ADVANCED TOPICS IN SCIENCE AND TECHNOLOGY IN CHINA

Weidong Geng

The Algorithms and Principles of Non-photorealistic Graphics

Artistic Rendering and Cartoon Animation

ZHEJIANG UNIVERSITY PRESS
浙江大学出版社
Springer

ADVANCED TOPICS
IN SCIENCE AND TECHNOLOGY IN CHINA

ADVANCED TOPICS
IN SCIENCE AND TECHNOLOGY IN CHINA

Zhejiang University is one of the leading universities in China. In Advanced Topics in Science and Technology in China, Zhejiang University Press and Springer jointly publish monographs by Chinese scholars and professors, as well as invited authors and editors from abroad who are outstanding experts and scholars in their fields. This series will be of interest to researchers, lecturers, and graduate students alike.

Advanced Topics in Science and Technology in China aims to present the latest and most cutting-edge theories, techniques, and methodologies in various research areas in China. It covers all disciplines in the fields of natural science and technology, including but not limited to, computer science, materials science, life sciences, engineering, environmental sciences, mathematics, and physics.

Weidong Geng

The Algorithms and Principles of Non-photorealistic Graphics

Artistic Rendering and Cartoon Animation

With 314 figures, mostly in color

ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

Springer

Author
Prof. Weidong Geng
State Key Lab of CAD&CG
Digital Media Technology Department
College of Computer Science
Zhejiang University, Hangzhou
310027, China
E-mail: gengwd@zju.edu.cn

ISSN 1995-6819 e-ISSN 1995-6827
Advanced Topics in Science and Technology in China

ISBN 978-7-308-06600-6
Zhejiang University Press, Hangzhou

ISBN 978-3-642-04890-6 e-ISBN 978-3-642-04891-3
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2009935705

Cover design: Frido Steinen-Broo, EStudio Calamar, Spain

Printed on acid-free paper

Springer is a part of Springer Science+Business Media (www.springer.com)

Preface

Non-photorealistic computer graphics is a multidisciplinary field in the research community, involving computer arts, computer graphics, computer vision, digital image/video processing and visual cognitive psychology. It aims at the computer generation of images and animations that are made in part “by hand” in appearance, and are characterized by their use of randomness, abstraction, ambiguity, or arbitrariness rather than completeness and adherence to the portrayed objects’ properties. In essence, it mimics the eyes and minds of artists and designers to create, view and depict the graphics world, effectively carrying-out the visual communication between computers and human beings.

Coverage and Audience

This book mainly focuses on the following five core issues in non-photorealistic computer graphics.

(1) How to create the paintings, artworks or sculptures from a digitized blank canvas or a standard shape with the tools simulated by the computer.
(2) How to convert a series of reference images into the resultant depiction with the desired visual effect.
(3) How to automatically generate the artistic rendition or technical illustrations from the 3D models in terms of the stylized parameters.
(4) How to produce the comprehensive and expressive visualizations from a set of graphical and textual information on the basis of the semantic meanings to be conveyed.
(5) How to speed up the production of cartoon animation by computer-assisted refinement of traditional pipeline and the exploration of novel approaches.

The author not only take a survey of the state-of-the-art research as well as trends and open-ended questions regarding the aforementioned five core issues, but also discuss the theoretical underpinnings of the field. This includes detailing a host of useful algorithms and addressing two applications of particular interest: artistic rendering and cartoon animation.

The book will be useful to practitioners in the field. It contains a wealth of examples, particularly in the form of images, which the authors hope will motivate the reader in the use of non-photorealistic computer graphics. The methods introduced are explained in enough detail so that programs can be written directly without a major conceptual effort.

Anothers use of the book is for reference by researchers in the field. The bibliographic references at the end of the chapters give the necessary pointers to the important publications. In the case of researchers in the field of non-photorealistic computer graphics, the methods that are built up are referenced appropriately, and a comprehensive index aids in selective readings.

Objective

Non-photorealistic computer graphics is a relatively young field, and new works are constantly being published. The intent of this book is to bring together a coherent conceptual framework for all of the research to date in the context of computer graphics, art history and theory, and cognitive psychology.

Although the field of non-photorealistic rendering has existed for more than two decades, it has for a long time not been taken seriously by large parts of the research community. The area has thus far been unstructured, making it increasingly difficult to identify and assess new open problems. Indeed, sometimes papers have even "reinvented the wheel," albeit in a different context and application concern. Recent years have seen many algorithms, papers, and software tools devoted to artistic rendering and computer-assisted cartoon animation. The time has become ripe for a systematic assessment of the literature. The following are our goals:

(1) To become the seminal reference for core issues surrounding artistic rendering and cartoon animation.
(2) To describe and review state-of-the-art advances in the field of non-photorealistic computer graphics, and to distill the breadth of cutting-edge non-photorealistic modeling, rendering and animation technologies into a coherent, accessible treatise.
(3) To provide the guidelines for researchers and software developers to assess and implement the best solution for their interactive arts application.

Acknowledgement

I would like to thank my family for their love, support, and patience, and a few of the great many people for their instrumental contributions in taking this book from an initial suggestion to a final product. I express my gratitude to Prof. Yunhe Pan, fellow of Chinese Academy of Engineering, for his time reviewing this book and contributing his many deep insights into the topic in the final chapter of the book.

A number of colleagues in the State Key Lab of CAD & CG at Zhejiang University gave the author support, offered constructive criticisms, or provided illustrations to include in the book. These include Profs. Hujun Bao, Yueting Zhuang, Dongming Lu, and Duanqing Xu. I am also indebted to the graduate students, faculty, and researchers in the research group of Computer Animation and Perception at Zhejiang University.

I wishes to thank those persons who provided the support to make this book appear, including the administrative, technical, and secretarial staff in our college who kept things up and running, even under adverse workloads (Ms Qi Shen, Ms Xuefang Zhang, etc.). I also thanks the graduate students at Zhejiang University who studied the topic with previous versions of the manuscript, and all of the colleagues and peers around the world who did great research and gave me the copyrights of their images.

The continuous research and development on non-photorealistic computer graphics in the past years were supported by the following grants: National Natural Science Foundation of China (Grant Nos.: 6960302, 69973044, 60373032, 60773183 and 60633070), National High Technology Research and Development Program of China (Grant Nos.: 2006AA01Z313 and 2006AA01Z335), New Century Excellent Talents in University (NCET-07-0743), and Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT0652).

Finally, I also acknowledges the prodigious work of the people at Zhejiang University Press and Springer.

Weidong Geng
Hangzhou, China
August 16, 2009

Contents

1 Introduction 1
1.1 The Brief History: from Photorealism to Non-photorealism . . 1
1.2 What is Non-photorealistic Computer Graphics 2
1.2.1 Image, Picture, and Visualization 3
1.2.2 Photorealistic versus Non-photorealistic Rendering . . . 4
1.3 The Framework for Non-photorealistic Computer Graphics . . 6
References 9

2 Simulating Artistic Media for Digitized Creation of Artworks 11
2.1 Stroke-based Artistic Drawing 11
2.1.1 Interactive Drawing Based on Brushstrokes 11
2.1.2 Pen-and-ink Illustration by Stroke Textures 15
2.1.3 Interactive Pencil Drawing 18
2.1.4 Simulating Wax Crayons 26
2.2 Oriental Calligraphy and Black Ink Painting 30
2.2.1 Modeling of Soft Brushes 31
2.2.2 Calligraphy with Soft Brushes 39
2.2.3 Oriental Black Ink Painting 43
2.3 Simulation of a Colored Painting 54
2.3.1 The Computational Model of Rendering Colored Pigments 55
2.3.2 Simulation of Watercolor Painting 58
2.3.3 Simulation of Oil Painting 68
2.4 Digitized Sculpting 77
2.4.1 Digitized Wood Sculpting 78
2.4.2 Digitized Metal Embossing 81
2.5 Creation of Artwork in a Virtual Environment 83
2.6 Summary 86
References 87

3 Computer-aided Design of Art Patterns 91
3.1 The Overview of Art Pattern Design 92
3.2 Art Pattern Creation by Fractals 94
3.3 Art Pattern Creation by Shape Grammars 96
3.4 Layout-based Creation of Art Patterns 100
3.4.1 Novel Layout Creation Based on Regular Structures .. 100
3.4.2 Layout Creation Integrated with Fabrication Craft ... 103
3.4.3 Art Pattern Creation by Stylized Layout 104
3.5 Knowledge-based Creation of Art Patterns 106
3.5.1 Aesthetical Knowledge Representation for Art Pattern Generation 107
3.5.2 Generation of Art Patterns by Synthesis Reasoning ... 108
3.6 Summary 111
References 111

4 Artistic Painting by Reference Images 113
4.1 Artistic Effect Generation by Pixel-level Image Processing ... 113
4.1.1 Artistic Processing via Digital Half-toning 114
4.1.2 Artistic Processing with Heuristic Rules 115
4.2 Converting Images into Artistic Painting by Strokes 118
4.2.1 Image-based Stippling Drawing 120
4.2.2 Image-based Mosaic and Stained Glass Simulation 124
4.2.3 Image-based Pen-and-ink Illustration 127
4.2.4 Image-based Pencil Drawing 131
4.2.5 Image-based Oriental Painting 137
4.2.6 Image-based Colored Painting 138
4.3 Artistic Transfer of Color and Texture from Reference Images 145
4.3.1 Artistic Transfer of Color 146
4.3.2 Artistic Transfer of Texture 149
4.4 Image-based Painting Driven by Examples 152
4.4.1 Painting Style Simulation by Image Analogy 153
4.4.2 Artistic Painting Generation by Stroke Templates from Examples 155
4.5 Summary 157
References 158

5 Artistic Rendering for 3D Object 161
5.1 Artistic Rendering Based on Traditional 3D Rendering Pipeline 162
5.1.1 Non-photorealistic Lighting Model 162
5.1.2 Non-realistic Projection 168
5.1.3 Non-photorealistic Texture Mapping and Synthesis ... 174
5.2 Non-photorealistic Rendering with Interim Images 179
5.2.1 Pen-and-ink Illustration from 3D Object 180

5.2.2 Pencil Drawing from 3D Polygonal Object 185
5.2.3 Chinese Painting from 3D Model 187
5.2.4 Colorful Painting from 3D Model 191
5.3 Artistic Silhouette and Line-drawing for 3D Model 192
5.3.1 Silhouette Rendering from 3D Polygonal Object 192
5.3.2 Line-drawing for 3D Surface 198
5.4 Artistic Rendering for 3D Landscape 201
5.4.1 Artistic Simulation by Placing Artistic Particles and Strokes on the 3D Surface 202
5.4.2 Artistic Rendering of Plants Based on Their 3D Structure 204
5.4.3 Artistic Rendition for Point-based Models 209
5.5 Artistic Illustration of Volume-based Models 212
5.5.1 Artistic Rendering of Surface Features of Volume Model 213
5.5.2 Artistic Illustration of Internal Structure of Volume Model 217
5.6 Summary 223
References 224

6 Expressive Rendering 229
6.1 Comprehensible Rendering 230
6.1.1 Enhancement of Rendering by Emphasis of Visual Cues 230
6.1.2 Cutaway Illustration 236
6.1.3 Comprehensive Rendering via Composite Viewpoints . 239
6.2 Shape-conveying Illustrations 241
6.2.1 Expressing Shape Features via Stroke Texture 241
6.2.2 Expressing Shape Features by Selective Depiction 243
6.3 Intent-based 3D Illustration 249
6.3.1 Intent Communication by Multimedia Presentation . . . 250
6.3.2 Interpreting Intent by Graphical Abstraction 252
6.3.3 Visual Explanation of Iconic Symbols 254
6.4 Expressive Rendering for Transparency 255
6.5 Summary 259
References 260

7 Computer-assisted Cartoon Animation by Traditional Production Pipeline 263
7.1 The Traditional Animation Process 263
7.2 The Role of the Computer in Traditional 2D Animation 268
7.3 Computer-assisted Coloring 270
7.3.1 Auto-coloring of Inbetweening Frames 271
7.3.2 Colorizing Black-and-white Cartoons 277
7.4 Computer-assisted Inbetweening 282

7.4.1 Generation of Inbetween Poses 282
7.4.2 Generation of inbetweening Facial Expression 288
7.5 Summary 290
References 291

8 Novel Approaches to Computer-assisted Cartoon Animation 293
8.1 Video Driven Cartoon Animation 294
8.1.1 Directly Converting Video Segment into Painterly Animation 294
8.1.2 Contour-based Animation Generation from Video 300
8.1.3 Video-driven Facial Animation with Style 303
8.2 Cartoon Production Integrated with 3D Geometric Elements . 310
8.2.1 Automated inbetweening by the 3D Approximate Model 311
8.2.2 Cartoon Animation by 3D Canvas 315
8.2.3 Cartoon Animation by View-dependent Geometry 318
8.3 Cartoon Generation by Temporal Coherent Rendering 322
8.4 Cartoon Generation Together with 3D Graphical Processing Techniques 327
8.4.1 Cartoon Production Integrated with 3D Transformation 327
8.4.2 Cartoon Animation Enhanced with Artistic Texture Mapping 330
8.4.3 Cartoon Motions Augmented by 3D Physical Models . . 331
8.4.4 Stylized Highlight and Shadow Generation for Cartoon Animation 334
8.5 Cartoon Production via Reuse of Traditional Animation 336
8.5.1 Reusing Visual Style for Cartoon Animation 337
8.5.2 Reusing Motions for Cartoon Animation 340
8.6 Summary 343
References 345

9 Perspectives of Non-photorealistic Computer Graphics . . . 349
References 353

Index 355

1

Introduction

Non-photorealistic computer graphics are used to imitate the eyes and minds of artists and designers to create, view, and depict the graphical world. These computer-generated graphics are used instead of achieving the illusion of photorealism via an optical camera. With the advent of many algorithms, papers, and software tools dedicated to generating the artistic and meaningful images, the entire field was exploded into existence in the 1990s. Now the field appears to be approaching maturity. Many questions remain open, but many have been settled. This book presents a detailed treatment of this field in a coherent conceptual framework.

1.1 The Brief History: from Photorealism to Non-photorealism

Photorealism in the context of computer graphics is a "faithful" rendering of the material world based on a number of depiction principles, such as convincing details, anatomical correctness, correct color rendition, and the correct perceptions of space, volume, and texture, etc. Therefore this field of computer graphics is also called *photorealistic rendering,* denoting algorithmic techniques that resemble the output of a photographic camera even make use of the physical laws being involved in the process of photography. A truly photorealistic image needs to be generated accurately from an extremely detailed object description requiring a great modeling effort. For the time being, a vast number of different computational models have been explored that to approximate these physical processes. The creation of realistic pictures has made great progress in the computer graphics community. This can be judged by viewing feature films and TV commercials, where it is often impossible for the audience to decide which are the virtual objects generated by the computer and which are the real objects captured by the camera.

It is no doubt that highly realistic graphics are very useful, e.g., they can support designers to evaluate and refine new products and turn computer games into a more enjoyable experience. But in general photorealism considers only part of the imagery traditionally used in simulation, design, entertainment, advertising, research and education, etc. For instance, it may be useful for designers to be able to generate photorealistic images of the finished product. But, during the design process they prefer to work with sketches and conceptual drawings that are better suited for explaining the basic concept of a new product or showing its inner structure. In educational course books, most of the pictures are not photographs, but rather diagrams and illustrations that are better able to communicate the important aspects of a topic. Furthermore, there are many research areas that can benefit from automatically generated images based on purely abstract data. But, how can one create photorealistic images of data that have no counterpart in the visual world? As computer graphic is getting closer to its holy grail of achieving photorealism, people finally realizes that there is more to images than realism, and, computer-generated imagery should not be restricted to photorealistic renderings.

Thus a new type of quest has emerged—creating imagery that is more effective at conveying information, expressive or beautiful—rather than just being physically realistic. Researchers started to explore alternative rendering techniques other than mimicking the effect of a traditional photographic camera. They needed to differentiate themselves from the rest of the computer graphics community, and *non-photorealism* was thus proposed. From the point of view of rendering an image, non-photorealistic images can be anything from a drawing or a diagram to a painting, as long as it helps to communicate the intended idea.

1.2 What is Non-photorealistic Computer Graphics

As with many new and young areas of scientific endeavor, there is no uniform definition of what we have called non-photorealistic computer graphics. The border between photorealism and non-photorealism is also fuzzy. Examining the primary literature on the topic, a number of different points of view have been summarized as follows [Gooch & Gooch, 2001; Strothotte & Schlechtweg, 2002]:

(1) The process of image production that is being mimicked (or non-photorealistic to be more precise, processes that are definitely not being mimicked): *non-photorealistic rendering*.
(2) The freedom not to have to reproduce the appearance of objects precisely as they are: *non-realistic rendering*.
(3) The process of adapting presentation to a dialog context and the dynamic informational wishes of users: *abstraction*.
(4) A specific drawing style: the terms *sketch rendering*, *pen-and-ink illustration*, and *stipple rendering* are examples.

(5) The effect a rendition has (or will hopefully have) on its viewers: *comprehensible rendering*.
(6) The use of renditions for conveying information, perhaps in the context of other media of expression: *illustrative rendering*, or *expressive illustration*.
(7) The possible deformations of images: *elastic presentations*.

In order to better explain non-photorealistic computer graphics, we will first explore the fundamental concepts of image, picture, and visualization for visual representation. We will then further discuss the essential aspects of non-photorealistic computer graphics by comparing the photorealistic and non-photorealistic computer graphics in terms of their goals and algorithmic techniques.

1.2.1 Image, Picture, and Visualization

Image, picture, and visualization are the different levels of visual representations. They are often mixed when used to describe the resulting output of a rendering in the computer graphics community. In order to help readers to better understand the rest of this book, these vocabularies should be clarified from the point of view of computer depiction. Computer depiction deals with all aspects of picture production, encompassing both photorealistic and non-photorealistic styles. Based on the definitions from the Webster dictionary, the differences between image, picture, and visualization are given as follows [Durand, 2002]:

(1) *Image*. An image is a "reproduction or imitation", or "the optical counterpart of an object" [Webster, 1983]. It is an optically formed duplicate, characterized by optical accuracy to a visual scene or object.
(2) *Picture*. A picture is "a design or representation", or "a description so vivid or graphic as to suggest a mental image or give an accurate idea of something" [Webster, 1983]. A picture is more loosely defined than an image, and it corresponds to both to the graphical object and to a representation. Pictures always have a purpose, which can be a message, collaborative work, education, aesthetics, emotions, etc. The term "picture" can be used to describe a visual representation of a visual scene, but this representation is not necessarily optically accurate. Moreover, a picture is not necessarily the representation of an existing real scene or object. The extreme example of impossible figures shows that a picture can superficially look like the representation of a 3D reality, while no objective scene that can be projected to such a picture.
(3) *Visualization*. Visualization is "the act or process of interpreting in visual terms or of putting in visual form" [Webster, 1983]. A visualization can represent visually data or subjects that are not themselves visual. Visualization therefore mainly relies on metaphors to communicate the meaningful information to the audience.

1.2.2 Photorealistic versus Non-photorealistic Rendering

The major goal of photorealistic rendering is to generate images that mimic the effect of a traditional photographic camera. It depicts only "What I See"—the extrinsic properties of objects such as outgoing light varying with light conditions, and the resultant output is a photography-like image. Its rendering process is a unidirectional optical projection of a 3D model onto a 2D plane. A scene consisting of 3D objects is illuminated by a number of virtual light sources, and images are generated by a virtual camera that is placed in the scene. The idea is to generate 2D images of the scene by emitting light from the light sources into the scene, computing the interaction of the light with the surface of the 3D models, and capturing that portion of the light that reaches the camera on a virtual film plate.

Non-photorealistic rendering (NPR), not only depicts "What I See", but also depicts "What I Know"—the intrinsic parameters and constancy that are invariant and constant properties of the objects such as reflectance and relative sizes. This gives freedom to encode an impression of the scenes rather than being forced to follow physical constraints. Its resulting output is a hybrid picture balanced between extrinsic and intrinsic properties of objects. The NPR process is a bi-directional interaction between a 3D model and a 2D plane, involving feedback and influence from the picture space to the object space. Therefore the NPR is essentially becoming a very complex optimization problem, producing the best picture with back-and-forth exchanges, given constraints, and goals linking the scene and the picture.

The function to minimize image information, and the degrees of freedom to vary it, heavily depend on the rendering of context and goal. For example, the goals and constraints for picture creation of art and craft are often set by the medium, the social context, the artistic fashion, clarity, representation of intrinsic vs. extrinsic qualities, 2D layout, etc. There are three main strategies to solve this optimization problem. The user can solve it, the computer can solve it, or the solution might involve both user and computer decisions. All approaches are of course not contradictory and can be blended. The frequently used case is the mixed one. The computer has to make decisions automatically, but the user needs to keep some control and influence the decisions. For example, in game and movie making, it is the equivalent of the movie director wanting to keep control of the style of the pictures, and the computer has to respond automatically to the user's interaction.

As a summary, the differences between photorealistic and NPR are investigated as follows:

(1) *Content of rendition.* Photorealistic rendering is merely based on the 3D geometry and topological information of the scene, and the resulting image is an "objective" depiction of that scene, and nothing else. In contrast, NPR encodes the "subjective" artifacts into the picture that clearly do not exist in the world. These artifacts may stem from the way

in which the geometric model represents the original object, or result from the manner, or style in which the geometric model is rendered.

(2) *Manners for presentation.* In photorealistic rendering, the external world is presented in an "objective" way. The depiction corresponds exactly to the object being modeled, following physical constraints and leaving nothing for the imagination. However, the presentation manner in NPR is a graphical abstraction such that the resultant picture comes from, and is "higher" than the underlying models, with certain features of the model being enhanced. It gives freedom to encode an impression of scenes, and introduce a broader variety of styles. This not only enables better recognition of certain features of the object being modeled by changing the model, but also enables selected features of the geometric model to be exaggerated in the rendition, in order to emphasize them. Moreover, it can show more of the relevant parts of an object than what would otherwise be possible, while less relevant parts may no longer be visible.

(3) *The cognition process of the resulting depiction.* The output of photorealistic rendering comes from the intuitive observation of the real world, and its cognition process is consistent with the visual perception of human beings in their daily lives. However, a reasoning process is needed to interpret non-photorealistic images. It is assumed that the viewers are able to build up a mental model of the object being portrayed with creative thinking and imagination, and then to perform the cognitive process for the visual understanding of the NPR results. This reasoning process gives the greatest communicative power for NPR.

(4) *The algorithmic mechanism.* The photorealistic rendering technique is based on the working model of another kind of machine, a camera. It simulates the particle-by-particle lighting exposure principle with the pixel-by-pixel rendering mechanism. The correspondence between pixels and the drawn primitive object is direct. In contrast to the pixel-by-pixel mechanism, NPR employs a relative global mechanism beyond pixels, and paints. The resulting picture is in a region-by-region mode. Each region has a set of pixels with attributes of shape, an area as a whole. These regions may be formed by a stroke, or more generally, may come from the interactional areas between the pen/brush with the canvas.

(5) *The interplay between 3D and 2D aspects of depiction.* Photorealistic rendering is a unidirectional projection from a 3D objective scene onto a 2D image. The typical object space inputs are a 3D geometric description of the objects, their material properties, and light sources. Perspective matrices, hidden-surface removal, and lighting simulation are then used to project this model onto the 2D image. However, the NPR is a complex bidirectional process between the 2D picture and the 3D model. A typical feedback loop is that the user and the computer work together, cooperatively generating an initial picture, viewing it, assessing the qualities, and then re-generating the new interim pictures via necessary modifica-

tions and refinements. The process is iterated, and the final picture is retouched until it looks right.

1.3 The Framework for Non-photorealistic Computer Graphics

The default tendency of non-photorealistic computer graphics is to generate imagery that superficially looks like that made by artists [Lansdown & Schofield, 1995]. It involves a fundamental issue of simulating the intelligence of artists, i.e., to emulate human facilities for producing an artist's handwork. Artists and other picture makers have developed a rich set of techniques to produce effective pictures. Non-photorealistic computer graphics should learn from this large body of knowledge, as well as from the analysis performed in the perception community. However, fine arts are still believed to be of a purely "metaphysical" nature and that there is no underlying theoretical knowledge of them. Every creative act is partly guided by intangible "forces" and "feelings" that are not easily translatable to algorithms.

Non-photorealistic computer graphics not only has been concerned with simulating traditional drawing and painting techniques, but also aims at improving visualization based on the findings from cognitive psychology. Conveying meaning is beyond scientific curiosity for pursuing NPR. There is ample evidence that non-photorealistic renditions are in fact more effective for communicating specific information than photographs of photorealistic renditions in many situations. Many studies have been carried out by cognitive and educational psychologists that attest to the superiority of such handmade graphics over photo-like images. NPR therefore enables users to lead human-computer dialogs with information exchange in a graphical form. The style of the picture generated should be flexible so as to be most appropriate for the dialog at hand. To this end, a model of information transfer must be assumed or developed. Methods and tools need to be developed to enable designers and programmers of interactive systems to have appropriate pictures rendered for their end users.

The core scientific problems in non-photorealistic computer graphics can be categorized as the following ones in terms of its input/output information.

(1) *How to create art crafts from a blank canvas.* When an artist sets out to paint a picture, he or she must have three types of physical tools. The first is a medium, such as oil paint, acrylic or watercolor, to be used to construct the picture. The second is some type of applicator or brush/pen for the application of the medium. The third is a surface, such as paper or canvas, on which the medium is applied. Therefore the computer should first model and simulate these authoring tools and the physical interaction among them, and then the user can employ the

digitized authoring tools to interactively or semi-automatically create the art craft of pen-and-ink illustration, watercolor/oil painting, or engraving.

(2) *How to convert the source images into pictures with the desirable visual effects.* By the techniques from image processing, analogical reasoning, computer vision, etc., it attempts to semi-automatically translate the input images into the resulting pictures with the desired artistic styles, which may be specified by numerical parameters, textual keywords, or the reference images.

(3) *How to generate artistic renditions from 3D models.* Its input is a 3D model of the scene, character, the viewpoint, etc., and the algorithmic steps for rendering are very similar to those in photorealistic renderings. However, the affine transformations, viewing projections, texture-mappings or lighting models are usually with non-photorealistic properties, and can help generate the output picture, which gives the visual impression of the artistic rendition styles specified by the user.

(4) *How to synthesize expressive pictures from textual, graphical or pictorial data.* The "expressive" picture embodies various levels of meaning for the communication among artists and designers. Its input might be a combination of 3D models, 2D images, or semantic text. It attempts to render objects and scenes to resemble how artists and designers might want to see them. The resulting pictures are made meaningful and comprehensible. The viewer is encouraged to make the same imaginative, perceptual contributions as those in the interpretive art.

(5) *How to accelerate the production of cartoon animation sequences with temporal coherence.* 2D animation can only be automated to the extent that the computer acts as an interactive assistant to the animator. The key problem is that the modeling, rendering and motion which are implicitly and tightly coupled in the animated drawings of key frames are unavailable. The temporal coherence problem will arise if we want to speed up the cartoon animation production by decoupling the modeling, rendering, and motion as that in 3D animation production. For example, some features of the frames in a cartoon sequence are chosen randomly (e.g., stroke placement for hatching), and they will look different in each frame. Although this may be desirable in some cases, it will in general distract the beholders' attention, and can put a considerable strain on the eyes. Similar artifacts appear if the rendering algorithm is unstable with respect to small changes of the viewing angle or small body deformations. It is therefore important to ensure that small (or no) changes in the scene result only in small changes in the cartoon animation production.

This book is accordingly organized in terms of the aforementioned five scientific core issues, providing a systematic, in-depth insight into non-photorealistic computer graphics. The structure of the book is shown in Fig. 1.1.

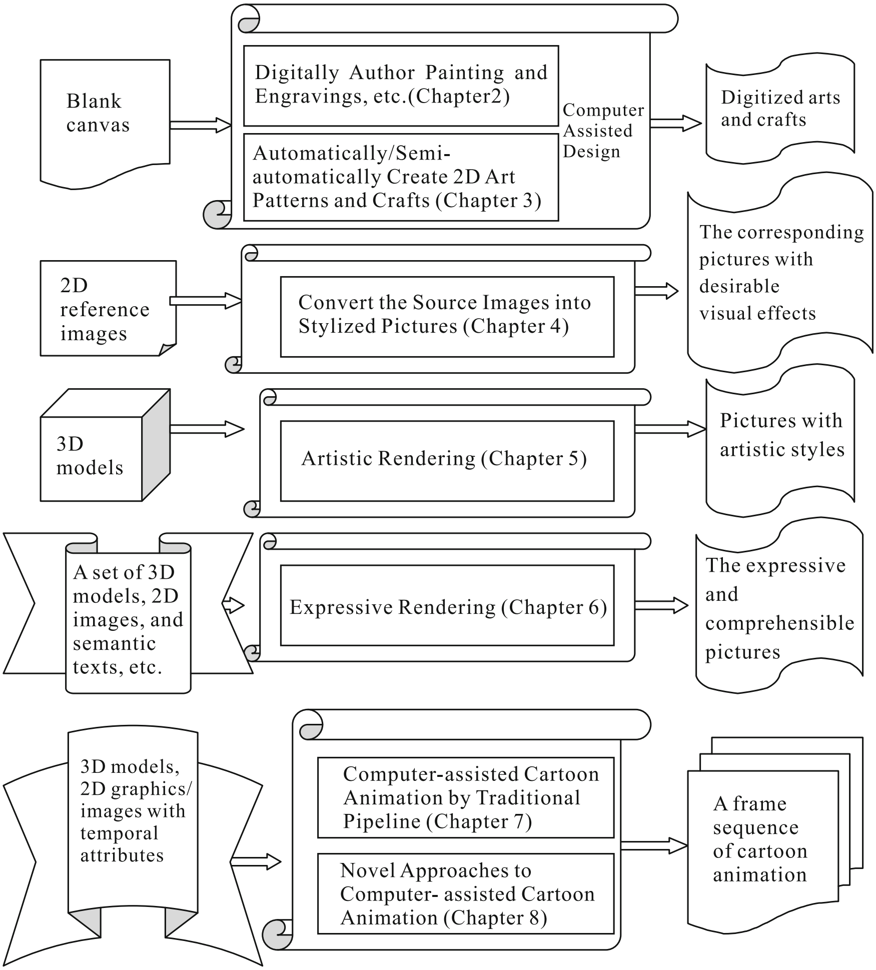


Fig. 1.1 The structure of the book based on a consistent framework of non-photorealistic computer graphics

An overview of the remaining chapters is given below:

Chapter 2 employs the *computer-aided design* (CAD) principles to enable the artist to explore the inner and outer world and our relation to them using the digitized media and tools. Its core technical issue is how to let the computer replace the paper/canvas, pigments, and pens/brushes by the natural media simulation.

Chapter 3 moves into how to automatically/semi-automatically synthesize the 2D art patterns in terms of the fractal computing, shape-grammars, spatial-layout, and the aesthetic knowledge/rules.

Chapter 4 deals with how to transform the source images into the resulting pictures with the desirable visual effects, which can be specified in three ways.

(1) The user explicitly specifies the resultant artistic style based on parameters or semantic keywords.
(2) The desirable visual effects are intuitively specified by a set of reference images/pictures.
(3) The user implicitly specifies the artistic transformation by the analogical mapping between pairs of images, instead of the final visual effect.

Chapter 5 discusses how to automatically/semi-automatically render the 3D objects or scenes into the artistic pictures. The rendition techniques include artistic simulation based on a traditional 3D rendering pipeline, conversion from the interim reference images, silhouette drawing, and the dedicated illustrational algorithms for 3D surfaces, 3D landscapes, and 3D volumes, respectively.

Chapter 6 structures and treats artistic communication methods that can visually convey meanings, purpose, intent, etc., including comprehensible rendering, expressing shape features, communications of design intent, and artistic presentation for transparent objects.

Chapter 7 describes a variety of computer-assisted cartoon animation techniques on the basis of traditional animation production pipeline, including computer-assisted auto-coloring, transforming black-and-white cartoon sequence into colorful ones, and computer- assisted "inbetweening" for cartoon characters.

Chapter 8 explores the novel approaches to assisting cartoon animation production beyond the traditional animation production pipeline, including video-driven cartoon animations, cartoon animation production guided by 2.5 D or approximate 3D geometry, cartoon animation production accelerated by artistic rendering with temporal coherence, and computer-assisted cartoon animation by reusing the graphical models, motions and rendition.

Chapter 9 concludes the book with a summary of research methodologies, scientific problems, current hot topics, and future directions in non-photorealistic computer graphics.

References

Durand F(2002) An invitation to discuss computer depiction. In: Proceedings of the 2nd International Symposium on Non-photorealistic Animation and Rendering 111–124

Lansdown J, Schofield S(1995) Expressive rendering: a review of non-photorealistic techniques. IEEE Computer Graphics and Applications 15(3): 29–37

Gooch B, Gooch A(2001) Non-photorealistic rendering. AK Peters, Ltd., Wellesly Massachusetts

Strothotte T, Schlechtweg S(2002) Non-photorealistic computer graphics: modeling, rendering, and animation. Morgan Kaufmann, San Francisco
Webster M(1983) Websters Ninth New Collegiate Dictionary. Merriam Webster, Springfield

2

Simulating Artistic Media for Digitized Creation of Artworks

One of the fundamental issues in non-photorealistic computer graphics is how to replace the natural media such as canvas, pigment, and pen/brush by the computer, in such a way that the artist can create the digital artworks by interactively manipulating these digitized artistic media. This chapter will provide in-depth coverage of algorithms for digitized drawing, painting and engraving. The relevant topics are broadly categorized into the following three parts:

(1) *Artistic tools modeling.* It describes the digitization of drawing tools (pen, pencil, crayon, charcoal, etc.), painting tools such as brushes, and engraving tools such as knives and chisels.
(2) *Modeling and simulation of natural medium and its interaction with tools.* It includes the modeling of natural media such as papers and canvases, and their interaction process with artistic tools such as pens and brushes.
(3) *Visual effect illustration for the final artwork.* It discusses the diffusion, sediment and drying process of pen-and-ink, water coloring and oil-painting, and the rendering of dried pigments.

2.1 Stroke-based Artistic Drawing

Strokes are the indivisible pictorial subunits or "building blocks" for constructing the new artistic images. It is of utmost importance to clarify how such pictorial subunits can be defined in terms of modeling their shape and attributes and how they can be rendered one by one. Here we will examine the different approaches for the creation of artistic lines, charcoal sketching, pen-and-ink, pencil-drawing and wax crayons.

2.1.1 Interactive Drawing Based on Brushstrokes

In the 1980s, the computer could generate the 2D graphical primitives in real-time. Then the graphics researchers began to extend the real-time drawing algorithms to mimic the brushstrokes by setting up the 2D lines, poly-lines, polygons and circles as the path of strokes and specifying the types of pixel distribution along the stroke path. The user can then interactively choose the various types of drawing strokes, and create the vivid pictures as shown in Fig. 2.1. The Paint on Windows platform is a typical application of it. The simplest brushstroke merely has a constant visual effect along the stroke path, ignoring the modeling of the paper and the temporal interplay of consecutive strokes.

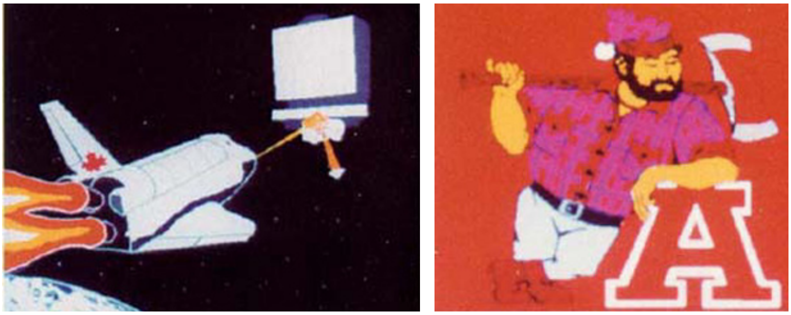


Fig. 2.1 Interactive drawing based on 2D graphical primitives [Beach *et al.*, 1982]. Copyright of ACM, used with permission

In order to enhance the painting effect of a brushstroke, the researchers started to mimic the drawing with the stroke of stylized lines. For instance, the brushstrokes can be illustrated as shining tubes by employing the depth information, and it looks like it's being drawn by a brush with a shiny ball, as shown in Fig. 2.2.

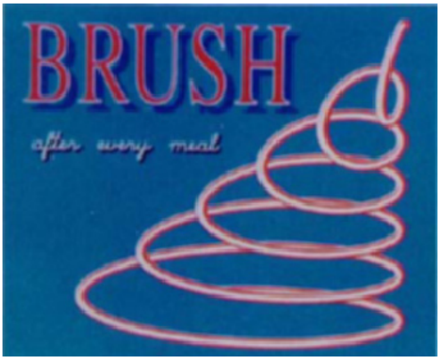


Fig. 2.2 Stylized line-drawing [Whitted, 1983]. Copyright of ACM, used with permission

From the point of view of interaction, the path of a brushstroke is usually controlled by a mouse or one of many tablet input devices that merely provide only (x, y) spatial information. The brushstroke itself is rendered with a single "brush shape" of a fixed size and orientation. However, a mouse-based interface does not support the fine control needed for detailed stroke work. A number of interruptions of the drawing act will make it difficult or impossible for the artist to maintain the kind of continuous control over his or her medium that is required for such tasks as changing "brush" shapes or the stroke styles. Thus digitized tablets with pressure-sensitive styluses become attractive devices for interactive drawing with brushstrokes. As the user draws on the tablet, the artist can change the shape of its contact point with the drawing surface. By changes in pressure exerted with the stylus against the surface, it is possible to control both the darkness (value) and width (weight) of the stroke. All of these factors are controlled by the artist in real time and with continuous feedback. Existing methods for modeling brushstrokes fall into two classes: "raster brushstroke", which models some brushstroke attributes at the pixel level and paints the result into a bitmap. The second class is the "vector brushstroke", which models a brushstroke outline and relies on scan-conversion for rendering.

The raster brushstroke approach is based on a digitization process called "brush extrusion". With the help of fast hardwired "BitBlt" operators, a bitmapped brush is dragged along a trajectory, leaving the image of the brushstroke. Fig. 2.3 shows a particular example of raster brushstroke sketching with a piece of a simulated charcoal drawing via a stylus. Raster brushstroke is well adapted to real-time sketching. Realistic models of paintings have been developed which make raster brushstrokes more expressive and simulate real paintings. Fig. 2.4 shows an example of a charcoal drawing. However, raster brushstrokes present two major drawbacks: they are resolution-dependent, and they cannot be edited. Resolution-dependence can be overcome by working at the maximum resolution of all possible output images. In any case, individual strokes cannot be easily, if at all, retouched nor edited.

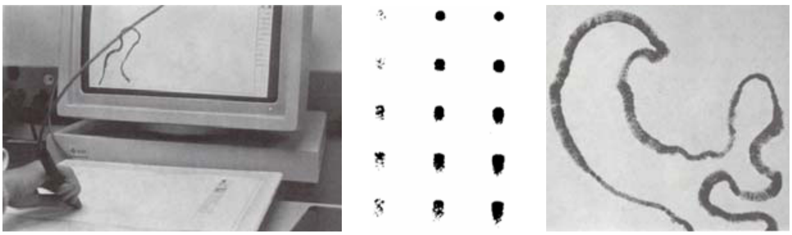
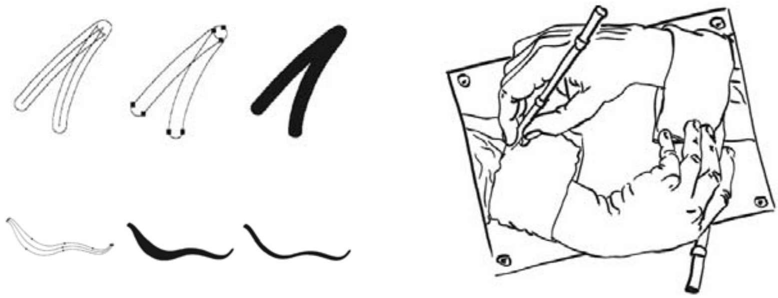


Fig. 2.3 Tilt stylus in use with dynamically brushed effects, varying with the magnitude and direction of tilt, the user pressure, etc. [Bleser *et al.,* 1988]. Copyright of ACM, used with permission

Fig. 2.4 An example of a charcoal drawing [Bleser *et al.,* 1988]. Copyright of ACM, used with permission

In the vector brushstroke approach, the stroke outline is computed from the brush outline and trajectory. Vector brushstrokes share two significant advantages of vector graphics: resolution-independence and editing capabilities. Strokes can be created, scaled, rotated or flipped very easily. Interactive retouching operations such as re-computing the stroke from the same trajectory using a different brush or different pressure data are straightforward. Moreover, the outline of the stroke is accessible to the user and can be edited like any other shape outline. However, vector brushstrokes have two major limitations. They often need heavier computations for curve fitting, as analytically solving the equations of the outline is too slow in the range of accessible shapes. Another limitation is that they sometimes require the user to explicitly enter some mathematical parameters, and thus are not suitable for sketching. Fig. 2.5 shows a set of typical vector brushstrokes and some instances of hand-sketching.

Furthermore, Finkelstein and Salesin proposed a multi-resolution vector brushstroke based on wavelets [Finkelstein & Salesin, 1994]. It requires no extra storage, and can support continuous levels of smoothing as well as direct manipulation of an arbitrary portion of the curve. Moreover, the control points and discrete nature of the underlying hierarchical representation are hidden from the user, which are preferred by hand-sketching. Fig. 2.6 shows some examples of multi-resolution brushstrokes.



(a) (b)

Fig. 2.5 Hand sketching with vector brushstrokes [Pudet, 1994]. (a) Vector brushstroke examples; (b) Hand drawing with vector brushstroke. Copyright of Blackwell Publishers, used with permission

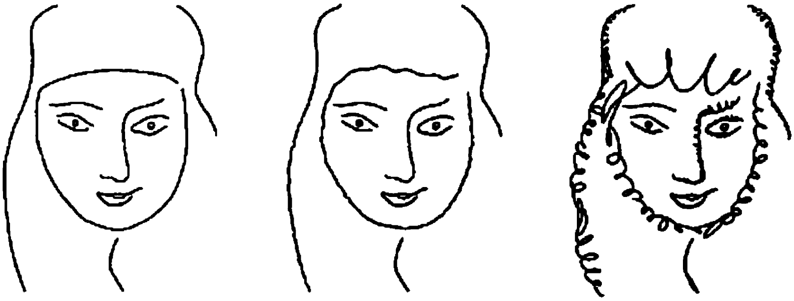


Fig. 2.6 Line-drawings with multi-resolution brushstrokes [Finkelstein & Salesin, 1994]. Copyright of ACM, used with permission

2.1.2 Pen-and-ink Illustration by Stroke Textures

Brushstrokes are to emulate traditional artists' tools, and the resulting artwork is composed of the strokes which are individually drawn by the user. However, pen-and-ink illustration incorporates a wealth of textures, tones, and styles formed by thousands of individual monochromatic strokes of the pen. The creation of pen-and-ink illustration will require a great deal of technical skills and patience. In order to remove the burden of placing individual strokes from the user, Salisbury *et al.* proposed to emulate pen-and-ink illustration via stroke textures [Salisbury *et al.*, 1994].

Stroke textures refer to collections of strokes arranged in different patterns. In pen-and-ink illustration, its tone and texture are not independent parameters, as every stroke contributes both tone (darkness) and texture. Furthermore, pen and ink strokes work together to express tone and texture. The pen artist must take care to convey both of these qualities simultaneously. Therefore Salisbury *et al.* set up the stroke texture in a pen-and-ink illustration as a collective result of many pen strokes, and a typical representation of pen and ink strokes is given below [Salisbury *et al.*, 1994]:

(1) *Pixels:* An arbitrary-size array of (x, y) pixel coordinate pairs, and x and y never change by more than ± 1 from one entry to the next.
(2) *Length:* The size of the pixels array.
(3) *Width:* The width of the stroke, in pixels.
(4) *Bbox:* The rectangular bounding box of the stroke's pixels.
(5) *Id:* The texture from which the stroke was derived.
(6) *Priority*: The ranking of a stroke, if in a prioritized texture.

The Salisbury's interactive pen-and-ink illustration system supports a library of user-defined stored stroke textures and built-in procedural stroke textures (as shown in Fig. 2.7). A stored texture is simply a collection of strokes. Drawing a texture at a given darkness is a matter of choosing from the collection a subset that has enough strokes to reach the desired tone. Procedural stroke textures are computed procedurally to depict interesting texture effects such as stippling (randomly distributed points or short strokes), parallel hatching, and curved strokes. To draw procedural stroke textures, the system simply generates appropriate candidate strokes under the region of the brush and tests them.

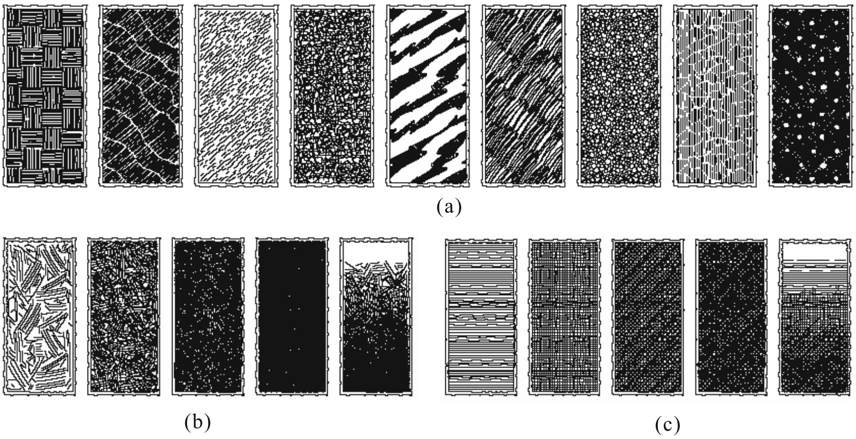


Fig. 2.7 Instances of stroke textures [Salisbury *et al.*, 1994]. (a) Assorted stored stroke textures; (b) A single texture drawn with several tone values; (c) The prioritized textures: the most significant strokes are drawn for light tone values, less important strokes are brought in to darken the texture. Copyright of ACM, used with permission

The overall paint pipeline in this system is given as follows:

Paint:
For Each brush position P
    While S←GenerateCandidateStroke(P)
        ClipStroke(S)
        If TestStrokeTone(S) **then**
            DrawStroke(S)
        End If
    End While
End For

GenerateCandidateStroke(P): At each brush position P, the system may in general try to draw many strokes. Each invocation of *GenerateCandidateStroke* returns the next stroke instance from a set of candidates. The next stroke returned may be generated dynamically based on the success of the previous strokes.

ClipStroke(S): The candidate stroke S is subjected to a series of clipping conditions such as to the bounds of the overall image, to the brush, to clipedges, etc. The clipping operations return a “first” and a “last” index into the stroke's pixels array, but before actually trimming the stroke, these indices are perturbed up or down by a small random amount to achieve ragged clipping.

TestStrokeTone(S): Two tests are performed to see how stroke S affects the image. First, the stroke's pixels in the image buffer are tested. If all the pixels are already drawn, the stroke has no effect on the image and is trivially rejected. Next, the effect of the stroke on the image tone is determined. The stroke is temporarily drawn into the image bitmap and the resulting tone is computed pixel-by-pixel along its length, by low-pass filtering each pixel's neighborhood. The stroke fails if it makes the image tone darker than the desired tone anywhere along its length.

DrawStroke(S): To draw stroke S, its pixels in the image bitmap are set, the display is updated, and an instance of S is added to the main stroke database. For stored stroke textures, the system checks to see if the new stroke S overlays an existing instance of the same stroke—such an occurrence could happen, for example, if the earlier stroke was clipped to the brush and the user has now moved the brush slightly. Rather than adding the new stroke, the previously-drawn stroke is extended to include the new stroke's pixels in order to avoid overwhelming the data structures.

Compared to the tedious individual stroke drawing, pen-and-ink simulation by stroke textures goes beyond emulating the traditional artists' tools. This enables the higher-level cumulative effect that the strokes can achieve: texture, tone, and shape. The user “paints” with a desired stroke texture to achieve a desired tone, and the computer draws all of the individual strokes. Fig. 2.8 shows some resulting pen-and-ink illustrations based on stroke textures.

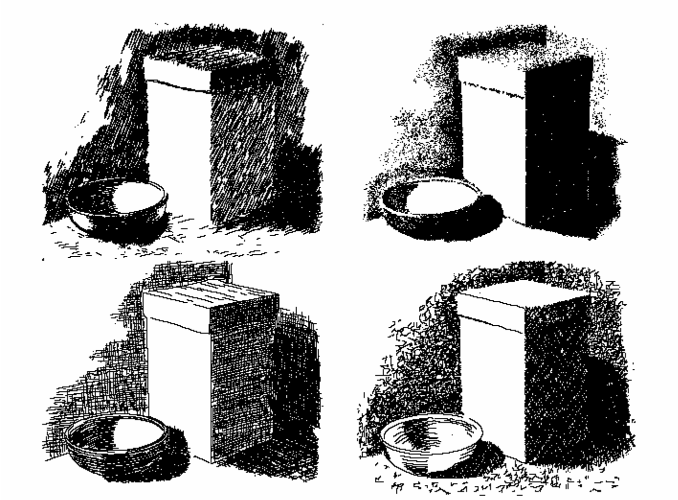


Fig. 2.8 Interactive pen-and-ink illustrations [Salisbury *et al.*, 1994]. Copyright of ACM, used with permission

2.1.3 Interactive Pencil Drawing

Pencil drawing is a flexible medium, providing a variety of styles of line quality, hand gestures, and tone building. It is excellent for preparatory sketches, and is popularly used in the contexts of scientific and technical illustrations, architectural and design drawings. From the point of view of media simulation, pen-and-ink illustrations are relatively simple, as we merely take into consideration the attributes of a pen-and-ink stroke such as width and contact area. But pencil drawings are much more complicated, as we should additionally consider the modeling of the pencil, drawing papers and the interactions between them. There are two types of pencils: graphite and colored. Thus we will discuss them respectively in the following subsections.

2.1.3.1 Graphite Pencil Drawing

The representative work on graphite pencil drawing comes from Sousa and Buchanan[Sousa & Buchanan, 2000]. They took the observational approach to simulating pencil drawing, capturing the essential physical properties and behaviors observed to produce quality pencil marks at interactive rates. Its