[redacted]

DISSERTATION

Interactive Image Matting

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

ao. Univ.-Prof. Mag. Dipl.-Ing. Dr. Margrit Gelautz
Institut 188 für Softwaretechnologie und Interaktive Systeme

und

Dr. Carsten Rother
Microsoft Research Cambridge

eingereicht an der Technischen Universität Wien
bei der Fakultät für Informatik

von

Mag. DI. Christoph Rhemann
Matrikelnummer: 9925853
Baumeistergasse 6/49/1
1160 Wien

Wien, im Jänner 2010

Summary

Image matting aims to extract a foreground object from a single natural image by recovering the partial transparency and corresponding color of the foreground object at each pixel in the image. The resulting transparency map is thereby denoted as alpha matte. The matting problem is severely ill-posed, and in this thesis we focus on matting approaches that utilize user interaction to make the problem tractable.

There are three fundamental challenges in interactive image matting research that are addressed in this thesis: (i) Providing a fast and intuitive user interface; (ii) finding a good cost function for matting; and (iii) providing a benchmark that allows a quantitative comparison of matting results.

In most previous approaches the user interacts with the algorithm by drawing an accurate trimap, which is a partition of the image into foreground, background and unknown regions. An accurate trimap is very tedious to create manually, hence we follow recent work and aim to automatically generate a trimap from very little user input. The novelty of our approach lies in a new cost function that describes the goodness of a trimap solution. Our cost function considers several image cues and incorporates four different types of priors that are used to regularize the result. We show that our method is fast and produces accurate results.

Given a trimap, the thesis then addresses the problem of extracting an alpha matte from a single photograph. We improve on previous image matting approaches by assuming that the majority of partial transparencies are induced by the imaging process. Hence we exploit a model where alpha is the convolution of a binary segmentation with the camera's point spread function. Based on this model, we propose new matting algorithms that generate high-quality results even for images where our assumption is not met completely.

Finally, we introduce a new benchmark test for image matting that enables a quantitative comparison of matting results. Our contributions are (i) a challenging, high-quality ground truth test set that builds the basis of our evaluation; (ii) a dynamic online benchmark system that allows other researchers to interactively analyze recent matting work and to complement the evaluation with new results; and (iii) perceptually motivated error metrics for image matting. We use this benchmark to confirm that our proposed matting algorithms outperform the current state-of-the-art.

Kurzfassung

Das Ziel von Image Matting ist es, ein Vordergrundobjekt aus einem Bild herauszulösen. Dabei müssen der Transparenzwert und die Farbe des Vordergrundobjektes an jedem Pixel im Bild bestimmt werden. Die resultierende Transparenzkarte wird auch als Alpha Matte bezeichnet. Das Matting-Problem ist mathematisch unterbestimmt, weshalb die meisten Algorithmen auf Benutzerinteraktion angewiesen sind, um das Problem einzuschränken.

Es gibt drei große Herausforderungen im Bereich Matting: (i) Die Entwicklung einer einfach und schnell zu bedienenden Benutzerschnittstelle; (ii) die Modellierung einer geeigneten Kostenfunktion, welche die Güte einer Alpha Matte beschreibt, und (iii) die Erstellung eines Benchmark-Tests, der einen quantitativen Vergleich von Matting-Algorithmen ermöglicht. In dieser Arbeit werden neue Ansätze in allen drei Bereichen präsentiert.

In den meisten vorangegangenen Arbeiten interagiert der Benutzer mit dem Algorithmus, indem er eine sogenannte Trimap zeichnet. Die Trimap ist eine Unterteilung des Bildes in Vordergrund, Hintergrund und einen unbekannten Bereich. Das händische Erstellen einer genauen Trimap ist jedoch sehr zeitintensiv. Daher folgen wir jüngsten Forschungsarbeiten und berechnen eine exakte Trimap anhand weniger Benutzereingaben. Die Neuheit in unserem Ansatz ist eine verbesserte Kostenfunktion, welche die Güte einer Trimap beschreibt. Unsere vorgeschlagene Kostenfunktion basiert auf Bildmerkmalen und a-priori Wissen über den Bildgebungsprozess, wodurch es ermöglicht wird, präzise Resultate mit wenig Benutzeraufwand zu erzeugen.

Der zweite Teil dieser Arbeit beschäftigt sich mit dem Extrahieren einer Alpha Matte unter Zuhilfenahme einer vom Benutzer spezifizierten Trimap. Unser Ansatz basiert auf der Annahme, dass die Transparenzen des Vordergrundobjektes vor allem durch den Bildgebungsprozess entstanden sind. Aufgrund dieser Annahme verwenden wir ein Modell, welches die Alpha Matte als Faltung einer binären Segmentierung mit der Punktspreizfunktion der Kamera beschreibt. Basierend auf diesem Modell werden in dieser Arbeit Matting-Algorithmen vorgestellt, welche qualitativ hochwertige Resultate erzeugen können, selbst wenn unsere Annahme nicht zur Gänze erfüllt wird.

Im letzten Teil dieser Arbeit entwickeln wir einen Benchmark-Test für Image Matting, der einen quantitativen Vergleich der Algorithmen ermöglicht. Der von uns entwickelte Benchmark umfasst (i) einen Testdatensatz mit qualitativ hochwertigen Referenzlösungen; (ii) ein dynamisches Online Benchmark System, welches für Forscher zugänglich ist, um bestehende Algorithmen zu analysieren und die Evaluierung mit neuen Resultaten zu ergänzen; und (iii) Fehlermetriken für Image Matting, welche auf der menschlichen Wahrnehmung basieren. Die Resultate des Benchmark-Tests bestätigen die ausgezeichnete Leistung unserer Matting-Algorithmen im Vergleich zum State of the Art.

Acknowledgements

First, I would like to thank my PhD advisors Margrit Gelautz and Carsten Rother, for their great support over the years. Margrit has supported me since my master studies and I am grateful that she nominated me for the Microsoft PhD scholarship. Without her help I would not have been able to start my PhD studies. I would like to thank Carsten for his great enthusiasm and for the countless hours that he devoted to this PhD project. He constantly supported and inspired this work with brilliant ideas. Without his help this thesis would look very different.

Many thanks go to my collaborators from different research institutions. In particular I would like to thank Michael Bleyer, Pushmeet Kohli, Alex Rav-Acha, Dheeraj Singaraju, Victor Lempitsky and Toby Sharp. Especially, I would like to thank Michael for many hours of useful discussions and Alex for sharing ideas that helped me to get started with my thesis.

I would also like to thank Microsoft Research Cambridge for the financial support through their PhD Scholarship Program and several travel grants. In particular, I would like to thank Fabien Petitcolas for initiating the collaboration with Carsten as well as for many interesting lectures, fun and excellent food at the scholarship events. I would also like to thank Christian Breiteneder for recommending me for the Microsoft PhD scholarship and for supporting my research visits abroad. Further thanks go to the academic relations manager of Microsoft Austria, Andreas Schabus, for supporting my scholarship application.

Last on this page but first in my heart, I thank my family, especially Mom, Dad, my grandparents and Judith, for their unconditional support and for making my life great everyday.

Contents

Summary **i**

Kurzfassung **iii**

Acknowledgements **v**

1 Introduction **1**
- 1.1 Motivation 1
- 1.2 Image Compositing and Matting 3
  - 1.2.1 Matting Using Specialized Imaging Setups 4
  - 1.2.2 Natural Image Matting 5
- 1.3 Contributions 5
  - 1.3.1 User Interaction 6
  - 1.3.2 Objective Function 7
  - 1.3.3 Matting Evaluation 8
- 1.4 Resulting Publications 9
- 1.5 Thesis Outline 10

2 Related Work **11**
- 2.1 User Interaction 11
  - 2.1.1 Trimap Interface 11
  - 2.1.2 Scribble Interface 13
  - 2.1.3 Scribble-based Trimap Extraction 13
- 2.2 Objective Function 17

2.2.1 Propagation-based Approaches 17
2.2.2 Color-model-based Approaches 24
2.3 Matting Evaluation 29
2.3.1 Ground Truth Data 29
2.3.2 Online Evaluation Repository 30
2.3.3 Perceptually Motivated Error Functions 30

3 Alpha Formation Process **31**
3.1 Translucent Materials 32
3.2 Imaging Process 34
3.2.1 Motion Blur 34
3.2.2 Defocus Blur 36
3.2.3 Resolution Blur 39
3.3 Segmentation-based Model for Alpha 40

4 A New Trimap Extraction Method **45**
4.1 Overview 48
4.2 Model 49
4.2.1 Color 50
4.2.2 Class Prior 52
4.2.3 Sub-blur Kernel Structures 52
4.2.4 Sharp Boundary 54
4.2.5 Smoothness Prior 55
4.3 Parameter Training 56
4.4 Experimental Results and Evaluation 57
4.4.1 Test and Training Data 57
4.4.2 Comparison of Trimap Extraction Methods 59
4.5 Summary 61

5 Improved Color Modeling **66**
5.1 Overview 66
5.2 Collecting Candidate Samples 68

5.3 Selecting Best Candidate Samples 70
5.4 Experimental Results 73
5.5 Summary 74

6 Segmentation-based Prior for Matting 77
6.1 Deconvolution of Alpha Mattes - Related Work 78
6.2 Hybrid Deconvolution Approach 80
    6.2.1 Overview of our Matting Approach 81
    6.2.2 Estimating the Initial Alpha Matte 81
    6.2.3 Estimating the PSF 82
    6.2.4 Alpha Deblurring 82
    6.2.5 Binarization of the Deblurred Alpha Matte 84
    6.2.6 Re-estimating Alpha Using the Segmentation Prior 85
    6.2.7 Multi-resolution Estimation of the Matte 85
    6.2.8 Experimental Results 85
6.3 Segmentation-based Deconvolution 89
    6.3.1 Overview of our Matting Approach 91
    6.3.2 Estimating the Initial Alpha Matte 91
    6.3.3 Upsampling Alpha 91
    6.3.4 Estimating the Binary Segmentation from Alpha 91
    6.3.5 Estimating the PSF 95
    6.3.6 Re-estimating Alpha Using the Segmentation Prior 96
    6.3.7 Experimental Results 96
6.4 Model Analysis 99
6.5 Summary 102

7 An Evaluation System for Image Matting 104
7.1 Ground Truth Database 106
    7.1.1 Data Capture 106
    7.1.2 Image Properties 108
    7.1.3 User Input 108
7.2 Online Benchmark System 109

7.3 Perceptually Motivated Error Measures 111
7.3.1 User Study 113
7.3.2 Analysis of Results 115
7.3.3 Choice of Parameters 118
7.4 Evaluation Results 120
7.4.1 Performance on SAD and MSE 120
7.4.2 Performance on Gradient Error 124
7.4.3 Performance on Connectivity Error 125
7.5 Summary 126

8 Conclusions and Future Work 128
8.1 Conclusions 128
8.2 Future Research Topics 129

Bibliography 131

List of Tables

4.1 Quantitative comparison of trimap methods. 60

6.1 Model errors. 102

7.1 Error measure correlations. 118
7.2 Evaluation. 122

List of Figures

1.1 Why do we need matting? 2
1.2 Image compositing. 3

2.1 User input. 12
2.2 Trimap painting with soft scissors. 13
2.3 Scribble-based trimap extraction. 15
2.4 Matting components. 22
2.5 Collecting color samples in Bayesian Matting. 25
2.6 Collecting color samples in Robust Matting. 27

3.1 Matting ambiguities. 32
3.2 Interaction of light with matter. 33
3.3 Fractional alpha values originating from motion blur. 35
3.4 Relation of defocus to image blur. 36
3.5 Fractional alpha values originating from defocus blur. 38
3.6 Fractional alpha values originating from resolution blur. 40
3.7 Alpha model. 42
3.8 Sparse distribution of alpha. 43

4.1 Our interactive matting framework. 46
4.2 Comparison of scribble-based matting approaches. 47
4.3 Unary terms for trimap segmentation. 51
4.4 Sub-blur kernel structures. 53
4.5 Sharp boundary terms. 55

4.6 Visualization of the correlation between λ and the size of the $\mathcal{U}$ region. . . . 56
4.7 Training and test images. 58
4.8 Comparison of trimap methods (1). 62
4.9 Comparison of trimap methods (2). 63
4.10 Comparison of trimap methods (3). 64
4.11 Comparison of trimap methods (4). 65

5.1 Overview of our approach. 67
5.2 Collecting color samples. 69
5.3 Sample selection. 72
5.4 Qualitative comparison (1). 74
5.5 Qualitative comparison (2). 75
5.6 Qualitative comparison (3). 75

6.1 Deconvolution with binary prior. 80
6.2 Intermediate results of our matting approach. 83
6.3 Comparison of matting methods (1). 87
6.4 Comparison of matting methods (2). 88
6.5 Deconvolution with binary prior (more results). 90
6.6 Enforcing connectivity. 94
6.7 Comparison of matting methods (1). 97
6.8 Comparison of matting methods (2). 98
6.9 Limitation - large light-transmitting region. 100
6.10 Spatially varying PSF. 101

7.1 Online benchmark. 110
7.2 Motivation for perceptual error measures. 112
7.3 A test case used in our study. 115
7.4 Connectivity error. 118
7.5 Robustness of parameters. 120
7.6 Performance of Segmentation-based Deconvolution method. 124
7.7 Performance on gradient error. 125

7.8 Advantages of the connectivity prior. 127

Acronyms

COC	Circle Of Confusion
DSLR	Digital Single Lens Reflex
GMM	Gaussian Mixture Model
GT	Ground Truth
MAD	Mean Absolute Distance
MAP	Maximum A Posteriori
MRF	Markov Random Field
MSE	Mean Squared Error
PSF	Point Spread Function
QPBO	Quadratic Pseudo Boolean Optimization
RGB	Red Green Blue
SAD	Sum of Absolute Differences

Chapter 1

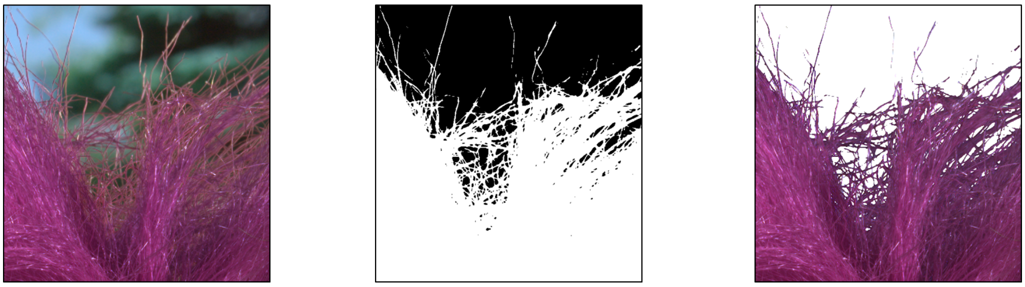
Introduction

1.1 Motivation

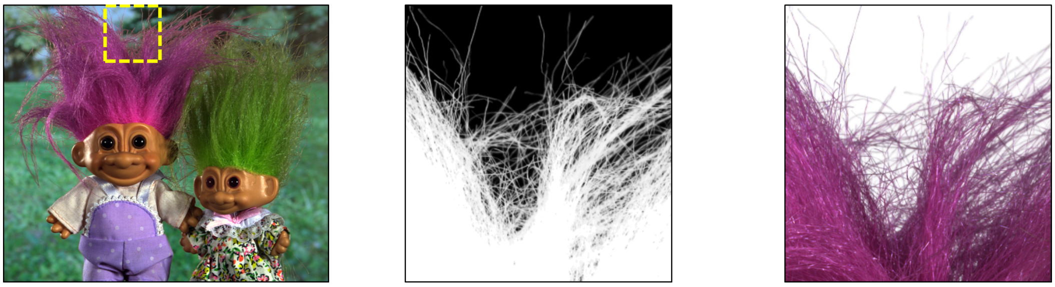
Separating a foreground object in an image from its background is a fundamentally important operation in image editing, with many applications in the entertainment industry. For instance, once an object has been separated from its background, it may be blended with another background scene.

To separate the foreground object, *binary segmentation techniques* like [BJ01, MB95] may be applied to the image. Such algorithms assign each pixel in the image to either the fore- or the background. The result is a binary mask, which defines the extent of the foreground object. For instance, the mask of the foreground of the image crop in figure 1.1(a) is depicted in figure 1.1(b). In the depicted mask, pixels which belong to the foreground are encoded in white, whereas background pixels are shown in black. Using this binary mask we blended the foreground of the image crop in figure 1.1(a) with a plain white background. The resulting image composition in figure 1.1(c) is imperfect, since the fine hair strands do not look visually integrated with the white background. This is because the colors at the boundary of the foreground object in figure 1.1(a) were mixed with the colors of the background during the image acquisition. Such mixed pixels cannot be clearly assigned to either the fore- or background, thus cannot be separated with a pure binary segmentation.

Hence, to achieve a more accurate separation of the foreground object, one has to infer the partial coverage of the foreground at mixed pixels. This task is known as *alpha matting*



(a) Crop of (d) (b) Binary foreground mask (c) Composite on white using (b)



(d) Image (e) Alpha matte (f) Composite on white using (e)

Figure 1.1: **Why do we need matting?** (a) The crop of the image in (d) shows purple hair strands in front of a blue and green background. Pixels close to the foreground object boundary in (a) were blended to the background during the image acquisition, hence only partially belong to the foreground. Using a binary mask (b) to blend the foreground in (a) with a white background gives a visually unpleasing image composition (c). An alpha matte (e) can account for this fractional foreground coverage, resulting in a visually integrated image composition (f).

and the resulting "soft segmentation" is referred to as the alpha matte. An example of an alpha matte is shown in figure 1.1(e) for the image crop in figure 1.1(a). The gray values of the alpha matte encode the fractional coverage of the foreground. Pixels which fully belong to the fore- and background are encoded in white and black, respectively. Using the alpha matte we can seamlessly blend the foreground object over an e.g. plain white background as demonstrated in figure 1.1(f).

In the following we will explain the image matting and compositing task in more detail.

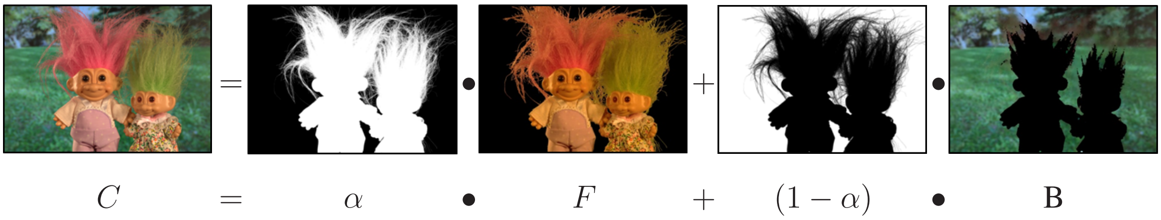


Figure 1.2: **Image compositing.** The foreground F is blended to the background B, according to the alpha matte α to give the composite C. Matting aims to reconstruct α, F and B, given C as input. See the text for a more detailed discussion.

1.2 Image Compositing and Matting

As we have seen in the previous section, the goal of image compositing is to combine two or more images from (different) sources such that the resulting image looks visually integrated [Bri99, Wri06]. One of the most useful operations in image compositing is the *over operation* [PD84], which aims to seamlessly place a foreground object over a background. The over operation is formalized in the *compositing equation* (see figure 1.2 for an illustration), which models the composite image C as a convex combination of the foreground color F and background color B:

$$C = \alpha F + (1 - \alpha)B. \tag{1.1}$$

Here, the *alpha matte* α defines the ratio to which the foreground object covers the background at each pixel. In regions where the foreground object fully covers the background, the value of alpha will be 1 (shown in white in figure 1.2), and 0 in those regions where the foreground object does not cover the background at all (shown in black in figure 1.2). However, close to the object boundaries, the foreground may only partially cover the background (an example is the hair of the soft toy in figure 1.2). Hence, the fore- and background colors are mixed together, which is modeled by fractional alpha values (i.e. $0 < \alpha < 1$). In general, this partial foreground coverage (i.e. mixing of the fore- and background colors) can be caused by multiple factors, such as translucent materials or the imaging process itself. We will discuss these sources in detail in chapter 3.

Compositing an image C using eq. (1.1) is a straightforward process. In sharp contrast, *alpha matting* is the inverse process of compositing and attempts to extract the foreground color F, background color B and alpha matte α, given only the observed color C of the composite (input) image. Assuming that we are working in the RGB-color space, C, F and B are three-dimensional vectors, with each dimension representing one color channel. Given only the composite image C, this leaves us with seven unknowns in only three equations. Clearly, this is a severely ill-posed problem, and without any further constraints there is an infinite number of solutions to the problem. For instance, one undesired solution that perfectly fits the compositing equation (1.1) is to set the alpha matte α to 1 and the foreground color F to the color of the input image C at each pixel in the image. Hence, the extracted foreground image equals to the input composite.

Thus, to solve the matting problem, further constraints are necessary. In section 1.2.1 we will discuss algorithms that constrain the problem by using a specialized imaging setup. However, such approaches cannot be used to derive alpha mattes for natural images that were captured with a standard photo-camera. In this thesis we aim to infer alpha mattes from such natural images, which in general requires the user to manually place constraints on the image (see section 1.2.2 for details).

1.2.1 Matting Using Specialized Imaging Setups

To solve for the unknown variables, some matting approaches impose additional constraints on the image setup. For instance, one approach that is extensively used in the film and entertainment industry is *Blue Screen Matting*. Here, the problem is simplified by photographing or filming the foreground object in front a known (usually constant-colored blue or green) background. Hence, the background color B in eq. (1.1) is known. Still the problem remains under-constrained (4 unknowns in 3 equations) and the remaining ambiguities are solved in practice by imposing ad-hoc assumptions on the foreground color channels [SB96].

In [SB96] an extension to this approach has been presented, which allows to obtain the true solution to the matting problem. This *Triangulation Matting* approach works by photographing the foreground object against at least two backgrounds, which differ in color at each pixel. This yields an overdetermined set of linear equations (i.e. 6 equations with only

4 unknown variables) that can be solved using a linear least squares method. Although the strict studio requirements of this approach prevent it from being used as a general-purpose matting system, it is very useful to derive reference solutions to the matting problem. In particular, we will use this technique to create a ground truth dataset for the purpose of evaluating matting algorithms in chapter 7.

Other previously proposed matting approaches have used additional information in the form of flash/no-flash image pairs [SLKS06], multiple synchronized video streams with different focus settings [MMP$^+$05], camera arrays [JMA06, WFZ02] or stereo cameras [BGRR09].

1.2.2 Natural Image Matting

The matting approaches described in section 1.2.1 can generate accurate results, but their practical use is limited since they either rely on specialized imaging setups or require the image to be captured in restrictive studio environments. In this thesis we will consider a practically more interesting, but in general more difficult case, where we aim to infer all unknown parameters in eq. (1.1) from a single natural image. In contrast to approaches that rely on a specialized imaging acquisition, *natural image matting* approaches rely on input by the user to restrict the space of possible solutions. Generally speaking, this is done by indicating those parts in the image where the fore- and background can be easily distinguished by the user. We will review different kinds of user interaction in chapter 2.1. Even after the user has placed constraints on the image, the matting problem remains ill-posed. Hence, to infer the unknown variables for the unconstrained pixels, further assumptions are necessary. One common assumption used in matting is local regularity on F and B, and we will review related work in chapter 2.2.

1.3 Contributions

There are three fundamental challenges in natural image matting research: (1) To provide a good way of user interaction; (2) to find a good objective function for matting; and (3) to evaluate the matting results. This thesis makes contributions in all three areas.

Most of our contributions rest upon the insight that a majority of mixed pixels (i.e. pixels where $\alpha \in \,]0,1[$) are caused by the imaging process. During the imaging process, the foreground can be mixed with its background due to defocus blur, motion blur or discretization. These blurring effects can be described by the camera's Point Spread Function (PSF). Based on this fundamental observation, we advocate a model for alpha that had previously been studied with respect to the super-resolution and deblurring tasks. In particular, we model the alpha matte as the convolution of a binary segmentation with the camera's PSF that accounts for the fractional alpha values. (See chapter 3 for details.) In this thesis we exploit this alpha model, based on a segmentation and the PSF, to overcome weaknesses of previously proposed matting approaches.

In the following, the contributions of this thesis are discussed in more detail.

1.3.1 User Interaction

As we have seen in section 1.2, natural image matting is a severely ill-posed problem and therefore user interaction is vital to solve it. The most common form of user interaction is the trimap interface, where the user manually partitions the image into foreground, background and unknown regions (see e.g. [CCSS01, RT00, WC07a, WAC07, GSAW05]). The matting problem is then solved for the pixels in the unknown regions only. If the unknown region of the trimap is very small, the resulting matte is usually of very high-quality. However, drawing an accurate trimap is a tedious process, and therefore matting algorithms have been developed that are also capable of working on very sparse trimaps, commonly denoted as scribbles (see e.g. [WC05, GCL$^+$06, LLW08, LRAL08]). These scribbles can also be regarded as a trimap with a large unknown region. Although scribbles are easier to create, the quality of the matting results is usually inferior to those obtained with a more accurate trimap. Recently, an intermediate solution has been suggested by Juan and Keriven [JK05] that aims to automatically generate an accurate trimap from sparse scribble input.

In this thesis we propose a novel method to extract an accurate trimap from only a few user defined scribbles in chapter 4. This is done by formulating the task of trimap segmentation as an energy minimization problem. The main contribution lies in a novel energy function that considers several image cues and incorporates four different types of priors. These priors are directly motivated from the segmentation-based alpha model (see above) and are used to regularize the result. We learn the parameters of the energy function from training data and optimize the energy with the parametric maxflow technique [KBR07]. This allows the user to interactively modify the size of the unknown trimap region in real-time, once the optimization procedure is finished. After the user has adjusted the size of the unknown trimap region, the matting algorithm of [WC07a] is invoked that computes the final alpha matte. Our method is intuitive and fast, and we show that it outperforms previous trimap extraction approaches.

1.3.2 Objective Function

Once the user has specified a trimap, the goal is to infer the alpha matte α and corresponding fore- and background colors (F, B) for the unknown trimap region. In this thesis we derive α by minimizing a cost function, subject to the user-defined constraints. One integral part of this cost function is to accurately model the distributions of the unknown model parameters (i.e. α, F and B).

To model the distributions of the fore- and background colors, several methods have been proposed in the past, e.g. [RT00, CCSS01, WC07a]. In this thesis we introduce a novel color modeling approach, which considerably improves the performance of matting algorithms. Previous work, like [WC07a], models the fore- and background colors at each pixel using nearby (local) estimates of F and B. The key idea of our approach is to exploit information from global color models to find better local estimates of the true fore- and background colors. In particular, we first gather a number of potential fore- and background color samples from user marked regions which are close in geodesic space (defined on the likelihood of a global color model). This is in contrast to previous approaches, like [WC07a], which simply collect samples from spatially nearby regions. In the next step we compute a confidence value for the sampled fore- and background colors. We present a new paradigm to compute the confidence by assuming that most alpha values in the image are either exactly 0 or 1 (this is a property of the segmentation-based alpha model). Finally, alpha values are computed from the color samples with the highest confidence. Details of this approach are given in chapter 5.

The second major challenge is to model alpha. Previous matting approaches either do not apply any prior on alpha or assume smoothness of the alpha matte. However, a smoothness prior is oftentimes not sufficient to overcome ambiguities in the solution space. The key idea is now to improve on previous matting approaches by modeling the matte as a convolution of a binary segmentation with a PSF.

We then propose two novel algorithms that infer alpha using this segmentation-based model. The first algorithm starts by computing an initial approximation of alpha based on a matting algorithm that uses a standard smoothness prior. From this initial alpha matte we infer the PSF and the binary segmentation using a novel Markov Random Field (MRF)-based segmentation technique. Afterwards we blur the binary segmentation with the PSF and use it to re-estimate the alpha matte. We show that this approach improves on the current state-of-the-art on a dataset of real matting scenes with known ground truth.

However, there are some drawbacks of this approach, which we aim to overcome in the second algorithm. First, we observed that the binary segmentation of thin structures is oftentimes overestimated by the first algorithm (i.e. the binary segmentation of thin structures is too wide). This is presumably because the segmentation was computed in a resolution where the underlying segmentation of thin structures is not yet binary. Therefore, in our second algorithm we propose to work on the higher-resolution (upscaled) alpha matte, where the underlying binary segmentation is more likely to be binary. Secondly, our improved method estimates the binary segmentation directly from the alpha matte as opposed to the first algorithm, where computationally expensive deconvolution methods were applied to alpha before binarization. Thirdly, we apply a different segmentation procedure, which enforces connectivity of the binary segmentation.

1.3.3 Matting Evaluation

An integral part of research is to evaluate the goodness of a proposed method on a standard benchmark. However, ground truth data for low-level vision benchmarks such as matting or stereo can only be obtained with a lot of effort. As a consequence, no benchmark has been so far developed for the task of image matting. In this thesis we present a new benchmark test that comprises three important contributions. First, we introduce a challenging, high-quality ground truth test set that builds the basis of our benchmark. Second, we establish a dynamic online benchmark system that provides all data and scripts, which enables researchers to interactively analyze recent matting work and to complement the evaluation with new results. Finally, we improve on the evaluation methodology for image matting by proposing perceptually motivated error functions.

We use our benchmark to evaluate the quality of our new matting algorithms, presented in this thesis. The benchmark confirms that our proposed algorithms perform favorably compared to the state-of-the-art. Also, our challenging test set reveals problems of existing algorithms that were not reflected in previously reported results.

1.4 Resulting Publications

Major parts of this thesis resulted in the following articles:

- C. Rhemann, C. Rother, P. Kohli, V. Lempitsky and M. Gelautz. Segmentation-based Alpha Matting. Under review.
- C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott. A Perceptually Motivated Online Benchmark for Image Matting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1826-1833, 2009.
- C. Rhemann, C. Rother, A. Rav-Acha, and T. Sharp. High Resolution Matting via Interactive Trimap Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1-8, 2008.
- C. Rhemann, C. Rother, and M. Gelautz. Improving Color Modeling for Alpha Matting. In *British Machine Vision Conference*, 2:1155-1164, 2008.

This thesis also inspired the following publications, whose contents are, however, not included in this thesis:

- C. Rhemann, M. Gelautz, and B. Fölsner. An Evaluation of Interactive Image Matting Techniques Supported by Eye-Tracking. In *SPIE Electronic Imaging*, volume 7242, 2009.

- M. Bleyer, M. Gelautz, C. Rother and C. Rhemann. A Stereo Approach that Handles the Matting Problem via Image Warping. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 501-508, 2009.
- D. Singaraju, C. Rother and C. Rhemann. New Appearance Models for Image Matting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 659-666, 2009.

1.5 Thesis Outline

This thesis is organized into 8 chapters. In chapter 2 we discuss work related to user interaction, the objective function and the evaluation of matting algorithms. In chapter 3 we investigate the alpha formation process and use the insights to develop a physically motivated model for alpha. Based on our physical model, we introduce a new approach to user interaction in chapter 4 and improve on the objective function for matting in chapters 5 and 6. In chapter 7 we present a new ground truth dataset and evaluation methodology for image matting and use it to quantitatively compare our matting approaches to the state-of-the-art. Finally, chapter 8 summarizes our contributions and highlights directions for future research.

Chapter 2

Related Work

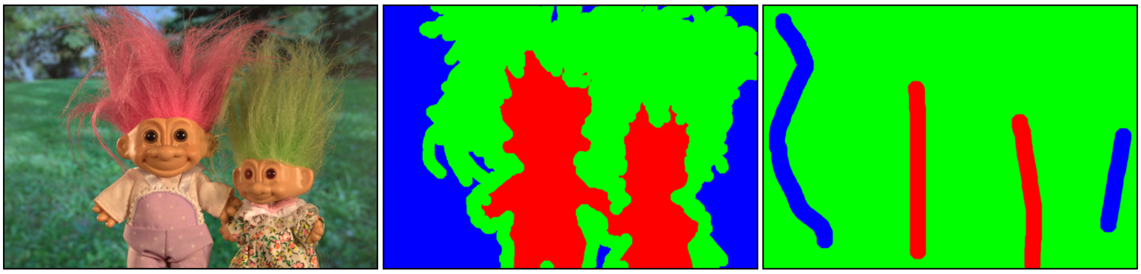
There are three main areas of related work, namely user interaction, the objective function and the evaluation of matting algorithms. In section 2.1, we review previous user interaction techniques. Recovering the alpha matte from the user-defined constraints according to an objective function, is discussed in section 2.2. Finally, in section 2.3, we deal with work related to the evaluation of matting methods.

2.1 User Interaction

As discussed in chapter 1, natural image matting is an ill-posed problem. Without any further constraints there is an infinite number of solutions to the problem. Hence, user interaction is vital to make the problem tractable. The most common forms of user interaction used in previous work can be broadly classified into three categories, namely *trimaps*, *scribbles* and *scribble-based trimap extraction* methods. In the remainder of this section, we will review all three of them.

2.1.1 Trimap Interface

The first class of interfaces that we discuss is based on trimaps (e.g. [CCSS01, RT00, WC07a, WAC07, GSAW05]). With the trimap interface the user manually assigns, as accurately as possible, each pixel to one of three classes: foreground ($\mathcal{F}$), background ($\mathcal{B}$)



(a) Input image (b) Trimap input (c) Scribble input

Figure 2.1: **User input.** For the input image (a) we show an example trimap (b) and scribble input (c). For both types of user input, the user manually marks each pixel as either foreground (shown in red), background (shown in blue) or unknown (shown in green).

or unknown ($\mathcal{U}$). An example trimap for the input image in figure 2.1(a) is shown in 2.1(b). The information from the known regions ($\mathcal{F}$, $\mathcal{B}$) is then used to predict for each unknown pixel the values for F, B and α. We call a trimap "perfectly tight" if the α values in $\mathcal{U}$ are above 0 and below 1 and the $\mathcal{F}$ and $\mathcal{B}$ regions comprise only α values which are exactly 1 and 0, respectively. It has been shown [WC07a] that, if the trimap is perfect (or nearly perfect), the resulting matte is of very high-quality. However, manually drawing such an accurate trimap is a very tedious and time-consuming process for the user.

To shift some of the burden from the user to the system, more sophisticated trimap "paint tools" like the soft scissors approach [WAC07] have been proposed. It builds on the intelligent scissors method [MB95], which computes a hard segmentation, i.e. $\alpha \in \{0, 1\}$. In soft scissors, the user marks the unknown region of the trimap by tracing the object boundary with a wide brush as illustrated in figure 2.2. The brush size is adapted according to the underlying data and intermediate results of the matte are shown, enhancing the user experience. The main drawback of such a brush tool is that objects with a long boundary or complicated boundary topology are very tedious to trace, e.g. a tree with many foreground holes.

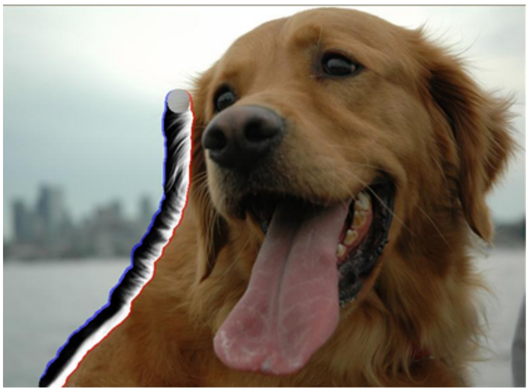


Figure 2.2: **Trimap painting with soft scissors.** With the soft scissors approach, the user creates a trimap by tracing the object boundary with a brush that automatically adapts to the underlying data. Figure from [WAC07].

2.1.2 Scribble Interface

Mainly because tracing the boundary can be very time-consuming, the trend of hard segmentation approaches has been to move from *boundary* selection tools like intelligent scissors [MB95] to scribble-based *region* selection tools [BJ01, RKB04]. This second class of interfaces is more user-friendly since only relatively few pixels have to be assigned to the foreground or background, which can be far away from the object boundary. Figure 2.1(c) shows a scribble input example for the image in figure 2.1(a). With scribble-based input, impressive results were achieved for hard segmentation [BJ01, RKB04, BS07] and also to some extent for matting [LLW08, WC05, LRAL08, GCL$^+$06]. However, in general the results obtained with an accurate trimap are qualitatively superior to those obtained with sparse scribbles. Hence, for difficult examples an accurate trimap is vital to obtain good results.

2.1.3 Scribble-based Trimap Extraction

To combine the accuracy of trimaps with the usability of scribbles, one can attempt to automatically generate an accurate trimap from sparse scribble input. A straightforward solution proposed in e.g. [RKB04, BS07] is to first obtain a binary segmentation of the image into $\mathcal{F}$ and $\mathcal{B}$ using some interactive image segmentation technique, e.g. GrabCut

[RKB04]. Figure 2.3(b) shows the segmentation obtained by [RKB04] using the scribbles in figure 2.3(a). In a second step, the unknown region is defined as a narrow band of uniform thickness, obtained by dilation of the segmentation boundary (see figure 2.3(c)).

Clearly, such an approach will fail for images where the foreground object has a complicated boundary. An example is shown in figure 2.3(d), where a band of uniform thickness (figure 2.3(e)) cannot cover all mixed pixels. Hence, a method is necessary which computes an adaptive band that respects the underlying data (see figure 2.3(f) for an example).

Recently, such a method has been suggested by Juan and Keriven [JK05]. In contrast to binary segmentation algorithms (e.g. GrabCut [RKB04]), the method in [JK05] segments the image into three classes (i.e. $\mathcal{F}$, $\mathcal{B}$ and $\mathcal{U}$) by minimizing the energy

$$E(\mathbf{x}) = \sum_i D(x_i) + \sum_{(i,j)\in\mathcal{N}} V(x_i, x_j), \tag{2.1}$$

where $x_i \in \{\mathcal{F}, \mathcal{B}, \mathcal{U}\}$ denotes the label of pixel i and $\mathcal{N}$ is the set of neighboring pixels. (For simplicity, sets and labels have the same name, e.g. $\mathcal{F}$). The vector $\mathbf{x}$ encodes the assignment of all pixels in the image. The term data D for pixel x_i is modeled as

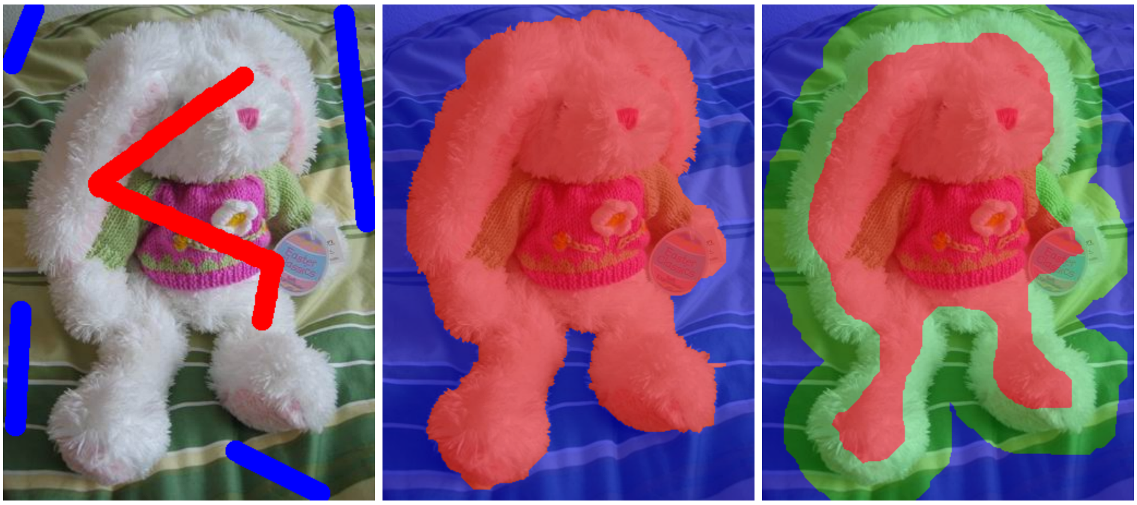
$$D(x_i) = \begin{cases} -logP(c_i|\theta_{\mathcal{F}}) & \text{if } x_i \in \mathcal{F} \\ -logP(c_i|\theta_{\mathcal{B}}) & \text{if } x_i \in \mathcal{B}. \end{cases}$$

Here $\theta_{\mathcal{F}}$ and $\theta_{\mathcal{B}}$ are the Gaussian Mixture Models (GMM) of the fore- and background, respectively. In the unknown region the color distribution of $D(x_i)$, with $x_i \in \mathcal{U}$, is represented by a third GMM $\theta_{\mathcal{U}}$ that is constructed by blending all combinations of fore- and background mixtures of the respective GMMs with the mixture coefficient α. In [JK05], it is assumed that α follows a uniform distribution, hence the blending coefficient is assumed to be linear.

The smoothness term V is defined as

$$V(x_i, x_j) = \lambda \frac{\delta(x_i = x_j)}{1 + \text{dist}\,(i,j)}, \tag{2.2}$$

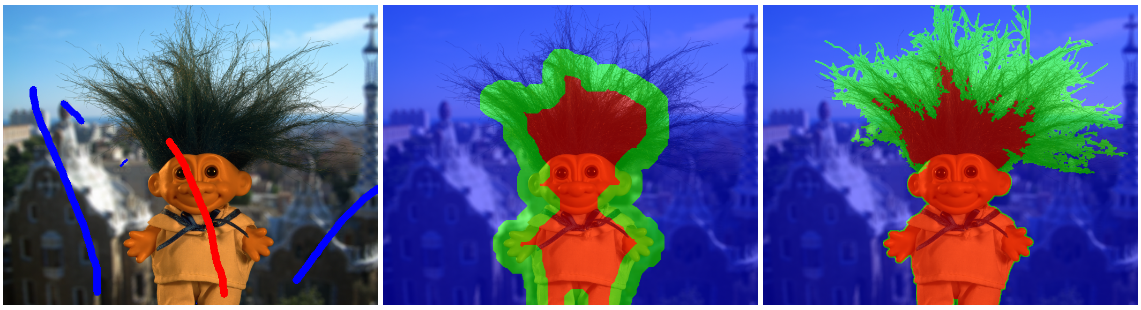
where δ is the Kronecker delta, $dist(i, j)$ is the distance between pixels i and j, and λ weights the smoothness term against the data term. The optimal labeling of the energy in



(a) Input image & scribbles (object with simple boundary)

(b) Binary segmentation using scribbles from (a)

(c) Trimap by dilation of the segmentation boundary from (b)



(d) Input image & scribbles (object with complex boundary)

(e) Trimap from eroded binary segmentation using scribbles from (d)

(f) Trimap extracted using the scribbles from (d)

Figure 2.3: **Scribble-based trimap extraction.** The image in (a) shows a soft toy with a rather simple object outline. In (b) we show the binary segmentation obtained with GrabCut [RKB04] using the user scribbles in (a). Dilation of the segmentation boundary gives a trimap where the unknown region is a band of uniform thickness (c). The trimap in (e) shows that a uniform band cannot capture the complex boundary (i.e. hair) of the object in (d). In contrast, trimap-extraction techniques infer a very accurate trimap (f) from only a few user scribbles (d).

eq. (2.1) can be obtained by computing a single minimum cut in a special graph [Ish03].

Motivated by [JK05] we present an improved trimap segmentation approach in chapter 4. A result of our method is depicted in figure 2.3(f), where we can see that the band of the unknown region adapts nicely to the complex object boundary.

2.2 Objective Function

As we have seen in the previous section, most natural image matting algorithms start by having the user place some constraints on the input image. Unfortunately, even with user input the matting problem remains underconstrained. Thus most matting algorithms additionally exploit the strong correlations between nearby pixels to estimate an alpha matte. Following [WC07a], we can roughly classify previous natural image matting methods into two classes according to how they model these correlations:

Propagation-based approaches exploit the correlation between nearby pixels by modeling that either the fore- and background colors or the alpha values are smooth within a small local neighborhood.

Color-model-based approaches use the correlation of neighboring pixels to first estimate the fore- and background color at each pixel. The optimal alpha value is then estimated individually for each pixel by evaluating the compositing equation (1.1) using the estimated color values. This pixel-wise estimate of alpha forms the data term of the objective function. The probably best state-of-the-art approaches, e.g. [WC07a], combine a propagation term with a color-model based data term in a single objective function.

In the remainder of this chapter we will briefly summarize previous approaches that are most relevant for this thesis. For a more detailed review, the interested reader is referred to the recent survey by Wang et al. [WC07b].

2.2.1 Propagation-based Approaches

Some matting methods are purely "propagation-based". This means that the given fore- and background regions define the boundary conditions for the alpha matte (i.e. 1 and 0) and the alpha values are propagated into the unknown region according to an objective function.

Poisson Matting

Poisson Matting [SJTS04] is based on the assumption that the foreground colors F and background colors B are locally smooth, hence the gradient of the fore- and background is relatively small. If this assumption is met, it has been shown in [MYT95, SJTS04] that the gradient field of the alpha matte can be approximated as:

$$\nabla\alpha \approx \frac{1}{F-B}\nabla C, \tag{2.3}$$

where ∇C denotes the gradient of the input image C. Using this approximation, the final alpha matte α can be obtained by minimizing the following variational problem by solving Poisson equations subject to user constraints:

$$\alpha = \arg\min_{\alpha} \int\int_{i\in\mathcal{U}} \left\| \nabla\alpha_i - \frac{1}{F_i - B_i}\nabla C_i \right\|^2 di, \tag{2.4}$$

where the fore- and background colors F and B for each pixel i of the unknown trimap region $\mathcal{U}$ are simply obtained by extrapolation from the fore- and background regions of a user marked trimap.

Although Poisson Matting is capable of producing impressive results, the quality decreases if the foreground or background colors are highly textured or if the fore- and background colors are locally very similar.

Random Walk Matting

In [GSAW05] the alpha value at a pixel in the unknown trimap region is defined as the probability that a "random walker", starting from the pixel under consideration, will first reach the user-defined foreground constraints before reaching the background constraints. Formally, these probabilities can be computed for all pixels in the unknown trimap region by minimizing a quadratic cost function defined on a weighted graph. In the graph, each pixel corresponds to a node and neighboring pixels are connected by edges (in [GSAW05] a pixel is connected to its four neighbors in the cardinal directions). The weight of an edge between two nodes i and j is defined by an affinity function $W_{i,j}$. An affinity function defines to which extent neighboring nodes are coupled (i.e. should have the same value for alpha). In other words, the affinity $W_{i,j}$ corresponds to the probability that a random walker transitions from node i to node j. Grady et al. [GSAW05] used an affinity function under which neighboring pixels are highly coupled if they have a similar color, and loosely coupled otherwise. Such affinities that enforce smoothness of the alpha matte in homogeneously colored regions are commonly used in image segmentation (e.g. [SM00, RKB04]) and can be formalized as

$$W_{i,j} = \exp\left(\frac{\|C_i - C_j\|^2}{\sigma^2}\right), \tag{2.5}$$

where C_i and C_j are the RGB color vectors at neighboring pixels i and j, and σ is a constant. For a better discrimination of the object boundary, this affinity was modified in [GSAW05] by applying a linear transformation to the RGB colors in eq. (2.5) using Locality Preserving Projections [HN04]. Finally, the alpha matte is computed by minimizing the cost function

$$J(\alpha) = \alpha^T L \alpha, \tag{2.6}$$

where α is a column vector of length N (where N is the number of pixels in the image). The matrix L is an $N \times N$ graph Laplacian matrix given by $L = D - W$. Here, W is a symmetric matrix whose off-diagonal entries are given by eq. (2.5) (after linear transformation of the RBG colors). The diagonal matrix D is given by $D(i,i) = \sum_j W_{i,j}$. The cost in eq. (2.6) is quadratic, hence can be solved by minimizing a system of linear equations.

Closed-form Solution

The Random Walk algorithm builds on an affinity function that was originally developed for binary segmentation. However, segmentation and matting are problems of different complexity. For matting very accurate models of the matte are necessary, whereas for segmentation color-based image features are oftentimes sufficient for extracting an object from its background. The main contribution of [LLW08] was a novel affinity function that is better suited to the matting problem and which can be solved in closed form. It builds on the observation that images can be locally described by as few as only two colors [OW04]. Based on this observation, the authors of [LLW08] assume that inside a small patch the colors of the fore- and background layer can be modeled by lines in the RGB color space. Using this assumption, [LLW08] derived a new "matting affinity" as

$$W_{i,j} = \sum_{k(i,j) \in w_k} \frac{1}{|w_k|}\left(1 + (C_i - \mu_k)\left(\Sigma_k + \frac{\varepsilon}{|w_k|} I_3\right)^{-1}(C_j - \mu_k)\right). \tag{2.7}$$

Here, Σ_k is a 3×3 covariance matrix, μ_k is the mean vector of the colors in a local window w_k, centered around pixel k. C_i and C_j denote the color vectors at pixels i and j in this local window w_k and I_3 denotes the 3×3 identity matrix. The constant ε weights a regularization term that enforces local constancy of the alpha mattes. This matting affinity has been shown to perform considerably better than the affinity used in [GSAW05].

Similar to Random Walk Matting [GSAW05], the final alpha matte can be derived by minimizing the cost $J(\alpha) = \alpha^T L \alpha$. In contrast to the cost used in Random Walk Matting (see eq. (2.6)), L denotes the *Matting Laplacian*, which is given by $L = D - W$, where the entries of W are given by eq. (2.7).

The Matting Laplacian has been shown to give impressive results and is commonly used in state-of-the-art matting approaches.

New Appearance Models for Matting

The Closed-form Matting approach [LLW08], reviewed in the previous paragraph, models the color distributions of the fore- and background layers as locally linear. It was observed in [SRR09] that if the colors of the fore- or background layer are locally constant (i.e. are a point in color space), the color line model is an over-fit and the quality of the generated alpha mattes decreases. The main contribution of [SRR09] was to extend the color line appearance model of [LLW08] such that it can handle any combination of point and line color models.

Similar to [LLW08], the authors of [SRR09] derived the alpha matte in a closed-form fashion. The closed-form solution derived in [SRR09] has a bias towards alpha values of 0 (or 1). This is in contrast to the approach by [LLW08], which has a bias towards constant solutions. The bias of [SRR09] is preferable for those parts of the unknown trimap region which are solely bounded by foreground constraints (this is a common case for objects with many holes, e.g. a tree). For such trimaps, the constant bias of [LLW08] tends to oversmooth the alpha matte, whereas the approach of [SRR09] has no such problems. On the other hand a bias towards constant solutions is preferable, if the user-defined constraints are very sparse. In such situations the approach of [SRR09] tries to fit fractional alpha even though the true alpha matte might be completely opaque or transparent in large parts of the image. In general this problem can be fixed by providing more accurate user constraints.

Spectral Matting

Spectral Matting extends spectral segmentation algorithms like [SM00, Wei99, YS03] that attempt to derive a fully automatic segmentation of the image into a collection of hard segments. Spectral segmentation techniques do this by finding the smallest eigenvectors of a symmetric semidefinite graph Laplacian matrix. The Matting Laplacian L used in [LLW08] for the task of interactive (supervised) image matting is of the same form as a graph Laplacian matrix, but builds on an affinity function that is better suited for matting.

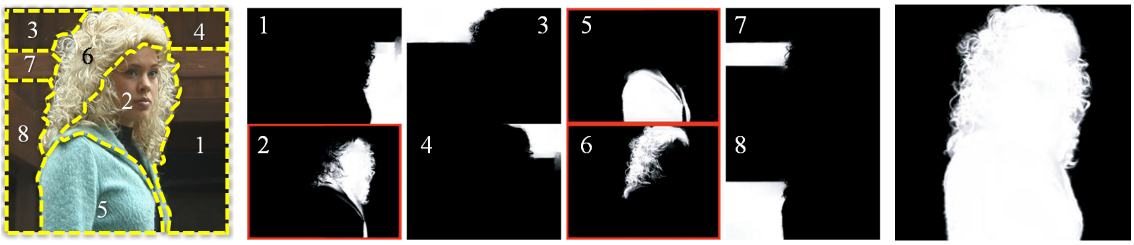
The key idea of Spectral Matting [LRAL08] is now to obtain an unsupervised segmentation of the image into K soft “matting components” α^k (in contrast to the “hard” segmentations components that would result from spectral segmentation techniques) which are spanned by the smallest eigenvectors of the Matting Laplacian. More precisely, the matting components are derived from the smallest eigenvectors of the Matting Laplacian by a linear transformation which ensures that (i) the resulting components sum up to an alpha value of 1 at each pixel; and (ii) as many pixels as possible are assigned to an alpha value of either 0 or 1. The second property of the transformation yields matting components that are mostly binary valued. This is motivated by the empirical observation that in most alpha mattes the majority of pixels is either opaque or completely transparent. This empirical observation can be manifested by the alpha formation process which we discuss in chapter 3.

Formally, given the $N \times K$ matrix $E = [e^1, \dots, e^K]$ comprising the K smallest eigenvectors of the Matting Laplacian L, the goal is to find the K transformation vectors y^k that minimize the following function over all pixels i and eigenvectors k

$$\sum_{i,k} |\alpha_i^k|^\gamma + |1 - \alpha_i^k|^\gamma \tag{2.8}$$

subject to $\sum_k \alpha_i^k = 1$, where $\alpha^k = Ey^k$. By using $0 < \gamma < 1$ (a typical value is $\gamma = 0.9$), the alpha values inside each matting component are encouraged to be distributed sparsely (i.e. as many pixels as possible are assigned to an alpha value of 0 or 1). The cost function in eq. (2.8) is a non-linear system and can be solved with re-weighted least squares.

The matting components obtained from the input image in figure 2.4(a) are depicted in figure 2.4(b). We can see that the matting components are an oversegmentation of the



(a) Input image. Dashed lines indicate components in (b)

(b) Matting components from (a)

(c) Alpha matte by grouping components in (b).

Figure 2.4: **Matting components.** Given an input image (a), the Spectral Matting approach automatically computes soft matting components (b) that are grouped to a final alpha matte (c). Figure modified from [LRAL08].

image. To obtain the desired alpha matte in figure 2.4(c), the matting components have to be grouped together in a second step. In [LRAL08] this is done in a completely automatic fashion by selecting the grouping that gives the lowest cost under the Matting Laplacian of [LLW08]. However, this automatic grouping might fail if the fore- or background object comprises several visually distinct components. Therefore, the authors of [LRAL08] propose to use an interactive scribble-based grouping approach or suggest a simple manual assignment of each matting component to the fore- or background.

Geodesic Matting

In [BS07] an alpha value for a pixel is computed based on its weighted (geodesic) distance to the user-defined fore- and background constraints. The geodesic distance D between two pixels is defined as the *shortest path* on a weighted graph with edge weight $W_{x,y}$:

$$D_{i,j} = \min_{\gamma_{i,j}} \sum_{x,y} W_{x,y}, \tag{2.9}$$

where $\gamma_{i,j}$ is a path connecting the pixels i and j, and x and y are two neighboring pixels on the path ($x, y \in \gamma_{i,j}$). The weight $W_{x,y}$ of the edge connecting two nodes x and y is defined as

$$W_{x,y} = |P_{\mathcal{F}}(x) - P_{\mathcal{F}}(y)|, \tag{2.10}$$

where $P_{\mathcal{F}}(x)$ is the likelihood that pixel x belongs to the foreground. This foreground likelihood is obtained by

$$P_{\mathcal{F}}(x) = p(C_x|\theta_{\mathcal{F}}) / \left(p(C_x|\theta_{\mathcal{F}}) + p(C_x|\theta_{\mathcal{B}})\right), \tag{2.11}$$

where $p(C_x|\theta_{\mathcal{F}})$ is the probability that the color C, at pixel x, was generated by the Gaussian Mixture Model $\theta_{\mathcal{F}}$ of the foreground, which is constructed from all pixels in the user-marked foreground region. The probability $p(C_x|\theta_{\mathcal{B}})$ is computed likewise.

Let $D_{i,\mathcal{F}}$ denote the geodesic distance of pixel i to the foreground user constraints. Then the alpha matte is computed by

$$\alpha = \frac{\omega_{\mathcal{F}}(i)}{\omega_{\mathcal{F}}(i) + \omega_{\mathcal{B}}(i)}, \tag{2.12}$$

where $\omega_{\mathcal{F}}(i) = D_{i,\mathcal{F}}^{-r} \cdot P_{\mathcal{F}}(i)$ and $\omega_{\mathcal{B}}(i)$ are computed similarly. The parameter r controls the smoothness of the matte and is typically set to $0 \leq r \leq 2$.

Fuzzy Matting

Similar to Geodesic Matting, the Fuzzy Matting approach [ZKY+08] is based on a weighted distance computation, denoted as *fuzzy connectedness*. The fuzzy connectedness FC between pixels i and j is defined as the *strongest path* on a weighted graph with edge weights $W_{x,y}$:

$$FC_{i,j} = \max_{\gamma_{i,j}} \left(\min_{x,y} W_{x,y} \right). \tag{2.13}$$

Here $\gamma_{i,j}$ denotes a path connecting the pixels i and j, and x and y are two neighboring pixels on that path ($x, y \in \gamma_{i,j}$). The "strength" of a path is defined by the weakest edge weight along the path ($\min$ in eq. (2.13)). The fuzzy connectedness FC between two pixels is the strength of the strongest path among all possible paths ($\max$ in eq. (2.13)). The $\min$/$\max$ metric in eq. (2.13) is independent of the length of the path, which is in contrast to the geodesic distance where the weights are summed up over the path (see eq.

(2.9)). In Fuzzy Matting, the edge weights (affinities) in eq. (2.13) are defined as

$$W_{i,j} = \lambda\psi_{i,j} + (1-\lambda)\phi_{i,j}, \tag{2.14}$$

where $\lambda \in [0,1]$ weights the terms ψ and ϕ. The term $\psi_{i,j}$ measures the color similarity between pixels i and j, and the function $\phi_{i,j}$ measures the color distance of pixels i and j to the colors of the user-marked fore- and background scribbles. The colors of those scribbles are thereby modeled by Gaussian Mixtures Models. Finally, [ZKY^{+}08] computes a matte as

$$\alpha = \frac{FC_{i,\mathcal{F}}}{FC_{i,\mathcal{F}} + FC_{i,\mathcal{B}}}, \tag{2.15}$$

where $FC_{i,\mathcal{F}}$ and $FC_{i,\mathcal{B}}$ denote the *fuzzy connectedness* of a pixel i to the user-marked foreground or background scribbles, respectively.

2.2.2 Color-model-based Approaches

Propagation-based approaches, presented in section 2.2.1, widely ignore the color distribution within the known fore- and background regions of the user-defined trimap. Additionally modeling these distributions can considerably improve propagation-based methods. Using the color model, for each pixel the optimal α value is then estimated *individually* and, ideally, also associated with a confidence value. The pixel-wise estimate for α forms the data term of the objective function that is oftentimes combined with a propagation term. Different approaches to color modeling have been suggested in the past and we will review the ones that are most relevant for our work in the following.

Bayesian Matting

Similar to the approach of [RT00], Bayesian Matting [CCSS01] models the local distributions of the foreground and background colors with a spatially varying set of Gaussians. More precisely, fore- and background color samples are gathered for each pixel from known trimap regions (that is, fore- and background) that lie within a radius r around the pixel under consideration (see figure 2.5 for illustration). Then a Gaussian model is fitted to the collected color samples at each pixel.

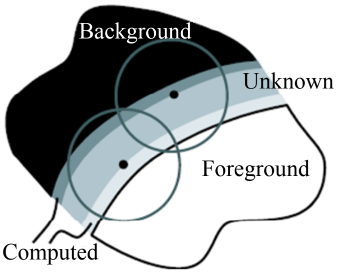


Figure 2.5: **Collecting color samples in Bayesian Matting.** Color samples are collected from known trimap regions and from previously computed colors that lie within a certain radius around a pixel. Figure modified from [CCSS01].

In order to make the color sampling more robust, the pixels are processed in an onion-peel fashion: First, the samples for pixels close to the user-marked trimap regions are estimated, hence previously computed colors are used as additional color samples for pixels which are further away from the user constraints (see figure 2.5). Finally, an alpha matte is obtained in a well defined Bayesian framework by iteratively estimating F, B and α.

Iterative Matting

In the Iterative Matting algorithm [WC05] a non-parametric color modeling approach is taken. This means that the collected color samples are directly used for alpha estimation. This is in contrast to, e.g., Bayesian Matting [CCSS01], where parametric models are fitted to the collected samples before the matte is estimated.

The sample set is obtained by selecting spatially close pixels in the user marked regions. However, these local color samples may not match the true fore- and background colors for pixels which are far away from the user-marked region (a common case when using sparse scribble input). In such cases the authors of [WC05] resort to a global color model. More precisely, they train a Gaussian Mixture Model on the fore- and background colors of the user scribbles and obtain color samples by randomly sampling each Gaussian.

Once the sample set is collected, each pair of fore- and background color samples is used to determine a likelihood for a set of k discretized alpha values (in [WC05] k is set to 25) at each pixel. This pixel-wise likelihood is combined with a pairwise smoothness term in an objective function. The alpha matte is then obtained by minimizing this objective function using Belief Propagation.

Easy Matting

In [GCL$^+$06] an "Easy Matting" approach was presented that is very similar to the Iterative Matting method of [WC05]. The main difference is that the authors of [GCL$^+$06] formulate the problem with a quadratic cost function which can be minimized by solving a set of linear equations. This formulation has the advantage that it allows to solve for a continuously valued alpha matte. Thus they can avoid the discretization of the alpha values necessary in the algorithm of [WC05]. Another difference is that in [GCL$^+$06] a dynamic weighting of the smoothness term is employed. The authors start the optimization with a large smoothness weight and decrease the weight during subsequent iterations of their algorithm. This is done in order to avoid that the algorithm gets stuck in a local minimum in early iterations.

Robust Matting

The before mentioned color modeling approaches, for instance [CCSS01], use either parametric models or take into account *all* color samples regardless of their reliability (e.g. [WC05, GCL$^+$06]). The key insight of [WC07a] is that better results can be achieved by selecting the "best" samples from the initial sample set. In the following, this approach is described in detail.

In Robust Matting [WC07a], an initial set of color samples is collected by spreading the sample set along the boundaries of the fore- and background regions of the user defined trimap. An example is illustrated in figure 2.6, which shows the foreground samples (yellow dots) and background samples (red dots) for the pixel marked in green.

In the next step the most confident samples are selected from this initial set. In [WC07a] a confident foreground/background sample pair (F^i, B^j) should meet the following two criteria: (i) F^i and B^j should fit the compositing equation (1.1); and (ii) F^i and B^j should be widely separated in color space, to allow for a robust estimation of α. In [WC07a] these two criteria are encoded in a distance ratio R:

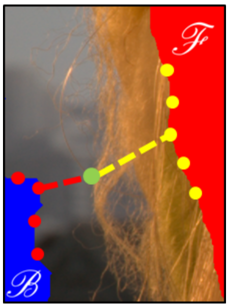


Figure 2.6: **Collecting color samples in Robust Matting.** The color samples are spread along the boundary of the user-defined fore- and background regions of the trimap.

$$R(F^i, B^j) = \frac{||C - (\widehat{\alpha} F^i + (1 - \widehat{\alpha})B^j)||}{||F^i - B^j||}, \tag{2.16}$$

where $\widehat{\alpha}$ is estimated by projecting the observed color C onto the line spanned by the sample pair (F^i, B^j) under consideration. Then a confidence value f for a sample pair is computed as

$$f(F^i, B^j) = exp\left\{ -\frac{R(F^i, B^j)^2 \cdot w(F^i) \cdot w(B^j)}{\sigma^2} \right\}, \tag{2.17}$$

where σ is a constant which was fixed to 0.1. In [WC07a] the two weights $w(F^i)$ and $w(B^j)$ are defined such that the confidence in eq. (2.17) is *low*, if the sample colors are close to the mixed color C:

$$\begin{aligned} w(F^i) &= exp\left\{ -||F^i - C||^2 / \max_s \left(\|F^s - C\|^2 \right) \right\} \\ w(B^j) &= exp\left\{ -||B^j - C||^2 / \max_s \left(\|B^s - C\|^2 \right) \right\}, \end{aligned} \tag{2.18}$$

where the function $\max_s$ return the maximum squared difference between the mixed color C and all corresponding fore- or background color samples, respectively. The confidence is computed for every pair of fore- and background samples, and the pairs with the highest confidences are used to compute a pixel-wise estimate of α. In [WC07a] it is assumed that pixels whose color is close to the color of the fore- and background samples are more likely to be fully foreground or background themselves. Thus in [WC07a] the alpha value of pixels with a low confidence value (i.e. those with a color similar to the samples) are biased towards 0 or 1.

This bias is encoded in an objective function, where the pixel-wise estimated alpha defines the data term that is combined with the smoothness term of [LLW08]. The final alpha matte is obtained by minimizing this objective function by solving a set of linear equations.

In chapter 5 we present an algorithm which improves on the algorithm of [WC07a] by using a new way to obtain the initial sample set and a new paradigm to compute the confidence value.

2.3 Matting Evaluation

We have seen in the previous section that many approaches to matting exist. Thus a quantitative benchmark for these methods becomes vital to reveal their strengths and weaknesses, thus providing the ground for novel research directions. A benchmarking system requires:

1. A challenging, high-quality ground truth (GT) test set.
2. An online evaluation repository that is dynamically updated with new results.
3. Perceptually motivated error functions.

In the following we will review related work in these areas.

2.3.1 Ground Truth Data

Recently, ground truth datasets for image matting have been published by Levin et al. [LRAL08] and Wang et al. [WC07a]. Levin et al. [LRAL08] captured three different soft toys in front of a computer monitor which displayed seven different (natural) background scenes. The ground truth alpha mattes for these images were then obtained with triangulation matting (see section 1.2.1 for details about triangulation matting). Unfortunately, the ground truth alpha mattes of [LRAL08] are considerably affected by noise, which might lead to unreliable evaluation results.

Another ground truth dataset was proposed by Wang et al. [WC07a]. In contrast to Levin et al. [LRAL08], which captured the ground truth data in a restricted studio environment, Wang et al. [WC07a] obtained ground truth information for real-world (outdoor) images. This was done by applying existing matting methods to the natural images, and their resulting alpha mattes were manually combined to a reference solution. Clearly, such reference solutions are biased towards the matting algorithms that were used to create the ground truth. A biased dataset might be not be suitable for a fair comparison of matting algorithms.

In chapter 7, we will propose a new dataset of 35 natural images whose reference solutions are of very high-quality (i.e. have a very high signal to noise ratio) and that were generated independently of previous matting approaches.

2.3.2 Online Evaluation Repository

Recently proposed benchmarks for computer vision problems such as stereo [SS02] or optical flow [BSL$^+$07], have been made freely available on the web. Such an online benchmark allows other researchers to interactively analyze recent work and to extend the evaluation with new results.

Unfortunately, no such online benchmark has been developed so far for the task of image matting. Thus we establish a dynamic online benchmark (described in chapter 7) that provides the ground truth data and scripts that enable the research community to complement our evaluation with new results.

2.3.3 Perceptually Motivated Error Functions

To quantitatively evaluate the performance of matting algorithms, their resulting alpha mattes have to be compared to the ground truth using an error metric. Ideally, we should use error metrics that correlate to the visual quality as perceived by humans. Although in other areas of computer vision, specialized perceptual distance measures exist for the task of image segmentation [PV08, CDGE02] or color constancy [GGL08], we are not aware of any perceptual distance metrics far for the task of image matting. Thus previous matting evaluations have been tied to simple pixel-wise error measures that are not necessarily correlated to the human perception.

In this thesis we go beyond these simple evaluation methodologies and develop quantitative error measures that are based on subjective human perception.

Chapter 3

Alpha Formation Process

In the last chapter we reviewed state-of-the-art approaches to matting. As we have seen, they usually exploit the local correlations between nearby pixels to infer alpha. However, ambiguities in the solution space are oftentimes not resolved correctly by these approaches.

For instance, it has been observed (e.g. [LRAL08]) that a major problem is that for insufficient user input (i.e. large trimap) the cost function used in [LLW08] has a large space of (nearly) equally likely solutions[1]. The resulting matte of [LLW08] is shown in figure 3.1(c), given the image and trimap in figure 3.1(b). The result is imperfect, since some hairs are cut-off.

To overcome this ambiguity in the cost function, e.g. Wang et al. [WC07a] additionally modeled the colors of the fore- and background regions of the trimap in the framework of [LLW08] [2]. However, the result is even worse, as shown in figure 3.1(d). In this case the problem is that some dark-green areas in the image background are explained as semi-transparent layers, i.e. dark-green is a mix of dark foreground with green background. The solution in figure 3.1(d), which shows large semi-transparent regions in the background, is plausible given the color observations. However, it is a solution which is physically very unlikely.

Hence, a physically valid model for alpha is necessary to restrict the solution space. Therefore, the goal of this chapter is (i) to analyze the physical sources that can cause a

---

[1] Another problem is that the color line model of [LLW08] does not hold for highly textured patches. From our experience, however, this seems less important.

[2] The framework of Levin et al. [LLW08] was originally introduced in [LLW06].

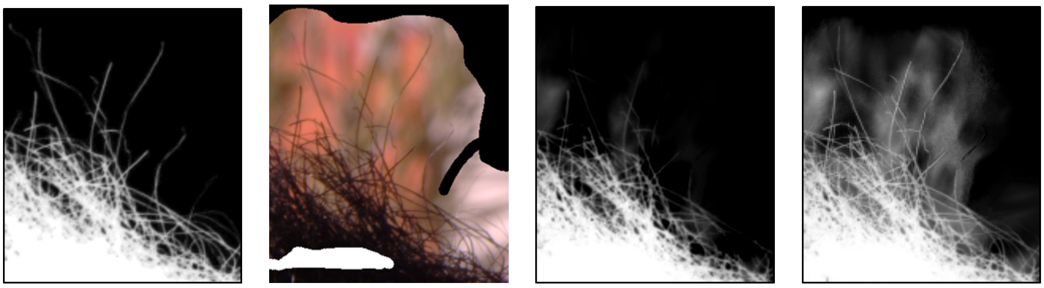

(a) Ground truth alpha matte (b) Input image with trimap (c) Result of [LLW08] (d) Result similar to [WC07a]

Figure 3.1: **Matting ambiguities.** Ambiguities in alpha matting are often not resolved correctly by state-of-the-art algorithms (c,d). Hence, modeling the prior probability of the alpha matte is necessary to restrict the solution space.

mixing of layer colors (i.e. fractional α values: $0 > \alpha < 1$) and (ii) to derive a physically motivated model for alpha, based on this analysis.

Generally speaking, there are two sources that cause the colors of the fore- and background layers to be blended:

- *Translucent materials* (e.g. window glass) and
- The *imaging process* (e.g. defocus blur).

In the following, we will analyze both sources in more detail. Afterwards, we will focus on fractional alpha values caused by the imaging process.

3.1 Translucent Materials

A mixing of the layer colors can be caused by a foreground object that is made up by translucent materials which only partially block the light from the background (i.e. they let the background "shine through"). In order to understand why certain objects appear translucent, we have to consider how light interacts with matter. Therefore, let us consider figure 3.2, which illustrates this process. We can see that light which impinges an object is (partially) *reflected* from its surface and the remaining light *propagates* through the

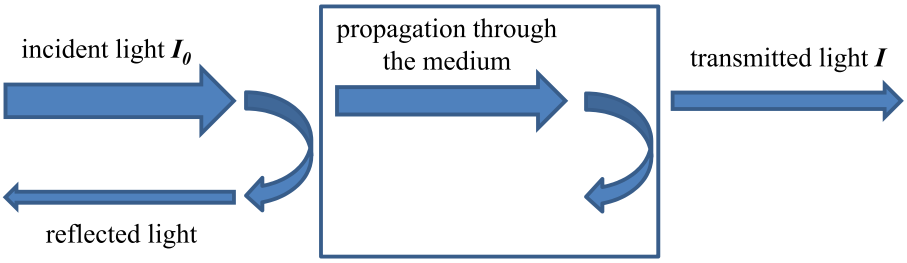


Figure 3.2: **Interaction of light with matter.** See the text for a detailed description. This figure is modified from [Fox01].

medium. During propagation through the medium, the light can be attenuated by *scattering* or *absorption*. Finally, if some remaining light reaches the back-surface of the medium, the light is either reflected again or *transmitted*.

A material appears opaque (i.e. $\alpha = 1$) to the human eye if it only absorbs, scatters or reflects all visible light (i.e. does not transmit any light) [Kat02]. On the other hand, a material is *transparent* if it transmits all light (i.e. $\alpha = 0$), and *translucent* (also denoted as *semi-transparent*) if it partially transmits light (i.e. $0 < \alpha < 1$). In this thesis we will use both the terms semi-transparency and translucency, to denote a mixing of layer colors caused by either light-transmitting materials or the imaging process.

The fraction of the incident light that is transmitted through a material is commonly denoted as *transmissivity* and is inversely proportional to α. Formally, the transmissivity T for a medium of thickness l is a function of the reflection and propagation properties of the object's material [Fox01]:

$$(1 - \alpha) \approx T = (1 - R_1)e^{-\mu l}(1 - R_2). \tag{3.1}$$

Here, R_1 and R_2 denote the fractions of light that are reflected from the front- and back surface of the object, respectively. If the reflection coefficients R_1 or R_2 are 1, then the corresponding surface reflects all light, whereas a coefficient of 0 means that no light is reflected. The middle term of eq. (3.1) accounts for the exponential attenuation of the light due to absorption and scattering, according to *Beer's law* (see e.g. [Fox01]). Thereby, the *attenuation coefficient* μ measures the fraction of light that is absorbed or scattered in a unit length of a medium. Materials with a small attenuation coefficient let more light be propagated through the medium, whereas larger values cause a blocking of the light.

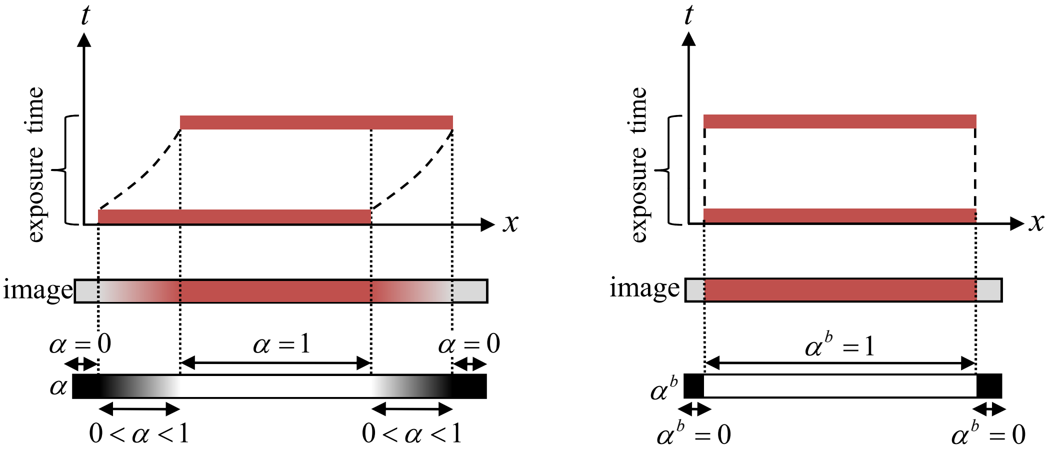
3.2 Imaging Process

Apart from material properties, fractional alpha values can be induced by the imaging process. When taking a picture, the light rays emanating from the scene are captured with an imaging system, which basically comprises a lens and a camera sensor. The *Point Spread Function* (PSF) models how an imaging system projects a point in the scene to the final image. In other words, the PSF describes how a point in the scene is spread over the image due to the deformations induced by the imaging process. The PSF is governed by blurring effects that are caused by motion of the camera or scene objects (motion blur), the camera lens (defocus blur) and the limited resolution of the camera sensor (resolution blur). In the following, we examine how these blurring effects cause a blending of the image layers and thus generate fractional alpha values.

3.2.1 Motion Blur

When capturing a digital photograph, the light travels through the lens and finally reaches the sensor plane. For simplicity, let us assume the lens to be an idealized pinhole, which projects all light rays through a common center of projection. Hence, each point in the scene is mapped to exactly one point in the image (i.e. there is no defocus blur). The light is exposed to the sensor for a certain time period, which is controlled by a shutter mechanism. In practice, this exposure time is a finite time period (e.g. $1/60$ second) and its choice depends on the lighting conditions. If the camera or a scene object is moving during exposure time, a point in the scene may be mapped to multiple points on the sensor, which causes a blending of colors. This is especially true for long exposure times (e.g. longer than $1/60$ second) and for fast moving objects (e.g. moving cars).

To illustrate the effect of motion blur, let us consider figure 3.3(a). It shows a 1-D scene with a red-colored solid (opaque) layer, which is photographed in front of a gray background (the background is not visualized here). While the scene is exposed to the



(a) Moving object (b) Static object

Figure 3.3: **Fractional alpha values originating from motion blur.** A red-colored solid layer is photographed. In (a) the foreground layer moves along the x-axis, hence occludes the background for a fraction of the full exposure time. Therefore, the layer colors are mixed around the boundary of the foreground layer. In (b) the foreground layer is static, thus fully occludes the background during the whole exposure time. As a consequence, the foreground colors did not mix with the background and a binary alpha matte α^b is generated. This figure is modified from [Jia07].

camera, the red layer is moving to the right along the x direction. The resulting image is shown directly below the scene. We can see that in between the dotted lines the color of the red foreground layer is mixed with the gray background. The mixing factor α at image point p is thereby determined by the percentage of time that the foreground layer occludes the background at point p:

$$\alpha(p) = t_{occ}(p)/t_{total}. \tag{3.2}$$

Here, t_{total} denotes the total exposure time and $t_{occ}(p)$ refers to the time span where the foreground layer occluded the background at point p. The alpha matte computed using eq. (3.2) is illustrated in the bottom row of figure 3.3(a).

Now let us consider figure 3.3(b). It shows the image resulting from a static (non-moving) layer. In contrast to the moving layer, the fore- and background colors did not mix. As a consequence, the corresponding alpha matte α^b is binary (i.e. $\alpha^b \in \{0, 1\}$).

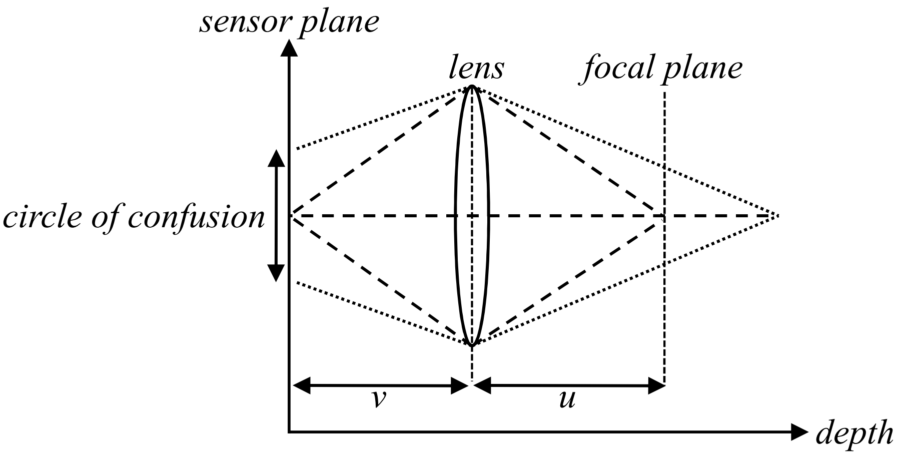


Figure 3.4: **Relation of defocus to image blur.** Points in the scene are projected to circular shapes on the camera sensor. The size of this circle depends on the distance of the scene point to the focal plane. (Figure after [FS07].)

Assuming planar front-to-parallel fore- and background layers with constant layer colors, we can derive the blurred image by convolving the unblurred (sharp) image with a spatially invariant motion blur kernel K_σ^{motion} [AFM98]. The spatial extent σ of the blur kernel thereby depends on the motion of the camera or scene object during the exposure time. Similarly, we can consider the alpha matte as a blurring of two constantly colored layers (i.e. a white foreground layer in front of a black background layer), thus we can model the alpha matte α as a convolution of the binary (unblurred) matte α^b (see figure 3.3(b)) with a spatially invariant motion blur kernel K_σ^{motion} (K_σ^{motion} is induced by the motion of the foreground layer) as $\alpha = \alpha^b \otimes K_\sigma^{motion}$.

3.2.2 Defocus Blur

In the previous section, we assumed an idealized pinhole camera in order to analyze motion blur. However, in practical imaging systems, a lens is used to map the light of the scene onto the camera sensor. The simplest model of a lens is the *thin lens model*, which is illustrated in figure 3.4. It shows an illustration of a lens at distance v from the camera sensor plane. The plane at distance u, which satisfies the thin lens law $1/u + 1/v = 1/F$

with focal length F, is called the *focal plane*[3]. A scene point that lies on the focal plane is projected to a single point on the camera sensor plane [BW80]. Other scene points will appear blurred, since they are projected onto a circular area on the sensor which is known as the *circle of confusion* (COC)[4]. The diameter of this circle is growing proportionally with the distance of the scene point to the focal plane.

In order to better see the relationship between defocus blur and the alpha matte, we will now reintroduce the lens model from a different viewpoint. Instead of constructing the image by projecting scene points onto the sensor plane, we can equivalently consider the intensity observed at each image point as being constructed by a mixture of the light that all scene points project to it.

This model can be considered as reversion of the model introduced in figure 3.4 and therefore it is called the *Reversed Projection Blurring Model* [AFM98]. An illustration of the reversed model is depicted in figure 3.5(a). It shows a 1-D scene with a red-colored solid layer in front of a gray background layer. The focal plane lies in between the two layers and therefore both are defocused. For every point on the sensor plane we define a *chief ray*, which is a straight line that starts at the sensor point and passes through the center of the lens (figure 3.5(a) shows the chief ray corresponding to image point p). The intersection of the chief ray with the focal plane defines the apex of a double cone (assuming a circular shaped aperture) that has its base at the lens. In figure 3.5(a) this double cone is illustrated by the green dashed lines.

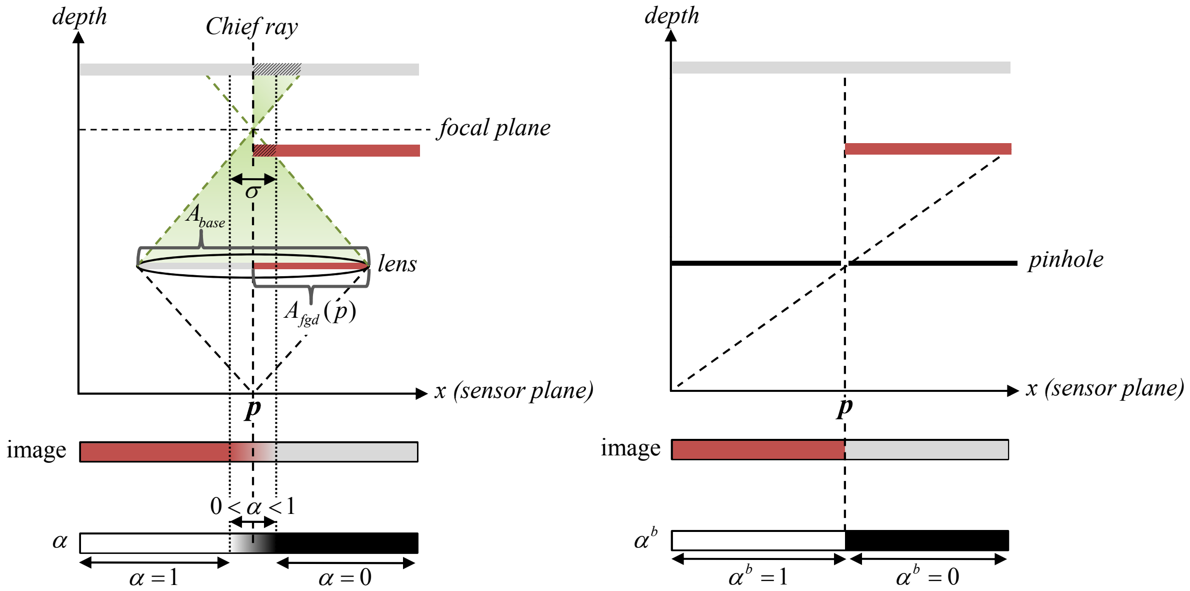
The key observation is now that the color at a sensor point p is a linear blend of the colors[5] of the layers that intersect this double cone [MLS06]. The blending factor (which corresponds to the alpha value of the foreground layer) is thereby given by the degree that the foreground layer occludes the background inside the double cone. More precisely, the blending factor α at point p is defined by the area $A_{fgd}(p)$ occupied by the foreground (fgd) layer after projecting it to the base of the double cone:

---

[3] The focal length F depends on the shape and material properties of the lens.

[4] In practice the circle of confusion is not perfectly circular, but its blur pattern is governed by the shape of the aperture.

[5] For simplicity we assume constant layer colors, which is sufficient to derive the relationship to the alpha matte.



Note that the lens projects the scene to the sensor, such that the resulting image (and corresponding alpha matte) appear inverted.

Note that the pinhole projects the scene to the sensor, such that the resulting image (and corresponding alpha matte) appear inverted.

(a) Lens camera

(b) Pinhole camera

Figure 3.5: **Fractional alpha values originating from defocus blur.** A red foreground layer is photographed in front of a gray background. In (a) the foreground layer is out-of-focus, which results in a mixing of the foreground colors with the background. In (b) the scene is captured through a pinhole. Hence, all pixels are in focus and a sharp image is generated.

$$\alpha(p) = A_{fgd}(p)/A_{base}, \tag{3.3}$$

where A_{base} is the area of the cone's base (i.e. the area of the lens cross section). In the example in figure 3.5(a), the foreground layer projected to the lens occupies exactly half of the lens area, hence $\alpha(p) = 0.5$.

Given the above analysis we can reconstruct the final image and alpha matte α, which we depict in the lower part of figure 3.5(a). We can see that in between the dotted lines the color of the red foreground layer is mixed with the gray background, hence there are fractional alpha values. The size of this blurred region (i.e. the distance between the dotted lines) corresponds to the diameter σ of the circle of confusion at the distance of the foreground layer. In contrast, figure 3.5(b) shows the image and alpha matte α^b that result from photographing the same scene with an idealized pinhole camera (i.e. without defocus blur). We can see that in this case the alpha matte α^b is a binary function (i.e. $\alpha^b \in \{0, 1\}$).

Similar to the motion blur case (discussed in section 3.2.1), we can derive the alpha matte α by convolving the unblurred α^b with the defocus blur kernel (circle of confusion) $K_\sigma^{defocus}$ with diameter σ: $\alpha = \alpha^b \otimes K_\sigma^{defocus}$.

3.2.3 Resolution Blur

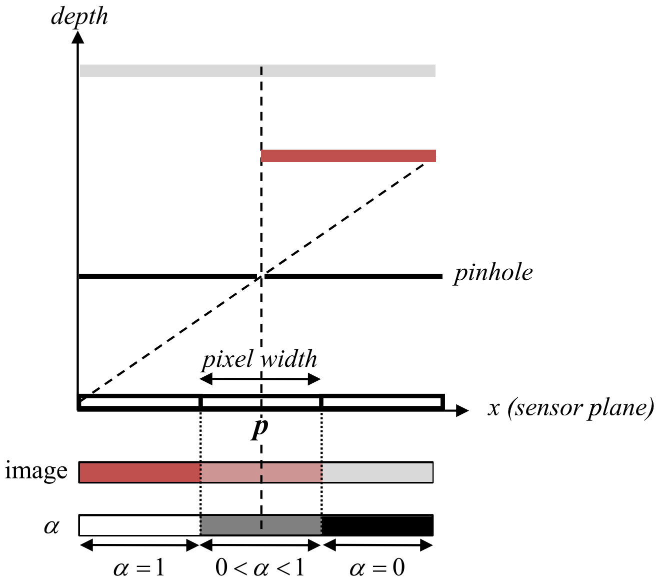
When taking a digital photograph, the light travels through the lens and finally reaches the imaging sensor where the incident photons are converted to an electrical signal. In the above analysis we assumed an idealized imaging sensor that has an infinitely high-resolution. However, real-world digital cameras compute the intensity at each image point by integrating the incident light (i.e. photons) over a finite sized sensor area, called a pixel. The larger the pixel area, the more photons will reach it during exposure time. As a consequence, larger pixels can use more photons to estimate the intensity and thus have a better signal-to-noise ratio [CCGW00]. A typical pixel size used in high-end digital cameras (e.g. Canon 1D MarkIII) is 7.2 x 7.2 μm [Inc08].

The integration of light over a sensor pixel is illustrated in figure 3.6. It shows a 1-D scene of a red layer in front of a gray background. The scene is projected through a pinhole onto the sensor plane, which consists of three pixels. The light rays which reach the middle sensor pixel originate from both of the two scene layers. Thus the color of the middle pixel is a linear combination of the two layer colors. The mixing factor $\alpha(p)$ at pixel p is given by the area A_{fgd} that the foreground layer occupies after projecting the scene onto pixel p.

$$\alpha(p) = A_{fgd}(p)/A_{total}, \tag{3.4}$$

where A_{total} is the total area of a sensor pixel.

Thus the observed (low-resolution) alpha matte α can be obtained by spatial integration of the binary segmentation α^b that would result from an idealized sensor with infinite high-resolution. Following [BK00] this spatial integration can be modeled by a convolution of



Note that the pinhole projects the scene to the sensor, such that the resulting image (and corresponding alpha matte) appear inverted.

Figure 3.6: **Fractional alpha values originating from resolution blur.** The colors of the fore- and background layer are blended because of the limited sensor resolution.

this binary segmentation with a box filter K_σ^{box}:

$$\alpha = \alpha^b \otimes K_\sigma^{box}, \tag{3.5}$$

where the width σ of the box function corresponds to the size of the sensor pixel. This convolution can also be regarded as downsampling of a high-resolution binary segmentation with a box filter.

3.3 Segmentation-based Model for Alpha

In the previous sections we have seen that fractional alpha values can either originate from light-transmitting materials or from the imaging process. Accurately modeling fractional alpha values caused by light-transmitting materials is hard, because they depend on numerous parameters of the scene (e.g. thickness and material properties of the foreground object) which are hard to predict.[6]

In contrast, the fractional alpha values that are induced by the imaging system can be modeled by a simple convolution of a binary (unblurred) matte with the system's PSF. Solving for the model parameters (i.e. binary segmentation and PSF) is a well known problem in image deblurring and a large body of literature exists on that topic (see e.g. [LWDF09, JSK08, Jia07, FSH$^+$06]).

Although the sources of the fractional alpha values depend on the photographed scene, the fractional alpha values in typical images (e.g. portraits of a humans) are mainly induced by the imaging process. It is also worth noting that almost all recent work on image matting was tested on images where the fractional alpha values where mainly caused by the imaging process. Therefore, the goal of this section is to introduce a model for the alpha matte that is based on the imaging process.

For the derivation of the blurring effects in section 3.2, we have so far assumed that the scene consists of only two planar front-to-parallel layers (i.e. foreground and background). We have seen that in this case the alpha matte of the foreground layer can be modeled as a convolution of a two-dimensional binary segmentation (i.e. the unblurred sharp scene as observed through a pinhole) with a spatially invariant PSF. However, in real world scenes, the foreground object may vary in depth and might be self-occluding (e.g. overlapping hair strands or overlapping legs). As a consequence, the PSF varies with the depth, and multiple PSFs contribute to pixels around occlusion boundaries. One can potentially account for depth-dependent blur by making the PSF spatially varying. Asada et al. [AFM98] investigated such a spatially varying PSF, but found it to neglect the simultaneous existence of multiple PSFs around occlusion boundaries. More sophisticated blur models [AFM98, Coo07] can account for multiple kernels, but they require the knowledge of the underlying three-dimensional segmentation (i.e. multiple depth values at each pixel). Reconstruction of this three-dimensional binary segmentation from only one single natural image is very hard in practice, and therefore we follow the deblurring literature (see e.g. [SXJ07, Lev06, JSK08]) and use an approximative model which accounts for depth-dependent blur with a spatially varying PSF.

---

[6] Note that by assuming that the material properties and thickness of the objects vary gradually, we could still model the alpha matte as a smooth function.

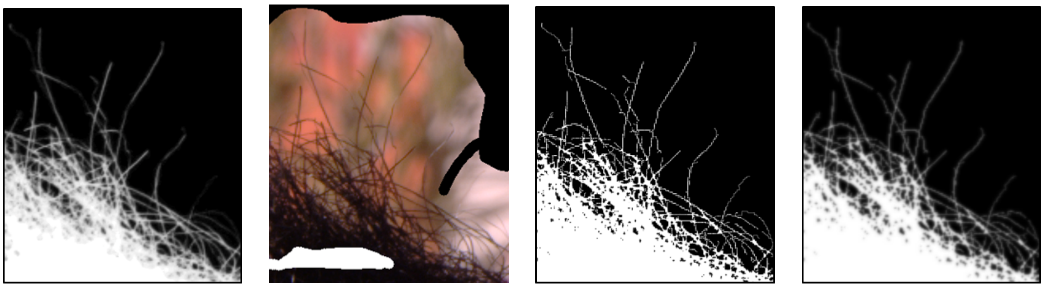

(a) Ground truth alpha matte (b) Input image with trimap (c) Binary segmentation (d) Result of (c) convolved with PSF.

Figure 3.7: **Alpha model.** We model the ground truth alpha (a) as a combination of a binary segmentation (c) and a PSF. The result (d) is close to the ground truth.

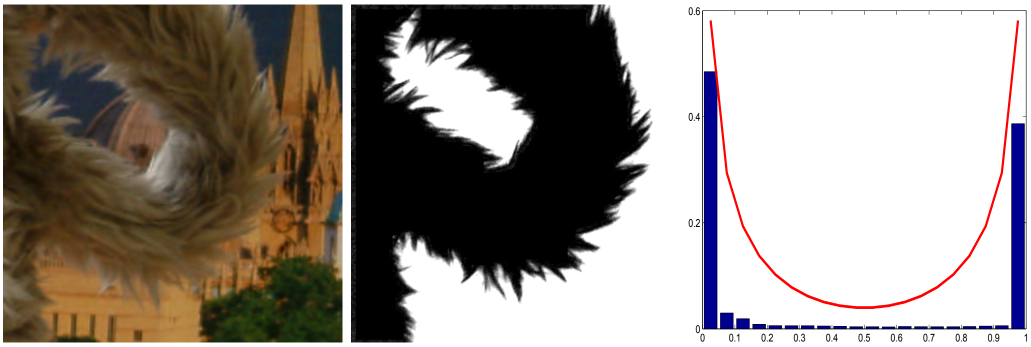
Our approximative model describes the observed alpha matte α as a convolution of an underlying, potentially higher-resolution, binary segmentation α^b with a spatially varying point spread function K, whose result is potentially downsampled:

$$\alpha = D(K \otimes \alpha^b). \tag{3.6}$$

Here, $\otimes$ denotes convolution and D is the downsampling function. In our formulation the kernel K accounts for all fractional alpha values induced by the imaging system. In chapter 6.4, we will qualitatively and quantitatively demonstrate that this approximative model is a good prior for many real alpha mattes.

To illustrate our approximative model, let us consider figure 3.7(a), which shows the ground truth (GT) alpha matte for the image crop in figure 3.7(b). We estimated the underlying binary segmentation α^b (figure 3.7(c)) using the GT alpha matte. Convolving this binary segmentation with our estimated PSF K gives the result in figure 3.7(d). It is very close to the true alpha matte and qualitatively and quantitatively better than the results of Levin et al. [LLW08] and Wang et al. [WC07a] on this crop (compare the results of [LLW08, WC07a] in figure 3.1).

Another major advantage of this model is the potential to easily incorporate prior knowledge. For example we know that the binary segmentation of objects which are non-occluded (a common assumption [VKR08]) are usually connected. In fact, we show in



(a) Image (b) Alpha matte (inverted) (c) Alpha distribution

Figure 3.8: **Sparse distribution of alpha.** For the image in (a) the ground truth alpha matte is depicted in (b). In (c) the histogram of the alpha matte in (b) is depicted, which follows a beta distribution (red line). Figure from [WFZ02].

chapter 7 that connectivity is an important factor for the human perception of alpha mattes. However, for continuous-valued alpha mattes, such a prior is significantly harder to formulate and to enforce than for binary masks.

Let us now discuss a property of our model that has been exploited by some previous matting approaches to model the global distribution of the alpha values. In particular, our segmentation-based model (see eq. (3.6)) generates alpha mattes whose fractional alpha values occur only at the boundary of an object and most parts of the matte have a value of either zero or one. Some work on matting (e.g. [WFZ02, WC07a, LRAL08]) exploited this insight to formulate a so-called sparsity prior on alpha. However, the respective authors did not motivate such a prior from the alpha formation process. Instead, these sparsity priors were based on empirical observations that fractional alpha values in real mattes were distributed sparsely.

For instance, [WFZ02] found that the global distribution of alpha in ground truth mattes follows a beta distribution (see figure 3.8). Based on this observation, [WFZ02] formulated a pixel-wise prior on alpha that pushes as many pixels as possible to 0 or 1. In the approach of Levin et al. [LRAL08], discussed in section 2.2.1, a similar prior as in [WFZ02] was employed. It prefers those “matting components” which contain most 0 and 1 values. Finally, in Wang et al. [WC07a], pixels whose estimated alpha values have a low confidence were biased towards an alpha value of 0 or 1 (see section 2.2.2 for a more detailed description).

It is important to note that these generic sparsity priors are employed to each pixel independently of the local context. Hence, they do not respect the alpha formation process. Ideally, the prior should be based on the approximative image formation model in eq. (3.6), which depends on the underlying binary segmentation. We will show in chapter 6 that such a segmentation-based prior is better than the previously used generic sparsity priors.

Chapter 4

A New Trimap Extraction Method

Given an input image, most natural image matting algorithms start by asking the user to specify a trimap, which is a partitioning of the image into foreground, background and unknown regions. Matting algorithms then compute an alpha matte for the unknown regions, while using the fore- and background regions of the trimap as boundary constraints. The quality of the resulting alpha matte, as well as the computational expenses required to obtain this matte, heavily depend on the accuracy of the user-provided trimap. Hence, to quickly derive high-quality results the user should assign as many pixels as possible to either the fore- or the background. However, manually drawing an accurate trimap is a very tedious and time-consuming process for the user.

Therefore, in this chapter, we present a new and efficient segmentation method that automatically infers an accurate trimap from only a rough indication of the fore- and background regions. An overview of our approach is given in figure 4.1. In the first step, the user indicates the fore- and background regions by placing only a few scribbles on the input image (figure 4.1 left). Using these scribbles, our algorithm automatically computes an accurate trimap (figure 4.1 middle). The user can refine this trimap and remove obvious mistakes. For instance, in our approach the user can interactively adjust the size of the unknown trimap region with a slider interface. In the final step, an alpha matte is computed using the previously proposed matting algorithm of Wang et al. [WC07a] (figure 4.1 right).

The advantage of our two-step process (i.e. trimap extraction followed by trimap-based alpha matting) over directly computing the alpha matte from the user-defined scribbles is

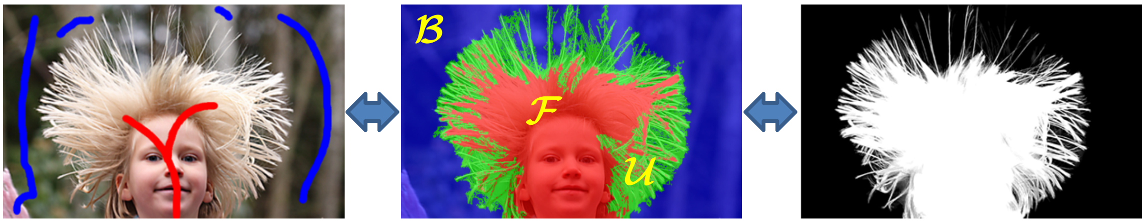
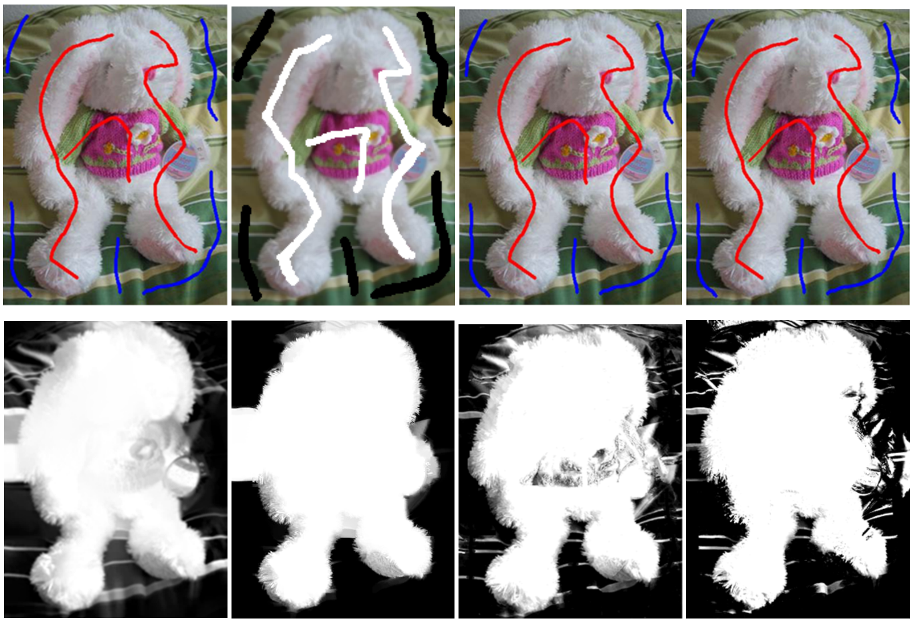


Figure 4.1: **Our interactive matting framework.** Given an input image and user-defined scribbles (left), we automatically compute an accurate trimap (middle). From the trimap an alpha matte (right) is computed, together with the true fore- and background colors (not shown here).

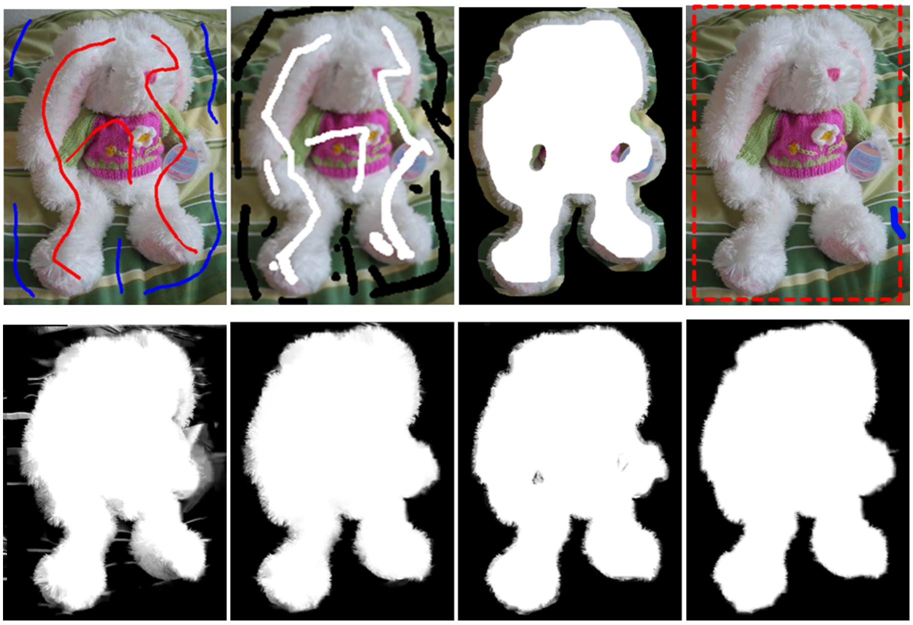
both speed and higher quality. The main benefit in speed comes from the observation that in a typical image most pixels belong solely to either the fore- or background (see chapter 3). For these fore- and background pixels, computationally expensive matting algorithms which recover the full range of fractional alpha values should not be invoked. For example, computing the alpha matte directly from the user-defined scribbles with the approaches of [WC05], [LLW08] and [GCL$^+$06] takes between 20 and 200 seconds for a typical low-resolution, 0.3 Mpix, image. Using the same image and user-defined scribbles, we first automatically extract a trimap with our approach and then employ the matting algorithm of [WC07a] to derive the alpha matte, which in total requires only 4.5 seconds.

Moreover, not only speed but also the quality of the matte is improved by our two-step process. This is demonstrated in figure 4.2, which shows the result of different matting methods for the image of a soft toy. We see that for this image, all tested matting approaches perform rather poorly if the matte is computed directly from sparse scribble input (see figure 4.2(a-e)). To achieve good results for this image, either many scribbles (figure 4.2(f)) or an accurate trimap (figure 4.2(g)) are necessary. In contrast, our high-quality result in figure 4.2(h) requires only a single bounding box selection and one additional background scribble as user input. Note that pixels outside the bounding box define the background constraints, and all other pixels are regarded as unknown. Hence, there are no foreground constraints.

The work most similar to our approach is that by Juan et al. [JK05], where the idea of extracting a trimap using a scribble-based interface has been introduced. However, the trimap extraction method of Juan et al. [JK05] is based on a model that mainly relies on



(a) Spectral Matting [LRAL08]
(b) Closed-form Matting [LLW08]
(c) Iterative Matting [WC05]
(d) Easy Matting [GCL^{+}06]



(e) Robust Matting [WC07a]
(f) Closed-form Matting [LLW08]
(g) Iterative Matting [WC05]
(h) Our approach

Figure 4.2: **Comparison of scribble-based matting approaches** (see text for discussion). Scribbles are marked in either red (fgd) and blue (bkg) or white (fgd) and black (bkg). Our result as shown in (h) was achieved by selecting the foreground object with a bounding box and drawing one additional background scribble. Note that our approach can also handle more challenging alpha mattes, e.g. as shown in figure 4.1. All results we show were either taken from the original papers or created with the original implementation of the respective authors.

color features. (See chapter 2.1 for a detailed review of [JK05].) Such a model is ambiguous for images where the colors of the fore- and background are overlapping.

The main contribution of our work is a new model that can resolve these ambiguities by considering several images cues, thus enabling us to extract trimaps of considerably higher quality. Our model draws its motivation from the alpha formation process, where the majority of fractional alpha values are induced by blurring the underlying binary segmentation of the foreground object with the camera's Point Spread Function (see chapter 3). Our approach predicts the structure of the underlying binary segmentation, which gives a good indication of the spatial extent of the unknown trimap region. Furthermore, we use a large set of images with known ground truth alpha mattes to train a classifier that predicts the ratio of unknown pixels in the trimap from the input image.

The remainder of this chapter is organized as follows. In section 4.1 we give an overview of our approach, and then we detail our model in section 4.2. The training of our classifier is discussed in section 4.3. Finally, in section 4.4, we qualitatively and quantitatively demonstrate that our approach outperforms the approach by Juan et al. [JK05] as well as other techniques that could be used to extract a trimap from sparse user input.

4.1 Overview

We start by defining some notation. Let $\mathcal{I}$ be the set of all pixels in the input image and α_i the alpha value at pixel i. We define a trimap as a partitioning of $\mathcal{I}$ into the three subsets $\mathcal{F}, \mathcal{B}$ and $\mathcal{U}$ (see figure 4.1 middle). The subsets are defined as $\mathcal{B} = \{i|\alpha_i < \epsilon\}$, $\mathcal{F} = \{i|\alpha_i > (1-\epsilon)\}$ and $\mathcal{U} = \mathcal{I} \backslash (\mathcal{F} \cup \mathcal{B})$, where we choose $\epsilon = \frac{5}{255}$. We also introduce two additional subsets $\mathcal{F}', \mathcal{B}'$ where $\mathcal{B}' = \{i|\alpha_i \leq 0.5\}$ and $\mathcal{F}' = \{i|\alpha_i > 0.5\}$. The subsets $\mathcal{F}'$ and $\mathcal{B}'$ define a binary segmentation of the image and the transition from $\mathcal{F}'$ to $\mathcal{B}'$ corresponds to the boundary of this segmentation. Obviously, it is $\mathcal{F} \subset \mathcal{F}'$, $\mathcal{B} \subset \mathcal{B}'$ and $\mathcal{F}' \cup \mathcal{B}' = \mathcal{F} \cup \mathcal{B} \cup \mathcal{U} = \mathcal{I}$.

To obtain a trimap, one could follow Juan et al. [JK05] and assign each pixel in the image to one of the three labels $\mathcal{F}, \mathcal{B}, \mathcal{U}$. (For simplicity, sets and labels have the same name, e.g. $\mathcal{F}$.) In our approach we additionally assign each pixel to one of the two labels $\mathcal{F}'$ and $\mathcal{B}'$. This has the main advantage that we can model the transition from $\mathcal{F}'$ to $\mathcal{B}'$, which usually coincides with strong edges in the image[1]. (Commonly, this transition is modeled by segmentation techniques such as [BJ01] or [RKB04] that aim to infer a binary segmentation of the image into fore- and background.)

The advantage of additionally modeling the transition from $\mathcal{F}'$ to $\mathcal{B}'$ is twofold. Firstly, it allows us to detect the spatial location of the unknown trimap region more accurately. Secondly, some parts of this transition usually coincide with clean, “sharp boundaries” of the foreground object, where the unknown trimap region can be modeled as a small band around this boundary. For instance, the sharp parts of the boundary of the object in figure 4.3(d) are marked in red. We discuss the detection of sharp boundaries in detail in section 4.2.4.

Ideally, we would like to define an energy function that models the optimal assignment of each pixel in the image to all 5 labels $\mathcal{F}, \mathcal{F}', \mathcal{B}, \mathcal{B}', \mathcal{U}$ and optimize this energy function globally. Instead, we employ a two-step process that allows a more comprehensive model and higher speed. In particular, we first obtain a binary segmentation into the sets $\mathcal{F}'$ and $\mathcal{B}'$ using the binary segmentation approach “GrabCut” proposed by Rother et al. [RKB04]. We use the energy function and parameter settings as defined in [RKB04], and the interested reader is referred to the respective paper for details. Following this binary segmentation, we further partition the image into the three labels $\mathcal{F}, \mathcal{B}$ and $\mathcal{U}$. In the following, we show that this trimap segmentation can be posed as a binary classification problem and that the corresponding energy function can be minimized with graph cuts [BK04].

4.2 Model

We start by assuming that each pixel in the image has been already classified into $\mathcal{F}'$ or $\mathcal{B}'$ by using the “GrabCut” algorithm [RKB04]. We now assign each pixel to one of the three labels $\mathcal{F}, \mathcal{B}$ or $\mathcal{U}$. Since $\mathcal{F} \subset \mathcal{F}'$ and $\mathcal{B} \subset \mathcal{B}'$, a *binary* classification into two labels $\mathcal{U}$ and $\bar{\mathcal{U}} = \mathcal{F} \cup \mathcal{B}$ is sufficient to derive the trimap. (Given $\mathcal{F}'$ and $\mathcal{B}'$, each pixel in $\bar{\mathcal{U}}$ is uniquely specified to be in either $\mathcal{F}$ or $\mathcal{B}$).

The goodness of a binary segmentation of the image into $\mathcal{U}$ and $\bar{\mathcal{U}}$ is characterized by

---

[1] Note that the other transitions, e.g. the transition from $\mathcal{B}$ to $\mathcal{U}$, do not depend on the image edges and are therefore harder to predict.

the following energy function E:

$$E(\mathbf{x}) = \sum_{(i,j)\in\mathcal{N}} \theta_b V^b(x_i,x_j) + V^s(x_i,x_j) \\ + \sum_i D^c(x_i) + D^p(x_i) + \theta_{b'} D^b(x_i) + \theta_s (D^s(x_i))^{\theta_{s'}}, \tag{4.1}$$

where $x_i \in \{\mathcal{U},\bar{\mathcal{U}}\}$ denotes the label of pixel i and the vector **x** encodes the labeling of all pixels in the image. The set of neighboring pixels (8-neighborhood) is denoted by $\mathcal{N}$, and $\theta_b, \theta_{b'}, \theta_s, \theta_{s'}$ define the model parameters. The energy can be locally optimized using graph cuts [RKB04] or the parametric maxflow technique [KBR07], depending on the choice of the free parameters during runtime (see below). The individual terms D^c,D^p and D^s are defined in subsections 4.2.1-4.2.3. The terms D^b and V^b are discussed in subsection 4.2.4 and, finally, the term V^s is defined in subsection 4.2.5.

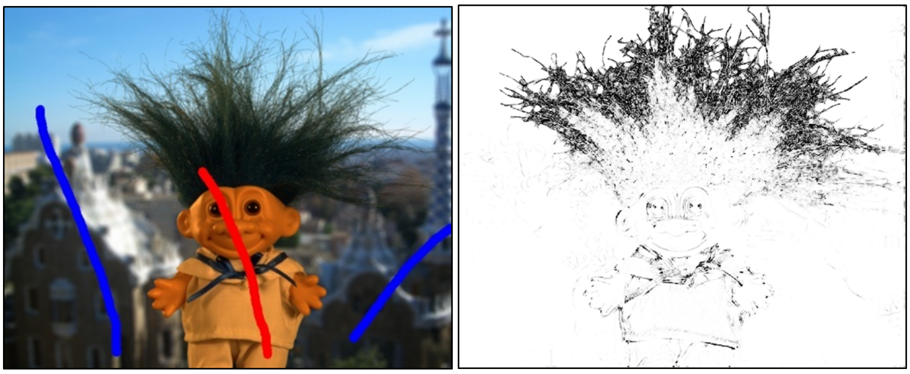
4.2.1 Color

The unary term D^c for pixels $x_i \in \bar{\mathcal{U}}$ models the color distribution of the fore- and background regions of the trimap as

$$D^c(x_i) = \begin{cases} -logP(c_i|\theta_{\mathcal{F}}) & \text{if } x_i \in \mathcal{F}' \\ -logP(c_i|\theta_{\mathcal{B}}) & \text{if } x_i \in \mathcal{B}', \end{cases}$$

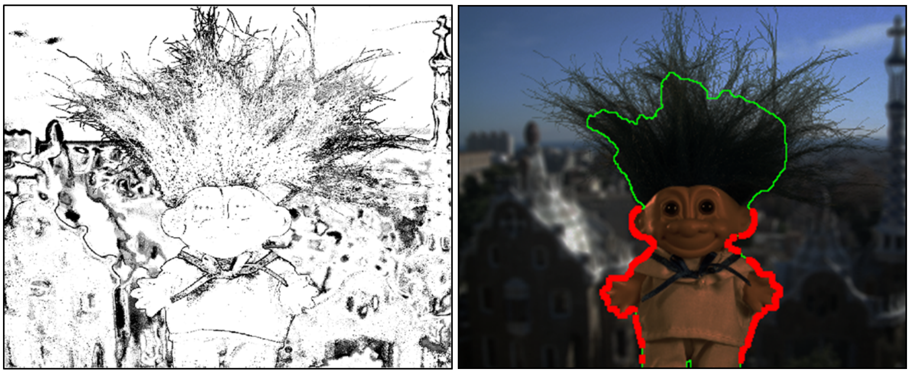
where c_i is the color of the input image at pixel i. Here $\theta_{\mathcal{F}}$ and $\theta_{\mathcal{B}}$ are the Gaussian Mixture Models (GMM) of fore- and background, respectively, that are initialized from the colors in $\mathcal{F}'$ and $\mathcal{B}'$, respectively.

In the unknown region, the color distribution of $D^c(x_i)$, with $x_i \in \mathcal{U}$, is represented by a third GMM θ_{GU} by blending all combinations of fore- and background mixtures of the respective GMMs as in [JK05]. This blending coefficient is correlated to the distribution of the alpha values over the image. Since the alpha values are likely to be distributed sparsely over the image (see chapter 3), we follow [WFZ02] and model the distribution of alpha values with a beta distribution whose two free parameters were derived as $(0.25, 0.25)$ from ground truth data. This is in contrast to [JK05], where the distribution of alpha (and



(a) Input image with scribbles

(b) Sub-blur kernel structures (unary energy)



(c) Color (unary energy)

(d) Physically sharp boundary detection

Figure 4.3: **Unary terms for trimap segmentation.** (a) Input image with user scribbles (red-foreground, blue-background). We show the unary energy for the sub-blur kernel structures term in (b) and the unary energy for the color term in (c). In (b,c) dark pixels indicates a low energy for $\mathcal{U}$ and white pixels a high energy. (d) Pixels in a small band around the $\mathcal{F}', \mathcal{B}'$ transition of GrabCut [RKB04] (green line) are classified into physically sharp boundary pixels indicated in bright red (the image was darkened for better visibility). The class prior is not visualized, since it is constant over the whole image.

therefore the distribution of the blending coefficient of the GMM) is modeled as linear. An example of the color energy computed for the input from figure 4.3(a) is shown in figure 4.3(c).

4.2.2 Class Prior

The class prior models the probability of a pixel to belong to the class $\mathcal{U}$ or $\bar{\mathcal{U}}$, which depends on the image. For instance, an image where the foreground object has been tightly cropped has a different proportion of $\mathcal{U}$ versus $\bar{\mathcal{U}}$ pixels than the original image. We model this ratio by a unary term as:

$$D^p(x_i) = \lambda\delta(x_i \neq \mathcal{U}) \; , \tag{4.2}$$

where δ is the Kronecker delta. Intuitively, a larger value for λ in eq. (4.2) gives a larger $\mathcal{U}$ region. We show that predicting λ during runtime improves the performance considerably. The learning of the predictor is discussed in section 4.3. Furthermore, the parameter λ (which corresponds to the size of the unknown trimap region) is also exposed to the user with a slider interface. To efficiently obtain the solution for *all* values of λ, we minimize our energy function in eq. (4.1) with the parametric maxflow technique [KBR07].

4.2.3 Sub-blur Kernel Structures

As we have demonstrated in chapter 3 there are different reasons for a pixel to have fractional alpha values (i.e. belong to $\mathcal{U}$). Here, we will assume that the main source of fractional alpha values is defocus blur. Thus our approach usually cannot detect fractional alpha values that are induced by motion blur or light-transmitting scene objects (e.g. a plastic bag). Our goal is to detect thin structures which have a width that is smaller than the size of the defocus blur kernel (e.g. the hair in figure 4.3(b)). Such "sub-blur kernel structures" are assumed to be originally opaque and show *solely* fractional alpha values after convolution with the defocus blur kernel. An example is given in figure 4.4 (top). It shows a 1-D example of a thin binary structure α^b on the left. It is convolved with the defocus blur kernel (here a 1-D averaging filter), which gives the alpha matte. Since the binary structure is smaller than the blur kernel, the resulting alpha matte comprises solely fractional alpha values (i.e. $\alpha < 1$).

In contrast, structures larger than the size of the blur kernel lead to "sharp boundaries".

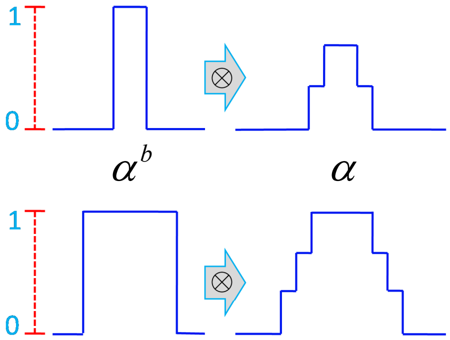


Figure 4.4: **Sub-blur kernel structures.** A 1-D example of a thin (top, left) and thick (bottom, left) binary structure (α^b). Convolution with the blur kernel delivers α (see text for details).

We can see an example in figure 4.4 (bottom). Blurring the spatially extended binary structure results in two smooth boundary transitions and the pixels between these transitions remain at an alpha value of 1. These transitions should ideally coincide with the boundaries of the binary segmentation into $\mathcal{F}'$ and $\mathcal{B}'$ (thus should be detected and handled by the sharp boundary term discussed in section 4.2.4).

In this section, we want to build a detector for the case where the structure is smaller than the size of the blur kernel. We have experimented with many different first and second order derivative filters and found the following to work best. Roughly speaking, the magnitude of the first derivative of a filter of size s should be low and at the same time the magnitude of the derivatives of the two $s/2$ sized filters, shifted by $s/2$, should be high. Here, s is twice the size of the blur kernel.We confirmed experimentally that $s = 5$ gives the lowest error rate over a training set with known ground truth alpha mattes.

Formally, we use $\max(0, |a-b| + |b-c| - |a-c|)$, where a, b, c are the left, center and right pixel values on a line segment of length 5. We made this (symmetric) detector orientation independent by taking the maximum response over four discrete angles $(0^o, 45^o, 90^o, 135^o)$. All three color channels were weighted equally. A further improvement was achieved by setting those filter responses to 0 for which not all pixels on the line segment were assigned to the same mixture in the GMM $\theta_{\mathcal{U}}$. The underlying assumption is that in a small window the true fore- and background colors are similar. We define D^s as the filter response. Figure 4.3(b) depicts a result for the image in figure 4.3(a). We found it to even work well for complicated interference patterns, e.g. many thin overlapping hairs. As desired, our filter has a lower response at sharp boundaries and inside the true $\mathcal{F}$ and $\mathcal{B}$ areas.

4.2.4 Sharp Boundary

The $\mathcal{F}', \mathcal{B}'$ transition determined by GrabCut [RKB04] often coincides with a clean, "sharp boundary". This means that in the vicinity of the detected boundary (defined by the spatial extent of the blur kernel) there is no other boundary transition that could lead to a sub-blur kernel structure. An example is the body (without hairs) of the object in figure 4.3(d). At such boundaries the width of $\mathcal{U}$ is equal to the width of the imaging system's PSF and thus is only a few pixels wide. To determine which parts of the $\mathcal{F}', \mathcal{B}'$ transition correspond to a sharp boundary, we have designed a simple classifier. We first derive an alpha matte with the matting algorithm of [LLW08] in a small band (twice the size of the blur kernel, centered on the $\mathcal{F}', \mathcal{B}'$ transition), which can be computed very efficiently. Then a pixel i in this band belongs to a sharp boundary if the following conditions hold for a small window w_i (twice larger than the blur kernel) centered on i:

1. The average alpha values of the pixels inside the window w_i are close to 0.5.
2. Half of the pixels inside w_i have an alpha value larger than 0.5.
3. At least half of the pixels in w_i are close to an alpha value of either 0 or 1.

Note that these conditions tolerate a shift of the boundary by half the size of the blur kernel. The classification error (percentage of misclassified pixels along the binary segmentation boundary) on our training set is 19.3%. Stronger conditions, which are computationally more expensive, could be considered in the future.

The result of this classifier is used to model the boundary terms D^b and V^b. Consider figure 4.5, which illustrates an in-focus region of a sharp boundary in a low-resolution alpha matte. The red line is the result of the binary segmentation into $\mathcal{F}'$ and $\mathcal{B}'$. We force all pixels adjacent to the $\mathcal{F}', \mathcal{B}'$ boundary to be in $\mathcal{U}$, which is encoded in D^b by using hard constraints (green pixels in figure 4.5). Also some pixels which are neighbors

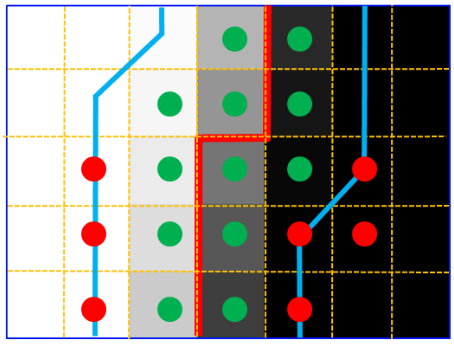


Figure 4.5: **Sharp boundary terms.** (See text for details.)

of green pixels are forced to be in $\bar{\mathcal{U}}$ (encoded in D^b by using hard constraints). These are those pixels which are close to a physically sharp boundary and are shown in red in figure 4.5. Intuitively, these pixels form a barrier which prevents the unknown region $\mathcal{U}$ from leaking out at sharp boundaries. The pairwise term V^b is shown as a blue line in figure 4.5. It forms an 8-connected path which follows tightly the green pixels. Intuitively, it enforces smoothness of this barrier. This means that it can close small gaps where our sharp boundary detector misclassified the $\mathcal{F}', \mathcal{B}'$ transition.

4.2.5 Smoothness Prior

The smoothness prior V^s encodes our assumption that pixels with similar colors should be assigned to the same label ($\mathcal{U}$ or $\bar{\mathcal{U}}$). Following [RKB04], the smoothness term is defined as

$$V^s(x_i, x_j) = \frac{\delta(x_i = x_j)}{\text{dist}\,(i, j)} \left(\theta_r \exp -\beta \left\|c_i - c_j\right\|^2\right), \tag{4.3}$$

where c_i and c_j are the colors at pixels i and j, respectively. The Kronecker delta is denoted by δ, and β is as defined in [RKB04]. The parameter θ_r is defined in section 4.3. As we show in section 4.4.2, our smoothness prior nicely preserves thin structures, e.g. hairs, inside the unknown region.

We also enforce the unknown region $\mathcal{U}$ to be 4-connected, which is true for 98.6% of the pixels in the ground truth database which we used to evaluate our algorithm. Since

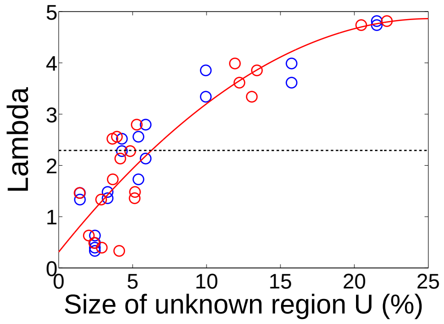


Figure 4.6: **Visualization of the correlation between λ and the size of the $\mathcal{U}$ region.** (See text for details).

minimizing our energy in eq. (4.1) under connectivity constraints was found to be $\mathcal{NP}$-hard by Vicente et al. [VKR08], we enforce connectivity by simple post-processing. In particular we remove all disconnected islands of $\mathcal{U}$.

4.3 Parameter Training

For training of the free parameters $\theta_b, \theta_{b'}, \theta_s, \theta_{s'}, \theta_r$ and λ, we have used the following heuristic error (loss) function, which counts false negatives twice compared to false positives:

$$error_{trimap} = \frac{100}{n} \sum_{i=1}^{n} 2 \cdot \delta(x_i^* = \mathcal{U} \wedge x_i \neq \mathcal{U}) + \delta(x_i^* \neq \mathcal{U} \wedge x_i = \mathcal{U}), \tag{4.4}$$

where δ is the Kronecker delta, x_i^* is the labeling of the ground truth trimap at pixel i and n is the number of pixels in the image. This is motivated by the fact that a missed unknown ($\mathcal{U}$) pixel in the trimap cannot be recovered during alpha matting. We see in section 4.4.2 that this error measure is correlated to the error for alpha matting. Based on our training dataset of 20 images with known ground truth (see section 4.4.1), we have hand-tuned the parameters $\{\theta_b, \theta_{b'}, \theta_s, \theta_{s'}, \theta_r\}$, defined in eqs. (4.1) and (4.3), to values of $\{2, 40, 1, 2, 0.1\}$. The training of the remaining parameter λ (defined in eq. (4.2)) is discussed in the following.

As we have seen in section 4.2.2, the parameter λ defined in eq. (4.2) is correlated to the size of the region $\mathcal{U}$. This correlation is visualized in figure 4.6, where the blue dots represent the optimal values for λ with respect to the size of the unknown region $\mathcal{U}$ of the ground truth trimaps of our training images. We see a correlation between the optimal values for λ and the size of the unknown trimap region. To exploit this correlation, we have built a predictor for the size of $\mathcal{U}$ (see below). The red dots in figure 4.6 show the optimal values for λ with respect to the size of the unknown region $\mathcal{U}$ that was predicted from our training images. The red curve is a quadratic function (3 parameters) fitted to the red dots. We see that the red (predicted) and blue (true) points are close-by. During runtime, the size of $\mathcal{U}$ is predicted using the test images, and the quadratic function provides the corresponding λ. The dashed line in figure 4.6 shows the average $\lambda = 2.3$, which was computed by averaging over all values *independently* of the size of $\mathcal{U}$. This straightforward averaging performs less well than predicting λ using our quadratic function, as we will see in section 4.4.2.

To predict the size of the unknown trimap region at runtime, we compute an initial trimap using the following heuristics. We use the data terms D^c, D^s, that are available after the user has placed the initial scribbles, and find the globally optimal trimap by simply thresholding this unary energy. On our set of training images with known ground truth, we have obtained an average prediction error for $\mathcal{U}$ of 1.5% relative to the image size.

4.4 Experimental Results and Evaluation

4.4.1 Test and Training Data

In order to quantitatively compare different interactive trimap extraction techniques, we first constructed a set of images with known ground truth alpha mattes denoted as α^*. We then obtained the ground truth trimap by partitioning the ground truth alpha matte α^* into a foreground region $\mathcal{F}$, background region $\mathcal{B}$ and unknown region $\mathcal{U}$ as $\mathcal{B} = \{i|\alpha_i^* < \epsilon\}$, $\mathcal{F} = \{i|\alpha_i^* > (1-\epsilon)\}$ and $\mathcal{U} = \mathcal{I}\backslash(\mathcal{F} \cup \mathcal{B})$, where we choose $\epsilon = \frac{5}{255}$.

To obtain natural images, which serve as input for the trimap segmentation algorithms, we recorded 27 images in a professional studio environment. These input images were obtained by photographing different foreground objects in front of a screen showing natural

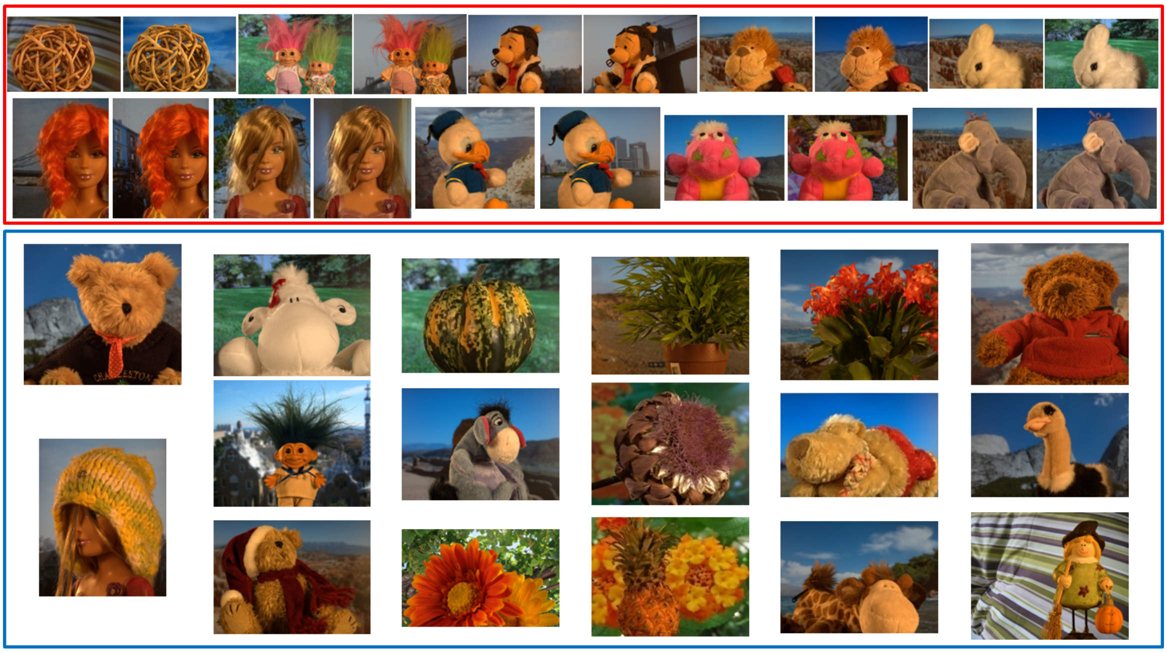


Figure 4.7: **Training and test images.** Our self-recorded training (images in red box) and test images (images in blue box).

images which served as background. The photographed objects have a variety of hard and soft boundaries and different boundary lengths, e.g. a tree with many holes (see figure 4.7 for an overview). For the input images, the ground truth alpha mattes were obtained with Triangulation Matting [SB96].

To increase the complexity of some input images, we replaced some of the backgrounds with more challenging, e.g. highly textured, backgrounds afterwards. These new compositions were simply generated by blending the ground truth foreground color with a new background using the ground truth alpha matte. The finally used background images show a varying degree of difficulty including color ambiguity of fore- and background and backgrounds with different degrees of blur. In total we employed in our tests a set of 10 training and 17 test images. An overview of the training and test images is depicted in figure 4.7.

Finally, we created for each image a set of potential user inputs by casually drawing large scribbles that intend to cover the major colors present in the image, while at the same time avoiding to draw the scribbles close to the object boundaries.

4.4.2 Comparison of Trimap Extraction Methods

In this section we compare our trimap extraction method to the method of Juan et al. [JK05]. In order to show that we also improve the quality of the final alpha matte, we further compare our approach to the 5 matting methods [WC07a, LLW08, LRAL08, GCL$^+$06] and [GSAW05][2]. Prior to applying the different methods, we first down-scaled our 6 Mpix input images to a size that most competing algorithms can handle, which was 0.3 Mpix (e.g. 700×560) - the limit of the publicly available system of [LLW08][3]. In addition we show results for the approach of [WC05] in figures 4.2 and 4.8. This approach had already been shown to be slightly outperformed by [GCL$^+$06], with the latter one being included in our test.

To quantitatively estimate the trimap quality, we use the metric defined in section 4.3, which measures the percentage of misclassified pixels with respect to the image size. To obtain a trimap error for systems which directly produce an alpha matte from the input scribbles, we transformed the resulting alpha matte into a trimap by thresholding. In order to derive an alpha matte from our computed trimaps, and those of Juan et al. [JK05], we use the matting approach of [WC07a].

The error for an alpha matte is defined as the following error function, which penalizes more heavily an over-estimation of alpha:

$$error_{alpha} = \frac{100}{n} \sum_{i=1}^{n} (1.5 \cdot \delta(\alpha_i \geq \alpha_i^*) + 0.5 \cdot \delta(\alpha_i < \alpha_i^*)) \cdot |\alpha_i - \alpha_i^*|, \tag{4.5}$$

where α_i and α_i^* denote the computed and the ground truth alpha matte at pixel i, respectively. The Kronecker delta is denoted as δ, and n is the number of pixels in $\mathcal{U}$.

Figures 4.8-4.11 compare the results of our trimap segmentation method with the results generated by the competing algorithms. We only show results of the best performing competing algorithms. The others were worse, both visually and in terms of error rates. For each method we show the computed trimaps and final composites. The results were

---

[2] We used the original authors' implementation of [WC07a, LLW08, LRAL08, GCL$^+$06, GSAW05] and our own implementation of [JK05].

[3] For [LRAL08] we had to even scale down the images to 0.15 Mpix. For comparison, the obtained result was then up-scaled to 0.3 Mpix images.

Method	av. error	worst 25%	time (seconds)
Grady et al. '05 [GSAW05]	24.3 (19.8)	33.6 (28.6)	5
Levin et al. '07 [LRAL08]	17.9 (9.5)	28.3 (17.8)	20
Guan et al. '06 [GCL$^+$06]	13.4 (9.0)	22.7 (16.5)	300
Levin et al. '06 [LLW08]	11.4 (6.9)	19.0 (13.3)	18
Wang et al. '07 [WC07a]	11.0 (8.4)	22.5 (19.0)	50
Juan et al. '05 [JK05]	7.6 (4.6)	13.8 (12.0)	5
Our (fixed $\lambda = 2.3$)	2.5 (1.2)	4.9 (2.3)	4.5
Our (predicted λ)	2.3 (1.0)	4.5 (1.9)	4.5
Our (user-tuned λ)	2.2 (0.7)	4.5 (1.5)	5.3
True trimap	0.0 (0.4)	0.0 (0.8)	-

Table 4.1: **Quantitative comparison of trimap methods.** We show the trimap error and corresponding alpha matte error in parentheses (definition of the error measures is given in the text). Depending on the columns, the numbers are averaged over all or the worst 25% of the test images. Times are in seconds and were measured on the same machine (2.2 GHz).

generated from the scribbles shown in the top row of (a) in the respective figure. An exception are the results shown in (e) of each figure. Here we adjusted λ and used extra scribbles (bottom images in (a)) to demonstrate the capability of our method to easily generate almost perfect results. In each figure, we see that our results outperform the competitors both visually and in terms of error rates. Further results are depicted in figure 4.1 and 4.2.

A quantitative comparison is shown in table 4.1. Let us first discuss the runtime performance of our algorithm in comparison to the other methods. We see that the matting systems [WC07a], [LLW08], [LRAL08] and [GCL$^+$06] are obviously considerably slower. Table 4.1 also shows that our method with the user-tuned value of λ takes on average 0.8 seconds longer to compute. This is because all solutions for the range of $\lambda \in [0, 5]$ have to be computed.

Let us now discuss the quality of the computed trimaps and alpha mattes. We see that our system clearly outperforms all other approaches both in terms of trimap errors and alpha matte errors[4]. We further see a correlation between the trimap- and alpha matte error, which motivates our heuristically defined error functions in eqs. (4.4) and (4.5). Also, the results

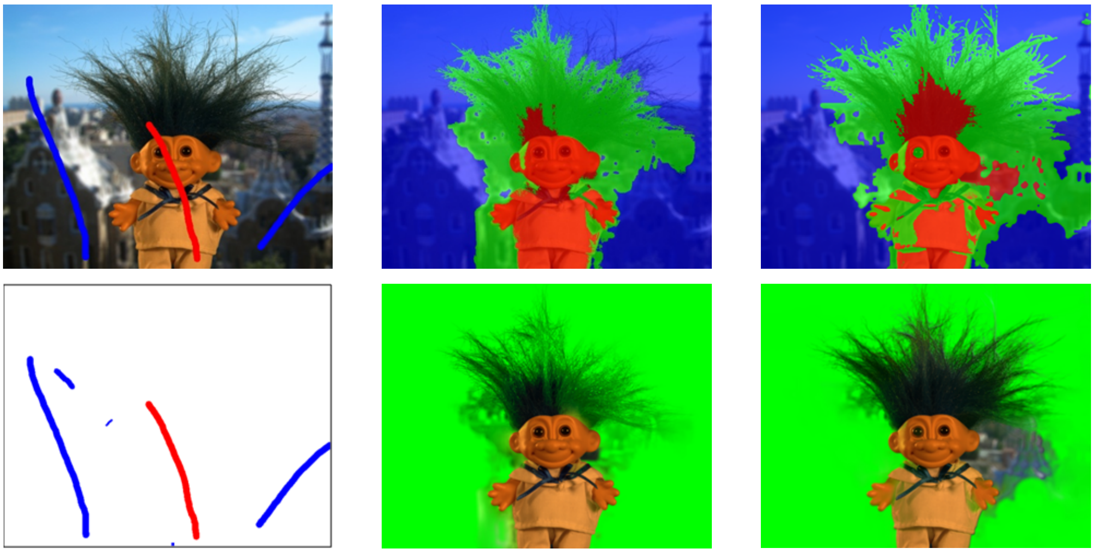
---

[4] The relatively low performance of [LRAL08] might be explained by the fact that this algorithm was not designed for a scribble-based interface, but a matting component picking interface.

confirm that predicting λ in our system works better than using a fixed λ. As expected, letting the user choose the λ value for each image, gives the best performance. Finally, by using the ground truth trimap to compute the alpha matte with the matting algorithm of [WC07a] (last row in table 4.1) gives by far the lowest alpha matte errors, which confirms that the problem of good trimap generation is vital for successful alpha matting.

4.5 Summary

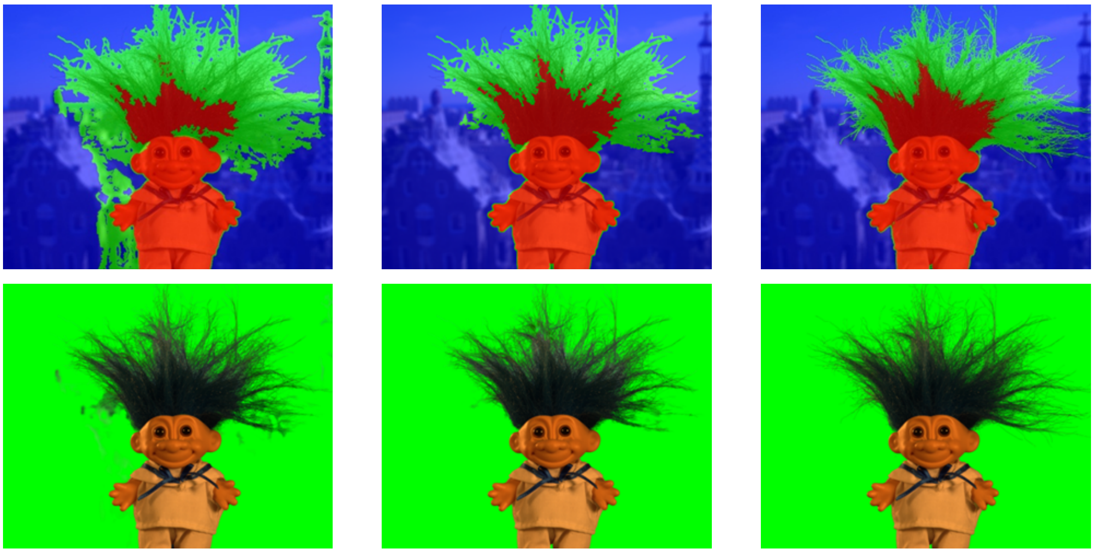
In this chapter we have presented a new method to obtain a trimap from only a few user-provided scribbles. The main contribution was a new energy function that resolves ambiguities in the trimap better than previous trimap extraction approaches. Our energy function considered several image cues that have drawn their motivation from the alpha formation process. For instance, we predicted the structure of the underlying binary segmentation, which indicates the spatial extent of the unknown trimap region. Furthermore, we used a large set of images with known ground truth alpha mattes to train a classifier that predicted the ratio of unknown pixels in the trimap from the input image. Our trimap extraction approach is fast, and we have confirmed that the quality of our computed trimaps and consequently alpha mattes improves on the state-of-the-art.



(a)
Top: 3 scribbles
Bottom: 6 scribbles

(b)
Closed-form Matting [LLW08]
(13.0;7.4)

(c)
Spectral Matting [LRAL08]
(11.3;12.5)

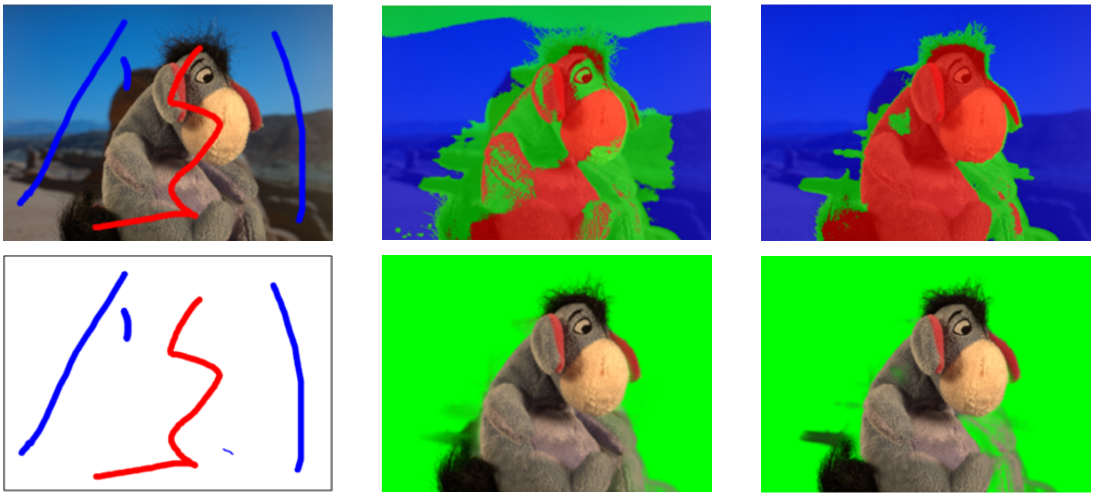


(d)
Our method
(predicted λ=3.5)
(5.9;2.0)

(e)
Our method
(user-tuned λ=3 & 6 scribbles)
(4.8;1.8)

(f)
Ground truth
(0.0;0.93)

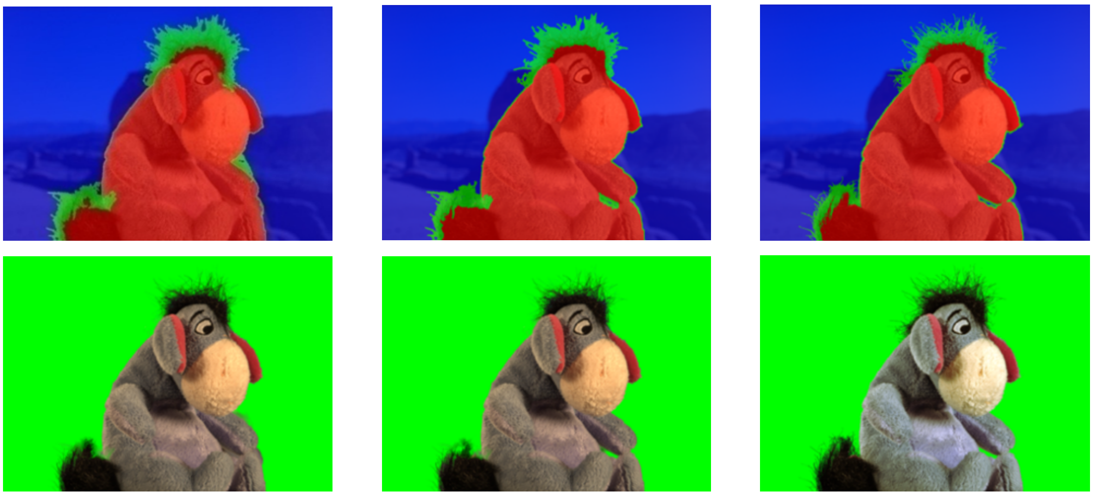
Figure 4.8: **Comparison of trimap methods (1).** This figure shows trimap segmentation results for a challenging image showing a soft toy with complex hair structure (a). For each result we show the trimap error and α matting error in parentheses.



(a)
Top: 4 scribbles
Bottom: 5 scribbles

(b)
Closed-form Matting [LLW08]
(6.1;3.5)

(c)
Robust Matting [WC07a]
(5.5;6.4)

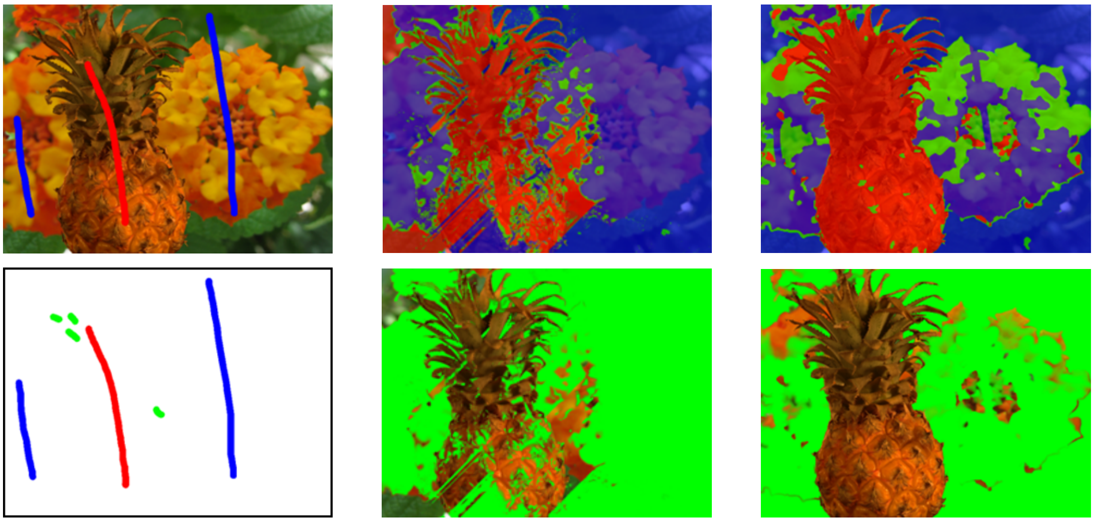


(d)
Our method
(predicted $\lambda = 2.2$)
(1.9;1.0)

(e)
Our method
(user-tuned λ=1.8 & 5 scribbles)
(1.8;0.7)

(f)
Ground truth
(0.0;0.0)

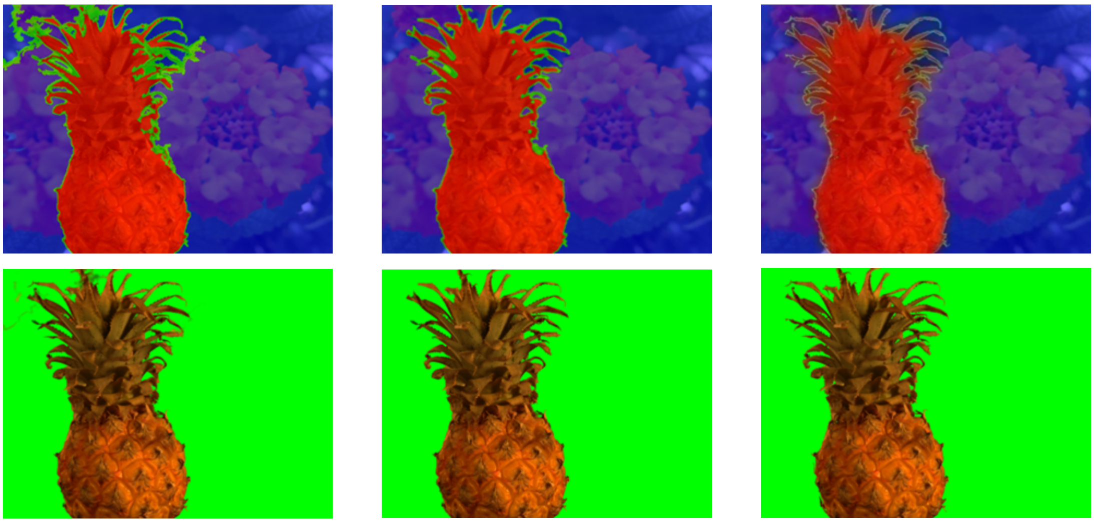
Figure 4.9: **Comparison of trimap methods (2).** Trimap segmentation results for an example with both simple and complex object boundaries. For each result we show the trimap error and α matting error in parentheses.



(a)
Top: 3 scribbles
Bottom: 7 scribbles

(b)
Easy Matting [GCL$^+$06]
(18.0;16.7)

(c)
Juan et al. [JK05]
(13.4;13.6)

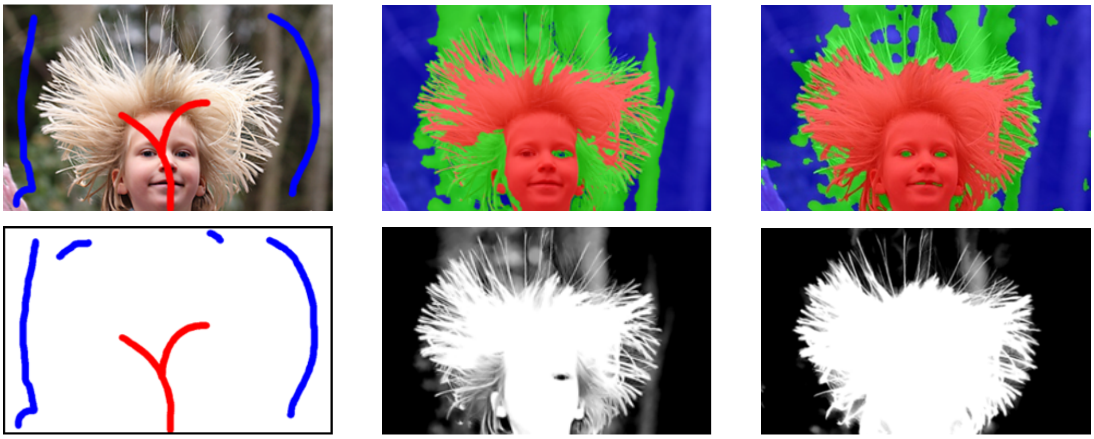


(d)
Our method
(predicted $\lambda = 1.3$)
(2.1;1.0)

(e)
Our method
(user-tuned λ=0.6 & 7 scribbles)
(1.2;0.6)

(f)
Ground truth
(0.0;0.0)

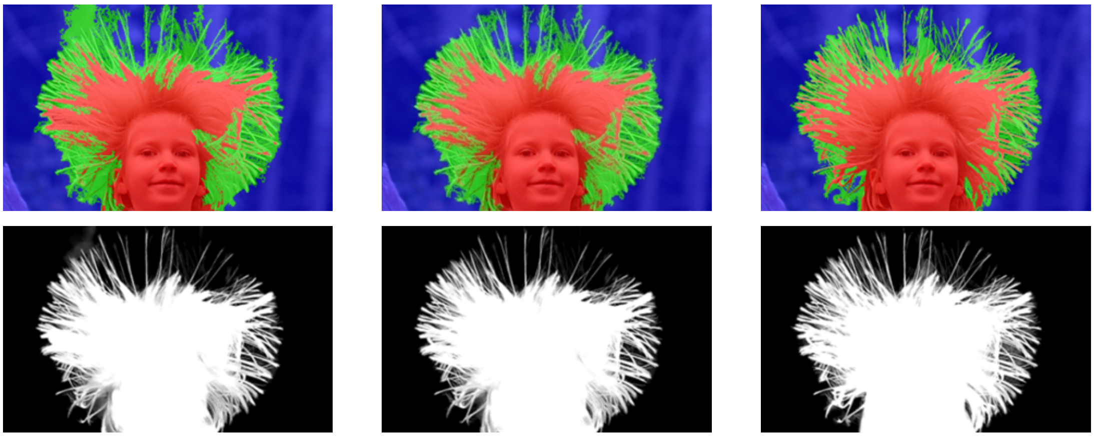
Figure 4.10: **Comparison of trimap methods (3).** Trimap segmentation results for examples with sharp boundaries and severely overlapping fore- and background color distributions. For each result we show the trimap error and α matting error in parentheses.



(a)
Top: 3 scribbles
Bottom: 5 scribbles

(b)
Robust Matting [WC07a]
(14.5;14.8)

(c)
Juan et al. [JK05]
(13.5;18.8)



(d)
Our method
(predicted $\lambda = 4.5$)
(10.1;2.8)

(e)
Our method
(user-tuned λ=5.8 & 5 scribbles)
(9.4;2.5)

(f)
Ground truth
(0.0;0.0)

Figure 4.11: **Comparison of trimap methods (4).** This figure shows trimap segmentation results for a difficult example used in the paper of [WC07a]. For each result we show the trimap error and α matting error in parentheses.

Chapter 5

Improved Color Modeling

In the previous chapter we have shown how to semi-automatically extract an accurate trimap with a small amount of user interaction. Given such a semi-automatically computed, or manually defined trimap, this chapter addresses the problem of extracting an alpha matte from a single photograph. More precisely, we concentrate on improving the color modeling step, in which for each pixel the fore- and background colors are estimated and then an optimal alpha value is computed for each pixel individually.

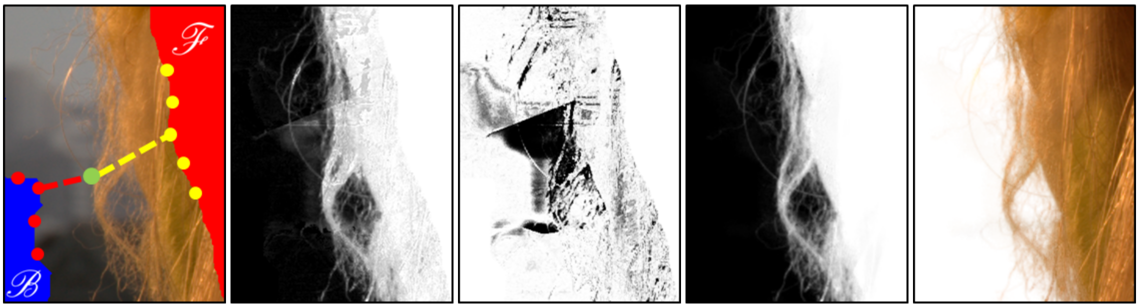
These alpha values form the data term that is combined with a smoothness term in an objective function. Minimizing this objective function yields the final alpha matte. We show that the alpha mattes obtained with our approach are superior to those computed with previous approaches.

5.1 Overview

Our approach builds on the "Robust Matting" approach of Wang and Cohen [WC07a] (see chapter 2.2 for details) and splits the color modeling task into two successive steps:

1. Collecting candidate color samples.
2. Selecting the best samples from the candidate set.

In the following, we first give an overview of the two different steps of our color modeling approach. Then both steps are explained in detail in section 5.2 and 5.3.



(a) Image crop with trimap (b) Pixel-wise alpha values (c) Confidence of alpha (Bright pixels indicate high confidence) (d) Final alpha (e) Composite

Figure 5.1: **Overview of our approach**. From an image crop (a), color samples are collected which give rise to an independently estimated alpha value at each pixel (b), together with its confidence (c). This forms the data term of our objective function. Combined with the smoothness term of [LLW08], it produces the final alpha matte (d). This alpha matte and corresponding foreground colors can be used to generate a new image composition as in (e).

Let us consider the image crop in figure 5.1(a). The user-defined trimap is indicated by the red (foreground $\mathcal{F}$) and blue (background $\mathcal{B}$) regions. For each pixel in the unknown region of the trimap, e.g. the green pixel in figure 5.1(a), we first gather a number of potential fore- and background color samples from the $\mathcal{F}$ and $\mathcal{B}$ regions. This is done by spreading the samples along the boundaries of the respective regions (red dots indicate background and yellow dots foreground samples). While previous approaches (e.g. [WC07a]) spread samples in an area which is *spatially* close to the green pixel, we use a spreading area which is close in *geodesic* space (see section 5.2).

In the next step a confidence value is computed for all possible pairs of sampled fore- and background colors. The confidence value reflects, among others, how well the sampled colors explain the mixed color of the pixel under consideration (i.e. fit the compositing equation (1.1)). A novel paradigm to compute the confidence value is presented in section 5.3. Then the corresponding sample pair with the highest confidence is selected for each pixel. Figure 5.1(b) shows the α values, computed from the selected sample pairs using eq. (1.1), with the corresponding confidence given in figure 5.1(c). This pixel-wise computed alpha matte is the data term of our objective function.

The confidence values in figure 5.1(c) are used to weight the data term with respect to the smoothness term of our objective function.[1] We use the smoothness term of [LLW08] and minimize the objective function by solving a sparse set of linear equations yielding a final α matte shown in figure 5.1(d). As we can see, the propagation removed many artifacts of the pixel-wise α in figure 5.1(b). The composition onto a white background (figure 5.1(e)) shows that fine details of the hair are nicely preserved.

5.2 Collecting Candidate Samples

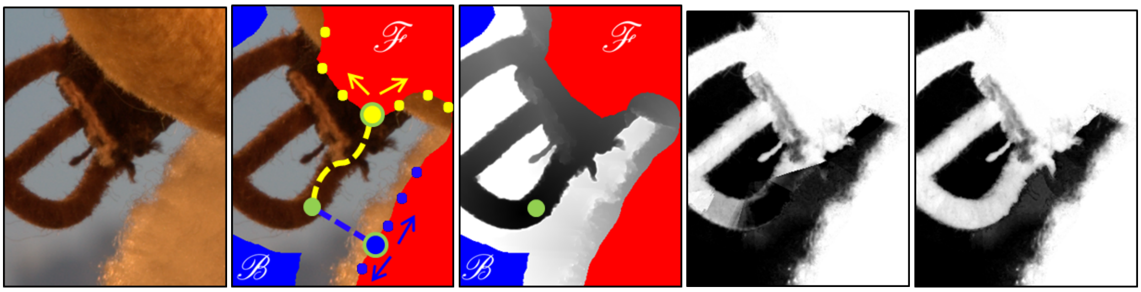
Let $\mathcal{I}$ be the set of all pixels in an image and let the subsets $\mathcal{F}$, $\mathcal{B}$ and $\mathcal{U}$ define the foreground, background and unknown region of the user-defined trimap. For each pixel $z \in \mathcal{U}$ we first collect a sample set of N (we use $N = 30$ in our implementation) fore- and background color samples: $\mathbf{F}_z = (F_z^1, ..., F_z^i, ..., F_z^N)$, $\mathbf{B}_z = (B_z^1, ..., B_z^j, ..., B_z^N)$, from $\mathcal{F}$ and $\mathcal{B}$, that are used to reason about the true fore- and background color at pixel z.[2] (For simplicity, we omit the subscript z if only a single pixel is under consideration).

Most previous approaches (e.g. [CCSS01, RT00]) reason about the fore- and background colors by fitting a parametric model to the sampled colors, e.g. a Gaussian Mixture Model. The key insight of [WC07a] was that better results can be achieved by simply selecting the "best" samples (defined in section 5.3) from the initial set. This circumvents a potential poor fit of the low-dimensional parametric model, and adds robustness with respect to outliers. The basic assumption of this approach is that the true fore- and background colors for every mixed pixel are present in the sample sets (or that the true colors are at least very close to the colors in the sample sets). This makes the collection of color samples a crucial part of the overall algorithm. In order to capture a large variation of colors, [WC07a] suggested to spread the samples along the boundaries of the known fore- and background regions (instead of collecting the spatially closest pixels from the fore- and background trimap regions). Let us improve on this idea.

---

[1] An important difference to [WC07a] is that they use a constant weighting of the data term, and in their approach the confidence value is used to bias pixels with high uncertainty towards an alpha value of zero or one. We show experimentally that our approach is superior.

[2] We use calligraphic letters for a set of pixels, e.g. $\mathcal{I}$, and bold letters for a set of color samples, e.g. $\mathbf{F}$.



(a) Image crop (b) Trimap with samples (c) Geodesic distance to green pixel (d) Pixel-wise alpha using blue samples (e) Pixel-wise alpha using yellow samples

Figure 5.2: **Collecting color samples** (detailed description in text). Collecting foreground candidate samples for image crop (a). (b) Trimap with "geodesic samples" in yellow and spatial samples in blue. The result with spatial samples (d) is worse than with geodesic samples (e).

Let us consider the image crop in figure 5.2(a) of a buckle which is part of a soft toy. Assume we aim to find a good set of foreground colors **F** for the green pixel in figure 5.2(b). A simple approach to gather **F** is to start spreading the sample set from the spatially nearest pixel in $\mathcal{F}$ (bold blue dot in figure 5.2(b)).[3] Unfortunately, this sample set includes only bright colors, which do not match the true foreground color (i.e. dark brown) of the pixel marked in green. Thus, this simple sampling scheme results in a poor estimation of α (figure 5.2(d)). Our basic idea is to improve the search for a suitable foreground color by assuming the foreground object to be spatially connected (a common assumption - see e.g. [VKR08]). The yellow path in figure 5.2(b) goes from the unknown pixel (marked green) to the bold yellow dot in $\mathcal{F}$ and passes solely through pixels that are very likely to belong to the foreground object. The bold yellow dot defines a better starting point to spread the foreground samples, since the sample set comprises colors that are similar to the true foreground color. This motivates to spread the sample set from the closest pixel in geodesic distance (figure 5.2(c)), which respects the shape of the foreground object and gives better results (figure 5.2(e)).

The geodesic distance is defined as the shortest path on a weighted graph from a given

---

[3] We believe that a similar method was used in [WC07a], although no details were given in the respective paper.

pixel $z \in \mathcal{U}$ to the foreground region $\mathcal{F}$ of the trimap. Similar to [BS07] we choose the weights of the edges to be the gradient of the likelihood, i.e. $\nabla P_{\mathcal{F}}(z)$. The likelihood $P_{\mathcal{F}}(z)$, for a pixel z to belong to $\mathcal{F}$, is obtained from the user-provided trimap as in [BS07]:

$$P_{\mathcal{F}}(z) = p(C_z|\theta_{\mathcal{F}}) / \left(p(C_z|\theta_{\mathcal{F}}) + p(C_z|\theta_{\mathcal{B}})\right), \tag{5.1}$$

where $p(C_z|\theta_{\mathcal{F}})$ is the probability that the color C, at pixel z, was generated by the Gaussian Mixture Model $\theta_{\mathcal{F}}$ of the foreground, which is constructed from all pixels in $\mathcal{F}$. The probability $p(C_z|\theta_{\mathcal{B}})$ is computed likewise. Note that the color models could also be built from local windows placed over the unknown region (similar to [CCSS01, RT00]). However, in practice we did not find a window-based approach to improve results.

To collect candidate samples for the background, we use the same approach as [WC07a], i.e. spread the sample set from the spatially nearest pixel in $\mathcal{B}$. This is based on the fact that the background region is usually not connected, due to occlusion by some foreground parts. We have seen experimentally that the performance can be improved even further by combining the “geodesic samples” with the samples of the spatially closest area. The reason could be that the likelihood $P_{\mathcal{F}}$ is not necessarily always perfect. Hence we gather in total 60 samples in each set $\mathbf{B}_z$ and $\mathbf{F}_z$ for every pixel z. In practice about $40 - 50\%$ of our “geodesic samples” contribute to the optimal sample pairs.

5.3 Selecting Best Candidate Samples

Given a candidate set of fore- and background colors ($\mathbf{F}_z$ and $\mathbf{B}_z$) for each pixel $z \in \mathcal{U}$ with color C_z, we first introduce our approach to compute the confidence for all sample pairs (F_z^i, B_z^j) from this initial set. Confident sample pairs should meet three criteria: (i) F^i and B^j should fit the linear model in eq. (1.1) (i.e. the mixed color C should lie on the line segment, in color space, spanned by F^i and B^j); (ii) F^i and B^j should be widely separated in color space, to allow for a robust estimation of α using the compositing eq. (1.1); (iii) Assuming that the alpha value of most pixels in the image is likely to be either 0 or 1, F^i or B^j are likely to be close in color space to C. (This is a reasonable assumption based on the alpha formation process described in chapter 3.)

Following [WC07a], we encode (i) and (ii) in a *distance ratio* $R(F^i, B^j)$ as

$$R(F^i, B^j) = \frac{\|C - (\hat{\alpha} F^i + (1 - \hat{\alpha}) B^j)\|}{\|F^i - B^j\|}, \tag{5.2}$$

where $\hat{\alpha}$ is estimated by projecting the observed color C onto the line spanned by the sample pair (F^i, B^j) under consideration. The numerator in eq. (5.2) represents the linear fit to the model, i.e. criterion (i), while the denominator encodes robustness (criterion (ii)).

For criterion (iii) we define two weights $w(F^i)$ and $w(B^j)$ that encourage individual fore- and background samples to be similar to color C. It should be noted that this approach is different to [WC07a], where two weights were defined that avoid samples which are similar to color C. In our approach, we define the two weights $w(F^i)$ and $w(B^j)$ as:

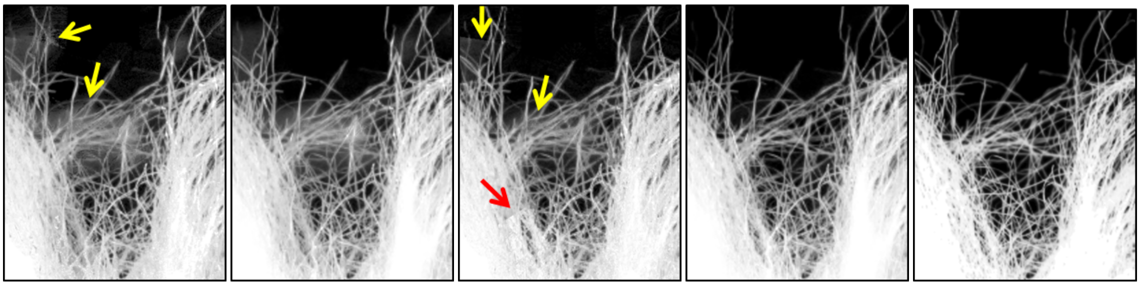
$$\begin{aligned} w(F^i) &= exp\left\{- \max_{s \in \{1,..,N\}} \left(\|F^s - C\|^2\right) / \left\|F^i - C\right\|^2\right\} \\ w(B^j) &= exp\left\{- \max_{s \in \{1,..,N\}} \left(\|B^s - C\|^2\right) / \left\|B^j - C\right\|^2\right\}, \end{aligned} \tag{5.3}$$

where N is the number of fore- and background samples, respectively. The function $\max_s$ returns the maximum squared difference between the mixed color C and all corresponding fore- or background color samples, respectively. Finally, a confidence value f for each sample pair is computed, as in [WC07a], by combining eqs. (5.2) and (5.3) to

$$f(F^i, B^j) = exp\left\{-\frac{R(F^i, B^j)^2 \cdot w(F^i) \cdot w(B^j)}{\sigma^2}\right\}, \tag{5.4}$$

where σ is set to 0.1. The confidence $f(F^i, B^j)$ is large if the distance ratio R is low or if the samples F^i or B^j are similar to color C. This is in contrast to [WC07a] where samples close to the mixed color C are assigned to a *low* confidence value and biased towards 0 or 1 in a later step.

We compute then a confidence value for each sample pair, and the pair with the highest confidence $\hat{f} = max_{i,j} \left(f\left(F^i, B^j\right)\right)$ is selected to obtain a pixel-wise estimation of α, denoted as $\hat{\alpha}$. Note, in practice it is computationally too expensive to evaluate eq. (5.4) for all 3600 pairs of samples. Therefore we prune each sample set from 60 to 15 using criterion (iii), i.e. by selecting those samples which are closest in color to the mixed color C. Hence, we obtain only 225 sample pairs. In our tests, this gave virtually no drop in performance.



(a) Pixel-wise alpha using [WC07a]. Arrows mark artifacts.
(b) Final alpha using [WC07a].
(c) Pixel-wise alpha of our method. Arrows mark artifacts.
(d) Final alpha using our method.
(e) Ground truth

Figure 5.3: **Sample selection** (detailed description in text). The pixel-wise estimation of alpha, based on the selected color samples, has fewer artifacts with our approach (c) than with [WC07a] (a). Given the data term, our final alpha matte (d) is close to the ground truth (e), while many artifacts remain in the final alpha matte of [WC07a] (b).

Figure 5.3(a) shows the pixel-wise computed matte obtained with the method of [WC07a], which contains considerable blurry artifacts and is of lower quality than the initial matte obtained with our approach (figure 5.3(c)).

In the next step, we use the pixel-wise estimated $\hat{\alpha}$ and its confidence $\hat{f}$ to define the data term. We use a quadratic function with the minimum at $\hat{\alpha}$. The data term is then combined with a smoothness term. Here, we use the Matting Laplacian L of [LLW08]. The complete objective function J is

$$J(\alpha) = \alpha^T L \alpha + (\alpha - \hat{\alpha})^T \hat{\Gamma} (\alpha - \hat{\alpha}), \tag{5.5}$$

where α and $\hat{\alpha}$ are treated as column vectors. The first term of eq. (5.5), $\alpha^T L \alpha$, defines the smoothness term and the second term is the data term. The diagonal matrix $\hat{\Gamma}$ defines the weighting between data and smoothness term. In contrast to [WC07a], where a constant weighting of the data term is used (i.e. the diagonal elements of $\hat{\Gamma}$ are set to a constant), we regulate each diagonal element $\hat{\gamma}_z$ of $\hat{\Gamma}$ with the confidence $\hat{f}_z$ of the pixel-wise estimated $\hat{\alpha_z}$: $\hat{\gamma_z} = \gamma \cdot \hat{f}_z$, where γ is a constant (we use 10^{-3} in our implementation). Thus our approach relies more on propagation (provided by the smoothness term) in low confidence regions. In order to deal with high-resolution (e.g. 6 Mpix) images, we solve the sparse linear system in a multi-resolution framework to obtain α mattes with reasonable time and memory consumption.

Figures 5.3(b) and (d) compare the final result of [WC07a] to our approach. We see that our result is close to the ground truth (figure 5.3(e)), while considerable blurry artifacts remain in the result of [WC07a], e.g. visible in the middle of figure 5.3(b).

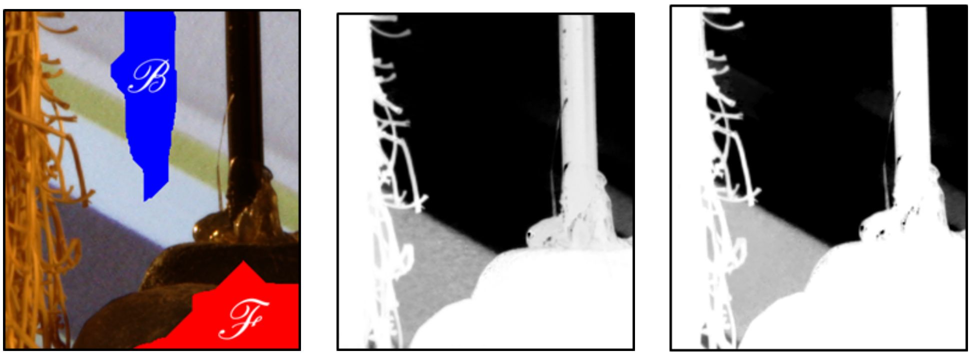
5.4 Experimental Results

To test our approach, we computed alpha mattes on a variety of different images and trimaps. In this section we present qualitative results that demonstrate the good performance of our algorithm. A quantitative comparison of our algorithm to the state-of-the-art in presented in chapter 7.

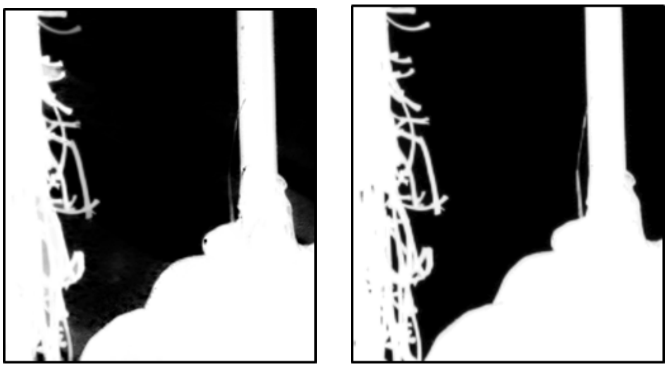
Figure 5.4 compares the results of our approach to two previous approaches. The alpha mattes were computed on an image crop, showing part of a solid toy and a fuzzy broom (figure 5.4(a)). The result of [LLW08] shows large semi-transparent regions, especially in the background (figure 5.4(b)). Similarly, the approach of [WC07a] has problems to correctly recover the background (figure 5.4(c)). In the result computed with our approach (figure 5.4(d)), most artifacts in the background are eliminated and the fuzzy broom was recovered well. The result is very close to the ground truth (e).

A second example is given in figure 5.5, which shows the crop of an artificial flower. We see that, similar to the example in figure 5.4, the methods of [WC07a] and [LLW08] generate large artifacts in the background (see figure 5.5(b,c)). Again, our approach (figure 5.5(d)) generates a more clear alpha matte and is close to the ground truth in figure 5.5(e).

Finally, we depict a very challenging example showing fuzzy hair in figure 5.6. We see that the approach of [WC07a] could not correctly reconstruct the hair (figure 5.6(b)), which is presumably due the color ambiguities. The method of [LLW08] (figure 5.6(c)) better recovers the hair, although our approach delivers slightly better results (see figure 5.6(d)).



(a) Input image with trimap (b) Closed-form Matting [LLW08] (c) Robust Matting [WC07a]

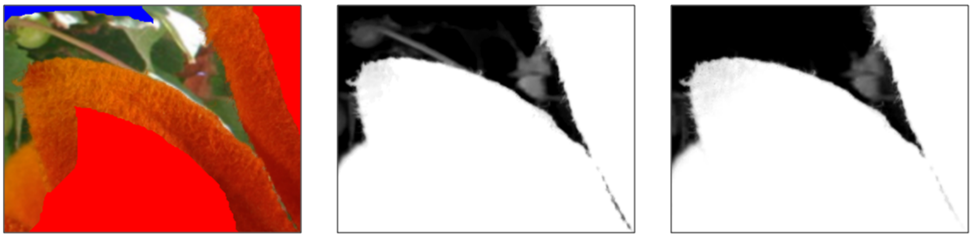


(d) Our final result (e) Ground truth

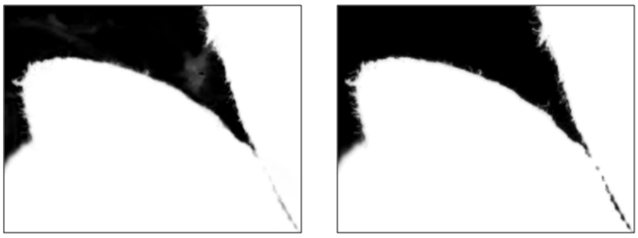
Figure 5.4: **Qualitative comparison (1).** Matting results for the input image in (a) are depicted in (b-d). See the text for a detailed discussion.

5.5 Summary

In this chapter we have presented a new approach to color modeling which relies on information from global color models to find better local estimates of the true fore- and background colors. In particular, we first gathered a number of potential fore- and background color samples from user-constrained regions which are close in geodesic space. This is in contrast to previous approaches which simply collect samples from spatially nearby regions. Furthermore, we have presented a new paradigm to compute a confidence value for the color samples which is motivated by the alpha formation process. Finally, we computed a pixel-wise alpha matte from the color samples with the highest confidence. This matte defines the data term of an objective function which is minimized to obtain the final alpha

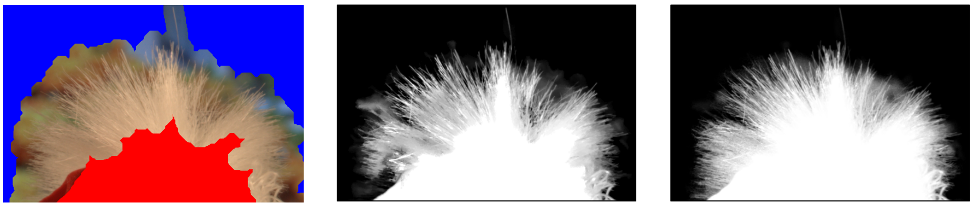


(a) Input image with trimap (b) Robust Matting [WC07a] (c) Closed-form Matting [LLW08]

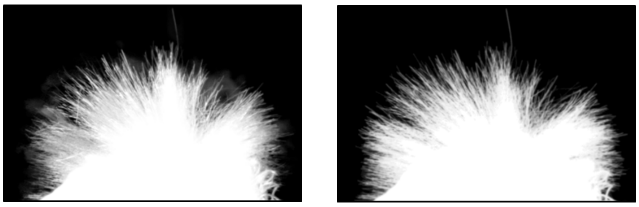


(d) Our approach (e) Ground truth

Figure 5.5: **Qualitative comparison (2).** Matting results for the input image in (a) are depicted in (b-d). See the text for a detailed discussion.



(a) Input image with trimap (b) Robust Matting [WC07a] (c) Closed-form Matting [LLW08]



(d) Our approach (e) Ground truth

Figure 5.6: **Qualitative comparison (3).** Matting results for the input image in (a) are depicted in (b-d). See the text for a detailed discussion.

matte. Finally, we have validated the good performance of our approach with compelling examples.

Chapter 6

Segmentation-based Prior for Matting

In this chapter, we aim to recover the alpha matte based on the assumption that in real world images, fractional alpha values are often induced during the imaging process, in particular, caused by the camera's Point Spread Function (PSF). If this assumption is met, we have seen in chapter 3 that one can model the prior distribution of the alpha matte α as a convolution of an underlying, potentially higher resolution, binary segmentation α^b with a kernel K that models the PSF. The result of this convolution may be downsampled afterwards by a function D:

$$\alpha = D(K \otimes \alpha^b). \tag{6.1}$$

In this chapter we model the kernel K as spatially constant, which can account for defocus blur in the presence of a large depth of field. It should be noted, however, that our framework could be extended with spatially varying blur kernels in the future.

To construct the prior, the key challenge is to solve the blind deconvolution problem, which is the reconstruction of the binary segmentation α^b and kernel K in eq. (6.1) from an input alpha matte α. (A deconvolution approach is commonly denoted as blind, if the kernel K is unknown, and denoted as non-blind otherwise.)

We will present two new approaches for the deconvolution of alpha mattes in sections 6.2 and 6.3. Our methods assume that the PSF is a kernel with a single peak, which is usually true for optical blur or very slight motion blur (a limitation is complex motion blur). If our assumption is met, it has been shown by Joshi et al. [JSK08] that the binary segmentation can be recovered from the edges in the blurred alpha matte. In this work, we infer the binary mask α^b and, consequently, the kernel K with new segmentation techniques from the initial alpha matte α. In our approaches, we compute this initial matte using the Improved Color Matting algorithm presented in chapter 5.

Convolving the recovered binary segmentation α^b with the PSF K gives a new alpha matte that is typically of high-quality. However, to account for potential artifacts in the matte, we use this convolved segmentation as prior in the Improved Color Matting method. The result is an alpha matte whose quality usually exceeds the current state-of-the-art, as we show in sections 6.2.8 and 6.3.7.

The remainder of this chapter is organized as follows. First, in section 6.1, we briefly review previous deconvolution approaches that can be used to solve for α^b and K, given a input alpha matte α. In sections 6.2 and 6.3 we introduce two new approaches for the deconvolution of alpha mattes and use the resulting binary segmentation and PSF as prior for image matting. Finally, section 6.4 demonstrates qualitatively and quantitatively that the model in eq. (6.1) is indeed a good prior for many alpha mattes of real images.

6.1 Deconvolution of Alpha Mattes - Related Work

Recovering the binary segmentation α^b and blur kernel K from an alpha matte is the task of blind deconvolution, and we discuss related work in the following. In this section we use the ground truth alpha matte α^* for comparing deconvolution methods. However, for the matting approaches described in most parts of sections 6.2 and 6.3, we use an alpha matte computed from the input image with the Improved Color Matting algorithm described in chapter 5. To ensure that the underlying segmentation α^b is more likely to be binary, in this test we upscaled α^* by a factor of 3 before applying the methods discussed below.

In theory one should be able to perfectly reconstruct α^b by deconvolution algorithms, given the true alpha matte α^* and the true blur kernel K^*, respectively. (We also confirmed this in a synthetic experiment.) However, in practice we found the results obtained with state-of-the-art (blind) deconvolution approaches[1] to be inappropriate for our purposes.

---

[1] We experimented with the non-blind deconvolution algorithms of [LFDF07, SJA08] and the blind method

More specifically, we observed that the deconvolved alpha mattes were usually far away from being binary. This empirical observation was recently confirmed in the work of Levin et al. [LWDF09] which shows that the simultaneous Maximum A Posteriori (MAP) estimation of both K and α^b mostly favors a solution where K is the delta kernel. To overcome this problem, Levin et al. [LWDF09] suggested to first estimate the PSF using the approach of [FSH$^+$06] and then perform (non-blind) deconvolution using [LFDF07]. We tested this approach, using the authors' implementations, but unfortunately the results were still non-binary. Hence, to obtain α^b we had to threshold the deconvolution results, which resulted in the loss of many details like hair strands. Since [FSH$^+$06] was mainly designed for large motion blur, we also used [JSK08] to initialize the PSF for [LFDF07], but found it to give non-binary results as well.

A possible explanation for this failure is that state-of-the-art deblurring approaches are based on natural image statistic priors that are not applicable to alpha mattes. In particular, the desired deblurred alpha matte is a two-tone image, thus has a much simpler structure than a natural image. Experiments in Levin et al. [LWDF09] suggest that a prior which favors two-tone images could potentially overcome the undesired solution where K is the delta kernel. Therefore, one could follow the approach of Jia [Jia07] and incorporate in the deconvolution process the assumption that the unblurred alpha matte is binary. The result of this approach[2] is shown in figure 6.1(b) for the ground truth alpha matte in figure 6.1(a). Unfortunately many fine details were lost.

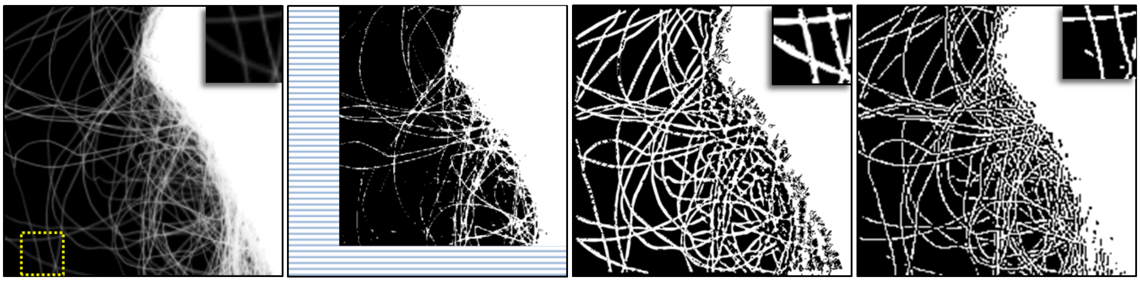
Another class of deconvolution approaches explicitly detect edges in the image to infer a binary segmentation. For instance, the recent approach by Joshi et al. [JSK08] detects the location of step edges in the (unknown) sharp image by applying a sub-pixel-accurate edge detector to the blurred image. If the deblurred image is two-toned (which is true for alpha mattes), the location and orientation of the sharp image edges is sufficient to infer α^b around the detected edges. We found this method to perform reasonably well on solid boundaries, but it severely over-estimated α^b in the presence of thin structures like hair strands, due to incorrect edge localization (see, e.g., figure 6.1(c)).

A straightforward approach to obtain α^b is to run binary segmentation methods like

---

of [SJA08]. When using [LFDF07, SJA08] we initialized K using [JSK08].

[2] The author of [Jia07] kindly applied his method on a crop of a ground truth matte of our test set.



(a) Ground truth alpha (b) α^b from [Jia07] (in high-resolution) (crop of (a) due to memory limits) (c) α^b from [JSK08] (in high-resolution) (d) α^b using our Hybrid Deconvolution approach (in low-resolution)

Figure 6.1: **Deconvolution with binary prior.** Deconvolution of the ground truth alpha (a) with the approach of [Jia07] results in the loss of many details like hair strands (b). (c) The deconvolution result of [JSK08] preserves most structures, but the segmentation of the hair is too wide. Our Hybrid Deconvolution approach (d) preserves thin structures and better recovers the width of thin structures. For a better visualization, we show a zoom-in of the yellow-marked region in (a) in the upper right corners of the result in (a), (c) and (d).

GrabCut [RKB04] on the image. However, current state-of-the-art segmentation methods oftentimes cannot recover very fine structures like hairs. Thus one could follow the approach described in chapter 4 and classify the segmentation borderline into sharp and soft boundaries. In the vicinity of a sharp boundary, fractional alpha values are likely to occur only in a small band around such a boundary (given by the width of the PSF) and pixels adjacent to this small band should be pushed towards an alpha value of 0 or 1. We tested this approach and found it to work well close to sharp boundaries, but clearly it did not work well if the foreground object exhibited fine soft-boundary structures like hair strands.

6.2 Hybrid Deconvolution Approach

As discussed in the previous section, the binary segmentation and the PSF may be derived using deconvolution approaches from an input alpha matte. In this section we propose a new Hybrid Deconvolution method that splits the task into two steps. We first deblur the alpha matte with the deconvolution approach by Levin et al. [LFDF07]. As we have observed in the previous section, the resulting deblurred alpha matte α^d is oftentimes not binary. Therefore, in the second step, we infer a binary segmentation α^b from the deconvolved alpha matte α^d with a new segmentation technique that was designed to preserve fine structures like hair strands in the segmentation result. Figure 6.1(d) shows α^b obtained with our method from the ground truth alpha matte in figure 6.1(a). We see that, in contrast to the methods proposed by Jia [Jia07] and Joshi et al. [JSK08], which infer the binary mask in a single step, our two-step process could better preserve the fine details.

6.2.1 Overview of our Matting Approach

Our matting approach comprises five steps: (i) Given an input image and trimap, compute an initial (usually) imperfect alpha matte α, with the Improved Color Matting approach presented in chapter 5; (ii) estimate the PSF from α; (iii) use the PSF to deblur α with the method by Levin et al. [LFDF07] to obtain a deconvolved (but usually non-binary) alpha matte α^d; (iv) estimate the binary segmentation α^b from the deconvolved α^d while preserving edges; (v) convolve the binary segmentation α^b with the PSF and use it to re-estimate the alpha matte α. We now describe each step in detail.

6.2.2 Estimating the Initial Alpha Matte

We have seen in section 6.1 that the binary segmentation may be derived with deconvolution methods from the ground truth alpha matte. To apply our approach to natural images where the ground truth is unknown, we infer the binary segmentation and PSF from an alpha matte computed from the natural image with a conventional matting algorithm. In particular, we use our Improved Color Matting approach, that was detailed in chapter 5. As shown in chapter 5, the Improved Color Matting algorithm first computes a pixel-wise estimation of alpha denoted as $\hat{\alpha}$, which defines the data term. The data term is then combined with the smoothness term of Levin et al. [LLW08], giving the following objective function J:

$$J(\alpha) = \alpha^T L\alpha + (\alpha - \hat{\alpha})^T \hat{\Gamma}(\alpha - \hat{\alpha}), \tag{6.2}$$

where α and $\hat{\alpha}$ are treated as column vectors and L is the Matting Laplacian of [LLW08].

The diagonal matrix $\hat{\Gamma}$ weights the data against the smoothness term. The objective function is minimized by solving a set of sparse linear equations, subject to the user defined input constraints, which gives the initial alpha matte α.

The initial matte computed for the image crop and trimap in figure 6.2(a) is shown in figure 6.2(b).

6.2.3 Estimating the PSF

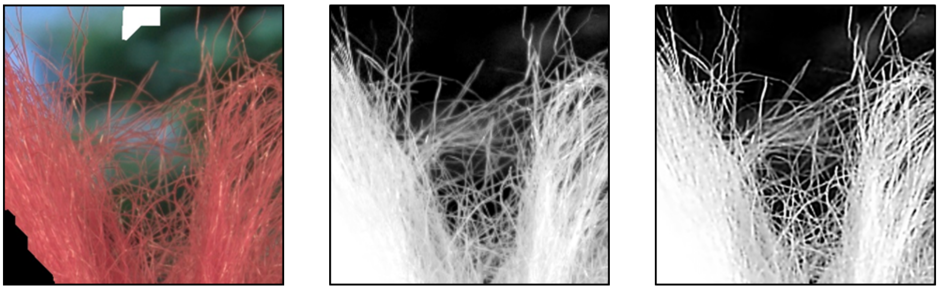
We model the PSF as a symmetric kernel K of size 9×9 pixels with non-negative elements which sum up to 1. Furthermore, we assume the PSF to be spatially constant. Given the alpha matte α, computed with the Improved Color Modeling method, we derive an initial approximation of α^b by thresholding α at 0.5 ($\alpha^b \approx \delta(\alpha > 0.5)$). We can then obtain an approximation of K by minimizing the linear system

$$||\delta(\alpha > 0.5) \otimes K - \alpha||^2. \tag{6.3}$$

Note that we can potentially refine the resulting kernel K, once we have computed a more accurate binary segmentation α^b (section 6.2.5). For this purpose, we re-estimate K, by solving the linear system in eq. (6.3) after replacing $\delta(\alpha > 0.5)$ in eq. (6.3) with α^b.

6.2.4 Alpha Deblurring

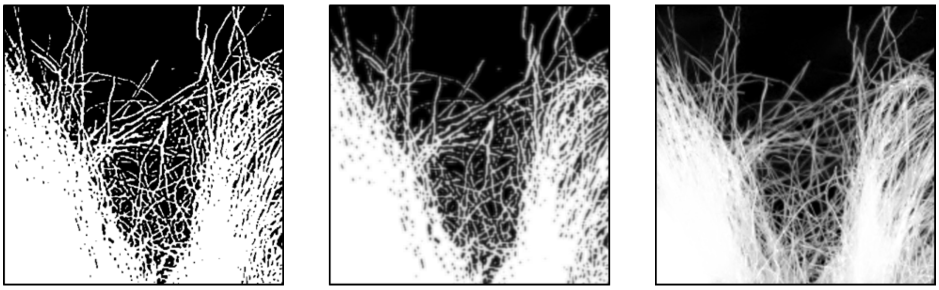
We now deblur the initially computed alpha matte α with the computed PSF K, using the algorithm by Levin et al. [LFDF07]. The resulting deblurred alpha matte α^d, derived from the initial alpha matte in figure 6.2(b), is shown in figure 6.2(c). We can see that the alpha values in the deconvolved matte are distributed more sparsely than those in the initial matte. Hence, this deblurred matte can serve as an initial approximation of the binary segmentation. However, the deconvolved matte is still far from being binary. To derive a binary segmentation one could simply threshold the deconvolution result. However, we observed that thresholding removes many fine details, such as hairs. Therefore, in the next section, we introduce a new binary segmentation method that we use to derive a much better binarization from α^d.



(a) Cropped image. Trimap: white-bkg; black-fgd

(b) Initial alpha matte computed with the Improved Color Modeling method

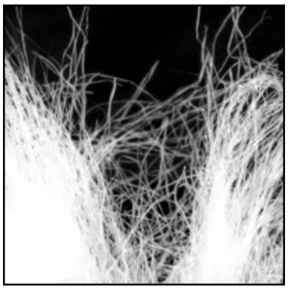
(c) α^d computed from (b)



(d) α^b computed from (c)

(e) Our prior (convolving (d) with the PSF)

(f) Final alpha obtained with our Hybrid Deconvolution approach



(g) Ground truth alpha

Figure 6.2: **Intermediate results of our matting approach.** For the input image (a) we first compute an initial alpha matte shown in (b). The alpha matte in (b) is then deconvolved giving the result in (c). We then binarize the deconvolved alpha matte using our new edge-preserving segmentation approach. The resulting binary segmentation is shown in (d). Convolving this binary segmentation with the PSF gives a prior for the alpha matte shown in (e). The final alpha matte (f), computed with the prior in (e), shows fewer artifacts than our initial matte (b).

6.2.5 Binarization of the Deblurred Alpha Matte

To derive the binary segmentation α^b from the deblurred alpha matte α^d, we formulate the following energy function E:

$$E(\boldsymbol{\alpha}^b) = \sum_{i \in \mathcal{I}} D(\alpha_i^b) + \theta_1 \sum_{\{i,j\} \in \mathcal{N}} V(\alpha_i^b, \alpha_j^b), \tag{6.4}$$

where $\alpha_i^b \in \{0, 1\}$ denotes the binary label of the image pixel i and the vector $\boldsymbol{\alpha^b}$ encodes the labeling on the set of image pixels $\mathcal{I}$. The set of neighboring pixels is denoted by $\mathcal{N}$ (we use an 8-connected neighborhood). We minimize the energy function in eq. (6.4) by finding the minimum cut in a specialized graph via Quadratic Pseudo Boolean Optimization (QPBO) [KR07]. The terms D and V in eq. (6.4) are given by

$$\begin{aligned} D(\alpha_i^b) &= |\alpha_i^b - \alpha_i^d| + \theta_2 |\alpha_i^b|; \\ V(\alpha_i^b, \alpha_j^b) &= V^{Potts}(\alpha_i^b, \alpha_j^b) + \theta_3 V^{edge}(\alpha_i^b, \alpha_j^b) \end{aligned} \tag{6.5}$$

where the constants $(\theta_1, \theta_2, \theta_3) = (5, 0.2, 0.002)$ weight the individual terms. The first part of the data term D encourages the labeling to be similar to α^d and the second part encodes a small preference towards an alpha value of zero. The preference towards zero was motivated by the empirical observation that the loss of some thin structures (e.g. hair strands) is visually less distracting than erroneously attaching parts of the background to the foreground object.

The pair-wise term V consists of two sub-terms V^{Potts} and V^{edge}. The first term V^{Potts} is defined as $V^{Potts}(\alpha_i^b, \alpha_j^b) = \delta(\alpha_i^b \neq \alpha_j^b)$, where the Kronecker delta δ encodes the standard Potts model. Thus V^{Potts} imposes a cost of 1, if two neighboring pixels i and j are assigned to different labels, and zero costs otherwise.

The second term, V^{edge}, was designed to preserve thin structures of the deblurred alpha matte α^d in the binary segmentation α^b. It is defined as $V^{edge}(\alpha_i^b, \alpha_j^b) = (\alpha_i^b - \alpha_j^b)(\alpha_j^d - \alpha_i^d)$. The term V^{edge} imposes no costs if two neighboring pixels i and j are assigned to the same label. However, if two neighboring pixels are assigned to different labels, the costs of V^{edge} depend on the values of pixels i and j in the deblurred alpha matte α^d.

Minimizing the energy function in eq. (6.4) gives a binary segmentation, for which an example is depicted in figure 6.2(d). We see that the fine hair strands were nicely preserved.

6.2.6 Re-estimating Alpha Using the Segmentation Prior

Given the computed binary segmentation α^b (figure 6.2(d)) and the computed blur kernel K, we now close the "loop" by improving the alpha matte using our alpha model in eq. (6.1) as prior. The prior P is constructed by convolving the binary α^b with K. The resulting prior is shown in figure 6.2(e). The prior is then simply added to the objective function by replacing $\hat{\alpha}$ in eq. (6.2) with this term:

$$\hat{\alpha}_i^{new} = \hat{\alpha}_i + \theta_4 P, \tag{6.6}$$

where $\theta_4 = 5$ is the relative weight of the new prior. The final alpha matte, depicted in figure 6.2(f), shows less blurry artifacts than the initial matte in (b).

6.2.7 Multi-resolution Estimation of the Matte

To obtain high-quality alpha mattes of high-resolution (e.g. 6 Mpix) images within reasonable time and memory requirements, we use a multi-resolution framework with three levels: 0.3 Mpix, 1.5 Mpix and 6 Mpix. The matte in lower resolutions is used as a weak regularization for higher resolutions. At the higher resolution, α is solved by processing the image in overlapping windows. Using the low resolution matte as regularization has two advantages: (a) it encourages a smooth transition between windows (for that reason, this prior gets a higher weight along window boundaries), (b) it pushes the solution towards the global optimum, which is essential for handling windows without user constraints.

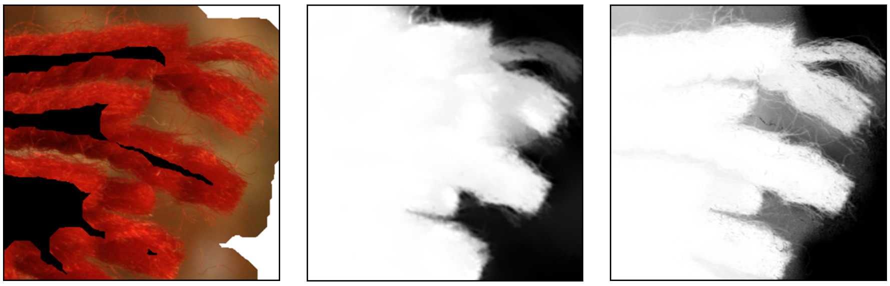
6.2.8 Experimental Results

To test the performance of our matting approach, we computed alpha mattes on different images and trimaps. In this section we present qualitative results to demonstrate the goodness of our method. A quantitative comparison of our algorithm to the state-of-the-art is presented in chapter 7.

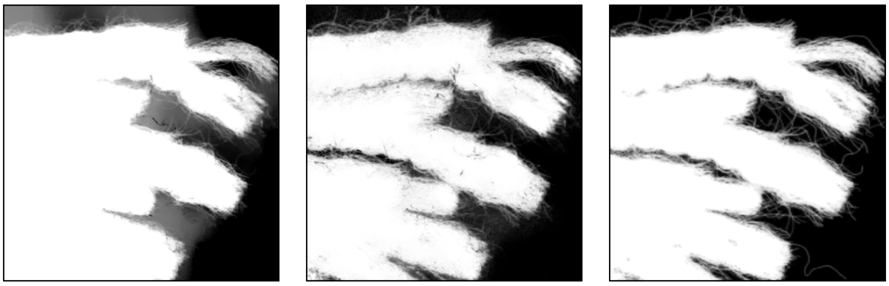
In figure 6.3, our approach is compared to its closest competitors [WC07a, LLW08, LRAL08] on a crop of a high-resolution image showing part of a wool scarf. We see that the results of all competitors show large blurry artifacts in the background. This is because the background colors are erroneously interpreted as semi-transparent layers. Even the pixel-independent sparsity priors of [LRAL08] and [WC07a] were not able to recover the correct alpha values. Our prior successfully removed the background artifacts (figure 6.3(f)), because large semi-transparent regions are very unlikely to occur in our segmentation-based model.

A further example is given in figure 6.4. The results were computed on the crop of a high-resolution image showing hairs of a toy. We see that the Spectral Matting approach gave the worst result (figure 6.4(c)). Even more severe, it seems that its pixel-independent sparsity prior biased the wrong pixels towards an alpha value of 0 and 1. The results of [LLW08] and [WC07a] are better but show large regions where the color was erroneously interpreted as semi-transparent. Again we can see that by explicitly committing to a specific alpha formation model, we can obtain a much better result since large blurry regions are unlikely to occur in our model (see figure 6.4(f)).

(a) High-resolution image



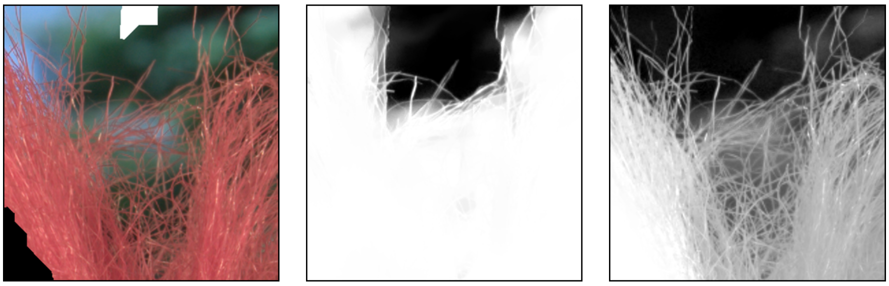
(b) Image crop and trimap (white: bkg; black: fgd) (c) Spectral Matting [LRAL08] (d) Closed-form Matting [LLW08]



(e) Robust Matting [WC07a] (f) Our Hybrid Deconvolution Matting (g) Ground truth

Figure 6.3: **Comparison of matting methods (1).** (b) Crop of the 7.7 Mpix image in (a) showing a region with a woolen scarf. The input trimap is superimposed: black (foreground) and white (background). (c-e) Results of various methods. Our result in (f) shows fewer artifacts in the background and is closer to the ground truth in (g).

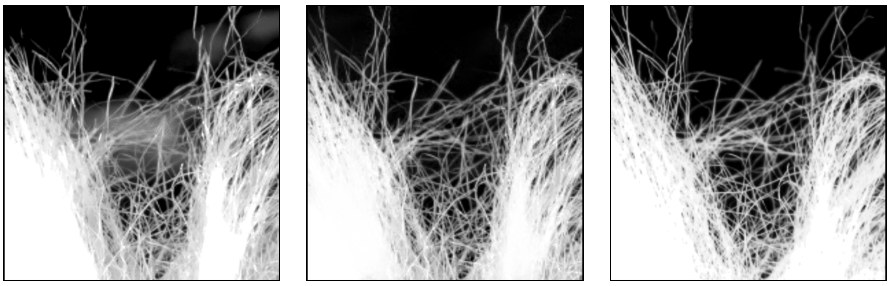
(a) High-resolution image



(b) Image crop and trimap (white: bkg; black: fgd)

(c) Spectral Matting [LRAL08]

(d) Closed-form Matting [LLW08]



(e) Robust Matting [WC07a]

(f) Our Hybrid Deconvolution Matting

(g) Ground truth

Figure 6.4: **Comparison of matting methods (2).** (b) Crop of the 7.6 Mpix image in (a) showing hair of a toy. The input trimap is superimposed: black (foreground) and white (background). (c-e) Results of various methods. Our result in (f) is closest to the ground truth in (g).

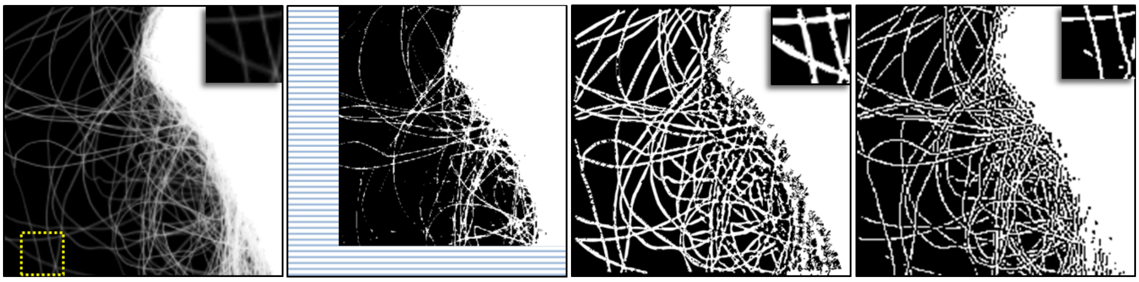
6.3 Segmentation-based Deconvolution

In the previous section we introduced a method to obtain the binary segmentation α^b from the deconvolved alpha matte using a new segmentation approach that preserves the edges in the deblurred alpha matte. We have seen that this method can effectively preserve thin structures like hair strands in the binary segmentation α^b. Let us now re-investigate the quality of this approach using the example from section 6.1. We use the ground truth alpha matte α^* for this comparison. However, for the matting algorithms described in sections 6.3.1-6.3.7, we use an alpha matte computed from the input image with the Improved Color Matting algorithm described in chapter 5.

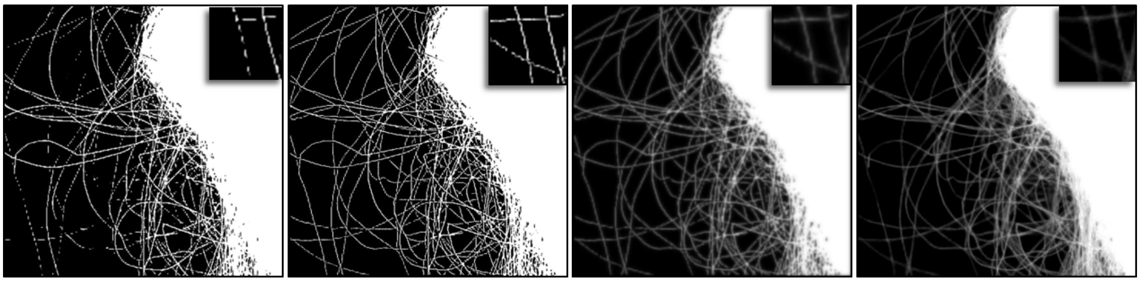
Figure 6.5(d) shows α^b estimated from the ground truth alpha (figure 6.5(a)) with our Hybrid Deconvolution method. We see that most details are better preserved in comparison to the results of [Jia07] and [JSK08] (figure 6.5(b) and (c)). However, α^b is overestimated (i.e. the segmentation of the hair strands is too wide) and originally connected hair strands appear fragmented (see e.g. the inlet in the upper right corner of figure 6.5(d)).

Therefore, the goal of this section is to improve on the segmentation results of the Hybrid Deconvolution approach in several important respects. Firstly, we propose to work on the higher-resolution (upscaled) alpha matte, where the underlying binary segmentation of thin structures is more likely to be binary (see a more detailed discussion in section 6.3.3). We also found that working in the higher resolution greatly improves the result of the Hybrid Deconvolution algorithm for which an example is depicted in figure 6.5(e). Secondly, our Segmentation-based Deconvolution approach estimates the binary segmentation α^b directly from the alpha matte, as opposed to the Hybrid Deconvolution method, where computationally expensive deconvolution methods were applied to alpha before binarization. Thirdly, we apply a different segmentation procedure, described in section 6.3.4, which enforces connectivity of the binary segmentation α^b.

Figure 6.5(f) shows α^b obtained with our Segmentation-based Deconvolution method from the ground truth α^*. We see that most of the fine details were nicely recovered and, in contrast to the Hybrid Deconvolution method, α^b is not overestimated. Furthermore, the segmentation of the foreground is connected, whereas this is not always the case for the Hybrid Deconvolution method. Convolving our computed α^b with our estimated PSF gives



(a) Ground truth alpha (b) α^b from [Jia07] (in high-resolution) (crop of (a) due to memory limits) (c) α^b from [JSK08] (d) α^b using our Hybrid Deconvolution approach (in low-resolution)



(e) α^b using our Hybrid Deconvolution approach (in high-resolution) (f) α^b using our Segmentation-based Deconvolution method (in high-resolution) (computed 13 times faster than (e)) (g) Prior from our Segmentation-based Deconvolution approach. (h) Final alpha matte with our Segmentation-based Deconvolution approach.

Figure 6.5: **Deconvolution with binary prior (more results).** (Extension of figure 6.1 in section 6.1.) Using the ground truth alpha (a), our segmentation approach (f) estimates the underlying binary segmentation better than previously proposed approaches for this task (b,c,d,e). For a better visualization, we show a zoom-in of the yellow-marked region in (a) in the upper right corners of the results in (c-h).

an alpha matte (figure 6.5(g)) which is very close to the ground truth, both visually and in terms of error rate. To further refine this result, we use it as prior in the Improved Color Matting approach (see result in figure 6.5(h)). This example shows that our Segmentation-based Deconvolution method has the potential to estimate the model parameters α^b and K with high accuracy.

6.3.1 Overview of our Matting Approach

Our matting approach comprises five steps: (i) Given an input image and trimap, compute an initial (usually imperfect) alpha matte α with the Improved Color Matting algorithm that was presented in chapter 5; (ii) upsample α to a resolution where the underlying segmentation is more likely to be binary, apart from discretization artifacts; (iii) estimate the binary segmentation α^b with an MRF; (iv) downsample α^b again and compute the PSF; (v) convolve α^b with the PSF and use the convolved α^b as a prior in the Improved Color Matting approach to estimate the final alpha matte α. In the following, each step is described in more detail.

6.3.2 Estimating the Initial Alpha Matte

Following our Hybrid Deconvolution approach, we compute an initial alpha matte α using the Improved Color Modeling algorithm proposed in chapter 5. To avoid overlap with section 6.2, we refer the reader to section 6.2.2 for details.

6.3.3 Upsampling Alpha

It is possible that small structures like hair strands project to a camera sensor area which is smaller than a pixel. To ensure that the underlying binary structure is at least of the size of one pixel, we compute α on a higher-resolution pixel grid. Thus we bicubically upscale the image to a resolution where the underlying segmentation is likely to be binary (i.e. where the underlying binary structures are at least on the order of the size of a pixel).

We found that a scaling factor of 3 was sufficient to preserve most details in our test images. However, further work could be conducted to learn the optimal scaling factor in a user study.

6.3.4 Estimating the Binary Segmentation from Alpha

Our approach recovers the binary mask α^b from the upscaled alpha matte α by solving the following energy function with a graph cut technique:

$$E(\boldsymbol{\alpha}^b) = \sum_{i \in \mathcal{I}} D(\alpha_i^b) + \theta_1 X(\alpha_i^b) + \theta_2 \sum_{\{i,j\} \in \mathcal{N}} V(\alpha_i^b, \alpha_j^b), \tag{6.7}$$

where $\alpha_i^b \in \{0, 1\}$ denotes the binary label of the image pixel i and the vector $\boldsymbol{\alpha}^b$ encodes the labeling on the set of image pixels $\mathcal{I}$. $\mathcal{N}$ denotes an 8-connected neighborhood on the set of image pixels $\mathcal{I}$. The constants θ_1 and θ_2 balance the terms in eq. (6.7) and were fixed to 200 and 0.005, respectively.

The data term D encourages α^b to be close to α:

$$D(\alpha_i^b) = \delta(\alpha_i^b = 1) \cdot L_i, \tag{6.8}$$

where δ is the Kronecker delta and L_i is the difference of the negative log likelihood that a pixel i with alpha value α_i belongs to the fore- or the background, respectively:

$$L_i = -\log(2\alpha_i) + \log(2(1 - \alpha_i)). \tag{6.9}$$

To detect edges and to preserve thin structures like hair strands in the segmentation, we use flux, which has been shown to be effective for segmenting thin objects in medical grayscale images [VS02] and has been demonstrated to be amenable for graph cut minimization [KB05]. The unary term X represents the flux of the gradient in L:

$$X(\alpha_i^b) = \delta(\alpha_i^b = 0) \cdot div\left(\nabla L_i \cdot \exp\left(-|L_i|/\sigma\right)\right), \tag{6.10}$$

where ∇ and div denote the gradient and divergence and σ was fixed to 2. In X, the exponential function is used to truncate the gradient in places where the foreground and background likelihoods in L_i are approximately equal.

Finally, our pairwise term V encourages neighboring pixel to be assigned to the same label:

$$V(\alpha_i^b, \alpha_j^b) = \delta(\alpha_i^b \neq \alpha_j^b). \tag{6.11}$$

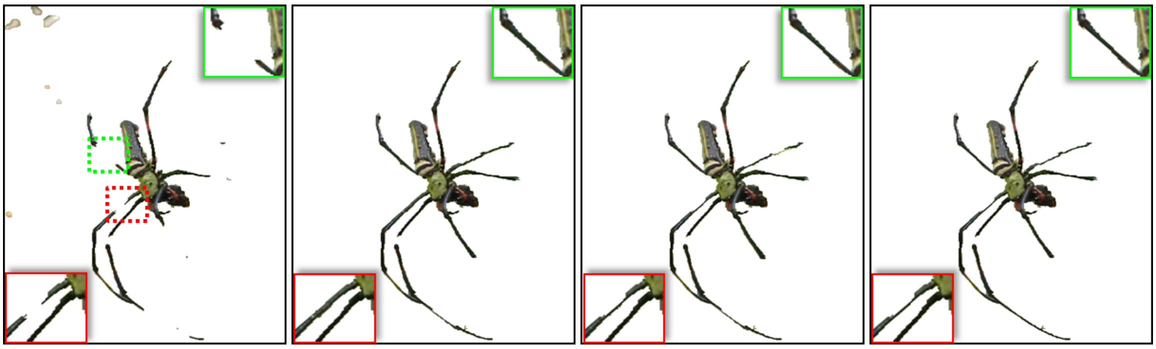
Enforcing Connectivity

In addition to the smoothness prior, discussed before, we enforce the foreground object to be a single 4-connected component. In general, this assumption holds for non-occluded objects, and also for the images used for evaluation in section 6.3.7. Recently, a solution to minimize energy functions, like eq. (6.7), under connectivity constraints has been presented by Nowozin et al. [NL09]. Unfortunately, their solution to this $\mathcal{NP}$-hard problem requires the image to be segmented into large super-pixels for computational reasons. Thus it is impractical for segmenting fine structures like hair strands. An interactive solution to this problem was proposed by Vicente et al. [VKR08]. They start by computing a segmentation without connectivity constraints (e.g. figure 6.6(a)). Then the user manually marks a pixel which has to be connected to the main part of the foreground object, and also manually selects a minimum width for the “connection path”. The method finds a connected component which fulfills these constraints.

In the following we propose a new approach to compute an entirely connected segmentation, which in contrast to previous work is very efficient and fully automatic. In essence, we automate the user interactions of [VKR08] and also make the core algorithm of [VKR08] much more efficient while keeping high-quality results.

More precisely, we first compute a segmentation $\hat{\alpha}^b$ by minimizing (6.7) without connectivity constraints (figure 6.6(a)). Then those regions in $\hat{\alpha}^b$ which are disconnected from a source region s are identified. We define s to be all pixels in $\hat{\alpha}^b$ that are 4-connected to the user-marked foreground pixels (e.g. spider body in figure 6.6(a)). Then for each disconnected region t a segmentation $\hat{\alpha}^{b'}$ is computed by minimizing (6.7) under the constraint that s and t must be connected. (This step is discussed in detail below.) We also determine an alternative solution $\hat{\alpha}^{b''}$ by simply removing region t from $\hat{\alpha}^b$. Now we keep the solution with lower energy, i.e. we keep, e.g., $\hat{\alpha}^{b'}$ if $E(\hat{\alpha}^{b'}) \leq E(\hat{\alpha}^{b''})$. In this manner all disconnected regions are processed, which gives the final result (figure 6.6(b)).

The difficult step in the above procedure is to find a segmentation subject to the condition that regions s and t are connected. Vicente et al. [VKR08] suggested a heuristic method called *DijkstraGC*. It works by computing the “shortest path” in a graph where the “distance” between two nodes measures the value of the energy (6.7) under the constraint that all pixels on the path from s to t belong to the foreground. Unfortunately, DijkstraGC is computationally very expensive, since it requires many calls to the maxflow algorithm to

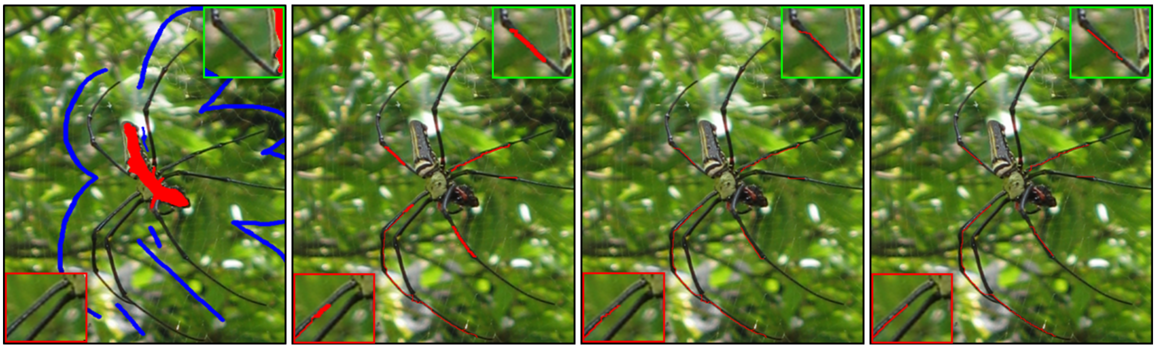


(a) Segmentation without connectivity

(b) Our final connected result

(c) Our connected result using a fixed minimum path width

(d) Result using DijkstraGC [VKR08] (40 times slower than ours)



(e) Image with scribbles (blue=bkg; red=fgd)

(f) Input image. Computed connected paths from (b) are marked red.

(g) Input image. Computed connected paths from (c) are marked red.

(h) Input image. Computed connected paths from (d) are marked red.

Figure 6.6: **Enforcing connectivity.** Given an input image and user constraints (e), an originally disconnected binary segmentation is computed (a). Our approach automatically connects (or excludes) disconnected islands in (a) to the foreground. Our final binary segmentation (b) includes most of the spider legs and shows no background artifacts. The result of our approach, where we disable the automatic estimation of the minimum width of the "connection path" (hence, we use a fixed minimum width of 1 pixel) is shown in (c). As expected it is worse than (b). Our results (b,c) are comparable to the result of DijkstraGC (d), which is, however, 40 times slower than our approach. We show the "connection paths" for the results in (b-d) in (f-h). (For this example we replaced eq. (6.7) with the energy in [RKB04] to compute the initial binary segmentation in (a), for reasons of compatibility with the original implementation of [VKR08].)

minimize function (6.7).[3] Hence, we found it impractical to compute a solution for many disconnected islands.

The key idea of our approach is to compute the shortest path on a graph where the weight of each node is its min-marginal energy under (6.7), which is given by

$$M(i) = \min_{\alpha^b, \alpha_i^b = 1} E(\alpha^b) - \min_{\alpha^b} E(\alpha^b), \tag{6.12}$$

where $\min_{\alpha^b, \alpha_i^b = 1} E(\alpha^b)$ returns the minimum energy when fixing the variable α_i^b to a value of 1, while minimizing over all other variables. The min-marginal energy can be computed very efficiently using graph recycling [KT06]. (The path to all disconnected islands can be computed in a single run of Dijkstra.) A segmentation is then computed by minimizing (6.7) under the constraint that all pixels on the shortest path in the min-marginals belong to the foreground. Hence, our approach approximates DijkstraGC but gives comparable results (for instance, compare our result in figure 6.6(b) with the result of DijkstraGC in figure 6.6(d)).

Finally, we address the problem of finding the minimum width of the "connection path". It has been observed in [VKR08] that DijkstraGC might result in undesired one-pixel-wide segmentations (see e.g. figure 6.6(c,d)). In [VKR08] this problem was fixed by manually specifying a minimum width for each connecting path (see [VKR08] for details). We automate this process by computing multiple shortest paths with different widths $\varphi \in \{1, .., 4\}$ for each disconnected island and choose that path which gives the segmentation with the lowest costs under (6.7). We encourage thicker paths by dividing the costs of paths where $\varphi > 1$ by a factor of 1.005.

6.3.5 Estimating the PSF

We model the PSF as a spatially constant kernel K of size $R \times R$ with non-negative elements that sum up to one (we use $R = 6$). Similar to [JSK08], we apply a smoothness prior to K that is given by $\gamma ||\nabla K||^2$, where $\gamma = R^2$ normalizes the kernel area. Given α^b and the

---

[3] In [VKR08] the computational burden was reduced by recycling flow and search trees [KT07]. But the authors of [VKR08] found that their effectiveness was significantly reduced, since nodes had to be (un)fixed in an unordered fashion.

ground truth alpha α, we can obtain K by minimizing the quadratic energy function:

$$||\alpha^b \otimes K - \alpha||^2/\sigma^2 + \theta_3\gamma||\nabla K||^2, \tag{6.13}$$

where σ denotes the noise level and θ_3 weights the smoothness prior. (In our implementation, we have chosen $\sigma = 0.005$ and $\theta_3 = 2$.) Note that for the Hybrid Deconvolution method (section 6.2) we used a very similar formulation to derive K. However, for the Segmentation-based Deconvolution algorithm we apply a smoothness constraint, since the symmetry constraint used in the Hybrid Deconvolution method cannot account for potential slight motion blur. For computational reasons we compute K in the original image resolution, thus we bicubically downsample α^b before PSF computation. We found this to give similar results as computing the PSF from the upscaled alpha matte and then downsampling the convolved result.

6.3.6 Re-estimating Alpha Using the Segmentation Prior

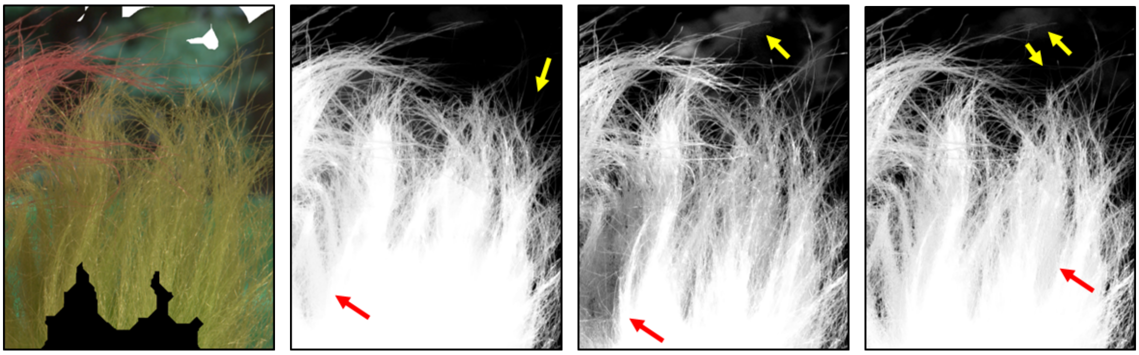
Following our Hybrid Deconvolution approach, we now construct the alpha prior P by convolving the computed binary α^b with the computed blur kernel K: $P = (\alpha^b \otimes K)$. We then re-estimate the alpha matte α by using P as data term in the Improved Color Matting approach as shown in section 6.2.6.

To obtain high-resolution alpha mattes with reasonable memory requirements, we solve for alpha in a window-based fashion similar to section 6.2.7. Since we only have to process pixels in the unknown trimap region, we found memory requirements reasonable.

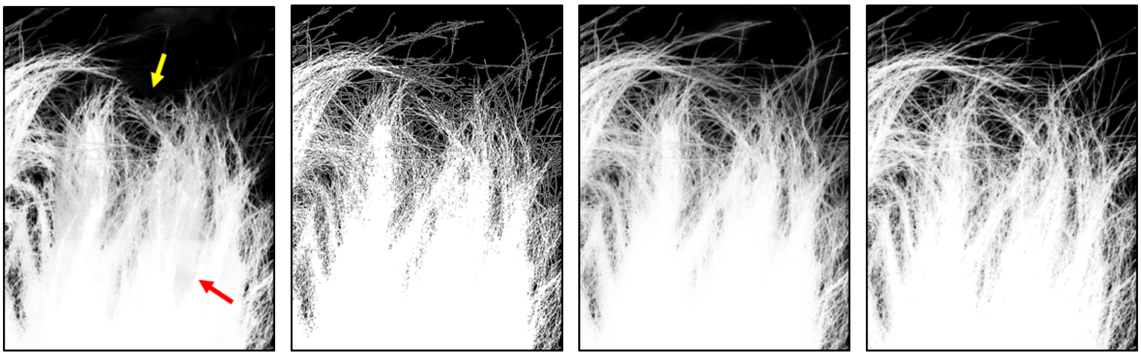
6.3.7 Experimental Results

In this section we demonstrate the good performance of our matting approach by presenting qualitative results on different images and trimaps. A detailed quantitative comparison of our algorithm to the state-of-the-art is presented in chapter 7.

Figure 6.7 compares our approach to its closest competitors for a crop of an image showing fuzzy hair (figure 6.7(a)). We see that the approach of [LLW08] (figure 6.7(b)) cuts off some hair and overestimates alpha in other regions. The approach of [WC07a]



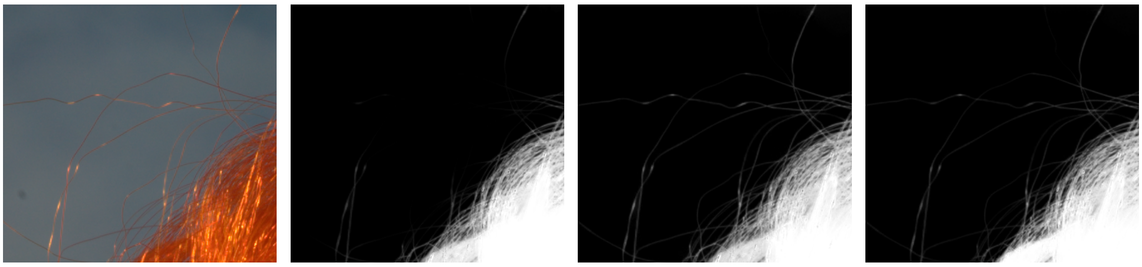
(a) Image crop + trimap (inverted) (b) Result of [LLW08] (c) Result of [WC07a] (d) Result of the Improved Color Matting



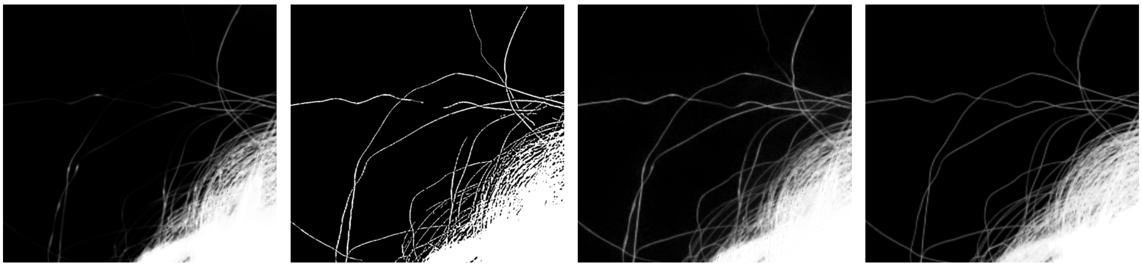
(e) Result of the Hybrid Deconvolution approach (f) Binary segmentation obtained with our Segmentation-based Deconvolution method (g) Final result obtained with our Segmentation-based Deconvolution method (h) Ground truth alpha

Figure 6.7: **Comparison of matting methods (1).** (b-g) Results for a crop of an image (a) showing the hair of a toy. Arrows point to artifacts. The input trimap is superimposed (trimap was inverted for better visibility). See the text for a discussion.

(figure 6.7(c)) better recovers the hair strands but introduces artifacts in the background and underestimates alpha in other places. The result of our Improved Color Matting approach (figure 6.7(d)) shows are cleaner background but the alpha in the foreground regions is still underestimated and some hair strands are missing. The Hybrid Deconvolution approach (figure 6.7(e)) could not improve the alpha matte. Even worse, it removed some of the hair strands. The result of our Segmentation-based Deconvolution algorithm (figure 6.7(g)) is based on the segmentation in figure 6.7(f) and is closest to the ground truth alpha matte



(a) Image crop + trimap (inverted)
(b) Result of [LLW08]
(c) Result of [WC07a]
(d) Result of the Improved Color Matting



(e) Result of the Hybrid Deconvolution approach
(f) Binary segmentation obtained with our Segmentation-based Deconvolution method
(g) Final result obtained with our Segmentation-based Deconvolution method
(h) Ground truth alpha

Figure 6.8: **Comparison of matting methods (2).** (b-g) Results for a crop of an image (a) showing hair of a toy. Arrows point to artifacts. The input trimap is superimposed (trimap was inverted for better visibility). See the text for a discussion.

(figure 6.7(h)). We can see that it preserves the hair strands and shows only small artifacts in the background.

Another example is depicted in figure 6.8, which shows results for a crop of an image showing hair strands of a doll (figure 6.8(a)). We see that [LLW08], [WC07a] as well as our Improved Color Matting approach underestimate the alpha values at the fine hair strands (figure 6.7(b-d)). Also our Hybrid Deconvolution approach could not completely recover the alpha matte. In contrast our Segmentation-based Deconvolution approach (figure 6.7(g)), based on the segmentation shown in figure 6.7(f), is closest to the ground truth (figure 6.7(h)), since it preserves the hair strands.

6.4 Model Analysis

In the previous sections we have introduced a prior that models the alpha matte α as a convolution of an underlying, potentially higher-resolution, binary segmentation α^b with a point spread function K (see eq. (6.1)). This prior is based on the assumption that the fractional alpha values are induced mostly by the imaging process, in particular, caused by by the camera's Point Spread Function (PSF). In the following, we quantitatively demonstrate that our model, based on a binary segmentation and PSF, can describe complex alpha mattes, originating from e.g. hairs or fur.

We evaluated the goodness of our model on a dataset of 27 natural images with corresponding ground truth alpha mattes (see chapter 7 for details about the dataset). First, as we aim at modeling alpha mattes for scenes without light-transmitting objects, we excluded one alpha matte that possessed such region from the test set (shown in figure 6.9). Thus we tested our model on the remaining 26 alpha mattes. We believe that these 26 images represent a typical set of photographs, and similar data were used for comparison in previous matting work. For evaluation purposes we split the set into two classes by careful manual inspection of the images. The first class comprises 7 images that show only *solid*, opaque objects with sharp boundaries. The second class comprises the remaining 19 *fuzzy* objects that have a boundary which potentially transmits light (e.g. hair or fur).

Clearly, the segmentation-based model is a good representation for the 7 opaque (solid) objects. So the key question is whether our model is also a good representation for the fuzzy objects. Although for these objects the majority of fractional alpha values could be caused by the PSF, parts of these objects might also transmit light (violating our assumption). To answer this question, we conducted the following experiment.

Given the ground truth alpha matte α^*, we computed the underlying binary segmentation α^b and blur kernel K with the Segmentation-based Deconvolution algorithm described in section 6.3. We implemented a further variant of this algorithm that can handle spatially varying defocus blur kernel, which further improved the results. In particular, given the ground truth alpha matte α^* and the computed binary segmentation α^b, we first derive the shape of the blur kernel K (e.g. Gaussian Kernel) by computing a single (spatially constant) kernel from all pixels in the image as described in section 6.3.5. Then we locally

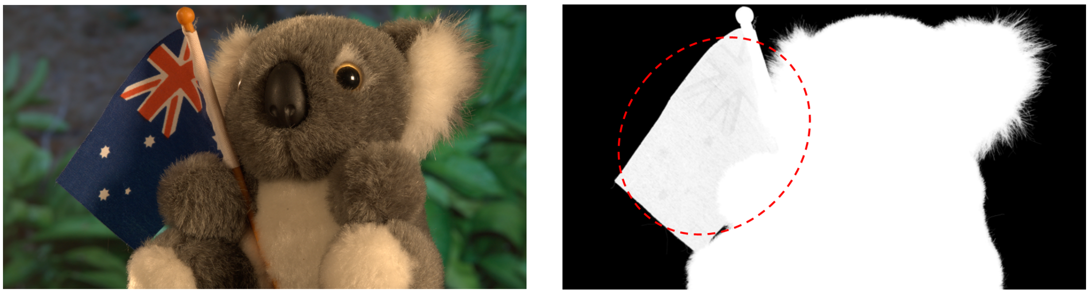
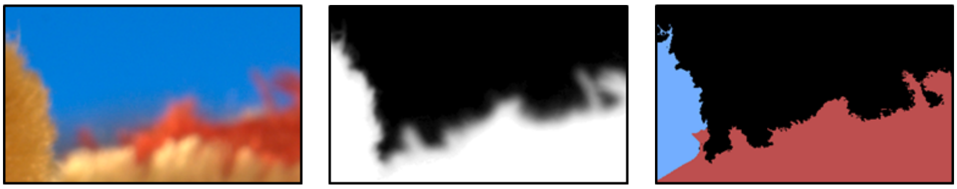

(a) Image (b) Ground truth alpha matte of (a)

Figure 6.9: **Limitation - large translucent area.** We did not validate our model on the ground truth alpha matte depicted in (b), because its fractional alpha values were mainly caused by the light transmitting flag that belongs to the foreground object (marked in red). Hence, for this alpha matte our assumption that the fractional alpha values are generated by the PSF is largely violated.

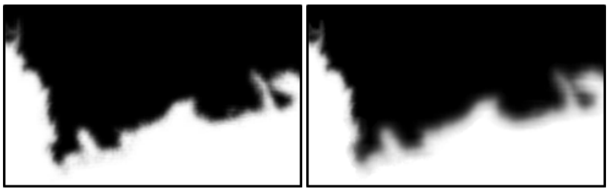
estimate the optimal scale S (which re-sizes K by a factor of S) of this kernel by minimizing $S_i = \arg\min_S(||\alpha^b \otimes K_S - \alpha^*||)$ over a local window centered at pixel i. Here K_S is the bicubically scaled kernel K with a scaling factor of S. (We used four scales $S = \{1, 2, 3, 4\}$.)

A result of this approach is depicted for the image crop in figure 6.10(a), which shows part of a soft toy whose background is more heavily blurred than the foreground, due to a narrow depth of field. Using the ground truth α^* (figure 6.10(b)), we computed the scale factor S for every pixel of the binary segmentation (figure 6.10(c)). Convolving α^b with the spatially varying kernel delivers results (figure 6.10(e)) close to the ground truth, whereas using a constant PSF cannot recover the larger amount of blur in the background (figure 6.10(d)).

Once the binary segmentation α^b and spatially varying PSF K were computed from the ground truth alpha matte, the goal was then to predict the ground truth alpha matte α^* by convolving the computed α^b with K. Clearly, the errors obtained by comparing the predicted alpha matte α with the ground truth α^* will not be zero. This is because α^b and K are still approximations of the reality. However, the error rate obtained for the class of truly opaque objects (for which we can assume that our model is a very good approximation of alpha) indicates some limit of how close we can approximate the ground truth alpha using



(a) Input image (b) Ground truth alpha (c) PSF scale: blue:1, red:2



(d) α^b convolved with constant PSF (e) α^b convolved with varying PSF

Figure 6.10: **Spatially varying PSF.** For the example in (a), the ground truth alpha (b) cannot be modeled with a constant PSF (d), due to depth-dependent defocus. We split the image in regions (c) where the scale of the PSF is approximately constant to recover the alpha matte in (e).

our computed α^b and K. Hence, if the error rates for the truly opaque objects are close to those obtained for the class of fuzzy objects (i.e. for objects that might be partially light transmitting), our model is very likely to be also a good representation for the latter group.

The obtained error rates for this experiment are shown in the upper part of table 6.1. It shows the errors between the computed alpha matte α and the ground truth α^* with respect to three error metrics that were averaged over the 7 and 19 test cases comprising only solid and fuzzy objects, respectively. The error metrics are defined as Mean Absolute Distance (MAD): $1/|\mathcal{U}| \sum_{i \in \mathcal{U}} |\alpha_i - \alpha_i^*|$, Mean Squared Error (MSE): $1/|\mathcal{U}| \sum_{i \in \mathcal{U}} (\alpha_i - \alpha_i^*)^2$ and Gradient Error (Grad): $1/|\mathcal{U}| \sum_{i \in \mathcal{U}} (\nabla \alpha_i - \nabla \alpha_i^*)^2$. Here $\mathcal{U}$ denotes the set of pixels in the unknown region of the trimap (which here corresponds to all pixels with a fractional alpha value in the ground truth matte α^*). The error rates were finally multiplied by a factor of 100 such that they correspond to percentage values.

The first row of table 6.1 shows errors for the results obtained by convolving α^b with K that we computed with our Segmentation-based Deconvolution approach. The results in the second row were obtained by additionally applying [LLW08], i.e. using $(\alpha^b \otimes K)$ as prior in the framework of [LLW08] (similar to section 6.3.6). We see that for both methods

Method	MAD	MSE	Grad
	solid ; fuzzy	solid ; fuzzy	solid ; fuzzy
Segm.-based Deconv.	4.5 ; 4.9	0.4 ; 0.4	1.0 ; 0.6
Segm.-based Deconv. + [LLW08]	**4.5 ; 4.5**	**0.4 ; 0.4**	**0.4 ; 0.2**
Joshi 08 [JSK08]	6.9 ; 16.3	0.9 ; 5.8	1.0 ; 1.9
Levin 08 [LFDF07]	8.8 ; 13.6	1.3 ; 3.0	1.1 ; 1.1
Hybrid Deconvolution (section 6.2)	5.4 ; 9.0	0.5 ; 1.5	0.8 ; 0.7

Table 6.1: **Model errors.** Errors averaged over solid; fuzzy objects suggest that our model is valid to a high degree of accuracy also on fuzzy objects. The gap between the error rate of our model and those obtained with practical algorithms (2 lines in the bottom) suggests that our model may improve alpha matting accuracy.

(first two rows in table 6.1) the average error on the opaque objects is very close to the error on the fuzzy class. This demonstrates that our model is also a good representation for fuzzy objects.

We further computed α^b and K with the approaches of [JSK08, LFDF07] as well as our Hybrid Deconvolution approach and compared the re-convolved result to the ground truth. For a fair comparison, we used the results of these methods as prior in the framework of [LLW08].[4] The errors of the so obtained alpha mattes are shown in the middle part of table 6.1. We see that, out of these three algorithms, the Hybrid Deconvolution method (section 6.2) performs best, but is still inferior to our Segmentation-based Deconvolution approach. The error rates also reflect that the method of [JSK08] performs reasonably well for solid objects but is not able to correctly recover α^b for the fine structures of the fuzzy objects.

6.5 Summary

In this chapter we proposed and tested a prior for alpha that is based on a model where the alpha matte is a convolution of a binary segmentation with the camera's Point Spread Function. We have seen that recovering the parameters of this model is related to the task of blind deconvolution. We presented two new deconvolution algorithms that recover the

---

[4] We thresholded the result of the deblurring algorithm [LFDF07] at 0.5 to obtain α^b.

underlying binary segmentation and PSF. Incorporating our new prior into a state-of-the-art matting technique produces results that are of considerably higher quality than those of previous matting algorithms.

The first algorithm, presented in section 6.2, starts by computing an initial approximation of alpha using a traditional matting algorithm. From this initial alpha matte, we infer the PSF and the binary segmentation using a novel segmentation technique that is effective in preserving thin structures like hair strands. Then we blur the binary segmentation with the PSF and use it to re-estimate an improved alpha matte. Our second algorithm (section 6.3) improves on this idea by computing the segmentation from the higher-resolution (up-scaled) alpha matte, where the underlying binary segmentation is more likely to be binary. Furthermore, we apply a different segmentation procedure, which enforces connectivity of the binary segmentation and considerably improves the computational performance. The high-quality of the resulting alpha mattes was demonstrated by showing results on natural images.

Chapter 7

An Evaluation System for Image Matting

To evaluate the performance of our proposed matting algorithms, a quantitative comparison to the state-of-the-art on a standard benchmark test would be highly useful. Unfortunately, no such standard benchmark test had been developed so far for the task of image matting. Therefore, the major goal of this chapter is to design and implement such a benchmark for image matting and to provide it to the scientific community.

A key requirement for a matting benchmark is a challenging test set with corresponding high-quality ground truth alpha mattes. Recently, some ground truth data sets have been proposed to provide a test bed for the matting algorithms in [LRAL08] and [WC07a]. Although these data sets are publicly available, they cannot be used for a benchmark test in a straightforward way, since they have serious flaws. For instance, the data in [LRAL08] is considerably affected by noise from the camera sensor, and the reference solutions in [WC07a] are biased towards some natural image matting algorithms that were used by [WC07a] to generate the ground truth. In contrast, we propose a dataset in section 7.1 with high-quality reference solutions that were generated independently of any previous natural image matting approaches. Our images show a large diversity of natural scenes with a variety of image properties (e.g. different focus settings, translucent scene objects, different depths of field). This dataset largely reflects the challenges inherent to real images and provides the basis for our comparison of matting algorithms.

Another issue addressed in this chapter is that none of the previously proposed datasets has emerged as an accepted standard. As a consequence, comparisons in subsequent work were not conducted on the same coherent set of data, thus lowering their informative value. This is presumably due to the lack of an appropriate online benchmark system that allows other researchers to include novel results. Thus, in section 7.2, we establish a dynamic online benchmark test which provides all data and scripts that enable the research community to complement our evaluation with new results. This will bring researchers in the favorable position to inspect previous work, which will hopefully inspire further research.

Our third contribution in this chapter - besides providing a new ground truth dataset and a dynamic online benchmark test - is to improve on the evaluation methodology for image matting. In previous work (e.g. [WC07b],[WC07a],[LLW08]), such evaluations have been usually tied to simple pixel-wise error measures that do not always correlate to the visual quality as perceived by humans. Thus we go beyond these evaluation methodologies and seek to develop quantitative error measures that are based on subjective human perception. More specifically, we concentrate on two properties of alpha mattes that considerably affect the visual quality of matting results, namely the connectivity of the foreground object and the preservation of gradients in the alpha matte. In section 7.3 we develop error functions that estimate the compliance of these properties, and in a user study we validate that our measures are correlated to human perception. This aspect of our work is related to research in other areas of computer vision where perceptual distance measures have been developed for e.g. image segmentation [PV08, CDGE02] or color constancy [GGL08].

Experimental results presented in section 7.4 show that our dataset is challenging and pronounces strengths and weaknesses of image matting algorithms that were not apparent in previous evaluations. Even more importantly, we show that our matting algorithms presented in chapters 5 and 6 considerably improve on the state-of-the-art.

The remainder of this chapter is organized as follows. In section 7.1 we discuss the construction of our ground truth dataset and analyze its properties. We explain the design of our online benchmark in section 7.2 and derive our perceptually motivated error functions in section 7.3. Finally, we evaluate and compare the performance of our matting algorithms to the state-of-the-art in section 7.4.

7.1 Ground Truth Database

Ideally, a ground truth dataset for image matting should feature several important properties. Firstly, the data should cover a variety of conditions found in real-world images such as color ambiguity, different focus settings, or high-resolution data. Secondly, the data should be challenging in order to further push the limits of current methods, and thirdly the data has to be paired with high-quality ground truth alpha mattes to allow for a meaningful comparison. We strive to construct a dataset that has all these properties.

To obtain ground truth information for real-world images, one could follow the approach of [WC07a], where existing matting methods were applied to natural images and their results were manually combined to a reference solution. We tested this approach on several challenging natural images, but found the resulting alpha mattes to be of low-quality. Furthermore we argue that such a dataset would be biased towards the algorithms that were used to construct the ground truth.

Since there seems to be no reasonable chance to derive alpha mattes with sufficient quality from real-world imagery, we decided to capture high-quality ground truth mattes in a restricted studio environment by triangulation [SB96] (see section 7.1.1). Our set of 35 images is considerably more challenging than previously used data and depicts natural (indoor) scenes that comprise of a variety of challenges one faces in real-world images, like different focus settings (see section 7.1.2). A large dataset might prevent other researchers to upload their own results. This was one of the reasons why we finally split up our data set into 8 test and 27 training images (see section 7.2).

7.1.1 Data Capture

To obtain a composite image that can serve as test image for evaluation purposes, we built up several natural background scenes that were then photographed with a foreground object. To derive high-quality ground truth alpha mattes for these composites, we carefully placed a monitor (Apple Cinema 30” HD) between the object and the scene, without moving neither the object nor the camera (all subsequent shots had to be perfectly aligned with the composition). On the monitor we displayed four single-colored backgrounds (i.e. black, red, green and blue) that were photographed with the foreground object. After capturing the object in front of the screen, the object was removed to photograph the plain single-colored backgrounds as well. This allowed us to extract a ground truth matte by triangulation matting [SB96] (see section 1.2.1 for a description). We obtained 8 images using this setup. For the remaining 27 images, a similar setup was used, but the image compositions were obtained by photographing the objects in front of a monitor which showed natural background images. By using a monitor for projecting the backgrounds, we could avoid the fragile process of placing the monitor between the object and the scene. This enabled us to generate alpha mattes with even higher quality.

All images were shot in unprocessed RAW format with a professional Digital Single Lens Reflex (DSLR) camera (Canon 1D MarkIII with a Canon 28-105mm zoom lens) at a resolution of 10.1 Megapixels with constant camera settings. To avoid camera shake, we locked the mirror of the camera (hence the shutter was the only moving part inside the camera) and used a remote control to trigger the shutter. This enabled us to take images that were registered to each other with sub-pixel accuracy. For computing the alpha matte, the RAW image data was transformed into RGB color images without gamma correction (i.e. linear gamma) in order to avoid the introduction of noise in dark areas. Finally, the images were cropped at a bounding box that was casually drawn around the foreground objects, resulting in test scenes with an average size of about 6 Megapixels.

To assure that our newly recorded ground truth mattes are indeed of high-quality, we evaluated their noise level. For this purpose, we manually marked regions which obviously were supposed to have an alpha alpha value of exactly 1 (i.e. truly foreground) and then computed the number of pixels in these regions with an alpha value lower than 0.97. For our 8 images that were captured by photographing a three dimensional scene, 3.4% of the pixels were below this threshold. For the 27 images where a monitor was used to project natural background images, only 0.3% of the pixels were below this threshold. These are very good values compared to the data in [LRAL08], where we found on average 26.7% of true foreground pixels with an alpha value below 0.97.

7.1.2 Image Properties

Our images exhibit many characteristics of real-world images, like highly textured backgrounds, different depths of field, as well as color ambiguity. We included a range of foreground objects that have different properties such as hard and soft boundaries, translucency or different boundary lengths and topologies (e.g. a tree with many holes).

Our dataset is challenging and exhibits various levels of difficulty. On our data set, the mean squared error (normalized over the number of pixels with unknown alpha values) of the alpha mattes computed using the algorithms of [WC07a] and [LRAL08] (averaged over the algorithms) varies between 0.3 and 21.8, with an average value of 4.2. This is considerably larger than the average error rates we computed with the same procedure on the datasets of [WC07a] and [LRAL08], which are 1.1 and 0.9, respectively. These results suggest that our data set shows a higher variation of difficulty and is more discriminative than previously proposed data sets.

7.1.3 User Input

As we have seen in chapter 2.1, the most common form of user interaction is the trimap interface, where the user manually partitions the image into foreground, background and unknown regions. Some matting algorithms are also capable of working on very sparse trimaps, commonly denoted as scribbles. However, scribbles are subject to an even higher variation of inputs, compared to trimaps, and are often only used to derive a more accurate trimap afterwards [JK05, BS07, RRRAS08].

Given the predominance of trimap input, we decided to simulate the user input by a set of three different trimaps for each test image. Two of them where generated automatically by dilating the unknown region of the high-resolution ground truth trimap by 22 and 44 pixels, respectively. To account for more natural user input, we also included a hand-drawn trimap for every test case. These were generated by an experienced user given a paint tool with a set of three brushes (i.e. unknown, foreground and background) and flood filling capability. The user was imposed a time constraint of 2 minutes per image, which we found sufficient to create a reasonable trimap for all images.

Although most matting algorithms accept trimap input, we plan to extend our benchmark with matting results that were generated by other forms of user interaction or in a completely automatic way (e.g. [LRAL08] supports a component picking interface and a completely unsupervised mode).

7.2 Online Benchmark System

An important reason that has led to the success of recently proposed benchmark tests in computer vision is that they have been made freely available on the web. Inspired by [SS02, BSL$^+$07] who have focused on stereo and optical flow algorithms, we designed an online benchmark that is accessible at www.alphamatting.com. Similar to other online benchmarks, a major advantage of our repository is that it can be dynamically updated with novel datasets or error measures, if needed in the future. We provide all scripts and data necessary to allow other researchers to submit new results. We hope that this will encourage the research community to participate in the competition. A screenshot of our online benchmark is shown in figure 7.1.

Selecting a Representative Test Set

A comprehensive benchmark test for matting algorithms should be carried out on a dataset that covers a large variation of different scenarios that are encountered in practical matting applications. Since we invite other researchers to submit their results to our benchmark, a very large dataset is unreasonable, especially when people process high-resolution images with unoptimized research code. For example, assuming an average computation time of two minutes per image, computing results for our dataset of 35 images on 3 different trimaps would require more than 3 hours. Hence, we need a dataset that is as small as possible but still largely maintains the same variations as the full set.

We decided to split up our database into a test and training set. The test set comprises 8 images for which the ground truth alpha mattes are hidden from the public, in order to largely prevent excessive parameter tuning. The remaining 27 images serve as training dataset with publicly available ground truth. This set may be used by other researchers for parameter learning.

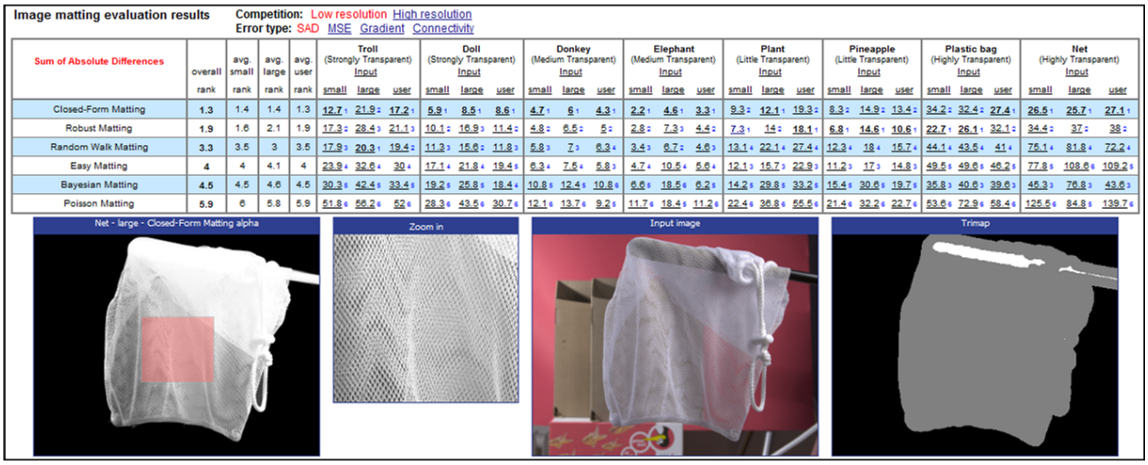


Image matting evaluation results

Competition: Low resolution High resolution
Error type: SAD MSE Gradient Connectivity

Sum of Absolute Differences	overall rank	avg. small rank	avg. large rank	avg. user rank	Troll (Strongly Transparent) Input			Doll (Strongly Transparent) Input			Donkey (Medium Transparent) Input			Elephant (Medium Transparent) Input			Plant (Little Transparent) Input			Pineapple (Little Transparent) Input			Plastic bag (Highly Transparent) Input			Net (Highly Transparent) Input		
					small	large	user	small	large	user	small	large	user	small	large	user	small	large	user	small	large	user	small	large	user	small	large	user
Closed-Form Matting	1.3	1.4	1.4	1.3	12.7^{1}	21.9^{2}	17.2^{1}	5.9^{1}	8.5^{1}	8.6^{1}	4.7^{1}	6^{1}	4.3^{1}	2.2^{1}	4.6^{1}	3.3^{1}	9.3^{2}	12.1^{1}	19.3^{2}	8.3^{2}	14.9^{2}	13.4^{2}	34.2^{2}	32.4^{2}	27.4^{1}	26.5^{1}	25.7^{1}	27.1^{1}
Robust Matting	1.9	1.6	2.1	1.9	17.3^{2}	28.4^{3}	21.1^{3}	10.1^{2}	16.9^{3}	11.4^{2}	4.8^{2}	6.5^{2}	5^{2}	2.8^{2}	7.3^{3}	4.4^{2}	7.3^{1}	14^{2}	18.1^{1}	6.8^{1}	14.6^{1}	10.6^{1}	22.7^{1}	26.1^{1}	32.1^{2}	34.4^{2}	37^{2}	38^{2}
Random Walk Matting	3.3	3.5	3	3.5	17.9^{3}	20.3^{1}	19.4^{2}	11.3^{3}	15.6^{2}	11.8^{3}	5.8^{3}	7^{3}	6.3^{4}	3.4^{3}	6.7^{2}	4.6^{3}	13.1^{4}	22.1^{4}	27.4^{4}	12.3^{4}	18^{4}	15.7^{4}	44.1^{4}	43.5^{4}	41^{4}	75.1^{4}	81.8^{4}	72.2^{4}
Easy Matting	4	4	4.1	4	23.9^{4}	32.6^{4}	30^{4}	17.1^{4}	21.8^{4}	19.4^{5}	6.3^{4}	7.5^{4}	5.8^{3}	4.7^{4}	10.5^{4}	5.6^{4}	12.1^{3}	15.7^{3}	22.9^{3}	11.2^{3}	17^{3}	14.8^{3}	49.5^{5}	49.6^{5}	46.2^{5}	77.8^{5}	108.6^{6}	109.2^{5}
Bayesian Matting	4.5	4.5	4.6	4.5	30.3^{5}	42.4^{5}	33.4^{5}	19.2^{5}	25.8^{5}	18.4^{4}	10.8^{5}	12.4^{5}	10.8^{6}	6.6^{5}	18.5^{6}	6.2^{5}	14.2^{5}	29.8^{5}	33.2^{5}	15.4^{5}	30.6^{5}	19.7^{5}	35.8^{3}	40.6^{3}	39.6^{3}	45.3^{3}	76.8^{3}	43.6^{3}
Poisson Matting	5.9	6	5.8	5.9	51.8^{6}	56.2^{6}	52^{6}	28.3^{6}	43.5^{6}	30.7^{6}	12.1^{6}	13.7^{6}	9.2^{5}	11.7^{6}	18.4^{5}	11.2^{6}	22.4^{6}	36.8^{6}	55.5^{6}	21.4^{6}	32.2^{6}	22.7^{6}	53.6^{6}	72.9^{6}	58.4^{6}	125.5^{6}	84.8^{5}	139.7^{6}

Figure 7.1: **Online benchmark.** A screenshot of our online evaluation table. The values in each cell correspond to the error generated by a specific method (rows) on a test image (columns). Moving the mouse over a specific error value displays the image of the corresponding alpha matte (leftmost image). To allow for a better inspection of the result, a zoom-in of the alpha values in the red box is shown next to it. The zoomed-in area can be easily changed by moving this box. Furthermore, we show the corresponding input image and trimap.

To select a representative test set from our full database, we applied the following strategy. Firstly, we manually assigned each image to one out of four available categories, depending on the amount of fractional alpha values in their respective ground truth matte. Then we computed error rates (mean squared error) for all images with a set of six matting algorithms (i.e. [WC07a, LRAL08, GCL^{+}06, SJTS04, CCSS01, GSAW05]). From each of the four categories, we selected those two images that were most challenging for the algorithms (i.e. images with a large average error and diverging quality of results).

To confirm that we had chosen a well-balanced subset, we compared the performance of various matting algorithms on our subset against their performance on the average subset. Therefore, we computed the average ranking of the 6 aforementioned algorithms over all possible subsets of 8 images. Indeed this ranking turned out to be identical to the one obtained on our particular subset. This suggests that we have chosen a subset that maintains similar properties as the full dataset. Furthermore, we computed the average correlation of rankings obtained from every possible subset of 8 images with the rankings on the full set, which gives a value of 0.91. This is very close to the correlation value of 0.87, which was found for our subset. Again, this demonstrates that we have chosen a well balanced subset.

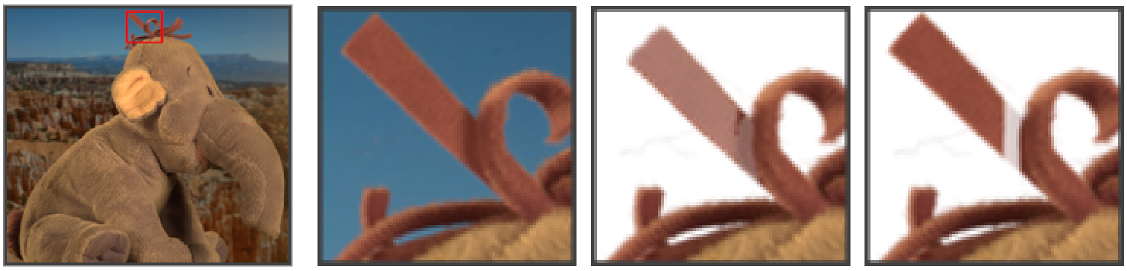
We finally downscaled the images of our dataset such that the longest image side is 800 pixels. This was done because most current matting algorithms are not capable of processing high-resolution images. Therefore, in this dissertation we restricted our evaluation (section 7.4) to low-resolution data. However, in the future we plan to complement the online benchmark with the original high-resolution images for those algorithms that can handle them.

7.3 Perceptually Motivated Error Measures

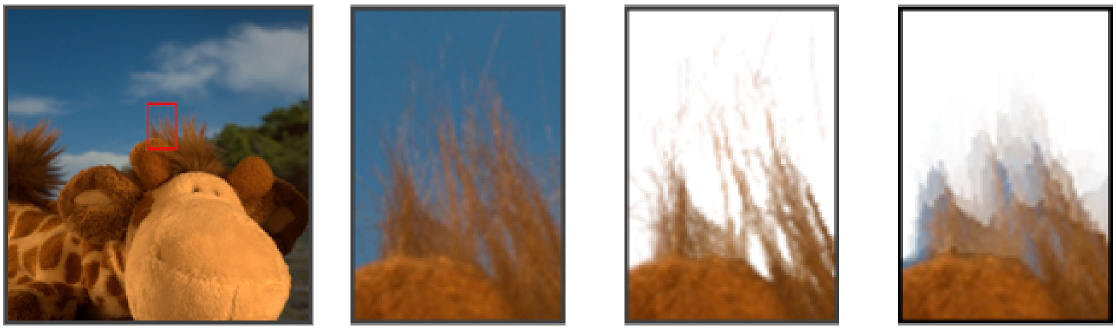
In order to quantitatively evaluate the performance of matting algorithms, their outputs (i.e. alpha mattes) have to be compared to the ground truth using an error metric. In previous work, simple metrics like the sum of absolute differences (SAD) or the mean squared error (MSE) have been used for this task. While these measures provide a good basis for comparison, they are not always correlated to the visual quality as perceived by a human observer. An example is depicted in figure 7.2, which shows two image compositions where the SAD error is not correlated to the visual quality. This motivates to study error metrics that are better suited for a perceptual comparison of matting methods.

Clearly, the development of perceptually driven distance measures depends on the target application and thus we will focus on the commonly used application scenario of compositing the extracted foreground object onto a new background (cut & paste). To further reduce the complexity, we will restrict ourselves to pasting onto a homogeneously colored background, which is an important application in the media industry (e.g. creating images for magazine covers).

Human observers judge the visual quality of image compositions by perceiving and weighing the different types of errors that appear in these images. This judgment depends on many different factors such as the color and texture of the resulting composite as well as the structure of the alpha matte. Ideally, one should learn a single visual error function over image regions that takes all these degrees of freedom into account. However, there are two major problems with this approach. Firstly, image regions that are big enough to



(a) (b) (c) SAD: 312 (d) SAD: 83



(e) (f) (g) SAD: 1215 (h) SAD: 806

Figure 7.2: **Motivation for perceptual error measures.** Two images (a; e) were cropped to give the images shown in (b; f). Matting methods have been applied to generate new compositions (c-d; g-h). In both cases, the average user ranking was exactly opposite to the error computed by SAD. This motivates to study perceptual error measures that better correlate to the visual quality. The top row (c-d) shows results for our connectivity set, and the bottom row (g-h) results for our gradient set.

preserve the context of the depicted scene, e.g. windows of size 100x100 pixels, have a very high number of potential colors and alpha values. Secondly, given the same patch of an image composite, people might disagree on the visual error. This indicates the need for a multi-modal error function. For instance, 12% of the participants in our study preferred figure 7.2 (d) over figure 7.2 (c), while 88% decided the other way round. To largely circumvent these challenges, we concentrate on developing perceptual error functions for two specific error categories for which previously used error metrics, like SAD, largely disagree with humans. In an explorative pre-study with 4 subjects (3 males and 1 female), two error categories emerged that seem to considerably degrade the visual quality of image composites: (i) *connectivity* errors, which are a result of disconnected foreground objects (for example, a disconnected piece of hair floating in the air); (ii) *gradient* errors, which are due to oversmoothing or erroneous discontinuities in the alpha matte (i.e. the gradient in the alpha matte diverges from the ground truth). Examples of each of these categories are depicted in figure 7.2.

In the remainder of this section, we first derive the visual quality of image compositions in a user study (section 7.3.1). In section 7.3.2, we design perceptual distance measures and show that their correlation to the visual quality is superior to previously used error measures, like SAD.

7.3.1 User Study

The main goal of our user study was to infer the visual quality of image compositions from human observers in the presence of connectivity and gradient artifacts.

Data

We performed our experiments on two sets of image compositions, each of them afflicted solely by either connectivity or gradient artifacts. To construct these sets, we applied a variety of matting algorithms to the input images of our ground truth database and created composites by pasting the extracted foreground object onto homogeneously colored backgrounds. We then carefully selected crops of these compositions that mainly exhibited either connectivity or gradient artifacts. The size of the crops was chosen such that they were small enough to isolate these error categories, but still big enough to provide the user with sufficient contextual information to judge their quality. In our pre-study, we found that crops with a size of about 100×100 pixels were a good trade-off between these two factors.

Compositions created from the same image crop (but with different matting algorithms) were arranged into a single test case. Figure 7.3 shows an example. To increase the number of composites per test case, we also included artificial images that we generated by interpolating some composites towards their ground truth. Note that by including these interpolations, the results of this study become more applicable to the output of future matting methods with higher quality results. From this pool of test cases, we have chosen only those examples whose composites could be sorted relatively easily according to their quality (to reduce potential ambiguities) and for which we expected traditional error measures (e.g. SAD) to diverge from the human perception. For the study we used a total number of 20 test cases (10 for each error category), with each test case consisting of 6 image compositions.

Study Procedure

The study was carried out with 17 participants (8 males and 9 females) whose ages ranged from 24 to 67 years, with an average age of 36. The study aimed to derive an ordering of the compositions associated with each test case, from the judgment of the participants. Such an ordering can be obtained by means of absolute (on a discrete scale) or relative rankings. We preferred to derive relative rankings, since they have been shown to significantly raise the agreement between users in the context of web page ranking [CBCD08]. Relative rankings can be obtained by a sequence of pairwise comparisons (the user selects one out of a pair of images) or by sorting the compositions at a glance. In our pre-study we observed that the participants preferred to rank the compositions at a glance, and therefore we decided for the following experimental setup shown in figure 7.3.

For each test case, the subjects were shown the associated 6 compositions in a list that they could interactively sort by moving the images on the screen (figure 7.3(left)). Each list element showed the original image crop (left) together with compositions on 4 homogeneously colored backgrounds (i.e. white and shades of red, green and blue). To provide the user with more contextual information, we also displayed the corresponding uncropped image (figure 7.3(right)). For every participant, the compositions in each test case were shown in random order. This was done to overcome any bias of subjects against any particular initial position of the list of images.

Prior to the study, the participants were told that they would be presented crops of photomontages that had been generated by inserting objects, extracted from a photograph, onto a single-colored background. Then we instructed the subjects to rank the results according to how realistic the image compositions appeared. The users were given the opportunity to indicate cases where two or more compositions could not be distinguished because they had the same quality. To reveal further details about the decision making process of the

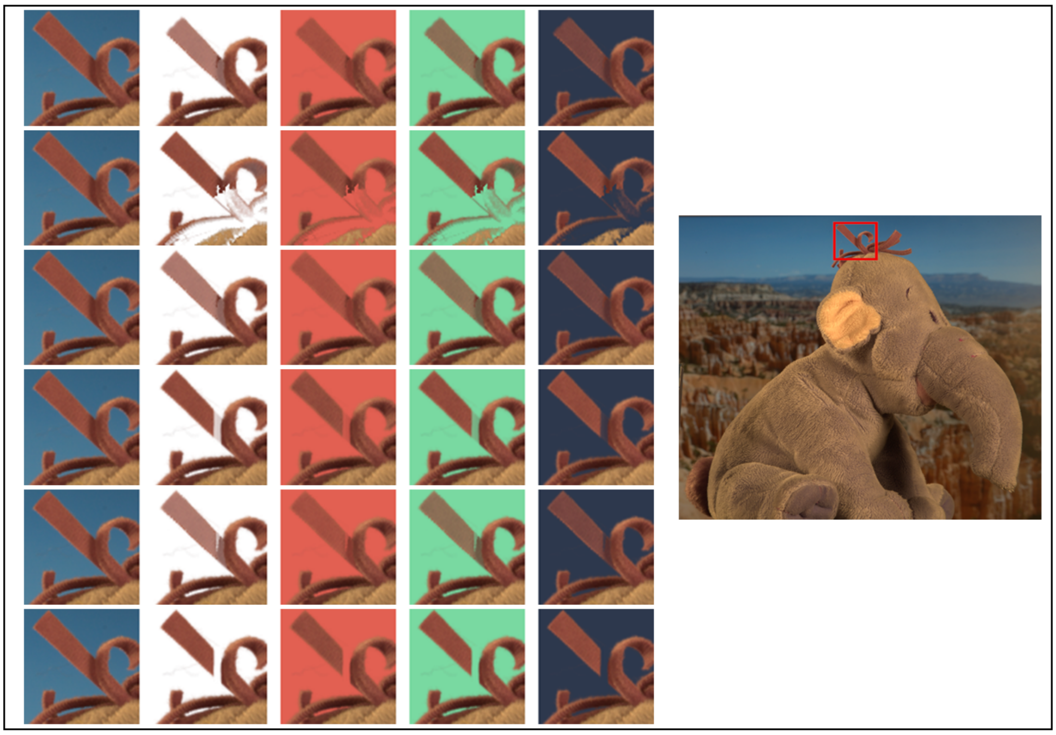


Figure 7.3: **A test case used in our study.** Explanation in text.

users, we also recorded their verbal feedback.

7.3.2 Analysis of Results

To obtain generalizable results, the study was evaluated with respect to the ranking of the "average user". In the average scores we accounted for image pairs that could not be clearly ranked (i.e. pairs for which the average ranks differed by less than 0.2) by assigning them to the same score (14% and 8% of pairs in the gradient and connectivity set were affected). To demonstrate that an analysis on the average observer basis is valid, we first analyzed the variability of the user judgments with respect to the average rankings. Then, we examined to which extent several distance measures were correlated to these average scores. Since the distance measures give absolute error values, we converted them to relative rankings beforehand. To measure the similarity between two rankings, we utilized the Kendall's τ measure [Ken55], which is commonly used in statistics for comparing the correlation of ordinal random variables [Joa02].

Agreement of Observers

The correlation of the individual participants (averaged over all test cases and users) with the average user ranking was 0.90 and 0.87 for the connectivity and gradient test set, respectively. These are reasonably high values compared to the zero coefficient that would be given to a random ranking. However, the remaining variation in the user judgments implies that even for the identical image composition, people disagreed on the visual error. This suggests that there is inherent ambiguity in the perception of errors and a single visual error function for image matting may not exist. Note that ambiguity in the perception of errors does not mean that there is no single global optimum (ground truth) for the alpha matte.

Error Measures

Our perceptual error measures are described in the following:

- **Gradient.** We tried a number of different gradient measures, including the commonly used angular error between the gradient vectors, but found the following measure to work best. The difference between the gradients of the computed alpha matte α and its ground truth α^* is defined as

$$\sum_i (\nabla\alpha_i - \nabla\alpha_i^*)^q , \tag{7.1}$$

where $\nabla\alpha_i$ and $\nabla\alpha_i^*$ are the normalized gradients of alpha at pixel i that we computed by convolving the mattes with first-order Gaussian derivative filters with variance σ (not shown in eq. (7.1)). The parameter q denotes the norm of the error metric. The values for σ and q will be defined in section 7.3.3.

- **Connectivity.** A considerable amount of work has been devoted to the problem of measuring connectivity [Ros83, VS91]. Following recent work in this area [BNG04], we define the degree of connectedness by means of connectivity in binary threshold images computed from the grayscale alpha matte.

In detail, we define the connectivity error of an alpha matte α with its corresponding ground truth α^* as

$$\sum_i \left(\varphi(\alpha_i, \Omega) - \varphi(\alpha_i^*, \Omega)\right)^p, \tag{7.2}$$

where φ measures the degree of connectivity for pixel i with alpha value α_i to a source region Ω. The parameter p denotes the norm of the error metric and its value will be defined in section 7.3.3. Consider figure 7.4, which illustrates the intensity function of a row of pixels in an alpha matte. The source region Ω is defined by the largest connected region where both the alpha matte as well as its ground truth are completely opaque (illustrated by the red line in figure 7.4). The degree of connectivity is based on the distance $d_i = \alpha_i - l_i$, where l_i is the maximum threshold level where pixel i is 4-connected to Ω (dashed line in figure 7.4). A pixel is said to be fully connected if $l_i = \alpha_i$. Finally, the degree of connectivity φ for pixel i is defined as

$$\varphi\left(\alpha_i, \Omega\right) = 1 - \left(\lambda_i \cdot \delta\left(d_i \geq \theta\right) \cdot d_i\right). \tag{7.3}$$

This means that a pixel is fully connected if $\varphi = 1$ and completely disconnected if $\varphi = 0$. The δ function enforces that very small values of d_i below θ are neglected. (The parameter θ will be defined in section 7.3.3.) We further weight d_i at disconnected pixels with their average distance λ_i to the source region:

$$\lambda_i = \frac{1}{|K|} \sum_{k \in K} dist_k(i), \tag{7.4}$$

where K is the set of discretized alpha values in the range between l_i and α_i. The function $dist_k$ gives the normalized Euclidean distance of i to the closest pixel that is connected to Ω at threshold level k. The intuition behind this is that unconnected parts that are further away from the connected region are visually more distracting.

Unfortunately, computing the connectivity under this metric is computationally rather expensive, since it requires to evaluate the function $dist_k$ at each threshold level k. To make the computation tractable, in our online evaluation system we use a slightly modified version of this metric, which neglects the distance of unconnected islands to the connected region. This was done by simply setting λ_i in eq. (7.3) to a constant value of 1.

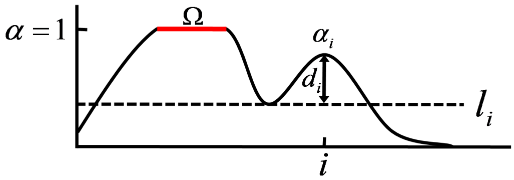


Figure 7.4: **Connectivity error.** See explanation in the text.

Data	Grad.	Conn.	MSE	SAD	User consent
Grad.	**0.75**	0.47 (0.41)	0.51	0.45	0.87
Conn.	0.40	**0.75** (0.77)	0.34	0.28	0.90

Table 7.1: **Error measure correlations.** The correlation coefficients of four error measures for the connectivity and gradient set. Correlations of the modified connectivity metric that we use for online evaluation are shown in parentheses.

Agreement of Error Measures

The agreement of our error measures on the gradient test set (first row of table 7.1) shows that the correlation of SAD and MSE with the average human observer is rather low (0.45 and 0.51, respectively). Our connectivity measure performs similarly with a correlation of 0.47. The correlation for our computationally less expensive connectivity measure (shown in parentheses in table 7.1) is 0.41. As expected our gradient measure outperforms all of them with a correlation of 0.75.

Analysis on the connectivity set (second row of table 7.1) shows that SAD and MSE exhibit an even lower correlation than on the gradient set (0.28 and 0.34) and also our gradient error (0.40) is not capable of capturing errors in the connectivity. As expected our measure for connectivity performs well with a correlation coefficient of 0.75. Interestingly, our modified connectivity metric, which neglects the distance of disconnected islands, performs even slightly better with a correlation of 0.77.

7.3.3 Choice of Parameters

We decided to choose the values for the four important parameters σ, q (eq. (7.1)), p (eq. (7.2)) and θ (eq. (7.3)) of our error measures according to their robustness and correlation with the user scores. The robustness of error measures with respect to noise in the data is a test commonly used in information retrieval [ZC06]. We added Gaussian noise (zero mean and variance σ_{noise} ranging from 0.001 to 0.005) to our alpha mattes and ranked the corrupted maps using our new perceptual error measures. We then computed the correlation coefficients between these rankings and the ones derived on undistorted data. We repeated this K times (we found $K = 200$ sufficiently large) and used the average correlation coefficient as robustness score. (The correlation coefficients range between a value -1 and 1, taking a value of exactly -1 or 1 if the data is completely uncorrelated or completely correlated, respectively.)

Let us consider figure 7.5 (left), which shows the robustness of our gradient measure for different values of the parameter σ, which is the variance of the Gaussian derivative filters used to compute the gradients. We can see that for $\sigma = 0.2$ (blue curve), the robustness (vertical axis) drops off quickly with increasing noise level. This is not surprising since a low σ makes the estimation of the gradient more sensitive to noise. For larger values of σ (1.4 and 3) the robustness is constantly high. Clearly, the choice of our parameters does not only depend on their robustness, but also on the correlation to the user scores. An example is depicted in figure 7.5 (right), which shows the correlation of the parameter σ to the user scores. We can see that although a large value of $\sigma = 3$ (green curve in figure 7.5 (left)) makes the measure robust to noise, the correlation of the gradient measure is rather low for this value. Thus we limited the parameters to a range where the error measures exhibit a robustness score of at least 0.9 and a correlation that is at worst 10% lower than its maximum value (averaged over all noise levels). Therefore a good choice would be $\sigma \in \{1.2, .., 2.0\}$, where our measure is robust and highly correlated to the user scores.

Accordingly, we can limit the remaining parameters of our error measures to the ranges $q \in \{1, .., 3\}$, $\theta \in \{0.13, .., 0.25\}$ and $p \in \{1, .., 2\}$. Finally we select the number in each range which gives the maximum correlation (i.e. $\sigma = 1.4$, $q = 2$, $\theta = 0.15$ and $p = 1$). Clearly, our approach for parameter selection assumes that the user rankings are invariant to small noise in the alpha mattes. Finally, it should be noted that instead of selecting the parameters according to the robustness and correlation to the user scores, one may also train the parameters directly from the data. However, for this purpose a larger training set would be necessary, thus we leave it for future work.

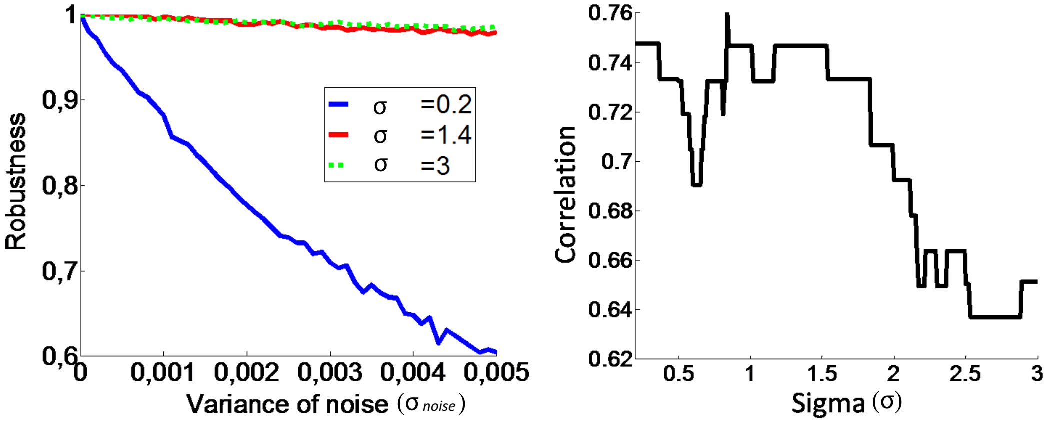


Figure 7.5: **Robustness of parameters.** See the text for explanation.

7.4 Evaluation Results

In this section, we compare our three matting approaches presented in chapters 5 and 6 against 9 algorithms that mostly represent the current state-of-the-art, namely *Bayesian Matting* [CCSS01], *Closed-form Matting* [LLW08], *Easy Matting* [GCL$^+$06], *Geodesic Matting* [BS07], *Iterative Matting* [WC05], *Poisson Matting* [SJTS04], *Random Walk Matting* [GSAW05], *Spectral Matting* [LRAL08] and *Robust Matting* [WC07a]. For all algorithms, we used the implementations of the respective authors, except for Poisson Matting [SJTS04], which we implemented ourselves. To offer a fair comparison, we set the parameters for all algorithms to the values reported in the respective papers.

7.4.1 Performance on SAD and MSE

We evaluated all of the above mentioned algorithms on our 8 test images, using three different trimaps as inputs (see section 7.1.3). We computed the accuracy of the resulting alpha matte with respect to the four error measures defined in section 7.3.2 (i.e. SAD, MSE, gradient and connectivity error). Each test case (image and trimap) gives a ranking of all algorithms. This rank, averaged over all test cases, is shown in table 7.2. Additional to the ranks of our three algorithms and its nine competing algorithms, we report the rank of our Segmentation-based Deconvolution method *without* connectivity prior in the third row of table 7.2.

One of the 8 test cases shows very large semi-transparent regions that originate from light transmitting materials (i.e. a translucent plastic bag), which largely violates the assumptions of our segmentation-based algorithms. Thus we additionally show the average ranking over the 7 remaining images in parentheses in table 7.2.

When analyzing the results with respect to the SAD and MSE error measure, we observe that all three matting methods proposed in this thesis perform better than the state-of-the-art. We also see that a segmentation-based prior (used in our Segmentation-based and Hybrid Deconvolution approaches, respectively) improves the quality of the alpha mattes. More precisely, we see that our Segmentation-based Deconvolution approach performs best on both error measures. The Hybrid Deconvolution approach shows lower performance on the MSE metric, because it can sometimes amplify strong errors in the alpha matte that is used to initialize the Hybrid Deconvolution algorithm.

The very good performance of our segmentation-based priors is even more clear if we look at the average rankings over the subset of 7 images which consists solely of foreground object that are largely opaque (shown in parentheses in table 7.2). In particular, the gap between the Segmentation-based Deconvolution method and its competitors is further increasing.

On the SAD and MSE metrics, the best performing previously proposed method is Closed-form Matting, followed by Robust Matting. We notice that the performance of Robust Matting, Bayesian Matting, Iterative Matting and Easy Matting, in comparison to the remaining algorithms, is lower than what was reported in previous evaluations [WC07b, WC07a]. These methods use a data term in their objective function, which is derived from global color models of true fore- and background regions. These data terms typically require to set a fair amount of free parameters. Hence, a potential over-fitting of these parameters to their respective test data might lead to a lower performance on our unseen data. Note that the test datasets for these methods in the original papers were mostly composed of images with smooth backgrounds, whereas our dataset contains examples of highly textured backgrounds. A detailed inspection of these data terms shows that they are fairly sensitive to the exact placement of the trimap (i.e. true fore- and background regions). This sensitivity can introduce large artifacts in the alpha matte. Pure propagation

Method	SAD	MSE	Grad.	Conn.
Segm.-based Dec. (chap. 6.3)	$\mathbf{2.7}_{1}$ ($\mathbf{2.1}_{1}$)	$\mathbf{3.0}_{1}$ ($\mathbf{2.5}_{1}$)	$\mathbf{2.1}_{1}$ ($\mathbf{1.3}_{1}$)	5.6_{5} (5.2_{5})
Hybrid Dec. (chap. 6.2)	3.2_{2} (3.1_{2})	3.4_{3} (3.3_{2})	3.5_{3} (3.2_{3})	4.4_{3} (3.6_{3})
Segm.-based Dec. without connectivity (chap. 6.3)	3.4_{3} (3.1_{2})	3.7_{4} (3.4_{3})	2.8_{2} (2.1_{2})	6.3_{7} (5.9_{6})
Impr. Color Matting (chap. 5)	3.7_{4} (4.0_{5})	3.3_{2} (3.5_{4})	4.1_{4} (4.5_{4})	4.7_{4} (4.3_{4})
Closed-form Matting [LLW08]	3.8_{5} (3.8_{4})	3.9_{5} (3.9_{5})	4.8_{5} (5.1_{5})	3.3_{2} (3.5_{2})
Robust Matting [WC07a]	5.3_{6} (5.9_{6})	5.0_{6} (5.4_{6})	5.0_{6} (5.2_{6})	7.5_{8} (7.5_{8})
Random Walk Matting [GSAW05]	8.0_{7} (7.8_{7})	8.0_{7} (7.9_{8})	8.1_{8} (8.5_{8})	$\mathbf{2.1}_{1}$ ($\mathbf{2.3}_{1}$)
Geodesic Matting [BS07]	8.5_{8} (8.6_{8})	8.7_{8} (8.9_{9})	9.8_{10} (9.8_{10})	9.2_{10} (9.5_{10})
Iterative Matting [WC05]	8.7_{9} (8.8_{9})	7.5_{9} (7.7_{7})	8.0_{7} (8.1_{7})	8.8_{9} (9.0_{9})
Easy Matting [GCL^{+}06]	9.1_{10} (9.0_{10})	10.3_{11} (10.1_{10})	10.4_{7} (10.1_{11})	9.9_{11} (10.5_{11})
Bayesian Matting [CCSS01]	10.0_{11} (10.4_{11})	9.9_{10} (10.1_{10})	10.4_{11} (10.4_{12})	11.7_{13} (12.2_{13})
Spectral Matting [LRAL08]	12.2_{12} (12.1_{12})	11.9_{12} (12.0_{12})	9.1_{11} (9.5_{9})	5.8_{6} (6.1_{7})
Poisson Matting [SJTS04]	12.2_{12} (12.1_{12})	12.4_{13} (12.3_{13})	12.9_{13} (12.9_{13})	10.6_{12} (11.4_{12})

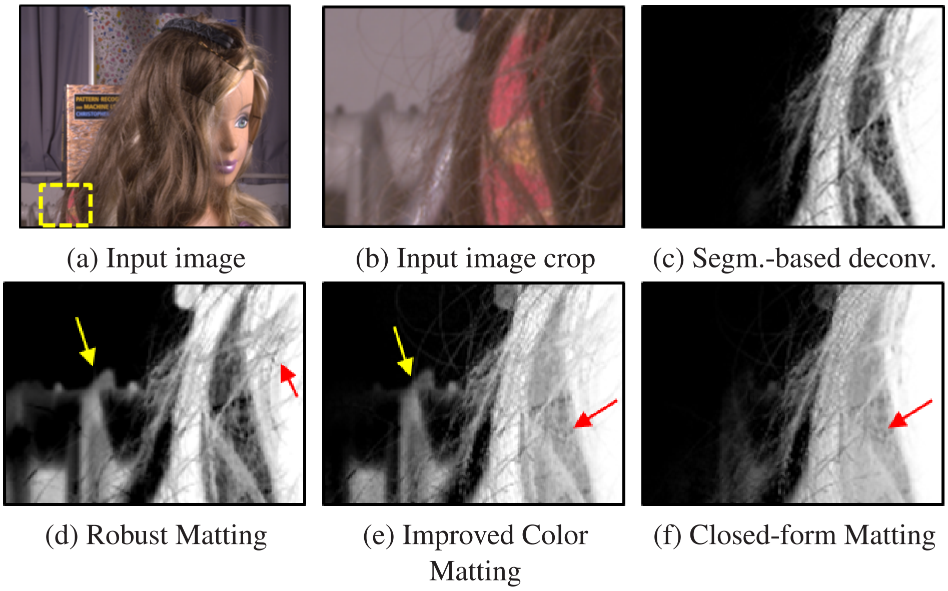
Table 7.2: **Evaluation.** The table reports the overall ranks of the different algorithms with respect to four error measures. These ranks were obtained by averaging the ranks over all test cases, i.e. all test image-trimap input pairs. In parentheses we show the rankings obtained after excluding one image (light-transmitting plastic bag) from the test set. The red lowercase numbers indicate the ordering of the algorithms in each column according to their ranking.

based approaches, like Closed-form Matting and Random Walk Matting, seem to suffer less from this problem. An exception is the propagation based Poisson Matting algorithm that performed constantly worse than its competitors, since its assumption of smooth fore- and background colors is rarely met on our dataset.

Although previously proposed methods that model the fore- and background colors rank slightly worse on our dataset, visual inspection of the results shows that these methods can sometimes overcome locally ambiguities in the fore- and background colors. For instance, Closed-form Matting (which does not have a global color model) tends to over-smooth holes in the foreground and shortens fine structures like hair. These structures were sometimes better captured by methods which have a global color model (e.g. Robust Matting).

In figure 7.6 we give an example that shows results on a crop of a challenging test image. We can see that the purely propagation-based Closed-form Matting (figure 7.6(f)) approach over-smoothed the hole in the foreground. Also our Improved Color Matting approach (figure 7.6(e)) could not recover the hole, whereas the Robust Matting approach (figure 7.6(d)) performed better. However, Robust Matting introduced large artifacts in the background. In contrast, our Segmentation-based Deconvolution approach (figure 7.6(c)) shows the fewest errors in the background and could also recover the hole in the foreground object.

Another observation is that the propagation-based Geodesic Matting [BS07] approach ranks only average among all methods. This is presumably because it was designed to run on very tight trimaps, whereas we evaluate the algorithms also on coarse trimaps. We also see that the propagation-based Spectral Matting approach shows a rather bad performance on our dataset. This might be explained by the fact that Spectral Matting is better suited for a component picking interface. Using a scribble or trimap-based interface to group the matting components might result in large errors, if the wrong matting components are grouped. Furthermore, Spectral Matting does not guarantee that the scribbles will be assigned to the user defined alpha value. Interestingly, Easy Matting, which builds on the Iterative Matting approach, shows worse results than Iterative Matting. A possible explanation is that Easy Matting changes the energy during optimization to better cope with coarse scribble input. However, this might not be suitable for our trimap input.


(a) Input image (b) Input image crop (c) Segm.-based deconv.

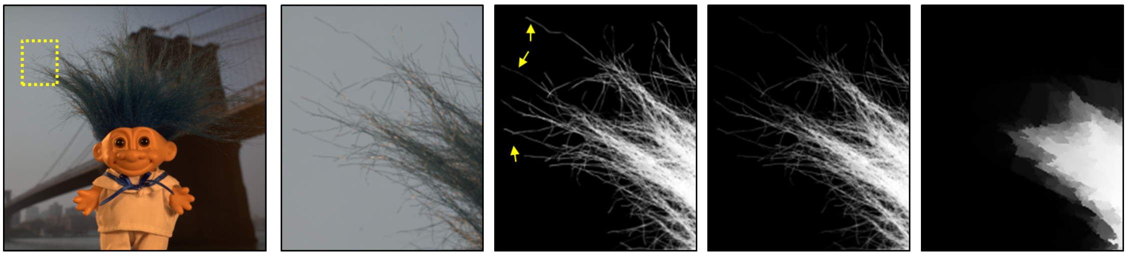
(d) Robust Matting (e) Improved Color Matting (f) Closed-form Matting

Figure 7.6: **Performance of Segmentation-based Deconvolution method.** The crop of this challenging example shows that our Segmentation-based Deconvolution approach (c) could remove background artifacts better than its competitors. This indicates that our segmentation-based prior can better resolve ambiguities in the solution space.

7.4.2 Performance on Gradient Error

When analyzing the scores with respect to the gradient error, we see that all of our proposed matting algorithms outperform the state-of-the-art. This indicates that in our results the gradient of fine structures like hair is better preserved. We also see that Closed-form Matting performs worse and is now almost on par with Robust Matting. This is because Closed-form Matting tends to attenuate or completely cuts off fine structures like hair strands in the alpha matte.

An example is shown in figure 7.7, where the Closed-form Matting approach (figure 7.7(d)) cuts off some hair of the soft toy. In contrast, our Hybrid Deconvolution method (figure 7.7(c)) could better recover the hair. We also see that Random Walk Matting (figure 7.7(e)), which performs reasonably well on the SAD metric, for this test image heavily oversmoothed the alpha matte. This is probably because its affinity function, which was originally designed for binary segmentation, oftentimes cannot recover the changes in the



(a) Input image (b) Crop of (a) (c) Hybrid Deconvolution Matting (d) Closed-form Matting (e) Random Walk Matting

Figure 7.7: **Performance on gradient error.** For the image crop in (b), our Hybrid Deconvolution method can nicely recover the fine hair strands (c), whereas the Closed-form Matting approach tends to cut off the hair (d). Random Walk Matting (e) heavily oversmoothes the hair, which is penalized by the gradient error metric.

alpha levels. This oversmoothing is, however, penalized by the gradient measure.

7.4.3 Performance on Connectivity Error

When considering the rankings based on our connectivity error measurement, we see that the Random Walk algorithm is clearly the best performer. This is not surprising, since alpha mattes generated by Random Walk are perfectly connected, i.e. they obtain a value of $\varphi = 1$ (eq. (7.3)) for each pixel. However, this does not mean that the connectivity error (which is the difference of the connectivity of the alpha matte with its ground truth) is necessarily zero, since the ground truth is usually not perfectly connected. As opposed to the connectivity error, however, the Random Walk algorithm shows quite large errors under the other metrics.

We also see that the Closed-form Matting approach takes the second place on the connectivity measure, which is presumably due to its bias towards constant solutions (see chapter 2.2.1). This has the advantage that the results show less background artifacts (which would cause a high penalty under the connectivity metric), but comes at the cost of oftentimes cutting off fine structures like hair.

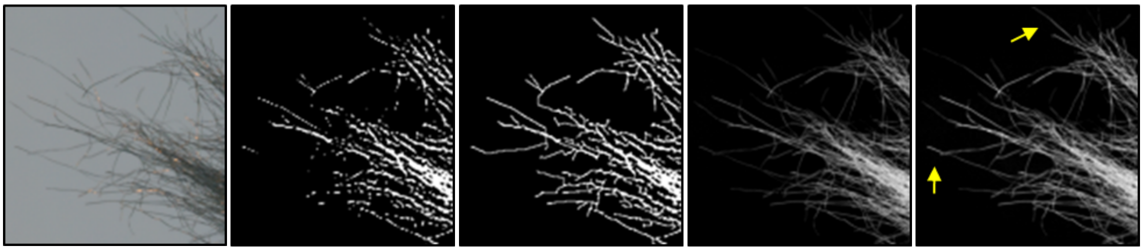
Our three approaches (rows 1, 2 and 4 in table 7.2) rank directly behind Random Walk and Closed-form Matting with respect to the connectivity measure. This is because our algorithms might erroneously fit fractional alpha values to the background texture. On the other hand, our methods can detect the alpha matte of thin structures that might be missed by the more "conservative" Random Walk and Closed-form methods.

Intuitively, this should be less of a problem for our Segmentation-based Deconvolution approach, which uses a connectivity prior on the underlying binary segmentation. However, it ranks worst out of our three algorithms on the connectivity metric. This is mainly because our connectivity prior might connect larger, originally disconnected regions to the foreground object with a small one-pixel-wide connecting path. In the final matte the alpha values along this path can be attenuated, hence regions which are connected in the binary segmentation appear disconnected (because of low alpha values on the connecting path) in the final alpha matte. (In practice also some background artifacts can be connected to the foreground in the same way, although they appear disconnected in the final matte.) To fix this problem, one could define a minimal width of the connecting path as in [VKR08]. However, a good single parameter for the minimum width is hard to define, since it depends on the local characteristics of the foreground object. For instance, hair strands should be connected by a thin path, whereas the feet of a human should be connected to the body by a wide path. In the future we would like to investigate methods that automatically find a minimum width for each connecting path.

To show that our connectivity prior is nonetheless important for the performance of our algorithm, we also report the performance our Segmentation-based Deconvolution method *without* connectivity prior (see third row of table 7.2). We can see that our connectivity prior considerably helps to improve the performance of our Segmentation-based Deconvolution method on all four error measures. This is because our connectivity prior closes gaps between fine hair stands and removes some undesired artifacts in the background (see figure 7.8 for an example).

7.5 Summary

We have presented a new benchmark test for the evaluation of image matting algorithms that is freely available on the web at *www.alphamatting.com*. We have shown that the matting algorithms presented in this thesis considerably improve on the state-of-the-art



(a) Image crop (b) Segmentation *without* connectivity prior (c) Segmentation *with* connectivity prior (d) Alpha from (b) (e) Alpha from (c)

Figure 7.8: **Advantages of the connectivity prior.** For the crop of the input image in (a), the segmentation without connectivity prior (b) shows many disconnected hair strands. The result with connectivity prior (c) could better recover the hair strands. The alpha matte computed using the disconnected segmentation as prior is shown in (d). Using the connected segmentation as prior yields alpha mattes where some hair strands are better preserved (e). (Arrows point to improvements.)

matting algorithms on our challenging dataset. Furthermore, our benchmark revealed failures of previously proposed algorithms on images containing highly textured backgrounds, and images where the fore- and background cannot be differentiated on the basis of color alone. Finally, we proposed and validated perceptually motivated error measures based on the connectivity and gradient of the alpha matte. We hope that our work will encourage researchers to develop new matting algorithms that pay more attention to visually important features such as spatial connectivity.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

In this thesis we have focused on interactive image matting, which is the problem of estimating a transparency map from a single natural image with the help of user interaction. In particular, we have investigated three important aspects of interactive image matting which are (i) providing a suitable way for user interaction; (ii) formalizing a good cost function that defines the goodness of an alpha matte; and (iii) the evaluation of matting algorithms.

The key idea that distinguishes this thesis from previous alpha matting work is that we have explicitly considered the image formation process. In particular, we have modeled the alpha matte as the convolution of a binary segmentation with the point spread function of the camera. We have shown that our alpha matting approach based on this model outperforms current state-of-the-art matting methods. Our work constitutes an important step towards unifying two areas of research: binary segmentation and alpha matting. We believe that in the near future commercial matting methods will be based on our model. In the following our contributions are discussed in more detail.

To provide a good way for user interaction, we have developed a fast algorithm that automatically generates a trimap from only a few user-defined scribbles that are placed on the input image. Our approach works by predicting the structure of the binary segmentation that underlies the alpha matte. We infer this binary structure using several image cues, like color and image edges. We have shown that our method is fast and produces results that exceed the quality of previous trimap extraction algorithms.

Given a trimap, we have introduced an approach that extracts an alpha matte by accurately modeling the fore- and background colors at each pixel in the unknown region of the trimap. The novelty of this "Improved Color Matting" approach was to exploit information from global color models to find better local estimates for the true fore- and background colors.

An important contribution was to further enhance the quality of the alpha mattes generated by this Improved Color Matting approach by incorporating a new prior which is based on the image formation process. In particular, we have modeled the prior probability of an alpha matte as the convolution of a binary segmentation with the point spread function of the camera. We have proposed two new approaches that can recover the prior model, given an approximate alpha matte. By incorporating the resulting prior model into our Improved Color Matting method, we are able to generate results that outperform current state-of-the-art matting methods.

Finally, we have introduced a new benchmark for the evaluation of image matting algorithms that is available to the research community on the web at www.alphamatting.com. We evaluated our matting algorithms on our challenging dataset and showed that they compare favorably to the current state-of-the-art. An important contribution of our work was the proposal and validation of perceptually motivated error measures based on the connectivity and gradient of the alpha matte. To the best of our knowledge, this was the first study that validated error measures for alpha matting. We hope that our work will encourage researchers to develop new matting algorithms that pay more attention to visually important features such as connectivity.

8.2 Future Research Topics

Although we have demonstrated that our methods generate very accurate results, our algorithms could be further improved in some respects. The most promising directions for future research are listed below.

- One limitation of our segmentation-based matting approaches is that we model the Point Spread Function (PSF) as a spatially constant kernel. In images where the PSF varies, e.g. due to depth variations, a spatially constant PSF is an oversimplification. However, recovering a spatially varying PSF, given only a single natural image as input is not straightforward and left for future work.

- Our segmentation-based matting approaches infer the binary segmentation from the edges in the alpha matte. Thus our approaches can correctly recover the segmentation if the PSF is a kernel with a single peak, which is mostly true for defocus blur or slight motion blur. However, strong motion blur can lead to blur kernels with multiple peaks, which cannot be handled by our approaches. Hence other methods could be investigated in the future which can overcome this limitation.
- We derive the binary segmentation that underlies the alpha matte from an initial approximation of the matte, computed with our Improved Color Matting algorithm. Clearly, the quality of the computed binary segmentation depends on the initial (usually imperfect) alpha matte. A promising direction of future research would be to infer the underlying binary segmentation directly from the input image.
- Future work could also concentrate on establishing more complex perceptual measures that take into account other factors such as color and texture of the image. Such an error function would be highly desirable, since it could be used by machine learning methods to train the parameters of alpha matting algorithms. However, more research is needed, since results of our user study indicate that the visual perception of errors is ambiguous and thus a single error function might be hard to establish.

Bibliography

[AFM98] N. Asada, H. Fujiwara, and T. Matsuyama. Seeing behind the scene: Analysis of photometric properties of occluding edges by the reversed projection blurring model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):155–167, 1998.

[BGRR09] M. Bleyer, M. Gelautz, C. Rother, and C. Rhemann. A stereo approach that handles the matting problem via image warping. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 501–508, 2009.

[BJ01] Y. Boykov and M.P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *IEEE International Conference on Computer Vision*, pages 105–112, 2001.

[BK00] S. Baker and T. Kanade. Limits on super-resolution and how to break them. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 372–379, 2000.

[BK04] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9), September 2004.

[BNG04] U. Braga-Neto and J. Goutsias. Grayscale level connectivity: Theory and applications. *IEEE Transactions on Image Processing*, 13(12):1567–1580, 2004.

[Bri99] R. Brinkmann. *The Art And Science Of Digital Compositing*. Morgan Kauffman, 1999.

[BS07] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[BSL$^+$07] S. Baker, D. Scharstein, J.P. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[BW80] M. Born and E. Wolf. *Principles of optics*. Pergamon Press, 1980.

[CBCD08] B. Carterette, P.N. Bennett, D.M. Chickering, and S.T. Dumais. Here or there. In *European Conference on Information Retrieval*, volume 4956, pages 16–27, 2008.

[CCGW00] T. Chen, P. Catrysse, A.E. Gamal, and B. Wandell. How small should pixel size be? In *Proceedings of the Society of Photographic Instrumentation Engineers (SPIE)*, volume 3965, pages 451–459, 2000.

[CCSS01] Y.Y. Chuang, B. Curless, D.H. Salesin, and R. Szeliski. A Bayesian approach to digital matting. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 264–271, 2001.

[CDGE02] A. Cavallaro, E. Drelie-Gelasca, and T. Ebrahimi. Objective evaluation of segmentation quality using spatio-temporal context. In *IEEE International Conference on Image Processing*, volume 3, pages 301–304, 2002.

[Coo07] J. D. Coombs. A novel defocus blurring model of layered depth scenes for computational photography. Master's thesis, University of Illinois at Urbana-Champaign, 2007.

[Fox01] A.M. Fox. *Optical properties of solids*. Oxford University Press, 2001.

[FS07] P. Favaro and S. Soatto. *3-D Shape Estimation and Image Restoration*. Springer-Verlag, 2007.

[FSH$^+$06] R. Fergus, B. Singh, A. Hertzmann, S.T. Roweis, and W.T. Freeman. Removing camera shake from a single photograph. *SIGGRAPH*, 25(3):787–794, 2006.

[GCL$^+$06] Y. Guan, W. Chen, X. Liang, Z. Ding, and Q. Peng. Easy matting: A stroke based approach for continuous image matting. In *Eurographics*, pages 567–576, 2006.

[GGL08] A. Gijsenij, T. Gevers, and M.P. Lucassen. A perceptual comparison of distance measures for color constancy algorithms. In *European Conference on Computer Vision*, volume 5302, pages 208–221, 2008.

[GSAW05] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann. Random walks for interactive alpha-matting. In *IASTED International Conference on Visualization, Imaging and Image Processing*, pages 423–429, 2005.

[HN04] X. He and P. Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.

[Inc08] Canon Inc. EOS digital for professionals brochure 2008. http://downloads.canon.com/cpr/software/camera/EOS_for_Pros_2008.pdf, 2008.

[Ish03] H. Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, 2003.

[Jia07] J. Jia. Single image motion deblurring using transparency. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[JK05] O. Juan and R. Keriven. Trimap segmentation for fast and user-friendly alpha matting. In *Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, volume 3752, pages 186–197, 2005.

[JMA06] N. Joshi, W. Matusik, and S. Avidan. Natural video matting using camera arrays. *SIGGRAPH*, 25(3):779–786, 2006.

[Joa02] T. Joachims. Optimizing search engines using clickthrough data. In *ACM Conference on Knowledge Discovery and Data Mining*, pages 133–142, 2002.

[JSK08] N. Joshi, R. Szeliski, and D.J. Kriegman. PSF estimation using sharp edge prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[Kat02] M. Katz. *Introduction to geometrical optics*. World Scientific, 2002.

[KB05] V. Kolmogorov and Y. Boykov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *IEEE International Conference on Computer Vision*, volume 1, pages 564–571, 2005.

[KBR07] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[Ken55] M. Kendall. *Rank Correlation Methods*. Hafner, 1955.

[KR07] V. Kolmogorov and C. Rother. Minimizing non-submodular functions with graph cuts - a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1274–1279, 2007.

[KT06] P. Kohli and P. Torr. Measuring uncertainty in graph cut solutions. In *European Conference on Computer Vision*, pages 30–43, 2006.

[KT07] P. Kohli and P. Torr. Dynamic graph cuts for efficient inference in Markov Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2079–2088, 2007.

[Lev06] A. Levin. Blind motion deblurring using image statistics. In *Conference on Neural Information Processing Systems*, pages 841–848, 2006.

[LFDF07] A. Levin, R. Fergus, F. Durand, and W.T. Freeman. Image and depth from a conventional camera with a coded aperture. *SIGGRAPH*, 26(3), 2007.

[LLW06] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 61–68, 2006.

[LLW08] A Levin, D Lischinski, and Y Weiss. A closed form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242, 2008.

[LRAL08] A. Levin, A. Rav-Acha, and D. Lischinski. Spectral matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10), 2008.

[LWDF09] A. Levin, Y. Weiss, F. Durand, and W.T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1964–1971, 2009.

[MB95] E.N. Mortensen and W.A. Barrett. Intelligent scissors for image composition. *SIGGRAPH*, pages 191–198, 1995.

[MLS06] S. McCloskey, M. Langer, and K. Siddiqi. Seeing around occluded objects. In *International Conference on Pattern Recognition*, volume 1, pages 963–966, 2006.

[MMP$^+$05] M. McGuire, W. Matusik, H. Pfister, J. F. Hughes, and F. Durand. Defocus video matting. *SIGGRAPH*, 24(3):567–576, 2005.

[MYT95] T. Mitsunga, T. Yokoyama, and T. Totsuka. Autokey: Human assisted key extraction. *SIGGRAPH*, pages 265–272, 1995.

[NL09] S. Nowozin and C.H. Lampert. Global connectivity potentials for random field models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 818–825, 2009.

[OW04] I. Omer and M. Werman. Color lines: Image specific color representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 946–953, 2004.

[PD84] T. Porter and T. Duff. Compositing digital images. *SIGGRAPH*, 18(3):253–259, 1984.

[PV08] B. Peng and O. Veksler. Parameter selection for graph cut based image segmentation. In *British Machine Vision Conference*, volume 1, pages 153–162, 2008.

[RKB04] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - Interactive foreground extraction using iterated graph cuts. *SIGGRAPH*, 23(3):309–314, 2004.

[Ros83] A. Rosenfeld. On connectivity properties of grayscale pictures. *Pattern Recognition*, 16(1):47–50, 1983.

[RRRAS08] C. Rhemann, C. Rother, A. Rav-Acha, and T. Sharp. High resolution matting via interactive trimap segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[RT00] M.A. Ruzon and C. Tomasi. Alpha estimation in natural images. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 15–25, 2000.

[SB96] A.R. Smith and J.F. Blinn. Blue screen matting. *SIGGRAPH*, pages 259–268, 1996.

[SJA08] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *SIGGRAPH*, 27(3), 2008.

[SJTS04] J. Sun, J. Jia, C.K. Tang, and H.Y. Shum. Poisson matting. *SIGGRAPH*, 23(3):315–321, 2004.

[SLKS06] J. Sun, Y. Li, S. B. Kang, and H. Y. Shum. Flash matting. *SIGGRAPH*, 25(3):772–778, 2006.

[SM00] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[SRR09] D. Singaraju, C. Rother, and C. Rhemann. New appearance models for image matting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 659–666, 2009.

[SS02] 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.

[SXJ07] Q. Shan, W. Xiong, and J. Jia. Rotational motion deblurring of a rigid object from a single image. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[VKR08] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[VS91] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.

[VS02] A. Vasilevskiy and K. Siddiqi. Flux maximizing geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1565–1578, 2002.

[WAC07] J. Wang, M. Agrawala, and M. F. Cohen. Soft scissors : An interactive tool for realtime high quality matting. *SIGGRAPH*, 26(3):9, 2007.

[WC05] J. Wang and M. F. Cohen. An iterative optimization approach for unified image segmentation and matting. In *IEEE International Conference on Computer Vision*, volume 2, pages 936–943, 2005.

[WC07a] J. Wang and M. F. Cohen. Optimized color sampling for robust matting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[WC07b] J. Wang and M.F. Cohen. Image and video matting: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(2):97–175, 2007.

[Wei99] Y. Weiss. Segmentation using eigenvectors: A unifying view. In *IEEE International Conference on Computer Vision*, volume 2, pages 975–982, 1999.

[WFZ02] Y. Wexler, A. Fitzgibbon, and A. Zisserman. Bayesian estimation of layers from multiple images. In *European Conference on Computer Vision*, pages 487–501, 2002.

[Wri06] S. Wright. *Digital compositing for film and video, Second Edition*. Focal Press, 2006.

[YS03] S.X. Yu and J. Shi. Multiclass spectral clustering. In *IEEE International Conference on Computer Vision*, volume 1, pages 313–319, 2003.

[ZC06] Y. Zhou and W.B. Croft. Ranking robustness: a novel framework to predict query performance. In *ACM Conference on Information and Knowledge Management*, pages 567–574, 2006.

[ZKY$^+$08] Y. Zheng, C. Kambhamettu, J. Yu, T. Bauer, and K. Steiner. Fuzzymatte: A computationally efficient scheme for interactive matting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.