CAMBRIDGE

Cambridge University Press
978-0-521-76687-6 - Computer Vision for Visual Effects
Richard J. Radke
Excerpt
More information

1 Introduction

> 43 of the top 50 films of all time are visual effects driven. Today, visual effects are the "movie stars" of studio tent-pole pictures — that is, visual effects make contemporary movies box office hits in the same way that big name actors ensured the success of films in the past. It is very difficult to imagine a modern feature film or TV program without visual effects.
>
> The Visual Effects Society, 2011

Neo fends off dozens of Agent Smith clones in a city park. Kevin Flynn confronts a thirty-years-younger avatar of himself in the Grid. Captain America's sidekick rolls under a speeding truck in the nick of time to plant a bomb. Nightcrawler "bamfs" in and out of rooms, leaving behind a puff of smoke. James Bond skydives at high speed out of a burning airplane. Harry Potter grapples with Nagini in a ramshackle cottage. Robert Neville stalks a deer in an overgrown, abandoned Times Square. Autobots and Decepticons battle it out in the streets of Chicago. Today's blockbuster movies so seamlessly introduce impossible characters and action into real-world settings that it's easy for the audience to suspend its disbelief. These compelling action scenes are made possible by modern visual effects.

Visual effects, the manipulation and fusion of live and synthetic images, have been a part of moviemaking since the first short films were made in the 1900s. For example, beginning in the 1920s, fantastic sets and environments were created using huge, detailed paintings on panes of glass placed between the camera and the actors. Miniature buildings or monsters were combined with footage of live actors using forced perspective to create photo-realistic composites. Superheroes flew across the screen using rear-projection and blue-screen replacement technology.

These days, almost all visual effects involve the manipulation of digital and computer-generated images instead of in-camera, practical effects. Filmgoers over the past forty years have experienced the transition from the mostly analog effects of movies like *The Empire Strikes Back* to the early days of computer-generated imagery in movies like *Terminator 2: Judgment Day* to the almost entirely digital effects of movies like *Avatar*. While they're often associated with action and science fiction movies, visual effects are now so common that they're imperceptibly incorporated into virtually all TV series and movies — even medical shows like *Grey's Anatomy* and period dramas like *Changeling*.

Like all forms of creative expression, visual effects have both an artistic side and a technological side. On the artistic side are visual effects artists: extremely talented (and often underappreciated) professionals who expertly manipulate software packages to create scenes that support a director's vision. They're attuned to the film-making aspects of a shot such as its composition, lighting, and mood. In the middle are the creators of the software packages: artistically minded engineers at companies like The Foundry, Autodesk, and Adobe who create tools like *Nuke*, *Maya*, and *After Effects* that the artists use every day. On the technological side are researchers, mostly in academia, who conceive, prototype, and publish new algorithms, some of which eventually get incorporated into the software packages. Many of these algorithms are from the field of computer vision, the main subject of this book.

Computer vision broadly involves the research and development of algorithms for automatically understanding images. For example, we may want to design an algorithm to automatically outline people in a photograph, a job that's easy for a human but that can be very difficult for a computer. In the past forty years, computer vision has made great advances. Today, consumer digital cameras can automatically identify whether all the people in an image are facing forward and smiling, and smartphone camera apps can read bar codes, translate images of street signs and menus, and identify tourist landmarks. Computer vision also plays a major role in image analysis problems in medical, surveillance, and defense applications. However, the application in which the average person most frequently comes into contact with the results of computer vision — whether he or she knows it or not — is the generation of visual effects in film and television production.

To understand the types of computer vision problems that are "under the hood" of the software packages that visual effects artists commonly use, let's consider a scene of a human actor fighting a computer-generated creature (for example, Rick O'Connell vs. Imhotep, Jack Sparrow vs. Davy Jones, or Kate Austen vs. The Smoke Monster). First, the hero actor is filmed on a partially built set interacting with a stunt performer who plays the role of the enemy. The built set must be digitally extended to a larger environment, with props and furniture added and removed after the fact. The computer-generated enemy's actions may be created with the help of the motion-captured performance of a second stunt performer in a separate location. Next, the on-set stunt performer is removed from the scene and replaced by the digital character. This process requires several steps: the background pixels behind the stunt performer need to be recreated, the camera's motion needs to be estimated so that the digital character appears in the right place, and parts of the real actor's body need to appropriately pass in front of and behind the digital character as they fight. Finally, the fight sequence may be artificially slowed down or sped up for dramatic effect. All of the elements in the final shot must seamlessly blend so they appear to "live" in the same frame, without any noticeable visual artifacts. This book describes many of the algorithms critical for each of these steps and the principles behind them.

1.1 COMPUTER VISION FOR VISUAL EFFECTS

This book, *Computer Vision for Visual Effects*, explores the technological side of visual effects, and has several goals:

Cambridge University Press
978-0-521-76687-6 - Computer Vision for Visual Effects
Richard J. Radke
Excerpt
More information

- To mathematically describe a large set of computer vision principles and algorithms that underlie the tools used on a daily basis by visual effects artists.
- To collect and organize many exciting recent developments in computer vision research related to visual effects. Most of these algorithms have only appeared in academic conference and journal papers.
- To connect and contrast traditional computer vision research with the real-world terminology, practice, and constraints of modern visual effects.
- To provide a compact and unified reference for a university-level course on this material.

This book is aimed at early-career graduate students and advanced, motivated undergraduate students who have a background in electrical or computer engineering, computer science, or applied mathematics. Engineers and developers of visual effects software will also find the book useful as a reference on algorithms, an introduction to academic computer vision research, and a source of ideas for future tools and features. This book is meant to be a comprehensive resource for both the front-end artists and back-end researchers who share a common passion for visual effects.

This book goes into the details of many algorithms that form the basis of commercial visual effects software. For example, to create the fight scene we just described, we need to estimate the 3D location and orientation of a camera as it moves through a scene. This used to be a laborious process solved mostly through trial and error by an expert visual effects artist. However, such problems can now be solved quickly, almost automatically, using visual effects software tools like *boujou*, which build upon structure from motion algorithms developed over many years by the computer vision community.

On the other hand, this book also discusses many very recent algorithms that aren't yet commonplace in visual effects production. An algorithm may start out as a university graduate student's idea that takes months to conceive and prototype. If the algorithm is promising, its description and a few preliminary results are published in the proceedings of an academic conference. If the results gain the attention of a commercial software developer, the algorithm may eventually be incorporated into a new plug-in or menu option in a software package used regularly by an artist in a visual effects studio. The time it takes for the whole process — from initial basic research to common use in industry — can be long.

Part of the problem is that it's difficult for real-world practitioners to identify which academic research is useful. Thousands of new computer vision papers are published each year, and academic jargon often doesn't correspond to the vocabulary used to describe problems in the visual effects industry. This book ties these worlds together, "separating the wheat from the chaff" and clarifying the research keywords relevant to important visual effects problems. Our guiding approach is to describe the theoretical principles underlying a visual effects problem and the logical steps to its solution, independent of any particular software package.

This book discusses several more advanced, forward-looking algorithms that aren't currently feasible for movie-scale visual effects production. However, computers are constantly getting more powerful, enabling algorithms that were entirely impractical a few years ago to run at interactive rates on modern workstations.

Finally, while this book uses Hollywood movies as its motivation, not every visual effects practitioner is working on a blockbuster film with a looming release date and a rigid production pipeline. It's easier than ever for regular people to acquire and manipulate their own high-quality digital images and video. For example, an amateur filmmaker can now buy a simple green screen kit for a few hundred dollars, download free programs for image manipulation (e.g., GIMP or IrfanView) and numerical computation (e.g., Python or Octave), and use the algorithms described in this book to create compelling effects at home on a desktop computer.

1.2 THIS BOOK'S ORGANIZATION

Each chapter in this book covers a major topic in visual effects. In many cases, we can deal with a video sequence as a series of "flat" 2D images, without reference to the three-dimensional environment that produced them. However, some problems require a more precise knowledge of where the elements in an image are located in a 3D environment. The book begins with the topics for which 2D image processing is sufficient, and moves to topics that require 3D understanding.

We begin with the pervasive problem of **image matting** — that is, the separation of a foreground element from its background (Chapter 2). The background could be a blue or green screen, or it could be a real-world natural scene, which makes the problem much harder. A visual effects artist may semiautomatically extract the foreground from an image sequence using an algorithm for combining its color channels, or the artist may have to manually outline the foreground element frame by frame. In either case, we need to produce an **alpha matte** for the foreground element that indicates the amount of transparency in challenging regions containing wisps of hair or motion blur.

Next, we discuss many problems involving **image compositing and editing**, which refer to the manipulation of a single image or the combination of multiple images (Chapter 3). In almost every frame of a movie, elements from several different sources need to be merged seamlessly into the same final shot. Wires and rigging that support stunt performers must be removed without leaving perceptible artifacts. Removing a very large object may require the visual effects artist to create complex, realistic texture that was never observed by any camera, but that moves undetectably along with the real background. The aspect ratio or size of an image may also need to be changed for some shots (for example, to view a wide-aspect ratio film on an HDTV or mobile device).

We then turn our attention to the detection, description, and matching of **image features**, which visual effects artists use to associate the same point in different views of a scene (Chapter 4). These features are usually corners or blobs of different sizes. Our strategy for reliably finding and describing features depends on whether the images are closely separated in space and time (such as adjacent frames of video spaced a fraction of a second apart) or widely separated (such as "witness" cameras that observe a set from different perspectives). Visual effects artists on a movie set also commonly insert artificial markers into the environment that can be easily recognized in post-production.

We next describe the estimation of **dense correspondence** between a pair of images, and the applications of this correspondence (Chapter 5). In general, this problem is called **optical flow** and is used in visual effects for retiming shots and creating interesting image transitions. When two cameras simultaneously film the same scene from slightly different perspectives, such as for a live-action 3D movie, the correspondence problem is called **stereo**. Once the dense correspondence is estimated for a pair of images, it can be used for visual effects including video matching, image morphing, and view synthesis.

The second part of the book moves into three dimensions, a necessity for realistically merging computer-generated imagery with live-action plates. We describe the problem of camera tracking or **matchmoving**, the estimation of the location and orientation of a moving camera from the image sequence it produces (Chapter 6). We also discuss the problems of estimating the lens distortion of a camera, calibrating a camera with respect to known 3D geometry, and calibrating a stereo rig for 3D filming.

Next, we discuss the acquisition and processing of **motion capture** data, which is increasingly used in films and video games to help in the realistic animation of computer-generated characters (Chapter 7). We discuss technology for capturing full-body and facial motion capture data, as well as algorithms for cleaning up and post-processing the motion capture marker trajectories. We also overview more recent, purely vision-based techniques for markerless motion capture.

Finally, we overview the main methods for the direct acquisition of **three-dimensional data** (Chapter 8). Visual effects personnel routinely scan the 3D geometry of filming locations to be able to properly insert 3D computer-generated elements afterward, and also scan in actors' bodies and movie props to create convincing digital doubles. We describe laser range-finding technology such as LiDAR for large-scale 3D acquisition, structured-light techniques for closer-range scanning, and more recent multi-view stereo techniques. We also discuss key algorithms for dealing with 3D data, including feature detection, scan registration, and multi-scan fusion.

Of course, there are many exciting technologies behind the generation of computer-generated imagery for visual effects applications not discussed in this book. A short list of interesting topics includes the photorealistic generation of water, fire, fur, and cloth; the physically accurate (or visually convincing) simulation of how objects crumble or break; and the modeling, animation, and rendering of entirely computer-generated characters. However, these are all topics better characterized as **computer graphics** than computer vision, in the sense that computer vision always starts from real images or video of the natural world, while computer graphics can be created entirely without reference to real-world imagery.

Each chapter includes a short **Industry Perspectives** section containing interviews with experts from top Hollywood visual effects companies including Digital Domain, Rhythm & Hues, LOOK Effects, and Gentle Giant Studios. These sections relate the chapter topics to real-world practice, and illuminate which techniques are commonplace and which are rare in the visual effects industry. These interviews should make interesting reading for academic researchers who don't know much about filmmaking.

Each chapter also includes several homework problems. The goal of each problem is to verify understanding of a basic concept, to understand and apply a formula, or to fill in a derivation skipped in the main text. Most of these problems involve simple linear algebra and calculus as a means to exercise these important muscles in the service of a real computer vision scenario. Often, the derivations, or at least a start on them, are found in one of the papers referenced in the chapter. On the other hand, this book doesn't have any problems like "implement algorithm X," although it should be easy for an instructor to specify programming assignments based on the material in the main text. The emphasis here is on thoroughly understanding the underlying mathematics, from which writing good code should (hopefully) follow.

As a companion to the book, the website `cvfxbook.com` will be continually updated with links and commentary on new visual effects algorithms from academia and industry, examples from behind the scenes of television and films, and demo reels from visual effects artists and companies.

1.3 BACKGROUND AND PREREQUISITES

This book assumes the reader has a basic understanding of linear algebra, such as setting up a system of equations as a matrix-vector product and solving systems of overdetermined equations using linear least-squares. These key concepts occur repeatedly throughout the book. Less frequently, we refer to the eigenvalues and eigenvectors of a square matrix, the singular value decomposition, and matrix properties like positive definiteness. Strang's classic book [469] is an excellent linear algebra reference.

We also make extensive use of vector calculus, such as forming a Taylor series and taking the partial derivatives of a function with respect to a vector of parameters and setting them equal to zero to obtain an optimum. We occasionally mention continuous partial differential equations, most of the time en route to a specific discrete approximation. We also use basic concepts from probability and statistics such as mean, covariance, and Bayes' rule.

Finally, the reader should have working knowledge of standard image processing concepts such as viewing images as grids of pixels, computing image gradients, creating filters for edge detection, and finding the boundary of a binary set of pixels.

On the other hand, this book doesn't assume a lot of prior knowledge about computer vision. In fact, visual effects applications form a great backdrop for learning about computer vision for the first time. The book introduces computer vision concepts and algorithms naturally as needed. The appendixes include details on the implementation of several algorithms common to many visual effects problems, including dynamic programming, graph-cut optimization, belief propagation, and numerical optimization. Most of the time, the sketches of the algorithms should enable the reader to create a working prototype. However, not every nitty-gritty implementation detail is provided, so many references are given to the original research papers.

1.4 ACKNOWLEDGMENTS

I wrote most of this book during the 2010-11 academic year while on sabbatical from the Department of Electrical, Computer, and Systems Engineering at Rensselaer Polytechnic Institute. Thanks to Kim Boyer, David Rosowsky, and Robert Palazzo for their support. Thanks to my graduate students at the time — Eric Ameres, Siqi Chen, David Doria, Linda Rivera, and Ziyan Wu — for putting up with an out-of-the-office advisor for a year.

Many thanks to the visual effects artists and practitioners who generously shared their time and expertise with me during my trip to Los Angeles in June 2011. At LOOK Effects, Michael Capton, Christian Cardona, Jenny Foster, David Geoghegan, Buddy Gheen, Daniel Molina, and Gabriel Sanchez. At Rhythm & Hues, Shish Aikat, Peter Huang, and Marty Ryan. At Cinesite, Shankar Chatterjee. At Digital Domain, Nick Apostoloff, Thad Beier, Paul Lambert, Rich Marsh, Som Shankar, Blake Sloan, and Geoff Wedig. In particular, thanks to Doug Roble at Digital Domain for taking so much time to discuss his experiences and structure my visit. Special thanks to Pam Hogarth at LOOK Effects and Tim Enstice at Digital Domain for organizing my trip. Extra special thanks to Steve Chapman at Gentle Giant Studios for his hospitality during my visit, detailed comments on Chapter 8, and many behind-the-scenes images of 3D scanning.

This book contains many behind-the-scenes images from movies, which wouldn't have been possible without the cooperation and permission of several people. Thanks to Andy Bandit at Twentieth Century Fox, Eduardo Casals and Shirley Manusiwa at adidas International Marketing, Steve Chapman at Gentle Giant Studios, Erika Denton at Marvel Studios, Tim Enstice at Digital Domain, Alexandre Lafortune at Oblique FX, Roni Lubliner at NBC/Universal, Larry McCallister and Ashelyn Valdez at Paramount Pictures, Regan Pederson at Summit Entertainment, Don Shay at Cinefex, and Howard Schwartz at Muhammad Ali Enterprises. Thanks also to Laila Ali, Muhammad Ali, Russell Crowe, Jake Gyllenhaal, Tom Hiddleston, Ken Jeong, Darren Kendrick, Shia LaBeouf, Isabel Lucas, Michelle Monaghan, and Andy Serkis for approving the use of their likenesses.

At RPI, thanks to Jon Matthis for his time and assistance with my trip to the motion capture studio, and to Noah Schnapp for his character rig. Many thanks to the students in my fall 2011 class "Computer Vision for Visual Effects" for commenting on the manuscript, finding errors, and doing all of the homework problems: Nimit Dhulekar, David Doria, Tian Gao, Rana Hanocka, Camilo Jimenez Cruz, Daniel Kruse, Russell Lenahan, Yang Li, Harish Raviprakash, Jason Rock, Chandroutie Sankar, Evan Sullivan, and Ziyan Wu.

Thanks to Lauren Cowles, David Jou, and Joshua Penney at Cambridge University Press and Bindu Vinod at Newgen Publishing and Data Services for their support and assistance over the course of this book's conception and publication. Thanks to Alice Soloway for designing the book cover.

Special thanks to Aaron Hertzmann for many years of friendship and advice, detailed comments on the manuscript, and for kindling my interest in this area. Thanks also to Bristol-Myers Squibb for developing Excedrin, without which this book would not have been possible.

During the course of writing this book, I have enjoyed interactions with Sterling Archer, Pierre Chang, Phil Dunphy, Lester Freamon, Tony Harrison, Abed Nadir, Kim Pine, Amelia Pond, Tim Riggins, Ron Swanson, and Malcolm Tucker.

Thanks to my parents for instilling in me interests in both language and engineering (but also an unhealthy perfectionism). Above all, thanks to Sibel, my partner in science, for her constant support, patience, and love over the year and a half that this book took over my life and all the flat surfaces in our house. This book is dedicated to her.

RJR, March 2012

Cambridge University Press
978-0-521-76687-6 - Computer Vision for Visual Effects
Richard J. Radke
Excerpt
More information

2 Image Matting

Separating a foreground element of an image from its background for later compositing into a new scene is one of the most basic and common tasks in visual effects production. This problem is typically called **matting** or **pulling a matte** when applied to film, or **keying** when applied to video.[1] At its humblest level, local news stations insert weather maps behind meteorologists who are in fact standing in front of a green screen. At its most difficult, an actor with curly or wispy hair filmed in a complex real-world environment may need to be digitally removed from every frame of a long sequence.

Image matting is probably the oldest visual effects problem in filmmaking, and the search for a reliable automatic matting system has been ongoing since the early 1900s [393]. In fact, the main goal of Lucasfilm's original Computer Division (part of which later spun off to become Pixar) was to create a general-purpose image processing computer that natively understood mattes and facilitated complex compositing [375]. A major research milestone was a family of effective techniques for matting against a blue background developed in the Hollywood effects industry throughout the 1960s and 1970s. Such techniques have matured to the point that blue- and green-screen matting is involved in almost every mass-market TV show or movie, even hospital shows and period dramas.

On the other hand, putting an actor in front of a green screen to achieve an effect isn't always practical or compelling, and situations abound in which the foreground must be separated from the background in a natural image. For example, movie credits are often inserted into real scenes so that actors and foreground objects seem to pass in front of them, a combination of image matting, compositing, and matchmoving. The computer vision and computer graphics communities have only recently proposed methods for semi-automatic matting with complex foregrounds and real-world backgrounds. This chapter focuses mainly on these kinds of algorithms for still-image matting, which are still not a major part of the commercial visual effects pipeline since effectively applying them to video is difficult. Unfortunately, video matting today requires a large amount of human intervention. Entire teams of rotoscoping artists at visual effects companies still require hours of tedious work to produce the high-quality mattes used in modern movies.

[1] The computer vision and graphics communities typically refer to the problem as matting, even though the input is always digital video.

We begin by introducing matting terminology and the basic mathematical problem (Section 2.1). We then give a brief introduction to the theory and practice of blue-screen, green-screen, and difference matting, all commonly used in the effects industry today (Section 2.2). The remaining sections introduce different approaches to the **natural image matting** problem where a special background isn't required. In particular, we discuss the major innovations of Bayesian matting (Section 2.3), closed-form matting (Section 2.4), Markov Random Fields for matting (Section 2.5), random-walk matting (Section 2.6), and Poisson matting (Section 2.7). While high-quality mattes need to have soft edges, we discuss how image segmentation algorithms that produce a hard edge can be "softened" to give a matte (Section 2.8). Finally, we discuss the key issue of matting for video sequences, a very difficult problem (Section 2.9).

2.1 MATTING TERMINOLOGY

Throughout this book, we assume that a color image I is represented by a 3D discrete array of pixels, where $I(x,y)$ is a 3-vector of (red, green, blue) values, usually in the range $[0,1]$. The matting problem is to separate a given color image I into a **foreground** image F and a **background** image B. Our fundamental assumption is that the three images are related by the **matting** (or **compositing**) **equation**:

$$I(x,y) = \alpha(x,y)F(x,y) + (1-\alpha(x,y))B(x,y) \tag{2.1}$$

where $\alpha(x,y)$ is a number in $[0,1]$. That is, the color at (x,y) in I is a mix between the colors at the same position in F and B, where $\alpha(x,y)$ specifies the relative proportion of foreground versus background. If $\alpha(x,y)$ is close to 0, the pixel gets almost all of its color from the background, while if $\alpha(x,y)$ is close to 1, the pixel gets almost all of its color from the foreground. Figure 2.1 illustrates the idea. We frequently abbreviate Equation (2.1) to

$$I = \alpha F + (1-\alpha)B \tag{2.2}$$

with the understanding that all the variables depend on the pixel location (x,y). Since α is a function of (x,y), we can think of it like a grayscale image, which is often called a **matte**, **alpha matte**, or **alpha channel**. Therefore, in the matting problem, we are given the image I and want to obtain the images F, B, and α.

At first, it may seem like $\alpha(x,y)$ should always be either 0 (that is, the pixel is entirely background) or 1 (that is, the pixel is entirely foreground). However, this isn't the case for real images, especially around the edges of foreground objects. The main reason is that the color of a pixel in a digital image comes from the total light intensity falling on a finite area of a sensor; that is, each pixel contains contributions from many real-world optical rays. In lower resolution images, it's likely that some scene elements project to regions smaller than a pixel on the image sensor. Therefore, the sensor area receives some light rays from the foreground object and some from the background. Even high resolution digital images (i.e., ones in which a pixel corresponds to a very small sensor area) contain fractional combinations of foreground and background in regions like wisps of hair. Fractional values of α are also generated by motion of the camera or foreground object, focal blur induced by the camera aperture, or