture the important, application-specific visual details of the input images.



(a) (b) (c)

(d) (e) (f)

(g)

Figure 6.5: Example 1 of the drawn-like image stylization: (a) Left camera image, (b) right camera image, (c) disparity image, (d) Edge Combination image, (e) disparity layers, (f) color segmented image, (g) drawn image.

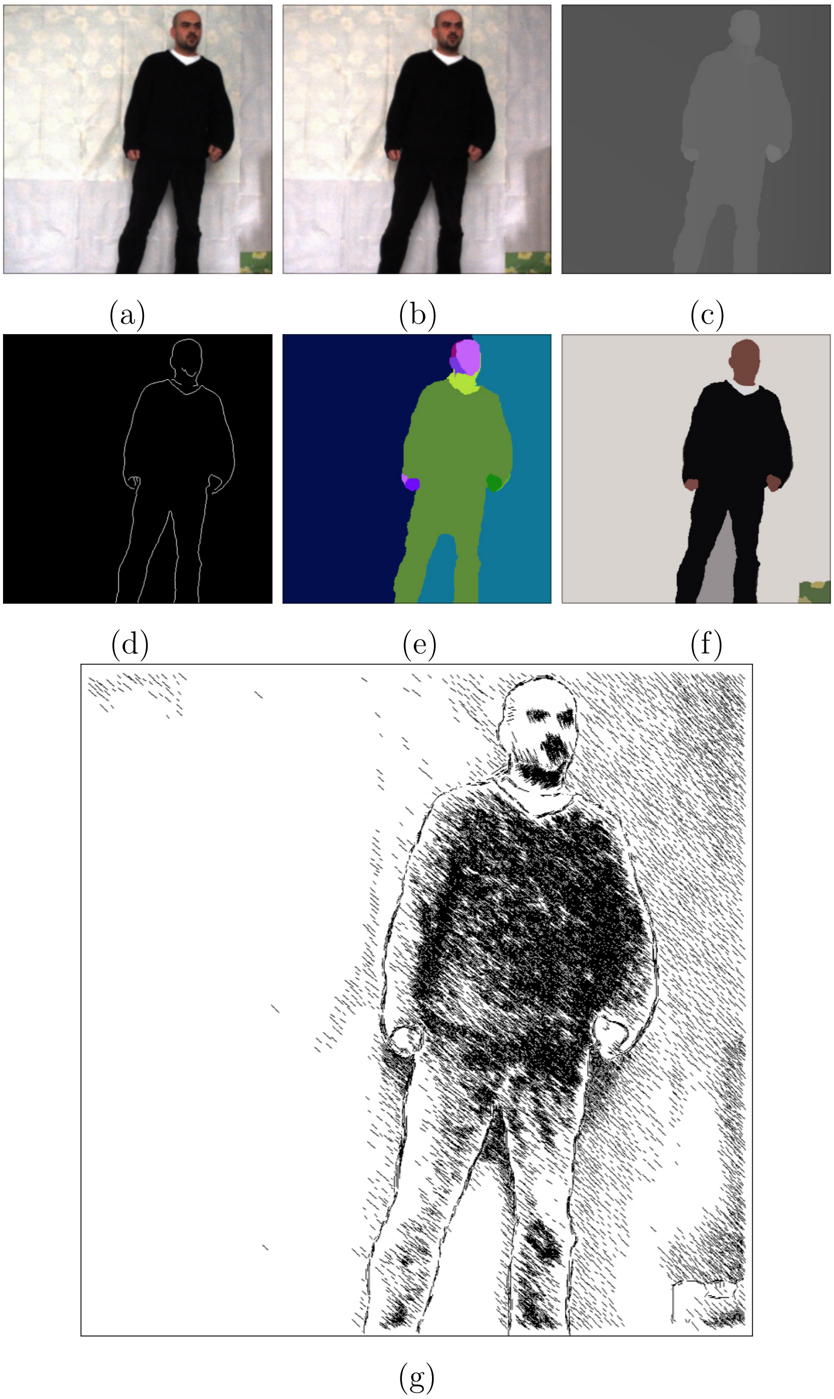


Figure 6.6: Example 2 of the drawn-like image stylization: (a) Left camera image, (b) right camera image, (c) disparity image, (d) Edge Combination image, (e) disparity layers, (f) color segmented image, (g) drawn image.

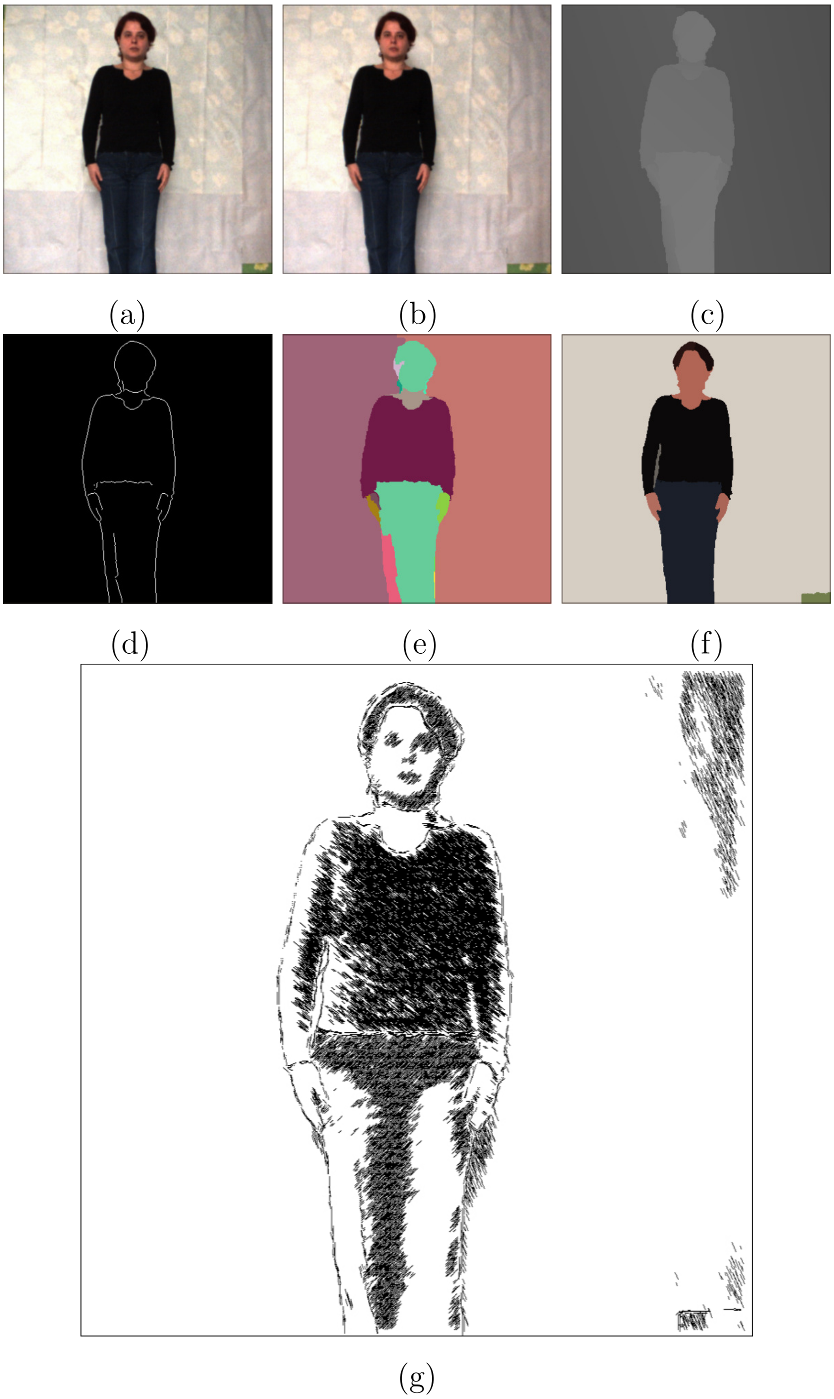
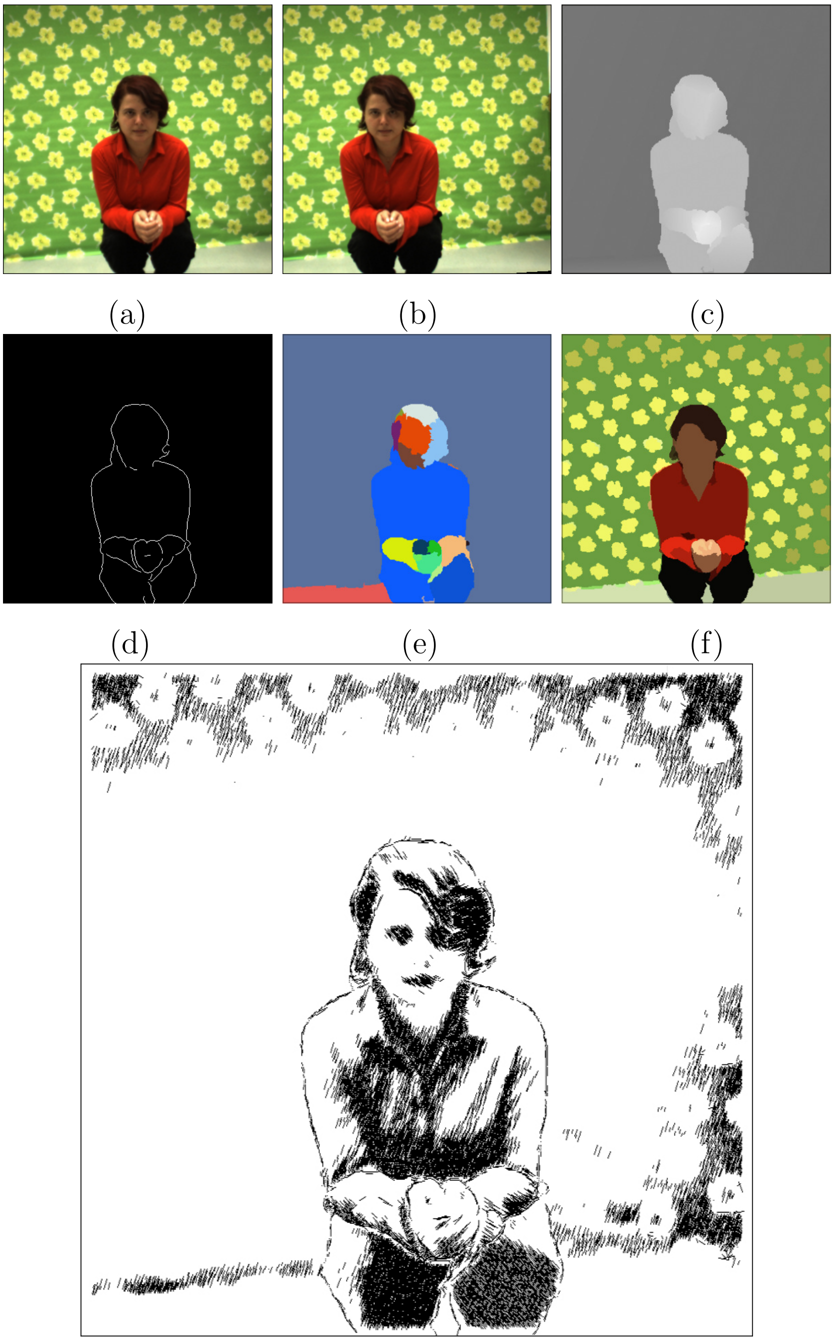


Figure 6.7: Example 3 of the drawn-like image stylization: (a) Left camera image, (b) right camera image, (c) disparity image, (d) Edge Combination image, (e) disparity layers, (f) color segmented image, (g) drawn image.

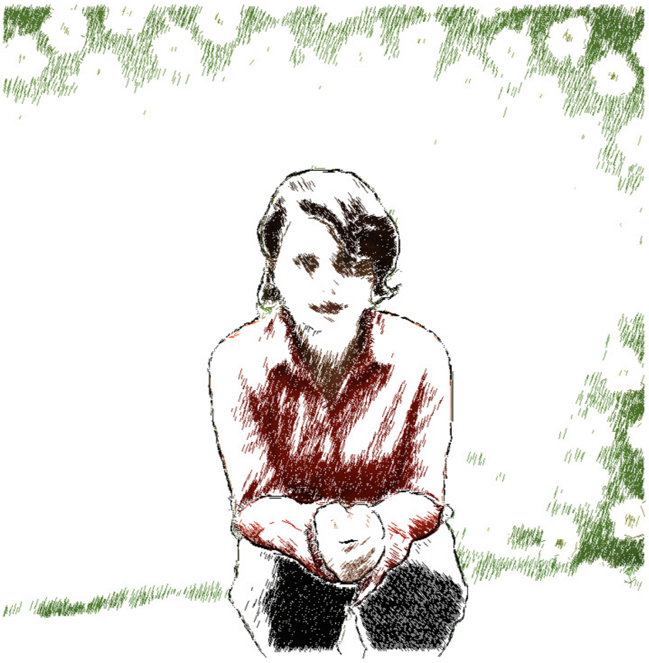


(a) (b) (c)

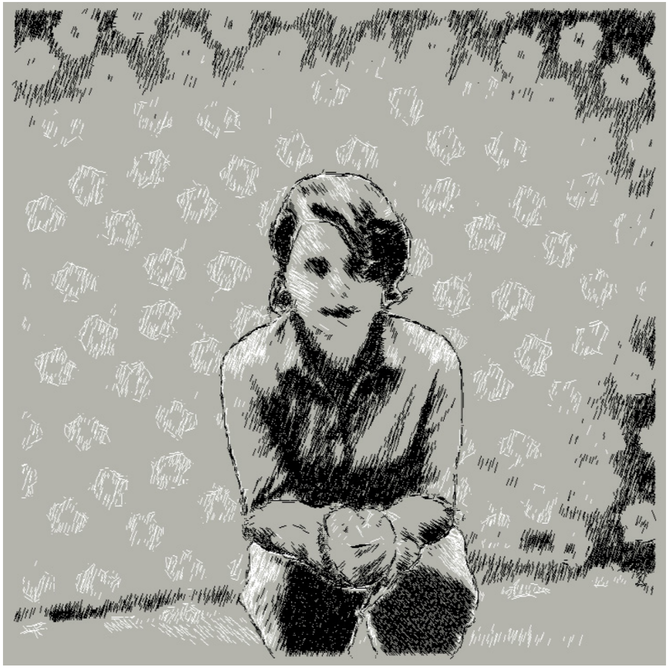
(d) (e) (f)

(g)

Figure 6.8: Example 4 of the drawn-like image stylization: (a) Left camera image, (b) right camera image, (c) disparity image, (d) Edge Combination image, (e) disparity layers, (f) color segmented image, (g) drawn image.



(a)



(b)

Figure 6.9: Results of the different drawn-like depiction styles for the image set given in Figure 6.8.

Chapter 7

Comics-like Motion Depiction

The heart of comics lies in the space between the panels where the reader's imagination makes still pictures come alive! [81]

7.1 Introduction

Although there is an ongoing discussion about historical roots of comics and their artistic contribution, reputed to be the first comic strip magazine is "Ally Sloper's Half Holiday" published in 1884. It was one of the most famous and most popular of all Victorian comics, featuring a regular character, Ally Sloper. However, the great boom that prompted the beginning of comics as an ongoing very popular art form was Richard Fenton Outcault's cartoon series "Hogan's Alley" for Joseph Pulitzer's New York World in 1895. The Yellow Kid, as he later came to be known for his yellow nightshirt, became the first comic character to serve as a marketing tool for the sale of newspapers. This set the bases for a new kind of art with the adventures of superheroes and monsters in stories written in a powerful language of visual symbols. [105, 51]

Comic books lost a lot of their popularity with the development and expansion of 3D animation and new media. Today a great variety of com-

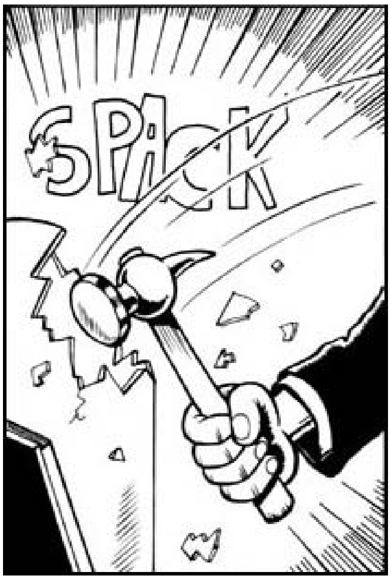


Figure 7.1: Example 1 from the "Alan Ford" comic book series. © Copyright 2005 by Max Bunker Press. Used by permission.

mercial applications give to the user the ability to create interesting artwork even without the requirement of good drawing skills. Contrary to this, comic books are usually created by skilled and creative artists who use a style of drawing unique for this type of art, as illustrated in Figures 7.1 and 7.4.

Comic books can be compared to silent movies, having dialogs in a written form and telling a story through sequences of images with a very important difference: comics use one image to present a sequence of frames. The form to visually depict the motion in just one image has been a big challenge for artists through history and resulted in different methods like: dynamic balance, multiple stroboscopic images, affine shear, photographic blur, and action lines [32]. Today, pictorial description of the motion is particularly interesting for image stylization and initiated the development of a number of systems [79, 18, 28, 68].

In this work we concentrate on a comic books' style of motion expression, where few lines suggest an action in the scene through motion lines and multiple contours. Similar to our work, Masuch et al. [79] depict motion, but different to our system they use a 3D scene as input, having access to the 3D geometry and precise motion information of the objects. The overall goal of our system is to enable a user to generate effective and attractive illustrations of dynamic natural scene recordings from stereo as input.

7.2 Algorithm

In this work, our goal is to stylize the image in a drawn-like form and to depict motion through motion lines and multiple contours. Thus, our algorithm consists of two major steps: image representation in a drawn-like form and comics-like depiction of motion. These steps are further explained in the following subsections.

7.2.1 Drawn-like Image Representation

To present an image in a drawn-like form, we make use of the work described in Chapter 6. The computed stroke density varies according to the tone of the intensity image region, from dense strokes in dark to no strokes in light areas. To better convey the shape of objects in natural scene images, which usually suffer from a bad contrast, we first perform a local image contrast enhancement, which is driven by the color segmentation result of the original image. An image frame of a test sequence shown in Figure 7.2 and a result that shows newly revealed details, obtained using the contrast enhancement step, is given in Figure 7.3. We then draw each stroke in a direction determined by the stereo derived disparity layers with the stroke density and distribution derived from a Poisson disc distribution. The disparity maps that we use in this work are results of the graph-cut based stereo-matching algorithm [14], which represents disparity by a set of planar layers. Such a representation of enhanced images using disparity to orient strokes results in a good distinction and shape depiction of objects in the scene.

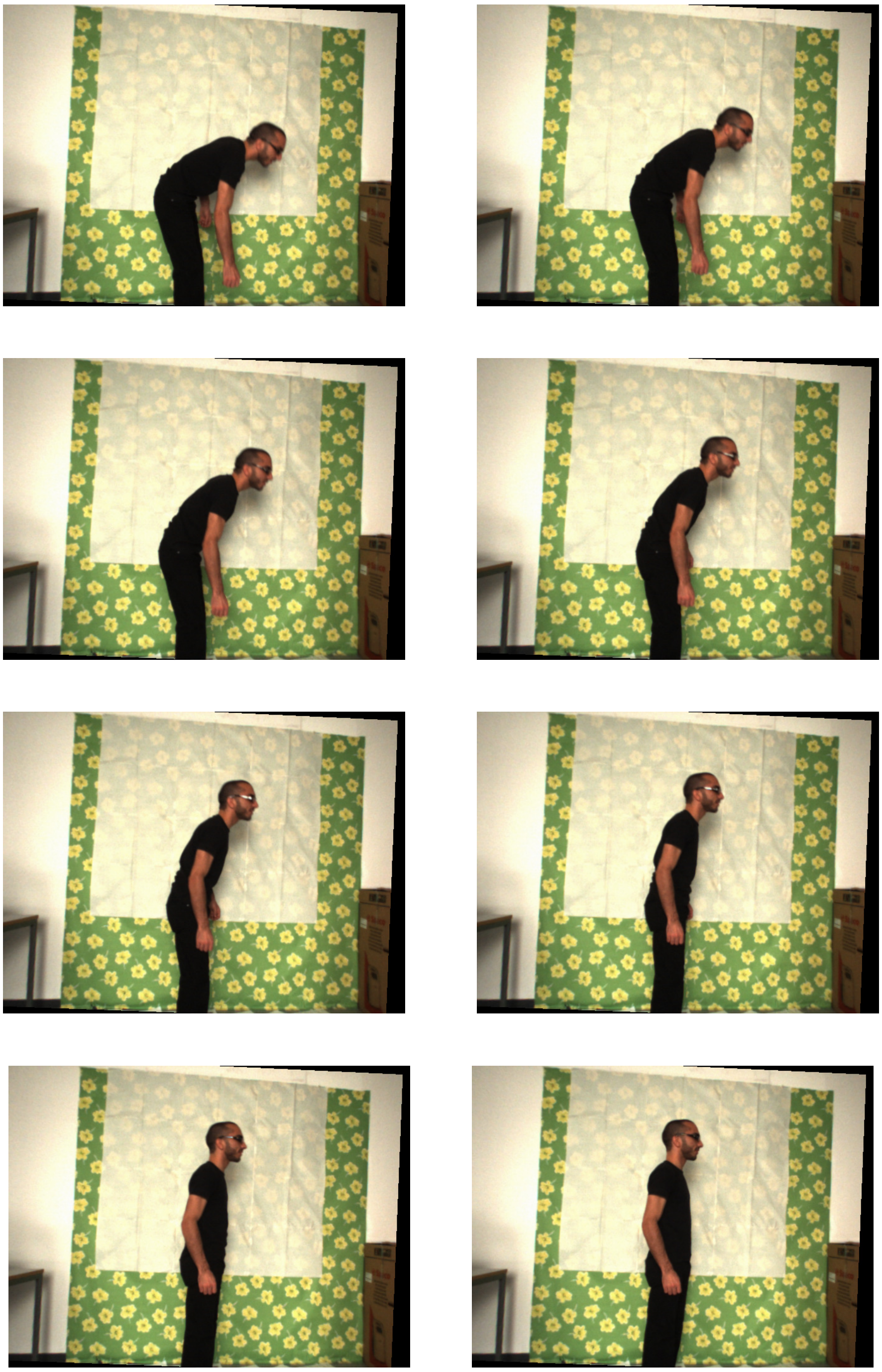
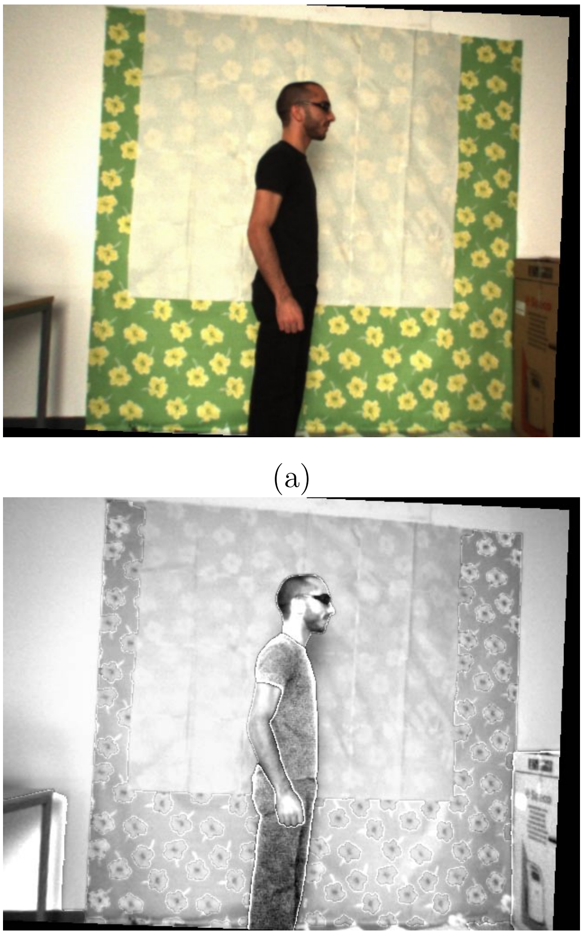


Figure 7.2: Intermediate frames (from top left to bottom) showing each fifth frame of the original sequence used for computation of the contours in our tests.



(a)

(b)

Figure 7.3: (a) Original image from the sequence shown in Figure 7.2, (b) enhanced original image, prepared for further stylization.

7.2.2 Tracking Motion

After creating the drawn-like representation of a reference image of the input stereo image pair sequence, the next very important step is the motion depiction of the objects present in the scene. Naturally, in order to depict the desired motion, we need to extract the motion information of the moving objects. To accomplish this task, various approaches can be used. In our work, for tracking the motion of objects in the scene, we employed the Kanade-Lucas-Tomasi (KLT) tracker, an implementation of the feature tracker described in [113].

In order to avoid the unnecessary computation of a dense motion field, the KLT tracker selects "good" features to track in the first image frame and tracks them throughout the following frames. A good feature represents a small window, optimal for tracking, with significant intensity variation in multiple directions. For the motion depiction in comics-like style, we need only information of distinct discontinuities in motion, which usually go along with depth discontinuities. By following this idea, among the point set selected automatically by the KLT tracker, we track only those points found to be close to edges in the Edge Combination image through all the frames of the user selected sequence. In this way we convey only the most important motion information, as desirable for a comics-like depiction style.

7.2.3 Depicting Motion

Depicting motion in comics has a meaningful purpose in story telling and strongly depends on an artist's imagination. In comics, motion is usually represented using either motion lines, multiple contours or a combination thereof, which results in a variety of styles. An example of motion depiction in a real comic through motion lines is given in Figure 7.1, and using both motion lines and multiple contours is shown in Figure 7.4. In our work, we aim at depicting motion by following these styles. Therefore, we will portray each one of them in more detail in the following sections.

Motion Lines

Motion lines, also known as action lines or speed lines, belong to the most common stylistic forms of depicting motion in comic books. Usually, the motion for each object is presented through one path in several motion lines (see Figure 7.1 and Figure 7.4). In a natural scene, one or more objects can be sources of motion. Representing just the motion of the object with, for instance, the longest path would be a great limitation. As a remedy, we give control to the user to choose a desired object and type of motion to depict by

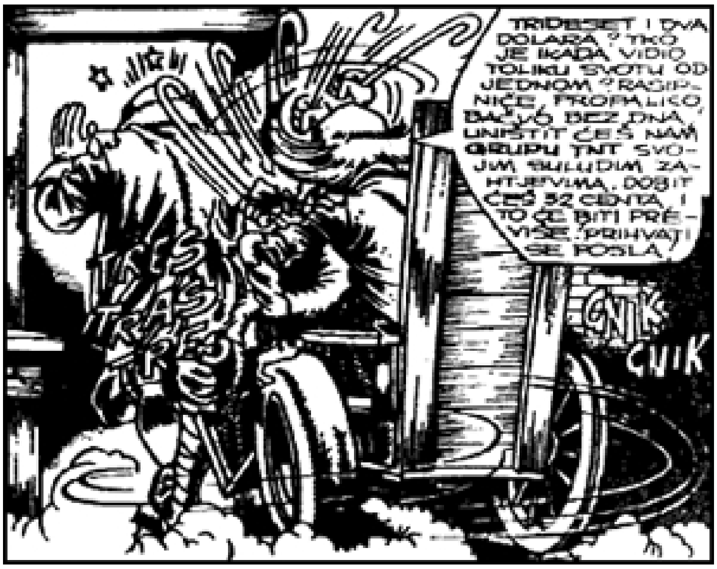
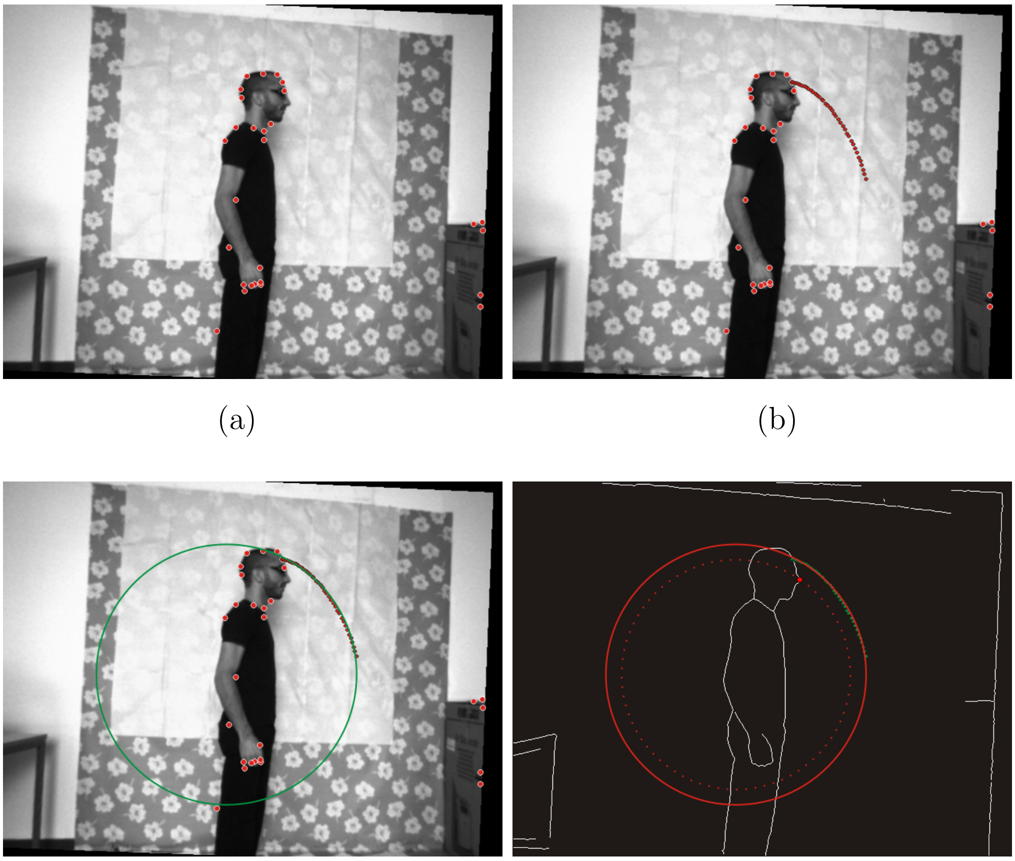

TRIDESET I DVA DOLARA? TKO JE IKADA VIDIO TOLIKU SVOTU OD JEDNOM? RASIPNIČE, PROPALICO, BAČVO BEZ DNA, UNIŠTIT ĆEŠ NAM GRUPU TNT SVOJIM SULUDIM ZAHTJEVIMA. DOBIT ĆEŠ 32 CENTA I TO ĆE BITI PREVIŠE. PRIHVATI SE POSLA!
GNIK
CUIK

Figure 7.4: Example 2 from the "Alan Ford" comic book series. © Copyright 2005 by Max Bunker Press. Used by permission.

selecting one of the object's tracked points. Figure 7.5 (b) shows a tracked point and its path selected from the tracking points in Figure 7.5 (a). To preserve the usual simple motion illustration in comics, we fit primitives (i.e. a circle, ellipse or straight line) to the path of the tracked point to get the final motion path and to have a possibility to extend its length. An example of fitting a primitive, in this case a circle, is shown in Figure 7.5 (c). By further selecting points on the contour in the Edge Combination image, Figure 7.5 (d), close to the tracked point, the user passes the same shape of the motion path to these points. In this way, more motion lines can be created from a single tracked path. For a more natural, hand-drawn look, the number of points in the final motion path, for each motion line, is reduced by removing a small random number of points at the beginning of the path.

In order to stylize the obtained motion path in a handcrafted-like fashion, we employed a strategy similar to the one described in the previous Chapter 5, in Section 5.3. Different to this procedure, instead of employing the computed error information obtained in the fitting process, we now use the user input parameters to specify the thickness of the stroke. Through these parameters,



(a) (b)

(c) (d)

Figure 7.5: (a) Points to track, (b) motion path of the selected point, (c) circle fitted to the motion path, (d) selected new point on the contour in the Edge Combination image and its path.

the user gets full control over the handcrafted-like appearance of the motion lines.

Multiple contours

Multiple contours technique is another, very important, stylistic form of motion representation which uses a complete or a part of a moving object's contour. To imitate this style we use Edge Combination images extracted from the frames in a regular interval and draw them on the final result. Regularity in frame selection gives a good impression of the object's speed. The variety of styles can be even larger by changing the transparency of contours through frames.

This is, in a way, an even simpler and very effective form of motion expression, which does not require the motion tracking step. On the other hand, this style of motion portrayal requires the computation of the Edge Combination image for depiction of all desired frames, which makes it dependable on the disparity map computation.

However, the motion depiction through multiple contours is a very expressive style. In this way presented motion of an object can be very comprehensible, as we will show through the obtained experimental results. This style is also very suitable for further experimenting in the creation of novel models of artistic expression and depiction.

7.3 Experimental Results

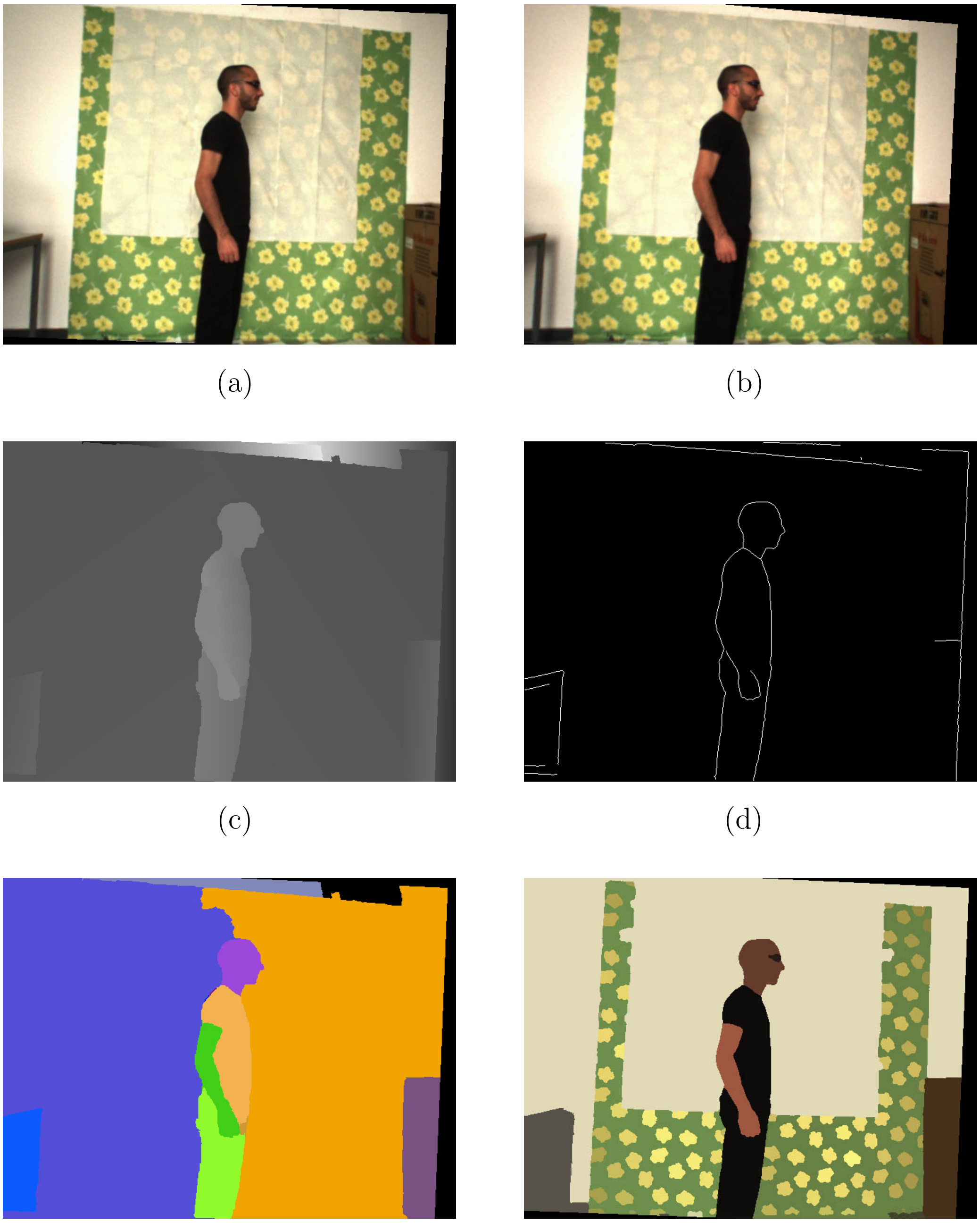
Example results of motion depiction using multiple contours, motion lines and their combination are given in Figures 7.7 - 7.10. The initial stereo image pair of the image sequence shown in Figure 7.2 is presented in Figure 7.6 (a) and (b). Their resulting disparity image computed by employing the stereo matching algorithm presented in [14] is given in Figure 7.6 (c), along with the labeled disparity layers shown in Figure 7.6 (e). This disparity image is further utilized to compute the Edge Combination image given in Figure 7.6 (d). The color segmentation result, see Figure 7.6 (f), is used in order to locally enhance the contrast to better convey the shape of the natural scene before further processing. Figure 7.7 (b) illustrates the output of our algorithm in the style of motion depiction using motion lines together with the drawn-like representation of the natural scene from Figure 7.7 (a). One can see that the circular motion lines describe well the actual motion of the person's upper body shown in Figure 7.2. Another example of the same motion depiction style is given in Figure 7.10 (c). In this case, the motion of a falling object is depicted using motion lines that are obtained by fitting straight lines to the motion path.

Two sets of images, presented in Figure 7.8 and Figure 7.9, demonstrate various styles of motion depiction through multiple contours and multiple contours joined with motion lines, respectively. The multiple contours styles that we illustrate in these figures are obtained by changing the transparency of contours through frames, using a complete or a part of a moving object's contour and their combination.

7.4 Summary and Outlook

In this chapter we have presented an algorithm to depict motion in comics-like form, with an artistic drawn-like representation of the scene, from a stereo image sequence. In order to truthfully simulate comics' style of motion portrayal, we devised several motion depiction techniques using motion lines and/or multiple contours. We have shown how silhouettes extracted by the Edge Combination algorithm can serve in motion depiction and also to portray the impression of the change of an object's speed.

In future steps, we intend to experiment with a spline representation of the motion path. This should give more flexibility and the possibility of depicting more complex motions, but it may also raise problems in motion path length extension. We also plan to experiment with other stylistic properties of comics and explore additional techniques of motion illustration in still images.

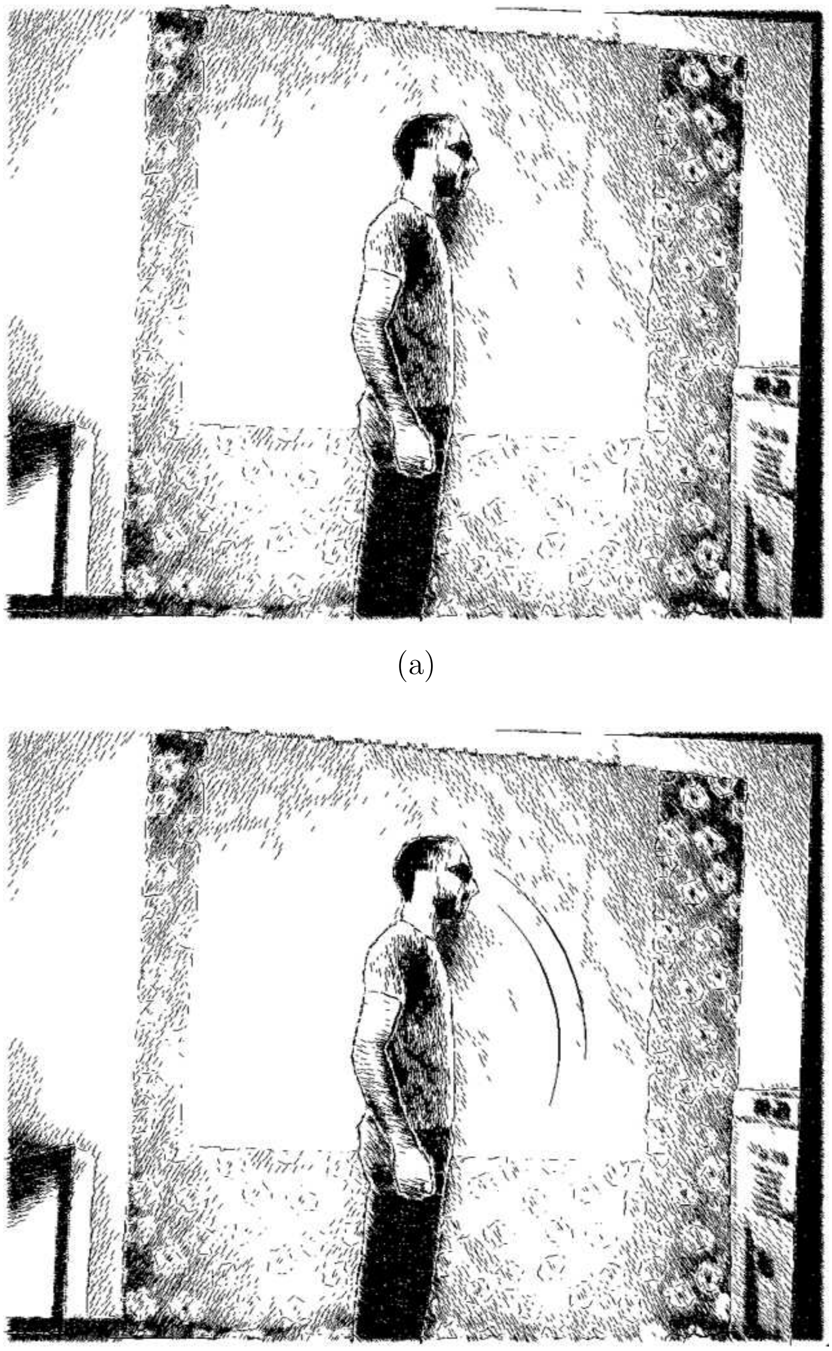


(a) (b)

(c) (d)

(e) (f)

Figure 7.6: (a) Left camera image, (b) right camera image, (c) disparity image, (d) Edge Combination image, (e) layers image, (f) color segmented image.



(a)

(b)

Figure 7.7: (a) Drawn-like image representation for the test set shown in Figure 7.6, (b) the same drawn-like image representation with motion depicted through motion lines.

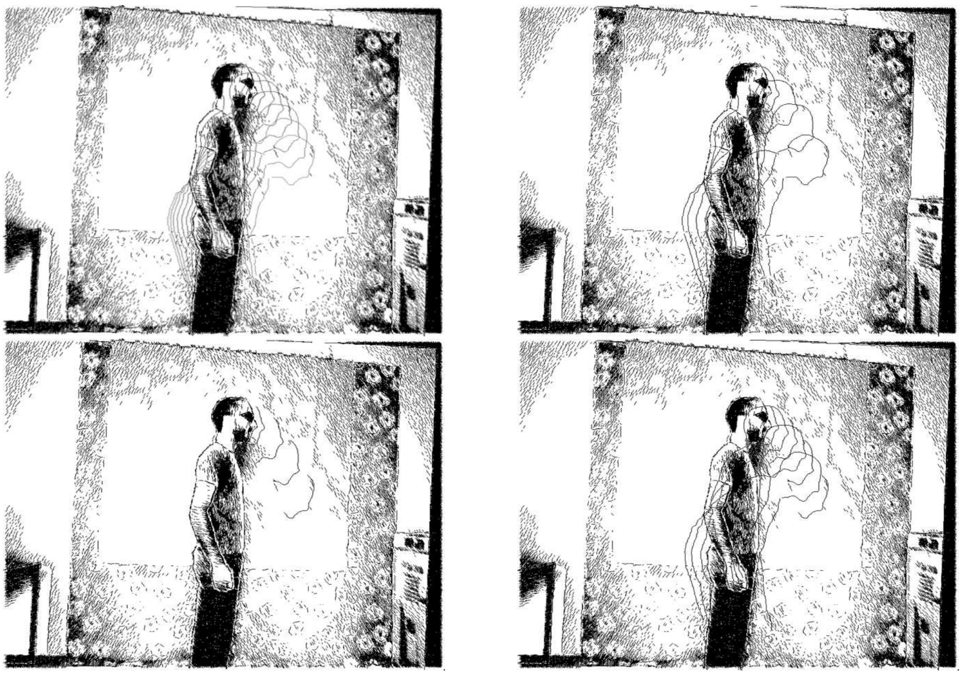


Figure 7.8: Results of motion depiction through multiple contours.

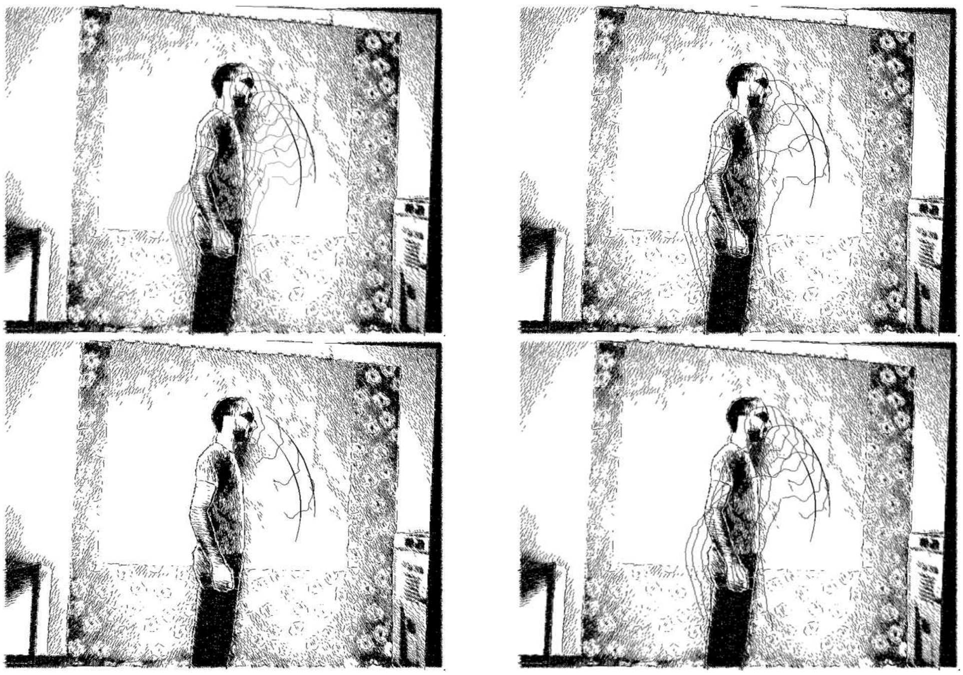
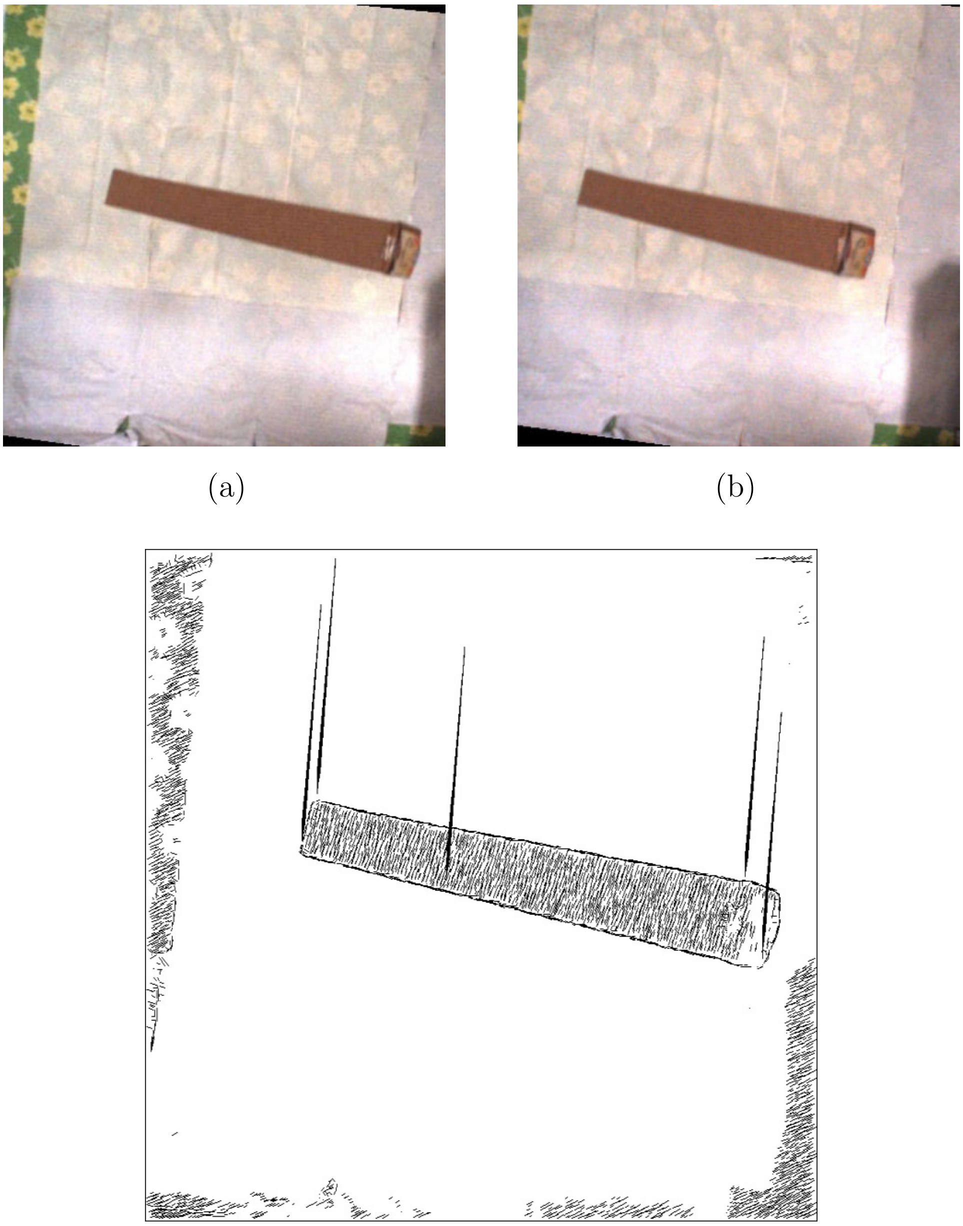


Figure 7.9: Results of motion depiction through multiple contours and motion lines.



(a) (b)

(c)

Figure 7.10: (a) Left camera image, (b) right camera image, (c) drawn image with depicted motion.

Chapter 8

Summary and Future Work

In this thesis we presented novel algorithms to stylize images of natural scenes in a form similar to creations of traditional artistic styles. The general goal of this work is to create a stylized representation of the real world with the attention on depiction concepts that preserve the natural visual aspect of a scene and its content appearance. Therefore, our algorithms focus on the utilization of stereo-derived disparity maps in order to obtain additional information about the 3D scene structure and objects' shape, which allows to produce stylization results in an expressive and comprehensive form.

First of all, we described a method to extract object silhouettes from stereo images of a natural scene. We showed that this so-called Edge Combination algorithm can improve the recognition of object contours from stereo images by combining the robustness of stereo-derived edges with the higher precision of the original intensity edges. Results of the Edge Combination algorithm show that the effects of stereo matching errors are largely suppressed, especially along object boundaries. In this way, a more reliable recognition of object contours is provided.

In subsequent chapters we presented several applications and experimental results that show the usefulness of the Edge Combination approach. In experiments illustrated in Section 4.7, we demonstrated that it can improve the performance of Active Contours for a more accurate object segmentation.

In Chapter 5 we described another type of application, which utilizes the Edge Combination image to depict a natural scene structure in an abstract fashion. We presented an algorithm to encode the extracted contour edges using an efficient Bézier curve representation for a subsequent image-based computer graphics rendering application in which we represent recordings of the natural scene in a sketched-like form. In this context, we implemented a fast curve approximation technique to convert the rasterized edges into a parametric form. The parametric form of the edges, outlining meaningful objects in the scene, is then stylized to produce sketches that have a handcrafted appearance. The smoothness of the stylized output is similar to that found in many concept drawings and sketches. The algorithm can be useful to assist an artist in creating the basis for a fine art work or it can help a beginner artist as a guiding tool on important features of a scene when learning visual fine arts.

Furthermore, we proposed a method to generate images in a drawn-like form derived from real scenes. Our approach employs the output of a stereo matching algorithm and utilizes tone stylization and object outlining techniques to synthesize images with a handcrafted appearance. The layered output of the stereo matcher is used to orient the strokes and in this way preserve the perspective impression of the image. In addition, we generate outlines using the Edge Combination algorithm that further enhance the image outlook and help to better apprehend the discrete scene structure.

Finally, we presented an algorithm to depict motion in comics-like form, from a stereo image sequence. We devised several motion depiction styles using motion lines and/or multiple contours. We showed that silhouettes extracted by the Edge Combination algorithm can serve in motion delineation and also to portray the impression of change in a moving object's speed.

Although, through the described algorithms, we have shown the usefulness of the Edge Combination approach, there is still space for further improvements to make it even more stable and less dependable on parameters and edge detector results. Using the robustness of disparity edges, the Edge

Combination algorithm relies on a stereo-matching result. This essentially means that with a better quality of the disparity map, a better result of the Edge Combination algorithm can be obtained. In our tests we used high-quality disparity maps delivered by stereo algorithms that reflect the current state of the art. A research topic of high practical relevance would be to explore the potential applications of the proposed approach to lower-quality stereo maps as, for example, computed by real-time stereo algorithms or by using low-resolution webcam images.

Another strong influence on the algorithm has the edge detector. In our work, we used the Canny edge detector, which gives very satisfactory results. However, some problems can arise, for example when dealing with Y-junctions, causing broken edges. Our algorithm comprises the important step of closing gaps of the "basic" Edge Combination image, created in the edge search process. The linking step is strongly dependable on the continuity of edges in the original edge image. This basically means that if the edge is broken in the original edge image, the edge in the Edge Combination image will also have a gap. Future work might overcome this weakness related to the edge detector in the Edge Combination approach.

Another focus of follow-up research could be to experiment with other possible applications based on the Edge Combination approach, related to object segmentation, detection, and tracking, as well as to explore various additional stylization techniques for natural scenes' depiction. This includes the development of "more advanced" algorithms that put more attention on, for instance, faces when depicting humans or use levels of details according to the disparity map when illustrating a more complex scene.

While in this thesis we have focused on the automatic simulation of traditionally created artwork, future research might also explore possibilities of devising a combination of traditional and modern. The automatic creation of computer-generated artistic work from natural images and video scenes offers high flexibility for devising novel styles. The goal of such automation is to support and assist users with different levels of art and computer software knowledge to create meaningful and aesthetically pleasing results.

Bibliography

[1] A. Agarwal and B. Triggs. 3D human pose from silhouettes by relevance vector regression. In *Proceedings of CVPR '04*, volume 2, pages 882–888, Washington, DC, USA, June 2004.

[2] A. Agarwala. SnakeToonz : A semi-automatic approach to creating cel animation from video. In *Proceedings of NPAR '02*, pages 139–146, Annecy, France, June 2002.

[3] A. Agarwala, A. Hertzmann, S. Seitz, and D. Salesin. Keyframe-based tracking for rotoscoping and animation. In *Proceedings of SIGGRAPH '04*, pages 584–591, Los Angeles, California, USA, August 2004.

[4] P. Alperson, editor. *The Philosophy of the Visual Arts.* Oxford University Press, 1992.

[5] "Art." Encyclopaedia Britannica. 2007. Britannica Concise Encyclopedia. 10 January 2007. http://concise.britannica.com/ebc/article-9389134/art.

[6] "Art." Microsoft® Encarta® Online Encyclopedia 2006. 12 June 2006. http://encarta.msn.com © 1997-2006 Microsoft Corporation. All Rights Reserved.

[7] B. A. Barsky. Parametric Bernstein/Bézier curves and tensor product surfaces. Technical Report UCB/CSD-90-571, EECS Department, University of California, Berkeley, May 1990.

[8] S. Battiato, G. Puglisi, and G. Impoco. Vectorialisation of raster colour images. In *Proceedings of 2a Conferenze Nazionale del Gruppo del Colore*, pages 20–22, Milano, Italy, September 2006.

[9] B. G. Baumgart. *Geometric modeling for computer vision.* PhD thesis, Stanford University, 1974.

[10] M. A.-A. Bhuiyan and H. Hama. Recovering the control points of Bézier curves for line image indexing. *Journal of Electronic Imaging*, 11(2):177–186, April 2002.

[11] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, 35(3):269–293, December 1999.

[12] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *Proceedings of ECCV 2004*, pages 428–441, Prague, Czech Republic, May 2004.

[13] G. D. Blasi and G. Gallo. Artificial mosaics. *The Visual Computer*, 21(6):373–383, July 2005.

[14] M. Bleyer and M. Gelautz. Graph-based surface reconstruction from stereo pairs using image segmentation. In *Proceedings of SPIE*, volume 5665, pages 288–299, San Jose, California, USA, January 2005.

[15] A. Bousseau, M. Kaplan, J. Thollot, and F. X. Sillion. Interactive watercolor rendering with temporal coherence and abstraction. In *Proceedings of NPAR '06*, pages 141–149, Annecy, France, June 2006.

[16] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proceedings of ICCV '01*, volume 1, pages 105–112, Vancouver, Canada, July 2001.

[17] M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):844–851, August 2000.

[18] G. J. Brostow and I. Essa. Image-based motion blur for stop motion animation. In *Proceedings of SIGGRAPH '01*, pages 561–566, Los Angeles, California, USA, August 2001.

[19] C. Buehler, W. Matusik, L. McMillan, and S. J. Gortler. Creating and rendering image-based visual hulls. Technical Report 780, MIT LCS, March 1999.

[20] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[21] V. Ceric. From simulation to algorithmic art. *Simulation News Europe*, (44/45):61–62, December 2005.

[22] H.-H. Chang and H. Yan. Vectorization of hand-drawn image using piecewise cubic Bézier curves fitting. *Pattern Recognition*, 31(11):1747–1755, November 1998.

[23] H. Chen, Z. Liu, C. Rose, Y. Xu, H.-Y. Shum, and D. Salesin. Example-based composite sketching of human portraits. In *Proceedings of NPAR '04*, pages 95–153, Annecy, France, June 2004.

[24] K.-M. G. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time part I: Theory and algorithms. *International Journal of Computer Vision*, 62(3):221–247, May 2005.

[25] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A Bayesian approach to digital matting. In *Proceedings of IEEE CVPR 2001*, volume 2, pages 264–271, Kauai, Hawaii, December 2001.

[26] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211–218, March 1991.

[27] J. P. Collomosse and P. M. Hall. Painterly rendering using image salience. In *Proceedings of EGUK '02*, pages 122–128, Leicester, UK, June 2002.

[28] J. P. Collomosse, D. Rowntree, and P. M. Hall. Cartoon-style rendering of motion from video. In *Proceedings of Video, Vision and Graphics (VVG)*, pages 117–124, Bath, UK, July 2003.

[29] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

[30] C. J. Curtis. Loose and sketchy animation. In *Proceedings of ACM SIGGRAPH '98*, page 145, Orlando, Florida, USA, July 1998.

[31] C. J. Curtis, S. E. Anderson, J. E. Seims, K. W. Fleischer, and D. H. Salesin. Computer-generated watercolor. In *Proceedings of SIGGRAPH '97*, pages 421–430, Los Angeles, California, USA, August 1997.

[32] J. E. Cutting. Representing motion in a static image: constraints and parallels in art, science, and popular culture. *Perception*, 31(10):1165–1193, 2002.

[33] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, 37(2):175–185, June 2000.

[34] P. Decaudin. Cartoon-looking rendering of 3D-scenes. Technical Report 2919, INRIA, June 1996.

[35] M. R. Delahunt. "High art." ArtLex Art Dictionary. 12 June 2006. http://www.artlex.com/.

[36] O. Deussen, P. Hanrahan, B. Lintermann, R. Měch, M. Pharr, and P. Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. In *Proceedings of SIGGRAPH '98*, pages 275–286, Orlando, Florida, USA, July 1998.

[37] O. Deussen and T. Strothotte. Computer-generated pen-and-ink illustration of trees. In *Proceedings of SIGGRAPH '00*, pages 13–18, New Orleans, Louisiana, USA, July 2000.

[38] Digital Art Museum - [DAM]. 10 January 2007. http://www.dam.org/.

[39] F. Durand, V. Ostromoukhov, M. Miller, F. Duranleau, and J. Dorsey. Decoupling strokes and high-level attributes for interactive traditional drawing. In *Proceedings of the 12th Eurographics Workshop on Rendering*, pages 71–82, London, UK, June 2001.

[40] O. Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, 1993.

[41] A. Finkelstein and M. Range. Image mosaics. In *Proceedings of EP '98/RIDT '98*, pages 11–22, St. Malo, France, April 1998.

[42] K. K. Flocken. *Silhouettes: Rediscovering the Lost Art*. Paperportraits.com Press, 2000.

[43] K. J. Freedman. *Teaching Visual Culture: Curriculum, Aesthetics and the Social Life of Art*. Teachers College Press, 2003.

[44] L.-P. Fritzsche, H. Hellwig, S. Hiller, and O. Deussen. Interactive design of authentic looking mosaics using Voronoi structures. In *Proceedings of the 2nd International Symposium on Voronoi Diagrams in Science and Engineering*, pages 1–11, Seoul, Korea, October 2005.

[45] M. Gleicher. Image snapping. In *Proceedings of SIGGRAPH '95*, pages 183–190, Los Angeles, California, USA, August 1995.

[46] B. Gooch, G. Coombe, and P. Shirley. Artistic vision: painterly rendering using computer vision techniques. In *Proceedings of NPAR '02*, pages 83–90, Annecy, France, June 2002.

[47] S. Green. *Introduction to Non-Photorealistic Rendering.* ACM SIGGRAPH 99 Course Notes, 1999.

[48] R. Gross. Run-on recognition in an on-line handwriting recognition system. In *Term Project*, University of Karlsruhe, 1997.

[49] S. R. Gunn and M. S. Nixon. A robust snake implementation; a dual active contour. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 19(1):63–68, January 1997.

[50] C.-E. Guo, S.-C. Zhu, and Y. N. Wu. Towards a mathematical theory of primal sketch and sketchability. In *Proceedings of ICCV '03*, volume 2, pages 1228–1235, Nice, France, October 2003.

[51] R. C. Harvey. *Children of the Yellow Kid: The Evolution of the American Comic Strip.* University of Washington Press, 1999.

[52] A. Hausner. Simulating decorative mosaics. In *Proceedings of SIGGRAPH '01*, pages 573–580, Los Angeles, California, USA, August 2001.

[53] J. Hays and I. Essa. Image and video based painterly animation. In *Proceedings of NPAR '04*, pages 113–120, Annecy, France, June 2004.

[54] D. Hearn and M. P. Baker. *Computer Graphics (2nd ed.): C Version.* Prentice-Hall, 1997.

[55] A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of SIGGRAPH '98*, pages 453–460, Orlando, Florida, USA, July 1998.

[56] A. Hertzmann and D. Zorin. Illustrating smooth surfaces. In *Proceedings of SIGGRAPH '00*, pages 517–526, New Orleans, Louisiana, USA, July 2000.

[57] P. Hillman, J. Hannah, and D. Renshaw. Alpha channel estimation in high resolution images and image sequences. In *Proceedings of IEEE CVPR 2001*, volume 1, pages 1063–1068, Kauai, Hawaii, December 2001.

[58] H. Hirschmüller, P. R. Innocent, and J. Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47(1-3):229–246, April 2002.

[59] S. N. Ho and R. Komiya. Real time loose and sketchy rendering in hardware. In *Proceedings of SCCG '04*, pages 83–88, Budmerice, Slovakia, April 2004.

[60] M. Hoch and P. C. Litwinowicz. A semi-automatic system for edge tracking with snakes. *The Visual Computer*, 12(2):75–83, February 1996.

[61] D. D. Hoffman and M. Singh. Salience of visual parts. *Cognition*, 63(1):29–78, April 1997.

[62] S. Ingram and P. Bhat. Automatic collage using texture synthesis. In *Proceedings of International Symposium on Smart Graphics '04*, pages 140–145, May 2004.

[63] H. H.-S. Ip, A. K. Y. Cheng, and W. Y. F. Wong. Affine invariant retrieval of shapes based on hand-drawn sketches. In *Proceedings of ICPR '02*, volume 2, pages 794–797, Quebec, Canada, August 2002.

[64] T. Isenberg, B. Freudenberg, N. Halper, S. Schlechtweg, and T. Strothotte. A developer's guide to silhouette algorithms for polygonal models. *IEEE Computer Graphics and Applications*, 23(4):28–37, July-Aug 2003.

[65] E. Izquierdo. Disparity/segmentation analysis: Matching with an adaptive window and depth-driven segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(4):589–607, June 1999.

[66] M. Kaplan and E. Cohen. Computer generated celtic design. In *Proceedings of EGRW '03*, pages 9–19, Leuven, Belgium, June 2003.

[67] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, January 1988.

[68] B. Kim and I. Essa. Video-based nonphotorealistic and expressive illustration of motion. In *Proceedings of Computer Graphics International (CGI '05)*, pages 32–35, Stony Brook, New York, USA, June 2005.

[69] J. Kim and F. Pellacini. Jigsaw image mosaics. In *Proceedings of SIGGRAPH '02*, pages 657–664, San Antonio, Texas, USA, July 2002.

[70] Y. Kuno, T. Watanabe, Y. Shimosakoda, and S. Nakagawa. Automated detection of human for visual surveillance system. In *Proceedings of ICPR'96*, volume 3, pages 865–869, Vienna, Austria, August 1996.

[71] L. Lam, S.-W. Lee, and C. Y. Suen. Thinning methodologies-a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):869–885, September 1992.

[72] N. Li and Z. Huang. A feature-based pencil drawing method. In *Proceedings of GRAPHITE '03*, pages 135–140, Melbourne, Australia, February 2003.

[73] W. Liu and D. Dori. A survey of non-thinning based vectorization methods. In *Proceedings of SSPR'98/SPR'98*, pages 230–241, Sydney, Australia, August 1998.

[74] T. Luft and O. Deussen. Real-time watercolor for animation. *Journal of Computer Science and Technology*, 21(2):159–165, March 2006.

[75] S. Manke, M. Finke, and A. Waibel. The use of dynamic writing information in a connectionist on-line cursive handwriting recognition system. *Advances in Neural Information Processing Systems 7 (NIPS)*, pages 1093–1100, 1995.

[76] X. Mao, Y. Nagasaka, and A. Imamiya. Automatic generation of pencil drawing from 2D images using line integral convolution. In *Proceedings of CAD/Graphics 2001*, pages 240–248, Kunming, China, August 2001.

[77] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194(4262):283–287, 1976.

[78] D. Martín, J.-D. Fekete, and J. C. Torres. Flattening 3d objects using silhouettes. *Computer Graphics Forum (EG'02)*, 21(3):239–248, September 2002.

[79] M. Masuch, S. Schlechtweg, and R. Schulz. Speedlines: Depicting motion in motionless pictures. In *Proceedings of SIGGRAPH '99*, page 277, Los Angeles, California, USA, August 1999.

[80] H. Matsui, H. Johan, and T. Nishita. Creating colored pencil style images by drawing strokes based on boundaries of regions. In *Proceedings of Computer Graphics International '05*, pages 148–154, Stony Brook, New York, USA, June 2005.

[81] S. McCloud. *Reinventing Comics: How Imagination and Technology Are Revolutionizing an Art Form.* Harper Paperbacks, 2000.

[82] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1(2):91–108, June 1996.

[83] T. McInerney and D. Terzopoulos. T-snakes: Topology adaptive snakes. *Medical Image Analysis*, 4(2):73–91, June 2000.

[84] S. Menet, P. Sait-Marc, and G. Medioni. Active contour models: Overview, implementation and applications. In *Proceedings of IEEE*

International Conference On Systems, Man And Cybernetics, pages 194–199, Los Angeles, California, USA, November 1990.

[85] M. Mignotte. Unsupervised statistical sketching for non-photorealistic rendering models. In *Proceedings of ICIP '03*, volume 3, pages 573–576, Barcelona, Spain, September 2003.

[86] J. L. Mitchell, C. Brennan, and D. Card. Real-time image-space outlining for non-photorealistic rendering. In *Proceedings of SIGGRAPH '02*, page 239, San Antonio, Texas, USA, July 2002.

[87] S. Mizuno, M. Okada, and J. Toriwaki. Virtual sculpting and virtual woodcut printing. *The Visual Computer*, 14(2):39–51, June 1998.

[88] E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In *Proceedings of SIGGRAPH '95*, pages 191–198, Los Angeles, California, USA, August 1995.

[89] B. A. Nayfeh. Cellular automata for solving mazes. *Doctor Dobb's Journal*, 18(2):32–38, February 1993.

[90] M. Nienhaus and J. Döllner. Sketchy drawings. In *Proceedings of AFRIGRAPH '04*, pages 73–81, Stellenbosch, South Africa, November 2004.

[91] M. Nixon and A. Aguado. *Feature Extraction and Image Processing.* Newnes (an imprint of Elsevier), 2002.

[92] J. Northrup and L. Markosian. Artistic silhouettes: A hybrid approach. In *Proceedings of NPAR '00*, pages 31–37, Annecy, France, June 2000.

[93] V. Ostromoukhov. Digital facial engraving. In *Proceedings of SIGGRAPH '99*, pages 417–424, Los Angeles, California, USA, August 1999.

[94] H. Pedersen and K. Singh. Organic labyrinths and mazes. In *Proceedings of NPAR '06*, pages 79–86, Annecy, France, June 2006.

[95] L. Prasad and A. N. Skourikhine. Vectorized image segmentation via trixel agglomeration. *Pattern Recognition*, 39(4):501–514, April 2006.

[96] E. Praun, H. Hoppe, M. Webb, and A. Finkelstein. Real-time hatching. In *Proceedings of SIGGRAPH '01*, pages 581–586, Los Angeles, California, USA, August 2001.

[97] T. Pudet. Real time fitting of hand-sketched pressure brushstrokes. *Computer Graphics Forum*, 13(3):205–220, August 1994.

[98] R. Raskar, K.-H. Tan, R. Feris, J. Yu, and M. Turk. Non-photorealistic camera: Depth edge detection and stylized rendering using multi-flash imaging. *ACM Transactions on Graphics*, 23(3):679–688, August 2004.

[99] R. Rosales and S. Sclaroff. Inferring body pose without tracking body parts. In *Proceedings of CVPR '00*, volume 2, pages 721–727, Hilton Head Island, SC, USA, June 2000.

[100] G. Rote. Computing the minimum Hausdorff distance between two point sets on a line under translation. *Information Processing Letters*, 38(3):123–127, 1991.

[101] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut"- interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, August 2004.

[102] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake. Digital Tapestry. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 589–596, San Diego, CA, USA, June 2005.

[103] D. Rudolf, D. Mould, and E. Neufeld. Simulating wax crayons. In *Proceedings of PG'03*, pages 163–172, Canmore, Alberta, Canada, October 2003.

[104] M. A. Ruzon and C. Tomasi. Alpha estimation in natural images. In *Proceedings of IEEE CVPR 2000*, volume 1, pages 18–25, Hilton Head Island, South Carolina, USA, June 2000.

[105] R. Sabin. *Comics, Comix & Graphic Novels: A History Of Comic Art.* Phaidon Press, 2001.

[106] S. Saito, A. Kani, Y. Chang, and M. Nakajima. Generation of varying line thickness. In *Proceedings of Computer Graphics International*, pages 294–299, Tokyo, Japan, July 2003.

[107] T. Saito and T. Takahashi. Comprehensible rendering of 3-D shapes. *Computer Graphics*, 24(4):197–206, 1990.

[108] M. Salisbury, M. Wong, J. Hughes, and D. Salesin. Orientable textures for image-based pen-and-ink illustration. In *Proceedings of SIGGRAPH '97*, pages 401–406, Los Angeles, California, USA, August 1997.

[109] M. P. Salisbury, S. E. Anderson, R. Barzel, and D. H. Salesin. Interactive pen-and-ink illustration. In *Proceedings of SIGGRAPH '94*, pages 101–108, Orlando, Florida, USA, July 1994.

[110] Y. Sato, T. Fujimoto, K. Muraoka, and N. Chiba. Stroke-based Suibokuga-like rendering for three-dimensional geometric models -Ten and Shun touches-. *The Journal of the Society for Art and Science*, 3(4):224–234, 2004.

[111] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1–3):7–42, 2002.

[112] A. Secord, W. Heidrich, and L. Streit. Fast primitive distribution for illustration. In *Proceedings of EGRW '02*, pages 215–226, Pisa, Italy, June 2002.

[113] J. Shi and C. Tomasi. Good features to track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593–600, Seattle, Washington, USA, June 1994.

[114] M. Shiraishi and Y. Yamaguchi. An algorithm for automatic painterly rendering based on local source image approximation. In *Proceedings of NPAR '00*, pages 53–58, Annecy, France, June 2000.

[115] M. Shugrina, M. Betke, and J. P. Collomosse. Empathic painting: Interactive stylization using observed emotional state. In *Proceedings of NPAR '06*, pages 87–96, Annecy, France, June 2006.

[116] M. C. Sousa and J. W. Buchanan. Computer-generated graphite pencil rendering of 3D polygonal models. *Computer Graphics Forum*, 18(3):195–207, September 1999.

[117] L. H. Staib and J. S. Duncan. Boundary finding with parametrically deformable models. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 14(11):1061–1075, November 1992.

[118] E. Stavrakis and M. Gelautz. Image-based stereoscopic painterly rendering. In *Proceedings of Eurographics Symposium on Rendering*, pages 53–60, Norrkoeping, Sweden, June 2004.

[119] S. L. Su, Y.-Q. Xu, H.-Y. Shum, and F. Chen. Simulating artistic brushstrokes using interval splines. In *Proceedings of CGIM '02*, pages 85–90, Kauai, Hawaii, August 2002.

[120] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. *ACM Transactions on Graphics*, 23(3):315–321, August 2004.

[121] D. Terzopoulos and K. Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):569–579, June 1993.

[122] L. Wang, T. Tan, H. Ning, and W. Hu. Silhouette analysis-based gait recognition for human identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1505–1518, December 2003.

[123] F. Wen, Q. Luan, L. Liang, Y.-Q. Xu, and H.-Y. Shum. Color sketch generation. In *Proceedings of NPAR '06*, pages 47–54, Annecy, France, June 2006.

[124] S. C. Wheatstone. Contributions to the physiology of vision.-Part the first. On some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philosophical Transactions of the Royal Society of London*, 128:371–394, 1838.

[125] D. J. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *CVGIP: Image Understanding*, 55(1):14–26, January 1992.

[126] G. Winkenbach and D. H. Salesin. Computer-generated pen-and-ink illustration. In *Proceedings of SIGGRAPH '94*, pages 91–100, Orlando, Florida, USA, July 1994.

[127] H. Winnemöller and S. Bangay. Rendering optimisations for stylised sketching. In *Proceedings of AFRIGRAPH '03*, pages 117–122, February 2003.

[128] M. T. Wong, D. E. Zongker, and D. H. Salesin. Computer-generated floral ornament. In *Proceedings of SIGGRAPH '98*, pages 423–434, Orlando, Florida, USA, July 1998.

[129] W. Woo, N. Kim, and Y. Iwadate. Object segmentation for z-keying using stereo images. In *Proceedings of IEEE WCC-ICSP'00*, volume 2, pages 1249–1254, Beijing, China, August 2000.

[130] C. Xu, D. L. Pham, and J. L. Prince. Medical image segmentation using deformable models. In *Handbook of Medical Imaging: Medical*

Image Processing and Analysis, chapter 3, pages 129–174. SPIE Press, 2000.

[131] C. Xu and J. L. Prince. Gradient vector flow: A new external force for snakes. In *Proceedings of CVPR '97*, pages 66–71, San Juan, Puerto Rico, June 1997.

[132] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, March 1998.

[133] H. Xu and B. Chen. Stylized rendering of 3D scanned real world environments. In *Proceedings of NPAR '04*, pages 25–34, Annecy, France, June 2004.

[134] H. Xu, N. Gossett, and B. Chen. Pointworks: Abstraction and rendering of sparsely scanned outdoor environments. In *Proceedings of EGSR'04*, pages 45–52, Norrkoeping, Sweden, Jun 2004.

[135] S. Yamamoto, X. Mao, and A. Imamiya. Colored pencil filter with custom colors. In *Proceedings of PG '04*, pages 329–338, Seoul, Korea, October 2004.

List of Figures

1.1 Ars Electronica Center Exhibition - "Gullivers World". © Pilo 5
1.2 Apparition - a dance and media performance for the stage where the dancer interacts with and controls the image and the musical environment. A collaboration between Klaus Obermaier and the Ars Electronica Futurelab © Pascal Maresch . 6
1.3 Example of form delineation. 7
1.4 Example of a silhouette portrait. 8
1.5 Montmartre, Paris: An artist cuts the silhouette portrait of a tourist. 9
1.6 Example of a shading tone scale (hatching technique). 11
1.7 Giovanni Battista Piranesi - *Veduta interna del sepolcro di S. Costanza*, etching, 1756. 12
1.8 A detail of the arch shown in Figure 1.7. 13
1.9 The example of an object handcrafted in sketched and in drawn form. 15

2.1 Realistic landscape presented in [36]. 19
2.2 Empathic Painting. An example of adjusting stylization output according to the emotional state of the viewer, presented in [115]. 21
2.3 Different styles of painterly rendering of a flower (top left), presented in [53]. 22

2.4 Loose and Sketchy Animation. The figure shows various styles that can be obtained by varying parameters as presented in [30]. 25
2.5 Illustration of (a) the input image of a flower plant and (b) a stylized texture de-emphasized rendering, presented in [98]. . . . 28
2.6 Example of a result of the real-time hatching algorithm result presented in [96]. 29
2.7 Illustration of (a) the input doll image and (b) a drawn-like representation in colored pencil style, presented in [80]. 30
2.8 A result of the drawn-like depiction approach presented in [112]. The output is a hatched rendering of Cezanne's *Still Life With Apples* used as input. 31

3.1 Wheatstone's (left) and Brewster's (right) stereoscope. 34
3.2 Example of the stereo image pair: (a) Left camera image, (b) right camera image. 36
3.3 Disparity and depth relation for parallel cameras. [40] 37
3.4 Pixel-to-Pixel stereo: Resulting disparity map for the input stereo pair shown in Figure 3.2. 39
3.5 Graph-cut based stereo: Resulting disparity map for the input stereo pair shown in Figure 3.2. 40
3.6 Example of a SnakeToonz animation along with its corresponding video frames as presented in [2]. 42
3.7 (a) Convergence of a snake using the GVF external forces shown in (b). A close view of forces within the concave boundary region is given in (c). (After [132].) 48
3.8 Bézier curve examples constructed using (a) four and (b) six control points. 51
3.9 A cubic Bézier curve from Figure 3.8, shown with its convex hull and variation diminishing property. 52

4.1 Overview of processing chain. 56

4.2 Searching process in constructing the “basic” Edge Combination image. 57
4.3 (a) Part of the original edge image, (b) Edge Combination image window before linking, (c) Edge Combination image window after linking and before cleaning, (d) Edge Combination image window after linking and cleaning. 58
4.4 Window cleaning. 59
4.5 Edge Combination example 1: (a) Left camera image, (b) right camera image, (c) disparity image, (d) disparity edge image, (e) original edge image, (f) Edge Combination image. 61
4.6 Edge Combination example 2: (a) Left camera image, (b) right camera image, (c) disparity image, (d) disparity edge image, (e) original edge image, (f) Edge Combination image. 62
4.7 Edge Combination example 3: (a) Left camera image, (b) right camera image, (c) disparity image, (d) disparity edge image, (e) original edge image, (f) Edge Combination image. 63
4.8 Object segmentation example 1: (a) Original image with snake initialization, (b) final snake on original image, (c) final snake on disparity image, (d) original image with snake from (c) overlaid, (e) final snake on Edge Combination image, (f) original image with snake from (e) overlaid. 67
4.9 Object segmentation example 2: (a) Original image with snake initialization, (b) final snake on original image, (c) final snake on disparity image, (d) original image with snake from (c) overlaid, (e) final snake on Edge Combination image, (f) original image with snake from (e) overlaid. 68

5.1 Direction and curvature estimation where $f = 1$. 74
5.2 Example of (a) a curve with (b) its Bézier curve representation and dominant points. 74
5.3 (a) The curve from Figure 5.2, (b) result of the simplification. 75
5.4 Control points initialization. 76

5.5 (a) Final maximum error measured by the Hausdorff distance between the approximated curve A and curve B, (b) final style of the line. 78
5.6 Example 1 of the sketched-like image representation: (a) Left camera image, (b) right camera image, (c) disparity image, (d) original edge image, (e) Edge Combination image, (f) disparity edge image, (g) sketched image. 80
5.7 Example 2 of the sketched-like image representation: (a) Left camera image, (b) right camera image, (c) disparity image, (d) original edge image, (e) Edge Combination image, (f) disparity edge image, (g) sketched image. 81
5.8 Example 3 of the sketched-like image representation: (a) Left camera image, (b) right camera image, (c) disparity image, (d) original edge image, (e) Edge Combination image, (f) disparity edge image, (g) sketched image. 82

6.1 Overview of processing steps. 85
6.2 (a) Left camera image, (b) enhanced image. 87
6.3 Process of disparity layer orientation estimation: (a) disparity image, (b) square window of the disparity layer, (c) rescaled values of the window in (b) from black to white, (d) estimated angle θ of the disparity layer. 90
6.4 Example of the drawn-like result without disparity information. 91
6.5 Example 1 of the drawn-like image stylization: (a) Left camera image, (b) right camera image, (c) disparity image, (d) Edge Combination image, (e) disparity layers, (f) color segmented image, (g) drawn image. 94
6.6 Example 2 of the drawn-like image stylization: (a) Left camera image, (b) right camera image, (c) disparity image, (d) Edge Combination image, (e) disparity layers, (f) color segmented image, (g) drawn image. 95

6.7 Example 3 of the drawn-like image stylization: (a) Left camera image, (b) right camera image, (c) disparity image, (d) Edge Combination image, (e) disparity layers, (f) color segmented image, (g) drawn image. 96
6.8 Example 4 of the drawn-like image stylization: (a) Left camera image, (b) right camera image, (c) disparity image, (d) Edge Combination image, (e) disparity layers, (f) color segmented image, (g) drawn image. 97
6.9 Results of the different drawn-like depiction styles for the image set given in Figure 6.8. 98

7.1 Example 1 from the "Alan Ford" comic book series. © Copyright 2005 by Max Bunker Press. Used by permission. 100
7.2 Intermediate frames (from top left to bottom) showing each fifth frame of the original sequence used for computation of the contours in our tests. 102
7.3 (a) Original image from the sequence shown in Figure 7.2, (b) enhanced original image, prepared for further stylization. . . . 103
7.4 Example 2 from the "Alan Ford" comic book series. © Copyright 2005 by Max Bunker Press. Used by permission. 105
7.5 (a) Points to track, (b) motion path of the selected point, (c) circle fitted to the motion path, (d) selected new point on the contour in the Edge Combination image and its path. 106
7.6 (a) Left camera image, (b) right camera image, (c) disparity image, (d) Edge Combination image, (e) layers image, (f) color segmented image. 109
7.7 (a) Drawn-like image representation for the test set shown in Figure 7.6, (b) the same drawn-like image representation with motion depicted through motion lines. 110
7.8 Results of motion depiction through multiple contours. 111
7.9 Results of motion depiction through multiple contours and motion lines. 111

7.10 (a) Left camera image, (b) right camera image, (c) drawn image with depicted motion. 112

Curriculum Vitae

Danijela Marković was born on 4th September 1970, in Niš, Serbia. She studied at the Faculty of Electronic Engineering, University of Niš, where she obtained her Dipl.Ing. degree in 1997. In 2002 she moved to Vienna and joined the Interactive Media Systems group at the Institute for Software Technology and Interactive Systems, Vienna University of Technology, where she was involved in the project "3D Video Analysis for Interactive Multimedia Applications (Video3D)", funded by the Austrian Science Fund (FWF). She was also involved in the project entitled "Bild- und Videoverarbeitung zur Erzeugung künstlerischer Effekte" (Image and Video Analysis for Artistic Effects), which was focused on the development of algorithms for non-photorealistic rendering of real images and video scenes, granted by the Hochschuljubiläumsstiftung der Stadt Wien. In 2005 she also joined Wissenschafterinnenkolleg Internettechnologien - WIT (Women's Postgraduate College for Internet Technologies), established at the Vienna University of Technology as a part of a series of projects promoting women in science and technology (fFORTE), initiated by the Austrian Council for Research and Technology Development. Her research interests are in computer vision and computer graphics. Particularly, she is interested in feature extraction, image based rendering, image stylization and in abstract representation of images of natural scenes.