Automatic Stereoscopic 3D Chroma-Key Matting Using Perceptual Analysis and Prediction

by

Ling Yin

Thesis submitted to the Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements for the M.A.Sc. degree in
Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

This research presents a novel framework for automatic chroma keying and the optimizations for real-time and stereoscopic 3D processing. It first simulates the process of human perception on isolating foreground elements in a given scene by perceptual analysis, and then predicts foreground colours and alpha map based on the analysis results and the restored clean background plate rather than direct sampling. Besides, an object level depth map is generated through stereo matching on a carefully determined feature map. In addition, three prototypes on different platforms have been implemented according to their hardware capability based on the proposed framework. To achieve real-time performance, the entire procedures are optimized for parallel processing and data paths on the GPU, as well as heterogeneous computing between GPU and CPU. The qualitative comparisons between results generated by the proposed algorithm and other existing algorithms show that the proposed one is able to generate more acceptable alpha maps and foreground colours especially in those regions that contain translucencies and details. And the quantitative evaluations also validate our advantages in both quality and speed.

Résumé

La thèse présentée ici concerne un nouveau cadre pour l'incrustation automatique et les optimisations pour la 3D stéréoscopique et le traitement en temps réel. Ce cadre simule d'abord la perception humaine comme processus d'adaptation pour séparer les objets au premier-plan à partir de la scène à l'aide d'analyse perceptive et de restauration du arrière-plan. Ensuite il prédit des couleurs et des facteurs alpha de mélange du premier-plan en fonction des informations perceptuelles plutôt que des échantillons directs. En outre, l'estimation de la correspondance stéréo dense est effectuée sur des paires d'image stéréoscopique. Par ailleurs, trois prototypes sont implémentés en accord avec leurs capacités hardware. Afin de parvenir aux traitements en temps réel, les procédures sont optimisées pour le traitement parallèle dans le processeur graphique et l'informatique hétérogène entre le processeur graphique et l'unité centrale de traitement. Les comparaisons qualitatives entre l'algorithme proposé et des algorithmes existants montrent que les couleurs et les facteurs alpha de mélange du premier-plan généré par l'algorithme proposé sont les meilleurs sur les régions qui sont transparentes ou qui ont des détails. Et les évaluations quantitatives valident nos avantages en matière de vitesse et de qualité.

Acknowledgements

This thesis is based on research conducted during my graduate study at the University of Ottawa. The work presented here is the result of highly collaborative efforts and I owe thanks to the people I have worked with.

I take this opportunity to express my heartfelt gratitude to Professor Jiying Zhao, for his guidance and providing me the chance to perform the research I am interested in. I greatly appreciate his patience and confidence in my research abilities.

I would like to give my sincere thanks to Watze and my next-of-kin for their love and support. I am also grateful to Wenyi Wang and Yu Zhang, my lab colleagues, for their suggestions, and sharing of knowledge and experience with me, as well as the encouragements they give.

Dedication

This thesis is dedicated to my loved ones.

Table of Contents

List of Tables

List of Figures

Nomenclature

Abbreviations

2D	Two Dimensional
3D	Three Dimensional
Arch.	Architecture
Bg.	Background
CAM	Colour Appearance Model
CIE	Commission Internationale de l'Eclairage (the International Commission on Illumination)
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
EXIF	Exchangeable Image File-format for digital still cameras
FP32	Single-Precision Floating-Point (32-bit)
Fg.	Foreground
FOV	Field-Of-View
GPU	Graphics Processing Unit
HD	High Definition

HDR	High Dynamic Range
HSL	A spherical coordinate notation by defining polar angle H as Hue, polar radial distance S as Saturation, and cylindrical axis L as Lightness
IPT	A Cartesian coordinate notation by defining I as luminance, P and T are two chrominance components
ms	millisecond
MSE	Mean Squared Error
Min.	Minimum
OpenGL	Open Graphics Library
OpenGL ES	Open Graphics Library for Embedded Systems
RGB	A Cartesian coordinate notation by defining R, G, and B as three primary colour lights – Red, Green, and Blue respectively
SAD	Sum of Absolute Differences
SMPTE	Society of Motion Picture and Television Engineers
UCS	Uniform Colour Space
VFX	Visual Special Effects
VS	Vertex Shader, a type of computer program running on GPU for vertex processing
XYZ	A Cartesian coordinate notation by defining Y as luminance, Z is quasi-equal to blue stimulation (or the S-cone response), and X is a linear combination of cone response curves chosen to be nonnegative
YUV	A Cartesian coordinate notation by defining Y as brightness, U and V are two chrominance components

Chapter 1

Introduction

1.1 Keying, Matting, and Compositing

Image keying, or video keying, is a category of techniques that merge multiple images or image sequences by switching among them based on the information attached to each picture element (e.g. pixel, in digital imaging) known as the 'key' [1]. The well-known case – chroma keying – uses the chrominance information as the 'key' and plays an indispensable role in today's image editing and video production applications. Utilizing 'key' is not only meaningful in the image and video processing scenarios, but also provides an extensive spectrum for other computer vision fields such as augmented reality applications which use depth information as the key, and ray tracing applications which use surface normal as the key.

The chroma keying is also known as the chroma-key compositing. It is the most widely used compositing technique in photography process, filmmaking, and television production industries because of its efficiency in replacing a near-monochromatic

background and particularly suitable for scenarios like virtual set compositing, film post-production, and even augmented reality compositing, as depicted in Figure 1.1. In the virtual set compositing example, it removes green background and composites the anchorwomen into the virtual set. In the file post-production example, the scene was shot with green screen and post-composited with the computer generated 3D building model. In the augmented reality example by Shoji, Michihiko, and Miura [2], the green covered robot is replaced with humanoid appearance and composited into an interactive session with human through head mounted display.

Figure 1.1: Examples of chroma keying applied in virtual set compositing (left, green screen plate courtesy of Hollywood Camera Work [3]), film post-production (middle, images courtesy of Moving Picture Company, ©DreamWorks, Paramount, and Warner Bros), and augmented reality (right, image courtesy of Shoji et al. [2]).

The underlying principals of chroma keying are alpha blending and alpha matting. In 1984, Thomas Porter and Tom Duff discussed the arithmetic for compositing digital images [4], which introduced the alpha channel and the basis of alpha blending for computer graphics. Subsequently, electronic chroma-keyers all use the alpha blending factor to represent their matte, and thus alpha matte became a popular terminology among the film and video production processes. Though in the field of computer graphics there are different alpha blending operations for different overlay mixing effect, the most widely used one is formulated as Equation 1.1, which generates the

composited colour $\mathbf{C}_p$ of pixel p by mixing source colour $\mathbf{S}_p$ and destination colour $\mathbf{D}_p$ with blending factor α_p in a linear combination manner:

$$\mathbf{C}_p = \alpha_p \mathbf{S}_p + (1 - \alpha_p)\mathbf{D}_p \tag{1.1}$$

Superimposing the foreground image onto background image can be generalized to alpha blending problem in most cases due to the additive property of physical light as illustrated in Figure 1.2, where the so-called translucent subject does not contain physical translucency or its dominant colour appearance does not mainly depends on the colour of light transported through the translucent medium.

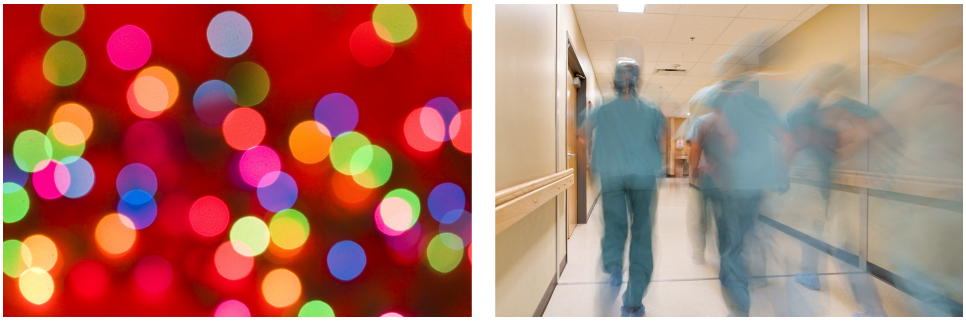


Figure 1.2: Illustration of additive colour mixing in means of defocus blur (left) and motion blur (right). As is observed, additive mixing of different colours always increase the brightness of the darker colour because multiple light source are projected on the same region and the imager or eye received a weighted sum of their energy.

In the case of foreground subjects contain physical translucency and its dominant colour appearance mainly depends on the light transported through the translucent medium, the alpha blending does not hold. The translucent medium functions as a filter that filters out partial spectrum, as illustrated in Figure 1.3. Such situation is considered beyond the scope of this research because of the physical limitation underlying the application, and which is usually avoided in real world application.

Figure 1.3: Illustration subtractive colour mixing in means of colour filtering, and selective reflection. Wine glasses with rainbow coloured liquids shot against white, green, and blue background. The colour liquids act as optical colour filters pass light of particular wavelength range. The green and blue background themselves selectively reflect light of particular wavelength range. As is observed, subtractive mixing of different colours always reduces the brightness because partial energy is absorbed and/or reflected.

Therefore, by defining foreground colour **F** as the source colour and pixel-wise background colour **B** as the destination colour, the approximate mixing colour as an image representation **I** can be calculated with a proper blending factor by applying

Equation 1.1, as shown in Equation 1.2, illustrated in Figure 1.4.

$$\mathbf{I} = \alpha\mathbf{F} + (1 - \alpha)\mathbf{B} \tag{1.2}$$

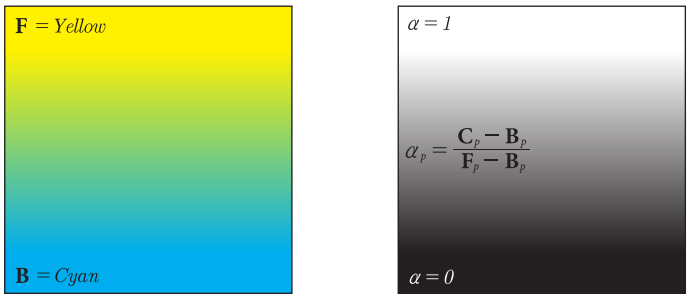


Figure 1.4: Illustration of alpha matting problem.

However, oftentimes the α and $\mathbf{F}$ are not known in prior, and thus the matting process is needed to extract the alpha matte and estimate foreground colours from the observation. In an ideal chroma-key setting, there would be significant difference between possible foreground colours and the background colour. But in real-world productions the range of foreground colours may overlap the range of background colours [5], and the background colours can be variant in luminance. As a result, $\mathbf{I}$ is the only known variable in Formula 1.2 and hence the estimation of α and $\mathbf{F}$ become comparably harder.

1.2 3D and Stereoscopy

In the mean time, 3D imaging and display technologies have been rapidly developing and becoming the future of motion picture and television. Today's 3D systems in widespread deployment utilize stereoscopic vision to capture and represent visual

contents, of which the underlying principle is 'stereopsis' – "the mind perceives an object of three dimensions by means of the two dissimilar pictures projected by it on the two retinæ" – as first defined and researched in a scientific manner by Wheatstone in 1838 [6]. Modern 'stereoscopes' have evolved out of the scope form by means of micro lens, parallax barrier or multiplexing, but all of them generate stereopsis as their ancestor did – showing two images with binocular disparity, as illustrated in Figure 1.5.

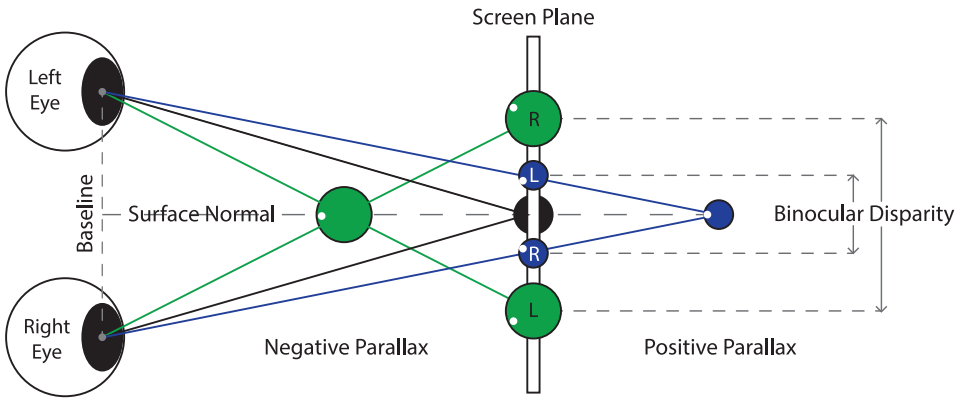


Figure 1.5: Illustration of binocular disparity and parallax.

A stereoscopic vision can provide the viewers with more perceivable information and enriches the viewing experience. Considerable adaptions from 2D to 3D have been made in the last decade for the creation of stereoscopic visual contents, though most of which, however, are focused on the post-production phase and involve a lot of manual adjustments. Today, some films are shot in stereoscopic 3D directly, but more films are still shot in 2D and are converted into stereoscopic 3D during their post-production due to budget constraints. For the 2D-to-Stereoscopic 3D conversion, the approaches can be classified into three schemes – manual, human-assisted and automatic conversion according to Zhang, Vazquez, and Knorr [7]. Both workflows

depend on 3D fixes, stereo matching, and depth grading as shown in Figure 1.6 that are also used in the industry-loved Technicolor's 3D post-production workflow [8] and The Foundry Visionmongers' stereoscopic workflow [9].

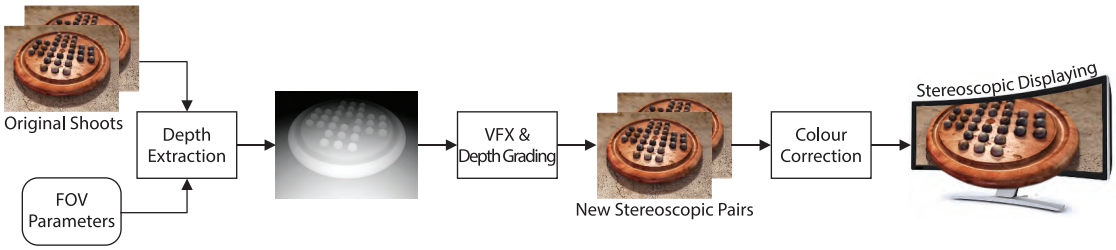


Figure 1.6: Illustration of stereoscopic 3D post-production workflow.

1.3 Existing Problems

The stereoscopic 3D chroma keying concept is relatively new to the industry, market, and academia. A common view is that both chroma keying and stereoscopy are for serving the augmented reality multimedia content. While stereoscopic 3D post-production workflows are applied to more and more films, few comparable live-production solutions have been released in the commercial market for television industry. The increasing demand of high quality visual experience, as well as the expanding market of stereoscopic 3D contents and apparatus push the live-production techniques to become more sophisticated and cost-effective. Foremost among them is the chroma keying which is vital to most visual mixing effects, and then the stereo matching.

As many television viewers may notice, sometimes there are artifacts like colour edges in those news, weather or other programs utilizing chroma keying technique. In fact, the presenters oftentimes lose their hair details, and they avoid wearing or using

translucent objects since most of the current chroma keying algorithms cannot well handle translucency. According to the Online Benchmark for Image Matting [10], several methods have achieved very high quality on generating the alpha matte for manually segmented image with arbitrary background. In a chroma keying setting, the background is more coherent than the ones of test images in the Online Benchmark for Image Matting. We believe that the chroma keying problem is a special case of alpha matting. Since those methods are not optimized for chroma keying, few of them claimed real-time performance even on the standard-definition test set, and their mathematically reasonable results may be even worse than state-of-the-art commercial chroma-keyers. Television productions such as news broadcasting and weather forecasting require a real-time process known as the live-production to composite the broadcaster(s) onto other scenes or into a virtual studio. To such live-production applications, chroma keying is worthless if impossible to achieve real-time performance. Even if the quality is high, if the viewer cannot observe a continuous context through the time, the information is hard to be comprehended. Furthermore, a reasonable foreground restoration from the mixing with background is also important to visual perception. In addition, from the operator view, a chroma keying system is better to be highly automatic than requiring miscellaneous manual adjustments.

When it comes to stereoscopic 3D, the time-consuming workflow introduced in the previous section is an even worse bottleneck in live-production. Powerful computing platforms may provide the potential to solve the problem, but the resources to fulfill the two demands are limited due to technology and budget. Practically, no network or studio would use multiple powerful mainframes to process state-of-the-art algorithms. Industry-leading solutions such as the Unreel from Ross Video [11] still focuses on

monocular 3D augmented reality due to considerations in cost-efficiency and the acceptance of stereoscopic television production. In their demonstrative videos [12] they dodged complex scenarios which include fine details and translucency but still exposed some quality issues because of their sub-optimal chroma-keying. Other notable solutions from SONY [13] and Rohde & Schwarz DVS [14] emphasize their capabilities on stereoscopic features like baseline alignment, disparity correction, and scenes integration, while their chroma keying part – as an essential mixing effect – seems to be independent from their 2D to 3D evolution and has not been improved much since its 2D epoch. Their problematic chroma keying inconsistency between views may cause binocular rivalry [15].

1.4 Objectives and Contributions

The increasing demand of high quality visual experience, as well as the expanding market of stereoscopic 3D contents and apparatus push the live-production techniques to become more sophisticated and cost-effective. Foremost among them is the chroma keying process. The objective of this research is to enhance the view experience of stereoscopic productions by the means of refining and adapting chroma keying composite for stereoscopic context. More specifically, it should not only process chroma keying on better than existing chroma-keyers in real-time, but also shows the ability on stereoscopic 3D capabilities and extensions – capabilities that can double the processing rate thus a pair of stereoscopic views can be processed simultaneously in their original frame interval, extensions that depth information can be extracted for enhanced visual effects and augmented reality.

The research in this thesis proposed a novel and stereoscopic-optimized framework for automatically processing stereoscopic 3D chroma keying based on human perception on colours and depth. Three prototypes on iPad, embedded development board, and personal computer have also been implemented and the entire procedures are optimized for different hardware to achieve different levels of real-time performance according to industrial standards. The qualitative and quantitative evaluations validate the advantages of the proposed framework and implementations in both quality and speed.

In addition, the following one paper has been published as a publication resulting directly from this research:

- L. Yin and J. Zhao, "Real-Time Automatic Chroma-Key Matting Using Perceptual Analysis and Prediction," in *Proceedings of the 27th IEEE Canadian Conference on Electrical Computer Engineering*, Toronto, ON, Canada, 2014, pp. 1–4 [16]

1.5 Organization of Thesis

The next chapter overviews the history of matting problem, the state-of-art chroma keying and alpha matting methods in both industry and academia, as well as the stereo matching problem. Then chapter 3 elaborates the proposed mechanism with illustrative intermediate results shown in each step, and chapter 4 describes the GPU optimized implementation. The final results and comparisons with other state-of-the-art chroma-keyers and alpha matting methods are given in chapter 5. The research work is summarized in chapter 6, which also concludes the research and discusses about the potential future work.

Chapter 2

Literature Review

In stereoscopic 3D production process, chroma keying is still a fundamental mixing effect as it was in 2D. In this chapter, the evolution of chroma keying and the development of alpha matting will be reviewed, and their advantages and drawbacks will also be analyzed during the reviewing. To reveal the underlying causes of imperfect chroma keying results, it is necessary to first look back into the history.

2.1 The History of Matting

In the early days of black and white filmmaking, manual matte was first introduced for compositing different shots in Georges Méliès' 1898 film "Un homme de têtes". It presented the magic of film matte and became one of the greatest black and white films. Then the Williams' Process [17] as illustrated in Figure 2.1 was patented by Williams in 1918, which shoots one foreground against black background and uses double exposure to composite. It was first successfully utilized in the 1927 film

"Sunrise" and got quite famously used in the 1933 film "The Invisible Man". This process is very similar to the luma key we use today. Because of the black screen nature, neither shadow nor dark foreground would survive in the matte.

Figure 2.1: Illustration of the Williams' Process. The foreground is shot against a black background (left) using negative film (middle), and the exposed negative film is intensified during developing to generate the mask (right). For negative films, the latent images were exposed darkens and forms a visible image during developing, the darker the less transparent. The smiling senior woman photo courtesy of Lammeyer [18].

In 1928, the first blue screen matting process had been invented by Dunning which was called the Dunning Process [19] and was patented in 1932 [20]. Its maiden shot – 1933 film "King Kong" – was groundbreaking and also one of the most critically acclaimed movies of its time. The Dunning Process is illustrated in Figure 2.2. First, the substituting background is shot on a double-coated film to generate both positive and negative exposures. Then the film is dyed with orange on the positive side and with blue on the negative side. Next, the foreground is shot in white lighting against blue coloured background with the substituting background film plate imposed. The blue background is filtered out by the orange dye and the global yellowish tone is adjusted by the blue dye, and thus a approximate white light level can be approximately restored. The foreground colour is filtered but does not affect view experience much since it is shot on black and white film, and this is also the reason why Dunning Process only worked with black and white film.

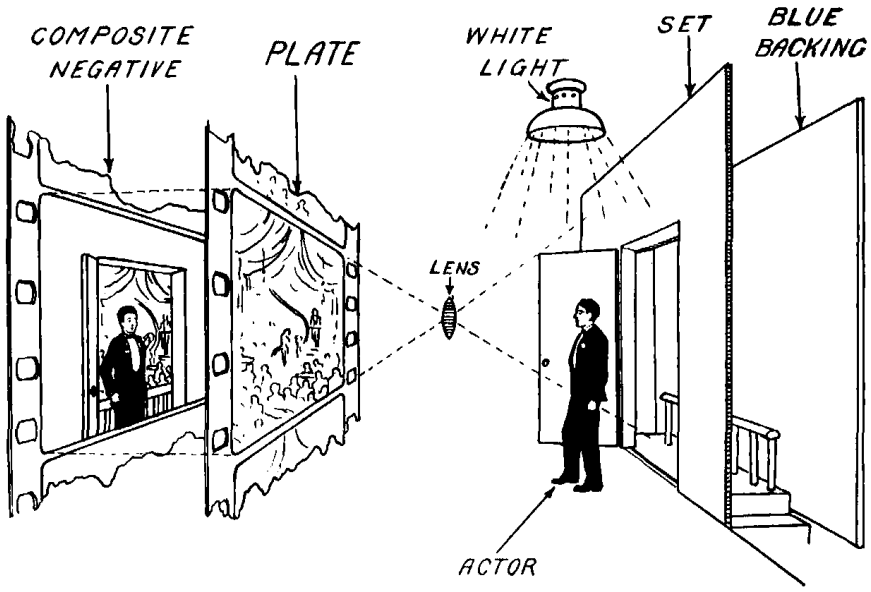


Figure 2.2: Diagram illustrating the Dunning process from [19]

When it came to the colour film era, new blue-screen travelling matte process was invented in 1940 by Butler and first used in the 1940 film "Thief of Bagdad". Butler also received the 13th Oscars Academy Award for Special Effects for the colour film matting process [21]. It was the unprecedented chroma keying process which properly deals with colours and has since become the standard process for chroma keying in colour filmmaking. It utilizes the three-colour cinematography Technicolor process [22] which splits three additive primaries – red, green, and blue – onto three separate films, and blue is chosen as the key since it is far from skin tones and has the finest film grain. The foreground is shot against blue background, then the blue channel film is intensified to generate the mask, as explained in Figure 2.3. Similar technique has also been applied on green screen, which was patented by Pohl in 1954 [23]. However, these approaches only provide the silhouette level separation of designated foreground and background, which is known as segmentation in today's

image processing field. Besides, this process is time consuming with optical printer at that time.

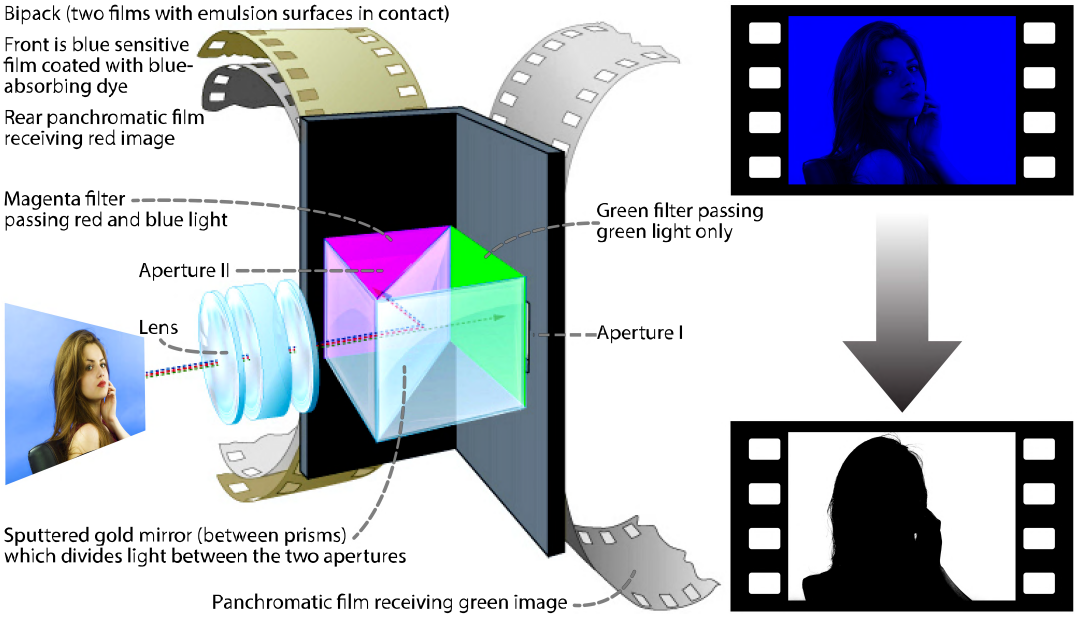


Figure 2.3: Direct blue-screen travelling matte process on Technicolor's three-colour cinematography (left). The blue channel film (upper right, illustrated in blue though it is monochromatic on film) is intensified to generate the mask (bottom right). The original illustration for three-colour cinematography Technicolor process was from [22] and the coloured remake courtesy of Sarah Steinbacher, Multimedia & E-Learning-Services, University of Zurich. The actor sitting in front of blue screen photo courtesy of Czamfir [24].

The limitation of the first blue-screen for colour film is due to the colour composition of foreground overlaps the blue range in visible light spectrum. Therefore, other background screens have been utilized, including invisible lighting screens. Two well-known solutions are the ultraviolet travelling matte process by Fred and Tondreau [25] and the sodium vapour process by Vlahos [26], though, meanwhile, similar approaches were used in Europe. These two processes are based on the observation that ultraviolet light and sodium vapour light have particular wavelength ranges and do not overlap the wavelength of common colours, thus they can be easily separated

from foreground colours by using filters. As illustrated in Figure 2.4, the background is an ultraviolet or sodium vapour illuminants lit diffusion screen. Both of them utilized the light splitting unit to duplicate the source light from the scene. The foreground light and background light captured simultaneously by two filmstrips, and the films contain background light are used to generate the matte. While they both have higher quality than previous techniques, the sodium vapour process show the advantage over the ultraviolet travelling matte process because the visible sodium vapour light generates similar light effects such as refraction and diffusion to other visible colours and thus the generated matte is more natural than the ones generated with ultraviolet [27]. The sodium vapour process produced some of the best films of the time such as the 1961 film "The Parent Trap" and the 1964 film "Mary Poppins".

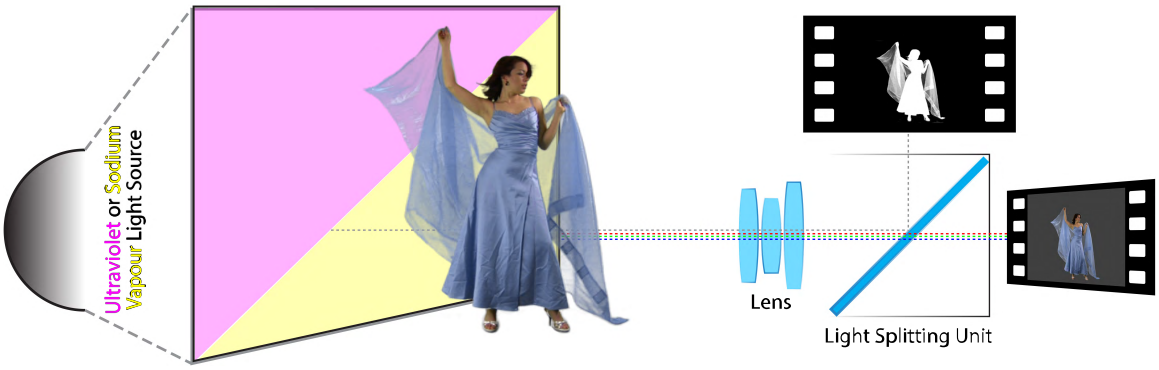


Figure 2.4: Unified illustration of ultraviolet travelling matte process (ultraviolet light depicted in magenta, though invisible to human eyes) and sodium vapour process (sodium vapour depicted in yellow) according to [25] and [26]. Actor photo courtesy of Hollywood Camera Work [3].

Despite that the ultraviolet travelling matte process and the sodium vapour process solved the blur and translucency problems, studios utilizing either process can be costly and both of them have the drawback that the actor can not be too close to

the background screen due to health hazard and light spill from high intensity backlight. Therefore, the industry started considering improvements on existing blue-screen process. In 1959, Vlahos first proposed the colour-difference approach based on the observation that most foreground colours have the same amount of green and blue, and thus earned the 37th Oscars Academy Scientific or Technical Award [21]. A patent was also issued in 1964 [28]. This process also utilizes both positive and negative films, first the scene is shot by a three-colour camera on positive films. Next, the green channel film is developed into negative and then overlapped the blue channel film. This double film pack is exposed under controlled lighting condition to generate the matte film, while the light transporting through both films acts as the channel difference calculation in mathematics and thus named colour-difference matte process, as sketched in Figure 2.5.

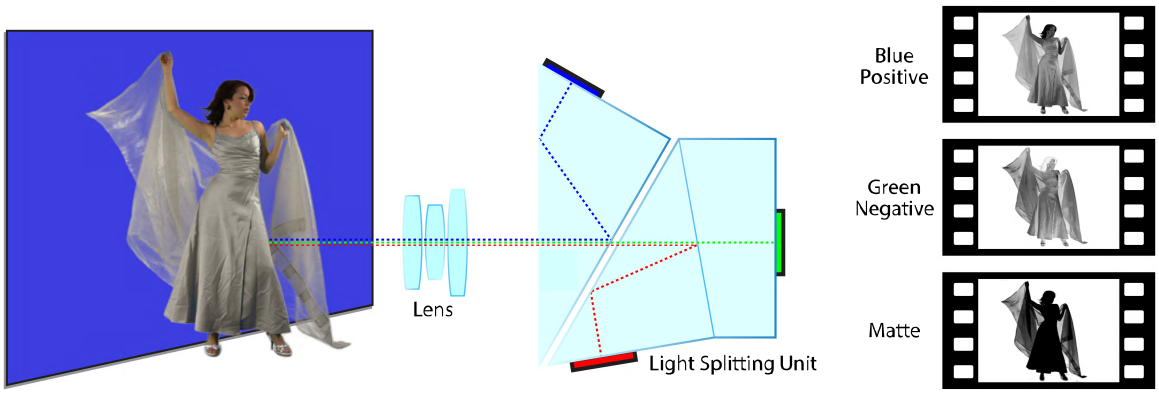


Figure 2.5: Illustration of colour-difference matte process with blue background screen. For simplicity, conversion from positive film to negative film is not drawn. Actor photo courtesy of Hollywood Camera Work [3].

The colour-difference matte process also solves the blur and achromatic translucency problems. The blue channel of foreground colours is corrected with the green channel. It dominated the filmmaking industry for almost 40 years, and was adapted

to electrical signal processors nourished the special effects development in television industry. Even today, many chroma-keys still based on its essential idea.

2.2 Chroma Keying in Industry

When filmmaking and television industries entered the digital era, different approaches have been deployed. Their algorithms are more specific to perform alpha matting against monochromatic background in green or blue, thus chroma keying is usually called the green-screen or blue-screen effect. The basic algorithms used in modern commercial chroma-keyers generally fall within the categories of colour difference key, colour vector key, HSL key (Hue-Saturation-Lightness, usually referred to as chroma key), and Multi-Space key (usually uses a combination of multiple colour space).

2.2.1 Colour-Difference Approach

Generating a matte using colour-difference approach has a long history dating back to the film epoch as briefly introduced in the previous section. The forerunner of studio level digital colour-difference keyer is the digital Ultimatte (including software and hardware implementations, referred to as Ultimatte in the remainder of this thesis) developed by Ultimatte Corporation. It adapts the colour-difference matte process originally developed by their founder Petro Vlahos for filmmaking since the basic idea is also applicable on digital colour representation, and it also takes red channel into consideration. The basic idea is very simple – the initial matte is determined by the difference of the key colour component and the remaining maximum colour component, pixel-by-pixel, and then the initial matte is scaled in range. Its sophisti-

cated versions utilize multiple adjusting parameters. According to the latest patent regarding colour-difference approach filed by Ultimatte Corporation [29], the matte generation equation is further developed into:

$$\mathbf{E}_c = K(\mathbf{B} - K_1) - K_2 \max(K_3\mathbf{G}, K_4\mathbf{R}) - \max(K_5(\mathbf{G} - \mathbf{R}), K_6(\mathbf{R} - \mathbf{G})) \quad (2.1)$$

where K and $K_i (i \in 1, 2, ..., 6)$ are adjusting parameters.

Among the contemporaries of the first Ultimatte unit, Devereux uses a similar approach by performing the colour-difference algorithm on the U-V chrominance plane of a planar YUV colour space (Y, U and V represent the luminance and two chrominance-luminance difference components) [30]. An advantage of this implementation is that the YUV colour space isolates the luminance and chrominance information into separate channels, and the polar coordinate conversion of the U-V chrominance plane turns the chrominance into a Hue-Saturation representation. Therefore, the hue angle of background is adjustable with flexibility. The industry-loved Keylight chroma-keyer by The Foundry Visionmongers [31] also applied the similar technique.

2.2.2 Colour Vector Approach

Chroma keying using colour vector is usually referred to as 3D colour-space keying – a vector from background colour to foreground colour is used to determine the alpha value of the background-foreground pair based on the position information of blended colour on it. The Primatte [32] by Photron uses its proprietary Primatte Polyhedral

Slicing Algorithm, which utilizes concentric, multi-faced polyhedrons in RGB colour space for colour clustering. It was originally presented at the 8th NICOGRAPH Conference [33] and patented in 1994 [34]. These multi-faced polyhedrons provide better fitting and isolation of different colour vectors than simple planes or spheres, as illustrated in Figure 2.6. The three polyhedrons separate the colours into four regions: absolute background colours, blended foreground and background colours around the boundaries, blended foreground colours with spilled background colours, and absolute foreground colours. Its definition on spilled regions further helped the colour restoration of foreground.

The commercial version of Primatte uses 128-faced polyhedrons to fit the varied colour dispersion, and achieves excellent results. The initial implementations on CPUs are reported to be slow and hard to handle. Later it came a GPU version of Primatte, which the performance manipulating polyhedrons are optimized through the 3D pipeline on GPU. There are also two simplified versions – one released as Primatte RT which only use one 1-faced separating surface, and another is Primatte RT Plus with two 6-faced polygonal surfaces and Primatte RT Plus – according to the Primatte plug-in reference in Fusion by eyeon Softerware [35]. Similar approach is used by Discreet Logic Incorporated (acquired by Autodesk Incorporated in 1998) in their Modular Keyer. One characteristic of vector approach is that user must manually select background regions or even foreground regions to perfect the result. This made it very similar to the pixel-sampling based approach in alpha matting users select background and foreground regions to generate rough segmentation (trimap), and the selected pixel samples are used to shape the separating surfaces, while the separating surfaces constrains the colour distribution approximation.

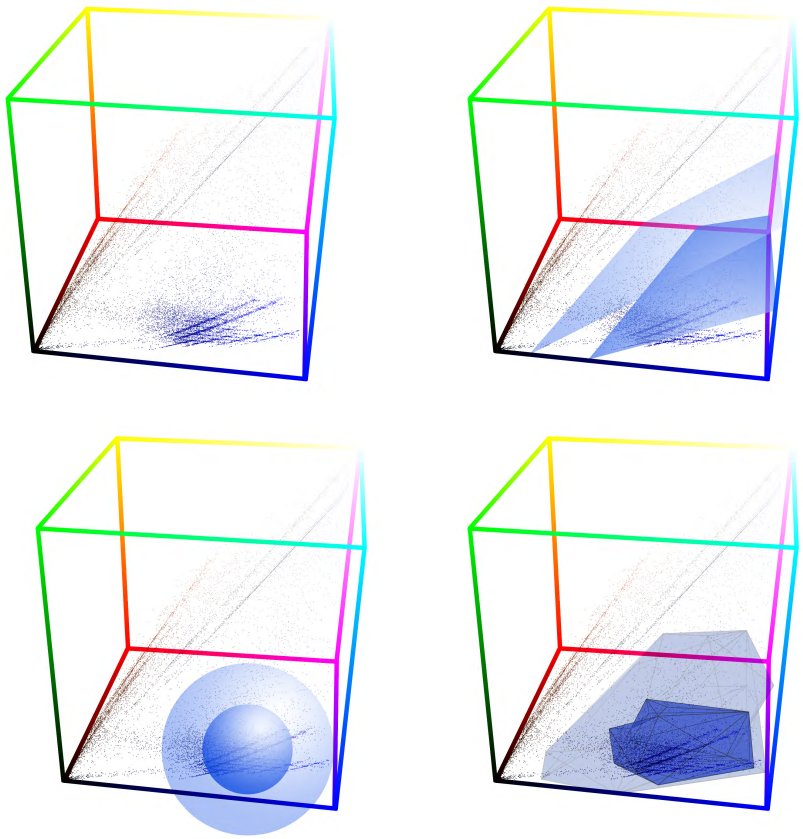


Figure 2.6: Comparison between different separation surfaces: the colour distribution inside an RGB cube (top left), separating with double planes (top right), separating with double spheres (bottom left), and separating with double polyhedrons (bottom right).

2.2.3 HSL Approach

The Hue-Saturation-Lightness approach for chroma keying is relatively straightforward to understand in comparison to the previous two approaches. In an ideal scenario, the background should be hue and saturation consistent because the hue and saturation are two lightness-independent properties that only related to the dye used in the background drop or paint. Therefore, the background colour can be clustered into a certain region and thus can be removed even under non-uniform achromatic

background lighting conditions. One well-known HSL chroma-keyer used in industrial application is the Discreet Keyer by Discreet Logic Incorporated, which is further evolved into the Diamond Keyer. The background colour range and blending tolerances are constrained by two Hue-Saturation-Lightness triplets, as depicted in Figure 2.7.

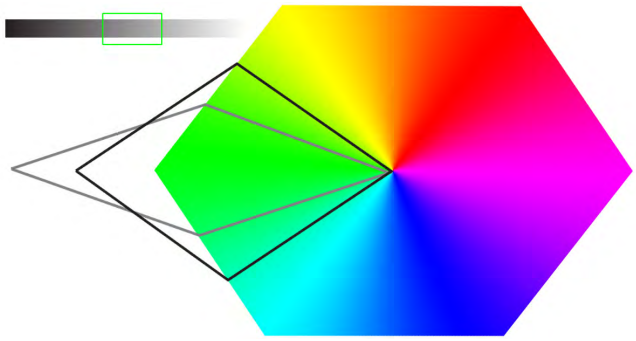


Figure 2.7: Illustration of soft segmentation on an HSL colour space: the grey polygon constrains absolute background chrominance, the black polygon constrains transition ranges, and the green rectangle on the greyscale gradient bar constrains background lightness.

However, the real world lighting is not pure white and thus may contaminate the hue-saturation consistency of the background. Besides, traditional Hue-Saturation-Lightness model such as RGB-HSL failed to model such perception related properties well [36].

2.2.4 Other Approaches

Among the commercial chroma-keyers, many of them do not expose their underlying principle through neither the patents nor the user interface. One of them with good quality is the Ultra Key by Serious Magic Incorporated, which claims 4-Space Vector Keying. Through the investigation, a Hue-Saturation-Lightness colour space based

spill suppression algorithm is used for foreground colour recovery, and sample based background fill in RGB colour space is used for background colour recovery. Considering its real-time performance on an average personal computer, high dimension future space is unlikely utilized in its algorithm. Thus, the other two spaces in its claim are still not clear.

2.3 Alpha Matting in Academia

In computer vision field, the classical algorithms for alpha matting are mainly based on pixel sampling, pixel affinity, or a combination of both [37]. Pixel sampling based approaches explicitly estimate the alpha values by sampling within the sets of certain foreground and background colours, yet insufficient pixel samples may cause matte discontinuities. Pixel affinity based approaches calculate the alpha value of a given pixel by analyzing its spatial neighbourhood, which usually result in local smoothness due to strong pixel correlations. In this section, we will briefly introduce some classical and extended approaches for alpha matting problem in academia. One noticeable difference between alpha matting approaches in academia and chroma-keyers in industry is that most of the commercial chroma-keyers use additional colour spill suppression algorithm to further correct the foreground colour, while the academic alpha matting approaches sample colours from existing known foreground.

2.3.1 Pixel Sampling Based Approaches

Soft matte approaches involve pixel sampling can date back to 1995 when Mitsunaga *et al.* constructed alpha values along object boundaries based on image gradient field

and linear approximation [38]. In 2000, Ruzon and Tomasi introduced a probabilistic view and an unmixed colour estimation to produce better matte [39]. Later, the Bayesian Matting [40] proposed by Chuang *et al.* in 2001 refined the solution with spatially varying sets of Gaussians and solves it using the Maximum A Posteriori (MAP) technique.

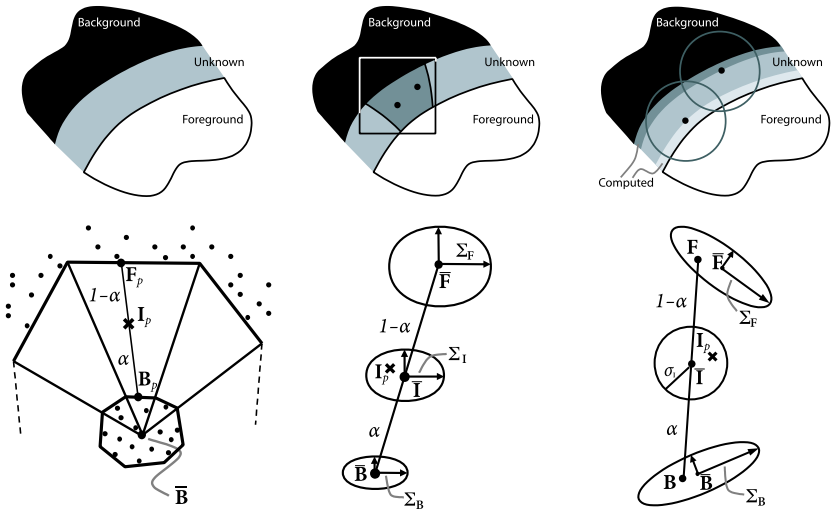


Figure 2.8: Sampling strategies (top) and corresponding colour models (bottom): Mitsunaga *et al.* (left), Ruzon and Tomasi (middle), and Chuang *et al.* (right). Diagrams courtesy of Chuang *et al.* [40].

Since then, most of the alpha matting approaches (according to the Online Benchmark for Image Matting [10]) do not solely use pixel sampling in matte generation except for the real-time version of Shared Sampling for Real-Time Alpha Matting [41] proposed by Gastal and Oliveira in 2010. This approach distributes the computations among neighbouring pixels and then shares their results within an affinity. Each pixel in unknown region searches along four different directions than its spatial neighbours, and only samples at most two foreground samples and two background samples, as shown in Figure 2.9

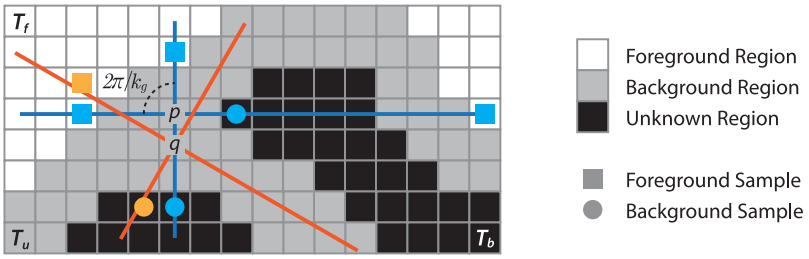


Figure 2.9: Illustration of shared sampling for real-time alpha matting.

During the process, it also measures image space statistic by integrating the image gradient from target pixel to sample pixel, and calculates the probability of a target pixel belongs to foreground. Such texture analysis ensures the geometric continuity from the determined foreground sample to the target pixel in unknown region, which is analogous to the sampling strategy proposed by Rhemann, Rother, and Gelautz in [42], as illustrated in Figure 2.10.

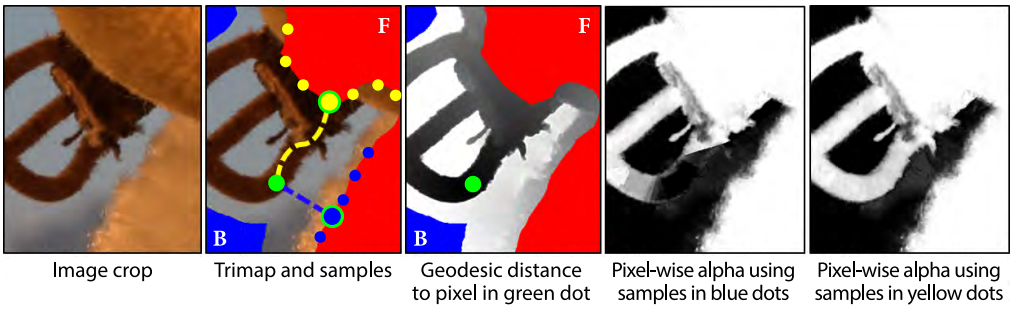


Figure 2.10: Sampling strategies and results using different geodesic metrics. In the third image, the geodesic distance from the pixel marked in green to its affinity is shown in greyscale, where lower value indicates shorter distance. Illustrations courtesy of Rother, and Gelautz [42].

Because the computations performed on the pixels are independent with each other, the overall processes can be executed in parallel and achieve real-time performance. However, large unknown regions that require big sampling radius may also

impact its performance.

2.3.2 Pixel Affinity Based Approaches

Sun *et al.* introduced the Poisson Matting [43] which estimates the alpha matte by solving a Poisson equation with the matte gradient field and Dirichlet boundary conditions (0 for background and 1 for foreground). Usually, the Poisson Matting is referred to its global version, which approximates the matte gradient by taking partial derivatives on Equation 2.2:

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

where $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ is the gradient operator. Since it is assumed that the transition between predefined $\mathbf{F}$ and $\mathbf{B}$ in a local neighbour is smooth, the $\alpha\nabla\mathbf{F} + (1 - \alpha)\nabla\mathbf{B} \ll (\mathbf{F} - \mathbf{B})\nabla\alpha$ and thus $\nabla\alpha$ can be approximated as $\nabla\mathbf{I}/(\mathbf{F} - \mathbf{B})$. Therefore, insufficient transition smoothness between foreground and background colours inside the unknown region may cause erroneous matte.

The Random Walk Matting [44] proposed by Grady *et al.* and the Closed-Form Solution to Matting Problem [45] proposed by Levin *et al.* have similar affinity function but different colour metrics. Both of them propagate the boundary constrains to the entire image by using symmetric Laplacian matrix and minimizing a quadratic cost function. Because of the careful analysis of the matting problem by Levin et al, the colour model in the Closed-Form Solution to Matting Problem is more reasonable and hence can generate relatively better matte. But due to the limitation in propagation, isolated unknown regions that only surrounded by foreground region or

background region may not have right alpha value, and large unknown region may have transitive propagation errors. Even they can be partially solved by indicating wider neighbourhoods, the memory and computational cost is also increasing.

Another example is the Iterative Transductive Learning for Alpha Matting proposed by He et al. [46] It uses asymmetric Laplacian matrix in comparison with the Random Walk Matting and the Closed-Form Solution to Matting Problem. It also utilizes an iterative process of transductive learning to optimize the alpha matte. Though their results are in high quality, the memory and computational cost is apparently higher because of the asymmetry of its Laplacian matrix and the iterative nature.

2.3.3 Joint Approaches

The combination of both pixel sampling and affinity may achieve better results, but it also can be time consuming.

The Optimized Color Sampling for Robust Matting by Wang and Cohen [47] first samples foreground and background colours along the corresponding boundaries of the unknown regions, and then refines the alpha matte using the modified random walk algorithm from Random Walk Matting with additional colour constraints applied. Nevertheless, the Shared Sampling for Real-Time Alpha Matting has a non real-time version that utilizes the matte Laplacian from the Closed-Form Solution to Matting Problem to smooth out the matte. With enough samples gathered and the addition smoothing, it can generate matte and estimate foreground colours with high quality, but also very unlikely to be in real-time on cost-efficient platforms in the near future.

As stated in the previous sections, other recent high ranking alpha matting algorithms on the Online Benchmark for Image Matting website [10][48] utilize both pixel sampling and affinity. The Learning Based Alpha Matting using Support Vector Regression [49] by Zhang, Zhu, and Xie determines alpha value from the labeled samples in the known regions. And the initial alpha values are further refined and smoothed using the Laplacian matrix from [45]. The Improving Image Matting using Comprehensive Sampling Sets [50] by Shahrian *et al.* introduced an optimized sampling method for wider colour distribution and utilizes the Laplacian matrix from [45] for refining alpha matte in its post-processing as well. The Image Matting with Local and Nonlocal Smooth Priors [51] by Chen *et al.*, which is also the website's highest ranked alpha matting approach at the time of compiling this thesis, uses K-Nearest Neighbours (KNN) sampling technique and also adapts the Laplacian matrix from [45] into RGBXY multi-dimensional feature space to enforce the local and non-local smoothness.

2.3.4 Other Approaches

Of course, there are some non-general solutions to the matting problem, which depend on specified capture devices to gather additional object information.

The Z-Key system [1] proposed by Kanade, Narayanan, and Rander in 1995 captures the pixel-wise depth information in the form of a depth map to determine the boundaries and position of foreground objects, which is actually hard segmentation based on depth-key. Similar work includes the Automatic Nature Video Matting with Depth [52] by Wang *et al.* in 2007, and the Image Matting with Color and Depth Information [53] by Lu and Li in 2012. Both of them have further optimized

the boundary transition by applying general alpha matting strategies. Wang *et al.* adapted the Poisson Matting, while Lu and Li applied the Bayesian Matting.

The triangulation matting algorithm proposed by Smith and Blinn in 1996 shoots the same foreground against two different background colours green and blue. Their solution mathematically solved the inverse alpha blending problem, but it is not practicable in filmmaking and television productions applications. Similar technique is also used in the ground truth extraction in the Online Benchmark for Image Matting [10]. As stated in the introduction chapter, the alpha blending does not account for physical translucency. As shown in Figure 2.11, the extracted foreground of translucent blue glass turns into emerald.

Figure 2.11: Triangulation against two different backgrounds, the blue glass acts as a colour filter and thus the original can not be recovered through triangulation. Images courtesy of Smith and Blinn [54].

The Flash Matting [55] proposed by Sun *et al.* in 2006 extracts the alpha matting from a flash photo and a non-flash photo. It based on the assumption that the appearance of foreground may change drastically under camera flash, and applied the Bayesian framework from Bayesian Matting jointly on the non-flash image and the flash-only image as depicted in Figure 2.12, where the term 'flash-only' referred to as the difference image between the flash image and non-flash image.

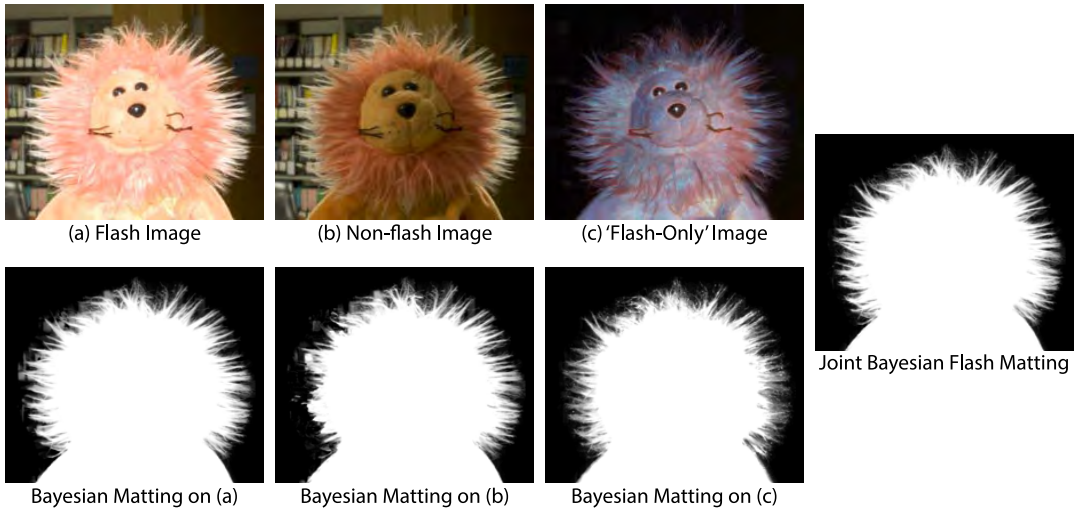


Figure 2.12: Flash matting images and corresponding alpha map generated with Bayesian framework. Images courtesy of Sun *et al.* [55].

Extracting foreground by utilizing light signals that do not affect colour appearance is also a good way for solving alpha matting problem. In 2000, Ben-Ezra proposed a polarization matting system [56], which utilizes polarized background and polarization filter to extract foreground. It is based on the assumption that the magnitude of light reflected by the foreground is omnidirectional. Since the background is lit by light within visible spectrum, the light transportation affected by the foreground is natural. This is big advantage than infrared or ultra-violet lit background

that may result in halos and other unnatural appearance because of the wavelength related physical properties such as fraction factor. Ben-Ezra also proposed the implementation of catadioptric camera design with solo and dual camera setting, which made the determination of physical translucency possible because the polarization filter is effective for all colour channels. He also proposed a sub-pixel polarization based on Bayer Pattern CCD filter, as depicted in Figure 2.13. In this case, the exact foreground colour can be determined precisely in real-time.

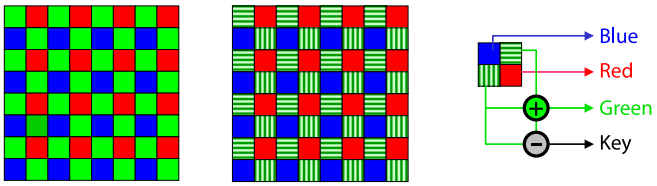


Figure 2.13: Polarized Bayer Pattern CCD filter. Illustration courtesy of Ben-Ezra [56].

Similar approach has been used in the Practical, Real-Time Studio Matting using Dual Imagers [57] by McGuire, Matusik, and Yerazunis in 2006. But the relatively high cost of polarized background, special light settings, and special optical components in comparison to traditional blue or green screen matting still make it unacceptable in daily studio use.

In 2012, Bentabet and Zhang adapts the Bayesian framework from Bayesian Matting jointly on the color image and the infrared thermal map [58]. However, the drawback is also apparent. Such system is only capable of extract foreground objects that have a different temperature range than the background's, such as human body.

Some other matting methods extend the scope into optical properties of foreground. A very interesting research is the Environment Matting and Compositing [59] by Zongker *et al.* in 1999, which performs ambient ray tracing along with alpha mat-

ting and can get very good result on translucency, refraction of the foreground objects, as illustrated in Figure 2.14.

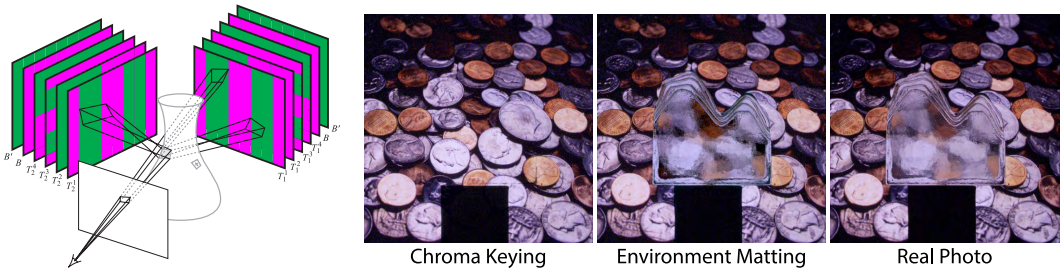


Figure 2.14: The illustration of Environment Matting process (left), and the comparisons between compositing results and the real photo (right). Illustration and images courtesy of Zongker *et al.* [59].

The drawback of such matting process is apparent. Despite the complex cave setting to wrap the foreground objects, the arbitrary light transporting paths such as diffuse reflection, which cause serious distortion of the modulation pattern. This problem is solved later by Chuang *et al.* through scanning [60], which is not a real-time approach in processing either because the ray tracing process is also time-consuming.

2.4 Discussion

In 2011, Luo adapted the Optimized Colour Sampling for Robust Matting to chroma keying problem [61] by using chrominance clustering on a rotated planar YUV colour space for automatic trimap generation. Despite the colour model is not good for image segmentation, the Robust Matting itself may not suitable for real-time application because of its large candidate sets. In 2012, Hao further improved it with adaptive trimap in temporal dimension based on Thin-Plate Spline [62]. However, the adap-

tive process is computationally intensive and is not suitable for dynamic foreground objects such as smoke clouds.

For commercial chroma keyers, Smith and Blinn [54] believed that the evolution of colour-difference matte is based rather on years of practical experiences and experiments than mathematical derivations. According to them, the colour-difference model can be cast into a colour vector approach that was defined and introduced before. In my opinion, the Hue-Saturation-Lightness based approaches can also be cast into colour vector based approaches. And because of the non-linear colour adaption of Hue-Saturation-Lightness colour space, the isolation surfaces can be more complex than the ones used in Primatte. In other words, most of the commercial chroma-keyers used in industry are implemented based on soft-segmentation with isolation surfaces.

Chapter 3

The Perceptual 3D Chroma Keying Framework

The process of human perception is a stimulus driven, experience guided closed-loop process [63]. In a chroma-key setting, the perceptual monochromatic background usually contains the greatest stimulus salience. With our general understanding to the purpose of chroma keying, our attention falls on the foreground elements through the cognition of colour differences. In addition, we may also focus on the regions that are likely to contain fine details such as fuzzy edges and trivial lines such as hairs because of the progressive nature of human recognition. Then, we start to analyze the original foreground colours that are contaminated with background colours using our experiential knowledge by assuming colours of occluded background regions have similar colours with other visible regions. Though different people may have different cognition of colour mixing, a reasonable prediction on foreground colours commonly acceptable is usually preferred. In the meantime, we may try to determine the distance

to the foreground objects. The dimensions of the objects and the binocular depth effects provide us the reference information for analysis, and thus we can get an approximate distance to the objects. Through these mental activities, we finally complete the perception of the scene.

The proposed framework consists of a set of carefully designed methods and algorithms that simulate the above human perception process in means of perceptual analysis, background restoration, foreground colour and alpha prediction, depth perceiving, and information fusion, as described in Figure 3.1. The information fusion will not be discussed here since in real-world applications it focuses on the generation of visual composites from colour, alpha, and depth, which varies case to case according to particular demands.

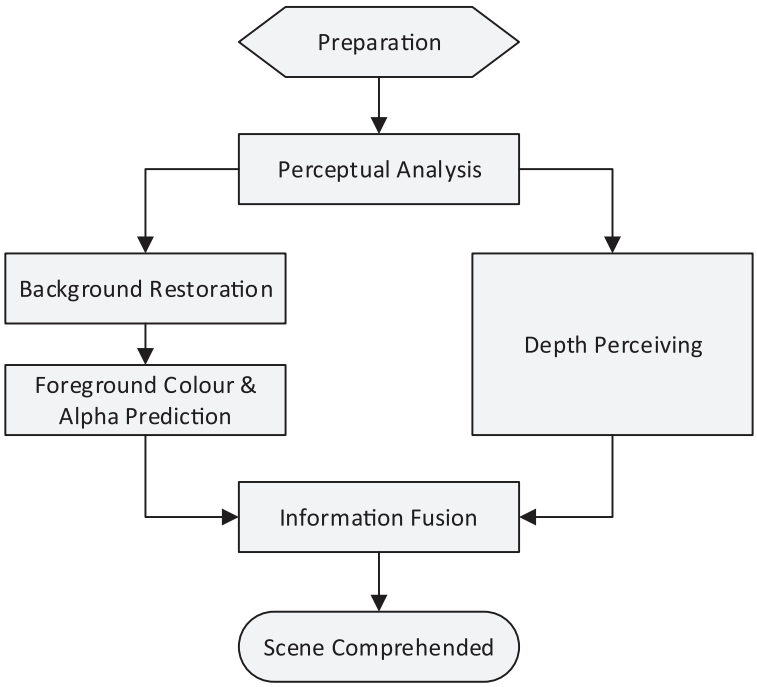


Figure 3.1: Workflow of proposed framework.

3.1 Perceptual Uniform Colour Space

To simulate this human perceptual process, it is necessary to find a perceptually modelled colour space that is similar to humans' colour sensation at first. To most humans, a colour can be decomposed into three main characters hue, saturation, and lightness. Natural lights are not homogeneous, which means the stimulated colour sensation is a result of lights with variety of different wavelengths. In fact heterogeneous lights with different spectral power distributions may stimulate the same colour sensation because they have same dominant wavelength (or equivalent wavelength) and sensible power density. The stimulated colour sensations by lights have such property are called metamers. Hue is determined by the dominant wavelength and is also closely related to the naming of colours. Though there is variability in colour naming due to environmental factors, often four colours that observers incline to perceive as pure colours – red, green, blue, and yellow, which are defined as the unique hues. Other composite hues such as orange, cyan, and purple (violet) are usually considered as a mix of unique hues to human perception even they do naturally exist as homogeneous lights [64][65]. The saturation is the perceived vividness, or the purity which of a given hue. From a physical view, it represents how much the intensity of dominant wavelength is distributed across the spectrum of different wavelengths. Lightness is more understandable to human because it describes the achromatic brightness - a property independent to colour naming and vividness. Thus nearly homogeneous light such as laser is often considered monochromatic and fully saturated though it may be dark or bright.

Several perceptually modelled colour spaces models have been proposed since the beginning of the 20^{th} century. The most notable and also the first colour system that

scientifically decomposed into a perceptual three-dimensional space based on hue, saturation, and lightness is the Munsell colour system [66]. It is a notation based colour system built up on strictly measured human visual responses to various colours, and was refined as time went by. This system has outlasted its contemporaries and is still widely used today [67]. In fact, a property named 'chroma' is used in the Munsell colour system as well as many of today's perceptual uniform colour spaces because it is related to lightness in those spaces. Modern well-known perceptual uniform colour spaces or appearance models include CIE 1976 (L^*, u^*, v^*), CIE 1976 (L^*, a^*, b^*) [68], IPT [69], CIECAM97s [70], and CIECAM02 [71]. However, the subtractive transform of CIE 1976 (L^*, u^*, v^*) is problematic, and the sub-optimal chromatic adaptation of CIE 1976 (L^*, a^*, b^*) weakens the hue constancy especially in the blue region [36]. The IPT colour space and two CIECAMs utilize LMS cone signals transform with modified exponent adaption function with different compress factors. The CIECAM02 performs as well as, or better than CIECAM97s in almost all cases and is considered a replacement of the CIECAM97 family [72].

Four of the colour models which are commonly used or have proven advances [72][73] are chosen as candidates for our perceptual process – CIE 1976 (L^*, a^*, b^*), IPT, CIECAM02, and CIECAM02-UCS. The CIE 1976 (L^*, a^*, b^*) are chosen due to its simplicity, though its poor hue constancy in the blue region will reflect badly on the blue screen chroma keying. The IPT colour space shows a slightly higher hue constancy in hue against different data sets [73], but its overall performance as a colour space is not as good as the derived uniform colour space (CIECAM02-UCS) from CIECAM02 [74]. However, the data sets used in their evaluations are standardized colour data sets. Even studio level lighting condition and cameras may

not acquire actual colours in practice. Therefore, the four candidate colour spaces are evaluated with profiled green screen backgrounds and blue screen backgrounds take by our self and collected from the Internet.

The colour space evaluation is focused on two aspect – hue constancy and saturation constancy. For CIE 1976 (L^*, a^*, b^*) colour space, the hue is taken from its cylindrical representation – CIE 1976 (L^*, C^*_{ab}, h_{ab}). Although the saturation property is not officially defined in CIE 1976 (L^*, a^*, b^*), we take the suggestion from Fairchild in [36] to approximate the perceived saturation by dividing its chroma (C^*_{ab}) by its lightness (L^*). Likewise for IPT colour space, the hue is also taken from its cylindrical representation, and the saturation is approximated by dividing its polar radial distance on the chrominance plane $(\sqrt{P^2 + T^2})$ by its luminance (I). Because the CIECAM02 colour appearance model and its derived CIECAM02-UCS colour space share the same hue angle and hue composition, the two related properties are evaluated once while their saturations are evaluated independently. In total, 1302 green screen images and 617 blue screen images (screen part only) from photos and video frames were used for evaluation. The saturation from CIECAM02-UCS is also evaluated. All the test images were converted to destination colour space according to their source colour space and white point described by their publisher or in their EXIF [75] tags, where the source colour spaces of the images evaluated fall in the set {sRGB, ITU-R Rec. BT.709}. The hue, hue angle, and hue composition are scale to $[0, 1]$ with a resolution of 0.0025 (equivalent to 0.9 degree angle). The saturation is also scaled to $[0, 1]$ with a resolution of 0.01 by dividing the calculated saturation by the corresponding saturation of ITU-R Rec. BT.709 [76] green primary (CIE 1931 chromaticity $x = 0.3000, y = 0.6000$) for green screen background, or ITU-

R Rec. BT.709 blue primary (CIE 1931 chromaticity $x = 0.1500, y = 0.0600$) for blue screen backgrounds, respectively.[1] Each test image was evaluated independently in hue and saturation in each colour space by measuring the proportion of background pixels that can be covered within a certain tolerance around their peak values. And the homogeneous properties are statistically analyzed among all test images. The results are intuitively shown in box whisker plots[2], shown in Figures 3.2, 3.3, 3.4, and 3.5.

The results confirmed the lack of hue constancy in blue region in CIE 1976 (L^*, a^*, b^*) colour space. As indicated in [73], the hue angle in CIECAM02 colour appearance model is not as constancy as the hue in IPT colour space. However, the hue composition in CIECAM02 colour appearance shows a better constancy in the performed evaluation. Besides, it is also obvious that the approximated saturations in CIE 1976 (L^*, a^*, b^*) colour space and IPT colour space are problematic, while the CIECAM02 saturation shows a slightly higher constancy to CIECAM02-UCS saturation. In addition, the advantage of CIECAM02 as a colour appearance model is that it provides six technically defined dimensions: brightness, lightness, colourfulness, chroma, saturation, and hue, making it easier to view colour properties of an image from different aspects. Therefore, we choose the hue composition (H), saturation (s), and lightness (J) in CIECAM02 colour appearance model as the perceptual uniform colour space, and CIE 1931 XYZ colour space as the linear blending colour space in our proposed framework[3].

[1] sRGB and ITU-R Rec. BT.709 share the same primary chromaticities.

[2] Please refer to Appendix A for a brief introduction on box whisker plot.

[3] The forward and reverse conversions between CIE 1931 XYZ colour space and CIECAM02 colour appearance model are implemented according to [71], please refer to Appendix B for details for further details.

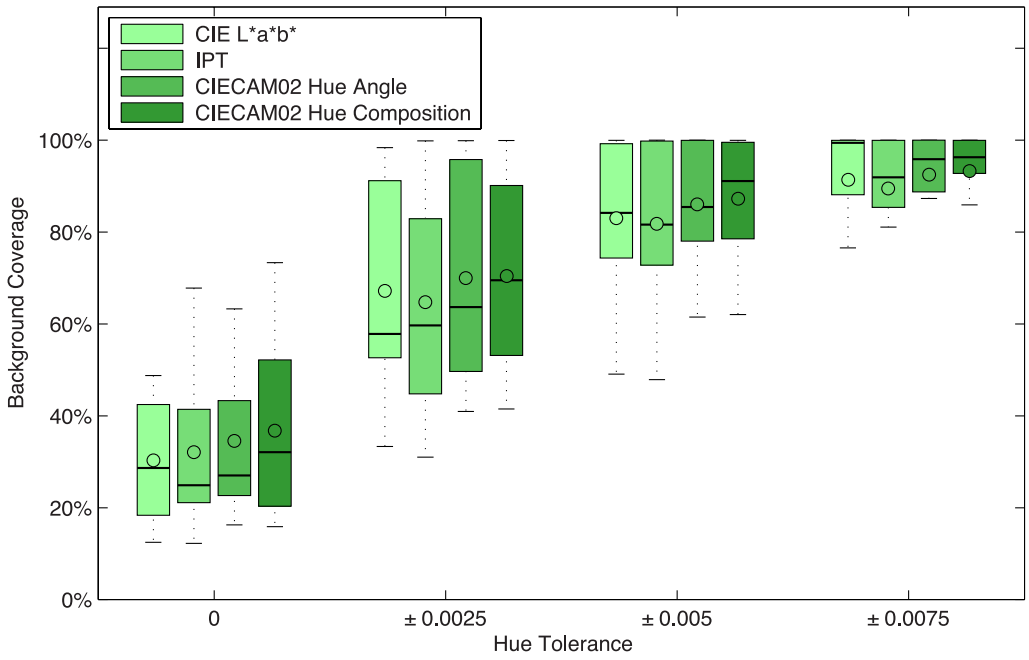


Figure 3.2: Hue uniformity evaluation on green screen backgrounds.

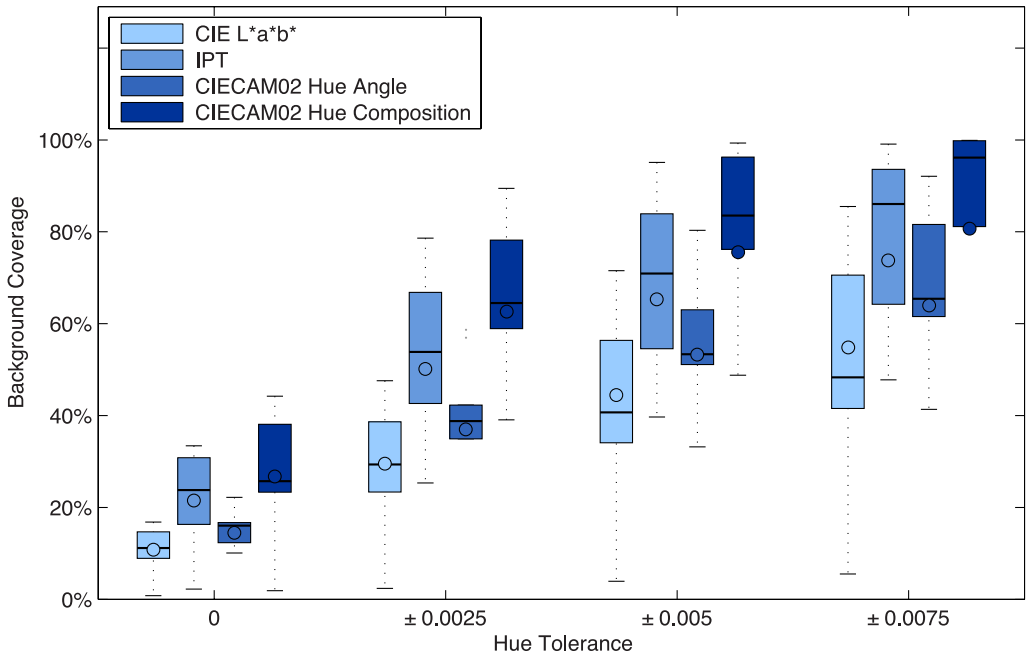


Figure 3.3: Hue uniformity evaluation on blue screen backgrounds.

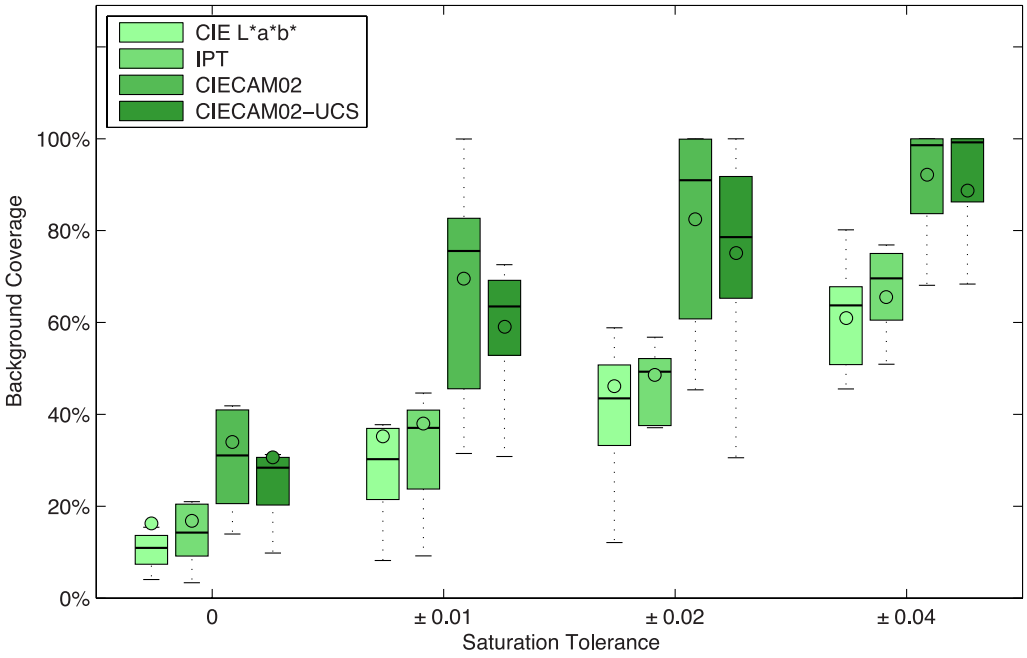


Figure 3.4: Saturation uniformity evaluation on green screen backgrounds.

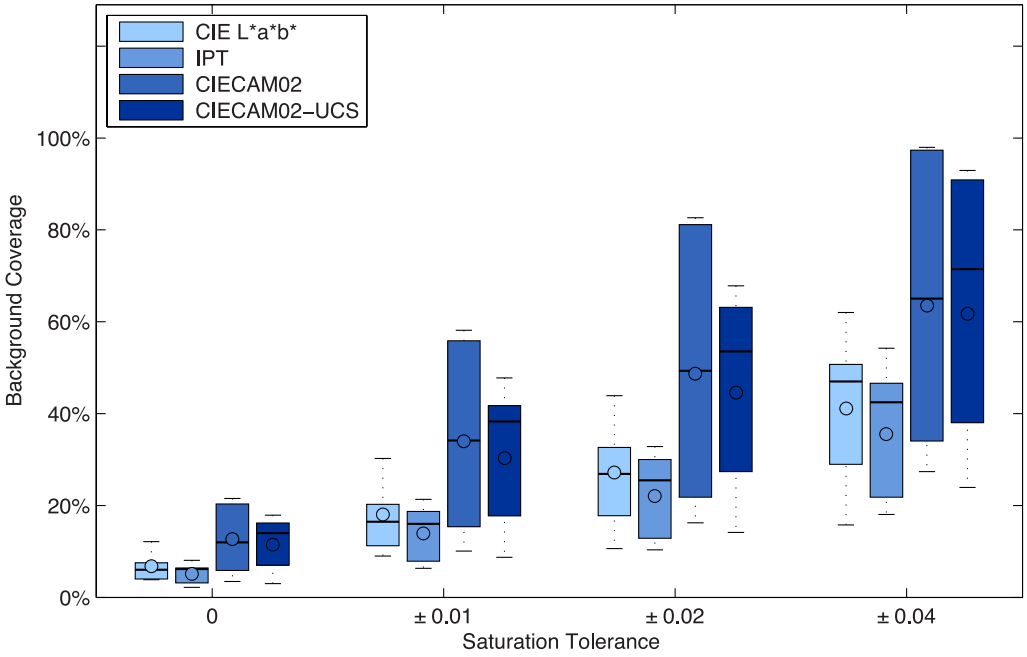


Figure 3.5: Saturation uniformity evaluation on blue screen backgrounds.

3.2 Perceptual Analysis

Three subsequent one-dimensional histogram and analysis stages on hue composition, saturation, and lightness respectively are designed for perceptual analysis. They rely on the conditional histogram as described in Table 3.1. The analysis of dominant hues is done by searching local maxima $H_{pk\mathrm{i}}$ ($\mathrm{i} \in \mathbb{N}$) in full hue composition range on the hue histogram. Then the saturation analysis determines the saturation constancies and dominant saturation ranges on each dominant hue composition. And the lightness analysis determines the lightness ranges of each dominant colour through independent lightness histograms on the previously determined chrominance ranges. The intermediate results of each stage is intuitively demonstrated in Figure 3.6, Figure 3.7, and Figure 3.8 respectively.

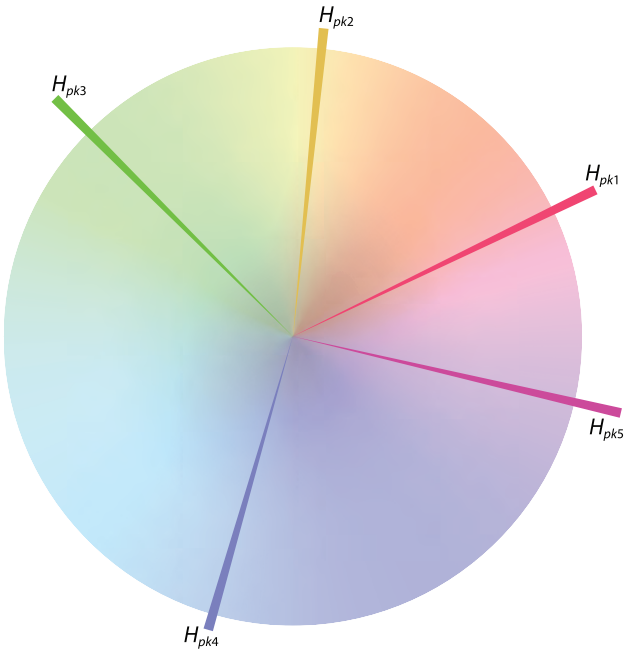


Figure 3.6: Dominant hues from hue histogram.

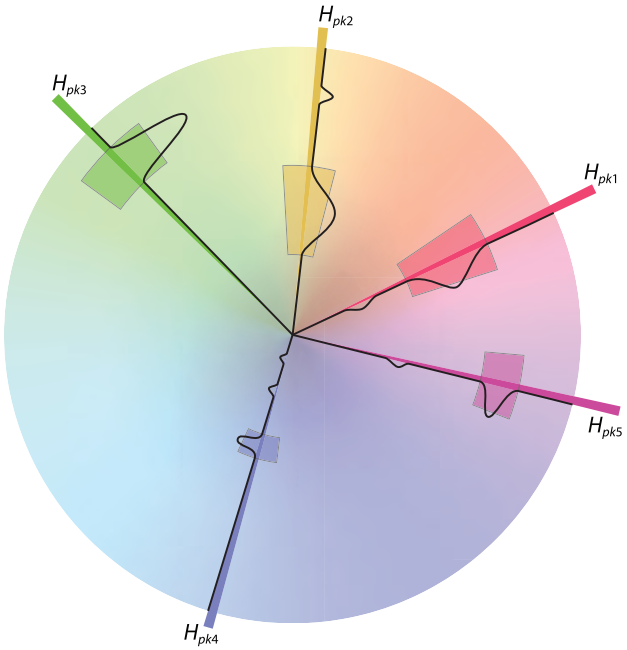


Figure 3.7: Dominant saturation ranges of dominant hues from saturation histograms. Curves indicate saturation distributions on each dominant hue. The *maxSatPeakCnt* is set to 1 for simplicity in illustration.

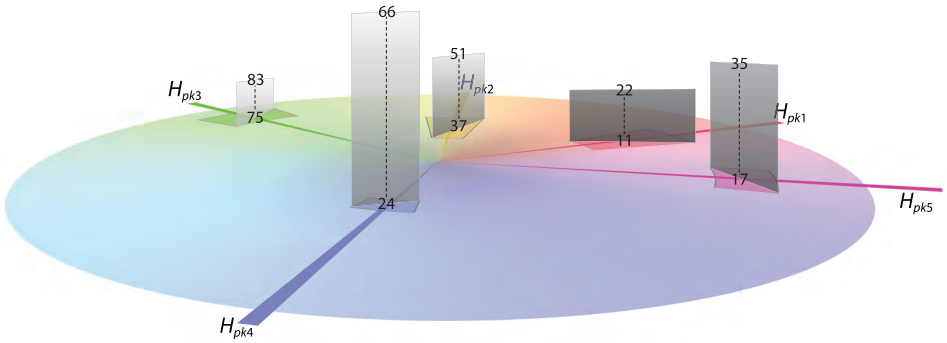


Figure 3.8: Lightness range of dominant colours from lightness histograms.

Table 3.1: Pseudo Code for Conditional Histogram

PROCEDURE Conditional_Histogram

Input(s) (

binInterval:	the bin interval for histogram
$I[\vec{\mathbf{x}}]$:	the Hue-Saturation-Lightness based image
targetComponent:	the colour component that need to be counted
conditionSet:	a set of conditions that need to be fulfilled

)

Output(s) (

histoArray:	an accumulator array containing the bin counts of the target colour component throughout I

)

Define *histoIdx* as the corresponding index of a histogram bin

Define *cFF* as the boolean flag indicating condition fulfilment

Initialize the *histoArray* to all zeros

Initialize the *cFF* to **true**

For each pixel $\in I$

  For each condition in *conditionSet*

    $cFF \leftarrow cFF \wedge$ the fulfillment of the current condition

  End-For

  If cFF is **true**

    histoIdx = the corresponding index of the target colour component of the current pixel

    Increase *histoArray*[*histoIdx*] by 1

  End-If

End-For

Considering the application constraints in a chroma-keying setting, the background colours are expected to be highly uniform in saturation because saturation is lightness independent property and dyes with same or similar chemical compounds have similar optical properties. Therefore, the dominant hue composition H_B of the background colour can be determined by the $H_{pk\mathrm{i}}$ with the highest degree of saturation constancy. But in some cases foreground elements have similar colours with background, thus the only perceptual difference is lightness. In this case, additional manual inputs on lightness range of background colours are required.

Based on the information obtained from the histogram and analysis stages, an input image or video frame can be then segmented into three categories of mutually exclusive regions – certain foreground regions (T_F), certain background regions (T_B), and uncertain regions (T_U), which resemble the traditional trimap. In order to preserve fine details, we define the fourth category – detail regions (T_D), which contain the possible foreground but may not be fully included in the unknown region. This is achieved by first edge preserving filtering and then edge detections in the certain background regions (T_B) and uncertain regions (T_U) to find out trivial stimuli like hairs and edges. Then the detail regions (T_D) are widen to preserve the softness of details. An example of such quartered segmentation map is illustrated in Figure 3.9. The procedural description of the whole perceptual analysis process is listed in Table 3.2, Table 3.3, and Table 3.4.

Table 3.2: Pseudo Code for Perceptual Analysis

PROCEDURE Perceptual_Analysis

Input(s) (

hueBinInterval:	the bin interval for hue histogram
satBinInterval:	the bin interval for saturation histogram
lhtBinInterval:	the bin interval for lightness histogram
$I[\vec{\mathbf{x}}]$:	the Hue-Saturation-Lightness based image
maxHuePeakCnt:	the maximal allowed count of hue peaks
hueTolerance:	the tolerance of hue inconsistency
maxSatPeakCnt:	the maximal allowed count of saturation peaks
sPeakWidMetric:	the metric for measuring saturation peak
edgeExpWid:	the width to expand from edges to background

)

Output(s) (

colourInfoList:	a list of elements containing information on dominant colours in terms of (peak hue, peak saturation, peak saturation count, saturation range, lightness range)
$quadMap[\vec{\mathbf{x}}]$:	the quartered segmentation map of I

)

Define *bgIndex* as the index of element determined belonging to background

Initialize *colourInfoList* to $\varnothing$

Initialize *bgIndex* to **null**

Conditional_Histogram(*hueBinInterval*, *I*, 'Hue', $\varnothing$) for hue histogram

Find the first *maxHuePeakCnt* local maxima from the hue histogram in an order-preserving manner

Continued on Table 3.3

Table 3.3: Pseudo Code for Perceptual Analysis (Continued 1)

For each local maximum in the hue histogram

Conditional_Histogram(*satBinInterval*, *I*, 'Saturation',

{the absolute hue difference between the current local hue maxima and the hue component of a pixel is less than or equal to hueTolerance}

)

Find the first *maxSatPeakCnt* local maxima from the current saturation histogram

For each local maximum in the current saturation histogram

Measure the width of the peak shape according to *sPeakWidMetric*

Add the current hue, peak saturation, and the range of saturation peak into *colourInfoList* with **null** lightness range

End-For

End-For

Determine the dominant hue and saturation of background based on the peak saturation count and saturation range properties of elements $\in$ *colourInfoList*, store the index into *bgIndex*

For each element in *colourInfoList*

Conditional_Histogram(*lhtBinInterval*, *I*, 'Lightness',

{the hue and saturation components of a pixel are within the chrominance range constrained by the current element and *hueTolerance*}

)

Find the lightness range from the current lightness histogram

Update the current element with measured lightness range

End-For

Continued on Table 3.4

Table 3.4: Pseudo Code for Perceptual Analysis (Continued 2)

Link the head and tail of *colourInfoList* together to build a circular list while keeping the existing indices, and make *colourInfoList* circularly indexable

For each $\vec{\mathbf{x}}$ in spatial space of I

If the absolute difference between the hue of the current pixel and the hue of *colourInfoList*[*bgIndex*] is less than or equal to *hueTolerance*

$quadMap[\vec{\mathbf{x}}] \leftarrow 0$

Else-If the hue of the current pixel is between the hue of *colourInfoList* [*bgIndex*−1] and of *colourInfoList*[*bgIndex*+1]

$quadMap[\vec{\mathbf{x}}] \leftarrow 0.5$

Else

$quadMap[\vec{\mathbf{x}}] \leftarrow 1$

End-If

End-For

Perform edge detection on I and mark edges with a value of 0.75 in the corresponding position in *quadMap*

Expand edges into background with a radius of *edgeExpWid*

A comparison is conducted between Figure 3.9 and a manually drawn segmentation map as well as other two state-of-the-art commercial software that provides segmentation maps as intermediate results. The original video frame and other three segmentation maps as shown in Figure 3.10. The criterion is to categorize at least 2 noticeable hairs as T_U while minimizing the area of T_U. It is obvious that the generated quartered segmentation map is the most similar one to the manually drawn segmentation map. It is much cleaner in T_B, and contains more details such as the almost invisible single hair on the calvaria, and covers less greenish sheen on the skirt that may cause translucency in solid foreground.

Figure 3.9: Quartered segmentation map of the 1st frame of Godiva medium sequence from [3]: certain foreground regions (T_F) in white, certain background regions (T_B) in black, uncertain regions (T_U) in dark grey, and detail regions (T_D) in light grey.

Figure 3.10: The original video frame (top left, the 1st frame of Godiva medium sequence from [3]), the manually drawn segmentation map (top right), the segmentation map from Photo Key 6 Pro [77] (bottom left), and the segmentation map from Keylight 1.2 [78] (bottom right). Arrows indicate the position of the hardly visible hair.

3.3 Background Restoration

When background is unable to be observed directly, we often assume that the occluded regions are similar to the remaining parts of the background. With a fine quartered segmentation from perceptual analysis, we can start estimating those occlusions with image inpainting. An ideal chroma-key studio should have almost uniformly illuminated background. Thus a good clean background can be generated by filling occluded regions with mean colour, which is appeared to be used in many commercial chroma-key softwares. However, the lighting conditions and camera quality in real-world chroma-key studios varied from one another. For a scene in the left of Figure 3.11, a low-end camera may not provide colour compensations to contrast the background during shooting. Besides, alpha matting against monochromatic background has been extended to outdoor applications such as the one shown in the right of Figure 3.11. Therefore, a good inpainting of background is very important.

Figure 3.11: Indoor (left, photo courtesy of Hollywood Camera Work [3]) and outdoor (right, photo courtesy of American Movie Classics) chroma keying setting.

Outdoor illuminant is usually the daylight, which results in approximately linear variation in local luminance intensity changes because the sun is far away from the

background surface. Therefore, a good result may be generated from linear interpolation. But for indoor scenarios, multiple spotlights near the background surface may flaw the uniformity of luminance. In this case, the approximately linear variation does not hold. The intensity variation isolines of a clipped spotlight shade can be modelled by conic sections, such variation is also known cone attenuation effect.

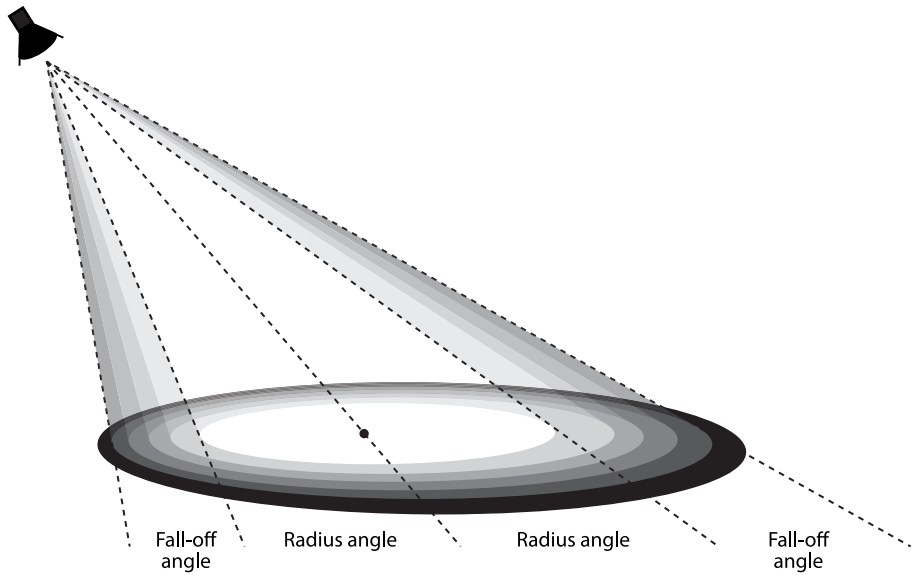


Figure 3.12: The most popular used model for cone attenuation effect of spotlight in computer graphics rendering.

The light intensity attenuates along the spotlight radius. However, because of the dynamic range compression of human eye, the centre region within a given radius appears to be much brighter and almost have full light intensity to human though it varies, while light intensity outside this region attenuates obviously. Therefore, in the field of computer graphics rendering, a model of spotlight effect widely used as shown in Figure 3.12. The light within the radius angle is assumed to have full intensity, and the attenuation outside is simulated with Hermite polynomials (3^{rd} or

higher order) as the interpolation function. Here we do not talk about its failure in physics but take the advantage of its relative simplicity and acceptable smoothness of gradient variation.

The 5^{th} order Hermite polynomial interpolation from Perlin [79] is used because of its human acceptable neutral and mathematically proven smoothness. Due to the fact that original paper only lists the final function, here it is mathematically derived. It starts with the generic 5^{th} order polynomial function $f(t)$:

$$f(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \tag{3.1}$$

Because the interpolation is between 0 and 1, there should be $f(0) = 0$ and $f(1) = 1$. Thus we can get:

$$\begin{cases} a_0 & = 0 \\ a_5 + a_4 + a_3 + a_2 + a_1 + a_0 & = 1 \end{cases} \tag{3.2}$$

Then, take first derivative of Equation 3.1:

$$f'(t) = 5a_5t^4 + 4a_4t^3 + 3a_3t^2 + 2a_2t + a_1 \tag{3.3}$$

Since the smooth transition at both endpoints of the interpolation function is desired, the first derivative of Equation 3.1 at both endpoints should be zero. By applying $f'(0) = 0$ and $f'(1) = 0$, we can get:

$$\begin{cases} a_1 & = 0 \\ 5a_5 + 4a_4 + 3a_3 + 2a_2 + 1a_1 & = 0 \end{cases} \tag{3.4}$$

Next, take second derivative of Equation 3.1:

$$f''(t) = 20a_5t^3 + 12a_4t^2 + 6a_3t + 2a_2 \tag{3.5}$$

To enforce the smoothness of transition at both endpoints of the interpolation function, the second derivative of Equation 3.1 at both endpoints should be zero. By applying $f''(0) = 0$ and $f''(1) = 0$, we can get:

$$\begin{cases} 2a_2 & = 0 \\ 20a_5 + 12a_4 + 6a_3 + 2a_2 & = 0 \end{cases} \tag{3.6}$$

Now, by solving Equation 3.2, 3.4, and 3.6 together, we can obtain the coefficients of the polynomial:

$$\begin{cases} a_0 = 0 \\ a_1 = 0 \\ a_2 = 0 \\ a_3 = 10 \\ a_4 = -15 \\ a_5 = 6 \end{cases} \tag{3.7}$$

Therefore, the designate 5^{th} order Hermite polynomial interpolation function for interpolating from a value a to another value b is:

$$
\begin{aligned}
H(t,a,b) &= \begin{cases} (b-a)f(t)+a, & (a<b) \\ (a-b)f(t)+b, & (a>b) \end{cases} \\
&= \begin{cases} (b-a)(6t^5-15t^4+10t^3)+a, & (a<b) \\ (a-b)(6t^5-15t^4+10t^3)+b, & (a>b) \end{cases}
\end{aligned}
\tag{3.8}
$$

A plot of the interpolation function is shown in Figure 3.13. An actinomorphic sampling method is deployed for both linear and Hermite polynomial interpolation. As illustrated in Figure 3.14, along each ray started from a given pixel $p \in \mathrm{T}_U \bigcup \mathrm{T}_D$,

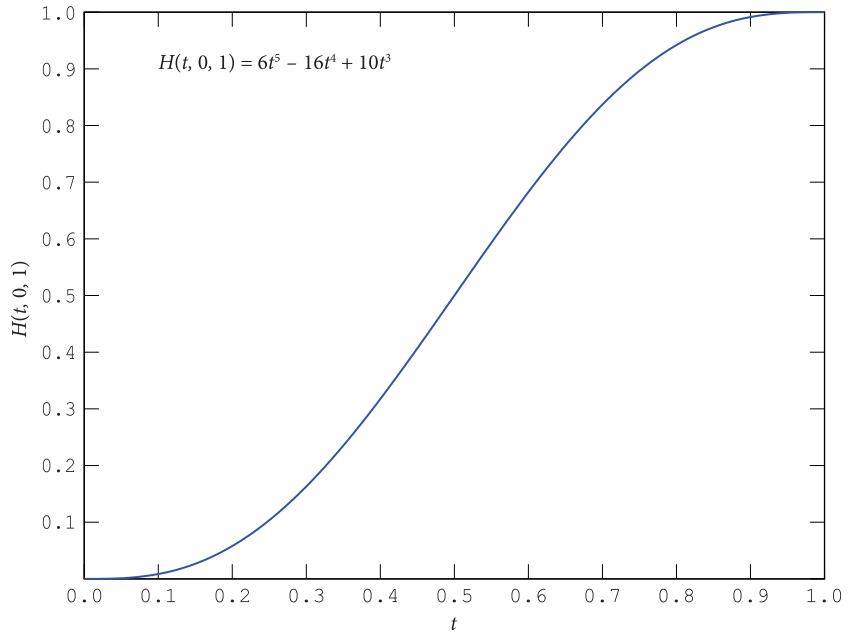


Figure 3.13: Plot of 5^{th} order Hermite polynomial interpolation function used in simulating cone attenuation effect.

Figure 3.14: Illustration of actinomorphic sampling. The dotted lines indicate searching paths, colour filled circles indicate gathered background samples. The radius for searching background samples is limited in a certain range, rays without gathering background samples are terminated with a red 'X'.

at most one spatially nearest background sample will be gathered within a certain searching radius. If a background sample is not found, the global average background colour $\bar{\mathbf{B}}$ is used. If the samples gathered along all rays have similar value to $\bar{\mathbf{B}}$, the linear interpolation is used. Otherwise, an approximation of fitting the cone attenuation effect is performed based on image space spatial weights. First, a spatially weighted average colour of gathered samples is calculated as in Equation 3.9.

$$\bar{\mathbf{B}}_q = \frac{\sum_i \mathbf{B}_{q_i} \|p - q_i\|}{\sum_i \|p - q_i\|} \tag{3.9}$$

where q_i denotes for the coordinates of gathered pixel with index i. And the estimated background is obtained as in Equation 3.10.

$$\hat{\mathbf{B}}_p = \frac{\sum_i H(\bar{\mathbf{B}}_q, \mathbf{B}_{q_i}, \bar{\mathbf{B}})\|p - q_i\|}{\sum_i \|p - q_i\|} \tag{3.10}$$

Though in the spotlight model, a dim or bright lightness level should be used accordingly instead of $\bar{\mathbf{B}}$, but using $\bar{\mathbf{B}}$ actually generates acceptable smooth gradient in restored background with less computation which is critical to real-time applications. Besides, constant hue and saturation in background colours are also assumed thus only lightness values are interpolated. Overall, the proposed inpainting method can be summarized as described in Table 3.5 and Table 3.6.

Figure 3.15 show the comparison between the results of the Robust Smoothing [80] with different parameters, the commercial software Teorex Inpaint version 6.0 [81], and the proposed method. Because the background is smooth in gradient, other inpainting algorithms solely based on image block or texture are not chosen. The Robust Smoothing with 1,000 and 20,000 iterations costs about 10 and 30 minutes respectively in its MATLAB implementation. The more iterations performed, the better smoothness generated. The Teorex Inpaint costs about 10 seconds but the result is not good. Several significant discontinuities can be noticed along the segmentation boundaries and the gradient of intra-segmentation smoothness appears unnature. While the former two algorithms do not provide the option to inpaint pixels in $\mathrm{T}_U \bigcup \mathrm{T}_D$ without sampling from T_F, the proposed method only inpaints pixels in $\mathrm{T}_U \bigcup \mathrm{T}_D$ by sampling from T_B with a single pass less than 10 milliseconds and generates relatively better and more visually reasonable result.

Table 3.5: Pseudo Code for 'Background Restoration on Demand'

PROCEDURE Background_Restoration_on_Demand

Input(s) (

$L[\vec{\mathbf{x}}]$:	the linear colour space image
$quadMap[\vec{\mathbf{x}}]$:	the quartered segmentation map
$barBg$:	$\bar{\mathbf{B}}$
$rayCnt$:	the number of searching direction(s)
$rayStep$:	the spatial increment along a searching direction
$rayRadius$:	the searching radius
$bgVarTh$:	the threshold of variance of background samples

)

Output(s) (

$hatBL[\vec{\mathbf{x}}]$:	the linear colour space image with restored background in demanded regions

)

Define *angInc* as the angular increment value

Define *rayDir* as the direction of the current searching path

Define *bgFound* as the boolean flag indicating whether background sample is found or not

Define *sampleInfoList* as a list of elements containing information on gathered samples in terms of (colour, distance)

Initialize *angInc* to $2\pi/rayCnt$

Initialize *rayDir* to a random number between 0 and *angInc*

Initialize *sampleInfoList* to $\varnothing$

For each pixel $\in L$

If the current pixel is categorized neither as foreground nor as background according to *quadMap*

bgFound $\leftarrow$ **false**

Continued on Table 3.6

Table 3.6: Pseudo Code for 'Background Restoration on Demand' (Continued)

Loop

Loop

Move along the *rayDir* with a step of *rayStep*

If the current position is background according *quadMap*

Sample the corresponding pixel from L

Add the current colour of sample and searching distance into *sampleInfoList*

bgFound $\leftarrow$ **true**

End-If

Until *bgFound* is **ture** or distance exceeds *rayRadius*

If *bgFound* is **false**

Add *barBg* and *rayRadius* into *sampleInfoList*

End-If

rayDir $\leftarrow$ *rayDir* + *angInc*

Until *rayDir* is greater than 2π

Calculate the variance of sampled background colours

If the variance is less than or equal to *bgVarTh*

Perform linear interpolation

Else

Apply Equation 3.9 and 3.10 to approximate the cone effect

End-If

Store the interpolated value into the corresponding position of *hatBL*

Else

Store the current pixel into the corresponding position of *hatBL*

End-If

End-For

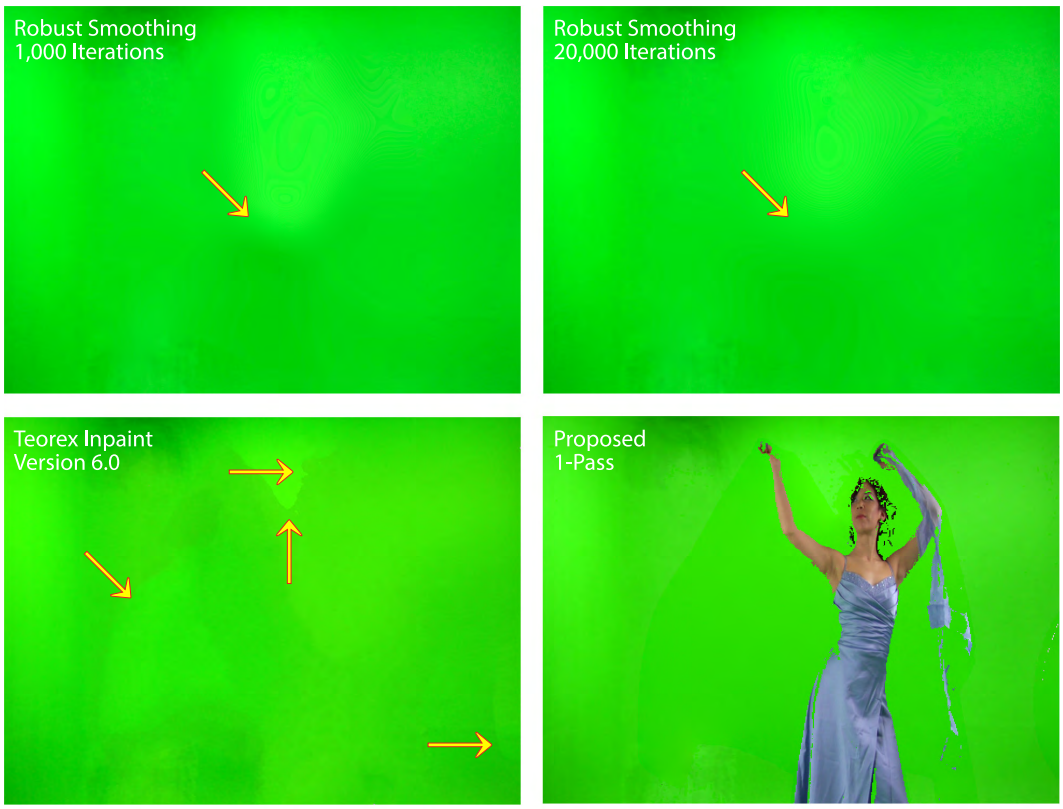


Figure 3.15: Comparison between different inpainting algorithms: Robust Smoothing with 20,000 iterations (top left), Robust Smoothing with 1,000 iterations (top right), Teorex Inpaint 6 (bottom left), and the proposed method (bottom right, partially inpainted to reduce computational cost). Arrows indicate the obvious flaws.

3.4 Foreground Colour and Alpha Prediction

The precise trimap and reasonable restoration of background make the estimation of alpha values and foreground colours relatively easier. Traditional sampling based approaches use limited number of samples from certain foreground (T_F) and certain background (T_B) regions, leading to mathematically right but sub-optimal colours to human visual preference and may cause discontinuity with its affinity. We gather

foreground samples not to form a colour sample set, but to determine the range of foreground colours. Observations show that colours inside a colour space can be indistinguishable to the average human eye in a certain vicinity known as MacAdam ellipse [82], and MacAdam ellipses are expected to be near-uniform in perceptually uniform colour spaces. Therefore, we can safely determine the saturation and lightness information around the foreground hue peaks but assume the same hues as the corresponding hue peak. It is believed that this approach is able to enrich the variety of estimated foreground colours in our ideology.

The possible foreground chrominance ranges of a given pixel $p \in \mathrm{T}_U \bigcup \mathrm{T}_D$ could be estimated based on the recovered background colour on the corresponding position $\hat{\mathbf{B}}_p$ and the observed colour of the pixel $\mathbf{O}_p = \mathbf{I}_p$ in conjunction with the chrominance tolerance regions on each dominant hues determined from the perceptual analysis. The saturation ranges of foreground colours is constrained by drawing a ray from the point of $\hat{\mathbf{B}}_p$ through the point of $\mathbf{O}_p$ on the hue-saturation chrominance plane in CIECAM02 colour appearance model, as depicted in Figure 3.16. Those chrominance tolerance regions that have intersections with the ray are preferred in the proposed method. Their centroids of the intersections are chosen as foreground chrominance candidates, and are used for further estimation on the lightness and the alpha value.

Because the possible foreground colour has a hue in either $(H_{\mathbf{O}_p}, H_{\mathbf{B}_p} - \pi]$ if $H_{\mathbf{O}_p} < H_{\mathbf{B}_p}$, or $(H_{\mathbf{O}_p}, H_{\mathbf{B}_p} + \pi]$ if $H_{\mathbf{O}_p} > H_{\mathbf{B}_p}$, the candidates can be further screened to half of its number statistically. The lightness estimation of foreground is done by de-blending the luminance values of $\mathbf{B}_p$ and $\mathbf{O}_p$ in CIE 1931 XYZ colour space with $\hat{\alpha}_p^{(H,\mathrm{i})}$ in an enumerated set $A_p \subset [0, 1]$ for estimation of α_p, where H is the estimated hue and i indicates the index of enumeration performed. The cardinality of A_p is

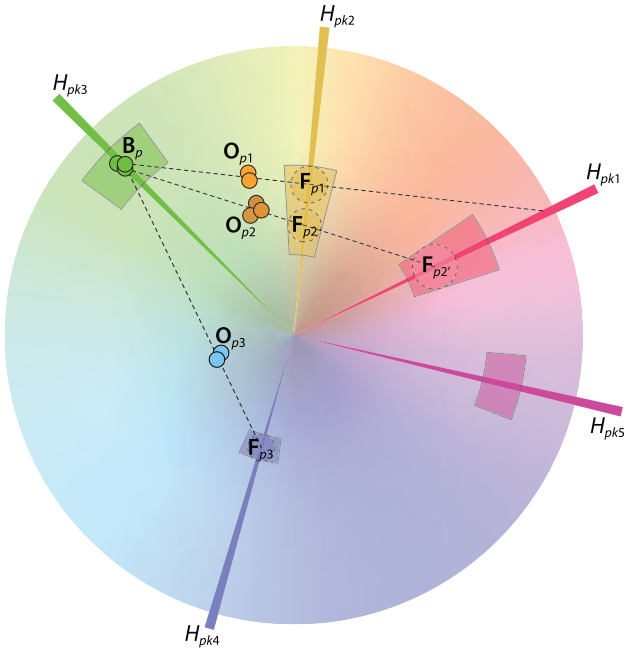


Figure 3.16: Chrominance prediction of foreground on the Hue-Saturation chrominance plane (range is limited in circle for illustration).

manually constrained for speed optimization by clipping the enumeration in small range around the r_p^λ, where r_p is the ratio of $\|\mathbf{O}_p - \mathbf{B}_p\|$ to $\|\mathbf{F}_p - \mathbf{B}_p\|$ projected on the Hue-Saturation chrominance plane and λ is an adjustment parameter. We manually add 0 and 1 to A_p in case they do not belong to the clipped range. The estimated luminance values are then converted to lightness values in CIECAM02 colour appearance model. The $\hat{\alpha}_p^{(H,\mathrm{i})}$ that produce lightness values in agreement with the lightness range of corresponding chrominance region from the perceptual analysis are defined as preferred alpha values and will be evaluated. The foreground colour estimation $\hat{\mathbf{F}}_p^{(H,\mathrm{i})}$ is calculated by paring the lightness values from estimated luminance values, and the foreground chrominance candidates. The best estimation of $(\hat{\alpha}_p, \hat{\mathbf{F}}_p)$ pair is obtained by minimizing the estimation to observation distortion function d_p:

$$d_p = \|\mathbf{O}_p - (\hat{\alpha}_p^{(H,\mathrm{i})}\hat{\mathbf{F}}_p^{(H,\mathrm{i})} + (1 - \hat{\alpha}_p^{(H,\mathrm{i})})\mathbf{B}_p\| \tag{3.11}$$

$$(\hat{\alpha}_p, \hat{\mathbf{F}}_p) = \underset{\hat{\alpha}_p, \hat{\mathbf{F}}_p}{\operatorname{argmin}}\ d_p(\hat{\alpha}_p^{(H,\mathrm{i})}, \hat{\mathbf{F}}_p^{(H,\mathrm{i})}) \tag{3.12}$$

If no preferred alpha value is found or the minimal d_p still exceeds the threshold, the conditions on lightness will be relieved. More estimated ($\hat{\alpha}_p$, $\hat{\mathbf{F}}_p$) pair will be evaluated and the one minimizes d_p will be chosen. The procedural description of the whole process is listed in Table 3.7, Table 3.8, Table 3.9, and Table 3.10.

Table 3.7: Pseudo Code for 'Foreground Colour and Alpha Prediction'

PROCEDURE Foreground_Alpha_Prediction

Input(s) (

$I[\vec{\mathbf{x}}]$:	the Hue-Saturation-Lightness based image
$L[\vec{\mathbf{x}}]$:	the image in linear colour space
$hatBL[\vec{\mathbf{x}}]$:	the linear colour space image with restored background in demanded regions
$quadMap[\vec{\mathbf{x}}]$:	the quartered segmentation map
colourInfoList:	the list of elements containing information on dominant colours in terms of (peak hue, peak saturation, peak saturation count, saturation range, lightness range)
emuRadius:	the radius for alpha enumeration
emuSize:	the radius for alpha enumeration
lambda:	the adjusting parameter λ in r_p^{λ}
dpThrshld:	the threshold of highest acceptable d_p

)

Output(s) (

$hatFL[\vec{\mathbf{x}}]$:	the linear colour space foreground estimation

Continued on Table 3.8

Table 3.8: Pseudo Code for 'Foreground Colour and Alpha Prediction' (Continued 1)

hatAlp[$\vec{\mathbf{x}}$]: the estimated alpha map

)

Define *hatB*[$\vec{\mathbf{x}}$] as the Hue-Saturation-Lightness based image with restored background in demanded regions

Define *cRatio* as the ratio of $\|\mathbf{O}_p - \mathbf{B}_p\|$ and $\|\mathbf{F}_p - \mathbf{B}_p\|$ projected on the Hue-Saturation chrominance plane

Define *aFList0* as a list of elements containing (foreground colour, alpha value) pairs of favourite candidates

Define *aFList1* as a list of elements containing (foreground colour, alpha value) pairs of potential candidates

Define *bestPair* as a (foreground, colour, alpha value) pair

Initialize *hatB* to the Hue-Saturation-Lightness based representation of *hatBL*

For each $\vec{\mathbf{x}}$ in the input image dimension

Initialize *aFList0* to $\varnothing$

Initialize *aFList1* to $\varnothing$

If the current pixel is categorized neither as foreground nor as background according to *quadMap*

Build a ray from the colour coordinate of *hatB*[$\vec{\mathbf{x}}$] through the colour coordinate of $I[\vec{\mathbf{x}}]$

Draw the projection of the ray on the chrominance plane

Find the foreground chrominance candidates (intersections of the ray and the chrominance ranges described by elements in *colourInfoList*)

For each foreground chrominance candidates

Calculate the *cRatio*

Enumerate *emuSize* alpha values between $cRatio^{lambda}$ - *emuRadius* and $cRatio^{lambda}$ + *emuRadius* with an interval of 2 * *emuRadius* / (*emuSize - 1*), add value 0 and value 1 into the enumeration

Continued on Table 3.9

Table 3.9: Pseudo Code for 'Foreground Colour and Alpha Prediction' (Continued 2)

For each enumerated alpha value

Estimate the linear colour space luminance by solving Equation 1.2 with $hatBL[\vec{\mathbf{x}}]$, $hatBL[\vec{\mathbf{x}}]$, and the current instance of alpha enumeration

Calculate the lightness value from estimated luminance value

Compare the lightness value with the lightness range of corresponding chrominance region described by *colourInfoList*

Get the linear colour space representation of the colour with current lightness value and chrominance values

If lightness value falls in the lightness range

Add the linear colour space representation and the current instance of alpha enumeration as a pair into *aFList0*

Else

Add the linear colour space representation and the current instance of alpha enumeration as a pair into *aFList1*

End-If

End-For

End-For

If *aFList0* is not $\varnothing$

Find out the element in *aFList0* containing the (foreground, colour, alpha value) pair that minimizes Equation 3.11, and store it into *bestPair*

End-If

If *aFList0* is $\varnothing$ or the minimal distortion according to Equation 3.11 exceeds *dpThrshld*

Find out the element in both *aFList0* and *aFList1* containing the (foreground colour, alpha value) pair that minimizes Equation 3.11, and store it into*bestPair*

Continued on Table 3.10

Table 3.10: Pseudo Code for 'Foreground Colour and Alpha Prediction' (Continued 3)

End-If

$hatFL[\vec{\mathbf{x}}] \leftarrow$ foreground colour of *bestPair*

$hatAlp[\vec{\mathbf{x}}] \leftarrow$ alpha value of *bestPair*

Else

$hatFL[\vec{\mathbf{x}}] \leftarrow L[\vec{\mathbf{x}}] * quadMap[\vec{\mathbf{x}}]$

$hatAlp[\vec{\mathbf{x}}] \leftarrow quadMap[\vec{\mathbf{x}}]$

End-If

End-For

3.5 Depth Perceiving

Human can perceive depth information of a scene with binocular disparity and experiential knowledge on the dimensions of objects. To simulate such process, a stereo matching is performed on the quartered segmentation map in conjunction with pixel-wise lightness and its gradient magnitude. The visual contents are also analyzed to provide the reference information of objects.

Stereo matching is a process that determines the disparities of pixels in a pair of stereoscopic images. Though there are over 100 algorithms listed in the Middlebury Stereo Evaluation website [83][84], the two main solutions to the problem are spatial domain based matching and transformed domain based analysis. The spatial domain based matching algorithms compare blocks of pixels in spatial features (e.g. sum of absolute difference, cross correlation, etc.) along the baseline direction to find the best matching position, while the transform domain based analysis uses phase

correlation in transform domain (e.g. Fourier transform domain, wavelet transform domain, etc.) to determine stereo correspondence. Most of the new work focused on the preprocessing, refinement and optimization using image warping, clustering, segmentation, and propagation.

In filmmaking and television industries, the depth information is used for special effects such as augmented reality interaction and focus blur, rather than sub-pixel level accuracy modelling. Thus an object level depth information is sufficient for such application. The chroma keying application provide us the most optimized solution on object segmentation, it is a big advantage for object level depth analysis. In addition, the chroma keying setting is a controlled environment and the stereoscopic cameras are set to be horizontally aligned, thus only horizontal disparity is concerned in the framework design.

The quartered segmentation map is almost textureless, making it harder to determine the stereo disparity. Considering the speed requirement of stereoscopic 3D chroma keying application, it is better to overcome this disadvantage without using big aggregation window. Therefore, a new feature-triplet map $T(Q, L, G_{mag})$ is defined where Q is the quartered segmentation map, L is the lightness, and G_{mag} is the gradient magnitude of lightness. A simple spatial domain horizontal matching using normalized cross-correlation is used for solving stereoscopic correspondence, as illustrated in the bottom of Figure 3.17. This block based method is chosen because of its pixel independency that may be efficiently implemented on GPU in parallel, which better fits the real-time requirement. In order to find where a given pixel $\vec{\mathbf{x}}$ in the left feature map T_L that appears in the right feature map T_R, two fixed-size matching windows in the same shape W sample around the given pixel at location $\vec{\mathbf{x}}$ on T_L and

the expected location $\vec{\mathbf{x}} + \vec{\mathbf{d}}$ on T_R respectively, where $\vec{\mathbf{d}}$ is the horizontal disparity that will be changing during the matching process. Then the similarity between samples from two matching windows is determined by the normalized cross-correlation function:

$$\mathrm{NCC}[\vec{\mathbf{x}}, \vec{\mathbf{d}}] = \frac{\sum_{\vec{\mathbf{i}} \in W} T_L[\vec{\mathbf{x}} + \vec{\mathbf{i}}] \cdot T_R[\vec{\mathbf{x}} + \vec{\mathbf{d}} + \vec{\mathbf{i}}]}{\sqrt{\sum_{\vec{\mathbf{i}} \in W} T_L^2[\vec{\mathbf{x}} + \vec{\mathbf{i}}] \cdot \sum_{\vec{\mathbf{j}} \in W} T_R^2[\vec{\mathbf{x}} + \vec{\mathbf{d}} + \vec{\mathbf{j}}]}} \tag{3.13}$$

and the $\vec{\mathbf{d}}$ that produces the highest correlation along a horizontal baseline are determined as the best matching disparity.

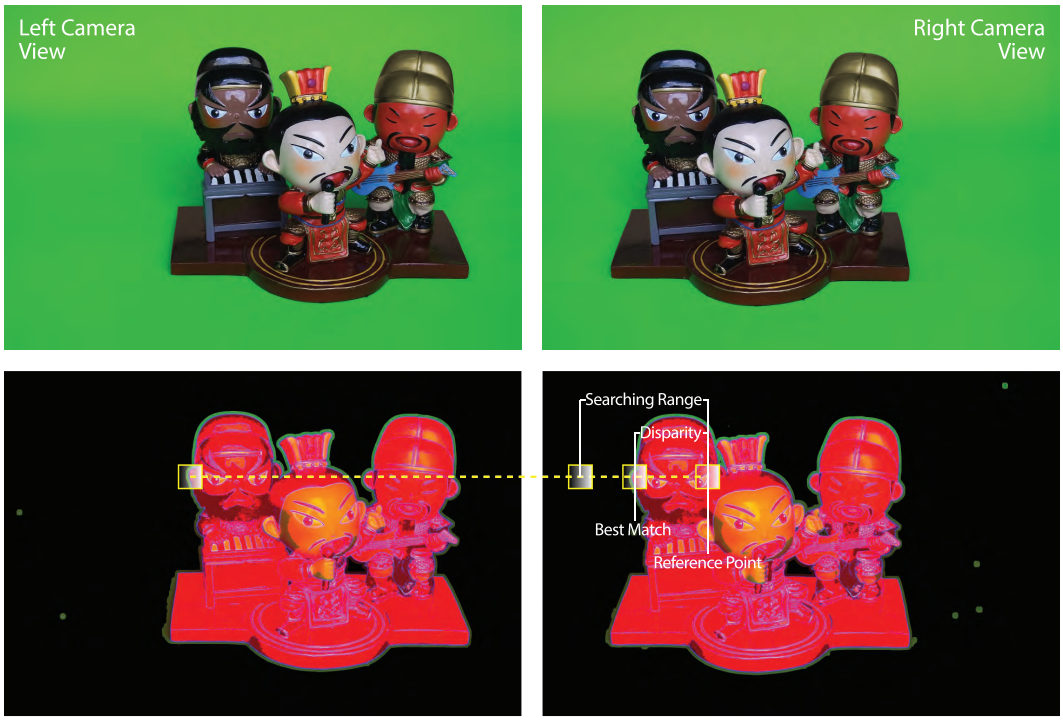


Figure 3.17: Illustration of stereo matching using normalized cross-correlation: the original stereoscopic pair (top), and the $T(Q_p, L_p, G_{mag})$ feature-triplet map (bottom). The matching window slides along the baseline within a certain range, the distance from the reference point to the best match is the relative disparity.

To further reduce the time consumption of stereo matching, the stereoscopic pair is down sampled to a lower resolution. By performing the matching pixel by pixel, the initial binocular disparity map is generated. Then the parameters of lens such as field-of-view and focal length are used to convert the initial disparity map into initial depth map. However, the initial low-resolution disparity map has significant discontinuities and does not have good object boundary constrains. Meanwhile, the chrominance distance from an observed colour to the dominant background colour on the Hue-Saturation plane has good boundaries of foreground objects. Therefore, using the chrominance distance as a reference is an advantageous way to constrain the object boundaries in the initial disparity map. In general stereo matching problem, propagation based methods are used to perform the optimization, one example achieving good results according to the Middlebury Stereo Evaluation website is the Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure [85]. But propagation based refinement methods are time-consuming, making it unacceptable in real-time applications. Another way is to use bilateral filtering [86], which is a spatial domain filter with weight preserving capability. When it is applied with external weightings, it jointly fuses the characters from both. It has been used in HDR tone-mapping up-sampling, colourization, and depth-map up-sampling [87].

A joint bilateral filtering is deployed in the proposed framework, which filters the initial depth map using the normalized chrominance distance from an observed colour to the dominant background colour on the Hue-Saturation plane as the adaptive kernel. The idea is to apply a spatial filter to the binocular disparity map $\mathbf{Z}$, while a corresponding range filter is jointly applied on the normalized chrominance distance.

Let $\hat{\mathbf{D}}$ denote the normalized chrominance distance from an observed colour to the dominant background colour on the Hue-Saturation plane, $\tilde{\mathbf{Z}}$ denote the refined depth map, and Ω_p denote coordinate set of the pixels in the small window around pixel p, the pixel-wise refinement can be obtained as:

$$\tilde{\mathbf{Z}}_p = \begin{cases} \dfrac{\sum_{q\in\Omega_p} \mathbf{Z}_q h(p,q)}{\sum_{q\in\Omega_p} h(p,q)}, & \hat{\mathbf{D}}_p \neq 0 \\ 0, & \hat{\mathbf{D}}_p = 0 \end{cases} \tag{3.14}$$

where $h(p,q) = f(\|p-q\|)g(\|\hat{\mathbf{D}}_p - \hat{\mathbf{D}}_q\|)$, and disparity of 0 indicates infinitive far point since no background is placed.

Chapter 4

The Efficient Implementation

Based on the designed framework of stereoscopic 3D chroma keying, a prototype system has been implemented with special optimizations for heterogeneous computing. Both GPU and CPU are utilized with carefully distributed tasks. The pixel-processing insensitive processes such as chroma keying and stereo matching are running on the GPU, while the CPU is responsible for perceptual analysis and process dispatching. The implementation shows its efficiency on both desktop computers and embedded systems. The detailed ideas and methods of implementation will be introduced and discussed in this chapter.

4.1 Asynchronous Parallelism for Heterogeneous Computing

Architecture is essential to support any framework that utilize it. To best fit the requirement of an application, a dedicated architecture is desirable. By analyzing the

characteristics of chroma keying application, it is found that the colour composition of both foreground and background does not change drastically between consecutive frames of a scene. This phenomenon suggests that frame-by-frame perceptual analysis is unnecessary, thus the perceptual analysis can be performed on a rate lower than frame rate. Lower analysis rate not only reduces the computational cost, but also saves the memory bandwidth for transferring frame data. Besides, once the quartered segmentation map has been generated, the stereo matching process can start immediately without waiting for background restoration and alpha estimation. Based on this idea, the architecture of the system is implemented. It consists of four dedicated process running in parallel – basic controlling and dispatching process, perceptual analysis process, chroma keying process, and stereo matching process. The interprocess parallelism and intra-process schedule is shown in Figure 4.1.

Ideally, this workflow is running optimally on a system with two CPUs and two GPUs when each processor is responsible for a dedicated process. The two GPUs can share their memories with technologies such as NVIDIA GPUDirect [88] to reduce copying overhead between each GPU accessible memory, which requires workstation level professional GPUs. However, due to research budget limitation, the workflow is implemented on only one GPU thus two GPU processes are run serially. Besides, by taking advantage in the proposed pixel-independent processing, the implementation is able to running on multiple GPUs without memory sharing by utilizing Split Frame Rendering (SFR) via NVIDIA SLI [89] or AMD CrossFire [90]. For a video at 60 frame-per-second, the maximum time from the end of video uploading to the rendering of final composite should less than 1/60 second. With the careful and detailed schedule, it is time to implement and optimize each task.

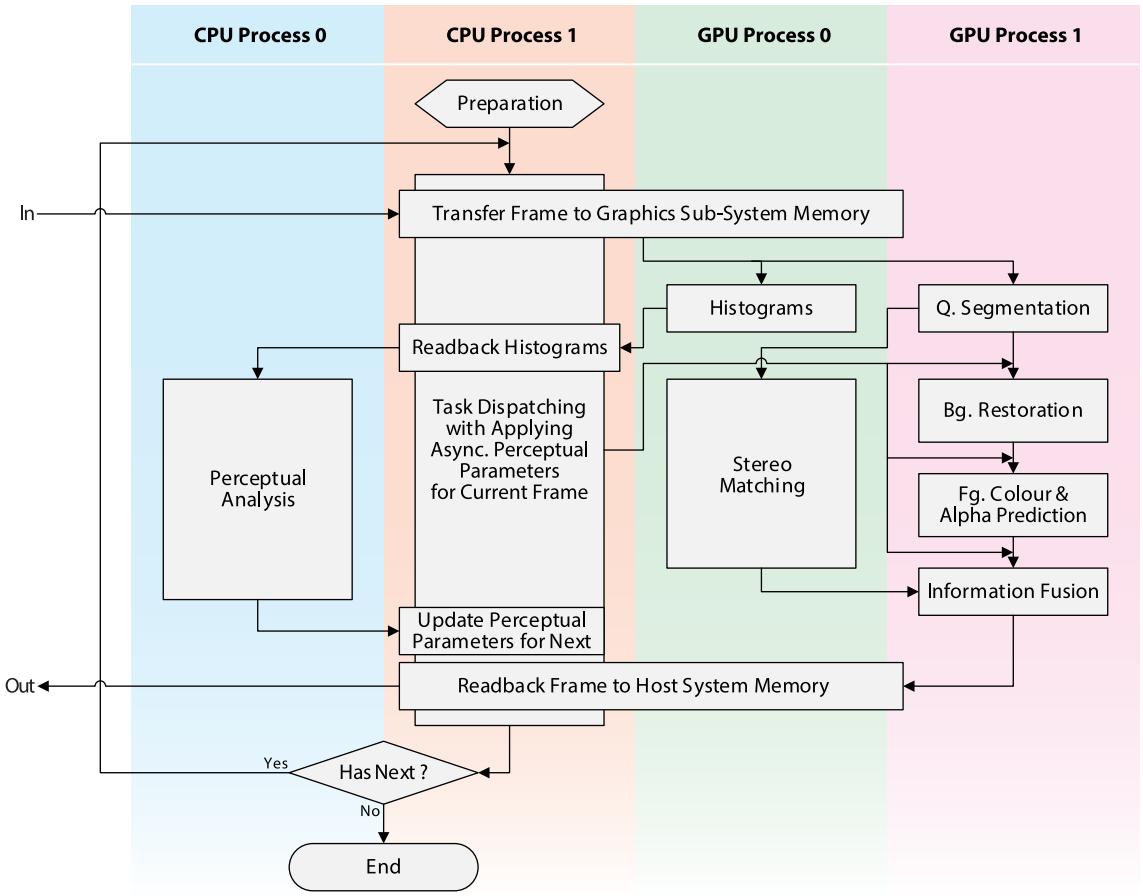


Figure 4.1: Workflow of implemented system.

4.2 Asynchronous Frame Transfer and Zero-Copy

Traditional rendering utilizes double-buffer scheme to parallel the displaying and processing of graphics, where the current frame starts loading, processing, and drawing to the current buffer during the displaying of the previous frame from the previous buffer. On the other hand, the video acquisition device is buffered and working independently with GPU and thus can be running in parallel naturally. Since the bit rate of an uncompressed video is constant, the time used for acquiring or displaying

one single frame is always same as the frame interval for a particular video stream. Besides, frame transmissions between sub-systems are usually synchronized to extra timing reference signal in video industry. Therefore, if the time used for single frame transferring and processing is less than the frame interval, the video stream is always continuous with at least a two-frame delay introduced by the two buffering sub-systems, as illustrated in Figure 4.2.

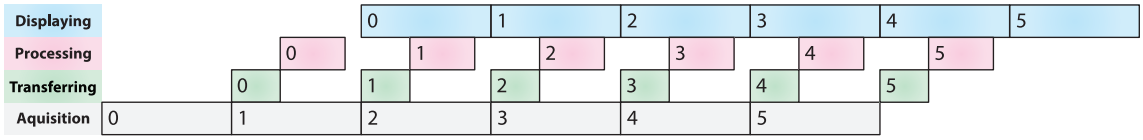


Figure 4.2: Asynchronous displaying buffer with vertical synchronization in traditional rendering: the time from the beginning of transfer to the end of rendering is less than one frame interval, a captured frame is always displayed at 2 frame intervals later.

But if the processing of a single frame is too long to keep within the frame interval, there will be frame loss as illustrated in Figure 4.3.

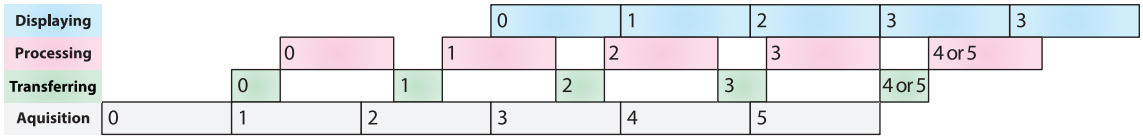


Figure 4.3: Asynchronous displaying buffer with vertical synchronization in traditional rendering: from the beginning of transfer to the end of rendering exceeds one frame interval, frame dropped according to acquisition buffering scheme, and previous frame is displayed again if current frame is not ready. If double-buffer scheme is used in acquisition, either frame 4 or frame 5 is transferred to the GPU accessible memory.

Modern researches on parallel computing focus on how algorithms can be distributed in parallel and then aggregated, while the data transfer between heterogeneous sub-systems (e.g. from host CPU to GPU) is usually not optimized. For production level video processing, the stream is uncompressed and there are huge

volumes of streaming data moving between heterogeneous sub-systems. Even a 2K (1920 px × 1080 px) resolution video in 10-bit 4:2:2 chroma-subsampling at 60 frame-per-second according to industrial standard SMPTE 274M [91] has a bit rate of 3G. It takes about 0.763 ms per frame, which is almost 1/20 of frame duration. Considering the computing density of the proposed framework, it is necessary to leave enough time for processing while the transferring is being done in parallel.

Therefore, the asynchronous transfer technique is utilized. A double-buffer scheme is deployed for asynchronous frame transfer from video acquisition device to GPU accessible memory. Thus, while the GPU is processing the previous frame from the previous buffer, the current frame from video acquisition device can be transferred to another. This technique provides more GPU time for data processing. However, the double-buffer scheme for video frame acquiring introduces one more frame delay for synchronous output, as illustrated in Figure 4.4, this is not optimal.

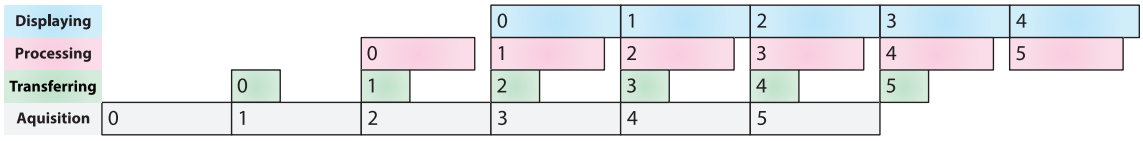


Figure 4.4: Asynchronous transfer rendering with vertical synchronization, resulting more GPU time for pixel processing but one more frame delay as well.

To cope with the additional frame delay, buffer sharing between the video acquisition device and the GPU is utilized, which is usually referred as the zero-copy technique. In this way, the video acquisition device can store data directly into the GPU accessible memory and thus the processing can start immediately after the last pixel is acquired. The pipeline becomes more condensed and time-efficient, as illustrated in the bottom of Figure 4.5.

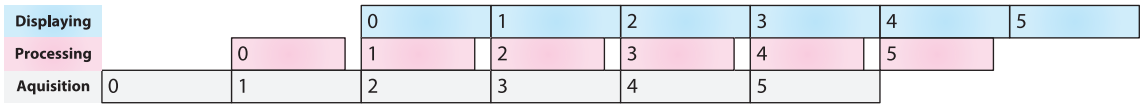


Figure 4.5: Zero-copy transfer rendering with vertical synchronization: frame is directly stored in the GPU accessible memory and thus no extra time is needed for copying frame between different sub-systems.

However, the zero-copy implementation is platform dependent. Most desktop GPUs that have dedicated graphics memory is not capable of direct zero-copy due to the different addressing of heterogeneous memories. Solutions from NVIDIA and AMD uses host system memory as a transit point and DMA controllers to free the burden from CPU and GPU [92][93], but extra delay may be introduced. For some embedded systems, CPU and GPU share the same memory and thus direct zero-copy can be achieved. The direct zero-copy is critical in embedded systems because GPUs for embedded systems usually have lower computational power. This technique helps the implementation of real-time chroma keying using the 2D part of the proposed chroma-keyer on an Imagination Technologies PowerVR Series 6 embedded GPU and NVIDIA GK20A embedded GPU.

4.3 Efficient Conditional Histogram Using Data Scattering on GPU

In the proposed stereoscopic 3D chroma keying framework, the three histograms play important roles in perceptual analysis. How to perform three histograms efficiently is critical to the overall performance of the proposed framework. A GPU have at least two kinds of processors that are called ‘shaders’ – vertex shaders, and pixel shaders.

Both of them are much simpler than general CPU thus the computational power of a single shader is limited. Modern GPUs usually integrate the two shaders into a unified shader. Therefore, a single GPU is a processing unit consists of multiple unified shaders optimized for vertex and pixel processing on predefined pixel coordinates, while the shaders are working in parallel, the aggregated computational power is huge. So it is not efficient to use them traversing a video frame in loops of pixel reading by using a few of the shader. However, reading back the data from GPU memory to CPU accessible memory is time consuming and will reduce the memory and bus throughput significantly. Besides, even CPU is more powerful to perform histogram, the performance is still not acceptable in a real-time application.

In 2007, Scheuermann and Hensley proposed the efficient histogram generation using scattering on GPUs [94]. They use High-level Shader Language (HLSL) [95] to build the vertex shader to place the vertex on bin position of a histogram according to the pixel value, and utilize additive blending to accumulate the pixel count. They gained a significant performance improvement from this strategy because each vertex can activate a vertex shader thread and thus the pixels are reading in parallel. Besides, the additive blending is an optimized operation on GPU with dedicated circuits, the cost is much lower than programming code running on shaders. Though there is NVIDIA CUDA [96] based histogram generator which claims higher performance with data wrapping [97], it is doubtful because they did not provide any comparison and they used programming code for accumulation. The drawback of using hardware accumulation is that the accuracy is implementation dependent. Some hardware only allow a summation within a certain range that is much smaller than usable, and some other hardware violates the data coherence (e.g. accumulation based on a previously

fetched value). However, for the proposed perceptual analysis, only approximate distributions of different perceptual components are needed.

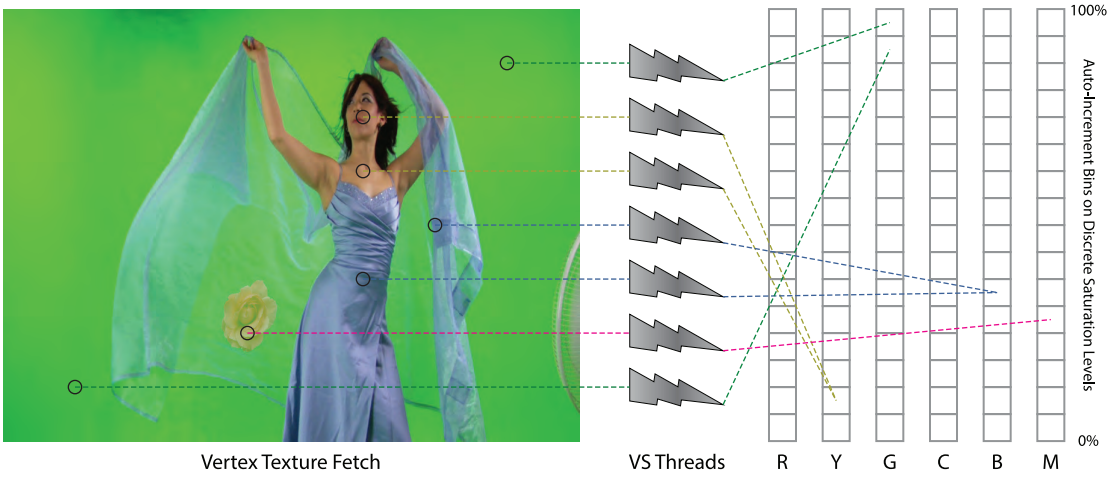


Figure 4.6: Illustration of GPU scattering for saturation histogram computations on each dominant hues which are assumed to be named colours as red, yellow, green, cyan, blue, and magenta only for illustration though they were determined by the preceding hue histogram.

In the proposed implementation, OpenGL Shading Language (GLSL) [98] is used. Since the Render to Vertex Buffer (R2VB) method in [94] requires additional rendering pass and thus introduces more delay in its whole process, the Vertex Texture Fetch (VTF) method is used. Different conditions on performing three kinds of histograms are efficiently switched using vertex shader subroutines [99] to avoid run-time conditional decisions on histogram procedure selection. And for saturation and lightness histograms, multiple output arrays are used, as illustrated in Figure 4.6 for saturation hisgograms. The test performed on multiple platforms including GPUs from NVIDIA (GK20A and GK107), Intel (HD Graphics 4000), and Imagination Technologies (PowerVR G6430) all returns acceptable result for the proposed perceptual analysis. A comparison of performance on a Full HD frame (1920 px $\times$ 1080 px) and 8 dominant hues assumed between implementations on CPU (Intel Core i7) and different GPUs

are listed in Table 4.1. The results also indicate that the more GPU cores can be assigned as vertex shaders and run in parallel during a certain period, the better performance it can achieve. This is because the operations are just a few scalar value comparisons and assignments, which do not require powerful computational unit. Even the Intel HD Graphics 4000 GPU has higher computational power but does not out-perform the inferior Imagination Technologies PowerVR G6430 GPU since Intel HD Graphics 4000 GPU only have 16 execution units at 1300-MHz while the PowerVR G6430 GPU has 64 inferior ones at 450-MHz. Besides, the number of texture mapping unit (TMU) which used for texture fetching, the bandwidth of memory, and cache and data path designs are also relative to the histogram performance.

Table 4.1: Time Consumption of Different Histogram Implementations (ms)

Processor	Core i7	G6430	GK20A	HD G4000	GK107
Frequency	2.7-GHz	450-MHz	852-MHz	1.3-GHz	900-MHz
Cores	4 (1)	64	192	16	384
FP32 Op/s	90G (22.5G)	115G	364G	332G	691G
TMUs	-	8	8	8	32
Hue	170.48	2.37	1.32	3.34	0.67
Saturation	236.28	2.40	1.37	3.52	0.82
Lightness	272.41	2.44	1.38	3.52	0.82
Total	679.17	7.21	4.07	10.38	2.31

The 'FP32 Op/s' property is the number of single precision (32-bit) float-point operations a processor can perform per second, which are given in approximations according to processor specifications. For CPU based histogram computation, only one CPU core is utilized for implementation simplicity because it will not be used in the framework implementation.

4.4 Efficient Spatial Filtering

Spatial filtering on pixels is generally considered time-consuming because its intensive data access within an affinity. Usually the pixel fetches inside the filter window is done with loops, which worsen the situation because run-time data access is not optimized by GPU. On the other hand, even access to cached data may not be cost-efficient because near-by pixels share almost the same affinity with each other while simultaneous access to cache is implementation dependent. There are three sub-processes in the proposed framework that need spatial filtering – source de-noising and chroma up-sampling, frame gradient field calculation, and joint bilateral filtering for depth map refinement. To achieve real-time processing, the reduction of run-time data access is needed.

4.4.1 Optimized Gradient Calculation

In the implementation, the Sobel operator is used to calculate approximations of the derivatives. By defining $\mathbf{L}$ as the lightness channel of the frame, and $\mathbf{G}_x$ and $\mathbf{G}_y$ as the approximations of the horizontal and vertical derivates respectively, the calculations are as follows:

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{L} \tag{4.1}$$

and

$$\mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & +1 \end{bmatrix} * \mathbf{L} \tag{4.2}$$

The Sobel kernels are also separable, they can be decomposed as the products of averaging and a differentiation kernel:

$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & +1 \end{bmatrix} \tag{4.3}$$

and

$$\begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & +1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} +1 & +2 & +1 \end{bmatrix} \tag{4.4}$$

Thus, by applying one rendering pass per direction with packing output data, only 6 texture accesses are need instead of 9 texture accesses in traditional 2D convolution using single rendering pass. It can be further reduced to 5 by using frame-buffer read-back extension in OpenGL ES [100][101]. In the meantime, the horizontal and vertical pixel coordinates are preprocessed in the vertex shader and thus the cache of texture in the affinity is optimized by the GPU pipeline providing a lower latency. The final magnitude of gradient can be formulated as:

$$\mathbf{G}_{mag} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2} \tag{4.5}$$

4.4.2 Separated Joint Bilateral Filtering

Unlike Sobel operator, the joint bilateral filtering is a run-time configured kernel that is not separable. This results in intensive data access during filtering, which is inefficient even with preprocessed vertex. Though there are some real-time implementation such as the Real-Time $O(1)$ Bilateral Filtering [102], they usually based on the assumption that constant time spatial filtering is implementable, which is hard to be efficiently implemented on GPU. Though they achieved real-time processing, huge resource are consumed, making it unsuitable for the proposed framework.

Pham and Vliet proposed the 'separable' bilateral filtering [103], which uses one dimensional bilateral filter along each dimension and shows comparable results to the original method, and better results with larger radius at lower cost. Though the 'separable' is used in [103], it is rather 'separated' because bilateral filter is not separable. Considering the purpose of using joint bilateral filtering in the proposed framework is relatively simple – to constrain depth boundary, to suppress noise, and to smooth depth gradient, the separated bilateral filtering is deployed with a larger radius of 9. Let $\underset{\text{PASS}(i)}{\tilde{\mathbf{Z}}}$ denote the refined depth map in each pass, the pixel-wise refinement can be obtained as:

$$\underset{\text{PASS}(i)}{\tilde{\mathbf{Z}}_p} = \begin{cases} \dfrac{\sum_{q\in\Omega_p} \underset{\text{PASS}(i-1)}{\tilde{\mathbf{Z}}_q} h(p,q)}{\sum_{q\in\Omega_p} h(p,q)}, & \hat{\mathbf{D}}_p \neq 0 \\ 0, & \hat{\mathbf{D}}_p = 0 \end{cases} \tag{4.6}$$

where $i \in [1,2]$ and $\underset{\text{PASS}(0)}{\tilde{\mathbf{Z}}} = \mathbf{Z}$, and Ω_p is limited in one dimension along the same row or column of pixel p, and $h(p,q) = f(\|p-q\|)g(\|\hat{\mathbf{D}}_p - \hat{\mathbf{D}}_q\|)$.

Chapter 5

Experimental Results and Evaluations

The framework implementation has been adapted to three different platforms as listed in Table 5.1. All of them shares the same algorithm on mathematical operations. The results are inspiring and show the advantages of the proposed framework.

Table 5.1: Platforms Used for the Proposed Implementations

#	**Platform**	**CPU Arch.**	**GPU Model**	**Video Acquisition**	**Zero-Copy**
I	Apple iPad ME279C/A	ARMv8	G6430	On-Device Camera	Yes
II	NVIDIA Dev. Board PM375	ARMv7	GK20A	External	Yes
III	Personal Computer	x86-64	GK107	External	No

For GPU specifications, please refer to Table 4.1.

5.1 Experimental Results

5.1.1 2D Chroma Keying

A test scene has been set up for demonstrating experimental results of the 2D chroma keying part. The test scene contains uneven lit background (lit under traditional fluorescent tubes in the laboratory), large transparent areas of the bottle and the cup, sheen on the camera body and the mug, and foreground colours of the label and the helmet that are close to background in some colour spaces, as shown in Figure 5.1.

Figure 5.1: Photo of test scene for 2D chroma keying experiments.

Figure 5.2 shows the experimental result of live chroma keying on the platform I with the implementation of proposed framework running on an Apple iPad. The

video frames are captured via the device camera in Full HD progressive format (1920 px × 1080 px) at 25 or 30 frame-per-second (depends on lighting condition). Because the on-device camera is not stereoscopic, only 2D chroma keying part is implemented on the iPad.

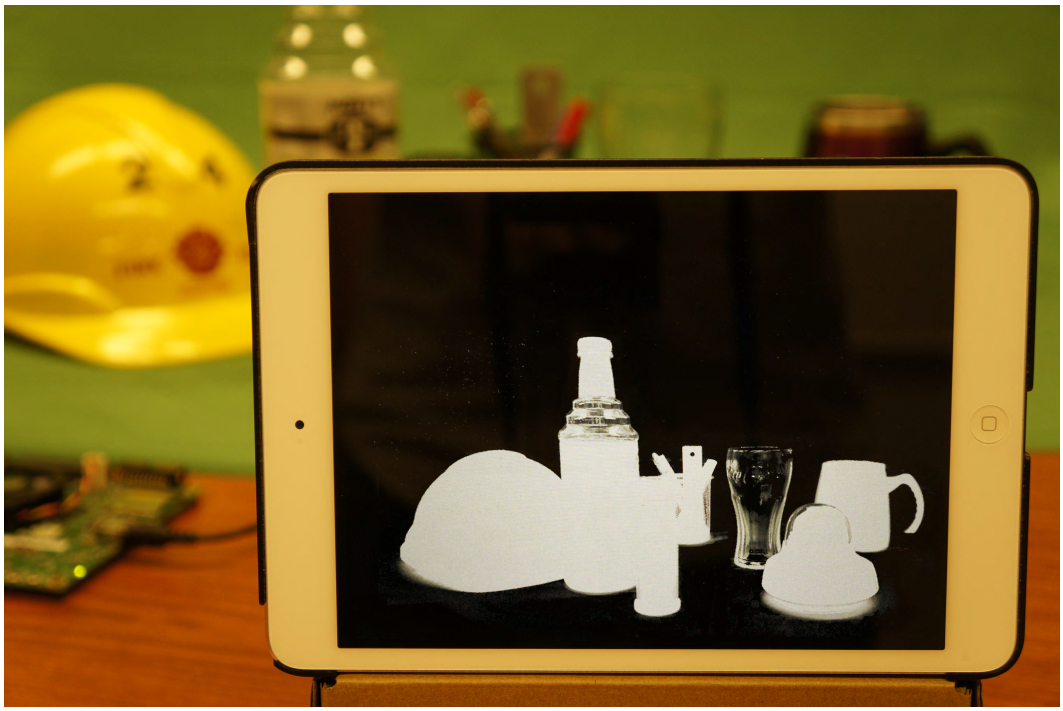


Figure 5.2: Photo of performing experiment on an Apple iPad, experimental result is shown on the iPad screen.

Figure 5.3 shows the experimental result of live chroma keying on the platform II with the implementation of proposed framework running on an NVIDIA PM375 Development Board. The video frames are captured via an external camera in Full HD progressive format (1920 px × 1080 px) at 60 frame-per-second. Though the stereoscopic part has been implemented, no stereoscopic camera is supplied at the time of demonstration. Therefore, only the result of 2D chroma keying part is shown.

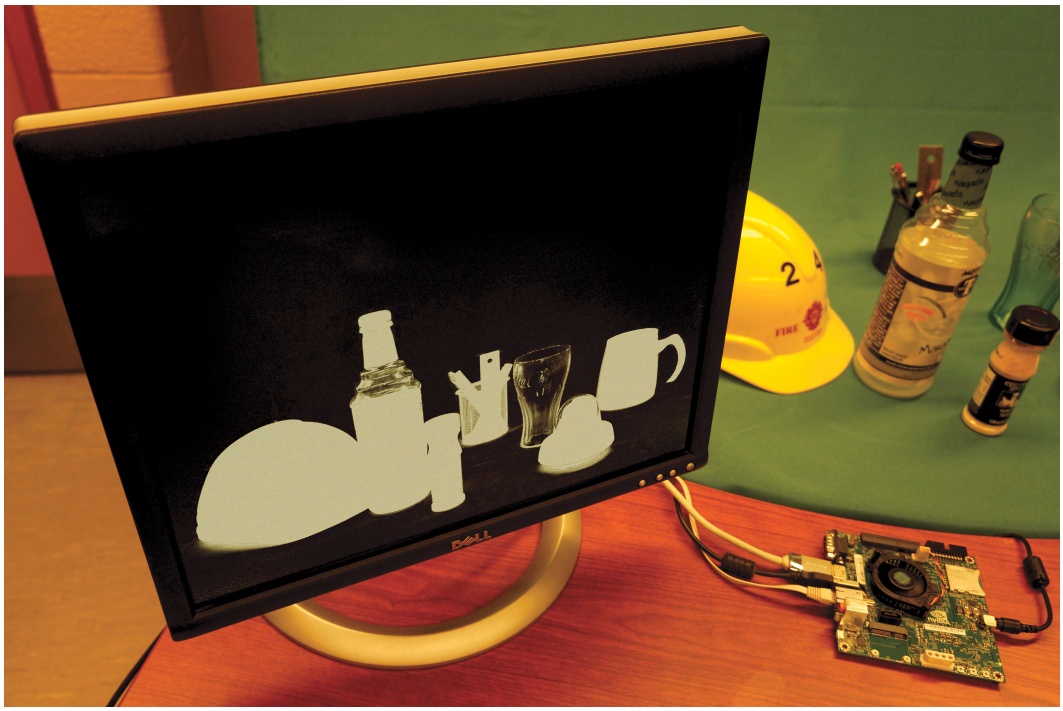


Figure 5.3: Photo of performing experiment on a PM375 development board, experimental result is shown on the discrete screen.

Because the same algorithm is used across different platforms, and the platform III with the implementation of proposed framework running on a personal computer shares the same feed of video frames with the platform II. Thus platform II and III generate the exactly same result on the 2D chroma keying part. This result will be further evaluated in the section 5.2.

5.1.2 3D Chroma Keying

The stereoscopic view based stereo matching algorithm cannot handle transparency well. On the other hand, the television industries use depth information for interactions with augmented reality which deals with human body that do not contain

transparency. Therefore, another test scene with opaque foreground has been set up for demonstrating experimental results of the 3D chroma keying part. Limited by the conditions, scene is shot by shifting the camera along the baseline with a distance of about 70 mm, and dolls are used instead of real humans, as shown in Figure 5.4.

Figure 5.4: Photo of test scene for 3D chroma keying experiments.

The stereoscopic frames are manually imported in to the system, and the alpha maps of both views are generated simultaneously on platform II and III, as shown in Figure 5.5.

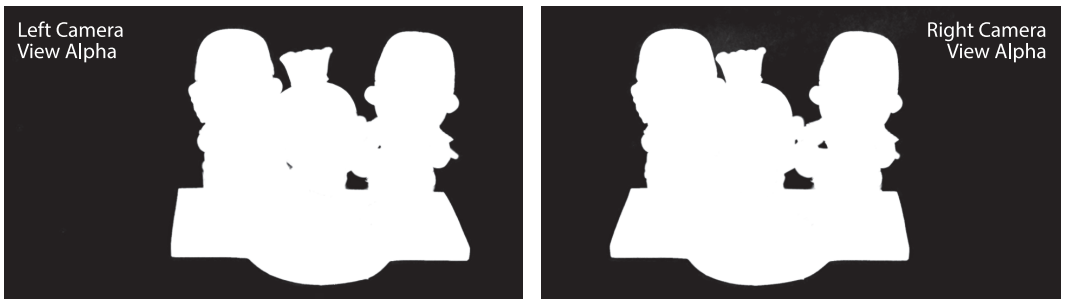


Figure 5.5: Stereoscopic alpha map pair of the 3D chroma keying test scene.

The stereo matching is also performed on the (Q_p, L_p, G_{mag}) feature-triplet representation, the final depth map is shown in Figure 5.6. The quality is sufficient for depth of filed simulation, as demonstrated in Figure 5.7.

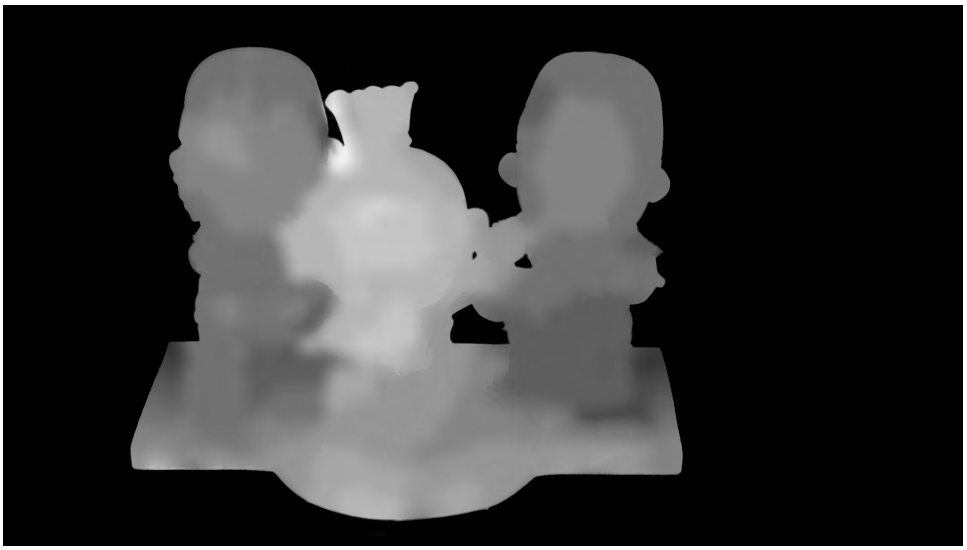


Figure 5.6: Final stereoscopic depth map of the 3D chroma keying test scene. The refined depth map has good foreground boundary and is smooth in depth transition.

Figure 5.7: Demonstration of depth of field rendering using experimental results. The composite is rendered at different focal lengthes.

5.1.3 System Performance

Table 5.2 shows the performance of the implementations on different platforms. The results show that the framework design and the optimized implementations fit well of its application.

Table 5.2: Performance Assessments of Proposed Implementations

#	**Input Format**	**Frame Processing Time (ms)**	**Standard Capability**
I	1920 × 1080 (2D)	30	1080P30
II	1920 × 1080 (2D)	15	1080P60
II	3840 × 2160 (2D)	46	-
II	1920 × 1080 (Half 3D SBS)	41	Half 3D SBS 1080P24
II	1920 × 1080 (Full 3D SBS)	56	-
III	1920 × 1080 (2D)	9	1080P60
III	3840 × 2160 (2D)	29	2160P30
III	1920 × 1080 (Half 3D SBS)	27	Half 3D SBS 1080P30
III	1920 × 1080 (Full 3D SBS)	39	Full 3D SBS 1080P24

'SBS' means Side-By-Side. Values are given in their worst (simulated with 100% uncertain pixels with 8 assumed dominant foreground colours, all pixels in the frame are processed with the alpha and foreground estimation as well as the stereo matching with a searching range of 1/2 frame width).

5.2 Evaluations and Comparisons on Chroma-Key Matting

Both qualitative and quantitative evaluations on the proposed implementation and other 8 state-of-the-art algorithms listed in Table 5.3, which cover source codes or executable binaries from recently published papers or industry-loved chroma-keyers in commercial software. All evaluations are performed on the platform III and the commercial software is running with their highest quality rendering setting. The proposed implementation, PhotoKey 6 Pro, and Primatte are set to automatic mode without any tuning, while other algorithms are more or less tuned to achieve better visual quality. For algorithms that need a trimap as user input, we provide our automatically generated quartered segmentation map with T_D set to T_U.

Table 5.3: Evaluated Algorithms and Environments

Algorithm	**Host Software**	**Min. User Input**
Proposed	Standalone	Auto
Shared Matting 1.0	Standalone [41]	Trimap
Robust Matting 1.45	Standalone [47]	Trimap
Photo Key 6 Pro	Standalone (FXHome)	Auto
Modular Keyer	Smoke (Autodesk) [104]	Colour Picker
UltraKey	Premiere Pro 8.0 (Adobe) [105]	Colour Picker
Keylight 1.2	After Effects 11.0 (Adobe) [106]	Colour Picker
Primatte	NukeX 8.0 (The Foundry) [107]	Auto
Ultimatte	NukeX 8.0 (The Foundry)	Colour Picker

5.2.1 Qualitative Evaluation

The test scene for 2D chroma keying experiments and several frame sequences from [3] are used for evaluation, as listed in Figure 5.8. They are not considered good enough for production and thus become challenges to chroma keying such as large transparency, heavy background colour spill, chroma sub-sampling, and compression. All frames are chroma sub-sampled at half the sample rate of luma ($Y' : C_b : C_r = 4 : 2 : 2$) in 8-bit percision due to the limitation of video acquisition device. Due to the fact that some of the algorithms only provide premultiplied foreground as $\alpha\mathbf{F}$, all foreground results are composited onto a red-yellow checkerboard background which consists the opponent colours of two widely used colours for chroma keying – green and blue.

Figure 5.9 shows the alpha maps and the composited results of the test scene frame, issues including uneven lit background, large area of semi-transparency and hardly visible edges of the bottle, glass, and camera cover, small area of background colour spill on the mug and bottles, and the vivid helmet. The results are expected to have clean background and correct transparency values on different objects.

Figure 5.10 shows the alpha maps and the composited results of the 1^{st} frame of the Godiva Medium sequence, issues including large area of seemingly translucency of the shawl, small area of background colour spill on the dress, and seemingly invisible fine hair. The results are expected to have opaque body and dress with smooth transition in seemingly translucent areas. Besides, though the fine hairs have very low alpha values, they should be visible in final composite.

Figure 5.11 shows the alpha maps and the composited results of the 21^{st} frame of the Godiva Wide sequence, issues including uneven lit background, and large area of background colour spills on the wall which appears in the similar colours as the

rotating fans. The results are expected to have clean background, opaque wall and surroundings, and seemingly transparent rotating fan blades.

Figure 5.12 shows the alpha maps and the composited results of the 3rd frame of the Motion Blur sequence. The main issue is motion blur. The results are expected to have smooth transition on the blur with reasonable foreground colour restoration.

Figure 5.13 shows the alpha maps and the composited results of the 249th frame of the Smoke sequence, issues including large area of seemingly translucency achromatic smoke, and large area of background colour spills on the wall and the ladder. The results are expected to clean background, white smoke cloud, smooth transition on the edges of the smoke cloud, opaque wall, and opaque ladder body.

Figure 5.14 shows the alpha maps and the composited results of the 44th frame of the Sheen sequence, issues including large area of background colour spills on the arms and the clothes. The results are expected to have opaque foreground, while the sheen should be removed and the foreground colours should be correctly recovered.

Figure 5.15 shows the alpha maps and the composited results of the Tinted Glasses single frame, issues including background colour spills on the hand and spectacles, as well as the semi-transparent glasses that have similar colours to the sheen. The results are expected to have opaque hand and spectacles. The sheen should be removed while the transparency of the glasses should be kept.

Figure 5.16 shows the alpha maps and the composited results of the Jeep Side Roto single frame, issues including uneven lit background, large area of background colour spills on the car seats, and semi-transparent windows with ambient reflections. Results are expected to have clean background, opaque car seats, and reasonable transparency on the windows.

Test Scene 2D

Godiva Medium 001

Godiva Wide 021

Motion Blur 003

Smoke 249

Sheen 055

Tinted Glass

Jeep Side Roto

Figure 5.8: Images and video frames used in the qualitative evaluation.

Figure 5.9: Alpha map & foreground comparison of the test scene. Because of the ill lighting condition, the shadows have larger chrominance shift. Shadows are more or less kept in all results while the proposed one still has a better overall appearance.

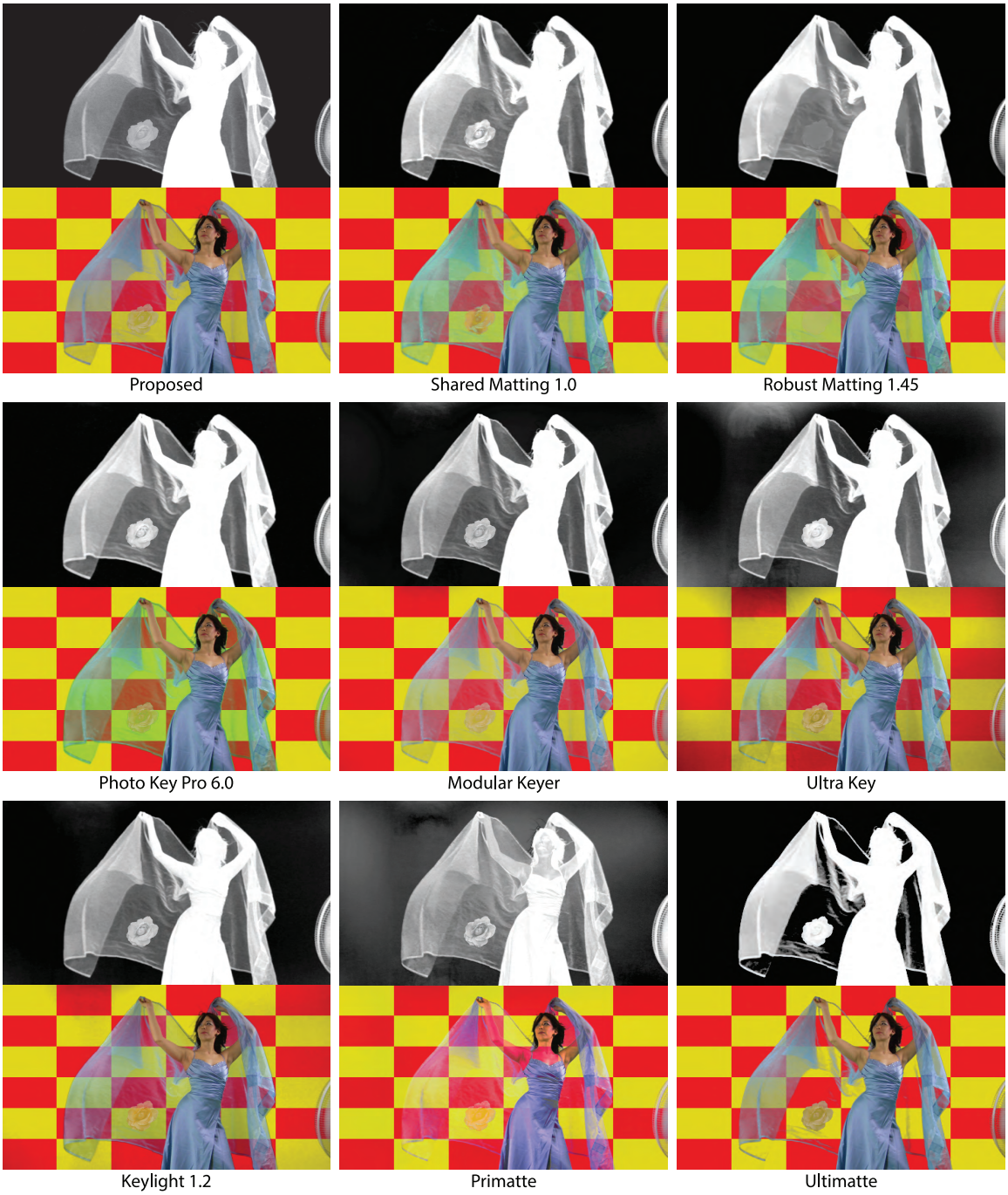


Figure 5.10: Alpha map & foreground comparison of the 1st frame of the Godiva Medium sequence. The proposed one efficiently removed the background shadows while kept the fine hairs.

Figure 5.11: Alpha map & foreground comparison of the 21st frame of the Godiva Wide sequence. It is hard to remove all sheen on the achromatic wall because its sheen only varies in saturation and lightness as the shadows. Keeping the wall opaque makes the alpha values of the shawl in the proposed one slightly higher than others.

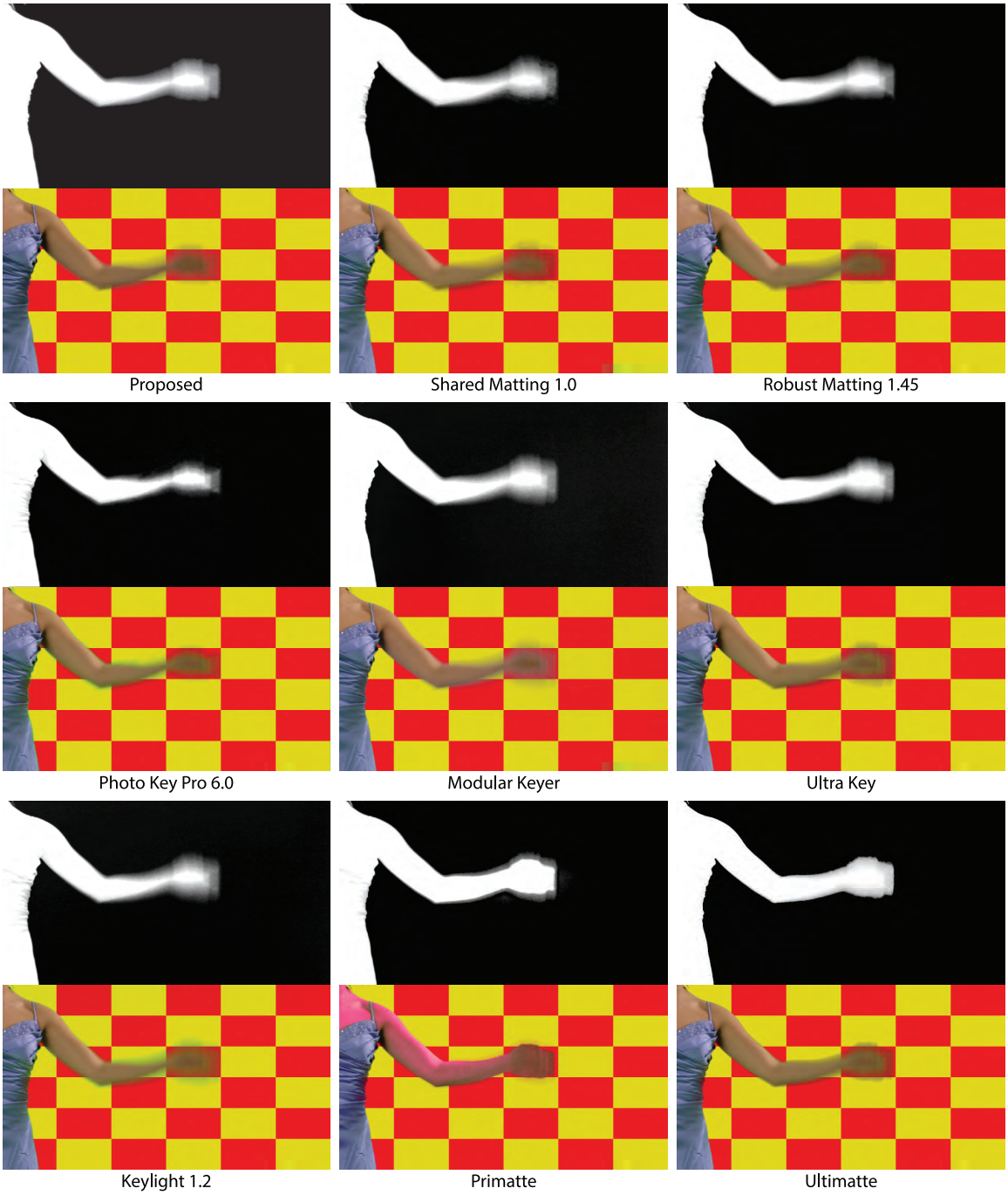


Figure 5.12: Alpha map & foreground comparison of the 3rd frame of the Motion Blur sequence. The proposed one has opaque body with reasonable foreground colour restoration, while the transition on the blur is still above average.

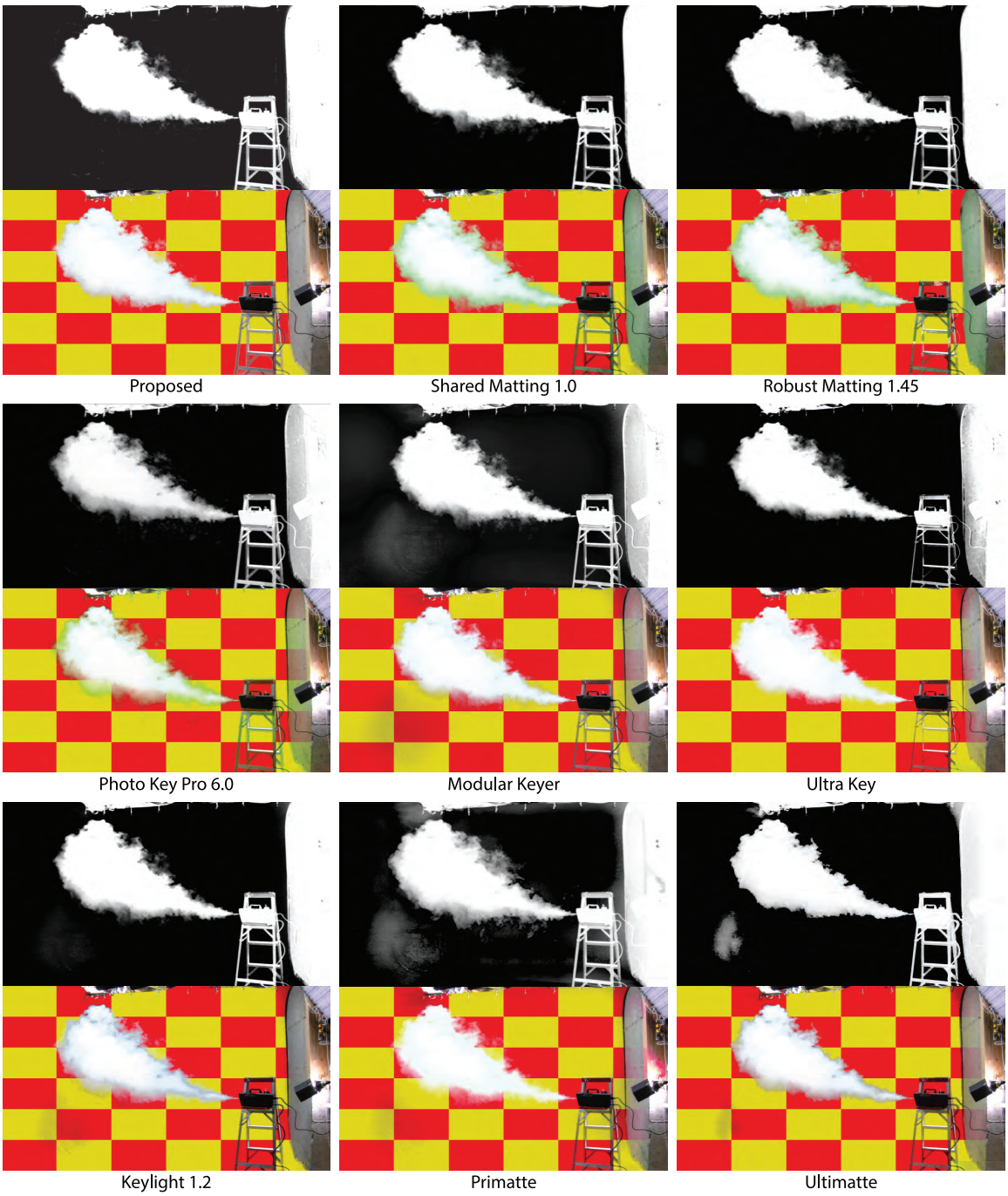


Figure 5.13: Alpha map & foreground comparison of the 249$^{\text{th}}$ frame of the Smoke sequence. Still, it is hard to distinguish achromatic smoke, ladder body, and wall. The proposed one is not the best in all aspects but the overall appearance is still very good.

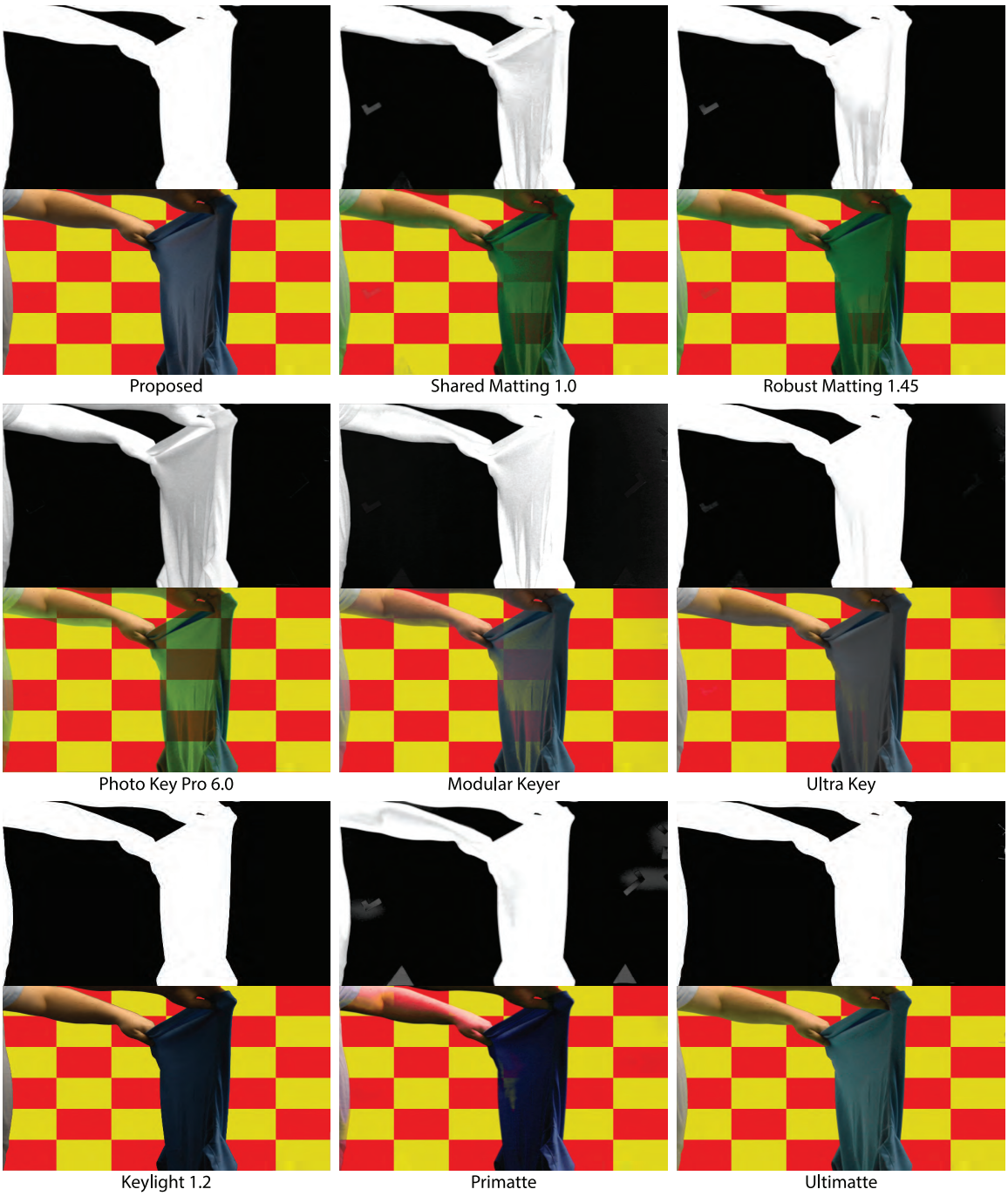


Figure 5.14: Alpha map & foreground comparison of the 44$^{\text{th}}$ frame of the Sheen sequence. In this turn, some traditional chroma-keyers have good results on alpha maps, but the proposed one still have better foreground restoration. It is apparent that there are unwanted colour fringes along the objects' boundary, which is also observed by its publisher [108]. Some software shrinks the matte so that the fringes do not appear in the final result.

Figure 5.15: Alpha map & foreground comparison of the Tinted Glasses frame. Again, the proposed one has a better overall appearance though the Modular Keyer has a little better transition on the glasses.

Figure 5.16: Alpha map & foreground comparison of the Jeep Side Rote frame. The proposed one shows its ability on keep fine details such as the fraction effect on the windshield, while the sheen on the car seats is almost removed.

5.2.2 Quantitative Evaluation

A quantitative evaluation has also been conducted on computer generated composites, by using the ground-truth foreground images and alpha maps from the Online Benchmark for Image Matting datasets [109]. The computer generated composites are shown on the top of Figure 5.17, ground-truth alpha maps are shown in both Figures 5.17 and 5.18. Each of them consists of 6.912×10^5 pixels. To reflect the key challenges in chroma keying problem, the evaluations are presented on four typical images that feature fuzzy hairs, overlapped colour distributions, ambient diffusions, and uneven lit background. Due to the fact that precisions varied in different implementations, all alpha values are normalized by scaling between 0 and 1. The numerical results are listed in Table 5.4.

Table 5.4: Quantitative Evaluation of Alpha Mattes in terms of SAD(10^3) and MSE(10^{-3})

	GT04		**GT08**		**GT13**		**GT25**	
Algorithm	SAD	MSE	SAD	MSE	SAD	MSE	SAD	MSE
Proposed	**10.70**	**1.55**	**5.93**	**0.63**	**2.38**	**0.22**	2.79	0.49
Shared Matting 1.45	11.05	1.98	9.21	1.63	4.31	0.66	4.69	1.33
Robust Matting 1.0	12.19	2.67	28.20	17.58	45.41	44.07	13.66	13.78
Photo Key 6 Pro	13.64	2.12	9.91	1.61	36.26	7.19	**2.50**	**0.33**
Modular Keyer	17.81	2.37	21.82	3.22	19.32	2.34	31.84	3.70
UltraKey	22.07	4.49	24.38	8.56	16.83	2.13	23.63	3.66
Keylight 1.2	20.94	3.56	10.37	1.62	61.12	13.3	6.59	0.74
Primatte	22.13	6.59	18.57	6.23	12.49	1.74	8.65	3.26
Ultimatte	25.81	10.26	19.85	7.11	7.64	2.63	3.78	1.17

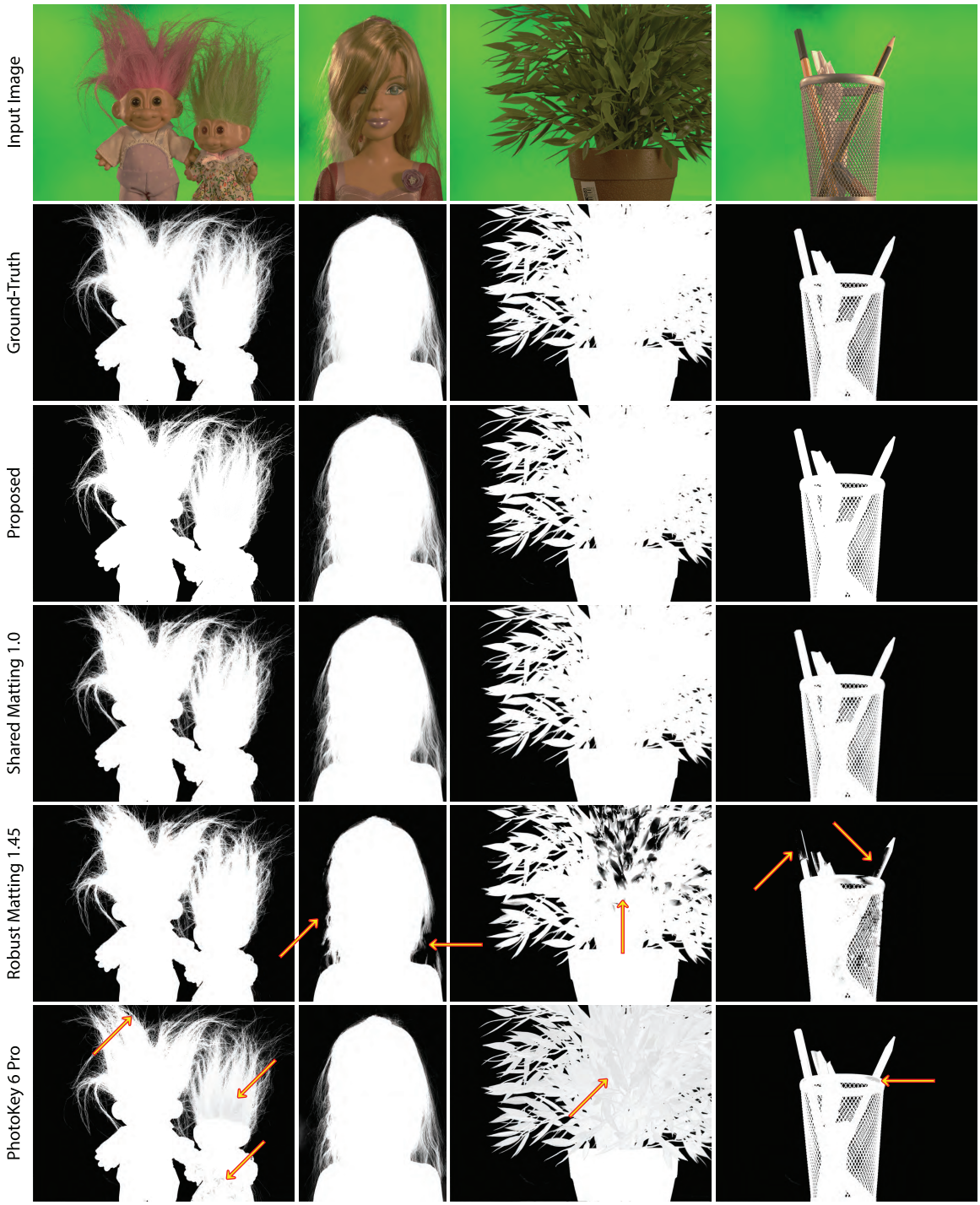


Figure 5.17: Alpha map comparison of quantitative evaluation. From left to right: GT04, GT08, GT13, and GT25. Arrows indicate obvious local flaws.

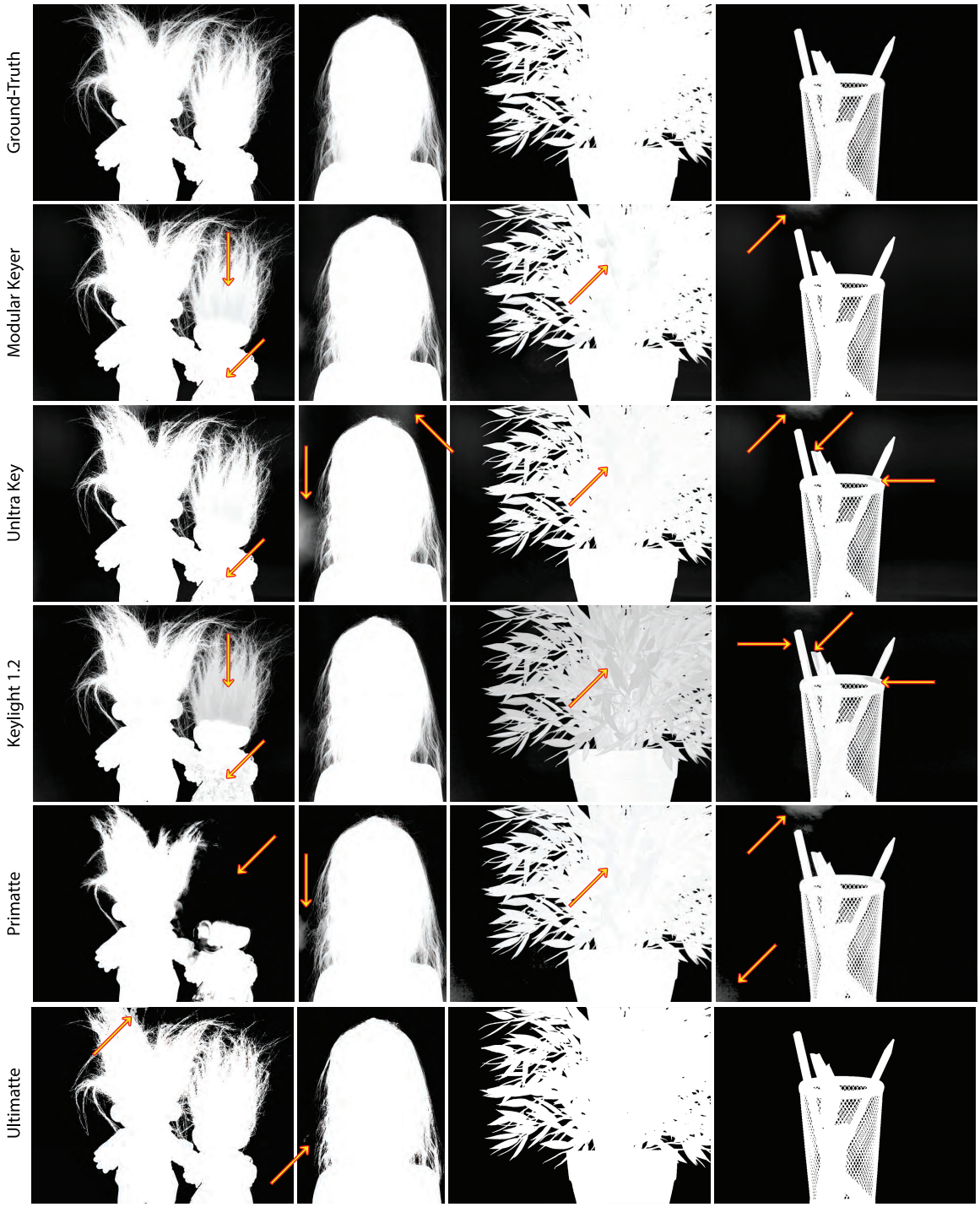


Figure 5.18: Alpha map comparison of quantitative evaluation (continued). From left to right: GT04, GT08, GT13, and GT25. Arrows indicate obvious local flaws.

It is not surprised that for solid objects as the ones in GT25, some commercial chroma-keyers like PhotoKey may generate alpha mattes with slightly lower SAD and MSE than the proposed implementations. This is because of the edge optimizations such as edge softening and matte chocking used in those commercial chroma-keyers. In general, the overall performance of the proposed implementations is much better than others across all data sets in terms of both alpha map accuracy and foreground colour rationality.

5.3 Discussion

During the evaluations, some of the evaluated algorithms have better results when feeding linear RGB inputs. Most commercial chroma-keyers can automatically convert gamma encoded formats into linear formats according to the colour profiles of the input or manual settings. In the proposed implementations, the procedures for colour space conversions are strictly followed according to their specifications. However, there are some slight colour differences between the results from different algorithms can be observed. This may be because of the spill suppression algorithm used in some commercial chroma-keyers. On the other hand, the correctly converted colours from non-computer generated sources may fail in alpha blending because the alpha blending does not really simulates colour filtering effects. In addition, the input dynamic range, noise level, colour subsampling, interpolation algorithm, and quantization error in the camera may also account for the failures in alpha blending.

Chapter 6

Conclusion and Future Work

In this thesis, an automatic high performance stereoscopic 3D chroma-key matting framework has been proposed. Based on the simulation of human perception of colours, the generated quartered segmentation map can make the subsequent processes more efficient because of its smaller unknown regions and richer details. The single-pass background inpainting technique significantly reduces the computational cost, and provides smoother and cleaner background plate in less time. The invalid foreground-background colour combinations during the prediction process are also safely eliminated based on colour mixing psychology. The attempt of using normalized cross-correlation based stereo matching also gives the depth map in expected object level precision.

The non-iterative and pixel-independent nature makes the proposed framework efficient in parallel computing environment. Three GPU-based prototypes on different platforms have also been implemented. The qualitative and quantitative evaluations show that the prototypes with low-end hardware are able to process High-Definition

and/or Ultra High-Definition chroma keying in real-time with better composites and overall more precise alpha maps.

There are some limitations in the proposed framework. The foreground colour and alpha prediction is not very reliable in low saturation regions in the input image or video frame. This is partially because the CIECAM02 colour appearance model used does not have very good performance in low saturation colours, and another reason is that low saturation colours are very sensitive to camera sensor noise. Besides, the normalized cross-correlation based stereo matching is not robust enough for textureless and translucent objects. In addition, though the proposed framework may avoid determining reflections as transparencies, but it does not recover the reflections automatically. This is because a rational mapping of ambient lighting cannot be directly reconstructed from the new background.

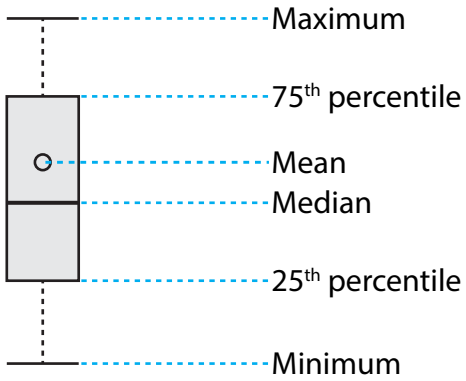
Future work includes improvements in prediction and self-evaluate metrics by taking spatial information into account. The relationship between human perception and different colour spaces will still be investigated, new colour models may be applied. For the reflection reconstruction, a rational mapping of ambient lighting may be reconstructed from the new background. Another of the possible extensions is utilizing the CIECAM02 colour appearance model for post-processing on blending different images or video frames to make the final composite more realistic and coherent.

APPENDICES

Appendix A

Brief Intro to Box Whisker Plot

Box whisker plot is a convenient way of graphically depicting groups of numerical data through their quartiles. It displays variation in samples of a statistical population without making any assumption of the underlying statistical distribution. The plots shown in this thesis uses the following representation:



Appendix B

CIE Colour Appearance Model: CIECAM02 [71]

Part 1: The Forward Mode

Input: X, Y, Z (under test illuminant $X_{\mathrm{w}}, Y_{\mathrm{w}}, Z_{\mathrm{w}}$)

Output: Correlates of lightness J, chroma C, hue composition H, hue angle h, colourfulness M, saturation s and brightness Q

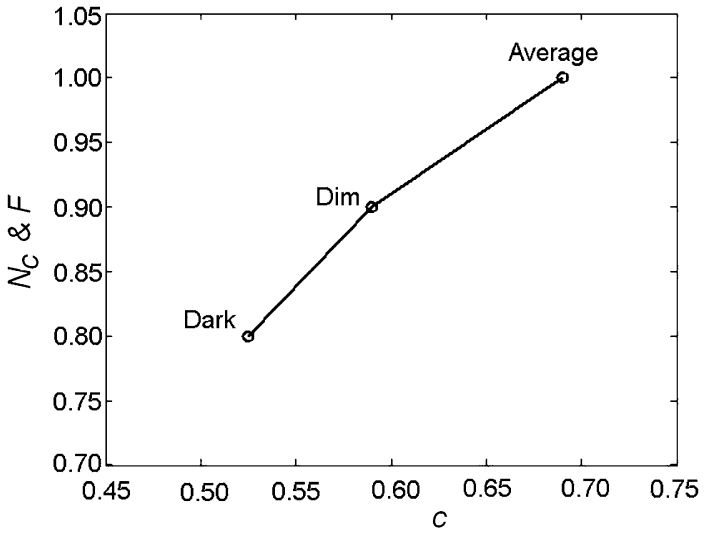
Illuminants, viewing surrounds set up and background parameters
(See the note at the end of this Appendix for determining all parameters)

Adopted white in test illuminant: $X_{\mathrm{w}}, Y_{\mathrm{w}}, Z_{\mathrm{w}}$
Background in test conditions: Y_{b}
(Reference white in reference illuminant: $X_{\mathrm{wr}} = Y_{\mathrm{wr}} = Z_{\mathrm{wr}} = 100$, which are fixed in the model)
Luminance of test-adapting field (cd/m^2) : L_{A}
All surround parameters are given in Table B1
Note that for determining the surround conditions, see the note at the end of this Appendix. N_c and F are modelled as a function of c, and can be linearly interpolated as shown in the Figure B1, using the above points

Table B1 Surround parameters

	F	c	N_c
Average	1.0	0.69	1.0
Dim	0.9	0.59	0.9
Dark	0.8	0.535	0.8

Figure B1 N_c and F varies with c



Step 0: Calculate all values/parameters which are independent of input samples

$$\begin{pmatrix} R_{\mathrm{w}} \\ G_{\mathrm{w}} \\ B_{\mathrm{w}} \end{pmatrix} = M_{\mathrm{CAT02}} \cdot \begin{pmatrix} X_{\mathrm{w}} \\ Y_{\mathrm{w}} \\ Z_{\mathrm{w}} \end{pmatrix}, \quad D = F \cdot \left[1 - \left(\frac{1}{3.6} \right) \cdot \mathrm{e}^{\left(\frac{-L_{\mathrm{A}}-42}{92} \right)} \right].$$

Note if D is greater than one or less than zero, set it to one or zero, respectively.

$$D_{\mathrm{R}} = D \cdot \frac{Y_{\mathrm{w}}}{R_{\mathrm{w}}} + 1 - D, \quad D_{\mathrm{G}} = D \cdot \frac{Y_{\mathrm{w}}}{G_{\mathrm{w}}} + 1 - D, \quad D_{\mathrm{B}} = D \cdot \frac{Y_{\mathrm{w}}}{B_{\mathrm{w}}} + 1 - D,$$

$$F_L = 0.2\, k^4 \cdot (5L_A) + 0.1(1-k^4)^2 \cdot (5L_A)^{1/3}, \quad \text{where } k = \frac{1}{5 \cdot L_{\mathrm{A}} + 1}.$$

$$n = \frac{Y_{\mathrm{b}}}{Y_{\mathrm{w}}}, \quad z = 1.48 + \sqrt{n}, \quad N_{\mathrm{bb}} = 0.725 \cdot \left(\frac{1}{n}\right)^{0.2}, \quad N_{\mathrm{cb}} = N_{\mathrm{bb}},$$

$$\begin{pmatrix} R_{\mathrm{wc}} \\ G_{\mathrm{wc}} \\ B_{\mathrm{wc}} \end{pmatrix} = \begin{pmatrix} D_{\mathrm{R}} \cdot R_{\mathrm{w}} \\ D_{\mathrm{G}} \cdot G_{\mathrm{w}} \\ D_{\mathrm{B}} \cdot B_{\mathrm{w}} \end{pmatrix}, \quad \begin{pmatrix} R'_{\mathrm{w}} \\ G'_{\mathrm{w}} \\ B'_{\mathrm{w}} \end{pmatrix} = M_{\mathrm{HPE}} \cdot M^{-1}_{\mathrm{CAT02}} \cdot \begin{pmatrix} R_{\mathrm{wc}} \\ G_{\mathrm{wc}} \\ B_{\mathrm{wc}} \end{pmatrix},$$

$$M_{\mathrm{CAT02}} = \begin{pmatrix} 0.7328 & 0.4296 & -0.1624 \\ -0.7036 & 1.6975 & 0.0061 \\ 0.0030 & 0.0136 & 0.9834 \end{pmatrix},$$

$$M_{\mathrm{HPE}} = \begin{pmatrix} 0.38971 & 0.68898 & -0.07868 \\ -0.22981 & 1.18340 & 0.04641 \\ 0.00000 & 0.00000 & 1.00000 \end{pmatrix},$$

$$R'_{\mathrm{aw}} = 400 \cdot \left(\frac{\left(\frac{F_{\mathrm{L}} \cdot R'_{\mathrm{w}}}{100}\right)^{0.42}}{\left(\frac{F_{\mathrm{L}} \cdot R'_{\mathrm{w}}}{100}\right)^{0.42} + 27.13} \right) + 0.1,$$

$$G'_{\mathrm{aw}} = 400 \cdot \left(\frac{\left(\frac{F_{\mathrm{L}} \cdot G'_{\mathrm{w}}}{100}\right)^{0.42}}{\left(\frac{F_{\mathrm{L}} \cdot G'_{\mathrm{w}}}{100}\right)^{0.42} + 27.13} \right) + 0.1,$$

$$B'_{\mathrm{aw}} = 400 \cdot \left(\frac{\left(\frac{F_{\mathrm{L}} \cdot B'_{\mathrm{w}}}{100}\right)^{0.42}}{\left(\frac{F_{\mathrm{L}} \cdot B'_{\mathrm{w}}}{100}\right)^{0.42} + 27.13} \right) + 0.1,$$

$$A_{\mathrm{w}} = \left[2 \cdot R'_{\mathrm{aw}} + G'_{\mathrm{aw}} + \frac{B'_{\mathrm{aw}}}{20} - 0.305 \right] \cdot N_{\mathrm{bb}}.$$

Note that all parameters computed in this step are needed for the following calculations. However, they depend only on surround and viewing conditions; hence, when processing pixels of image, they are computed once for all. The following computing steps are sample dependent.

Step 1: Calculate (sharpened) cone responses (transfer colour-matching functions to sharper sensors)

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = M_{\mathrm{CAT02}} \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix},$$

Step 2: Calculate the corresponding (sharpened) cone response (considering various luminance level and surround conditions included in D; hence, in D_{R}, D_{G} and D_{B})

$$\begin{pmatrix} R_{\mathrm{c}} \\ G_{\mathrm{c}} \\ B_{\mathrm{c}} \end{pmatrix} = \begin{pmatrix} D_{\mathrm{R}} \cdot R \\ D_{\mathrm{G}} \cdot G \\ D_{\mathrm{B}} \cdot B \end{pmatrix},$$

Step 3: Calculate the Hunt-Pointer-Estevez response

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = M_{\mathrm{HPE}} \cdot M^{-1}_{\mathrm{CAT02}} \cdot \begin{pmatrix} R_{\mathrm{c}} \\ G_{\mathrm{c}} \\ B_{\mathrm{c}} \end{pmatrix},$$

Step 4: Calculate the post-adaptation cone response (resulting in dynamic range compression)

$$R'_{\mathrm{a}} = 400 \cdot \left(\frac{\left(\frac{F_L \cdot R'}{100} \right)^{0.42}}{\left(\frac{F_L \cdot R'}{100} \right)^{0.42} + 27.13} \right) + 0.1.$$

If R' is negative, then

$$R'_{\mathrm{a}} = -400 \cdot \left(\frac{\left(\frac{-F_L \cdot R'}{100} \right)^{0.42}}{\left(\frac{-F_L \cdot R'}{100} \right)^{0.42} + 27.13} \right) + 0.1$$

and similarly for the computations of G'_{a}, and B'_{a}, respectively.

Step 5: Calculate Redness–Greenness (a) , Yellowness–Blueness (b) components and hue angle (h):

$$a = R'_{\mathrm{a}} - \frac{12 \cdot G'_{\mathrm{a}}}{11} + \frac{B'_{\mathrm{a}}}{11},$$
$$b = \frac{(R'_{\mathrm{a}} + G'_{\mathrm{a}} - 2 \cdot B'_{\mathrm{a}})}{9},$$
$$h = \tan^{-1}\left(\frac{b}{a}\right)$$

make sure h between 0 and 360^{o}.

Step 6: Calculate eccentricity (e_t) and hue composition (H), using the unique hue data given in Table B2; set $h' = h + 360$ if $h < h_1$, otherwise $h' = h$. Choose a proper i(i =1,2,3 or 4) so that $h_i \leq h' < h_{i+1}$. Calculate

$$e_t = \frac{1}{4} \cdot \left[\cos\left(\frac{h' \cdot \pi}{180} + 2\right) + 3.8\right],$$

which is close to, but not exactly the same as, the eccentricity factor given in Table B2.

$$H = H_i + \frac{100 \cdot \frac{h' - h_i}{e_i}}{\frac{h' - h_i}{e_i} + \frac{h_{i+1} - h'}{e_{i+1}}}.$$

Table B2 Unique hue data for calculation of hue quadrature

	Red	Yellow	Green	Blue	Red
i	1	2	3	4	5
h_i	20.14	90.00	164.25	237.53	380.14
e_i	0.8	0.7	1.0	1.2	0.8
H_i	0.0	100.0	200.0	300.0	400.0

Step 7: Calculate achromatic response A

$$A = \left[2 \cdot R'_{\mathrm{a}} + G'_{\mathrm{a}} + \frac{B'_{\mathrm{a}}}{20} - 0.305\right] \cdot N_{\mathrm{bb}}.$$

Step 8: Calculate the correlate of lightness

$$J = 100 \cdot \left(\frac{A}{A_{\mathrm{w}}}\right)^{c \cdot z}.$$

Step 9: Calculate the correlate of brightness

$$Q = \left(\frac{4}{c}\right) \cdot \left(\frac{J}{100}\right)^{0.5} \cdot (A_{\mathrm{w}} + 4) \cdot F_{\mathrm{L}}^{0.25}.$$

Step 10: Calculate the correlates of chroma (C), colourfulness (M) and saturation (s)

$$t = \frac{\left(\frac{50000}{13} \cdot N_{\mathrm{c}} \cdot N_{\mathrm{cb}}\right) \cdot e_t \cdot \left(a^2 + b^2\right)^{1/2}}{R'_{\mathrm{a}} + G'_{\mathrm{a}} + \left(\frac{21}{20}\right) \cdot B'_{\mathrm{a}}},$$

$$C = t^{0.9} \cdot \left(\frac{J}{100}\right)^{0.5} \cdot (1.64 - 0.29^{n})^{0.73},$$

$$M = C \cdot F_{\mathrm{L}}^{0.25},$$

$$s = 100 \cdot \left(\frac{M}{Q}\right)^{0.5}.$$

Part 2: The Reverse Mode

Input: J or Q; C, M or s; H or h
Output: X, Y, Z (under test illuminant $X_{\mathrm{w}}, Y_{\mathrm{w}}, Z_{\mathrm{w}}$)

Illuminants, viewing surrounds and background parameters are the same as those given in the forward mode. See notes at the end of this Appendix calculating/defining the luminance of the adapting field and surround conditions.

Step 0: Calculate viewing parameters
Compute all $F_{\mathrm{L}}, n, z, N_{\mathrm{bb}} = N_{\mathrm{bc}}, R_{\mathrm{w}}, G_{\mathrm{w}}, B_{\mathrm{w}}$, D, $D_{\mathrm{R}}, D_{\mathrm{G}}, D_{\mathrm{B}}$, $R_{\mathrm{wc}}, G_{\mathrm{wc}}, B_{\mathrm{wc}}$, $R'_{\mathrm{w}}, G'_{\mathrm{w}}, B'_{\mathrm{w}}$ $R'_{\mathrm{aw}}, G'_{\mathrm{aw}}, B'_{\mathrm{aw}}$ and A_{w} using the same formulae as in Step 0 of the Forward model. They are needed in the following steps. Note that all data computed in this step can be used for all samples (e.g., all pixels for an image) under the viewing conditions. Hence, they are computed once for all. The following computing steps are sample dependent.

Step 1: Obtain J, C and h from H, Q, M, s
The entering data can be in different combination of perceived correlates, i.e., J or Q; C, M, or s; and H or h. Hence, the followings are needed to convert the others to J, C, and h.

Step 1–1: Compute J from Q (if start from Q)

$$J = 6.25 \cdot \left[\frac{c \cdot Q}{(A_{\mathrm{w}} + 4) \cdot F_{\mathrm{L}}^{0.25}} \right]^2 .$$

Step 1–2: Calculate C from M or s

$$C = \frac{M}{F_{\mathrm{L}}^{0.25}} \text{ (if start from } M\text{)}$$

$$\mathrm{Q} = \left(\frac{4}{c} \right) \cdot \left(\frac{\mathrm{J}}{100} \right)^{0.5} \cdot (\mathrm{A_w}{+}4.0) \cdot F_L^{0.25}$$

and $C = \left(\frac{s}{100}\right)^2 \cdot \left(\frac{Q}{F_{\mathrm{L}}^{0.25}}\right)$ (if start from s)

Step 1–3: Calculate h from H (if start from H)
The correlate of hue (h) can be computed by using data in Table B2 in the Forward mode.
Choose a proper i ($i = 1,2,3$ *or* 4) so that $H_i \le H < H_{i+1}$.

$$h' = \frac{(H - H_i) \cdot (e_{i+1} h_i - e_i \cdot h_{i+1}) - 100 \cdot h_i \cdot e_{i+1}}{(H - H_i) \cdot (e_{i+1} - e_i) - 100 \cdot e_{i+1}}.$$

Set $h = h' - 360$ if $h' > 360$, otherwise $h = h'$.

Step 2: Calculate t, e_t, p_1, p_2 and p_3

$$t = \left(\frac{C}{\sqrt{\frac{J}{100}} \cdot (1.64 - 0.29^n)^{0.73}} \right)^{\frac{1}{0.9}},$$

$$e_t = \frac{1}{4} \cdot \left[\cos\left(h \cdot \frac{\pi}{180} + 2 \right) + 3.8 \right],$$

$$A = A_{\mathrm{w}} \cdot \left(\frac{J}{100} \right)^{\frac{1}{c \cdot z}},$$

$$p_1 = \left(\frac{50000}{13} \cdot N_c \cdot N_{\mathrm{cb}} \right) \cdot e_t \cdot \left(\frac{1}{t} \right), \text{ if t} \neq 0,$$

$$p_2 = \frac{A}{N_{\mathrm{bb}}} + 0.305,$$

$$p_3 = \frac{21}{20},$$

Step 3: Calculate a and b

If $t = 0$, then $a = b = 0$ and go to Step 4

(be sure transferring h from degree to radian before calculating $\sin(h)$ and $\cos(h)$)

If $|\sin(h)| \geq |\cos(h)|$, then

$$p_4 = \frac{p_1}{\sin(h)},$$

$$b = \frac{p_2 \cdot (2 + p_3) \cdot \left(\frac{460}{1403}\right)}{p_4 + (2 + p_3) \cdot \left(\frac{220}{1403}\right) \cdot \left(\frac{\cos(h)}{\sin(h)}\right) - \left(\frac{27}{1403}\right) + p_3 \cdot \left(\frac{6300}{1403}\right)},$$

$$a = b \cdot \left(\frac{\cos(h)}{\sin(h)} \right).$$

If $|\cos(h)| > |\sin(h)|$, then

$$p_5 = \frac{p_1}{\cos(h)},$$

$$a = \frac{p_2 \cdot (2+p_3) \cdot \left(\frac{460}{1403}\right)}{p_5 + (2+p_3) \cdot \left(\frac{220}{1403}\right) - \left[\left(\frac{27}{1403}\right) - p_3 \cdot \left(\frac{6300}{1403}\right)\right] \cdot \left(\frac{\sin(h)}{\cos(h)}\right)},$$

$$b = a \cdot \left(\frac{\sin(h)}{\cos(h)}\right).$$

Step 4: Calculate $R'_{\mathrm{a}}, G'_{\mathrm{a}}$ and B'_{a}

$$R'_{\mathrm{a}} = \frac{460}{1403} \cdot p_2 + \frac{451}{1403} \cdot a + \frac{288}{1403} \cdot b,$$

$$G'_{\mathrm{a}} = \frac{460}{1403} \cdot p_2 - \frac{891}{1403} \cdot a - \frac{261}{1403} \cdot b,$$

$$B'_{\mathrm{a}} = \frac{460}{1403} \cdot p_2 - \frac{220}{1403} \cdot a - \frac{6300}{1403} \cdot b.$$

Step 5: Calculate R', G' and B'

$$R' = \mathrm{sign}(R'_{\mathrm{a}} - 0.1) \cdot \frac{100}{F_L} \cdot \left[\frac{27.13 \cdot |R'_{\mathrm{a}} - 0.1|}{400 - |R'_{\mathrm{a}} - 0.1|}\right]^{\frac{1}{0.42}}.$$

Here, $\mathrm{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$, and similarly computing G', and B' from G'_{a}, and B'_{a}.

Step 6: Calculate $R_{\mathrm{C}}, G_{\mathrm{C}}$ and B_{C} (for the inverse matrix, see the note at the end of the Appendix)

$$\begin{pmatrix} R_{\mathrm{c}} \\ G_{\mathrm{c}} \\ B_{\mathrm{c}} \end{pmatrix} = M_{\mathrm{CAT02}} \cdot M_{\mathrm{HPE}}^{-1} \cdot \begin{pmatrix} R' \\ G' \\ B' \end{pmatrix}.$$

Step 7: Calculate R, G and B

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} \frac{R_c}{D_R} \\ \frac{G_c}{D_G} \\ \frac{B_c}{D_B} \end{pmatrix}.$$

Step 8: Calculate X, Y and Z (for the coefficients of the inverse matrix, see the note at the end of the Appendix)

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = M_{\mathrm{CAT02}}^{-1} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

Notes to Appendix B

1. It is recommended to use the matrix coefficients given below for the inverse matrix M_{CAT02}^{-1} and M_{HPE}^{-1}:

$$M_{\mathrm{CAT02}}^{-1} = \begin{pmatrix} 1.096124 & -0.278869 & 0.182745 \\ 0.454369 & 0.473533 & 0.072098 \\ -0.009628 & -0.005698 & 1.015326 \end{pmatrix},$$

$$M_{\mathrm{HPE}}^{-1} = \begin{pmatrix} 1.910197 & -1.112124 & 0.201908 \\ 0.370950 & 0.629054 & -0.000008 \\ 0.000000 & 0.000000 & 1.000000 \end{pmatrix}$$

2. For implementing the CIECAM02, the testing data and the corresponding results from the forward and reverse modes can be found from [71].
3. The L_{A} is computed using

$$L_{\mathrm{A}} = \left(\frac{E_{\mathrm{W}}}{\pi}\right) \cdot \left(\frac{Y_{\mathrm{b}}}{Y_{\mathrm{W}}}\right) = \frac{L_{\mathrm{W}} \cdot Y_{\mathrm{b}}}{Y_{\mathrm{W}}},$$

where $E_{\mathrm{w}} = \pi \cdot L_{\mathrm{w}}$ is the illuminance of reference white point in lux unit, L_{w} is the luminance of reference white point in cd/m^2 unit, Y_{b} is the luminance factor of the background, and Y_{w} is the luminance factor of the reference white point.

References

[1] T. Kanade, P. J. Narayanan, and P. W. Rander, "Virtualized reality: Concepts and early results," in *Proceedings of the 1995 IEEE Workshop on Representation of Visual Scenes*, Washington DC, USA, 1995, pp. 69–76.

[2] M. Shoji, K. Miura, and A. Konno, "U-tsu-shi-o-mi: The virtual humanoid you can reach," in *Proceedings of the ACM 33rd Annual Conference on Computer Graphics and Interactive Techniques*, Boston, MA, USA, 2006.

[3] Hollywood Camera Work LLC. (2008) Green Screen Plates. [Online]. Available: http://www.hollywoodcamerawork.us/greenscreenplates.html

[4] T. Porter and T. Duff, "Compositing digital images," in *Proceedings of the ACM 11th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1984, pp. 253–259.

[5] J. Foster, *The Green Screen Handbook: Real-World Production Techniques*, 1st ed. Indianapolis, IN, USA: Wiley Publishing, 2010.

[6] C. Wheatstone, "Contributions to the Physiology of Vision. Part the First. On Some Remarkable, and Hitherto Unobserved, Phenomena of Binocular Vision,"

Philosophical Transactions of the Royal Society of London, vol. 128, pp. 371–394, Jan. 1838.

[7] L. Zhang, C. Vazquez, and S. Knorr, "3D-TV Content Creation: Automatic 2D-to-3D Video Conversion," *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 372–383, Jun. 2011.

[8] Technicolor SA, "3D Post-Production Workflow." [Online; accessed on 2013-02-14]. Available: http://www.technicolor.com/en/solutions-services/technology/technology-licensing/3d-excellence/3d-post-production-workflow

[9] The Foundry Visionmongers Ltd., "What is stereoscopic 3D." [Online; accessed on 2013-08-02]. Available: http://www.thefoundry.co.uk/products/ocula/about-stereoscopic-3d

[10] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott, "A perceptually motivated online benchmark for image matting," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009, pp. 1826–1833.

[11] Ross Video Ltd., "Augmented Reality (AR) Solution." [Online; accessed on 2014-06-30]. Available: http://www.rossvideo.com/solutions/augmented-reality

[12] ——, "Ross Video YouTube Uploads." [Online; accessed on 2014-06-30]. Available: http://www.youtube.com/user/RossVideoLtd

[13] Sony Electronics, Inc., "SONY Stereo Image Processing Platform." [Online; accessed on 2014-06-30]. Available: https://pro.sony.com/bbsc/video/channels-switchers/video-cinealta_mpe200_3d_image_processing_platform

[14] Rohde & Schwarz DVS GmbH, "3D with VENICE and STAN." [Online; accessed on 2014-06-30]. Available: http://www.dvs.de/products/video-systems/venice/venice/3d-with-stan.html

[15] B. B. Breese, "Binocular rivalry," *Psychological Review*, vol. 16, no. 6, pp. 410–415, Nov. 1909.

[16] L. Yin and J. Zhao, "Real-time automatic chroma-key matting using perceptual analysis and prediction," in *Proceedings of the 27th IEEE Canadian Conference on Electrical Computer Engineering*, Toronto, ON, Canada, 2014, pp. 1–4.

[17] F. Williams, "Method of taking motion-pictures." U.S. Patent 1 273 435, Jul. 23, 1918.

[18] Lammeyer. Smiling senior woman in front of black background. Accessed through SS SPV LLC (d/b/a BigStock and BigStockPhoto). [Online]. Available: http://www.bigstockphoto.com/image-60541955

[19] C. D. Dunning, "Composite photography," *Transactions of the Society of Motion Picture Engineers*, vol. 12, no. 36, pp. 975–979, Sep. 1928.

[20] C. H. Dunning, "Method and means of producing composite photographs with sound accompaniment," U.S. Patent 1 858 767, May 17, 1932.

[21] Oscars, "The Official Academy Awards®Database." [Online]. Available: http://awardsdatabase.oscars.org

[22] J. A. Ball, "The technicolor process of three-color cinematography," *Journal of the Society of Motion Picture Engineers*, vol. 25, no. 2, pp. 127–138, Aug. 1935.

[23] W. E. Pohl, "Traveling matte," U.S. Patent 2 693 126, Nov. 2, 1954.

[24] Czamfir. Studio portrait of beautiful woman with long hair on blue background. Accessed through SS SPV LLC (d/b/a BigStock and BigStockPhoto). [Online]. Available: http://www.bigstockphoto.com/image-49289564

[25] F. W. Gage and A. W. Tondreau, "Process of composite photography," U.S. Patent 2 651 233, Aug. 10, 1953.

[26] P. Vlahos, "Composite photography utilizing sodium vapor illumination," U.S. Patent 3 095 304, Jun. 25, 1963.

[27] J. Foster. (2009) The Masters of Compositing Series - Part 1: Petro Vlahos. [Online; accessed on 2013-02-20]. Available: http://provideocoalition.com/pvcexclusive/story/the_masters_of_compositing_series_-_part_1_petro_vlahos

[28] P. Vlahos, "Composite color photography," U.S. Patent 3 158 477, Nov. 24, 1964.

[29] P. Vlahos, A. Dadourian, and G. Sauve, "Method and apparatus for adjusting parameters used by compositing devices," U.S. Patent 5 907 315, May 25, 1999.

[30] V. G. Devereux, "Television animation store: Digital chroma, key and mixer units," British Broadcasting Corp., London, United Kingdom, BBC-RD 1984/16, Dec. 1984.

[31] The Foundry Visionmongers Ltd. (2008) User Guide for Keylight on Nuke. [Online]. Available: http://www.thefoundry.co.uk/support/user-guides#keylight

[32] Photron Ltd. Primatte Whitepapers. [Online; accessed on 2013-08-23]. Available: http://www.primatte.com/content.cfm?n=whitepapers

[33] Y. Mishima, "A software chromakeyer using polyhedric slice," in *Proceedings of the NICOGRAPH 1992 Conference*, Tokyo, Japan, 1992, pp. 44–52.

[34] ——, "Soft edge chroma-key generation based upon hexoctahedral color space," U.S. Patent 5 355 174, Oct. 11, 1994.

[35] The Foundry Visionmongers Ltd. (2008) Fusion Tool Reference – Primatte: Part 4. [Online]. Available: http://eyeonmanual.com/fusion/tool-reference/primatte/primatte-part-4

[36] M. D. Fairchild, *Color Appearance Models*, 3rd ed., ser. The Wiley-IS&T Series in Imaging Science and Technology. USA: Wiley Publishing, 2013.

[37] J. Wang and M. F. Cohen, "Image and video matting: A survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 2, pp. 97–175, Jan. 2007.

[38] T. Mitsunaga, T. Yokoyama, and T. Totsuka, "Autokey: Human assisted key extraction," in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1995, pp. 265–272.

[39] M. Ruzon and C. Tomasi, "Alpha estimation in natural images," in *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head Island, SC, USA, 2000, vol. 1, pp. 18–25.

[40] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski, "A bayesian approach to digital matting," in *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, HI, USA, 2001, vol. 2, pp. 264–271.

[41] E. S. L. Gastal and M. M. Oliveira, "Shared sampling for real-time alpha matting," *Computer Graphics Forum*, vol. 29, no. 2, pp. 575–584, May 2010.

[42] C. Rhemann, C. Rother, and M. Gelautz, "Improving color modeling for alpha matting," in *Proceedings of the 2008 British Machine Vision Conference*, UK, 2008, pp. 115.1–115.10.

[43] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum, "Poisson matting," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 315–321, Aug. 2004.

[44] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann, "Random walks for interactive alpha-matting," in *Proceedings of the 5th IASTED International Conference on Visualization, Imaging, and Image Processing*, Palma de Mallorca, Spain, 2005, pp. 423–429.

[45] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 228–242, Feb. 2008.

[46] B. He, G. Wang, C. Shi, X. Yin, B. Liu, and X. Lin, "Iterative transductive learning for alpha matting," in *Proceedings of the 20th IEEE International Conference on Image Processing*, Melbourne, Australia, 2013, pp. 4282–4286.

[47] J. Wang and M. Cohen, "Optimized color sampling for robust matting," in *Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, USA, 2007, pp. 1–8.

[48] Rhemann, C. and Rother, C. and Jue Wang and Gelautz, M. and Kohli, P. and Rott, P., "Alpha Matting Evaluation Website." [Online; accessed on 2013-11-25]. Available: http://www.alphamatting.com

[49] Z. Zhang, Q. Zhu, and Y. Xie, "Learning based alpha matting using support vector regression," in *Proceedings of the 19th IEEE International Conference on Image Processing*, Orlando, FL, USA, 2012, pp. 2109–2112.

[50] E. Shahrian, D. Rajan, B. Price, and S. Cohen, "Improving image matting using comprehensive sampling sets," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, USA, 2013, pp. 636–643.

[51] X. Chen, D. Zou, S. Zhou, Q. Zhao, and P. Tan, "Image matting with local and nonlocal smooth priors," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, USA, 2013, pp. 1902–1907.

[52] O. Wang, J. Finger, Q. Yang, J. Davis, and R. Yang, "Automatic natural video matting with depth," in *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, Maui, HI, USA, 2007, pp. 469–472.

[53] T. Lu and S. Li, "Image matting with color and depth information," in *Proceedings of the 21st International Conference on Pattern Recognition*, Tsukuba, Japan, 2012, pp. 3787–3790.

[54] A. R. Smith and J. F. Blinn, "Blue screen matting," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1996, pp. 259–268.

[55] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum, "Flash matting," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 772–778, Jul. 2006.

[56] M. Ben-Ezra, "Segmentation with invisible keying signal," in *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head, SC, USA, 2000, vol. 1, pp. 32–37.

[57] M. McGuire, W. Matusik, and W. Yerazunis, "Practical, real-time studio matting using dual imagers," in *Proceedings of the 17th EuroGraphics Conference on Rendering Techniques*, Nicosia, Cyprus, 2006, pp. 235–244.

[58] L. Bentabet and H. Zhang, "Bayesian image matting using infrared and color cues," in *Proceedings of the 5th International Conference on Image and Signal Processing*, Agadir, Morocco, 2012, pp. 1–8.

[59] D. E. Zongker, D. M. Werner, B. Curless, and D. H. Salesin, "Environment matting and compositing," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1999, pp. 205–214.

[60] Y.-Y. Chuang, D. E. Zongker, J. Hindorff, B. Curless, D. H. Salesin, and R. Szeliski, "Environment matting extensions: Towards higher accuracy and real-time capture," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 2000, pp. 121–130.

[61] Z. Luo, "Color range determination and alpha matting for color images," Master's thesis, University of Ottawa, Ottawa, ON, Canada, 2011.

[62] C. Hao, "Improved sampling-based alpha matting in images and video," Master's thesis, University of Ottawa, Ottawa, ON, Canada, 2012.

[63] E. B. Goldstein, *Sensation and Perception*, 8th ed. Belmont, CA, USA: Wadsworth Publishing, 2009.

[64] R. G. Kuehni, "Variability in unique hue selection: A surprising phenomenon," *Color Research & Application*, vol. 29, no. 2, pp. 158–162, Apr. 2004.

[65] K. Allen, "Locating the unique hues," *Rivista di Estetica: Ontologia dei colori*, vol. 50, no. 43, pp. 13–28, Jan. 2010.

[66] A. H. Munsell, "A pigment color system and notation," *The American Journal of Psychology*, vol. 23, no. 2, pp. 236–244, Apr. 1912.

[67] E. Landa and M. Fairchild, "Charting color from the eye of the beholder," *American Scientist*, vol. 93, no. 5, pp. 436–443, Sep. 2005.

[68] Commission Internationale de l'Eclairage, *Colorimetry*, 2nd ed., ser. CIE Technical Report. Commission Internationale de l'Eclairage, CIE Central Bureau, 1986.

[69] F. Ebner and M. D. Fairchild, "Development and testing of a color space (ipt) with improved hue uniformity," in *Proceedings of the 6th Color Imaging Conference: Color Science, Systems and Applications*, Scottsdale, Arizona, 1998, no. 1, pp. 8–13.

[70] M. R. Luo and R. W. G. Hunt, "The structure of the cie 1997 color appearance model (ciecam97s)," *Color Research & Application*, vol. 23, no. 3, pp. 138–146, Jun. 1998.

[71] P. J. Alessi, G. Dispoto, M. D. Fairchild, X.-F. Feng, R. W. Hunt, H. Komatsubara, L. C., M. R. Luo, M. M., Belgium, N. Moroney, T. Newman, and H. Yaguchi, "A color appearance model for color management systems: CIECAM02," Tech. Committee 8-01, Commission Internationale de l'Eclairage, Vienna, Austria, CIE Technical Report 159:2004, 2004.

[72] C. Li, M. R. Luo, R. W. G. Hunt, N. Moroney, M. D. Fairchild, and T. Newman, "The performance of ciecam02," 2002-01-01, vol. 2002, no. 1, pp. 28–32.

[73] Y. Xue, "Uniform color spaces based on ciecam02 and ipt color difference equations," Ph.D. dissertation, Rochester Institute of Technology, New York, NY, USA, 2008.

[74] M. R. Luo, G. Cui, and C. Li, "Uniform color spaces based on ciecam02 color appearance model," *Color Research & Application*, vol. 31, no. 4, pp. 320–330, Aug. 2006.

[75] *Exchangeable image file format for digital still cameras: Exif Version 2.2*, Japan Electronics and Information Technology Industries Association, Standard JEITA CP-3451.

[76] *Parameter values for the HDTV standards for production and international programme exchange*, International Telecommunication Union, Recommendation BT.709-5, 2002.

[77] FXHOME Ltd. PhotoKey 6. [Online; accessed on 2013-09-30]. Available: http://fxhome.com/photokey-6

[78] The Foundry Visionmongers Ltd., "Keylight." [Online; accessed on 2013-09-30]. Available: http://www.thefoundry.co.uk/products/plugins/keylight

[79] K. Perlin, "Improving noise," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 681–682, Jul. 2002.

[80] D. Garcia, "Robust smoothing of gridded data in one and higher dimensions with missing values," *Computational Statistics & Data Analysis*, vol. 54, no. 4, pp. 1167–1178, Apr. 2010.

[81] Teorex. Inpaint Version 6.0. [Online; accessed on 2014-06-30]. Available: http://www.theinpaint.com

[82] D. L. MacAdam, "Visual sensitivities to color differences in daylight," *Journal of the Optical Society of America*, vol. 32, no. 5, pp. 247–274, May 1942.

[83] D. Scharstein and R. Szeliski. Middlebury Stereo Evaluation. [Online; accessed on 2014-07-30]. Available: http://vision.middlebury.edu/stereo/eval

[84] ——, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 7–42, Apr. 2002.

[85] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *Proceedings of the 18th International Conference on Pattern Recognition*, Hong Kong, China, 2006, vol. 3, pp. 15–18.

[86] C. Tomasi and R. Manduchi, "Bilateral filtering for grey and color images," in *Profeedings of the 6th International Conference on Computer Vision*, Bombay, India, 1998, pp. 839–846.

[87] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007.

[88] NVIDIA Corp. NVIDIA GPUDirect. [Online; accessed on 2013-10-26]. Available: https://developer.nvidia.com/gpudirect

[89] ——. NVIDIA SLI Technology. [Online; accessed on 2013-08-15]. Available: http://www.nvidia.com/page/sli.html

[90] Advanced Micro Devices, Inc. AMD Crossfire Technology. [Online; accessed on 2013-08-15]. Available: http://www.amd.com/en-us/innovations/software-technologies/crossfire

[91] *For Television – 1920 × 1080 Image Sample Structure, Digital Representation and Digital Timing Reference Sequences for Multiple Picture Rates*, The Society of Motion Picture and Television Engineers, ST 274:2008, 2008.

[92] NVIDIA Corp. Unified Memory in CUDA 6. [Online; accessed on 2014-06-30]. Available: http://devblogs.nvidia.com/parallelforall/unified-memory-in-cuda-6

[93] Advanced Micro Devices, Inc. AMD Pinned Memory. [Online; accessed on 2013-10-26]. Available: https://www.opengl.org/registry/specs/AMD/pinned_memory.txt

[94] T. Scheuermann and J. Hensley, "Efficient histogram generation using scattering on gpus," in *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, Seattle, WA, USA, 2007, pp. 33–37.

[95] Microsoft Corp. HLSL. [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/bb509561(v=vs.85).aspx

[96] NVIDIA Corp. Parallel Programming and Computing Platform. [Online]. Available: http://www.nvidia.ca/object/cuda_home_new.html

[97] C. Nugteren, G.-J. van den Braak, H. Corporaal, and B. Mesman, "High performance predictable histogramming on gpus: Exploring and evaluating algorithm trade-offs," in *Proceedings of the 4th Workshop on General Purpose Processing on Graphics Processing Units*, Newport Beach, CA, USA, 2011, pp. 1:1–1:8.

[98] The Khronos Group. OpenGL Shading Language. [Online]. Available: http://www.opengl.org/documentation/glsl

[99] ——. Shader Subroutine. [Online]. Available: http://www.opengl.org/registry/specs/ARB/shader_subroutine.txt

[100] Advanced Micro Devices, Inc. The Standard for Embedded Accelerated 3D Graphics. [Online; accessed on 2014-04-10]. Available: http://www.khronos.org/opengles

[101] Apple, Inc. Shader Framebuffer Fetch. [Online; accessed on 2014-04-10]. Available: https://www.khronos.org/registry/gles/extensions/EXT/EXT_shader_framebuffer_fetch.txt

[102] Q. Yang, K.-H. Tan, and N. Ahuja, "Real-time o(1) bilateral filtering," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009, pp. 557–564.

[103] T. Q. Pham and L. J. van Vliet, "Separable bilateral filtering for fast video preprocessing," in *Proceedings of the 2005 IEEE International Conference on Multimedia and Expo*, New York, NY, USA, 2005, pp. 1–4.

[104] Autodesk, Inc. Professional Video Effects Software - Smoke. [Online; accessed on 2013-10-16]. Available: http://www.autodesk.com/products/smoke

[105] Adobe Systems, Inc. Adobe Premiere Pro. [Online; accessed on 2013-10-16]. Available: http://www.adobe.com/products/premiere.html

[106] ——. Adobe After Effects. [Online; accessed on 2013-10-16]. Available: http://www.adobe.com/products/aftereffects.html

[107] The Foundry Visionmongers Ltd. Nuke. [Online; accessed on 2013-10-16]. Available: http://www.thefoundry.co.uk/products/nuke-product-family/nuke

[108] Hollywood Camera Work LLC, "Is HD really HD." [Online]. Available: http://www.hollywoodcamerawork.us/is_hd_really_hd.html

[109] Rhemann, C. and Rother, C. and Jue Wang and Gelautz, M. and Kohli, P. and Rott, P., "Alpha Matting Evaluation Datasets." [Online; accessed on 2013-02-10]. Available: http://www.alphamatting.com/datasets.php