Robust Chroma Keying System based on Human Visual Perception and Statistical Color Models

Wenyi Wang

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

in partial fulfillment of the requirements

for the Doctorate in Philosophy degree in Electrical and Computer Engineering

Ottawa-Carleton Institute of Electrical and Computer Engineering

School of Electrical Engineering and Computer Science

Faculty of Engineering

University of Ottawa

Ottawa, Canada

Abstract

In this thesis, we propose a chroma keying system that automatically estimates the alpha map and the reliable intrinsic color of foreground objects in front of solid background. Our system is designed to be capable of distinguishing the transparent foreground from the reflective foreground and shaded background, thereby making the artifacts of the composited image less conspicuous. Specifically, we assume that the transparent region tends to be with higher saturation and lightness compared with region reflecting background light. With this assumption, a threshold function (TF) on a saturation-lightness plane is defined according to human visual experiments. The pixels with color mixed with the background light (conventional unknown pixels) are now further categorized into reflective pixels and transparent pixels according to TF. In this case, the reflective and the transparent regions are separated to improve the alpha matte quality.

Furthermore, a new color representation model is proposed to estimate the intrinsic color of each pixel according to the global color distribution of the image. The underlying assumption of our proposed model is that all colors in a natural image can be approximated by a limited number of chrominance values (dominant colors). Specifically, the color statistics are counted by 2D histogram analysis. Then, we approximate the color distribution by the sum of a set of Gaussian mixture functions (GMF), whose centroids are the dominant colors (D_c) of the image. By choosing colors around each D_c, the possible intrinsic colors for each pixel can be comprehensively and efficiently selected.

Considering the fast development of IP-based video broadcasting, we present a data hiding scheme that can protect the chroma keying results when the image/video data is recorded in the JPEG/H.264 format.

According to our simulation, the proposed chroma keying system generates high quality composited images that are little affected by reflecting, background shading, and intrinsic color missing.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Jiying Zhao, for bringing the problem of chroma keying to me, and for his patience, motivation, and immense knowledge. His guidance helped me in every steps of my research and thesis writing.

In addition, I would like to thank the rest of my thesis committee: Prof. Shahram Shiran, Prof. Richard Dansereau, Prof. Éric Dubois, and Prof. Abed El Saddik, for their insightful comments and encouragement, and also for the questions which inspire me to widen my research from various perspectives.

I thank my fellow labmates, for the great time we worked together, and for all the fun we have had in the last four years.

I would deeply appreciate the encouragement from my parents. It is very important to have their support during my PhD research.

Table of Contents

List of Tables

List of Figures

1

Introduction

1.1 History of image compositing

The technique of **"image/video compositing"** [3] is the process that combines multiple elements into one image/frame with the concern of occlusion relations between elements. Despite the fact that compositing was proposed decades ago, this technique has been still playing an irreplaceable role in modern broadcasting, film making, *etc.* The origin of image compositing dates back to the 19th century, which was almost the same time when film was invented. In 1898, Georges Méliès played a visual trick by using mattes for multiple exposures in his film "Our Heads Are Better Than One". The usage of mattes here can be regarded as the naive beginning of what we now think of as green-screen compositing. In the production of this film, a piece of glass (i.e., matte) is partially painted into black to prevent part of the film being exposed.

After the first round of filming, Méliès would rewind the film and expose the frame part that had been under the matte earlier. By using double exposure, two and even more shots can be combined into just one frame. After this compositing idea had been proposed, it was gradually widely used in film making from the early silent film era to the modern digital film era.

Given this compositing technique, one fair question is why we need to use image/video compositing instead of recording the film in the real scene. In the 19th century, orthochromatic filming required large amount of light, therefore making indoor filming much more practical and reliable. With techniques improving, outdoor filming becomes feasible because of the improved camera sensing technology. The film compositing, however, is still indispensable because of its many advantages:

- Different elements in the scene can not get pleasant exposure within one shot. Take the actor standing in the dark environment for example, if we want to expose the environment clear enough, the front actor would get over exposed, and vice versa.
- Sometimes, a stunt could be too dangerous for people to act on location.
- In fantasy or science-fiction movies, the scene and the location does not exist and needs to be computer generated.
- Video/image compositing can save large amount of the budget by avoiding outdoor filming.

Because of all these advantages from image/video compositing, research on this technique is still very active in both academic and industry fields. In the process of image compositing, one essential problem is to extract target elements from the original image. Generally speaking, this problem can be categorized into image segmentation, which will be briefly summarized and introduced in the following section.

1.2 Image segmentation in computer vision

Image segmentation is one of the most fundamental problems in computer vision [6] [7] [8]. A convincing and reliable image partition is usually the output from complex hierarchical processes that involves low-level, mid-level, and high-level visions as shown in Fig. 1.1. Despite decades of extensive research, image segmentation is still far from perfect in human perspective.

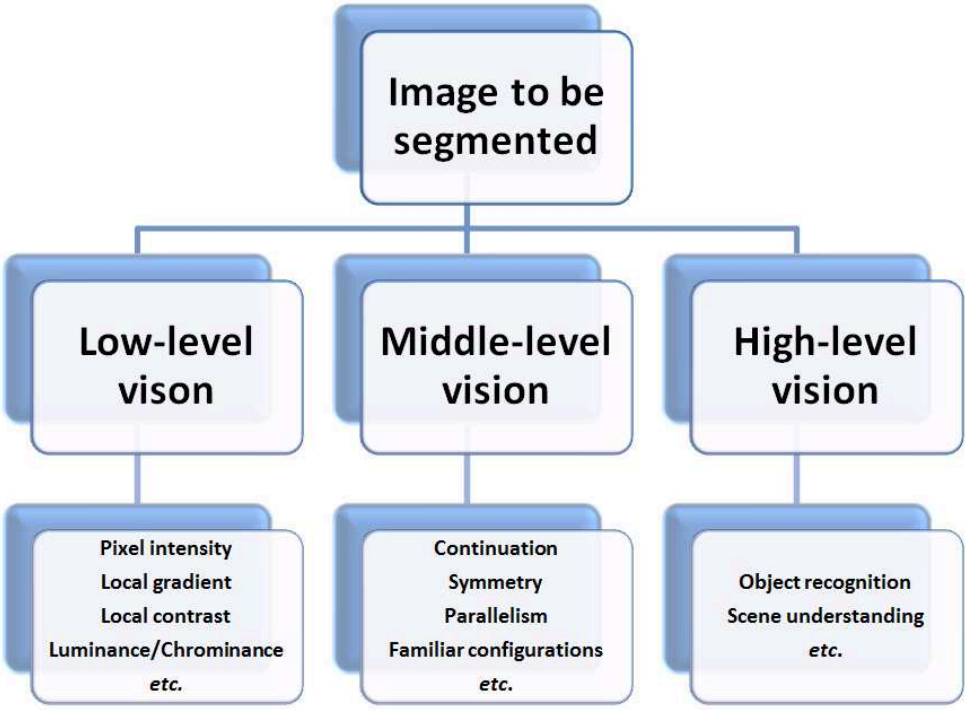


Figure 1.1: Hierarchical processing levels in computer vision.

In low-level computer vision methods, the properties of image **local intensity** (i.e., luminance, chrominance, gradients, texture, *etc.*) are directly used to group the pixels in the image into homogenous regions. However, an image segmentation based on low-level information alone is known to be ambiguous and inflexible. In Fig. 1.2, the example patches cannot provide reliable cues for objects discrimination. Specifically, the patches taken from the same object (i.e., left ones from the stones, and right ones from lizard) can have different textures and colors. Meanwhile, the

patches with the same index number in Fig. 1.2 have similar low-level characters (e.g., texture and color) even though they are from different objects. In this case, it would be impossible to generate reliable image segmentation just from low-level information [9] [10].

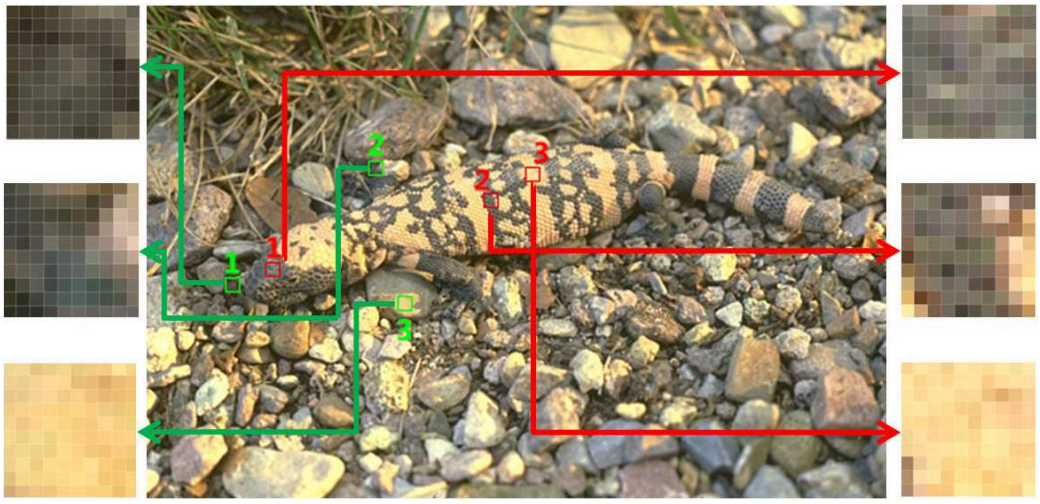
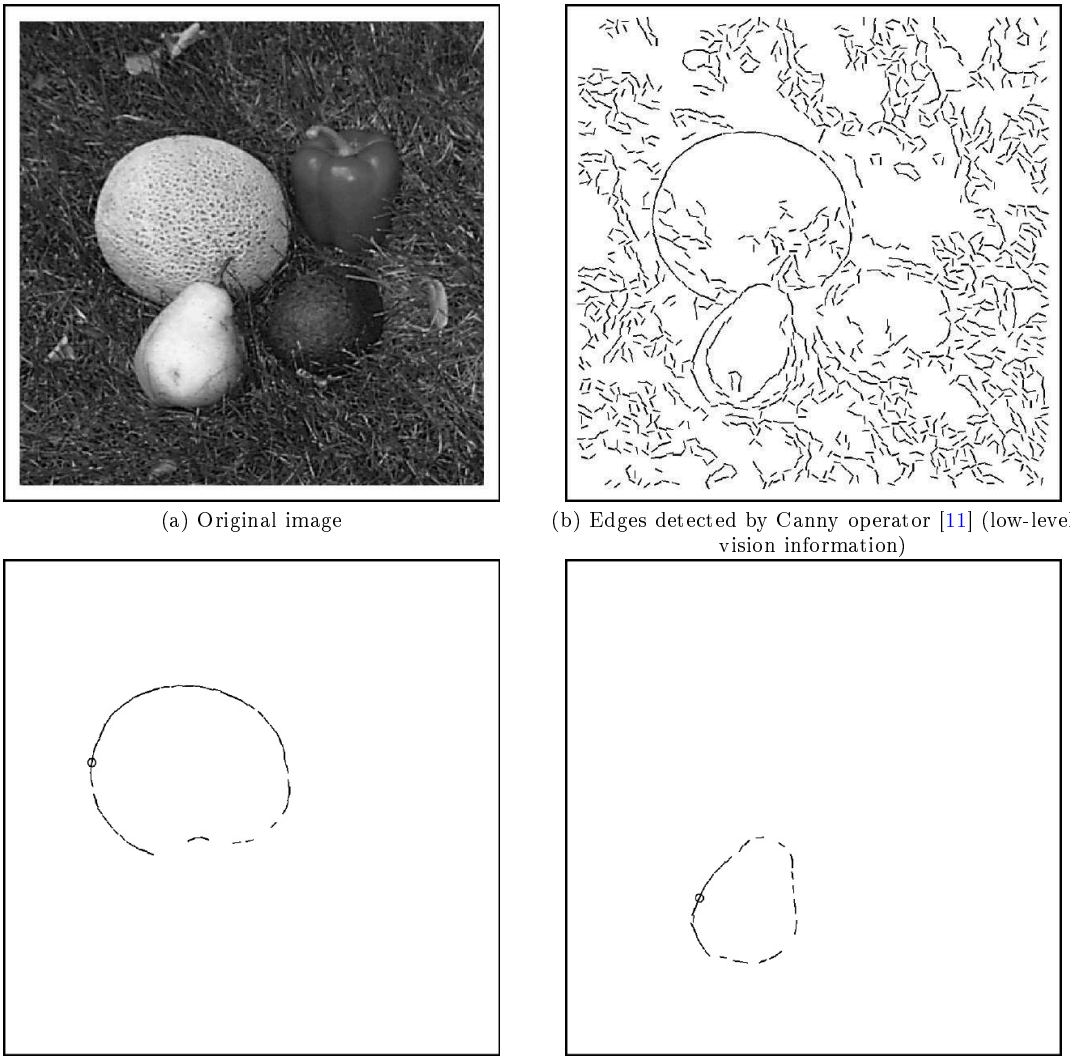


Figure 1.2: Ambiguous problems in low-level vision.

In order to overcome the limitations of low-level vision, middle-level vision problems are researched to extract advanced feature information that provides more robust cues for object discrimination. Compared to low-level vision that concentrates on pixels, middle-level vision aims to represent the world by means of objects and structures with various orientation, illumination, spatial occlusion, and movement. Based on low-level vision information, middle-level information extraction often involves the analysis of the interaction between local image features, which is referred as **context**. In order to model the context in an image, different quantitative metrics are designed to define each middle-level vision cue, such as continuation, symmetry, parallelism, familiar configurations, *etc.* [1]. As shown in Fig. 1.3 (b), the edges in the original image are the low-level vision information which is extracted by the local pixel intensity variations. Based on edge information, objects can be further identified with assumptions and knowledge of middle-level vision cues as shown in

Fig. 1.3 (c) (d).



(a) Original image

(b) Edges detected by Canny operator [11] (low-level vision information)

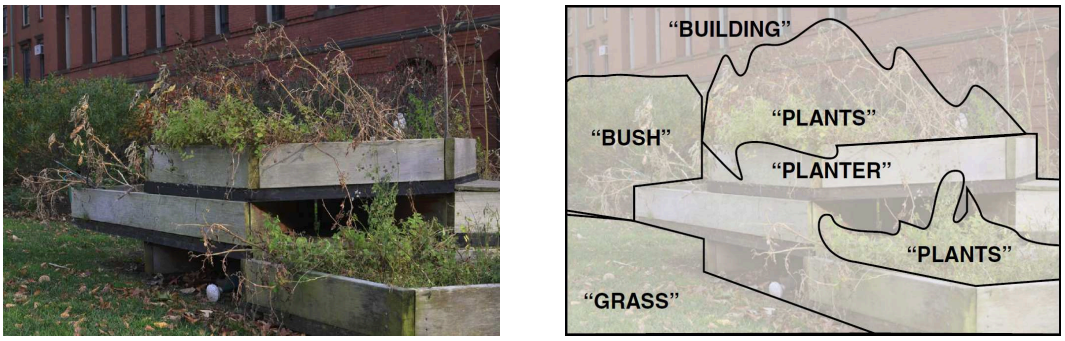
(c) The first detected object (mid-level vision information)

(d) The second detected object (mid-level vision information)

Figure 1.3: Objects detection by using middle-level vision cues. The experimental results are from [1].

In addition to low-level and middle-level vision problems, it comes to the most advanced task, which is known as high-level vision problems. **Semantic labeling**, which is considered as one of the most important high-level vision problems, aims to identify and analyze the identities of objects in an image in spite of the position, size, lighting condition, the presence of nearby object, *etc.* Besides the intuitive identity labeling, semantic labeling can involve more comprehensive analysis [2] of

the identified objects, such as the size, depth, and materials. As shown in Fig. 1.4, the original image (a) is segmented into different regions in (b) with respect to object identities in the scene. Such image segmentation involves the task of scene understanding, which makes the computer vision more consistent with human vision.



(a) Original image

(b) Image segmentation with object identities (high-level vision segmentation).

Figure 1.4: Image segmentation in high-level vision. The results are from [2].

1.3 Alpha matting in natural image

Other than general image segmentation, some specific segmentation problems are also compelling for decades. One of such problems is **"alpha matting"** [12] [13], which refers to the problem of finely extracting interested objects from still natural images or video sequences. Generally, the objects to be extracted are referred as **"foreground object"**, while the rest of the image/frame is referred as **"background region"**. Since foreground/background separation can be regarded as semantic labeling, the matting problem can be categorized as a high level vision problem.

The object extraction here is not a rough separation between foreground and background. Instead, this problem involves pixel-level segmentation, occlusion representation, and intrinsic foreground color restoration. Specifically, color ambiguity between foreground/background pixels is inevitable at fuzzy/blurred edges (1.5 (a1)), in reflective regions (1.5 (b1)), and in transparent regions (1.5 (c1)). As shown in Fig. 1.5, the observed foreground color could be the result of mixing the intrinsic

foreground color with the background color in many different situations. Besides the visual appearance of color ambiguity presented in Fig. 1.5 (a1) (b2) (c1), the color distribution in RGB color space is also presented in Fig. 1.5 (a2) (b2) (c2). From the color distributions, it is obvious that the colors of foreground and background are not separately distributed. Instead, there is a transition region between the foreground and background colors in the color space. This transition region makes the color distribution continuous, therefore introducing difficulties for reliable and accurate hard segmentation between the foreground color and background color. An example of the dilemma problem in hard segmentation is as shown in Fig. 1.6. The hard segmentation here means a binary mask is used so that the image is exclusively segmented into two parts: absolute foreground and absolute background. If we want to remove the background region as much as possible, it is highly possible that the fine details of the foreground objects are also removed as shown by Fig. 1.6 (d). If we want to keep the foreground details as much as possible, a halo with background color is very likely to be observed around the extracted foreground as shown by Fig. 1.6 (e).

In order to finely separate the foreground object from the background and to restore the original foreground color for mixed pixels, it is important to model the way that foreground/background color mix. This problem was first mathematically described by the landmark paper proposed by Porter and Duff in 1984 [3]. Generally, the observed pixel color is modeled by a convex combination of the foreground object color and the background color:

$$C_{(i,j)} = \alpha_{(i,j)} F_{(i,j)} + (1 - \alpha_{(i,j)}) B_{(i,j)} \,, \tag{1.1}$$

where (i, j) refers to the pixel coordinates, $F_{(i,j)}$ and $B_{(i,j)}$ are the foreground and background colors, and $\alpha_{(i,j)}$ is the blending factor that varies from 0 (completely background) to 1 (completely foreground).

By solving Equ. (1.1), the observed pixel color is decomposed into two parts:

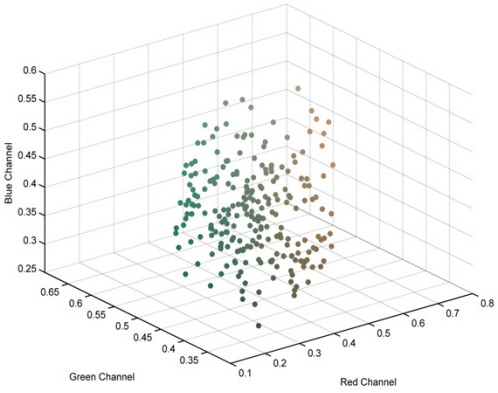
foreground color and background color. In the region of absolute foreground objects, the observed pixel color is exclusively contributed by foreground color, the value of α is equal to one. In absolute background region, the observed pixel color is exclusively contributed by background color, the value of α is therefore equal to zero. In regions where foreground/background color mix together, the α value varies in the range of $0-1$. The particular α value for each mixed pixel is determined by the local color mixture condition. By applying such color mixing model, the foreground objects can be accurately and softly extracted from the background, as shown by the alpha map in Fig. 1.6 (c). Furthermore, the intrinsic foreground color before color mixing can also be restored as shown by the extracted foreground in Fig. 1.6 (f). It is obvious in the provided example that soft segmentation using alpha matting is superior to hard segmentation because alpha matting keeps fine details of the foreground objects and removes the mixed background color from the foreground at the same time.

However, the problem of alpha matting is severely under constrained, and thus there is no unique solution of Equ. (1.1). This is because there are two color vectors ($F_{(i,j)}$, $B_{(i,j)}$) and one blending factor ($\alpha_{(i,j)}$) to be solved with just one equation. In this case, prior assumptions and side information are always required in existing alpha matting methods. Generally, **"trimap"** [14] [15] and **"scribbles"** [16] [17] are two most commonly used side information in alpha matting solutions. An example of these two side information can be found in Fig. 1.7. In the provided example, the image regions covered by red and green in Fig. 1.7 (b) are predefined to be absolute foreground and background regions, respectively. In this case, alpha values of pixels in absolute foreground/background regions can be directly set to be 1 or 0. In the yellow region, which is referred to as the unknown region, the alpha values are estimated based on prior assumptions about pixel color, pixel location, and local texture. Another type of side information is the scribbles as shown in Fig. 1.7 (c). Similar as trimap, part of the pixels in the original image are labeled as absolute foreground/background by the white/black scribbles. Alpha values of the remained

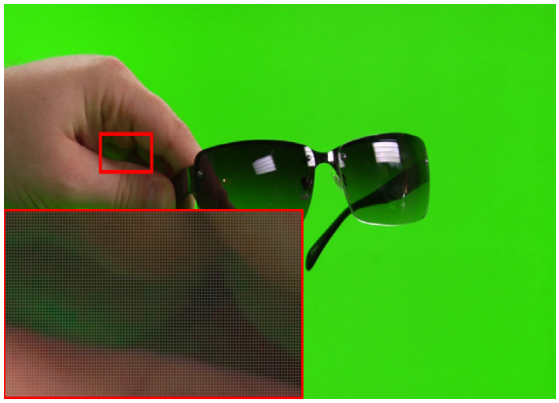
pixels are then estimated based on the color/texture/geometric information.

Since "trimap" and "scribbles" are both used to provide reliable cues for foreground/background color statistics, their quality can significantly affect the final matting result. On the other hand, "trimap" and "scribbles" are often manually drawn by users. In this case, a well specified trimap can involve significant amount of user efforts, which is not friendly to normal users in most situations. In addition, the requirement of manual trimap/scribbles makes it very difficult to do alpha matting on video sequences. Although some previous research was made to do alpha matting on natural video sequences, complete automatism is still difficult to achieve, and strong prior assumptions are often required.

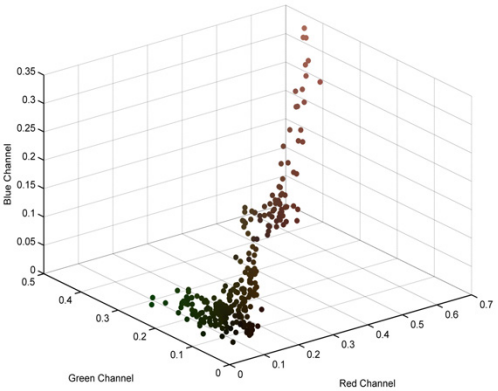
(a1) Foreground with fuzzy edges.



(a2) The RGB color distribution at fuzzy edges.

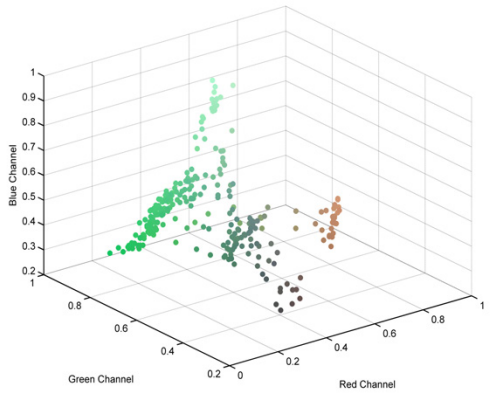


(b1) Foreground with reflection from environment.



(b2) The RGB color distribution in reflective regions.

(c1) Foreground with transparent parts.



(c2) The RGB color distribution in transparent regions.

Figure 1.5: Color mixture between foreground and background under different situations.

(a) Hard Foreground segmentation: mask 1.

(b) Hard Foreground segmentation: mask 2.

(c) Soft Foreground segmentation: alpha.

(d) Extracted foreground with mask 1.

(e) Extracted foreground with mask 2.

(f) Extracted foreground with alpha map.

Figure 1.6: Foreground object extraction using hard segmentation and alpha matting.

(a) Original Image

(b) Image segmented by trimap.

(c) Image labeled by scribbles.

Figure 1.7: Example of trimap and scribbles used to label the image.

1.4 Chroma keying in image/video with monochromatic background

Image/video composition mentioned in Section 1.1 [3] is the technique which accurately and smoothly combines the foreground objects (such as actors/actresses and broadcasters) and the background scene that is taken somewhere else or generated by computer. This technique plays an important role in TV broadcasting, film production, augmented reality and virtual environment because of its good performance, flexible capability of integrating real objects with computer generated scenes as well as the cost saving by avoiding outdoor filming.

Before the foreground objects are composited with the desired background scene, these foreground objects must be finely segmented from their original image/video, especially for reflective parts, transparent regions and boundaries with color spill. Although alpha matting [12] [13] is a good choice for still image editing, it is not always feasible in different application scenarios, such as video editing.

Because of the requirement of manual labels (i.e., trimap/scribbles), it is difficult to achieve the automation and efficiency by using conventional alpha matting methods designed for still natural images. In this case, the most reliable and applicable video matting strategy is **chroma keying** which is the special case of alpha mating. The technology of chroma keying stresses accuracy, efficiency, and automation at the cost of extra efforts on background setup. The images/videos to be chroma keyed are generated by picturing the foreground objects in front of solid color background (usually blue or green, see Fig. 1.8) with the constraint that there should not be similar colors to background on the foreground objects. With the simple background and separable foreground-background color distribution, the chroma keying system can efficiently extract the foreground objects along with their transparency property by removing the background color in each pixel.

Although it is assumed that the background is monochromatic and evenly lighted,

the color on background can never be completely constant in practice because of the lighting condition and background setup. In professional studio environment as shown in Fig. 1.8 (a), the background can be with almost constant color with multiple carefully positioned lighting source. However, the situation becomes challenging when it turns to simple background setup which is the major circumstance for normal users. As shown by Fig. 1.8 (b), a simple green background setup can adversely affect the background color in many different ways, such as uneven lighting condition, pale color appearance, and crumpled background surface. Besides the luminance diversity on the background, the scenes in practice can break the assumption that there are no similar colors to background on the foreground. This often happens when the background light partially passes through the foreground, or reflects on the foreground.

With these problems, background color removal involves three different situations: completely transparent, semi-transparent, and color spill. As shown in Fig. 1.9, there are three indexed regions representing different background color mixture situations. Region 1 is completely transparent (alpha values should equal 0) even though the color in this region is not completely even due to the lighting variation and crumpled background; region 2 is semi-transparent (alpha values should be in the range of 0 - 1); region 3 is completely opaque (alpha values should equal 1) although it reflects the background color and looks similar to semi-transparent region (e.g., region 2). The problem of reliably distinguishing these different situations is one of the most important and challenging problems in a chroma keying system.

(a) Professional studio environment for chroma keying.

(b) Home made environment for chroma keying.

Figure 1.8: The background environment setup by expert and normal user respectively.

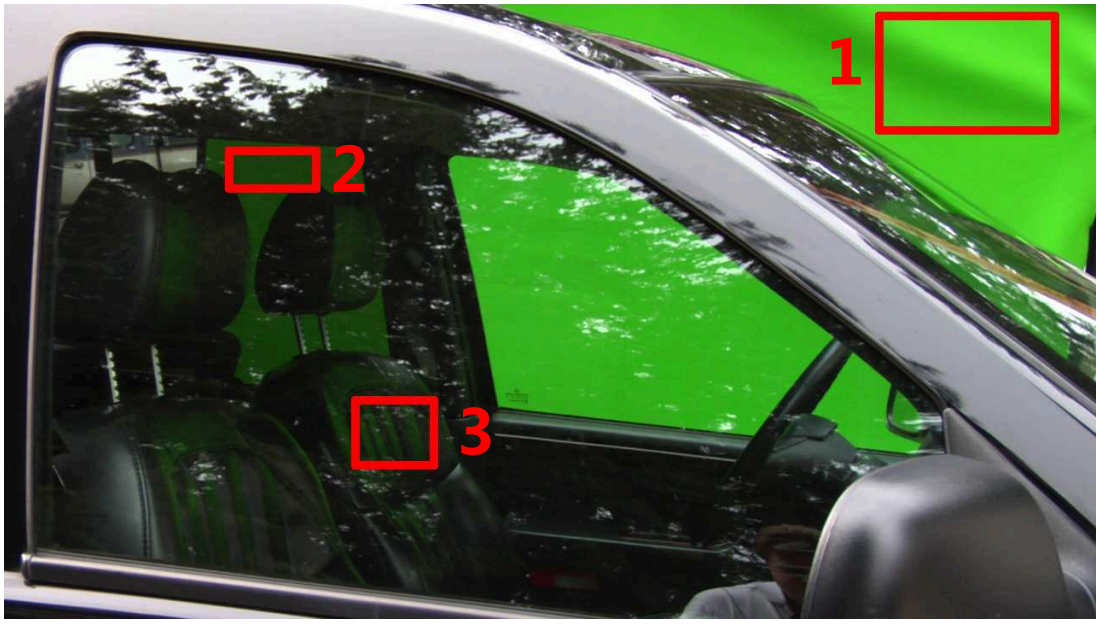


Figure 1.9: Example of three different background color mixture situations happen in one image.

1.5 Benchmark and matting quality

The matting problems, including alpha matting (natural images) and chroma keying (solid background images), have been widely researched for many years. Matting quality evaluation metrics and tools are desirable.

Rhemann and his colleagues contributed an online benchmark [18] [19] for the quality evaluation and comparison of alpha matting algorithms. In their work, a free access online benchmark provides testing images, groundtruth alpha maps and predefined trimaps to global users and researchers. The users can upload the estimated alpha maps of the testing images by using their own algorithms and get the quality evaluation of their results. The metrics used for alpha quality evaluation in Rhemann's work includes the conventional ones such as SAD (sum of absolute differences), and MSE (mean squared errors). However, SAD and MSE may not be the optimal metrics for alpha map quality because of the fact that the alpha map is a property map instead of an image to be directly viewed. The composited image based on alpha map is the actual result to be presented to users. In this case, gradient distance and connectivity distance are proposed as the human perceptual driven distance measurements. The gradient distance measures the over-smoothing and erroneous discontinuities in the generated alpha maps. The connectivity distance measures the undesirable disconnected foreground objects, such as fractured hair floating in the air.

In industry, the matting quality is mostly evaluated for the results from chroma keying although there is still no standard quantitative metrics. Instead, the engineers prefer to test audience ranking from normal and professional viewers. Despite of the lack of quantitative metrics, there are still multiple objective concerns in industry for the matting quality of chroma keying [20]:

- The capability of keeping foreground objects complete opaque while clearly removing all background regions despite of the light condition, shadow, and

noises.

- The capability of keeping the details on foreground objects' boundary, such as hair foaming in the air.
- The uniformity of the lightness and chrominance along the object contour.
- The artifacts caused by noise and image textures.
- The accurate and sharp details in angles and small enclosed spaces, such as small holes and corners.
- The reliable transparency estimation in fabrics, glass, *etc.*
- The reliable differentiating between transparency and reflective (color spill).
- The foreground color restoration in the transparent and reflective regions.

1.6 Scope and structure of the thesis

In this thesis, we focus on developing a robust chroma keying system involving foreground/background region auto detection, color modeling, transparency estimation, color spill suppression, foreground restoration, and alpha covert transmission. Our system is proposed to not only perform well in a professional studio environment but also robustly perform under a poor background setup from normal users. We believe such a chroma keying system would be more friendly to normal users and be more flexible in different applications.

In order to have a comprehensive understanding of different matting algorithms, we give an insight review and evaluation of the existing matting algorithms including the alpha matting methods and the chroma keying methods. The working principles of each method are demonstrated, and the performance of each method is also presented along with detailed explanations of the underlying theories and experimental results. The advantages and disadvantages for each method are also discussed from

both theoretical and experimental perspectives. The related fundamental knowledge about matting and image processing used in this thesis are introduced in **Chapter 2**. In **Chapter 3**, the literature review of existing matting algorithms are presented.

Based on the literature review in Chapter 3, it can be observed that the foreground color estimation is a critical step for the reliable alpha estimation and the subsequent image compositing. Most matting methods only considered local foreground color properties, while the other few methods tried to consider the foreground color in global range at the cost of huge computational cost. In order to comprehensively and efficiently find the foreground colors, we propose a novel color representation model in **Chapter 4**. In our color representation model, the global color distribution of an image is first estimated by using Gaussian mixture models (GMMs). After that, a small number of major color components can be extracted to represent the image with high quality. This approach can adaptively compress the colors in an image into a few unique ones without the loss of color diversity. Therefore, we can globally estimate the foreground colors while the computational cost is low. The experimental results are also provided to illustrate that our proposed model can accurately and reliably represent the image with a small color set.

Besides the foreground color estimation, we tried to solve several other unsettled chroma keying problems, such as removing uneven background, and managing the dilemma between transparency and reflection. Given the uneven background, it would be difficult to completely detect and remove the whole background region while foreground objects are kept completely opaque. For the dilemma between transparency and reflection, we need a reliable mechanism to differentiate the two situations despite that the color mixture could appear very similar. In this case, we propose a quad-map based chroma keying method in **Chapter 5**. Based on the color statistics and the thresholds from human visual experiments, the image to be chroma-keyed is automatically segmented by a quad-map into four exclusive regions: foreground, background, transparent, and reflective regions. With this initial segmentation, we

can better manage the dilemma between reflective and transparent foreground, and avoid the reflective foreground being estimated to be transparent. Given the known background from the quad-map, the background color in the remained image is estimated based on the global lightness variation so that the estimation would be robust to uneven lighting conditions. Since our proposed chroma keying system can reliably estimate foreground and background colors and differentiate between reflection and transparency, we can generate high quality alpha maps and restore reliable intrinsic foreground colors in complex lighting condition.

In recent years, an important change has been taking place in the video composition technology of broadcasting. The broadcast video engineers have tended to adopt IP (Internet Protocol)-based technologies to replace the SDI (Serial Digital Interface) -based technologies to transmit and distribute the videos to be composited. Although the new technologies are currently designed on LAN (local area network) for video sources synchronization and cost saving, this could be a promising beginning for the live video to be streamed through a wider range of the Internet. For digital information transmitted online, it is important to consider the copyright protection and user access control. Inspired by digital watermarking technique, we proposed a hidden transmission method by inserting the alpha map into the covert JPEG image or H.264 video in **Chapter 6**. This combination is done by using a reversible watermarking algorithm in quantized DCT (discrete cosine transform) domain, so that only the authorized user knowing the watermarking algorithm can get access to the alpha map for further compositing.

In **Chapter 7**, we conclude the thesis and give suggestions for future research work on chroma keying and alpha matting problems.

1.7 Contributions

The researches illustrated in this thesis are mainly focused on the topic of robust chroma keying algorithms. The contributions include the following work:

1. In chroma keying or alpha matting algorithms, reliable foreground color estimation is essential for correct alpha map estimation and further image compositing. Normally, the foreground color estimation is done by examining the color distribution by local sampling. However, the local approach may fail if there is no reliably foreground color cue nearby. On the other hand, the computational cost would be very high if global sampling is involved, especially when the image size becomes large. In order to obtain the most comprehensive foreground color information and keep the computational cost low, we proposed a new image representation model. In our proposed image representation model, a color image can be represented by the combination of a limited number of chrominance, which are called dominant colors in our thesis. Our experimental results showed that the image quality in our representation model can be over 35 dB when the number of dominant colors are between 30-80. In this case, the possible foreground color candidates for alpha estimation can be reduced to a small set, which significantly improves the efficiency and reliability of foreground color estimation. This work was accepted as a conference paper as follows.

   [1] W. WANG, Y. Luo, J. Hu, and J. Zhao, "A Novel Perceptual Oriented Image Color Representation", in *IEEE International Conference on Instrumentation and Measurement Technology (I2MTC'16)*, 2016.

2. Although chroma keying and alpha matting are two closely related research fields, there is one obstacle severely restricts the application of alpha matting algorithms to video chroma keying. The problem is the requirement of manual trimap in conventional alpha matting schemes. By taking the advantage of simple background used in chroma keying, we proposed different auto trimap generation algorithms. Based on our proposed methods, most of the existing alpha matting algorithms are applicable to video chroma keying. These works are published as two conference papers and one journal paper as follows.

[1] Z. Luo, W. WANG, J. Zhao, and Y. Liu, "Color Range Determination and Alpha Matting for Color Images", in *Proceedings of IEEE International Conference on Imaging Systems and Techniques (IST'13)*, pp. 142-145, 2013.

[2] C. Hao, W. WANG, and J. Zhao, "Priority-concerned Chroma-keying", in *IEEE international Symposium on Haptic, Audio and Visual Environments and Games (HAVE'14)*, pp. 130-133, 2014.

[3] C. Hao, W. WANG and J. Zhao, "Video Chroma Keying via Global Sampling and Trimap Propagation", *Multimedia Systems*, pp. 1-15, Oct. 2015.

3. Based on the proposed image representation model and the auto trimap generation methods, we proposed our chroma keying system. In our system, the background color can be estimated and propagated according to the environment lighting condition. The trimap is further refined into quadmap so that the reflective and the transparent regions can be separated even though their color appearance is similar. Therefore, the reflecting foreground object will no longer be estimated as transparent, and the visual quality of the composited image/video could be significantly improved. These works are published as one conference paper and one journal paper as follows.

   [1] W. WANG, and, J. Zhao, "Chroma-Keying based on Global Weighted Sampling and Laplacian Propagation", in *Proceedings of IEEE International Conference on Computational Science and Engineering (CSE'14)*, pp. 851-854, 2014.

   [2] W. WANG and J. Zhao, "Robust Image Chroma-Keying: A Quadmap Approach Based on Global Sampling and Local Affinity", *IEEE Transactions on Broadcasting*, vol. 61, no. 3, pp. 356-366, Apr. 2015.

4. Considering the secure transmission and storage of transparency information,

we proposed a covert transmission mechanism for alpha information. By using the reversible watermarking in quantized DCT (discrete cosine transform) domain, we hide the alpha map into its associated color image/video in JPEG/H.264 format. At the receiver side, only the customer knowing the watermarking algorithm can extract the alpha map for further compositing. This covert transmission proposed here is based on the concern of copyright protection and user access control. This part of work is based on our pervious work on depth information hiding, which are published as two conference papers and one journal paper as follows.

[1] W. WANG, J. Zhao, W.J. Tam, F. Speranza, and Z. Wang, "Hiding Depth Map into Stereo Image in JPEG Format using Reversible Watermarking", in *Proceedings of ACM International Conference on Internet Multimedia Computing and Service (ICIMCS'11)*, pp. 82-85, 2011.

[2] W. WANG, J. Zhao, W.J. Tam, F. Speranza, "Hiding Depth Information into H.264 Compressed Video using Reversible Watermarking", in *Proceedings of ACM Multimedia International Workshop on Cloud-based Multimedia Applications and Services (CMBAS-EH'12)*, pp. 27-32, 2012.

[3] W. WANG and J. Zhao, "Hiding Depth Information in Compressed 2D Image/Video using Reversible Watermarking", *Multimedia Tools and Applications*, pp. 1-19, Feb. 2015.

5. Some other contributions: the choice of color spaces for chroma keying and the GPU parallelized chroma keying implementation are discussed in the following papers.

[1] L. Yin, W. WANG, and J. Zhao, "Stereoscopic Chroma Key Matting using Statistical Analysis in CIECAM02 Color Space", in *IEEE international Symposium on Haptic, Audio and Visual Environments and Games (HAVE'15)*, 2015.

[2] L. Yin, W. WANG and J. Zhao, "Real-time Video Chroma Keying: A Parallel Approach based on Local Texture and Global Color Distribution", *IET Image Processing*, vol. 10, pp. 638-645, Sep. 2016.

2

Fundamental theories and techniques

In this chapter, we introduce the fundamental theories and techniques used in the existing and our proposed matting algorithms.

2.1 Matting problem: a physical perspective

From Chapter 1, we already know that the most accepted matting model is the linear combination:

$$C_{(i,j)} = \alpha_{(i,j)} F_{(i,j)} + (1 - \alpha_{(i,j)}) B_{(i,j)} \,, \tag{2.1}$$

where (i, j) refers to the pixel coordinates, $F_{(i,j)}$ and $B_{(i,j)}$ are the foreground and background colors respectively, and $\alpha_{(i,j)}$ is the blending factor which varies from 0 (completely background) to 1 (completely foreground).

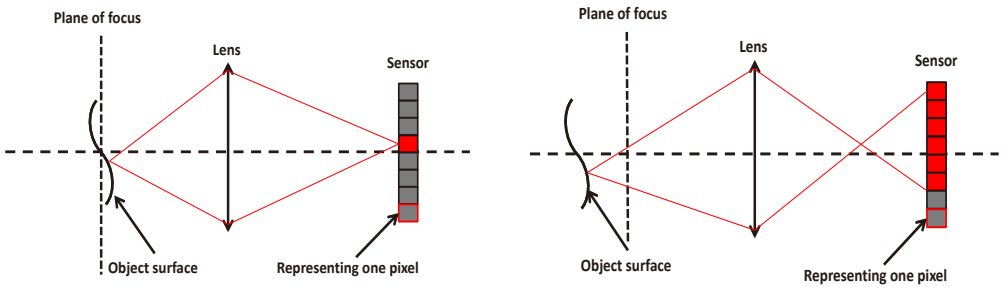
Based on this model, numerous alpha matting algorithms were proposed in the last decade [12] [13]. When the α value in Equ. (2.1) is between 0 and 1, foreground and background colors mix together. The color mixture is usually explained by the transparency effect [21]. It should be noticed that the word "transparency" here covers a wider range than its literal meaning. The "transparency" in matting problems is mostly caused by the following situations, which are also the main concerns in this thesis.

- The material looks semi-transparent if it is made of dense mesh, such as fabrics.
- The material is transparent if it allows light to pass through it (i.e., optically transparent). It should be noticed that martial like colored glass does not follow the color mixture model described by Equ. (2.1), and it is not discussed in this thesis.
- The foreground boundary looks blurred with background color if it only partially covers a pixel.
- The moving object looks blurred with background color.
- The foreground object looks blurred if it is out of focus.

In the aforementioned situations, the finite pixel resolution and the inevitable defocus are two major causes of the color mixture. In this case, we will discuss the mechanism of focus/defocus and the relation between defocus and matting model.

Without loss of generality, we take the basic lens model as shown in Fig. 2.1 for explanation. In Fig. 2.1 (a), it is assumed that the foreground surface is located at the focus plane and the points on the surface can be perfectly focused. In this case, the emitted light from one point on the foreground surface will exactly concentrate on one point on the image plane. On the other hand, we assume that the imaging sensor is constructed by a set of individual sensing elements (the squares in the sensor in

Fig. 2.1), which are corresponding to the pixel values in the image. If the foreground object is perfectly focused, the received light at each sensing element is from a small area on the foreground surface, as shown by the red square in Fig. 2.1 (a). Therefore, the pixel value at that sensing element is only corresponding to a very small part of the foreground. In this case, there is less color mixture if the resolution of the sensing elements (i.e., pixel) is high. On the contrary, color mixture becomes severer in images with lower resolution. In Fig. 2.1 (b), it shows the case that the foreground object is not on the focus plane, therefore the emitted light from one small part of the foreground object goes to multiple sensing elements in the sensor (i.e., red squares in Fig. 2.1 (b)). In this case, the light (i.e., color) of the foreground object spread on the image, and results in color mixture.



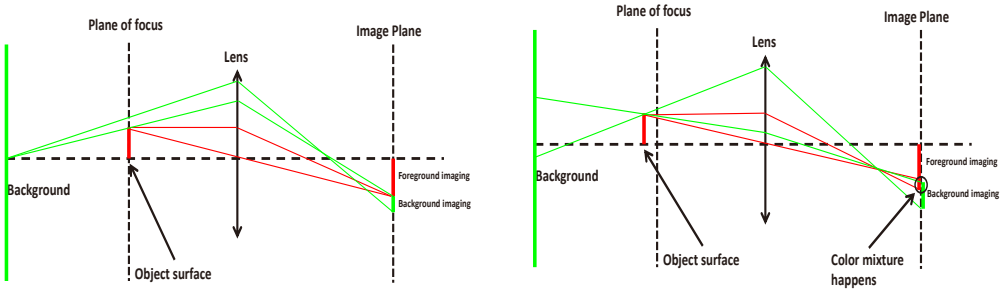
(a) Ideal situation where the object is perfectly focused and the diffraction is ignored. In this case, each small part on the foreground object only stimulate one element of the imaging sensor, which is regarded as one pixel.

(b) Defocus situation where the object is defocused and the diffraction is ignored. In this case, each small part on the foreground object will stimulate multiple elements of the imaging sensor, which are regarded as multiple pixels.

Figure 2.1: A simplified example of the focus and defocus situation of the foreground object.

After the discussion of foreground focus/defocus, we take into account the foreground/background together as shown in Fig. 2.2. In the case that the foreground is perfectly focused, the extreme light path is plotted in Fig. 2.2 (a) from the foreground to the lens aperture, and from the background to the foreground boundary. It can be observed that the foreground imaging and the background imaging does not overlap. On the contrary, the case that the foreground object is out of focus is presented in Fig.

2.2 (b). Due to the defocus light spreading, the foreground imaging and background imaging are partially overlapped, therefore causing the color mixture.



(a) Ideal situation that the object is perfectly focused and the diffraction is ignored. In this case, the background imaging is not overlapped with the foreground.

(b) Defocus situation that the object is defocused and the diffraction is ignored. In this case, the background imaging is partially overlapped with the foreground.

Figure 2.2: The foreground-background color mixture caused by defocus.

In order to mathematically model the color mixture caused by defocus, the reverse ray-tracing used in [22] is introduced here as shown in Fig. 2.3. In this figure, S_F and S_B represent the surface of the foreground and background respectively. Function β_F and β_B represent the foreground transparency and the transparency effect on background, respectively.

As illustrated in this figure, only the light from the objects (including foreground and background) in the cone behind point $X_{I'}$ can go through this point and be converged by the lens at the point X_I on the image plane. The total light power arrived at X_I determines the local pixel intensity in the digital image. M_F and M_B are two masking functions indicate the portions of foreground and background that are in the cone respectively. For the points on foreground/background in the cone, M_F and M_B are equal to 1, otherwise they are equal to 0. Vector $\mathbf{n}$ is the normal vector orthogonal to the local background surface. Vector $\mathbf{z}$ is a unit vector parallel with the optical axis of the lens.

In order to determine the pixel intensity at X_I, the received power at X_I needs to be modeled and accumulated. In [23], the spectral radiance of light is defined as

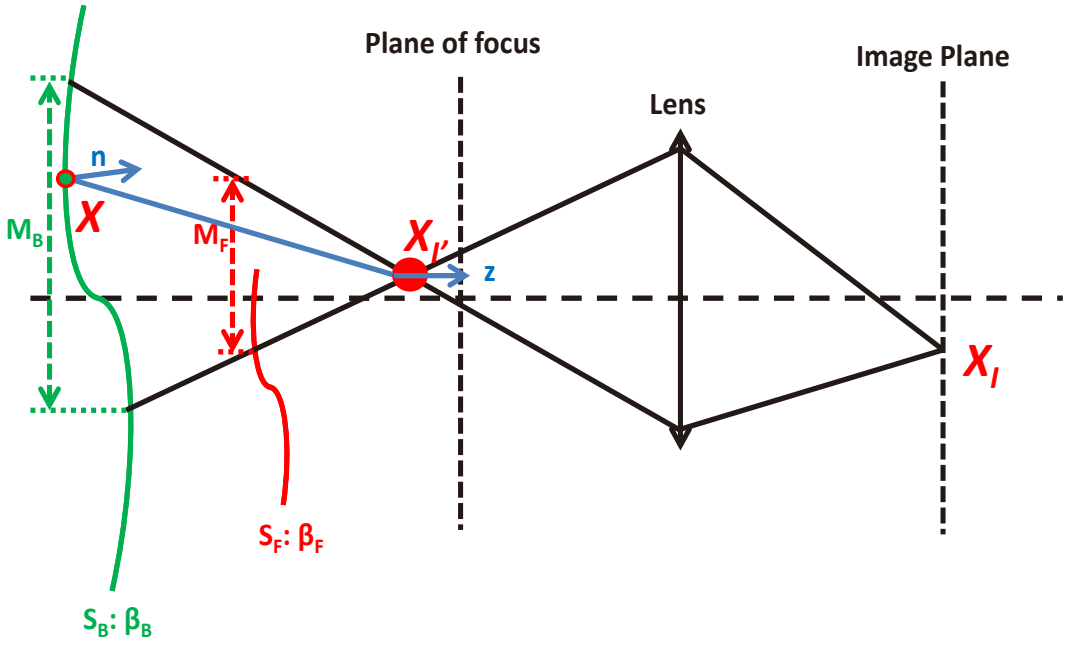


Figure 2.3: Reverse ray-tracing.

the power per projected area per solid angle:

$$L(\lambda, \omega, \boldsymbol{x}) = \frac{d^2 P(\lambda, \omega, \boldsymbol{x})}{d\omega dA^{\perp}}, \tag{2.2}$$

where $P(\lambda, \omega, \boldsymbol{x})$ is the power in direction ω, λ is the light wavelength, $\boldsymbol{x}$ is the position of a point on the object surface, $d\omega$ is the differential solid angle, and $dA^{\perp}$ is the differential area perpendicular to the power direction ω.

We also assume that the light that passes through point $X_{I'}$ does not attenuate before it arrives at point X_I. Therefore, the received power at X_I is equivalent to the incoming power at $X_{I'}$. Given the definition of spectral radiance, the received power $P_{X_I}(\Delta_{X_I})$ at the differential area of X_I can be integrated as follows:

$$P_{X_I}(\Delta_{X_I}) = \int_C \int_\Omega L(\lambda, \omega, \boldsymbol{x}) < \mathbf{n}, \mathbf{l} > d\omega dx, \tag{2.3}$$

where Δ represents the differential area element, C is the foreground/background

surface behind the cone of X_I', Ω is the solid angle from location x to the differential area $\Delta_{X_I'}$, $\mathbf{n}$ is the normal vector orthogonal to the object surface at x, $\mathbf{l}$ is the direction from x to X_I':

$$\boldsymbol{l} = \frac{\boldsymbol{X_{I'}} - \boldsymbol{x}}{\|\boldsymbol{X_{I'}} - \boldsymbol{x}\|}\,. \tag{2.4}$$

According to [22], Equ. (2.3) can be derived and rewritten in the form of Equ. (2.5):

$$I(\lambda) = m^2 \int_C L(\lambda, \boldsymbol{l}, \boldsymbol{x}) \frac{< \boldsymbol{n}, \boldsymbol{l} >< \boldsymbol{z}, \boldsymbol{l} >}{\|\boldsymbol{X_I'} - \boldsymbol{x}\|} d\boldsymbol{x}\,, \tag{2.5}$$

where

$$m = \frac{f}{f - v}\,, \tag{2.6}$$

and f is the focal length, v is the image distance.

After the total power arrives at point X_I is formulated by Equ. (2.5), it can be further divided into two parts according to [22]: the power from foreground and the power from background, as presented in Equ. (2.7):

$$\begin{aligned} I(\lambda) = m^2 \Bigg\{ & \int_{S_F} M_F(\boldsymbol{x})(1 - \beta_F(\boldsymbol{x})) L_F(\lambda, \boldsymbol{x}) \frac{< \boldsymbol{n}, \boldsymbol{l} >< \boldsymbol{z}, \boldsymbol{l} >}{\|\boldsymbol{X_I'} - \boldsymbol{x}\|} d\boldsymbol{x} \\ & + \int_{S_B} M_B(\boldsymbol{x}) \beta_B(\boldsymbol{x}) L_B(\lambda, \boldsymbol{x}) \frac{< \boldsymbol{n}, \boldsymbol{l} >< \boldsymbol{z}, \boldsymbol{l} >}{\|\boldsymbol{X_I'} - \boldsymbol{x}\|} d\boldsymbol{x} \Bigg\}\,, \end{aligned} \tag{2.7}$$

where S_F and S_B are the set of points on the foreground and background surfaces respectively; $\beta_F(\boldsymbol{x})$ and $\beta_B(\boldsymbol{x}) \in [0, 1]$, represent the foreground/background transparency properties, the relation of which is presented by Equ. (2.8):

$$\beta_F(\boldsymbol{x_F}) = \beta_B(\boldsymbol{x_B})\,, \tag{2.8}$$

where $\boldsymbol{x_F}$ and $\boldsymbol{x_B}$ are the points on foreground and background surfaces; and the point positions satisfy the constraint that $\boldsymbol{x_F} = \gamma \boldsymbol{x_B} + (1-\gamma)\boldsymbol{x_{I'}}$ with $\gamma \in [0, 1]$.

Since the pixel intensity on the image is determined by the received light, the derived power function in Equ. (2.7) can be used to represent the pixel intensity. According to [22], this power function can be rewritten into similar form with Equ. (2.1) if the following conditions and assumptions are met.

- The foreground/background surface has the property of Lambertian reflection, which shows similar brightness to an observer regardless of the viewing angle (i.e., $L(\lambda, \omega, \boldsymbol{x}) = L(\lambda, \boldsymbol{x})$).
- The surfaces of the foreground and background are parallel to the image plane with the distances of u_F and u_B.
- The spectral radiance varies little on C_F and C_B, which infers that $L_F(\lambda, \boldsymbol{x}) \approx L_F(\lambda, C_F)$, and $L_B(\lambda, \boldsymbol{x}) \approx L_B(\lambda, C_B)$.

With these assumption, Equ. (2.7) can be rewritten as follows:

$$I(\lambda) = \alpha F(\lambda) + (1-\alpha)B(\lambda)\,, \tag{2.9}$$

where

$$F(\lambda) = \frac{m^2 \sigma(C_F)}{(u_F + mv)^2} L_F(\lambda, C_F)\,, \tag{2.10}$$

$$B(\lambda) = \frac{m^2 \sigma(C_B)}{(u_B + mv)^2} L_B(\lambda, C_B)\,, \tag{2.11}$$

$$\alpha = \begin{cases} \dfrac{\sigma(S_F \cap C_F)}{\sigma(C_F)}, \text{foreground is opaque} \\ \dfrac{\sigma(S_F \cap C_F)}{\sigma(C_F)}(1-\beta_F), \text{foreground is partially transparent} \end{cases}, \tag{2.12}$$

and C_F represents the foreground region behind the cone of $X_{I'}$ in Fig. 2.3, $\sigma(\bullet)$ denotes the area of its augment.

According to Equ. (2.1), Equ. (2.7), and Equ. (2.9), it can be approved that the linear α blending model, which is used in this thesis and as well as many other image matting algorithms, has solid physical ground.

2.2 Matting problem in computer graphics

In the computer graphics, it is not an efficient strategy to render an entire image in a single program since different image elements (i.e., the foreground objects, and the background) may need different rendering strategies. In addition, the render errors in one image element should not affect the rendering results of other elements. The compositing problem arises when it is required to separate the entire image into multiple elements that can be rendered independently. In image compositing, different image elements are combined together in one single image while soft edges, semi-transparency, fine details, and relative postilions are all well kept without introducing aliasing.

A comprehensive and detailed description about alpha based image compositing was proposed in the landmark paper by Porter and Duff in 1984 [3]. The concept of integral alpha channel was first invented by Ed Catmull and Alvy Ray Smith in late 1977, and then it was fully developed by Thomas Porter and Tom Duff. The values in α channel indicate the coverage extent of image elements at each pixel. In the work of [3], vectors called "quadruple (r, g, ,b, α)" are given to the elements in an image at every pixel. The first three values in quadruple are the R, G, B colors of the image element, the fourth value, which is called α, indicates in what extent the current pixel is covered by this element – fully covered, half covered, or quartered covered. If we want to represent a pixel that is fully covered by a red object, a quadruple value with (1,0,0,1) is used. Meanwhile, a quadruple with the value of (1, 0, 0, 0.5) is used to represent that this pixel is only half covered by a red object, and this may happen

at object boundary or semi-transparency region.

With the usage of α channel, we can analytically represent the pixel color that is affected by multiple image elements. Since we are talking about the joint color appearance of multiple elements, it is desirable to consider the relative positions between these elements. In [3], the relative positions are summarized into the following cases: over, in, out, atop, and xor.

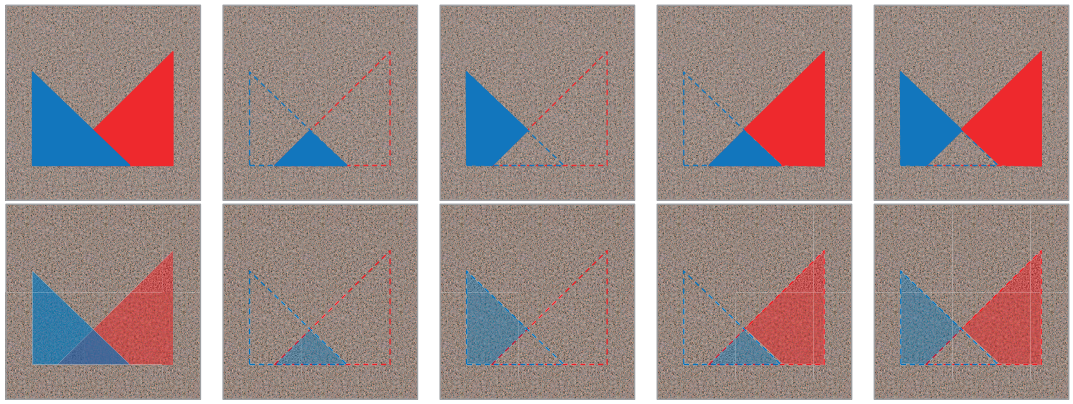
- Over: "A over B" represents the placement that A is in front of B.
- In: "A in B" represents the part of A that is inside of B when A is over B.
- Out: "A out of B" represents the part of A that is outside of B.
- Atop: "A atop B" represents the union part of "A in B" and "B out of A" when A is over B.
- XOR: "A xor B" represents the union part of "A out of B" and "B out of A" when A is over B.

An example of these relative positions is given in Fig. 2.4. The blue triangle represents an element A, and the red triangle represents an element B. The background is white noisy so that the transparency effect is easier to be observed. The elements A and B are both opaque in the first row of Fig. 2.4, while they are both semi-transparent in the second row.

Given the α channels and the relative positions of each image element, the compositing color of each image pixel can be calculated as follows:

$$I_{(i,j)} = \frac{\alpha_{A_{(i,j)}} I_{A_{(i,j)}} + \alpha_{B_{(i,j)}}(1-\alpha_{A_{(i,j)}}) I_{B_{(i,j)}}}{\alpha_{A_{(i,j)}} + \alpha_{B_{(i,j)}}(1-\alpha_{A_{(i,j)}})}\,, \tag{2.13}$$

where (i,j) refers to the pixel coordinate, $I_{A_{(i,j)}}$ and $I_{B_{(i,j)}}$ are the intrinsic colors of element A and B, and $\alpha_{A_{(i,j)}}$ and $\alpha_{B_{(i,j)}}$ are the values in α channels of element A and B, and $I_{(i,j)}$ is the final compositing color if the relative position is A-over-B.



(a) A over B (b) A in B (c) A out B (d) A atop B (e) A xor B

Figure 2.4: The object relative positions in digital matting [3].

The conventional alpha matting and chroma keying problems are special cases in general image compositing because there are only two image elements to be considered (i.e., foreground and background). Their relative position is "foreground over background". In this case, Equ. (2.13) can be rewritten into the standard α blending equation:

$$I_{(i,j)} = \alpha_{A_{(i,j)}} I_{A_{(i,j)}} + (1 - \alpha_{A_{(i,j)}}) I_{B_{(i,j)}} , \tag{2.14}$$

where $I_{A_{(i,j)}}$ is the foreground color, and $I_{B_{(i,j)}}$ is the background color.

Although the matting problem described in Equ.(2.1) and Equ. (2.14) is a simplified case of multiple layer image compositing, foreground-background composition is the basic problem which multiple layer compositing can be decomposed to. Therefore, we focus on the solution of foreground-background matting problem described by Equ.(2.1) in this thesis.

2.3 Human visual system and color spaces

In human visual system, the color sensation is initiated by a specialized type of retinal neuron, which is known as the photoreceptor cell. It is known that there are two types of photoreceptor cells directly contributing to human sight: rod cells,

and cone cells [4] [24] [25]. The rod cells mainly locate at the edges of the retina. They are responsible for the night vision because this type of cell is sensitive in dark condition and does not respond in bright condition. On the other hand, the cone cells mainly locate at the macula of retina. They are responsible for the color vision in relatively light condition. In addition, there are three types of cones which have different reaction wavelength ranges:

- S-Cone: S is the abbreviation for short, this type of cone is sensitive to short wavelength light (400-500 nm).
- M-Cone: M is the abbreviation for medium, this type of cone is sensitive to medium wavelength light (450-630 nm).
- L-Cone: S is the abbreviation for long, this type of cone is sensitive to long wavelength light (500-700 nm).

With (S-, M-, L-) cones, human eyes have different reactions to light with different energy distributions. This is also the reason that we can perceive colors. The reaction from S-Cones can be regarded as how much "red" is perceived because the wavelength of "visually red" light is normally between 625-740 nm; the reaction from M-Cones can be regarded as how much "green" is perceived because the wavelength of "visually green" light is normally between 525-565 nm; and the reaction from L-Cones can be regarded as how much "blue" is perceived because the wavelength of "visually blue" light is normally between 435-500 nm [26]. Fig. 2.5 illustrates the reaction functions of different photoreceptor cells for the light with different wavelength.

We already know that the human color sensation is the result from three different cone cells' reactions. In this case, it is an intuitive way to represent arbitrary color by using three base colors, such as red, green, and blue used in RGB color space. One general expression for this color representation is presented in Equ. (2.15):

$$[C] = C_1[P_1] + C_2[P_2] + C_3[P_3]\,, \tag{2.15}$$

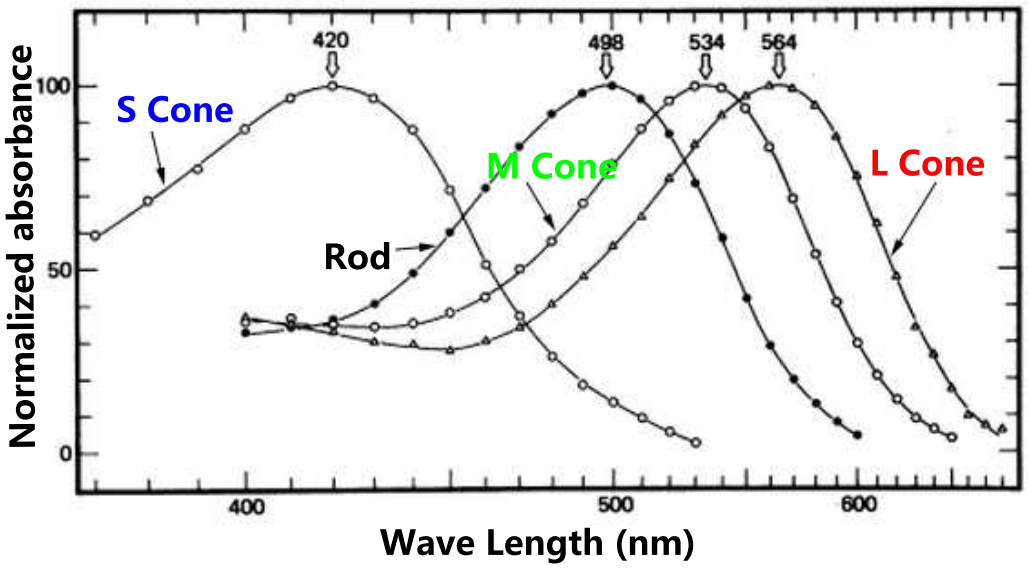


Figure 2.5: The normalized mean absorbance spectra of four types of human photoreceptors. The numbers at each curve represent the wavelength at which the photoreceptor has the peak response. The data and figure are from the work of [4].

where [C] is the color to be represented; $[P_1]$, $[P_2]$, and $[P_3]$ are the base colors, which are also called as primaries; C_1, C_2, and C_3 are called as tristimulus values, which represent how much the corresponding base colors contribute to the observed color [C].

A color space described by Equ. (2.15) is a linear color space, in which the sum of two colors can be calculated by summing up the associated tristimulus values. The tristimulus values of a color in any two linear color spaces can be transformed by using a 3 by 3 matrix. For example, the tristimulus values of a color in sRGB space and in CIE-XYZ space can be directly transformed by using the following transform matrix:

$$\begin{bmatrix} C_R \\ C_G \\ C_B \end{bmatrix} = \begin{bmatrix} 0.4184 & -0.1587 & -0.0828 \\ -0.0912 & 0.2524 & 0.0157 \\ 0.0009 & -0.0026 & 0.1786 \end{bmatrix} \begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix}, \tag{2.16}$$

where (C_R, C_G, C_B) are the tristimulus values of a color in sRGB color space, and (C_X, C_Y, C_Z) are the tristimulus values of the same color in CIE-XYZ color space.

Although the RGB based color spaces are consistent with the underlying mechanisms of human color perception, there are two drawbacks which introduce difficulties in computer vision problems:

- The Euclidean distance in RGB color space can not isotropically and evenly represent the color difference in the sense of human. This problem makes it difficult to design robust metrics to separate different colors or to group similar colors in an image.

- Neither RGB nor XYZ color space separates the lightness from the chrominance. Such color descriptions does not follow the way that human beings describe the color because we normally describe the colors by using the color name (e.g., red, cyan, etc.) and the color lightness (i.e., bright and dark).

In order to simulate the human color sensation and to avoid the aforementioned two drawbacks, many HVS (human visual system) based color spaces are proposed. Generally, HVS based color spaces are often derived from sRGB or CIE-XYZ color space [27] by using linear or non-linear transformations. After the transformation, the lightness of the color can be separated from the chrominance by using an independent scale value. Meanwhile, two coordinates are used to represent the chrominance of the color on a color plane. These two coordinates can be pure numeric index on a color plane which is carefully designed to be perceptually uniform, such as YC_bC_r, YUV, $CIE\text{-}Lab$.

The transformation between RGB color space and YC_bC_r color space that is used in HDTV is shown as follows:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.183 & 0.614 & 0.062 \\ -0.101 & -0.339 & 0.439 \\ 0.439 & -0.399 & -0.040 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} . \tag{2.17}$$

The transformation between RGB color space and YUV color space that is used in PAL and NTSC video is shown in Equ. (2.18):

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} . \tag{2.18}$$

The transformation between XYZ color space and $CIE-Lab$ color space is shown as follows:

$$\begin{aligned} L &= 116 f(Y/Y_n) - 16 \,, \\ a &= 500[f(X/X_n) - f(Y/Y_n)] \,, \\ b &= 200[f(Y/Y_n) - f(Z/Z_n)] \,, \end{aligned} \tag{2.19}$$

where

$$\begin{aligned} f(t) &= t^{(1/3)}, \text{ when } t > 0.008856 \,, \\ f(t) &= 7.787t + 16/116, \text{ otherwise} \,. \end{aligned} \tag{2.20}$$

On the other hand, chrominance coordinates can be specifically defined to be closely related to human color sensation in spaces like HSL, HSV, and $CIE-CAM02$. In such representation model, two terms are often used to code the color chrominance: **hue** and **saturation**. The hue can be regarded as the color name, which is often a degree to which a color can be described as different from or similar to

colors such as red, green, blue. Besides the color name (**hue**), the color chrominance also depends on the colorfulness which is often represented by the saturation. For example, the color blue can be further categorized as light blue, pastel blue, and vivid blue. By using hue and saturation together, the chrominance of a color can then be represented according to human color sensation. Take the **HSV** color space for example, it is one basic color model that is widely used to separate lightness from chrominance. The color coordinates in HSV space can be derived from the color coordinates in RGB space by using the transformation functions in Equ. (2.21), Equ. (2.22), and Equ. (2.23):

$$H = \begin{cases} 0, & \text{if } C = 0\,, \\ 60^\circ \times \Big(\dfrac{G-B}{C} \bmod 6\Big), & \text{if } M = R\,, \\ 60^\circ \times \Big(\dfrac{B-R}{C} \bmod 6 + 2\Big), & \text{if } M = G\,, \\ 60^\circ \times \Big(\dfrac{R-G}{C} \bmod 6 + 4\Big), & \text{if } M = B\,, \end{cases} \tag{2.21}$$

$$S = \begin{cases} 0, & \text{if } C = 0\,, \\ \dfrac{C}{V}, & \text{otherwise}\,, \end{cases} \tag{2.22}$$

$$V = M\,, \tag{2.23}$$

where

$$\begin{aligned} M &= \max(R, G, B)\,, \\ m &= \min(R, G, B)\,, \\ C &= M - m\,. \end{aligned} \tag{2.24}$$

In this thesis, the global statistics about colors in a natural image will be analyzed in the HVS (human visual system) based color spaces. Based on statistical data,

Gaussian models will be used to reveal the underlying color distribution. Depending on the color distribution, we will propose our solution to the aforementioned matting problem. Before presenting our matting proposal, the existing matting solutions will be reviewed in the next chapter.

3

Literature review

Image/video matting refers to the problem of accurately extracting interested elements in the image or video sequence. The extracted elements can be further composited into other images or video sequences. The image matting and the image compositing are two inverse problems and they are playing important roles in many image and video editing applications. The matting problem would turn into the most fundamental case if there was only one interested element in the original image/video. In this case, the interested element is also called the foreground object. As we introduced in Chapter 1.3, this matting problem can be regarded as solving color vectors $F_{(i,j)}$, $B_{(i,j)}$ and blending factor $\alpha_{(i,j)}$ from Equ. (1.1). If we rewrite Equ. (1.1) into

the matrix form

$$\begin{bmatrix} C_R \\ C_G \\ C_B \end{bmatrix} = \alpha \begin{bmatrix} F_R \\ F_G \\ F_B \end{bmatrix} + (1-\alpha) \begin{bmatrix} B_R \\ B_G \\ B_B \end{bmatrix}, \tag{3.1}$$

it is easy to see that there are 7 unknowns to be solved from just one equation. Therefore, matting problem is inherently under constrained, and it has no unique solution.

In the previous matting approaches, this under constrained problem is further categorized into two cases: 1) the natural image matting [28] that the foreground is extracted from complex background with a manual tag map (i.e. trimap or scribbles); 2) the blue screen matting [29] that the foreground is extracted from monochromatic background. These two matting approaches have different advantages and disadvantages. The natural image matting has fewer requirements on image content but it often needs an extra manual input that initially labels the major parts of the foreground and the background. The blue screen matting requires less human intervention but special environment setup is required to guarantee that the background color is as uniform as possible.

Depending on the applications, the matting problems can also be classified by other means, such as video matting [30] [31], shadow matting [32] [33], and environment matting [34] [35].

In this thesis, we organize the literature review based on blue screen matting and the natural image matting. In the following sections, we will overview the previous works on matting problems.

3.1 Blue screen matting

Since video compositing is a compelling technique in film industry, simple and efficient matting techniques have been researched for decades. It is easy to understand that

the foreground extraction would be easier from simpler background. In this case, the early researches mainly focused on blue screen matting problem, in which the foreground object is recorded in front of a solid colored background. The term "solid" means that the chrominance is unique and the lightness does not change very much on the background. Although the term "blue screen" is used here, it does not infer that the background color can be only blue. Actually, the background color can be chosen differently based on the applications and the techniques. The time-line of blue screen matting development is presented in Fig. 3.1. In a blue screen matting system, the background color is the most important side information that is used for foreground-background separation. Therefore, blue screen matting is often called as **chroma keying**. In this thesis, we do not differentiate blue screen matting and chroma keying. Since chroma keying is mostly developed for commercial usage such as film making and TV-broadcasting, this technique is often patent protected. We will introduce the most important and classic patents and techniques in the following section.

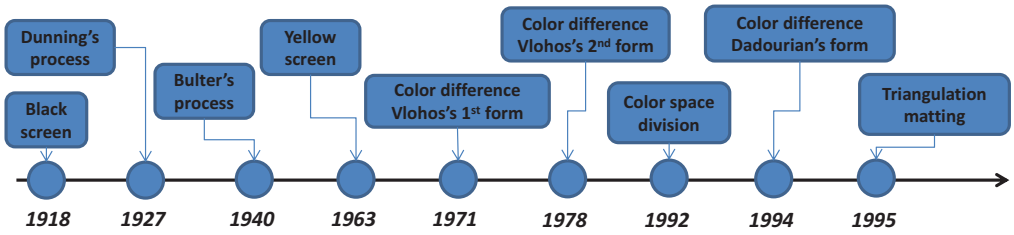


Figure 3.1: Time-line of the techniques for blue screen matting.

3.1.1 Early approaches of blue screen matting

In the early years of film industry, the film was often recorded in analog ways such as negative film. In this case, digital processing, which is widely known and used today, was not a choice back then. Therefore, many physical methods were investigated and

used for matting problems. As we introduced in Chapter 1.1, the earliest matting solution was made by placing a black glass (i.e. matte) in front of the cameras so that only part of the film would get exposed. However, the matte in this solution is **fixed** in front of the camera and the foreground object to be extracted should not move across the fixed matte boundary – the "hopefully" invisible boundary line between the matte painting and the live foreground.

In order to make the matte be capable of moving with the live foreground, the black matting process was invented and patented by F. D. Williams in 1918 [36]. This is also the first time that the color (actually the luminance) is used for the foreground-background separation. In Williams's process, the foreground objects were photographed against a pure black background. Then the film would be copied to increasingly high light sensitive negative films until a black and white segmentation map emerged. The black matting process used two assumptions for the matting solution: 1). the dark background would not make the negative film exposed; 2). the light from the foreground would eventually make the negative film over exposed if the negative was sensitive enough to the incoming light. The obtained segmentation map is the cast of foreground object, therefore it can move along with the live actions. Due to the fact that the matte can move with the foreground, this method is the first place where the term "traveling matte" comes from. However, the shadows on the foreground object would get lost because of the lack of lightness level.

In 1927, an alternative matting solution was proposed by C. D. Dunning [37]. In Dunning's process, color light sources were used to lighten the background and foreground into different colors. For example, the background is lighten to be blue, and the foreground actors are lighten to be yellow. Given the differently colored foreground and background, a color filter can be used to split the light from foreground and background. Therefore, a traveling matte can be generated. The major problem of Dunning's process was that this method can only be applied to black and white films. This is because that the foreground color was distorted when the colored lights

was applied.

In order to deal with matting problems in color films, Petro Vlahos began his matting work from 1950s, and eventually became to one of the giants in the world of compositing for his tremendous achievements [5] [38] [39] [40] [41]. In Vlahos's early matting solution, the Sodium vapor process [5] was proposed and extensively used by the Walt Disney Studios in the 1960s and 1970s. In this matting method, the actors are required to stand in front of a white screen which was lit by power Sodium vapor lights. The Sodium vapor light is an orange light source with very specific wavelength and narrow bandwidth–averaging 589.6 nanometers. By using a specifically designed selective light divider, the light from the background with wavelength near 589.6 nanometer is reflected to one film recorder while the light with other wavelength (i.e. the light from the foreground) passes through this light divider to another film recorder. In this case, the foreground and background lights can be physically separated. The reflection function of the selective light divider and the energy distribution of Sodium vapor light are presented in Fig. 3.2 from Vlahos's patent. It can be observed that the selective light divider has another reflection band at 475 nanometer in the blue color range. In this case, the light from the foreground with wavelength near 475 nanometer will get a penalty in this Vlahos's proposal.

Up to now, the matting processing was mainly based on physical approaches. In the following section, the difference matting will be introduced. This matting approach can be regarded as the beginning that matting problem was mathematically analyzed and modeled.

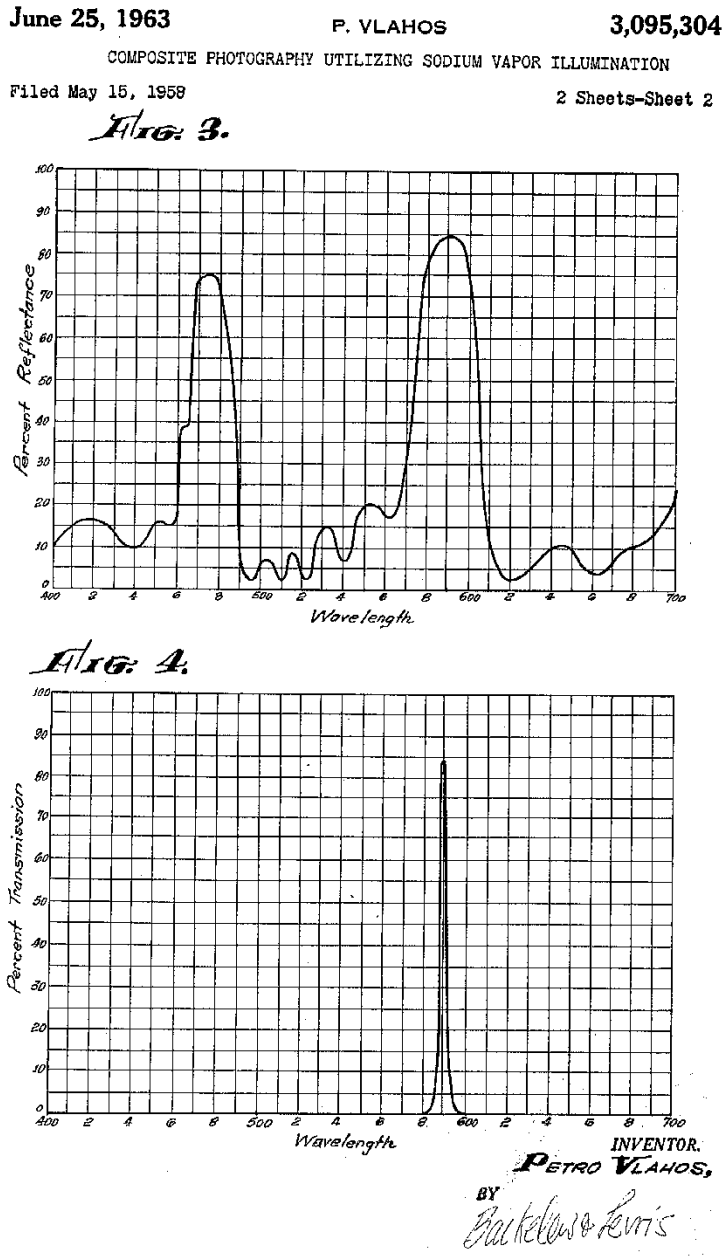


Figure 3.2: The reflection curve of selective light divider (top figure), and the energy curve of Sodium vapor light (bottom figure). These two figures are from the patent of Vlahos [5].

3.1.2 Difference matting

As we mentioned in Chapter 3.1.1, the Sodium vapor process performed well and was extensively used by Disney for years. However, only one selective light divider was ever made. In this case, only one camera, which is owned by Disney, can do this process. Vlahos then proposed difference matting which can be used and applied in wider range. In this matting proposal, the color of the observed object is modeled by the following equation:

$$C_0 = C_f + (1 - \alpha_0)C_b \,, \tag{3.2}$$

where C_0 is the observed color, C_f is the color of foreground object, and C_b is the color of background.

Note that this equation is slightly different from the conventional α matting equation we use today (Equ. (1.1)). Actually, these two equations are consistent if we refer to C_f in Equ. (3.2) as the composited foreground, which is already multiplied by α.

In Vlahos' new matting method, he assumed that the RGB colors on the foreground object would obey the following relation:

$$B \leq aG \,, \tag{3.3}$$

where B and G represent the values in blue channel and green channel for each pixel; a is a variable which usually ranges from 0.5 to 1.5 [41].

Based on this assumption, Vlahos proposed his new alpha estimation equation

$$\alpha_0 = \max(\min(1 - a_1(B_0 - a_2 G_0), 1), 0) \,, \tag{3.4}$$

where B_0 and G_0 are the blue and green components of a pixel. In practice, the B_0

is suggested to be replaced by $\min(B_0, B_k)$, where B_k is the minimum value in the blue channel of the whole background region.

After the α value is estimated, the foreground color is further refined by adjusting blue channel intensity:

$$\begin{aligned} R_f &= R_f \,, \\ G_f &= G_f \,, \\ B_f &= \min(B_f, a_2 G_f) \,. \end{aligned} \tag{3.5}$$

In 1978, Vlahos proposed his second version of α estimation [41]:

$$\alpha_0 = 1 - a_1(B_0 - a_2(a_5 \max(r, g) + (1 - a_5)\min(r, g))) \,, \tag{3.6}$$

where $r = a_3 R_0$, and $g = a_4 G_0$; R_0, G_0, and B_0 are the red, green, blue components of a pixel; a_i are tunable parameters.

In addition to Vlahos's efforts on color difference matting, further refinements on difference matting have been proposed. One recent published work is as shown by Equ. (3.7) [42]:

$$\alpha_0 = 1 - ((B_0 - a_1) - a_2 \max(r, g) - \max(a_5(R_0 - G_0), a_6(G_0 - R_0))) \,, \tag{3.7}$$

where the coefficients are the same as what we defined in Equ. (3.6).

It can be observed that the color difference matting involves tedious work on tuning multiple parameters. In addition, the equations used for color difference matting were derived based on years of experiments and experiences. However, the lack of firm mathematical and theoretical derivation is a major problem.

3.1.3 Polyhedron based matting

The global color distribution is also investigated for reliable chroma keying [43] [44]. In [43], two polyhedral approximations of spheres, which are centred at the average background color, are used to partition the RGB color space. One polyhedron sphere (sphere A) is defined with the assumption that it covers all background pixel colors with the shortest radius. The other polyhedron sphere (sphere B) is defined with the assumption that it does not cover any of the foreground pixel color with the longest radius. As shown in Fig. 3.4, pixel with color in sphere A (green dot in Fig. 3.4) is the background pixel ($\alpha = 0$); pixel with color outside of sphere B (red dot in Fig. 3.4) is the foreground color ($\alpha = 1$); pixel C with color located between sphere A and sphere B (blue dot in Fig. 3.4) is the edge or transparent pixel ($\alpha \in (0, 1)$). By drawing a straight line from the sphere centroid to pixel color C, the intersection points with sphere A and B are respectively regarded as the background color and the foreground color. In this case, the α value will be estimated with respect to the relative position of the background color and the foreground color. However, the assumption that the foreground and background colors can be separated by using spheres is not reliable for many images and videos. In practice, the colors in sRGB space are not easy to be grouped, especially when they are grouped by sphere clusters.

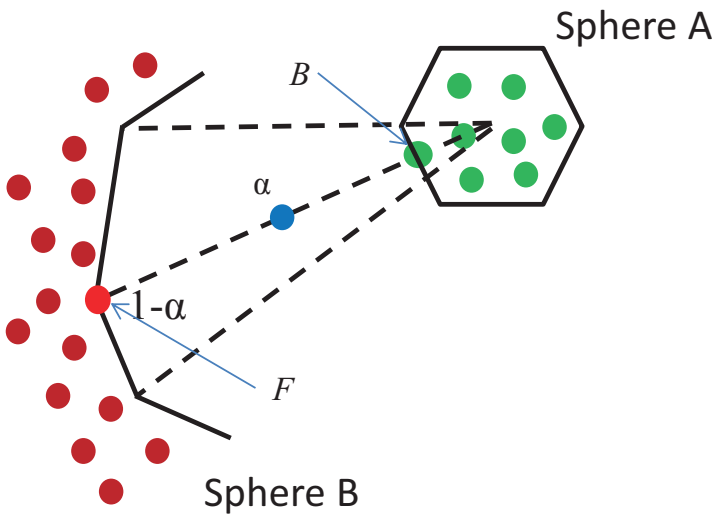


Figure 3.3: α estimation in Mishima's Polyhedron based matting.

3.2 Natural image matting

Besides the chroma keying techniques used for solid background, recent researches have been focused on **natural image matting**, which extracts foreground object from natural images with complex background. Natural image matting has wider application range compared with chroma keying because it is capable of dealing with arbitrary background. On the other hand, this advantage is usually achieved at the cost of heavy computation and lack of automaticity. In this case, both chroma keying and natural image matting have their own advantages and disadvantages, depending on the applications. Although this thesis is dealing with the problem of chroma keying, the methods of natural image matting are also reviewed because these two techniques are closely related and the ideas in natural image matting can be very helpful for chroma keying. The time-line of natural image matting development is presented in Fig. 3.1. The representative algorithms are listed with regard to the years when they were proposed. Generally, the natural image matting can be

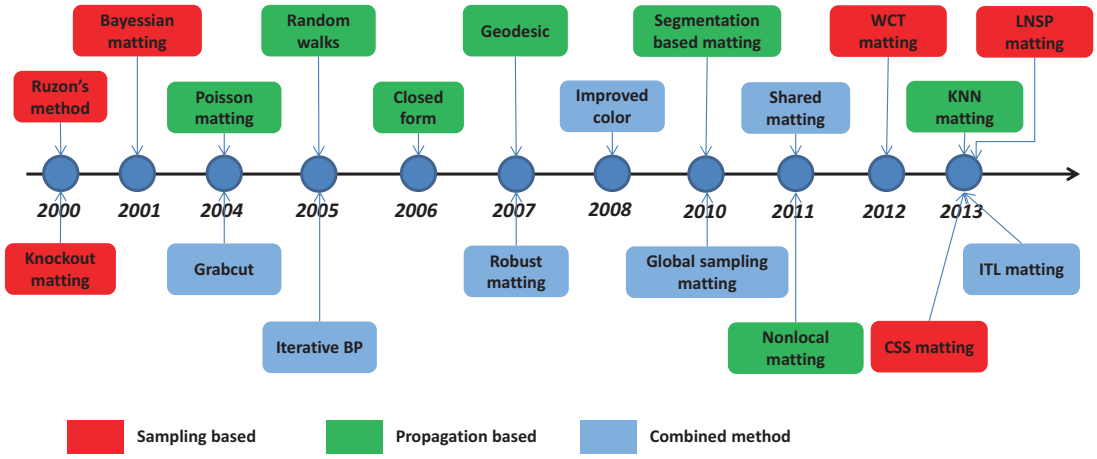


Figure 3.4: Time-line of the methods of natural image matting.

classified into three categories:

- Matting methods based on sampling.
- Matting methods based on propagation.
- Combination methods based on sampling and propagation.

The representative methods in each category will be introduced in the following sections.

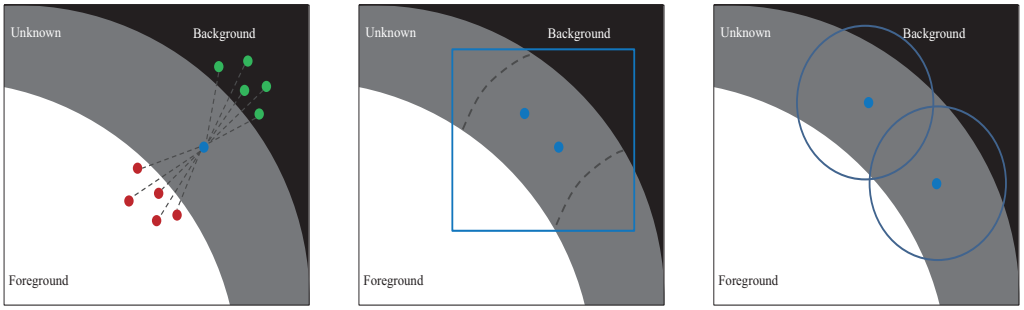
3.2.1 Sampling based natural image matting

As shown in Equ. (3.1), the solution of the matting problem is to find for each pixel the foreground color, background color, and the corresponding α value. In sampling based matting methods, the foreground color and the background color of a pixel are estimated by choosing samples from the original image. More specifically, the sampling strategies can be considered from two aspects:

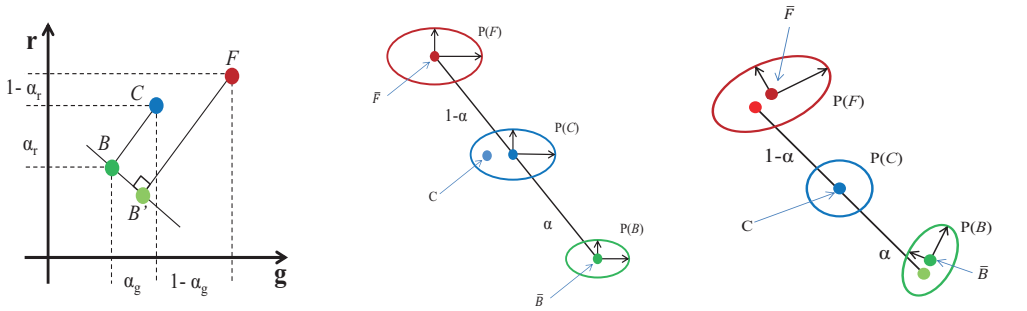
- How to estimate the foreground color, background color and the α value based on the samples collected from the image.

- How to choose reliable samples, and how to define the reliability of chosen samples.

In classical sampling based image matting, the foreground and background samples of pixel C are often locally collected from the region nearby C. As shown in Fig. 3.5, three representatives for early sampling based matting methods are presented. In these three methods, a trimap is manually specified to roughly indicate the foreground and background regions. As shown in Fig. 3.5 (a) (b) (c), the white/black regions are the predefined foreground/background regions while the gray region is the unknown region where F, B, and α need to be estimated for each pixel.



(a) Sampling strategy of Knockout [45] [46]
(b) Sampling strategy of Ruzon's method [47]
(c) Sampling strategy of Bayesian matting [48]



(d) Alpha estimation in Knockout [45] [46]
(e) Alpha estimation in Ruzon's method [47]
(f) Alpha estimation in Bayesian matting [48]

Figure 3.5: Classic sampling based natural image matting.

Given a pixel C in unknown region, Knockout method [45] [46] collected the foreground and background samples from the spatial neighbors of pixel C according to the trimap. The foreground color F is estimated by calculating the weighted sum. The weights used here are proportional to the spatial distances between the samples

and pixel C. The initial background color B is also estimated by the same way. In [46], this initial background color is further refined to B' as shown in Fig. 3.5 (d). With the estimated foreground and background colors, the α values in each channel can be calculated by Equ. (3.8):

$$\alpha_i = \frac{C_i - B'_i}{F_i - B'_i}, \tag{3.8}$$

where $i \in \{ R, G, B\}$ represents R/G/B channel, respectively. Afterwards, the final α can be calculated by averaging the α values in every channels.

In Ruzon's matting method [47], the foreground and background samples are collected in a rectangular window enclosing three categories of pixels: foreground pixels GF, background pixels GB, and unknown pixels GU. The pixel colors in GF and GB are modelled by using Gaussian probability functions P(F) and P(B). As shown in Fig. 3.5 (e), the symmetric axis of the Gaussian functions are parallel to the coordinate axis. Given an unknown pixel C, its value is related to a probability function P(C), which is an intermediate Gaussian distribution between foreground Gaussian P(F) and background Gaussian P(B). In this case, the foreground color F_{op} and background color B_{op} of an unknown pixel C are approximated by the mean value of P(F) and P(B). The optimal alpha value α_{op} is the one that yields the highest probability of P(C'), where C' is the composited color by using F_{op}, B_{op}, and α_{op}.

In Bayesian matting [48], the foreground and background samples are collected by using a sliding window moving along the boundaries of the trimap. As shown in Fig. 3.5 (c), the blue circles represent the sliding window at different locations. Similar as [47], the collected foreground and background samples are used to model Gaussian functions. As shown in Fig. 3.5 (f), the symmetric axis of the fitted Gaussian does not have to be parallel to the coordinate axis. In this case, this Gaussian function is a more general one and can better fit the color distribution. In addition, the α matte is solved by using maximum a posteriori (MAP) technique based on a well-

defined Bayesian framework. Given an unknown pixel with color C, its corresponding alpha, foreground and background colors are estimated by maximizing the posterior probability:

$$\begin{aligned} &\arg\max_{F,B,\alpha} P(F,B,\alpha|C)) \\ =&\arg\max_{F,B,\alpha} P(C|F,B,\alpha)P(F)P(B)P(\alpha)/P(C) \\ \propto&\arg\max_{F,B,\alpha} L(C|F,B,\alpha)+L(F)+L(B)+L(\alpha)\,, \end{aligned} \tag{3.9}$$

where

$$L(F) = logP(F) = -\frac{1}{2}(F-\mu_F)^T\Sigma_F^{-1}(F-\mu_F)\,, \tag{3.10}$$

$$L(B) = logP(B) = -\frac{1}{2}(B-\mu_B)^T\Sigma_B^{-1}(B-\mu_B)\,, \tag{3.11}$$

and

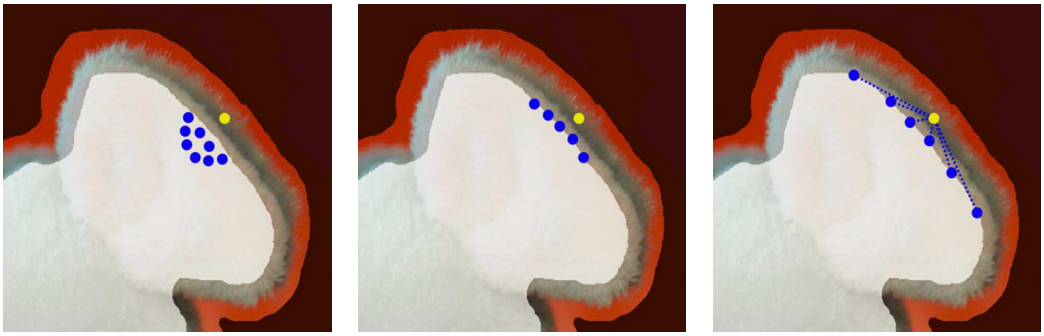
$$L(C|F,B,\alpha) = logP(C|F,B,\alpha) = \frac{-\|C-\alpha F-(1-\alpha)B\|^2}{\sigma^2}\,. \tag{3.12}$$

In Equ. (3.10), Equ. (3.11), and Equ. (3.12), μ is the mean value, σ is the local color variance, and Σ is the Gaussian covariance.

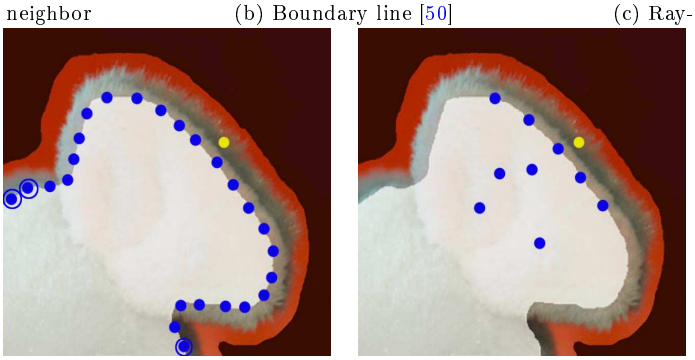
Based on Bayesian matting, the improved work is also proposed. In [49], the foreground and background color are no longer modelled by Gaussian function. Instead, the probability terms P(F), P(B), and P(α) are estimated based on the spatial distance and color difference. In this case, the estimation is more robust to sampling outliers.

However, these local sampling methods often fail to collect correct foreground/background color samples due to the limited sampling range. In this case, many recent researches

proposed different sampling strategies that collect comprehensive color information.



(a) Nearest neighbor (b) Boundary line [50] (c) Ray-based[51]



(d) Global [52] (e) Comprehensive [53]

Figure 3.6: Different sampling strategies.

In [50], Wang and Cohen pointed out that nearby searching (as shown by Fig. 3.6 (a)) may fail to collect samples that fit the matting equation. This is often caused by the complex color patterns in the natural image. In order to extend the searching range and collect more comprehensive color samples, Wang and Cohen proposed to collect color samples along the boundaries (Fig. 3.6 (b)) of the unknown region so that more colors can be collected. Based on Wang's work, Bai [54] [55] proposed to use the geodesic distance instead of Euclidean distance to find the nearby colors. The geodesic distance is defined as the path on a weighted graph from the unknown pixel to the boundary of the unknown region. The weights on the graph are calculated based on the probability that a pixel belongs the foreground or background region.

In order to further extend the sampling range, ray-based sampling was proposed in [51]. This method draws rays from the unknown pixel to different directions and collects foreground and background samples at the pixel locations where the rays

intersect with the region boundary. An example of such sampling method can be found in Fig. 3.6 (c). Although the rays can reach wider sampling range compared with Wang's method, ray-based sampling may still miss good samples due to the shape of the trimap [52].

In order to obtain comprehensive color sample, the global sampling was proposed in [52]. As shown in Fig. 3.6 (d), all pixel colors on the unknown region boundary are collected as the color samples in the global method. Therefore, a very comprehensive samples set is generated. The main problem in [52] is the huge computation cost introduced by large sample set. In order to efficiently pick out the best foreground-background sample pair, the generalized PatchMatch [56] algorithm is used to find the optimal sample pair.

The aforementioned sampling methods have two major problems:

- Only color information is considered. In this case, it is difficult to deal with the overlapped color distribution of foreground and background.
- The color samples are usually collected around the unknown region boundary. In this case, the shape of the trimap can significantly affect the sampling result.

In order to overcome these problems, more comprehensive sampling method was proposed based on weighted color and texture [53] [57] [58] [59]. In this method, the color samples are collected in global and local ranges at the same time. In addition, the samples are collected inside the foreground and background regions instead of just collecting them on the trimap boundary. An example of this comprehensive color sampling can be found in Fig. 3.6 (e). Furthermore, the texture information is used as the complementary of color information to distinguish the foreground-background samples with similar colors. By considering the comprehensive color information and the texture information, this method currently produces the state-of-the-art matte result on the benchmark data set [19].

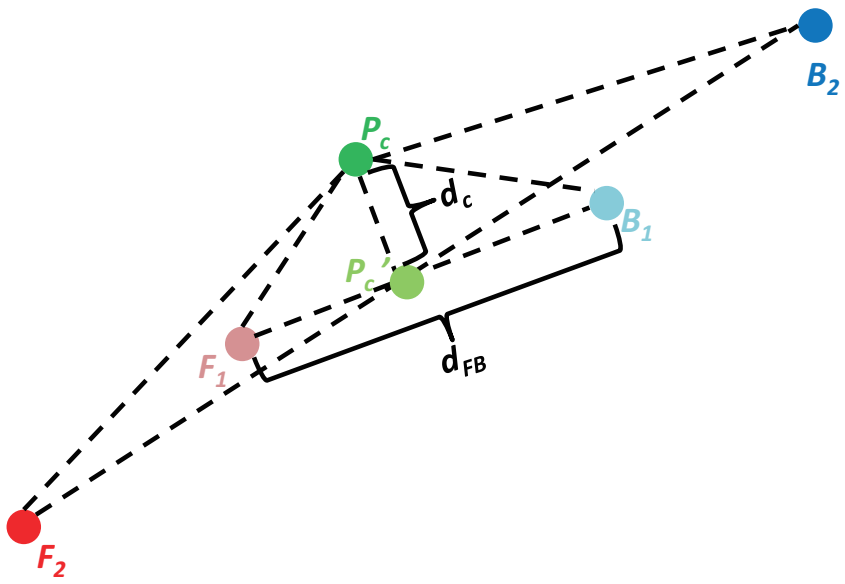


Figure 3.7: The linearity among foreground, background and unknown pixels' color.

Besides the sampling strategy which aims to provide comprehensive foreground-background sample sets, another problem is to find the most reliable foreground-background color pair in the sample set. In other words, it is required to robustly and convincingly pick out the most reliable foreground-background color pair for each unknown pixel from the collected color samples. In [50], Wang and Cohen proposed the confidence value $f(F, B)$ to evaluate the reliability of foreground-background sample pair according to the linearity ($R(F, B)$) and the color similarity ($w(F)$, $w(B)$) which are measured as follows:

$$R(F,B) = \frac{\|C - (\alpha F + (1-\alpha)B))\|}{\|F - B\|}, \tag{3.13}$$

$$w(F) = exp\left(-\frac{\|F - C\|^2}{D_F^2}\right), \tag{3.14}$$

$$w(B) = exp\left(-\frac{\|B - C\|^2}{D_B^2}\right), \tag{3.15}$$

and

$$f(F, B) = exp\left(\frac{-R(F, B)^2 \cdot w(F) \cdot w(B)}{\sigma^2}\right), \tag{3.16}$$

where σ is a tuning parameter, D_F and D_B are the minimum color distances from unknown pixel to foreground and background, respectively.

The sample pair with highest $f(F, B)$ will be chosen as the final estimation of foreground and background colors. The definition of $f(F, B)$ is based on the following two assumptions:

- Linear assumption: for an unknown pixel C, a good foreground and background color estimation would have good linear relation with the color of C according to their linear compositing equation (Equ. (3.1)). Take Fig. 3.7 as example, the sample pair (F_2, B_2) would be a better choice than (F_1, B_1) because points F_2, B_2, P_c have better linear relation.

- Color similarity assumption: for an unknown pixel C, a foreground sample with similar color to C would more likely be a reliable foreground sample; and the background color selection also obeys this assumption.

In conclusion, sampling based matting methods can provide intuitive and straightforward estimation of the foreground and background colors. Such method, however, can not work well if the image color becomes complex. Since most of the sampling methods collect samples around the trimap boundaries, the quality of trimap can significantly affect the final matting result. If only spatial and color information is used, sampling based methods are inherently inadequate when the foreground and background color distributions overlap.

3.2.2 Propagation based natural image matting

Because of the difficulties encountered by sampling based matting methods, many propagation based matting methods were proposed for natural images [60] [61] [62]. In propagation based methods, the affinities between local and nonlocal pixels are investigated to solve the matting equation. There are two most important problems in propagation based matting methods: 1) the definition of the affinity between image pixels; 2) the mathematic model used to propagate the α value from known region to unknown region. The classic and recent propagation based matting methods will be introduced as follows.

In [63], Poisson matting was proposed based on the assumption that foreground and background color change smoothly in neighboring pixels. Given this assumption, the gradient of the α map is proportional to the gradient of the original image. Mathematically, this relation can be formulated as follows:

$$\nabla \alpha_{(x,y)} \simeq \frac{1}{F_{(x,y)} - B_{(x,y)}} \nabla C_{(x,y)} \,, \tag{3.17}$$

where (x, y) is the image coordinate, ∇ is the gradient operator, C is the pixel color, F and B are foreground and background colors. In addition, this equation can be further formulated as

$$\Delta \alpha_{(x,y)} \simeq div \left(\frac{\nabla C}{F_{(x,y)} - B_{(x,y)}} \right) \,, \tag{3.18}$$

where Δ is the Laplacian operator, and div is the divergence operator.

In Poisson matting, Equ. (3.18) was solved by using Gauss-Seidel iteration with over-relaxation. In each iteration, the foreground color and background color are estimated by local sampling, which can not perform well in image with complex color texture. The improvement was proposed in [64]. This improved Poisson matting first applied a series of filters to extract the image textures, and used these textures to

guide the matting procedure. The original divergence based Poisson equation was reformulated into eigenvector based equation. The results in [64] showed that better matte can be generated in texture images compared with original Poisson matting.

In Poisson matting, it is assumed that the local foreground and background color vary smoothly. Compared with that, a looser assumption was made in closed form matting [65] [66]. it assumes that every foreground colors in a local image window can be represented by linear combinations of two constant colors. In other word, it infers that the foreground colors in a local image window lie on the same line in the RGB color space. In addition, it assumes that the local background colors obey the same assumption. Based on this assumption, the α value for each pixel can be calculated by

$$\alpha_i = \sum_c a^c I_i^c + b^c, \ \forall i \in w \,, \tag{3.19}$$

where i represents the image coordinate, c denotes the channel of RGB color space, w is a small local window in the image, a and b are two constants in window w.

Now the alpha matting turns into the problem of finding the optimal a, b, and α that minimize the energy function defined in the following equation:

$$J(\alpha, a, b) = \sum_{j \in I} \left(\sum_{i \in w_j} \left(\alpha_i - \sum_c a_j^c I_i^c - b_j^c \right)^2 + \varepsilon \sum_c (a_j^c)^2 \right) . \tag{3.20}$$

By eliminating a^c and b^c [65], Equ. (3.20) can be reformulated into the following quadratic function:

$$J(\alpha) = \alpha^T L \alpha \,, \tag{3.21}$$

where L is the **matting Laplacian matrix**, which is a square matrix with number of rows equal to the number of pixels in the image. Specifically, the (i, j)th element

of L is generated by

$$L(i,j) = \sum_{k|(i,j)\in w_k} \left[\delta_{ij} - \frac{1}{|w_k|} \left(1 + (I_i - \mu_k) \left(\Sigma_k + \frac{\varepsilon}{|w_k|} I_3 \right)^{-1} (I_j - \mu_k) \right) \right] , \quad (3.22)$$

where $|w_k|$ is the number of pixels in window w_k, Σ_k is the covariance matrix of pixel colors in window w_k, μ_k is the mean value of the pixel colors in w_k, I_3 is a 3 by 3 identity matrix, and δ_{ij} is the Kronecker delta, which is defined by

$$\delta_{ij} = \begin{cases} 0, \ i \neq j \, , \\ 1, \ i = j \, . \end{cases} \quad (3.23)$$

Given the user constraint such as the trimap or the scribbles, the α map can be calculated by Equ. (3.24):

$$\hat{\alpha} = \arg\min \alpha^T L \alpha, \ \ s.t. \ \hat{\alpha}_i = \alpha_i \ \forall i \in S \, , \quad (3.24)$$

where $\hat{\alpha}$ is the estimated α value, S refers to the set of pixels in predefined foreground and background region, α_i refers to the α value (0 or 1) in foreground and background region.

The propagation based matting methods, which use matting Laplacian matrix, can work well if the color in the local window obey the linear assumption. In order to reduce the computational cost, the window size in Equ. (3.22) is usually small (e.g. 3 by 3). As an improved version of closed-form matting, the adaptive window size was proposed in [67]. In this work, it was found that small window size may cause over smoothing in image regions with complex structures. On the other hand, a large window size can sometimes improve the matting result. However, large window size increases the computational cost, and it also makes it more possible to break the color line assumption.

Another propagation based matting method is called as random walk matting

[60]. The same as closed form matting, random walk matting also utilized the local affinity to propagate the α value from known foreground-background region to the entire image. Specifically, it computes the probability that a random walker, which starts from pixel C, reaches a foreground pixel before it reaches a background pixel. This probability is treated as the α value of pixel C. The random walker is set to move on a weighted graph for which each pixel is a node. There are four edges connecting to each node so that every neighboring pixels are connected. The weight w_{ij} of the edge connecting pixel i and j infers the probability that a random walker at i moves to j, and it is determined by the normalized color similarity between pixels:

$$w_{ij} = \frac{exp\left(-\left\|C_i - C_j\right\|^2 / \sigma^2\right)}{\sum_k w_{ik}},\ k \in \text{neighbors of}\, i\,, \tag{3.25}$$

where σ is a tunable constant, and C is the transformed color [68] of the pixel. Given the weight between every neighboring pixel, the α matte can be calculated by the same way as closed form matting.

Similar with nonlocal closed form matting [69] [70], KNN matting [71] [72] is also based on the assumption of nonlocal similarity. In KNN matting, it is assumed that the pixels with similar color and texture appearance should be expected to share similar α values. In this case, a feature vector $X(i)$ is defined for pixel i by Equ. (3.26):

$$X(i) = (cos(H_i), sin(H_i), S_i, V_i, x_i, y_i)\,, \tag{3.26}$$

where H, S, V are the hue, saturation and lightness values in HSV color space, x and y are the spatial coordinate.

Given the feature vector for each pixel, the affinity between two pixels can be

established by

$$L(i,j) = 1 - \frac{\|X(i) - X(j)\|}{C}, \tag{3.27}$$

where C is a constant representing the least upper bound of $\|X(i) - X(j)\|$. Afterwards, the α matte can be calculated by the same way as closed form matting.

Compared with closed form matting, KNN matting performs better in texture regions with complex intensity variations. On the other hand, closed form matting outperforms KNN matting at smooth regions.

In conclusion, compared with sampling based method, the propagation based matting methods show better performance to the images with complex textures and plentiful details. This is because that the local texture is considered in propagation methods by constructing local affinity. However, the performance of propagation methods heavily rely on the assumption made in the local affinity. If the assumption breaks, the matte result may be erroneous.

3.2.3 Combination methods for natural image matting

As aforementioned, the sampling based methods and the propagation based methods both have advantages and disadvantages. The sampling based method works well in simple images where reliable samples are easy to be collected. On the other hand, such method may fail in complex image regions where reliable samples are hard to be found and the foreground, background color distribution overlap. The propagation based method can work well in complex image regions because it investigate the local relations between neighboring pixels. However, this kind of method may fail if the underlying assumption can not be met. In order to achieve better matting results, recent researches [73] [74] preferred to combine these two kinds of matting techniques into one energy function, which can be solved by optimization processes.

The matting method based on iterative belief propagation [74] is one representa-

tive combination method. In this method, the α value is estimated iteratively. In each iteration, the algorithm locally collects foreground and background samples for each unknown pixel. With the color samples, a Markov random field (MRF) is constructed. every known pixels and reliably estimated pixels in last iteration are treated as the nodes in MRF. Then the nodes are connected to their neighbors. The possible α values for each unknown pixel are discretized into K-levels, which correspond to possible states in the MRF. The energy function used here involves two terms: the data term and the affinity term. The data term

$$D(\alpha_p^k) = 1 - \frac{L_k(p)}{\sum_{k=1}^{K} L_k(p)} \tag{3.28}$$

represents the character of the pixel itself, such as the reliability of the foreground-background sample pair. In Equ. (3.28), k represents the state in MRF, α_p^k is the estimated α for pixel p in state k, $L_k(p)$ represents the reliability of the foreground-background sample pair used to estimate α_p^k. The affinity term

$$L(\alpha_1, \alpha_2) = 1 - exp\left(-\frac{(\alpha_1 - \alpha_2)^2}{\sigma_s^2}\right) \tag{3.29}$$

represents the relations between the current pixel and its neighborhood. In Equ. (3.29), σ is a constant, α_1 and α_2 are the α values of two neighboring pixels.

With the constructed MRF model, the algorithm used loopy belief propagation [75] to solve the α map by finding the global minimum of the energy function.

4

Proposed GMM based color representation model

In this chapter, we propose an accurate and efficient method to represent color images based on Gaussian mixture model (GMM). This method is helpful in comprehensively collecting foreground/background color candidates in matting problems.

4.1 Color sampling in matting problems

In matting problems, foreground/background color prediction is always an essential part for accurate alpha estimation. Normally, researchers have to find the balance between efficiency and diversity when they are considering the way to choose foreground/background color candidates. Local sampling is a widely adopted method that efficiently collects the color candidates at the risk of missing important color

information. On the contrary, more comprehensive color candidates could be collected if the samples were globally selected in the entire image. The global sampling, however, costs much more computational time compared with local sampling.

In order to achieve efficiency and color diversity at the same time, we solve the sampling problem based on the knowledge of global color distribution, which is analyzed and represented by our proposed GMM color representation method. In the remaining part of this chapter, we will introduce our proposed GMM color representation in detail.

4.2 Color sparsity in natural images

The problem of representing image pixels in a way consistent with human perception is one of the essential problems in computer vision. An appropriate representation of pixels in an image can be of great help for the subsequent image analysis. A major kind of solutions for pixel color representation is to design novel color spaces from conventional sRGB color space so that the distance in the new color space can isotropically represent the color difference in the sense of human vision. However, the design and derivation of most existing color spaces are independent of image content. This might be problematic from the following perspectives:

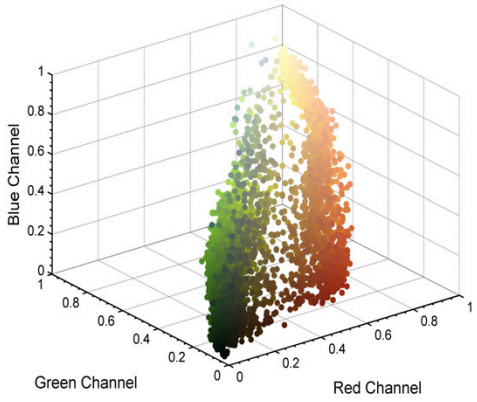
- The human vision system can always automatically adjust the sense of colors with respect to the view conditions, such as the lighting temperature, the color of the image background, and even the noise level in the image. This brings the question whether it is appropriate to deal with a color with the same metric in different situations.

- It is observed that the colors are significantly sparsely distributed in natural images [76]. In this case, we need to ask whether it is necessary for a color representation model to be able to represent all possible colors even if most of them do not show up in the image.

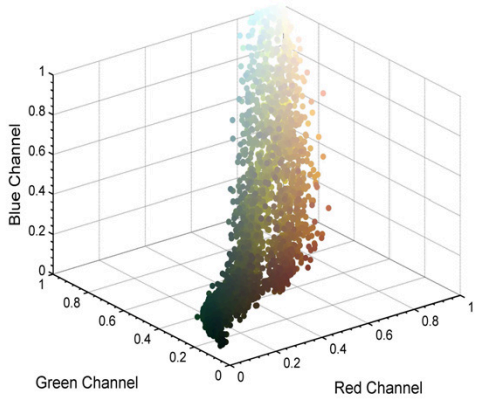
In order to illustrate the fact that the colors are sparsely distributed in natural images, we take two test images as an example from the Berkeley segmentation database [10] [77]. The global color distribution of these two test images can be found in Fig. 4.1 (c) and (d). Specifically, the colors in the whole test images are plotted as colored dots in sRGB color space, and it can be found that the colors in one natural image only occupy a small part of the whole sRGB color space.

(a) Example image #1

(b) Example image #2



(c) The global color distribution of image # 1.



(d) The global color distribution of image # 2.

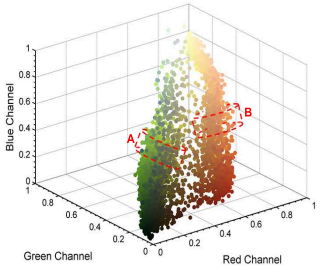
Figure 4.1: The global color distribution of two test images.

Besides the global color distribution, we also count the local color statistics in sRGB color space. As shown in Fig. 4.2 (a) and (d), two local regions marked by red cylinders are selected for each test image. The local color distribution of image #1 is presented in Fig. 4.2 (b) - (c), while the local color distribution of image #2 is presented in Fig. 4.2 (e) - (f).

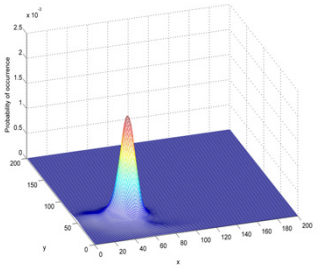
Two facts can be observed in the given example:

- The colors in an image only take a very small part in the entire color space (Fig. 4.1 (a) and (d)).
- The color distribution is highly concentrated (Fig. 4.2 (b) (c) (e) and (f)).

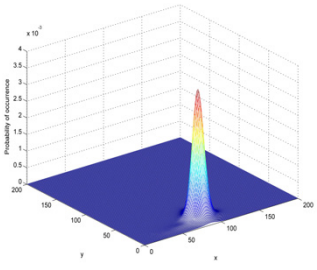
Given the fact that colors in a natural image usually sparsely distribute, it is not necessary to uniquely represent every possible color when we are dealing with one specific image. For example, the sRGB color space can represent 2^{24} unique colors if there are 256 different intensity levels in each color channel. However, it is very unlikely that human vision can distinguish so many different colors from just one image.



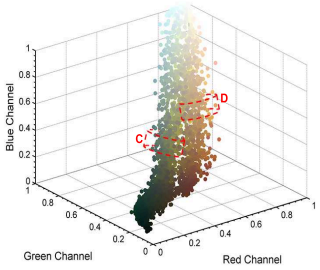
(a) The global color distribution with two marked local color regions (image # 1).



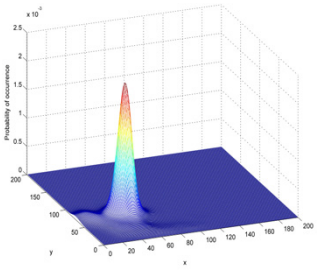
(b) The local color distribution in region A.



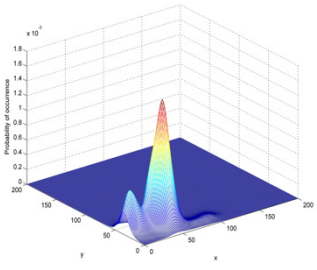
(c) The local color distribution in region B.



(d) The global color distribution with two marked local color regions (image # 2).



(e) The local color distribution in region C.



(f) The local color distribution in region D.

Figure 4.2: The sparse color distribution in natural images.

In this chapter, we propose a novel image color representation model to represent a color image by using a small set of colors D_c. The colors in D_c will be adaptively determined by the global color distribution of the image, so that this color representation model is self-adaptive to the content of the image to be represented.

4.3 Related works

Color quantization is the technique that reduces the number of unique colors in a digital image, while the visual quality of the quantized image is similar to the original one. The quantized image is often recorded by a **color palette** that contains a set of quantized colors, and an **index map** that stores the entry index of each pixel to the color palette. Therefore, a typical color quantization often involves two steps: color palette generation, and pixel-palette entry index assignment.

This technique has been investigated for decades because of its wide applications in image displaying, image printing, lossy compression, segmentation, image analysis and many other tasks. In order to find appropriate palette, different clustering methods were used.

Uniform quantization [78] independently divides each color axis into equal sized subspaces. The entire color space is divided into a set of boxes by planes that perpendicularly pass through the division points on the axis. The centroid of each box is regarded as one of the palette color, and all colors in the box are quantized and represented by this centroid color. The number of boxes is dependent on the quantization levels on each color axis. For example, one previous approach [78] quantized the whole color space into 256 subspaces by dividing the red and green channel into 8 segments and the blue channel into 4 segments. The color palette of uniform color quantization does not have any relation with the content of the image. In this case, such color palette is universal and easy to use. However, the visual quality of the quantized image is often not satisfactory when the size of color palette is small.

Another form of uniform quantization is the popularity algorithm [78]. If the required size of the palette is N, the popularity algorithm does not directly divide the whole color space into N boxes. Instead, it initially divides the color space into many more boxes (i.e., 262144 boxes) by using much smaller box size (i.e., $4 \times 4 \times 4$). The pixels in the original image are then mapped to the boxes that their colors fall in.

The representative color of each box is the average color of the pixels that mapped to it. Finally, the N palette colors are chosen by selecting the N most popular representative colors. The popularity here is determined by the number of pixels falling in each box.

Contrary to uniform quantization, non-uniform quantization is more preferable because it can divide the color space according to the color distribution of the original image. The median cut method [79] is one of the most widely used color quantization methods that adaptively split the color space. The underlying principal of the median cut method is to make sure that every palette color represents the same number of pixels in the original image. Specifically, this method starts with finding a smallest box that contains all pixel colors of the original image in the color space. The enclosed colors are then sorted along the longest axis of the box. After that, the original box is split into two boxes at the median of the sorted list. The above processes will repeat until the number of boxes equals to the desired size of color palettes. Within each box, the average color is calculated and it is used as one of the palette colors for the image. The recursive bipartition used in median cut is identical to the problem of $k-d$ trees. By using the same $k-d$ tree framework, variance minimization is proposed [80] [81] for better quantization performance.

Besides the aforementioned splitting methods, different conventional clustering methods were also used for color quantization. The k-means [82] is one of the most fundamental clustering methods used in color quantization. After the initial centroids are set, the pixel colors are grouped according to their nearest centroids. Then the color centroids are updated to the average color in each group. The above process is repeated until convergence is achieved. The grouping result of k-means highly relies on the initial condition. In this case, fuzzy c-mean (FCM) [83] was proposed to decrease the adverse effect of the initial condition. The FCM generally generates better clustering results than the k-means does and the FCM is more robust to the existence of outliers in the original data set [84].

Although color quantization works for reducing the number of unique colors in an image, it is still an inevitable fact that better color approximation is achieved by larger size of palette. Considering natural images, a sufficient color palette could contain hundreds of different colors. In the case of foreground/background color estimation, which is also the reason that we investigate color quantization in this thesis, hundreds of foreground/background color candidates make the alpha estimation not robust and introduce large calculation cost. In this case, we propose our color image representation model that can robustly categorize the colors in a natural image into $20 - 50$ groups. Based on these color groups, the original image can be represented with high quality.

4.4 Overall structure of the proposed color representation model

In our proposed color representation model, the pixel value is parameterized with respect to the global color distribution in the current image. The underlying assumption of our proposed representation is the fact that the chrominance in a natural image is limited and sparse in the sense of human perception, and we call those colors as dominant colors D_c in the image. We further assume that all the colors, which appear in a natural image, can be modeled based on those dominant colors. Specifically, we first approximate the global image color distribution by the sum of a series of mixture Gaussian functions. The centroids of these Gaussian functions are regarded as the dominant colors D_c of the image. In order to further model the colors not near to any of the Gaussian centroids, a simple linear model is proposed. Our proposed color representation explains the image color in a semantic way, and it can be easy to use for image analysis, such as segmentation, color editing, and compression. The overall flowchart of the proposed representation method is as shown in Fig. **??**.

Given an image in sRGB color space, the pixel colors are first transformed into HSV-based color space. The HSV color space could be any color space that separates the lightness from the chrominance. V denotes the lightness channel; H (Hue)

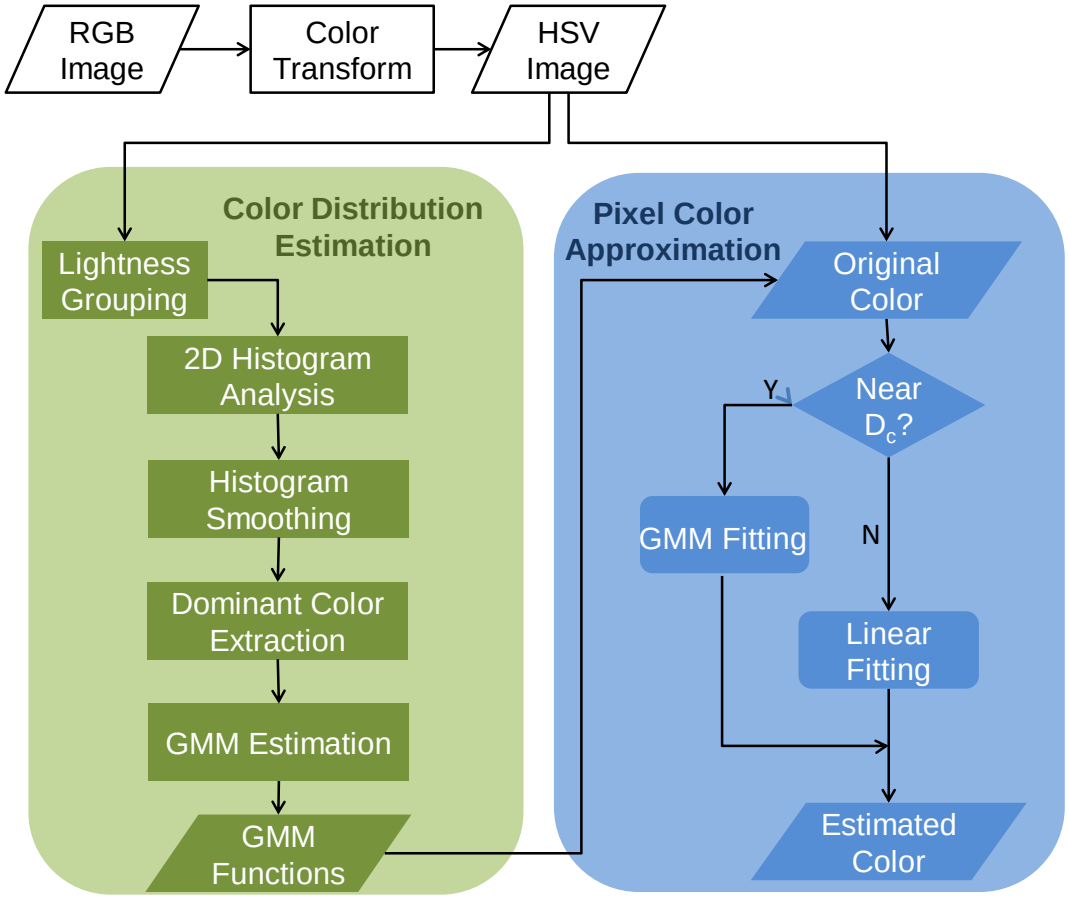


Figure 4.3: The overall flowchart of the proposed GMM based color representation model.

and S (Saturation) denote the polar coordinates of the chrominance plane. Then the pixels are grouped with respect to their lightness values. The pixel colors with similar lightness values are histogram analyzed on the Hue-Saturation plane. Given the histograms, the dominant colors in each lightness level will be estimated by finding out the local maximum of the smoothed histogram. Around each dominant color, a Gaussian mixture model (GMM) will be applied to fit the local chrominance distribution. By doing this in each lightness level, we can obtain a set of GMM clusters which are used to approximate the global color distribution.

Now we already have a parameterized model (a set of GMMs) of the global color distribution, the next step is to reconstruct each pixel value based on this model.

Given the color of one pixel, the probability that this color belongs to each GMM is calculated. If this color has a high enough probability to one GMM, this color will be reconstructed based on that GMM and its associated dominant color. If this color does not have high probability to any of the GMMs, a linear combination model will be applied. Specifically, this color will be represented by a linear combination of two different dominant colors.

The detailed descriptions for each step aforementioned will be introduced in the remaining sections of this chapter.

4.5 Global color distribution estimation

In most widely used color spaces, the color representation is based on fixed transformation from RGB or CIE-XYZ space. In the perspective of human color sensation, fixed transformation may not be suitable because the human visual system can adjust the color perception based on the viewing environment (i.e., the color content in the image). Besides, the color of one pixel is represented independently from other pixels in the image in most color spaces. The lack of global view of image color components results in that the pixel color is easily affected by noises. With this concern, the global color distribution of the image is estimated to provide a concrete prior before we represent the pixel colors in this image.

4.5.1 Hierarchical histogram analysis based on lightness level

Since objects with similar intrinsic color may appear differently under different light conditions, it will be more robust to analyze colors with similar lightness levels. With this concern, it is desirable to separate the lightness value from RGB values. Although there are many human visual system based color spaces (e.g., *Lab*, YUV, YC_bC_r, *etc.*) from which we can extract the lightness level, we find that the results of dominant colors extraction are similar in different color spaces. Therefore, we use the HSV based color space whose color transformation is simple and efficient. The

color transformation between the sRGB space and the HSV space is given as follows:

$$C = \max(R, G, B) - \min(R, G, B)\,, \tag{4.1}$$

$$H = \begin{cases} 0, & \text{if } C = 0\,, \\ 60^\circ \times \left(\dfrac{G-B}{C} \bmod 6 \right), & \text{if } M = R\,, \\ 60^\circ \times \left(\dfrac{B-R}{C} \bmod 6 + 2 \right), & \text{if } M = G\,, \\ 60^\circ \times \left(\dfrac{R-G}{C} \bmod 6 + 4 \right), & \text{if } M = B\,, \end{cases} \tag{4.2}$$

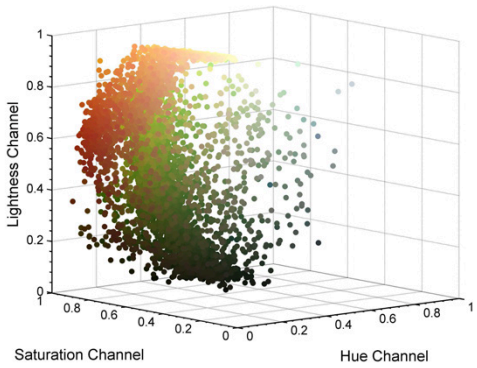
$$S = \begin{cases} 0, & \text{if } C = 0\,, \\ \dfrac{C}{V}, & \text{otherwise}\,, \end{cases} \tag{4.3}$$

$$V = \max(R, G, B)\,, \tag{4.4}$$

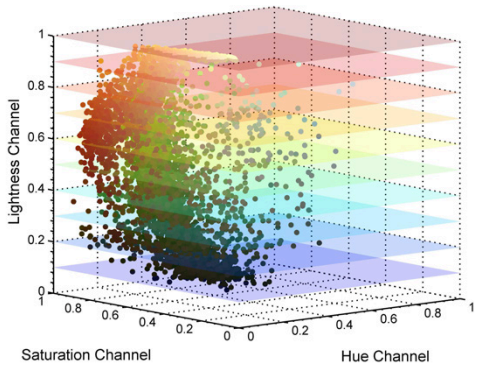
where V denotes the lightness channel, H and S denote the polar coordinates of the chrominance plane. After the transformation, the color distribution is estimated in the HSV color space with respect to different lightness levels as shown in Fig. 4.4.

By evenly dividing the lightness range (0 - 1) with an interval of 0.1, the pixel colors in the image is categorized into 10 groups according to their lightness levels, which are represented as $\{L_i | i = 1, 2, \ldots, 10\}$. In Fig. 4.4 (c) and (d), we present the color distribution of two groups with different lightness levels. In each group, the color distribution appears to be simpler than the global distribution, which ease the estimation of the color distribution.

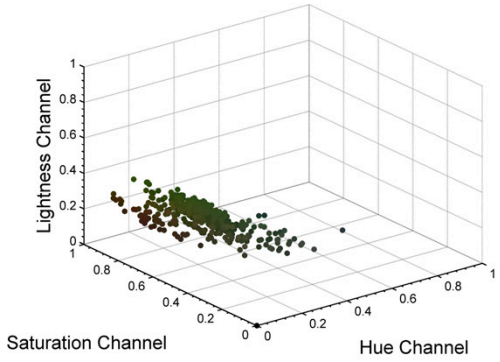
Given the colors in one group, 2D histogram analysis is applied in Hue-Saturation plane to estimate the color distribution in the current lightness level. The histogram is a representation of the occurrence distribution of continuous or quantitative variable



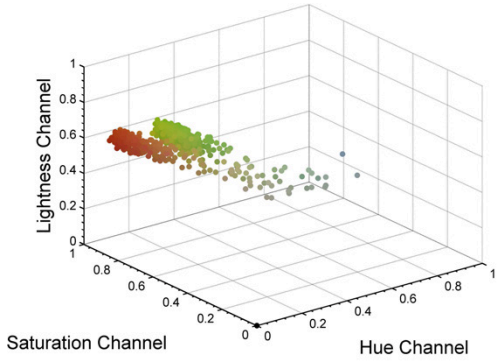
(a) The global color distribution of an image.



(b) Color space is evenly divided according to lightness level.



(c) Color distribution with $L \in (0.2, 0.3]$



(d) Color distribution with $L \in (0.6, 0.7]$

Figure 4.4: The color grouping based on lightness level.

[85]. In order to estimate the histogram of an observed data set, the first step is to set the bins, which divide the entire value range of the observed data into a series of intervals. After that, we count how many observed data falling into each interval (i.e., bin). This occurrence number is then set to be the height of each bin. Usually, the bins are consecutive, non-overlapping, and with the size width. Mathematically, the histogram of variable X is

$$H(i) = C(X|X \in [t_i - 0.5w, t_i + 0.5w]) \,, \tag{4.5}$$

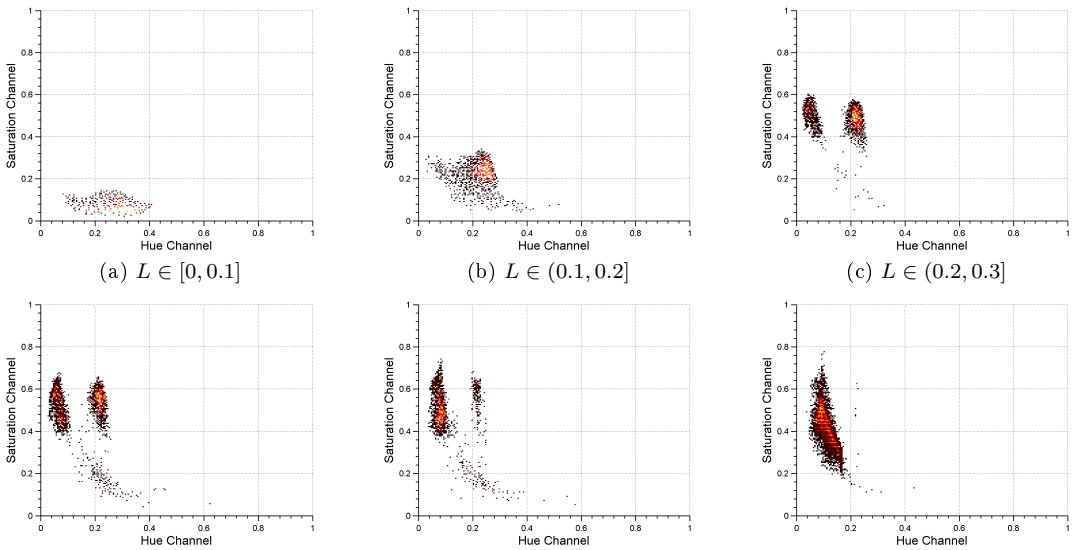
where $H(i)$ is the height of the ith bin, X is the set of observed data, t_i is the center value of the ith bin, w is the width of the bins, and $C(x)$ is the function counting the number of x.

In this thesis, we set the bin size as 0.005 by 0.005 to analyze the 2D histogram on the Hue-Saturation plane. Therefore, Equ. (4.5) can be rewritten into

$$H_{i,j} = C((H,S)|H \in [t_i - 0.0025, t_i + 0.0025],\ S \in [t_j - 0.0025, t_j + 0.0025]), \quad (4.6)$$

where $(t_i, t_j) \in \{0.0025, 0.0075, 0.125, \ldots, 0.9975\}$, H and S represent the hue value and saturation value of a pixel respectively.

According to Equ. (4.6), the color distribution on each lightness level can be estimated by 2D histogram as shown in Fig. 4.5 and Fig. 4.6. In Fig. 4.5, the histogram is viewed from the top of the Hue-Saturation plane. In this case, it is represented in 2D manner. In this figure, the brighter the bin is, the colors occur more frequently in this bin. In Fig. 4.6, the histograms are presented in 3D manner by viewing them from side of the Hue-Saturation plane. In this figure, the magnitude of each bin represents the occurrence frequency of the colors in the bin.

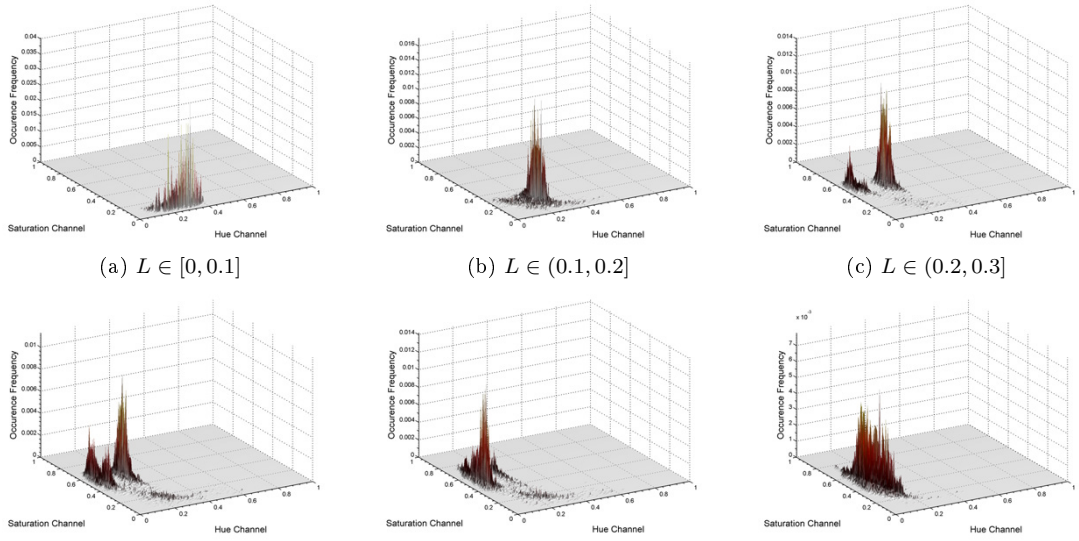


(a) $L \in [0, 0.1]$ (b) $L \in (0.1, 0.2]$ (c) $L \in (0.2, 0.3]$

(d) $L \in (0.6, 0.7]$ (e) $L \in (0.7, 0.8]$ (f) $L \in (0.8, 0.9]$

Figure 4.5: The color distribution in 6 different lightness levels, viewing from top of Hue-Saturation plane.

In Fig. 4.5 and Fig. 4.6, it can be observed that the histogram of the original image is not smooth. This phenomenon could be caused by reasons such as sensing



(a) $L \in [0, 0.1]$ (b) $L \in (0.1, 0.2]$ (c) $L \in (0.2, 0.3]$

(d) $L \in (0.6, 0.7]$ (e) $L \in (0.7, 0.8]$ (f) $L \in (0.8, 0.9]$

Figure 4.6: The color distribution in 6 different lightness levels, viewing from side of Hue-Saturation plane.

distortion at the camera, the compression distortion at the encoder, and the down-sampling/upsampling distortion. In this case, we smooth the 2D histogram in the following section.

4.5.2 2D histogram smoothing using first and second order difference penalties

In order to smoothly and robustly model the global color distribution, we refine the raw color histogram by constructing and minimizing an objective function with first and second order differential penalties:

$$z_s = \arg\min_z |y - z|^2 + \gamma^2 |D_2 z|^2 + 2\gamma |D_1 z|^2 , \tag{4.7}$$

where y is the original histogram, z is the smoothed histogram, and z_s is the optimal smoothed histogram with first and second order difference penalties. D_1 is the first order differential operator, and D_2 is the second order differential operator. γ is the factor controlling the strength of smoothing, the larger γ is, the smoother histogram is

generated. Equ. (4.7) was originally proposed to smooth one dimensional histograms in [98]. This equation can be efficiently solved by rewriting it into the form of linear system equation as follows:

$$(I + \gamma^2 D_2' D_2 + 2\gamma D_1' D_1) Z = Y \,, \tag{4.8}$$

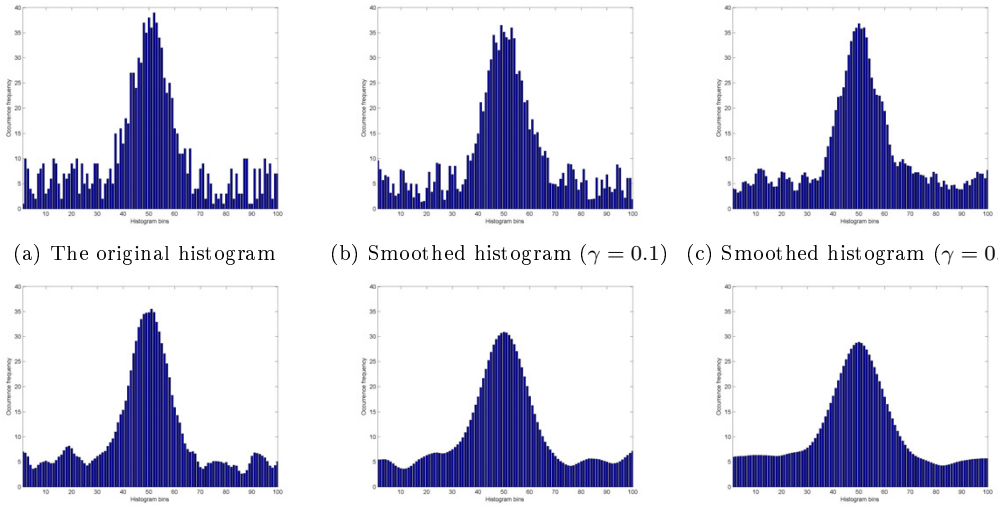
where I is a N by N identity matrix, Y is a N by 1 matrix containing the original values of the histogram, Z is a N by 1 matrix containing the smoothed values of the histogram. Here N is the number of bins in the histogram, D_1 and D_2 are the first and second order differential operators.

Therefore, the values in the smoothed histogram can be calculated as follows:

$$Z = (I + \gamma^2 D_2' D_2 + 2\gamma D_1' D_1)^{-1} Y \,. \tag{4.9}$$

Fig. 4.7 shows the smoothed histogram by using different γ values. It can be observed that the histogram becomes smoother when γ value increases. Meanwhile, the overall shape of the original histogram is always kept so that the underlying distribution of the original data can be estimated based on the smoothed histogram.

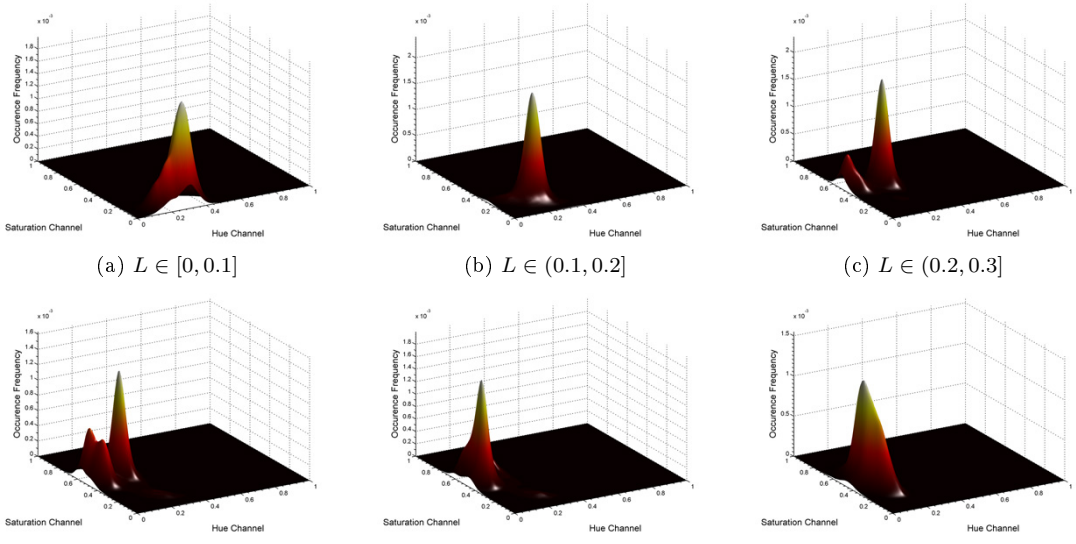
In the case of 2D histogram smoothing on Hue-Saturation plane, we apply Equ. (4.9) to each row and column in the matrix of Hue-Saturation histograms so that the 2D histograms can be smoothed. Similar to what we did in Fig. 4.6, we also provide the smoothed histogram from the side view of H-S plane (Fig. 4.8). Compared with the original histograms in Fig. 4.6, the smoothed histograms in Fig. 4.8 have better boundary and shape property.



(a) The original histogram (b) Smoothed histogram ($\gamma = 0.1$) (c) Smoothed histogram ($\gamma = 0.5$)

(d) Smoothed histogram ($\gamma = 1$) (e) Smoothed histogram ($\gamma = 5$) (f) Smoothed histogram ($\gamma = 10$)

Figure 4.7: Histogram smoothing using different γ values in 1D case.



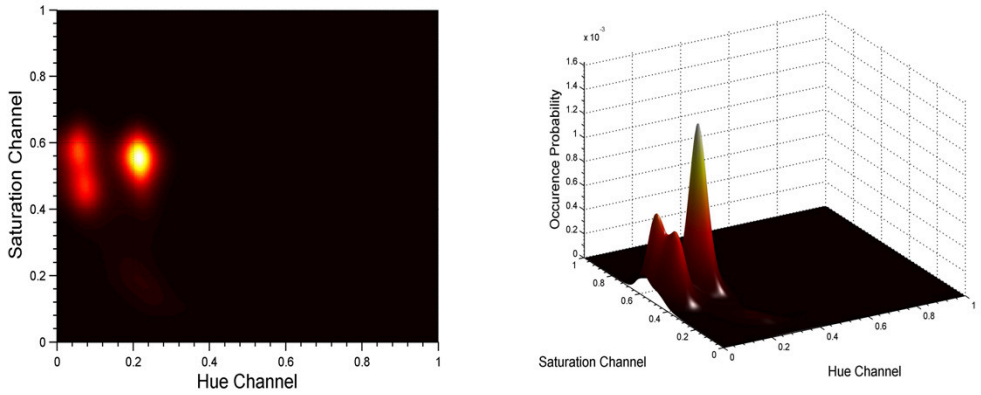
(a) $L \in [0, 0.1]$ (b) $L \in (0.1, 0.2]$ (c) $L \in (0.2, 0.3]$

(d) $L \in (0.6, 0.7]$ (e) $L \in (0.7, 0.8]$ (f) $L \in (0.8, 0.9]$

Figure 4.8: The smoothed color distribution in 6 different lightness levels, viewing from side of Hue-Saturation plane.

4.5.3 Histograms grouping based on watershed algorithm

The Gaussian model is used in this thesis to fit the observed colors (i.e., the smoothed histogram) in the image. At the same time, it is obvious that the observed colors sometimes distribute in several Gaussian clusters. In the worse cases, those Gaussian clusters may overlap with each other. Fig. 4.9 provides an example of such case, where the observed colors are distributing in three overlapped clusters.

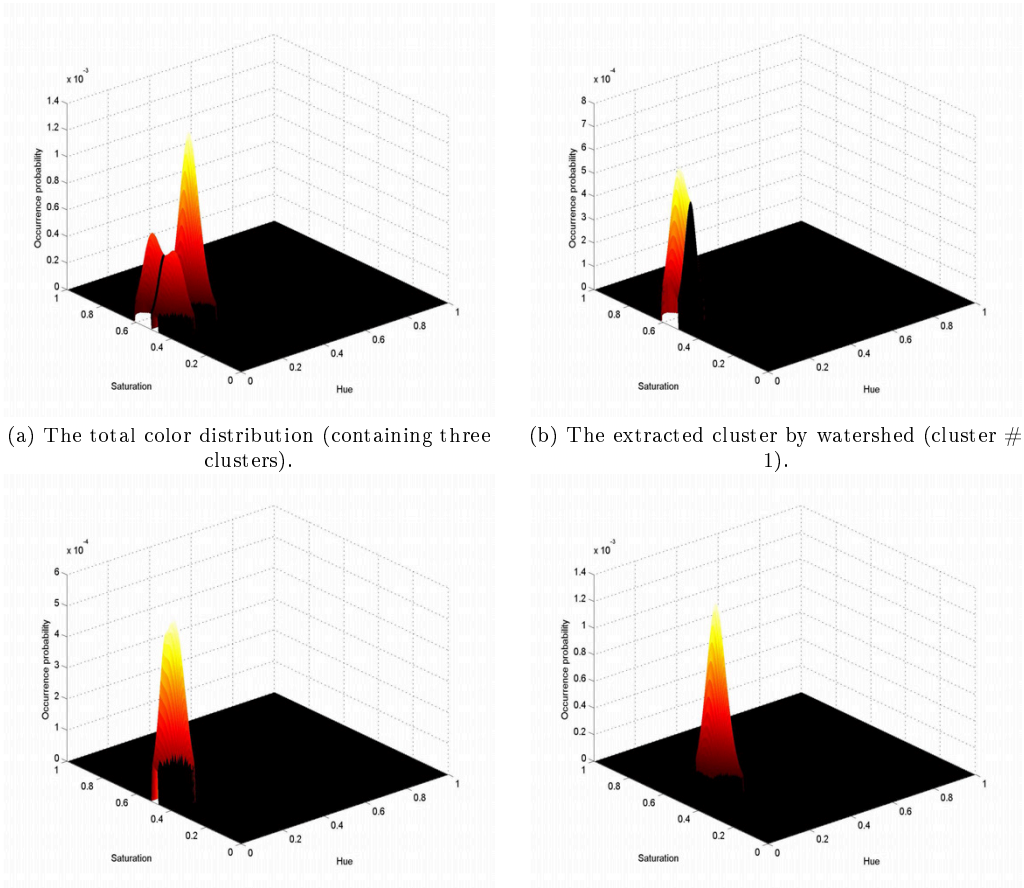


(a) Multiple Gaussian clusters (viewing from top) (b) Multiple Gaussian clusters (viewing from side)

Figure 4.9: The example of multiple Gaussian clusters in one lightness level.

In this case, it is desirable to group the observed colors into three sets, so that they can be separately Gaussian fitted. In this thesis, the watershed algorithm proposed in [99] is used to group the observed colors. Since the detailed derivations and proofs of the watershed algorithm in [99] is out of the scope in this thesis, only a brief introduction to the watershed algorithm will be made here. Generally speaking, the watershed algorithm has mainly worked for image segmentation in mathematical morphology since it was proposed in 1979 [103]. Normally, this method works for gradient images by detecting the catchment basins of all gradient minimum. An intuitive explanation of the watershed method can be made by considering the gradient image as a topographical relief. The relief is flooded by injecting the water at every local minimum. The flood increases with uniform speed all over the relief until the moment that the floods filling two adjacent catchment basins start to merge together.

At this moment, a dam is established to prevent the fusion of the floods. The set of all dams is called the watershed lines, which are also the boundaries of the image segment.



(a) The total color distribution (containing three clusters).

(b) The extracted cluster by watershed (cluster # 1).

(c) The extracted cluster by watershed (cluster # 2).

(d) The extracted cluster by watershed (cluster # 3).

Figure 4.10: Cluster extraction by using watershed.

As shown by Fig. 4.10, the overlapped clusters can be reliably extracted by using watershed. In this case, the Gaussian fitting can be applied to each of these clusters.

4.5.4 Color distribution fitting based on Gaussian mixture model (GMM)

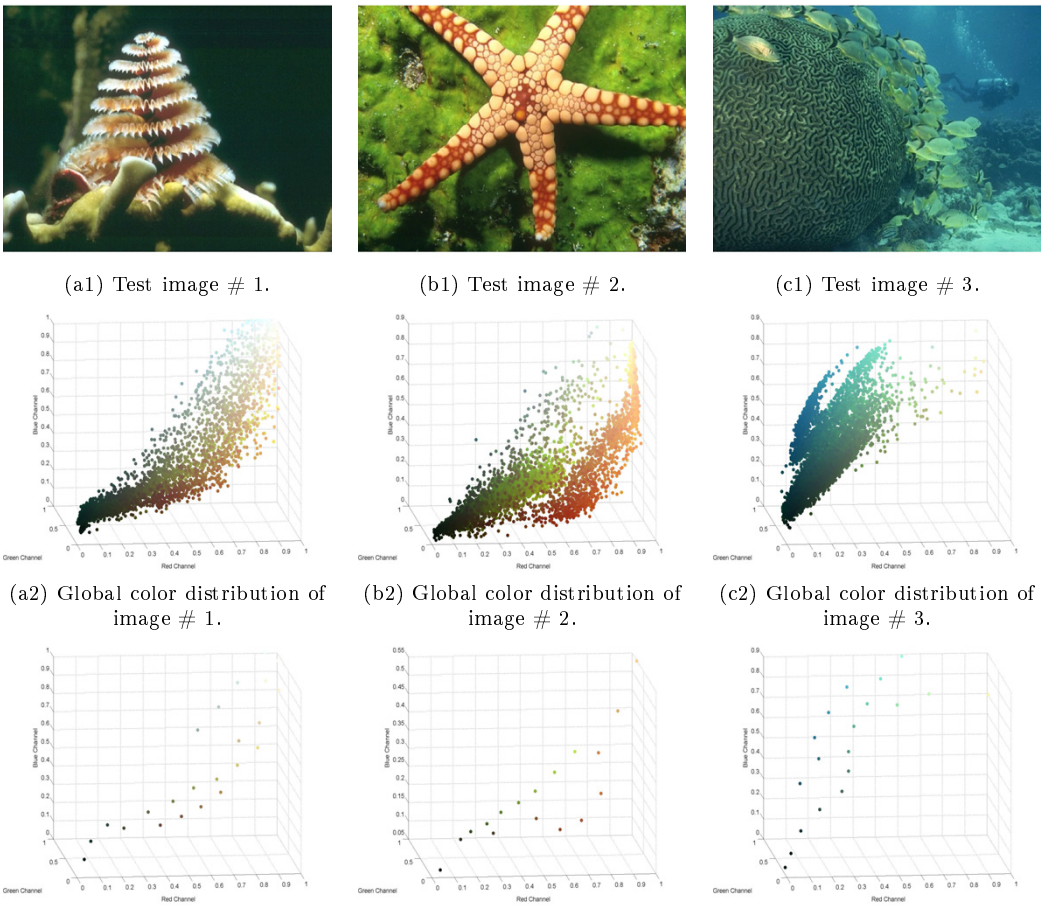
With the smoothed color distribution in each lightness level, we introduce our proposed dominant color extraction and modeling in this section. Since the global color distribution is already smoothed, it is easy to locate the local maximums of the color

occurrence frequency. In each lightness level i, the color values at local maximums are called dominant colors D_{ci}, and the number of dominant colors in this lightness level is N_i. After we find out dominant colors in all lightness levels, the complete set of dominant color is $\{D_{ci}|i = 1, 2, \ldots, M\}$, where M is the number of lightness levels. Therefore the number of total dominant colors will be $\sum_{i=1}^{i=M} N_i$. The extracted dominant colors in different test images are shown in Fig. 4.11 and Fig. 4.12. It can be observed that the major colors in the image (i.e., red, yellow, green, and brown) are reliably represented by the dominant colors.

Take the test images #5 and #6 for example. In image #5, although the global colors distribute in a narrow region in RGB space, the extracted dominant colors appear to have three major chrominance with different lightness: the pale green (grass color), the dark green (tree color), and the gray (the color of road and summerhouse). In image #6, the color distribution is more separable. The extracted dominant colors also appear to have three major chrominance with different lightness: the red (red flower), the purple (purple flower), and the green (the color of grass, trees and the river). It can be seen that the dominant colors can reliably represent the major chrominance in an image. Meanwhile, these dominant colors are capable of representing the overall chrominance of an image from the perspective of human sensation.

Although the colors in a natural image are most concentrated towards the dominant colors, it is still insufficient to represent the whole image colors by just a few dominant colors. If we simply approximate the pixel colors by its nearest dominant color, our proposed model would be another version of group based color quantization.

In other words, we need to represent the colors around each dominant colors with higher diversity and accuracy. According to our experiments, it was observed that the Gaussian function can well fit the color distribution around the dominant colors. In this case, we use Gaussian distribution to represent the color variation around dominant colors in this thesis.



(a1) Test image # 1. (b1) Test image # 2. (c1) Test image # 3.

(a2) Global color distribution of image # 1. (b2) Global color distribution of image # 2. (c2) Global color distribution of image # 3.

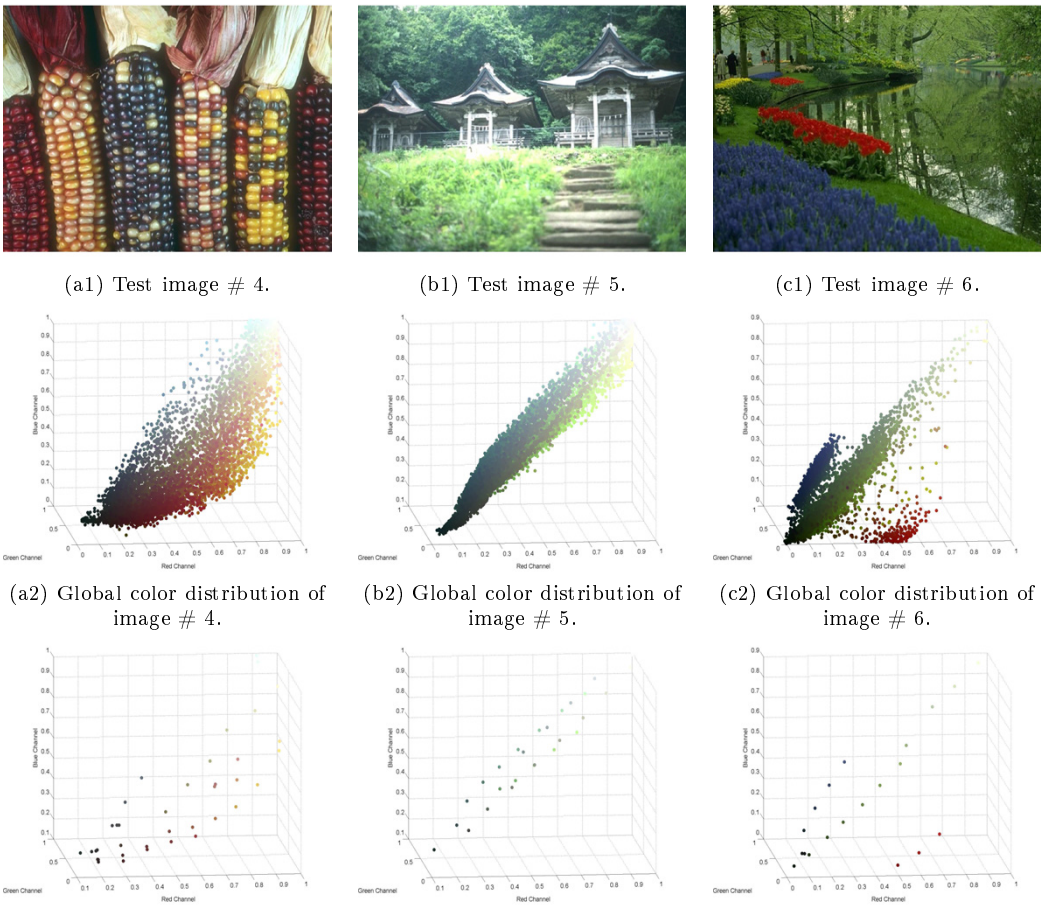
(a3) Dominant colors of image # 1. (b3) Dominant colors of image # 2. (c3) Dominant colors of image # 3.

Figure 4.11: The extracted dominant colors for test images (part 1).

It would be simple if we model the color distribution around each dominant color by using one 2D Gaussian function

$$f(x,y) = Aexp\left(-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)\right), \tag{4.10}$$

where $[x_0, y_0]$ is the center point, σ_x and σ_y are the variances in x and y directions. However, the approximation would be inaccurate if the local color distribution is not symmetric. As shown by the example in Fig. 4.13, it is obvious that the original color distribution in (a) is not symmetric, and the fitting result shown in (b) is not accurate by using just one Gaussian function.



(a1) Test image # 4. (b1) Test image # 5. (c1) Test image # 6.

(a2) Global color distribution of image # 4. (b2) Global color distribution of image # 5. (c2) Global color distribution of image # 6.

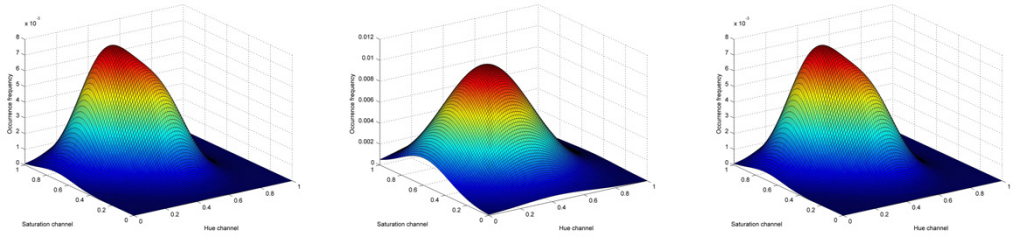
(a3) Dominant colors of image # 4. (b3) Dominant colors of image # 5. (c3) Dominant colors of image # 6.

Figure 4.12: The extracted dominant colors for test images (part 2).

In this case, we propose to use the Gaussian mixture model (GMM) in this thesis to represent the local color distribution. Around each dominant color, the local color distribution is modeled by a GMM, which is the sum of two Gaussian functions whose centroids are close to each other:

$$\begin{aligned} f(\boldsymbol{x}) = & A_1 exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_1})^T\Sigma_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu_1})\right)+\dots \\ & A_2 exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_2})^T\Sigma_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu_2})\right), \end{aligned} \tag{4.11}$$

where $\boldsymbol{x}$ is the color on H-S color plane, $\boldsymbol{\mu_1}$ and $\boldsymbol{\mu_2}$ are the mean values of two Gaussian functions, Σ_1 and Σ_2 are the covariance matrix of the two Gaussian functions.



(a) Color distribution to be fitted. (b) The fitted distribution using one Gaussian function. (c) The fitted distribution using GMM.

Figure 4.13: Example of color distribution fitted by GMM and single Gaussian.

By using GMM, local color distribution of natural images can be better fitted when the color components are complicated. Compared with the fitting result in Fig. 4.13 (b), the fitting result in Fig. 4.13 (c) is more accurate when GMM is applied.

After the mixture Gaussian model is applied to each dominant color, the global color distribution can be roughly represented by the combination of a set of mixture Gaussian functions ($\{MGF_i | i = 1, 2, \ldots, N\}$) centered at dominant colors ($\{D_{ci} | i = 1, 2, \ldots, N\}$) as shown in Fig. 4.14. The value of N here is the number of dominant colors extracted from the original image. In Fig. 4.14, the colored points represent the dominant colors (D_{ci}) in the image. The gray ellipsoid around each dominant color represents the region (SR_i) where each MGF_i is defined since only pixel colors nearby D_{ci} will be represented by MGF_i. If we only need to roughly estimate the colors in the image, the pixel's color can be directly approximated by the nearest D_{ci}. If high accuracy is required, the colors around each D_{ci} can be finely represented by using D_{ci} as the new coordinate origin. We will take a pixel color C with tristimulus values (0.62, 0.63, 0.87) as example. In order to represent this color in a conventional color space, the resolution of each color channel has to be at least 0.01. If the values in each channel range from 0 to 1, we need 100 different intensity levels for each channel. In our model, we assume that there is a dominant color D with tristimulus values (0.58, 0.60, 0.83) near the pixel color C. And the support region SR of D is a sphere centered at D with the radius of 0.05. In this case, the shift is (0.04, 0.03,

0.04) from C to D. Because the shift is only in the range of [-0.05, 0.05], we only need 11 different levels to represent these values with a resolution of 0.01. Furthermore, it is not the optimal choice to evenly divide the color space because we already know the local color distribution from mixture Gaussian functions MGF_i. In the following section, we will introduce the method that finely represents the colors according to MGF_i.

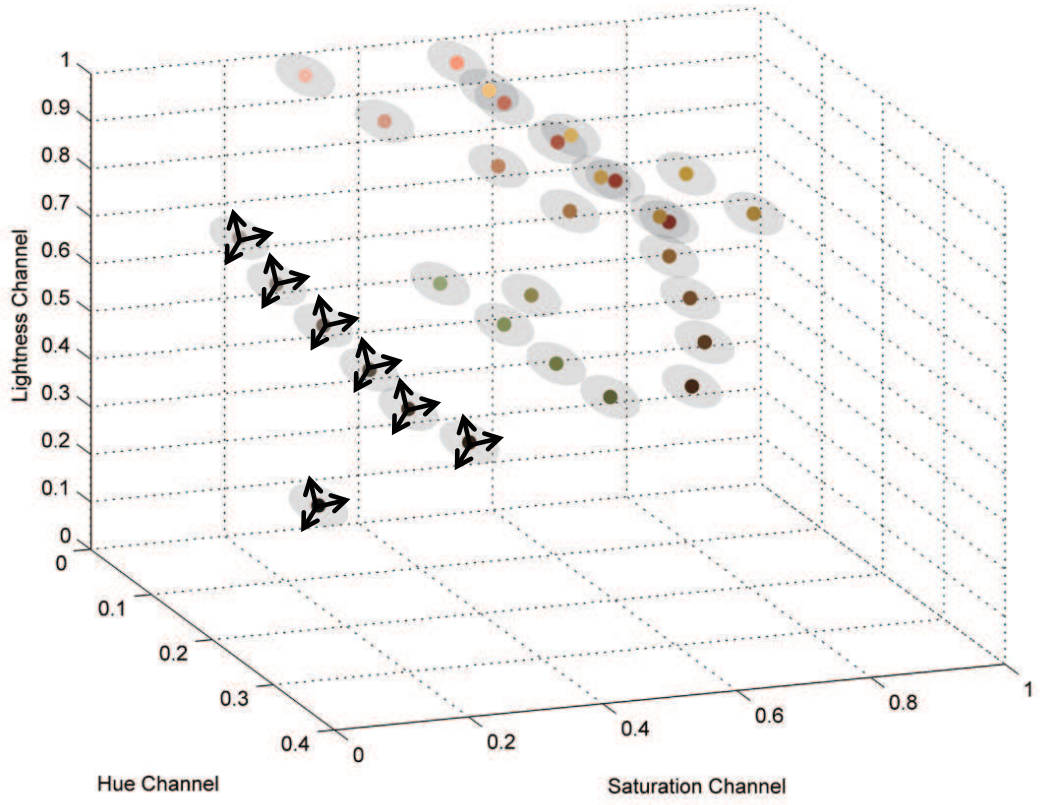


Figure 4.14: Mixture Gaussian model representing global color distribution.

4.6 Color quantization

In Section 4.5, the colors in an image are grouped into a set of dominant colors D_{ci} and each dominant color is associated with a GMM distribution GMM_i. Now we

will introduce the way to discretely represent the colors around each dominant color according to local Gaussian distribution GMM_i.

As illustrated in Fig. 4.15, we start to solve this problem from the simplest case: 1D Gaussian distribution with zero mean and unit variance. In order to discretely represent the values along the x-axis with N different values, it is desirable to determine a set of decision points $\{l_i | i = 1, 2, \ldots, N+1\}$, and a set of representative levels $\{p_i | i = 1, 2, \ldots, N\}$. In this case, the continuous values on the x-axis can be represented by discrete values according to the quantization function

$$f(x) = p_i, \ l_i \leq x < l_{i+1}\,, \tag{4.12}$$

where x is the values on x-axis, p_i is the quantized value of x, l_i is the decision points.

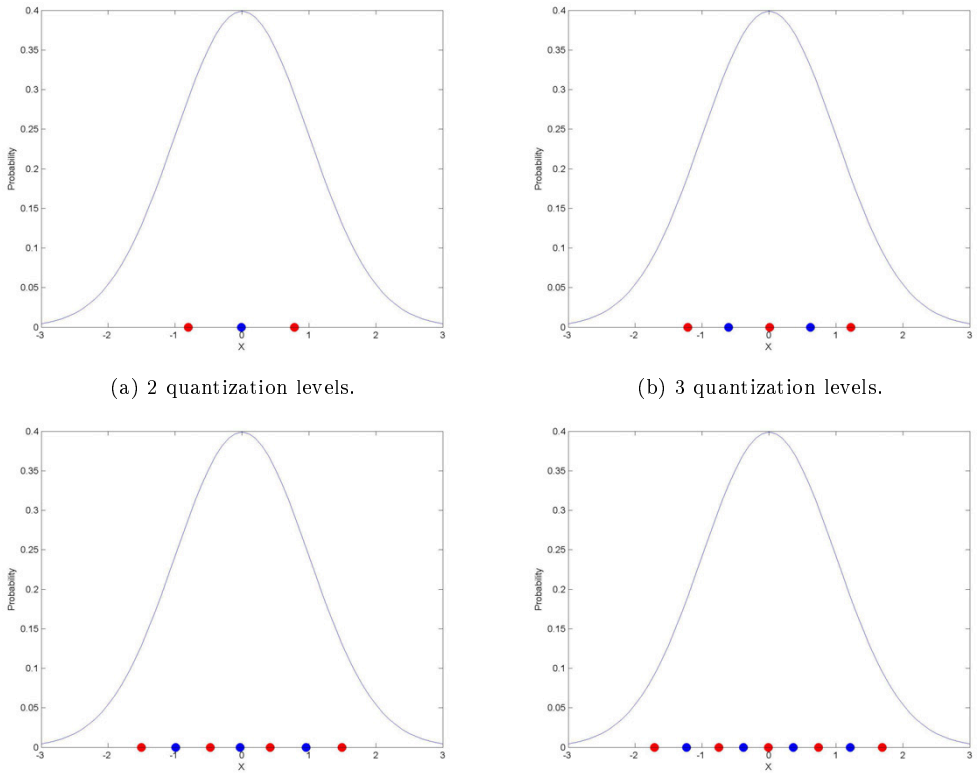
In order to find the optimal decision points and representative levels, the Lloyd-Max scalar quantizer [101] [102] is utilized. This algorithm uses an iterative process to minimize the mean square distortion of the quantization. As shown in Fig. 4.15, it is very easy to find the decision points and representative levels under different quantization levels.

The solution in the 1D case can be extended to the 2D case as shown in Fig. 4.16. In this thesis, the 2D solution is obtained by separately solving the 1D case horizontally and vertically.

With the decision points and the representative levels for 2D standard Gaussian, the last problem is to find the decision points and the representative levels for any arbitrary 2D Gaussian. Given any Gaussian distribution, its covariance matrix is denoted as Σ, which can be represented by its eigenvectors V and eigenvalues L:

$$\Sigma V = VL\,, \tag{4.13}$$

where V is the matrix whose columns are the eigenvectors of Σ, and L is the diagonal matrix whose non-zero elements are the corresponding eigenvalues.



(a) 2 quantization levels.

(b) 3 quantization levels.

(c) 4 quantization levels.

(d) 5 quantization levels.

Figure 4.15: Quantization for Gaussian distribution. Blue dots are the decision points; red dots are the representative levels.

In this case, the covariance matrix Σ can be rewritten as a function of its eigenvectors and eigenvalues

$$\Sigma = VLV^{-1}, \tag{4.14}$$

which is also called the eigen-decomposition of the covariance matrix and can be obtained by using singular value decomposition algorithm.

The eigenvectors V in Equ. (4.14) represent the directions of the largest variance of the data, the eigenvalues L represent the magnitude of this variance in those directions. In other words, V represents a rotation matrix, while $\sqrt{L}$ represents a scaling matrix from standard Gaussian to the current Gaussian. Now the covariance

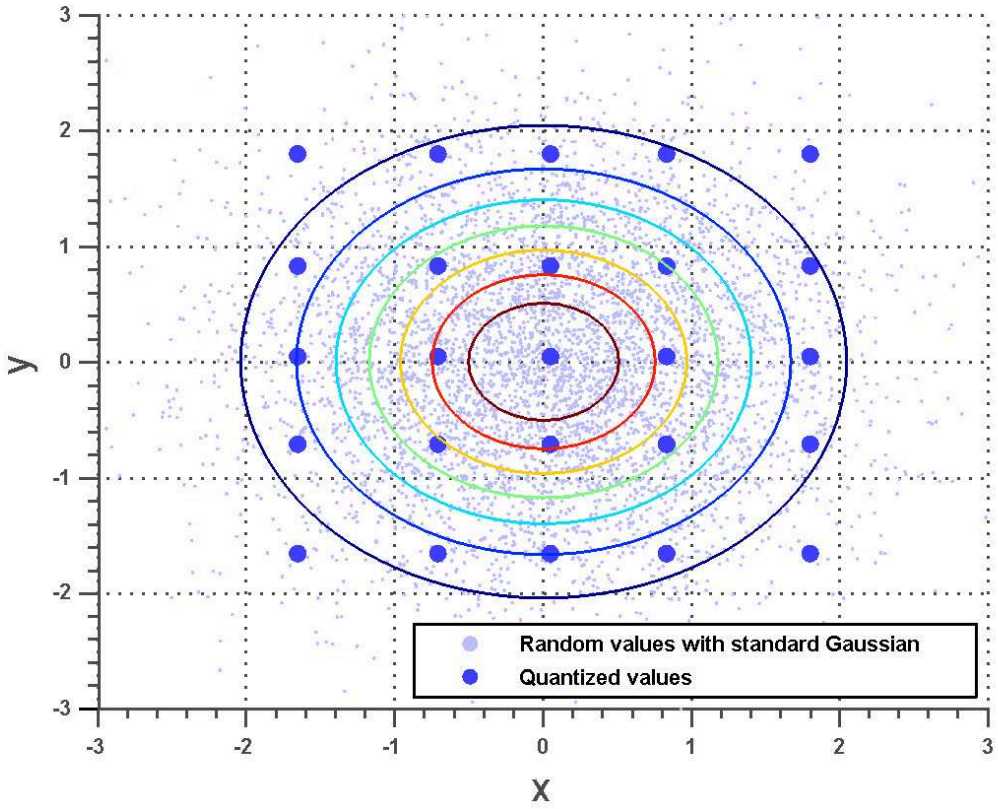


Figure 4.16: The representative levels for 2D standard Gaussian. The grey dots are randomly generated points according to 2D standard Gaussian distribution. The blue dots are the representative points for grey dots according to Lloyd-Max scalar quantizer. The contours represent values with probabilities of $0.14, 0.12, 0.10, 0.08, 0.06, 0.04, 0.02$, respectively.

matrix can thus be rewritten to

$$\Sigma = RSSR^{-1} = TT^T\,, \tag{4.15}$$

where $R = V$ is the rotation matrix, $S = \sqrt{L}$ is the scaling matrix; $T = RS$ is the linear transformation matrix that transforms a standard Gaussian to arbitrary Gaussian with covariance Σ.

In this case, the representative points for any arbitrary Gaussian can be obtained

by using the following equation:

$$P_{trans} = TP\,, \tag{4.16}$$

where T is the transformation matrix from covariance matrix Σ, P is a 2 by N matrix containing the representative points for standard Gaussian, N is the number of representative points, and P_{trans} is the representative points for Gaussian distribution with covariance matrix Σ. One example of such representative points can be found in Fig. 4.17.

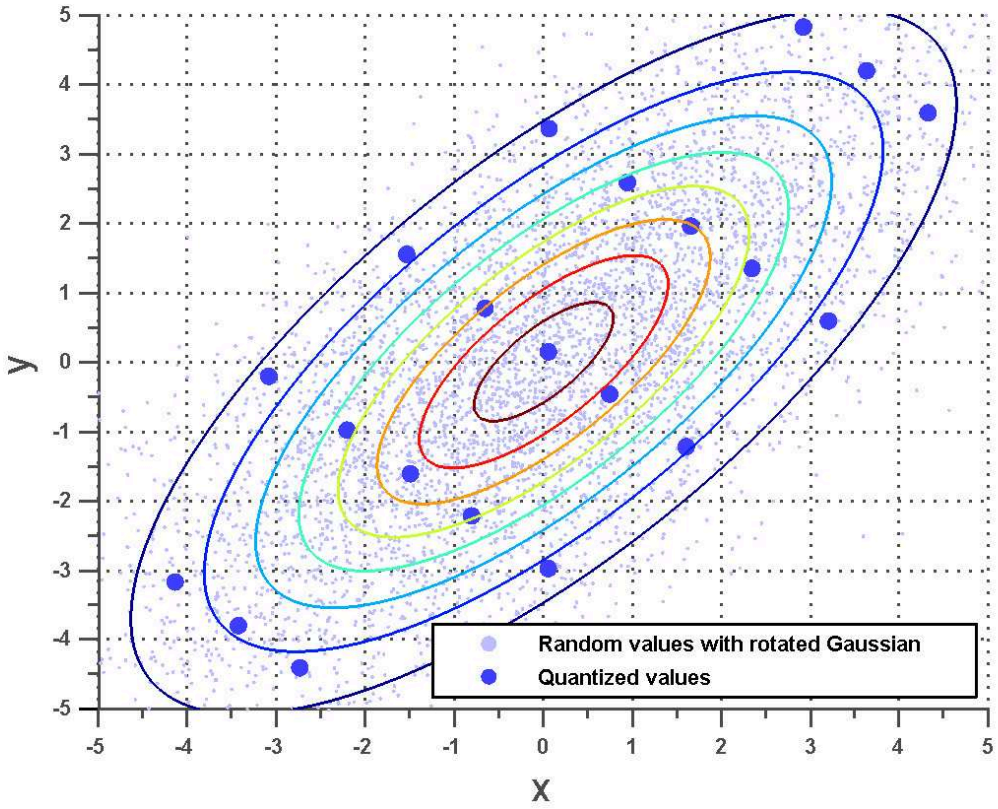


Figure 4.17: The representative levels for arbitrary 2D Gaussian. The gray dots are randomly generated points according to 2D Gaussian distribution with covariance matrix Σ. The blue dots are the representative points for gray dots according to Lloyd-Max scalar quantizer. The contours represent values with probabilities of $0.14, 0.12, 0.10, 0.08, 0.06, 0.04, 0.02, 0.005$, respectively.

Now we can reliably describe the colors around each dominant color by using the GMM functions. In this case, thousands of colors can be accurately represented by using only 20-60 dominant colors.

4.7 Linear model for outlier

Although the color distribution can be roughly represented by the combination of MGF_i, it is observed that there are still pixel colors not belonging to any of MGF_i. If these colors are directly approximated by one of the dominant color, the approximation error would be large. In this section, these pixel colors are called outlier colors (OC_i). Meanwhile, we found that outlier colors are very likely to be the result of mixing two dominant colors as shown in Fig. 4.18. In this figure, D_{c1} and D_{c2} are two dominant colors, and the gray spheres represent the regions where mixture Gaussian functions are defined. All colors inside the gray spheres can either be accurately represented by mixture Gaussian function, or they can be reliably approximated by the dominant color. For outlier colors outside of the gray sphere, it is no longer reliable to approximate them by single dominant color.

In this case, these outlier colors are approximated by using a linear combination model

$$OC_m = \alpha D_{ci} + (1-\alpha) D_{cj}\,, \tag{4.17}$$

where OC_m is the outlier color, D_{ci} and D_{cj} are two different dominant colors in the image, α is the blending factor to approximate OC_m from D_{ci} and D_{cj}. Given the color of OC_m, D_{ci}, and D_{cj}, the value of α can be estimated by Equ. (4.18):

$$\hat{\alpha} \;=\; \frac{(OC_m - D_{cj})(D_{ci} - D_{cj})}{\|D_{ci} - D_{cj}\|^2}\,, \tag{4.18}$$

where $\hat{\alpha}$ is the estimated value of α. In order to find out the most reliable dominant

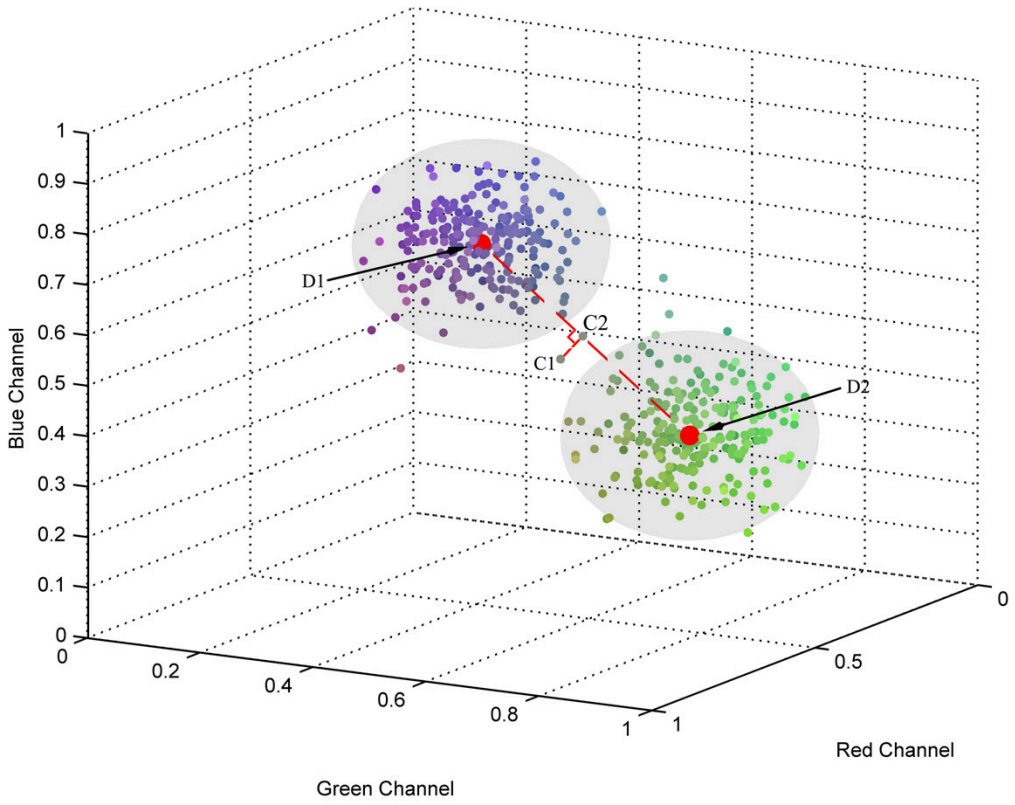


Figure 4.18: Linear combination model representing outlier colors.

color pair to represent each outlier color, the metric

$$R_d(D_{ci}, D_{cj}, OC_m) = \frac{\|OC_m - (\hat{\alpha} D_{ci} + (1 - \hat{\alpha}) D_{cj})\|}{\|D_{ci} - D_{cj}\|} \tag{4.19}$$

is used to examine the linearity among OC_m, D_{ci}, and D_{cj}. The dominant color pair with the lowest R_d value will be chosen to represent the outlier color.

Take the outlier color C_1 in Fig. 4.18 as an example. This color is projected to the connecting line between dominant color D_{c1} and D_{c2}, the projection point C_2 is then used to represent color C_1 in our proposed linear combination model.

By using this linear combination model as a supplement to the mixture Gaussian model, every pixel color in an image can be reliably represented with respect to the color content in this image. In this case, our proposed color model is self-adaptive and

robust to many image distortions. Besides, the pixel color is modeled by probability functions, which can provide important information for further image processing and editing.

4.8 Experimental results

For the performance illustration and the comparison, four standard images as shown in Fig. 4.19 are modeled by using our proposed color representation method.

(a) Image "Lena" (b) Image "F-16"

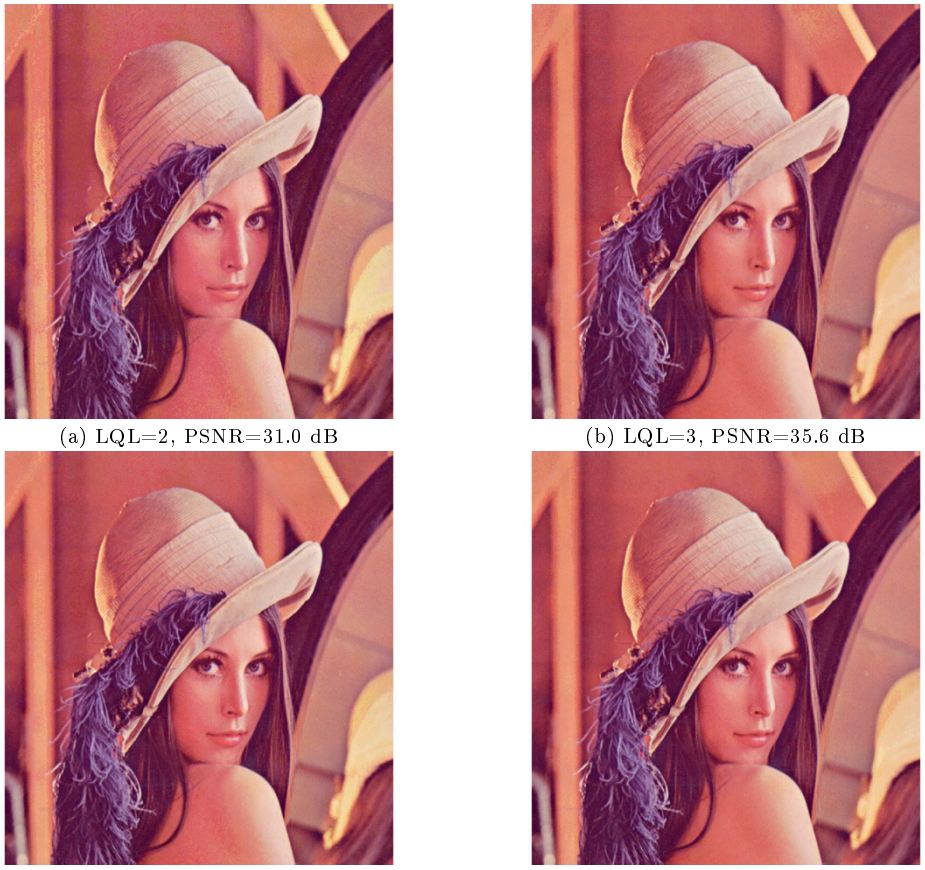
(c) Image "Peppers" (d) Image "Mandrill"

Figure 4.19: Test images for quality comparison.

For each test image, the dominant colors are first extracted and the global color distribution is then approximated by mixture Gaussian functions. Besides, the optimal dominant color pair is also chosen for each outlier color. By doing these, all the pixels in the test image are represented by the mixture Gaussian model and the

linear combination model. For each test image, four different local quantization levels ($LQL = 2, 3, 4, 5$) are used to represent the pixel colors around each dominant color. This means that the Hue and Saturation are respectively divided into LQL segments according to the local GMM distribution.

In Fig. 4.20, Fig. 4.21, Fig. 4.22, and Fig. 4.23, the reconstructed images are presented along with their PSNR values in sRGB color space. It can be observed that the reconstructed images have high quality in visual experience and numeric metrics. Besides, increasing the local quantization level can effectively improve the image quality.



(a) LQL=2, PSNR=31.0 dB

(b) LQL=3, PSNR=35.6 dB

(c) LQL=4, PSNR=36.2 dB

(d) LQL=5, PSNR=37.8 dB

Figure 4.20: Reconstructed "Lena" by using different local quantization levels (LQLs).

Other than providing the reconstructed images by using our proposed method, we also compare our results with the results from other color quantization methods:

(a) LQL=2, PSNR=38.2 dB (b) LQL=3, PSNR=40.4 dB

(c) LQL=4, PSNR=41.9 dB (d) LQL=5, PSNR=43.1 dB

Figure 4.21: Reconstructed "F-16" by using different local quantization levels (LQLs).

uniform quantization (UQ) [78], minimum variance quantization (MVQ) [80], K-means quantization (KMQ) [82], and fuzzy c-means quantization (FCMQ) [83]. The quality comparisons can be found in Table 4.1 and Table 4.2, which list the PSNR values of the reconstructed images in sRGB and CIE-LAB color spaces, respectively.

In Table 4.1 and Table 4.2, the color images are first reconstructed by using color quantization methods (i.e., UQ, MVQ, KMQ, and FCMQ) with different palette sizes (QL). In addition, these color images are also reconstructed by using our proposed model that includes dominant color representation and linear blending. Here ND represents the number of dominant colors found in the color image, and LQL is the local color quantization levels used in our proposed model. Comparing with other color quantization methods, our proposed method can represent the images with

(a) LQL=2, PSNR=34.5 dB (b) LQL=3, PSNR=35.1 dB

(c) LQL=4, PSNR=35.5 dB (d) LQL=5, PSNR=36.0 dB

Figure 4.22: Reconstructed "Peppers" by using different local quantization levels (LQLs).

higher quality while the number of dominant colors are small, which benefits the sampling efficiency in our matting algorithm.

In order to test the robustness of our proposed method, 200 test images from [77] are modeled by using our proposed color representation method. For each test image, the dominant colors are first extracted and the global color distribution is then approximated by the Gaussian mixture functions. Besides, the optimal dominant color pair is also chosen for each outlier color. By doing these, all the pixels in the test image are represented by a mixture Gaussian and linear combination model.

The experiments here will mainly focus on two aspects: quality of the reconstructed image, and the efficiency of the color representation.

(a) LQL=2, PSNR=31.1 dB

(b) LQL=3, PSNR=35.3 dB

(c) LQL=4, PSNR=36.0 dB

(d) LQL=5, PSNR=37.8 dB

Figure 4.23: Reconstructed "Mandrill" by using different local quantization levels (LQLs).

In order to demonstrate the effectiveness of the proposed linear combination model, the reconstructed image quality is presented in Fig. 4.24 with respect to the usage of linear combination and different local quantization levels. If the linear combination is not used, only the Gaussian mixture model is applied to extract the dominant colors and each pixel color in the image will be represented by its nearest dominant color and the associated GMM model. If linear combination is used, the non-outlier pixels will be still represented by their nearest dominant colors while the outlier pixels will be represented by the linear combination model. As shown by the horizontal dashed lines in Fig. 4.24 (a), the average PSNR value of the 200 test images is 37.49 dB with the linear combination model applied while the average PSNR

Table 4.1: The results of PSNR values in sRGB color space

Images	QL	UQ	MVQ	KMQ	FCMQ	Proposed	ND	LQL
Lena	16	16.5 dB	25.7 dB	26.0 dB	26.1 dB	31.0 dB	16	2
	32	20.3 dB	28.4 dB	28.8 dB	28.9 dB	35.6 dB		3
	64	24.4 dB	30.9 dB	31.3 dB	31.4 dB	36.2 dB		4
	128	29.0 dB	33.1 dB	33.5 dB	33.7 dB	37.8 dB		5
F-16	16	15.0 dB	26.7 dB	26.3 dB	27.2 dB	38.2 dB	25	2
	32	18.3 dB	29.8 dB	30.2 dB	30.4 dB	40.4 dB		3
	64	26.9 dB	31.9 dB	32.3 dB	32.4 dB	41.9 dB		4
	128	28.2 dB	33.9 dB	33.9 dB	33.3 dB	43.1 dB		5
Peppers	16	15.9 dB	22.2 dB	22.6 dB	22.6 dB	34.5 dB	32	2
	32	20.0 dB	24.8 dB	25.0 dB	25.0 dB	35.1 dB		3
	64	25.2 dB	27.0 dB	27.3 dB	27.4 dB	35.5 dB		4
	128	28.8 dB	29.1 dB	29.6 dB	30.0 dB	36.0 dB		5
Mandrill	16	16.2 dB	20.6 dB	21.0 dB	21.0 dB	31.1 dB	63	2
	32	20.5 dB	23.0 dB	23.5 dB	23.6 dB	35.3 dB		3
	64	24.7 dB	25.5 dB	26.0 dB	25.9 dB	36.0 dB		4
	128	28.9 dB	27.6 dB	28.1 dB	28.2 dB	37.8 dB		5

reduces to 33.58 dB if the linear combination model is not applied. In Fig. 4.24 (b), application of the linear combination model can still improve the quality of the reconstructed image when more local quantization levels (LQLs) are used. Therefore, it is obvious that the linear combination model can improve the performance of the dominant color approximation.

Besides image quality in PSNR, the efficiency of our proposed method is also demonstrated by comparing the number of dominant colors with the number of total colors in the original image. The number of dominant colors for each image is shown by left y-axis in Fig. 4.25, while the number of total colors in each image is shown by right y-axis in Fig. 4.25. In our experiment for the 200 test images, the number of dominant colors range from 9 to 120 with a mean value of 38.6, and the number of total colors in an image range from 1822 to 19422 with a mean value of 8347. In this case, the huge color redundancy is eliminated by using dominant colors.

From the comparisons in Fig. 4.24 and Fig. 4.25, it can be observed that the quality of reconstructed images is good when only a limited number of dominant

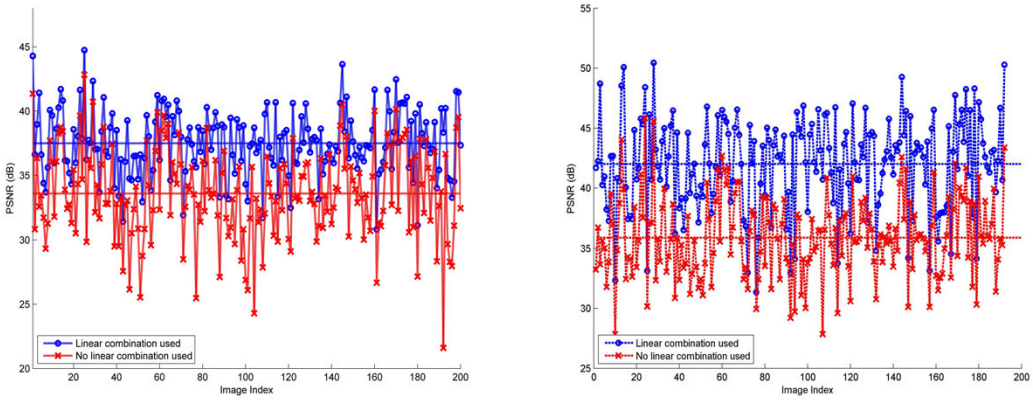
Table 4.2: The results of PSNR values in CIE-Lab color space

Images	QL	UQ	MVQ	KMQ	FCMQ	Proposed	ND	LQL
Lena	16	20.1 dB	27.1 dB	27.8 dB	28.1 dB	35.4 dB	16	2
	32	23.7 dB	30.4 dB	31.4 dB	32.3 dB	39.1 dB		3
	64	28.0 dB	31.6 dB	33.6 dB	34.4 dB	41.7 dB		4
	128	32.2 dB	34.7 dB	36.1 dB	37.2 dB	42.9 dB		5
F-16	16	23.7 dB	28.5 dB	28.3 dB	29.5 dB	42.5 dB	25	2
	32	25.3 dB	30.9 dB	32.1 dB	32.6 dB	44.1 dB		3
	64	30.8 dB	33.7 dB	34.5 dB	35.0 dB	46.3 dB		4
	128	35.4 dB	35.4 dB	35.1 dB	36.7 dB	47.9 dB		5
Peppers	16	21.5 dB	24.8 dB	25.0 dB	24.6 dB	39.0 dB	32	2
	32	24.1 dB	26.1 dB	27.5 dB	27.9 dB	39.8 dB		3
	64	30.9 dB	30.0 dB	29.8 dB	30.7 dB	40.7 dB		4
	128	34.0 dB	31.8 dB	32.7 dB	32.9 dB	41.5 dB		5
Mandrill	16	21.2 dB	22.5 dB	23.1 dB	23.0 dB	35.3 dB	63	2
	32	22.4 dB	25.2 dB	25.4 dB	25.9 dB	39.3 dB		3
	64	25.9 dB	27.7 dB	28.0 dB	28.3 dB	41.6 dB		4
	128	27.8 dB	29.6 dB	30.1 dB	30.7 dB	43.0 dB		5

colors are used to represent the colors for all the pixels in the image.

In addition to the objective comparison for the 200 test images, we also choose two images to provide the visual comparison between the original images and our reconstructed images in Fig. 4.26. The visual difference between the reconstructed image and the original is hardly noticeable, which supports our objective quality results in Fig. 4.24.

Up to now, we have introduced our proposed image color representation model based on global color distribution and linear formation. By extracting the dominant colors in an image, the basic color information of the image is obtained. Based on these dominant colors, the global color distribution is approximated by means of Gaussian mixture models (GMMs). Given the covariance matrix of GMM, the number of unique colors near each dominant color can be reduced based on the Lloyd-Max scalar quantizer. For the outlier color, we further proposed the linear combination model to represent it by two different dominant colors. The experimental results show that our proposed model can reliably represent the original image by



(a) PSNR values of the reconstructed images when local quantization level (LQL) equals to 1.

(b) PSNR values of the reconstructed images when local quantization level (LQL) equals to 3.

Figure 4.24: The PSNR values of the reconstructed images (200 natural images are tested). The horizontal axis represents the image index in the dataset.

a limited number of dominant colors. In this case, the color of an image can be simplified and be robust to noises and distortions. By using this color representation model, the foreground color estimation in chroma keying problem can be done reliably and efficiently, which will be introduced in the next chapter.

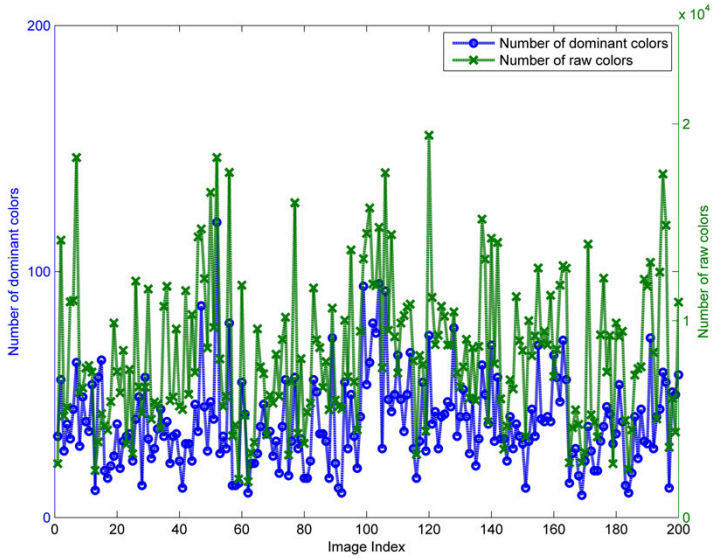


Figure 4.25: Number of dominant colors in the modeled images compared with the number of total colors in the original image.

(a) Original image #49

(b) Original image #166

(c) Reconstructed image (PSNR=36.51dB)

(d) Reconstructed image (PSNR=41.63dB)

Figure 4.26: Visual comparison between original images and reconstructed images. The image numbers in (a) and (b) are the index on horizontal axis in Fig. 4.24.

5

Proposed quad-map based robust chroma-keying

5.1 Introduction to the proposed robust chroma keying system

Chroma-keying is the technique used to replace solid-colored background of images or video frames. This technique is widely used in TV broadcasting, film production, augmented reality and virtual environment. This thesis focuses on proposing a new chroma-keying method, which can automatically remove the background color in an image and accurately segment the foreground objects along with their transparency property. Compared to conventional chroma-keying methods based on color clustering, color difference or thresholding, the proposed method takes into account more

comprehensive mechanisms to robustly generate a high quality alpha map in different background setups. The overall structure of the proposed chroma keying system is shown in Fig. 5.1.

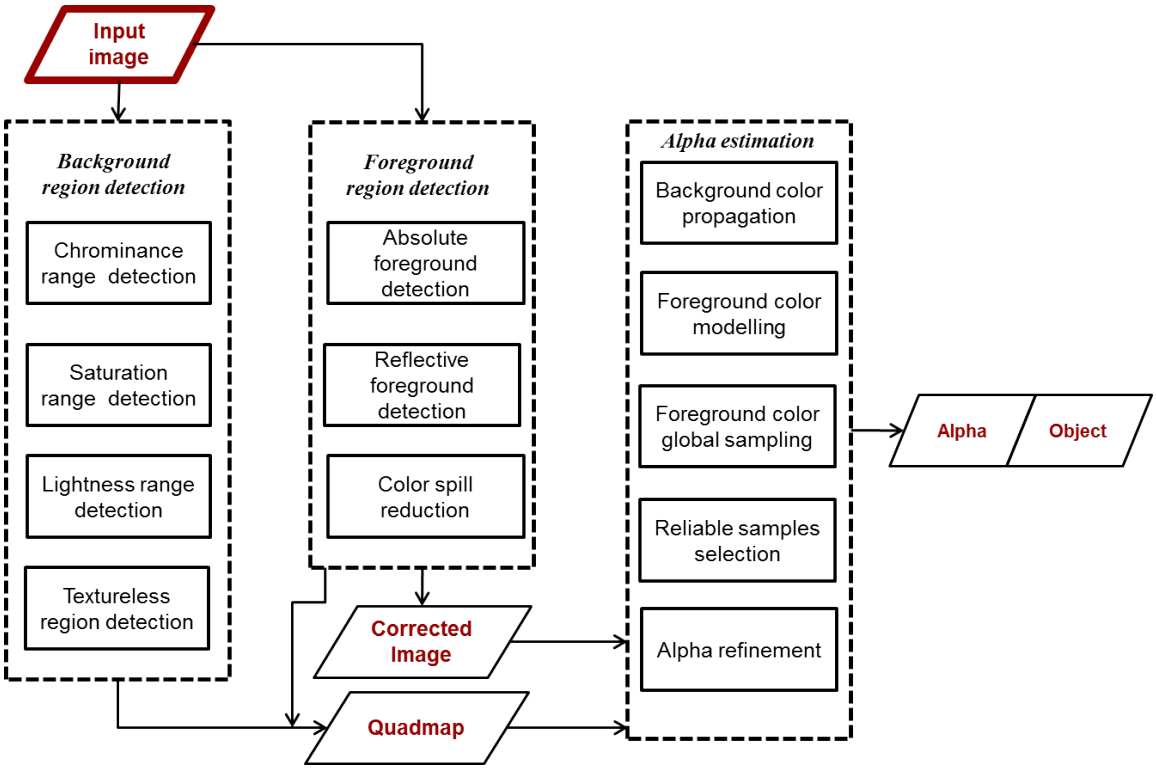


Figure 5.1: The proposed chroma keying system.

As shown in Fig. 5.1, the color (i.e., Hue, Saturation, Lightness) range of the image background is first auto estimated by analyzing the global color distribution. Besides, the spatial entropy is calculated in lightness channel to locate the textureless region of the image. An image pixel will be regarded as background only if its color is in the estimated background color range, and this pixel locates in a textureless region with respect to the spatial entropy.

The foreground region detection involves two parts: absolute foreground detection, and reflective foreground detection. The absolute foreground includes the pixels with chrominance far away from the background color range. The reflective foreground includes the pixels with low saturation and low lightness. We observe that

the color of reflective pixels is often slightly mixed by the background color even though it is not transparent. In this case, color correction is applied to reflective pixels so that the mixed background color can be removed. This process is also called color spill reduction in this thesis. With the background region, absolute foreground region, reflective region, and the remaining transparent region, the input image is segmented into four different regions. This segmentation map is called as "quadmap", which significantly improves the matting result in this thesis. By using quadmap, the proposed chroma keying system can differentiate between transparent and reflective regions. This has always been a challenging problem in conventional chroma-keying or α matting systems. As a result, there can be less constraint for foreground scene used in TV-broadcasting or film making.

In this section, we estimate the α values by solving the following α blending equation:

$$C_{(i,j)} = \alpha_{(i,j)} F_{(i,j)} + (1 - \alpha_{(i,j)}) B_{(i,j)} \,, \tag{5.1}$$

where (i, j) refers to the pixel coordinates, $F_{(i,j)}$ and $B_{(i,j)}$ are the foreground and background colors respectively, and $\alpha_{(i,j)}$ is the blending factor which varies from 0 (completely background) to 1 (completely foreground).

In order to estimate the pixel-wise background color, the known background color is propagated to the entire image by using local affinity. The global foreground color distribution is modeled by using the color representation model we proposed in the last chapter. With the knowledge of global foreground color distribution, a foreground color sampling strategy is proposed. Given the pixel-wise background color and the set of foreground color candidates, a confidence function is used to choose the optimal foreground-background color pair to estimate the α value.

In order to further improve the smoothness of the α map, an affinity based propagation is used at last to iteratively improve the matting result.

In the following sections, the aforementioned parts will be introduced in details.

5.2 Automatic background region detection

As mentioned in Section 1.4, the background in the chroma keying problem is supposed to be solid and simple. However, a completely even background is very difficult to get because of lighting condition, noise, background setup, *etc.* In this case, it is desirable to define a color range that includes most of the background pixels, and excludes all of the foreground pixels. Fixed or manually tunable thresholds are the simplest choice, which may not be the appropriate and convenient choice on the other hand. This is because the background color distributions are different across images even if their basic tone is the same (e.g., green). In this case, a self-adaptive method based on background color statistics is proposed here to estimate the thresholds used for background color determination.

5.2.1 Background detection based on global color variation

The background color range $R = [T_{min}, T_{max}]$ is adaptively specified in this section by estimating thresholds T_{min} and T_{max}. With color range R, the background region can be detected as shown in Fig. 5.2.

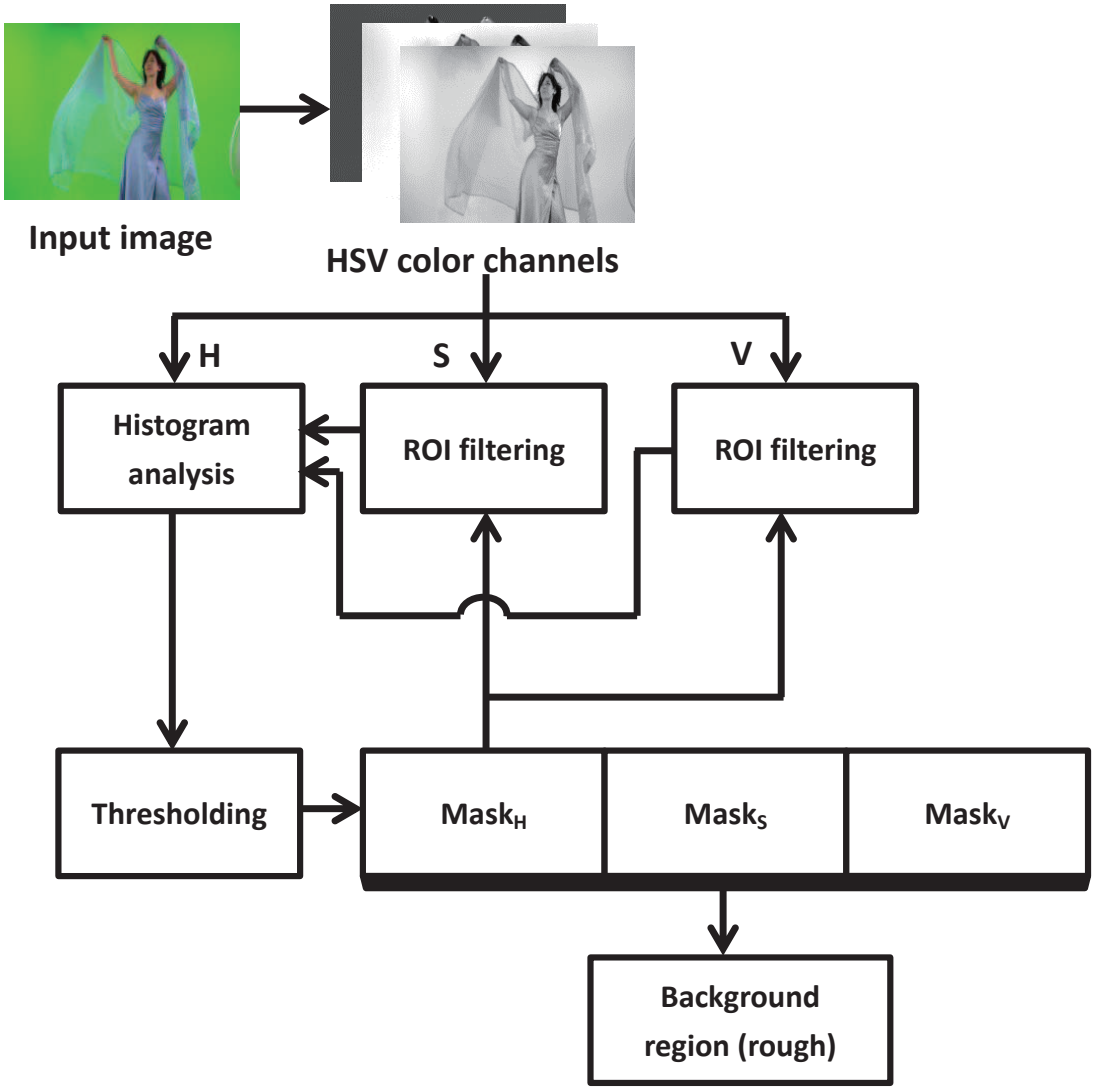


Figure 5.2: Flowchart for background detection (rough).

It is a reasonable assumption that the number of pixels with color in range R is significantly larger than the number of pixels with other colors. In this case, we are supposed to see a manifest "hill" on the color histogram near the range of the background colors. With this assumption, the background color range can be estimated as follows.

Given one channel of the original image, all pixel values are first normalized to the range $[0, 1]$. M bins with interval r are used to uniformly cover the entire range $[0, 1]$. We denote the bins by their centres' locations as presented in vector

$\mathbf{c^0} = \{c_i^0 | i = 1, 2, \ldots, M\}$. The range for each bin is $[c_i^0 - r/2, c_i^0 + r/2]$. The total number of pixels with values falling into each bin is counted as occurrence frequency $p(c_i^0)$. The overall statistical color distribution of original image is represented by vector $P(\mathbf{c^0}) = \{p(c_i^0) | i = 1, 2, \ldots, M\}$, which is the histogram of the image. $P(\mathbf{c^0})$ will dramatically change at bins whose locations are in the range $R = [T_{min}, T_{max}]$. The background color range R is estimated as follows.

First, we define an empty set CR^0, which is updated in each iteration. The bins to be analyzed is initially $\mathbf{c^0}$ and updated in each iteration. If we denote the bins to be analyzed in kth iteration as $\mathbf{c^k} = \{c_i^k | i = 1, 2, \ldots, N^k\}$, their associated occurrence frequency distribution $P(\mathbf{c^k})$ is shown as follows:

$$\begin{aligned} P(\mathbf{c^k}) &= P(\mathbf{c^{k-1}} | c_i^{k-1} \notin CR^{k-1}) \\ &= \{p(c_i^k) | i = 1, 2, \ldots, N^k)\}. \end{aligned} \tag{5.2}$$

Note that $\mathbf{c^k}$ is always a subset of $\mathbf{c^0}$. Given vector $P(\mathbf{c^k})$, its variance v^k is calculated by using Equ. (5.3):

$$v^k = \frac{\sum_{c_i^k = r/2 | c_i^k \notin CR^{k-1}}^{1-r/2} \left(p(c_i^k) - P_{ave}^k \right)^2}{N^k}, \tag{5.3}$$

where c_i^k is the values in vector $\mathbf{c^k}$, P_{ave}^k is the average of vector $P(\mathbf{c^k})$, N^k is the number of bins not in set CR^{k-1}, r is the interval of each bin.

The following two conditions are used to determine when iterations should be ended:

$$v^k < v_{amp} * v^0, \tag{5.4}$$

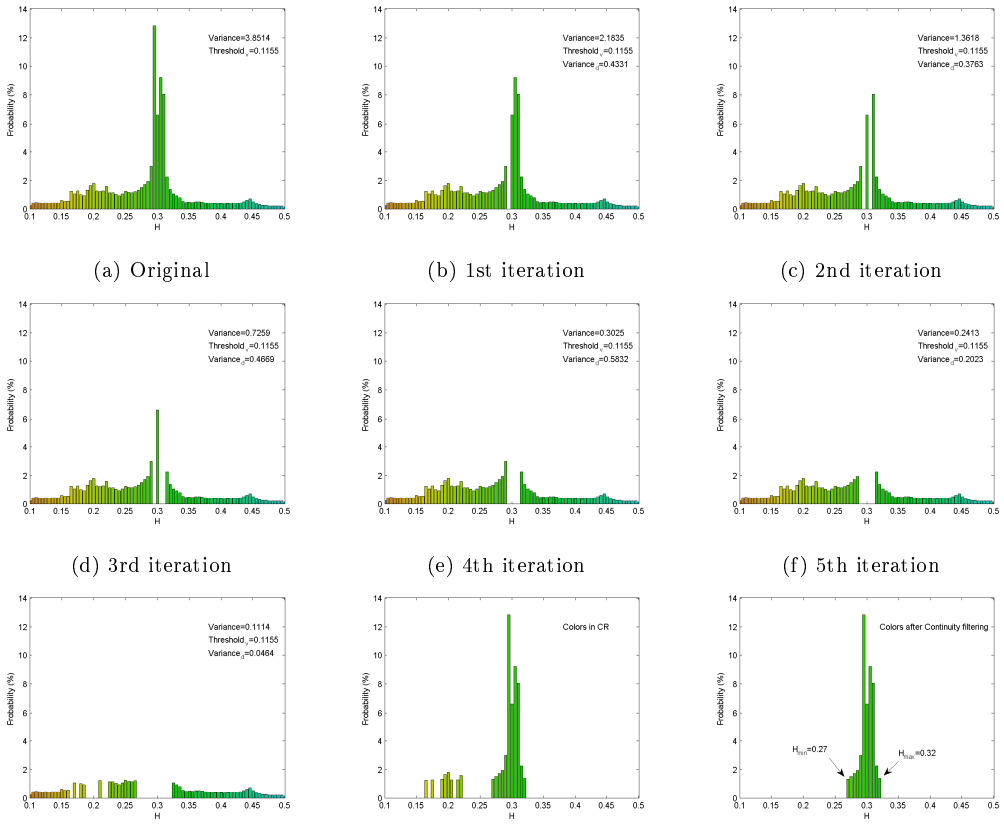
and

$$\frac{|v^k - v^{k-1}|}{v^{k-1}} < T_{grad}\,, \tag{5.5}$$

where v^0 is the variance of $P(\mathbf{c^0})$ for original data, v^k is the variance of $P(\mathbf{c^k})$ in kth iteration, v_{amp} is a constant factor that ensures the current variance is small enough, T_{grad} is a constant value that ensures the variance change between two iterations is small enough.

If the specified conditions are not met, set CR^{k-1} is updated to set CR^k by adding into it the bin c^k_{max} with highest occurrence frequency $p(c^k_{max})$. Then the estimation proceeds to $(k+1)th$ iteration. If conditions are met at Mth iteration, the final set $CR^{M-1} = \{c^1_{max}, c^2_{max}, \ldots, c^{M-1}_{max}\}$ is used to estimate the thresholds. Given CR^{M-1}, only consecutive bins including the one with highest occurrence frequency are kept to obtain CR^f. Finally, the thresholds are estimated by $T_{min} = \min(CR^f)$, $T_{max} = \max(CR^f)$.

In order to demonstrate histogram analysis intuitively, thresholds estimation in H channel of image "Roto" is as shown in Fig. 5.3. In each figure, "$Variance$" is the numeric value of v^k in each iteration; "$Threshold_v$" is the variance threshold calculated from the right side of Equ. (5.4); "$Variance_d$" is the numeric value calculated from the left side of Equ. (5.5) in each iteration. The color of each histogram bin represents the color of the corresponding Hue value. In each iteration, the bin with the highest occurrence frequency is removed until the conditions are met. Fig. 5.3 (h) shows all bins in final set CR^{M-1}. Fig. 5.3 (i) shows consecutive bins CR^f in set CR^{M-1}. And the background color thresholds H_{min} and H_{max} in H channel are obtained as shown in Fig. 5.3 (i).



(a) Original (b) 1st iteration (c) 2nd iteration

(d) 3rd iteration (e) 4th iteration (f) 5th iteration

(g) 19th iteration (h) candidates (i) back color range

Figure 5.3: Histogram analysis for background region detection.

By applying the proposed histogram analysis method to each HSV channel, the background region can be extracted. The overall working procedures to roughly segment background region are as shown in Fig. 5.2. Given an image, the pixel values are transformed to HSV color space. First, the Hue values close to background color (e.g., 0.1-0.5 for green) are histogram analyzed to estimate Hue thresholds H_{min} and H_{max}. A background map $Mask_H$ is consequently generated according to $R_H = [H_{min}, H_{max}]$:

$$Mask_H(i,j) = \begin{cases} 1 & H(i,j) \in R_H\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{5.6}$$

Afterwards, the S and V channels are respectively ROI (region of interest) filtered according to $Mask_H$. The remaining pixels in S and V channels are also respectively histogram analyzed. The color range (i.e., $R_S = [S_{min}, S_{max}]$ and $R_V = [V_{min}, V_{max}]$) in S and V channels are then estimated to generate $Mask_S$ and $Mask_V$:

$$Mask_S(i,j) = \begin{cases} 1 & S(i,j) \in R_S, Mask_H(i,j) == 1\,, \\ 0 & \text{otherwise}\,, \end{cases} \tag{5.7}$$

$$Mask_V(i,j) = \begin{cases} 1 & V(i,j) \in R_V, Mask_H(i,j) == 1\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{5.8}$$

Finally, the background mask can be generated by $Mask_{back}(i,j) = Mask_H(i,j) * Mask_S(i,j) * Mask_V(i,j)$.

5.2.2 Background refinement based on local color entropy

In some situations, background detection solely based on histogram analysis is not enough. It is because some tiny or severely transparent part at the boundary of foreground object may be detected as background. Three images (see Fig. 5.4) with such fuzzy boundary will be used to illustrate this problem. Some of the fuzzy boundaries in the test images are marked by the red rectangles.

Figure 5.4: Test images with fuzzy boundary.

The detected background region in the HSV color range determined by our proposed histogram analysis method is shown in Fig. 5.5, Fig. 5.6, and Fig. 5.7. It is not difficult to see that the fuzzy boundary, such as the hair floating in the air, is erroneously detected as the background. This is because the color of the hair is very similar to the background, especially with the shadow regions in the background. In this case, the detected background needs to be further refined.

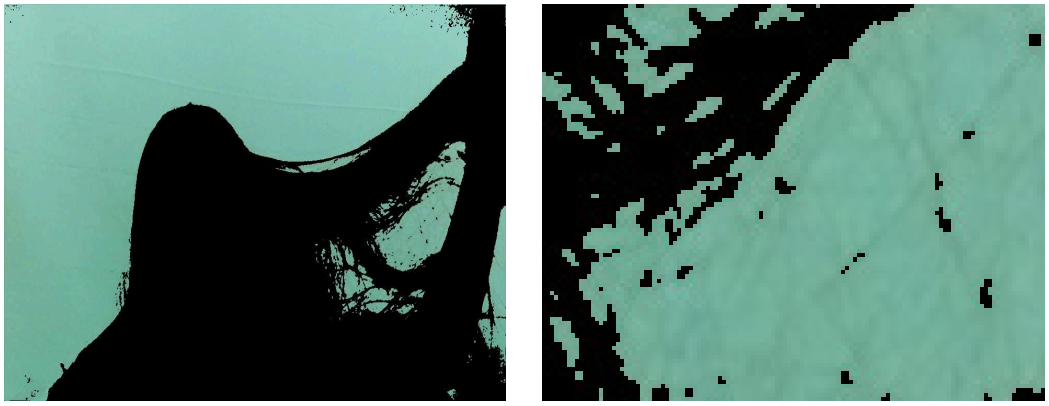


Figure 5.5: The detected background region of the first image in Fig. 5.4 by just using the proposed histogram based color range determination. The enlarged part is the marked region in Fig. 5.4.

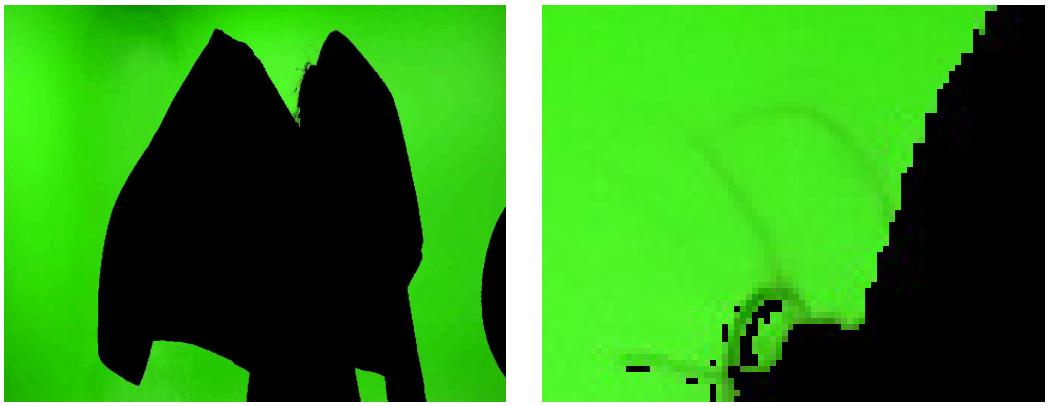


Figure 5.6: The detected background region of the second image in Fig. 5.4 by just using the proposed histogram based color range determination. The enlarged part is the marked region in Fig. 5.4.

Despite the color similarity between the fuzzy edge and the background, there are always lightness variations at the fuzzy edge. And this is also the reason why

Figure 5.7: The detected background region of the third image in Fig. 5.4 by just using the proposed histogram based color range determination. The enlarged part is the marked region in Fig. 5.4.

humans can notice such fuzzy, tiny or thin structure. Here we propose to use two metrics to evaluate the local lightness variation: lightness gradient, and lightness spatial entropy.

The gradient calculation is as presented in Equ. (5.9):

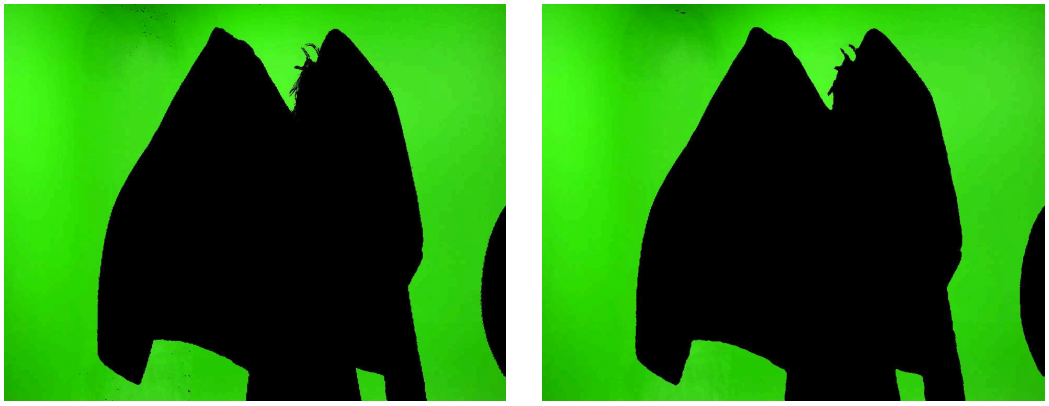
$$G_{(x,y)} = \left\| \left(\frac{\partial V_{(x,y)}}{\partial x}, \frac{\partial V_{(x,y)}}{\partial y} \right) \right\| , \tag{5.9}$$

where (x, y) is the image coordinates, and $V(x, y)$ is the lightness value of the pixel. If a pixel belongs to background, the lightness gradient is very likely to be low because the background is normally flat.

The spatial entropy, which measures the randomness of the variable in a block of the image, tends to be low in the background because the lightness does not vary much within a background block. The spatial entropy can be calculated as follows:

$$E_{B_i} = -\sum_{j=1}^{j=N} P(V_j) log_2(P(V_j)) , \tag{5.10}$$

where B_i is the ith block in the image, N is the number of different lightness values in this block, $P(V)$ is the occurrence frequency of V.



(a) Background region after fuzzy removal using gradient

(b) Background region after fuzzy removal using entropy

(c) Enlarged part of (a)

(d) Enlarged part of (b)

Figure 5.8: The detected background region of the second image in Fig. 5.4 after removing the fuzzy edge. The enlarged part in (c) (d) are the marked region in Fig. 5.4.

In Fig. 5.8, both of the gradient based and entropy based methods can give us a clean background without having fuzzy edges. At the same time, it is also observed that gradient based method is easier to be affected by the image noise. In this case, we choose to use entropy based method in this thesis to refine our background detection.

5.3 Foreground region detection

After the background region is detected, we come to detect the foreground region in this section. The foreground detection involves two parts: absolute foreground detection and reflective foreground detection, which will be introduced as follows.

5.3.1 Absolute foreground region detection

The underlying principle behind absolute foreground detection is quite straightforward. Since the absolute foreground should be with colors which are far enough from the background color. Meanwhile, the background hue range $[H_{MIN}, H_{MAX}]$ has already been estimated in Section 5.2.1. In this case, we can use a simple threshold based method to efficiently extract the foreground regions with color absolutely different from the background. Specifically, we use a predefined threshold T_f to control the hue distance between absolute foreground and background. Because Hue values are continuous in a circle ranged from 0 to 1, the foreground color determination involves modulo operation. Therefore, the foreground color range FR is determined by a piecewise function

$$FR = \begin{cases} [0, H_{MIN} - T_f] \bigcup [H_{MAX} + T_f, 1]; & H_{MIN} > T_f, H_{MAX} < 1 - T_f\,, \\ [H_{MAX} + T_f - 1, H_{MIN} - T_f]; & H_{MAX} > 1 - T_f\,, \\ [H_{MAX} + T_f, 1 + H_{MIN} - T_f]; & H_{MIN} < T_f\,, \end{cases} \tag{5.11}$$

where H_{MIN} and H_{MAX} are the lower and upper bound of the background Hue range, which are estimated in Section 5.2.1.

(a) Original image (b) Detected foreground

Figure 5.9: The detected absolute foreground region based on Hue distance.

As shown in Fig. 5.9 (b), it is the detected foreground region based on Hue distance to the background color. It is obvious that this foreground extraction is not satisfactory because large part of the foreground object is not detected based on Hue distance. This problem is caused by lack Hue information in foreground regions such as dark hair and white sweater. In this case, we further propose a method to deal with a foreground region that has limited chrominance information.

5.3.2 Reflective foreground region detection and color spill reduction

As indicated by its name, the chrominance information is the most important cue to separate the background from the foreground in a chroma keying system. However, chrominance information is not sufficient to completely detect the foreground region because there could be colorless (i.e., grayish) regions in the foreground object. And unfortunately, such colorless region is not rare in natural images, such as black hair, and white clothes.

In order to extract the colorless foreground region, it is desirable to define what a "colorless" color is. A completely gray color with equal R, G, B values is colorless. However, this constraint is too strict in practice and we hope to detect the colorless pixels according to human sensation. In this case, we define a Saturation threshold function $S(V)$ as shown in Fig. 5.10 according to our visual experiments by human viewers. In Fig. 5.10, there is a saturation threshold $S(V)$ corresponding to every lightness level V. Given a pixel with lightness V_i, it will be regarded as colorless if its saturation S_i is smaller than the corresponding saturation threshold $S(V)$

In order to determine the saturation threshold function shown in Fig. 5.10, we invited five human viewers to view the color plates under different lightness levels. In our visual experiment, these views told us the saturation threshold under which there is insufficient chrominance information. By doing this separately under different lightness levels, we can obtain a saturation threshold function as shown in Fig. 5.10. Note that the final function we used here is the average from the results of our five

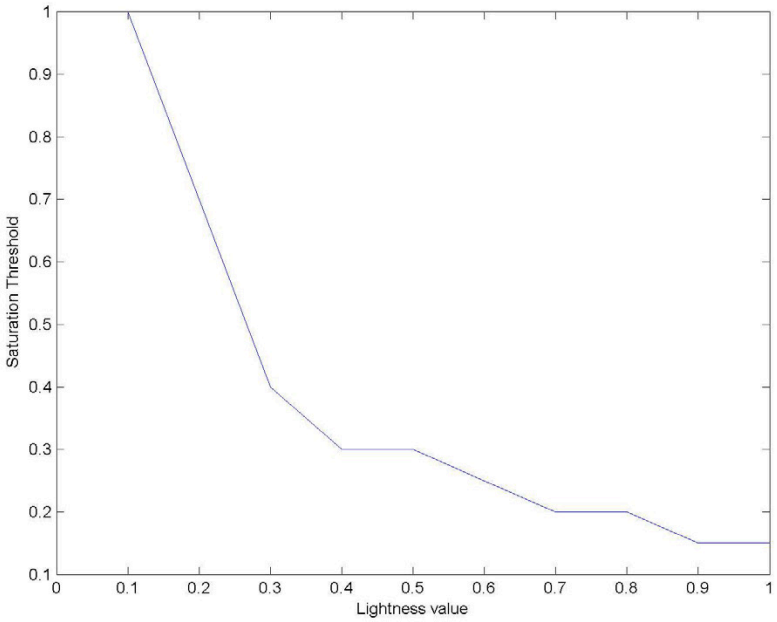
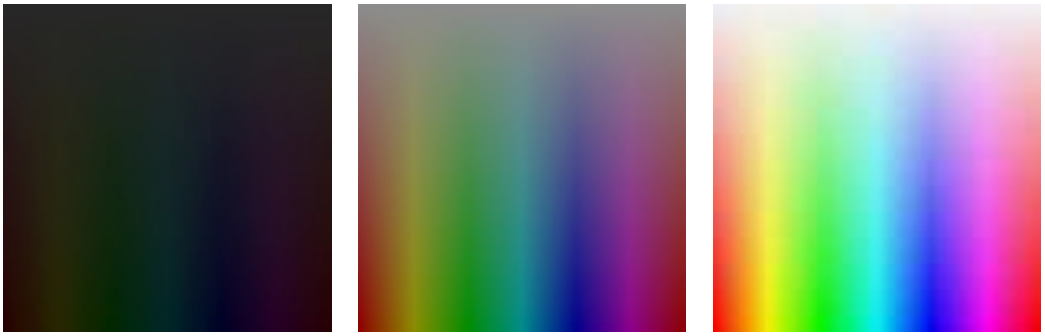


Figure 5.10: The Saturation thresholds for colorless pixels under different lightness levels.

volunteers. And this visual experiment can also be done in different human visual system based color spaces, such as Lab, YUV, $YCbCr$. In this thesis, we choose to use HSV color space because our other color analysis is done in this space with the concern of efficient implementation. Some examples of these color plates are as shown in Fig. 5.11. In our visual experiment, 20 color plates under different lightness levels ($V_i \in 0.05, 0.1, 0.15, \ldots, 0.95, 1$) were used.

Given an input image, we can now estimate a saturation threshold for each pixel to determine whether it is colorless or not based on its lightness level. In the example of Fig. 5.12 (b), we provide a saturation threshold map. The pixel values in Fig. 5.12 (b) are the saturation thresholds for every pixel, depending on their lightness value. It can be observed that the saturation thresholds tend to be high for dark pixels on the hair. This is reasonable for human vision because we prefer to consider dark pixels to be less vivid. In this case, a dark pixel is more likely to be colorless and therefore its saturation threshold should be higher.



(a) H-S color plate with lightness=0.15 (b) H-S color plate with lightness=0.55 (c) H-S color plate with lightness=0.95

Figure 5.11: The example of color plates used for threshold determination of grey color saturation. In each color plate, the Hue varies from 0 to 1 from left to right; the Saturation varies from 0 to 1 from top to bottom.

Given the actual S channel of the image, and the saturation thresholds map ST as shown in Fig. 5.12 (b), we can calculate the gray confidence (GC) of each pixel by Equ. (5.12):

$$GC_{(x,y)} = ST_{(x,y)} - S_{(x,y)}\,, \tag{5.12}$$

where (x, y) is the image coordinates, S is the image saturation, and ST is the saturation thresholds map. A pixel would tend to be colorless if its GC value is high.

As shown in Fig. 5.13, most of the colorless foreground can be extracted by using our proposed saturation thresholds function. By observing the extracted foreground in Fig. 5.13 (b), we find the "colorless" foreground is actually not real colorless, especially considering the greenish environment in this image. For dark pixel, its intrinsic lightness and chrominance are both very low, which makes the pixel color very easy to be affected by the environment, such as the green color of the background. In Fig. 5.13 (b), it is not difficult to find that the hair is not completely black or brown, instead it looks greenish which can not be the intrinsic color of the hair of an Asian girl. Technically, this phenomenon is also called as **color spill, which represents the color reflecting from the back screen and casting a noticeable tint on**

(a) Original image (b) The saturation threshold of each pixel

Figure 5.12: The pixel-wise Saturation thresholds based on the pixels' lightness. In (b), the intensity value at each pixel location represents the corresponding Saturation threshold.

the foreground object.

In order to suppress the color spill, we propose the following suppression function:

$$G_{(x,y)} = \max(R_{(x,y)}, B_{(x,y)}), \text{ if } G_{(x,y)} > \max(R_{(x,y)}, B_{(x,y)}), \tag{5.13}$$

where the green intensity of a colorless pixel is suppressed if the value of G channel is the largest one among R, G, B channels. By using this color spill suppression function, the greenish effect on the foreground object can be significantly removed. In Fig. 5.14, we present the full foreground object, including the absolute foreground and the gray foreground after color spill. Compared with the foreground in Fig. 5.13 (b), the hair in Fig. 5.14 is no longer affected by the green background, which provides us more reliable foreground color information.

Since the colorless region of the foreground considered in this section is often affected by the reflecting light from the background, **we also call this colorless region as reflective region** in this thesis.

In this case, we divided the image to be chroma keyed into four different regions: background, absolute foreground, reflective foreground, and the remaining unknown (i.e., transparency). Such segmentation is called a **quadmap**, which can significant-

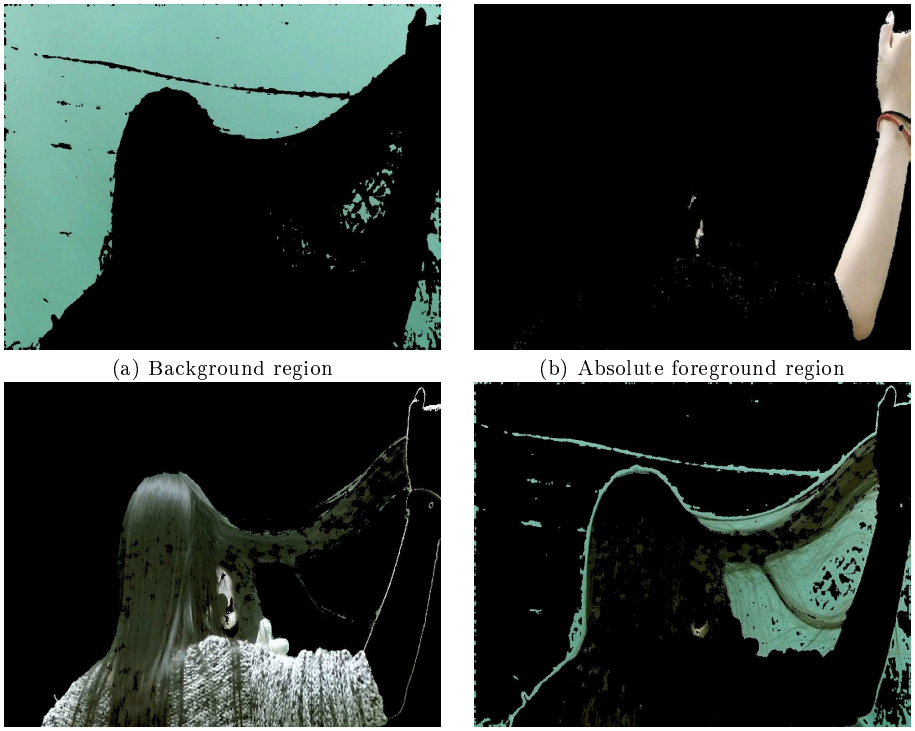
(a) GC map calculated from Equ. (5.12)

(b) The extracted colorless foreground by taking pixels with GC values higher than 0.05

Figure 5.13: The gray confidence map and the extracted colorless foreground.

ly improve the keying result because it differentiates the reflective part from the unknown part. An example of these four image regions are as shown in Fig. 5.15. We only need to estimate the α values in the remained unknown region in Fig. 5.15 (d). It can be observed that although most of the reflective region is correctly identified as foreground, a small part of the reflective foreground region (e.g., hair) is still grouped into the unknown region. In this case, we need to estimate the α values in these regions. If an α value not close to 1 is assigned to pixels in these region, it will introduce significant visual artifact after image compositing because the foreground object will be partially transparent. In order to avoid this problem as much as possible, we carefully designed the method for α value estimation, which will be introduced in the following section.

Figure 5.14: The complete foreground after color spill suppression.



(a) Background region

(b) Absolute foreground region

(c) Reflective foreground region

(d) The remained unknown region where α values needs to be estimated

Figure 5.15: The quadmap segmentation of the image to be chroma keyed.

5.4 Alpha channel estimation

From the aforementioned proposals, the image can already be automatically segmented into four different regions and only the α values of pixels in unknown regions need to be estimated. This problem is still highly under constrained because we need to solve 2 color vectors (i.e., F and B) and one α value from just one equation as presented by Equ. (5.1).

In this case, we choose to estimate the background and foreground color for each pixel ahead of the estimation of α value in this thesis. The detailed description for background/foreground color estimation and the final α calculation will be introduced as follows.

5.4.1 Background color propagation

Despite that the background color in images/frames to be chroma keyed is simple, we still can not use a constant background color for all pixels in the image. This is because the background color can change along with lighting condition, noise, and uneven background surface. In this case, we propose in this section that the background color is estimated by propagating background color from known background region to the whole image based on local affinity, which is modeled by a Laplacian equation as shown in Equ. (5.14):

$$\Delta f(x,y) = \frac{\partial^2 f(x,y)}{\partial^2 x} + \frac{\partial^2 f(x,y)}{\partial^2 y} = 0\,. \tag{5.14}$$

In the discrete domain, Equ. (5.14) refers that a pixel value equals to the average value of its neighboring pixels. In this case, the background color can be estimated by minimizing Equ. (5.15):

$$E = \gamma \sum_{i \in back} \Big(b_i - B_i\Big)^2 + \sum_{i=1}^{M} \Big(b_i - \sum_{j \in N_i} W_{i,j} b_j\Big)^2, \tag{5.15}$$

where i is the index for all pixels in the image, M is the number of all pixels in the image, N_i is the neighbors of pixel i, b_i is the estimated background color, B_i is the known background color, and γ is a large control factor ensuring the consistency between the estimated background and the known background. The weight $W_{i,j}$ between neighboring pixels is determined by Equ. (5.16):

$$W_{i,j} = \begin{cases} 1/3 & j \in N_i,\ i \text{ locates at image corner,} \\ 1/5 & j \in N_i,\ i \text{ locates at image side,} \\ 1/8 & j \in N_i,\ i \text{ does not locate at image side,} \\ 0 & j \notin N_i. \end{cases} \tag{5.16}$$

The energy function in Equ. (5.15) can be further written in the matrix form

$$\begin{aligned} E = (B_{est} - B_{known})^T \Lambda (B_{est} - B_{known}) ... \\ + ((I - W)B_{est})^T (I - W)B_{est}\,, \end{aligned} \tag{5.17}$$

where I is $M \times M$ identity matrix, B_{est} is an $M \times 1$ matrix containing estimated background color, B_{known} is an $M \times 1$ matrix containing known background color. Λ is an $M \times M$ matrix that is defined by Equ. (5.18):

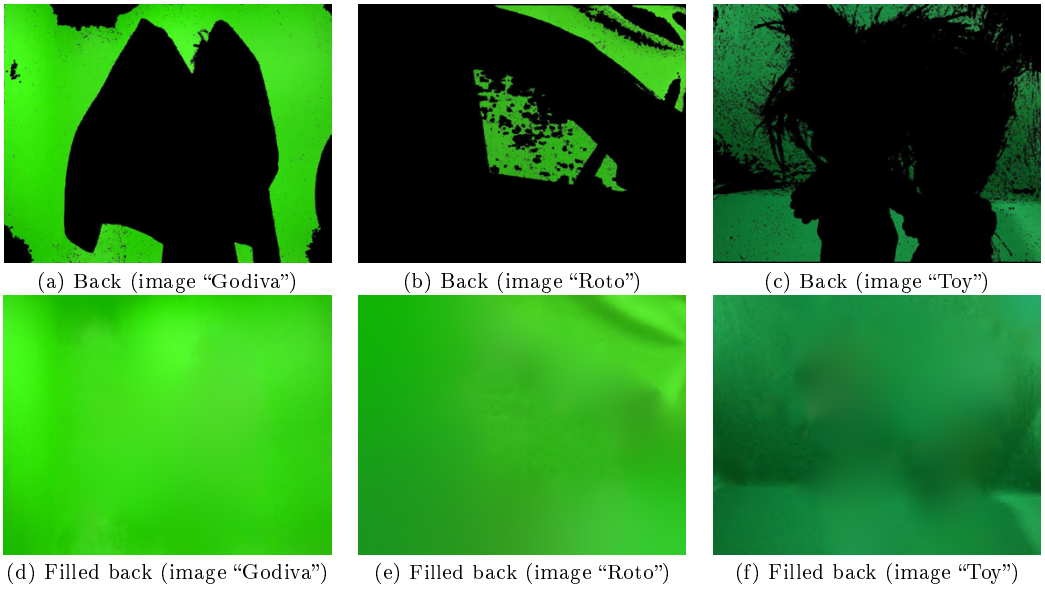
$$\Lambda_{ii} = \begin{cases} \gamma & i \in \text{back,} \\ 0 & i \notin \text{back,} \end{cases} \tag{5.18}$$

where γ is a large constant value which guarantees the consistency between estimated and known background.

Finally, matrix B_{est} can be calculated by

$$B_{est} = [(I - W)^T (I - W) + \Lambda]^{-1} \Lambda B_{known}\,. \tag{5.19}$$

Therefore, the background color is propagated to the whole image as shown in Fig. 5.16.



(a) Back (image "Godiva") (b) Back (image "Roto") (c) Back (image "Toy")

(d) Filled back (image "Godiva") (e) Filled back (image "Roto") (f) Filled back (image "Toy")

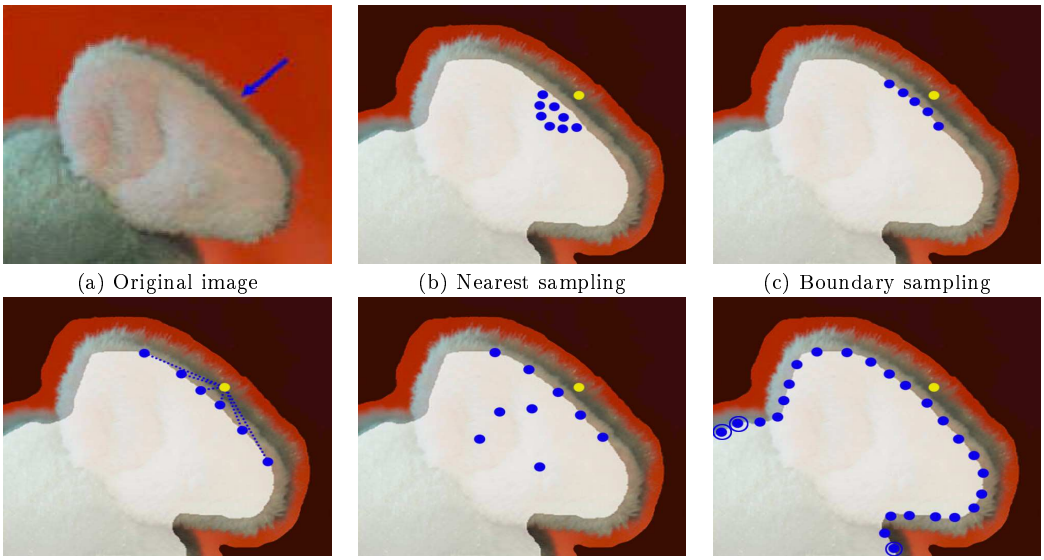
Figure 5.16: Background propagation.

5.4.2 Global foreground color modeling and sampling

In this section, we introduce our proposed foreground estimation strategy for each pixel in the unknown region. Conventionally, foreground color is estimated by choosing samples from known foreground regions. Different sampling methods were presented as shown in Fig. 5.17. In Fig. 5.17 (b) - (f), the bright region refers to foreground region; the dark region refers to background region; the yellow dot represents an unknown pixel whose α value needs to be estimated; the blue dots represent foreground color candidates. In Fig. 5.17 (b), foreground samples are selected from the nearest locations in the foreground region; In Fig. 5.17 (c), foreground samples are selected along boundary between foreground and transparent regions; In Fig. 5.17 (d), a set of rays are drawn from the unknown pixel, and the intersection points of rays and foreground boundary are selected as foreground samples; In Fig. 5.17 (e), foreground samples are not only selected along boundaries but also selected in foreground re-

gions; In Fig. 5.17 (f), foreground samples are globally selected along the foreground boundaries.

In the special case as shown in Fig. 5.17, a reliable foreground color (i.e., dark grey in this case) cannot be found by using local sampling methods as shown in Fig. 5.17 (b) - (e). Instead, only global sampling with enough searching range can find correct foreground color samples as shown by blue circles in Fig. 5.17 (f). The drawback for global sampling is the high computational cost. In order to realize the global sampling as shown in Fig. 5.17 (f), hundreds and even thousands of foreground samples may be chosen. This will become a huge burden when we come to look for the best foreground-background sample pair for each unknown pixel.



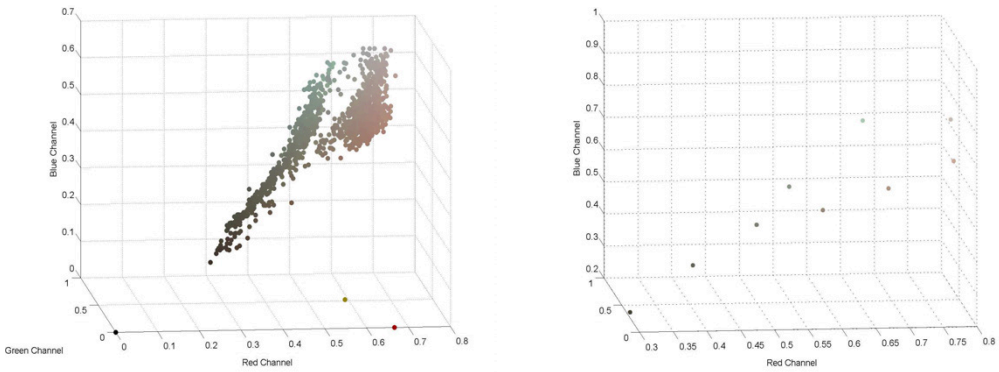
(a) Original image (b) Nearest sampling (c) Boundary sampling

(d) Shared sampling (e) Comprehensive sampling (f) Global sampling

Figure 5.17: Different foreground sampling methods.

In this case, we propose in this section a new foreground color estimation strategy, which is different from sampling. As we proposed in Chapter 4, the major color component, which is also called as the dominant colors, can be reliably extracted from the natural image. In this case, we can directly use these dominant colors as our foreground color candidates, instead of sampling in the image. Compared with local sampling, our dominant colors can represent much more comprehensive

foreground color candidates. Compared with global sampling, which often involves hundreds of foreground samples, there are normally only 10-30 different dominant colors in a natural image, which significantly reduce the computational cost. Take Fig. 5.18 as an example. The global color distribution of the foreground object in Fig. 5.17 (a) is presented in Fig. 5.18 (a). Meanwhile, the extracted dominant colors are presented in Fig. 5.18 (b). In this case, there are only 9 dominant colors used as foreground color candidates. On the other hand, we can directly see from the original image that there are two major chrominance in the image: greyish pixels at the ear boundary and the elephant head, the pink pixels on the elephant ear. If we look back at the dominant colors in Fig. 5.17 (b), it is not difficult to find that these dominant colors are just grayish colors and pink colors with different luminance. Therefore, the dominant colors can be reliable foreground color candidates.



(a) The global color distribution of the foreground object in Fig. 5.17 (a)

(b) The extracted dominant colors of the foreground object in Fig. 5.17 (a)

Figure 5.18: Dominant colors of the foreground object.

Although we have the reliable dominant colors D_F for the foreground object, it is still not a good idea to directly use them as the foreground color candidates. This is because the actual colors in the image always varies around its dominant color. This is easy to understand since it is unlikely that there would be only 9 different colors in image Fig. 5.17 (a).

In order to give the most reliable foreground color candidates for each unknown

pixel. We refine the color of each dominant color when we deal with different unknown pixels. Given all the foreground pixels, they are already grouped into N subsets $\{F_i|i = 1, 2, \ldots, N\}$ with respect to its nearest dominant colors, the value of N is the number of dominant colors. For each subset F_i, it corresponds to one dominant color D_{F_i}. The foreground pixels in F_i are the nearest pixels to D_{F_i} in the sense of GMM probability as we proposed in Chapter 4.

When we come to deal with one unknown pixel U, we choose the nearest pixel to U in each of the foreground subset $\{F_i|i = 1, 2, \ldots, N\}$ as the final foreground candidates. By doing this, the number of foreground color candidates is still the number of dominant colors. Besides, the color of the candidates are not exactly the same as the dominant colors. They are more suitable choices because these colors are chosen from the geometrically nearby pixels. As shown in Fig. 5.19, the blue dots are the foreground color candidates chosen for the unknown pixel (the yellow one). By using our proposed foreground color sampling method, a comprehensive color set can be obtained because it was already shown in Chapter 4 that the dominant colors can reliably represent the colors in the whole image.

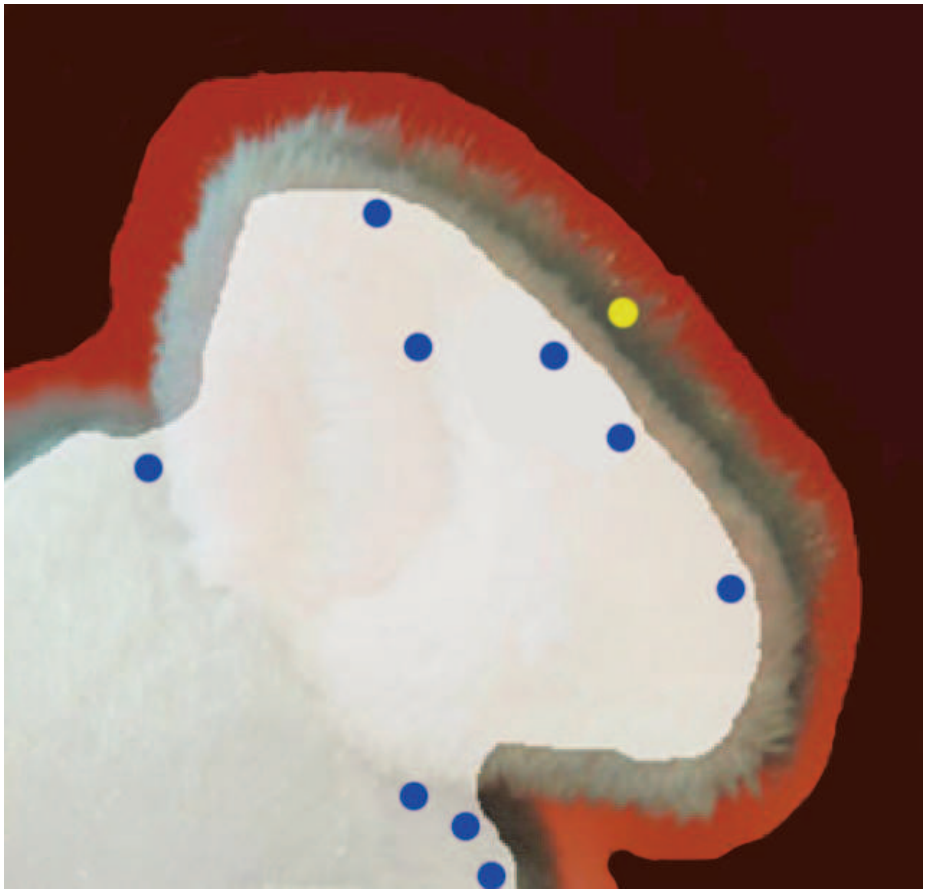


Figure 5.19: The proposed foreground color selection.

5.4.3 Linear cost

Now the background color and foreground color candidates are all prepared for each unknown pixel. The last step is to choose the best foreground-background color pair, so that the α value can be calculated by Equ. (5.20):

$$\hat{\alpha}_{(x,y)} = \frac{(C_{(x,y)} - B_{(x,y)})(F_{(x,y)} - B_{(x,y)})}{\left\|F_{(x,y)} - B_{(x,y)}\right\|^2}, \tag{5.20}$$

where (x, y) is the image coordinates, $\hat{\alpha}_{(x,y)}$ is the estimated α value, $C_{(x,y)}$ is the color of an unknown pixel; $B_{(x,y)}$ and $F_{(x,y)}$ are the background and foreground color of this unknown pixel.

As shown in Fig. 5.20, the foreground and background pair (F_2, B_2) is a better

choice than (F_1, B_1) for the unknown pixel P_c because (F_2, B_2, P_c) have better linear relationship and better fit Equ. (5.1).

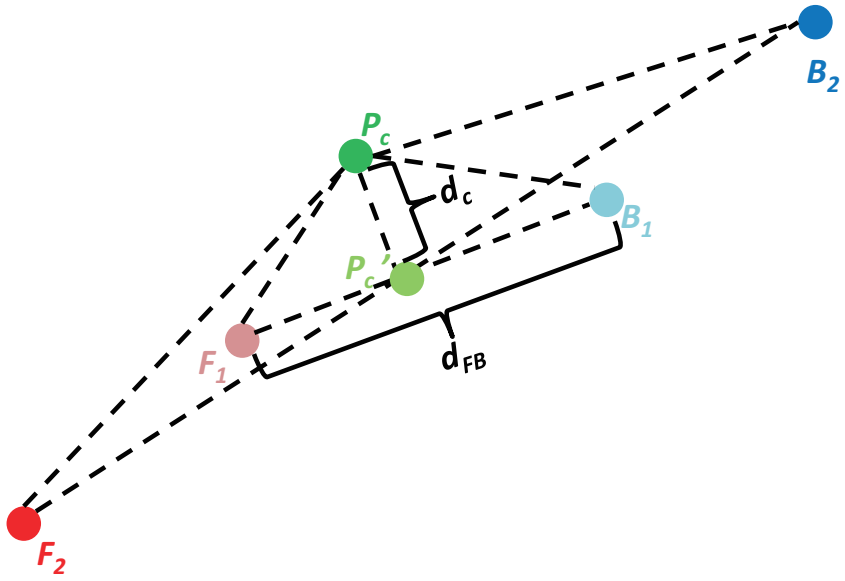


Figure 5.20: Linear relationship between foreground, background and unknown pixel.

In this case, the reliability of a foreground-background sample pair is measured by the linear cost R_d:

$$R_d(F^i, B^j) = \frac{\|C - (\hat{\alpha}F^i + (1-\hat{\alpha})B^j)\|}{\|F^i - B^j\|}, \tag{5.21}$$

where F^i and B^j are the foreground and background color candidates, C is the color of an unknown pixel, $\hat{\alpha}$ is the estimated α by using F^i and B^j. This linear cost becomes lower if there is better linear relation among the background, the foreground, and the unknown color. The foreground-background sample pair with the lowest linear cost will be selected to calculate the final α value.

5.5 Results of alpha estimation

In this part, we present the alpha mattes generated by our proposed method. These results will be further analyzed by comparing them with the results from other matting algorithms. The methods used to be compared include both recently published academic researches (i.e., closed form matting [65] and WCT matting [58]) and industry-established chroma-keyers in commercial software (i.e., Keylight [86] and Primatte [87]).

5.5.1 Visual quality comparison

Firstly, the visual quality between the matting results are compared by using the test video frames from [88]. The video frames in Fig. 5.21 - Fig. 5.25 are taken in real-life studio rooms. In this case we cannot calculate and compare the numeric image quality (i.e., PSNR and *etc.*) without the ground truth alpha maps as reference. However, the visual difference is obvious enough among the alpha maps from different matting algorithms. This is because the test frames used here contain many difficult problems that conventional matting methods cannot give reliable solutions. These problems will be shown and explained by comparing the matting results in the following figures.

In Fig. 5.21, the original video frame contains transparent region on black glasses, and reflective region on the man's hand. For most matting methods, the chrominance of the pixel is essential information for foreground color and alpha value estimation. And the chrominance appearance of monochromatic object can be easily affected by lighting and noise. In this case, it is often unreliable to predict the alpha values for monochromatic and transparent object (e.g., glasses in this example). In Fig. 5.21 (d), the glasses are erroneously estimated to be nearly opaque while it is highly transparent. The other problem is that the man's hand is reflecting green light from the background. This is a common and difficult problem in practice. If this problem can not be solved, the reflective opaque foreground object would be identified as

transparent object because the color appearance of reflective region is very similar to the appearance of transparent region. In Fig. 5.21 (c) and (f), part of the hand is erroneously estimated to be transparent because of the reflecting effect. It is also noticeable that there is significant artifact in Fig. 5.21 (f), this is because the WCT matting algorithm utilizes local sampling, which may not be able to collect right colors if the unknown region is large. In this example, our proposed method (Fig. 5.21 (b)) and the closed form matting (Fig. 5.21 (e)) can both provide reliable alpha estimation, which correctly represents transparency and reflection.

In Fig. 5.22, the original video frame contains very large transparent region, and little reflective region on the boundary of the actress's body. Firstly, it is easy to notice that the background region is not completely removed in Fig. 5.22 (c) and (d). This problem is a conventional challenge in chroma keying systems. In real life studio rooms, the background color varies along with lighting conditions. In this case, a good chroma keying system needs to be robust to the change of background lighting, and be able to remove the background regions even if their luminance is different. Besides, a small part of the background in Fig. 5.22 (e) is erroneously estimated to be semi-transparent. This is because the closed form matting is a propagation based method which cannot generate reliable result in "holes" of the trimap. The "hole" in this example is the small background region that is completed surrounded by unknown regions. In Fig. 5.22 (c) (e) and (f), the reflective region at the body boundary is again erroneously identified as transparent. In this example, our proposed method (Fig. 5.22 (b)) generates large smooth transparent region, completely removes the background region, and avoids the ambiguous between reflection and transparency.

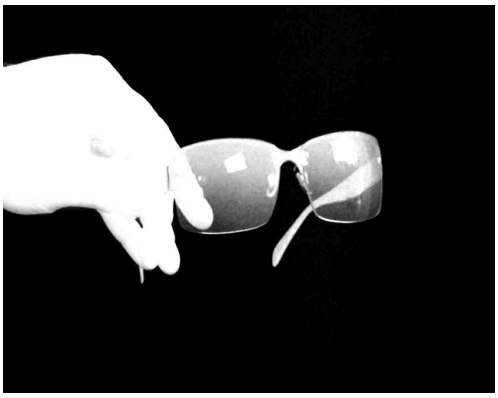
In Fig. 5.23, it is a similar example as Fig. 5.22. One major difference is that the background lighting is more uniform in Fig. 5.23 compared with Fig. 5.22. In this case, Keylight can effectively remove the background in Fig. 5.23 (c). Similar with Fig. 5.22, Primatte and closed form matting still do not perform well on large transparent regions. They estimate much higher alpha values to human perception.

The result from Keylight in Fig. 5.23 (c) still has problems with distinguishing reflection and transparency. In this example, our propose method (Fig. 5.22 (b)) and WCT matting (Fig. 5.22 (f)) both generate visually pleasing matting results.

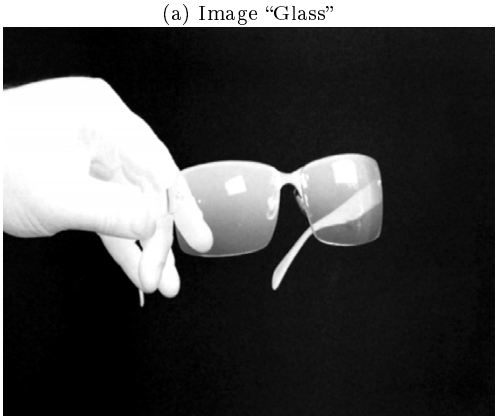
In Fig. 5.24, the original video frame contains background with noticeable lighting variation, and dark foreground object reflecting the environment light. In this example, propagation based method (Fig. 5.24 (e)) fails to estimate the correct alpha matte because the property of local color affinity is impaired by the dark environment in the car. Meanwhile, the matting results in Fig. 5.24 (c) (d) and (f) all erroneously identify the reflecting chair to be transparent. As for the result from our proposed method (Fig. 5.24 (b)), the reflection is not completely detected in this time. Although part of the reflecting chair is correctly estimated to be opaque, the remained part is erroneously identified to be transparent. This is because the color of the reflecting chair is almost identical to the color of the transparent back window. In this case, it is very difficult to make the window transparent while keeping the chair opaque.

In Fig. 5.25, the original video frame contains foreground objects that are severely reflecting environment light. The matting results in Fig. 5.25 (c) and (d) do not completely remove the background region. The results in Fig. 5.25 (c) and (e) erroneously estimate the foreground objects to be transparent. In Fig. 5.25 (f), the artifact arises again because large unknown region makes it difficult to find reliable foreground color candidates. In this example, our proposed method can completely remove the background region and correctly avoid the ambiguous between transparent and reflection.

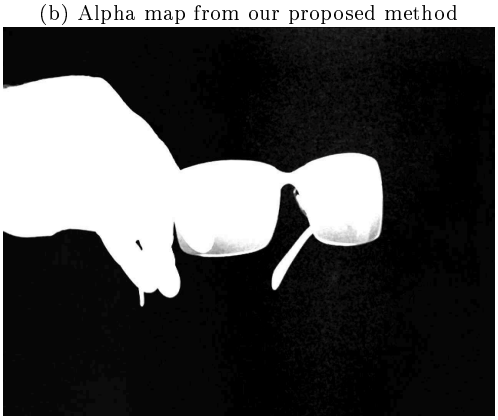
(a) Image “Glass”



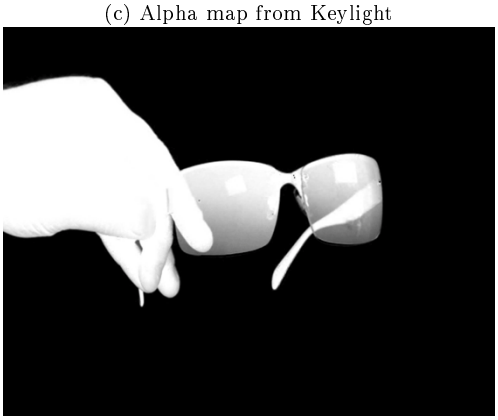
(b) Alpha map from our proposed method



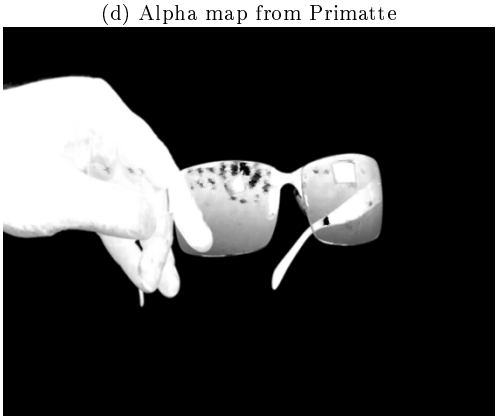
(c) Alpha map from Keylight



(d) Alpha map from Primatte

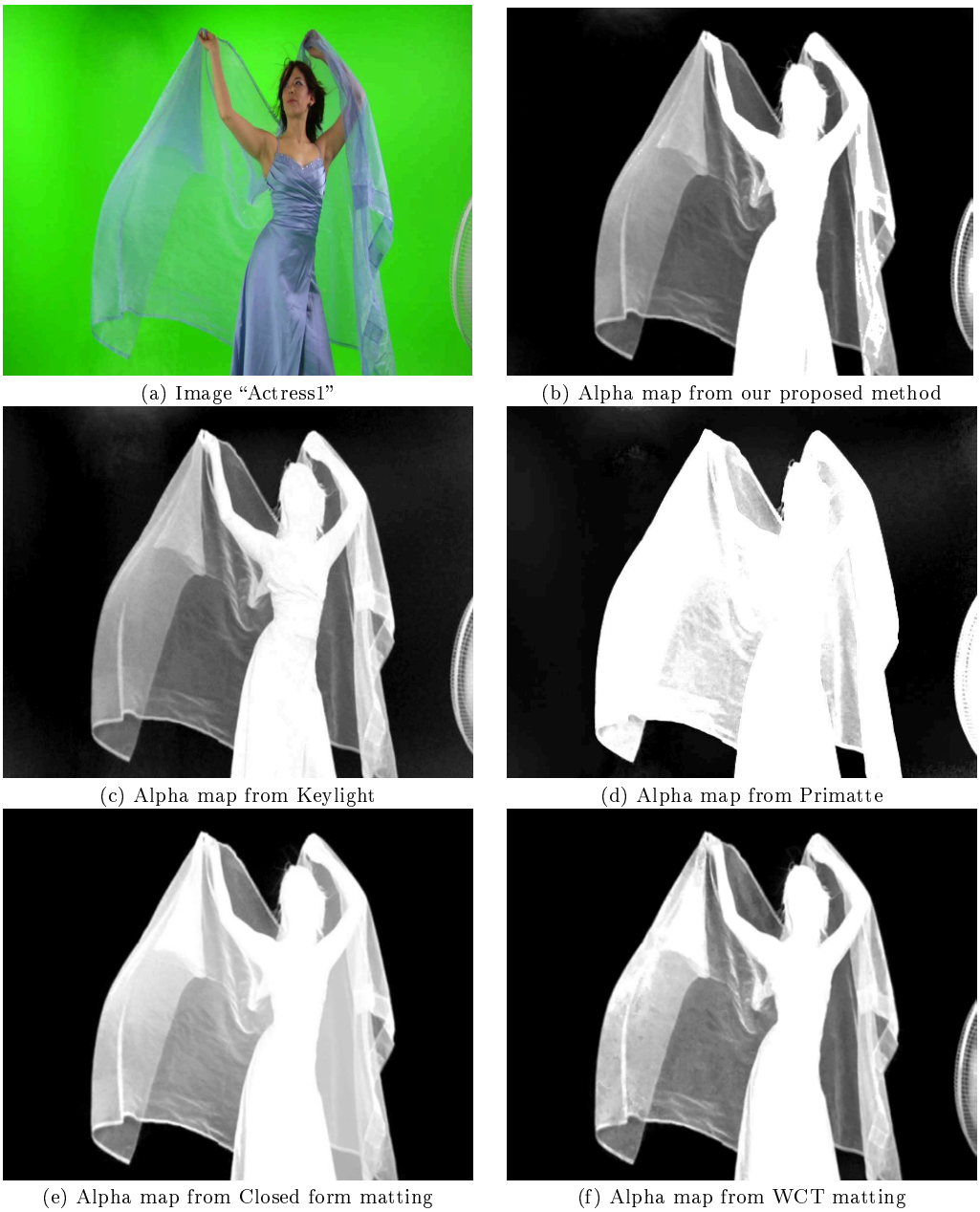


(e) Alpha map from Closed form matting



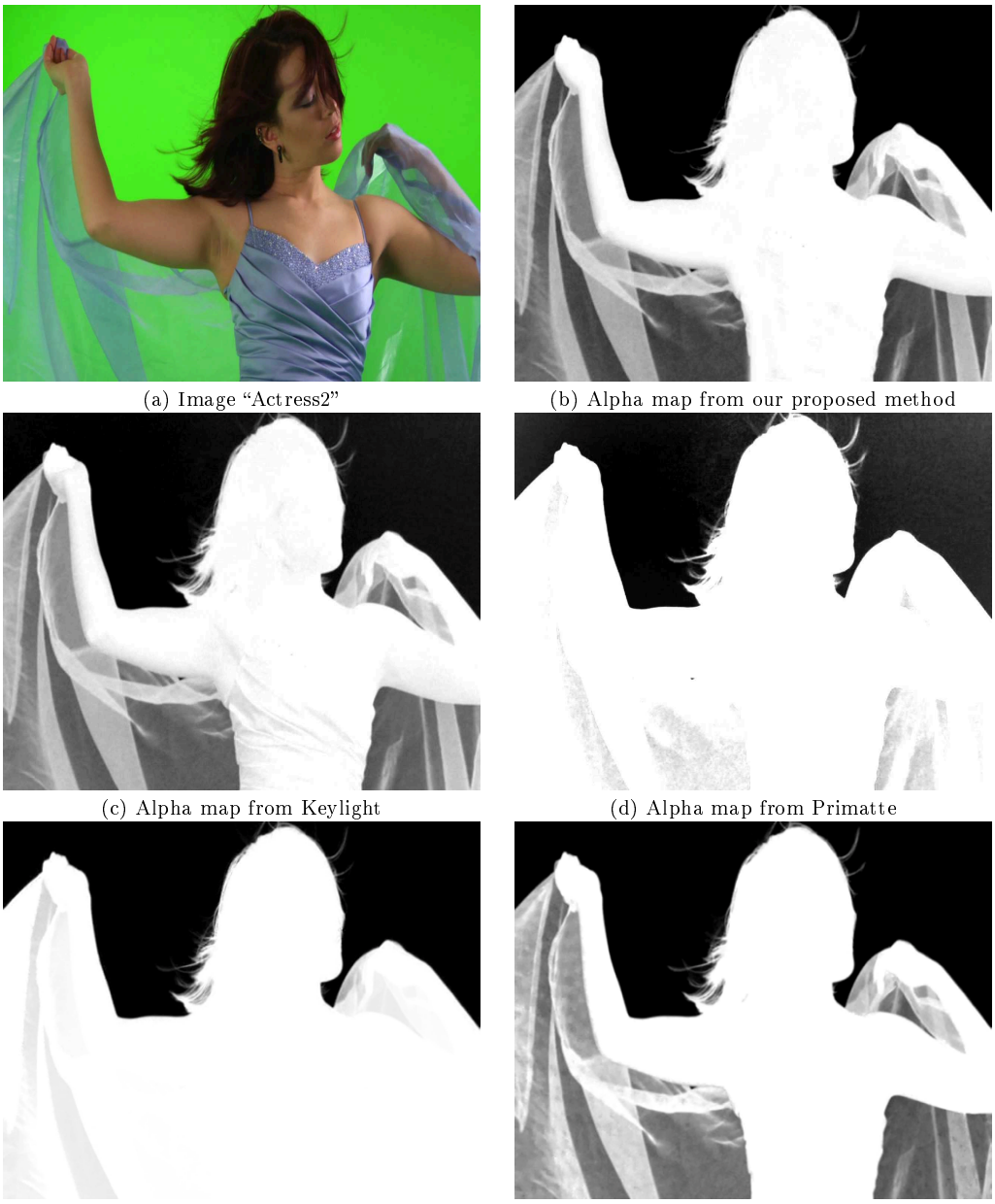
(f) Alpha map from WCT matting

Figure 5.21: The estimated α maps of image “Glass” by using different matting methods.



(a) Image "Actress1"
(b) Alpha map from our proposed method
(c) Alpha map from Keylight
(d) Alpha map from Primatte
(e) Alpha map from Closed form matting
(f) Alpha map from WCT matting

Figure 5.22: The estimated α maps of image "Actress1" by using different matting methods.



(a) Image "Actress2"

(b) Alpha map from our proposed method

(c) Alpha map from Keylight

(d) Alpha map from Primatte

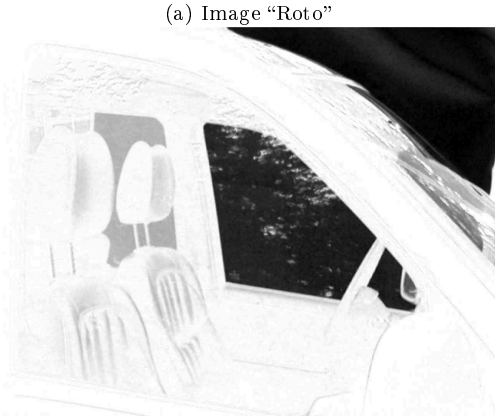
(e) Alpha map from Closed form matting

(f) Alpha map from WCT matting

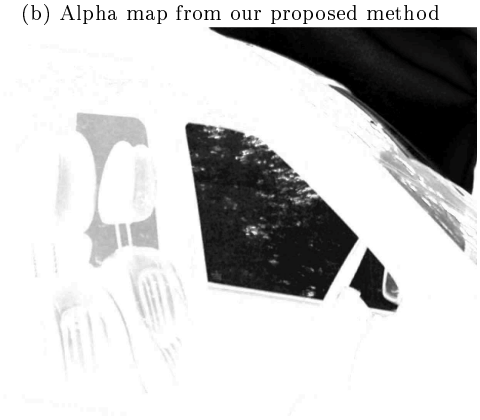
Figure 5.23: The estimated α maps of image "Actress2" by using different matting methods.

(a) Image "Roto"

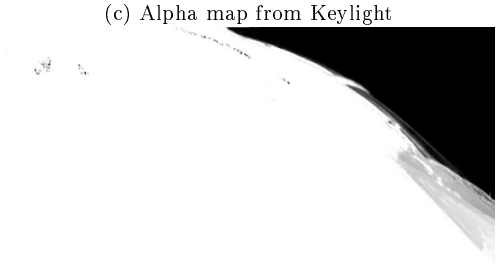
(b) Alpha map from our proposed method



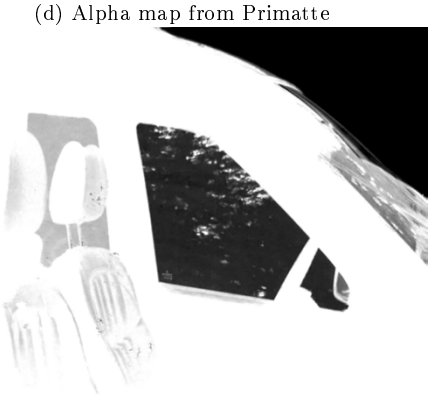
(c) Alpha map from Keylight



(d) Alpha map from Primatte

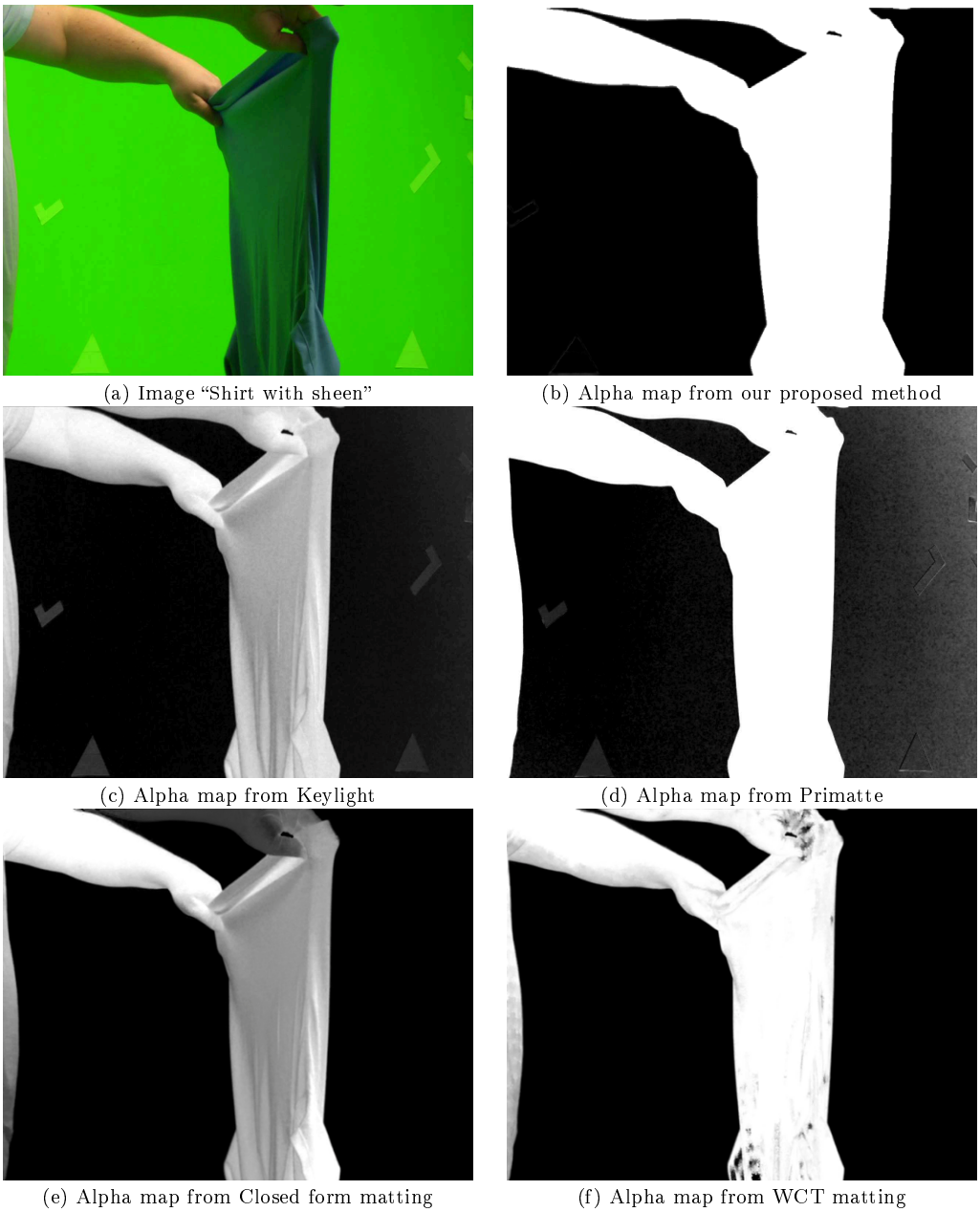


(e) Alpha map from Closed form matting



(f) Alpha map from WCT matting

Figure 5.24: The estimated α maps of image "Roto" by using different matting methods.



(a) Image "Shirt with sheen"

(b) Alpha map from our proposed method

(c) Alpha map from Keylight

(d) Alpha map from Primatte

(e) Alpha map from Closed form matting

(f) Alpha map from WCT matting

Figure 5.25: The estimated α maps of image "Shirt with sheen" by using different matting methods.

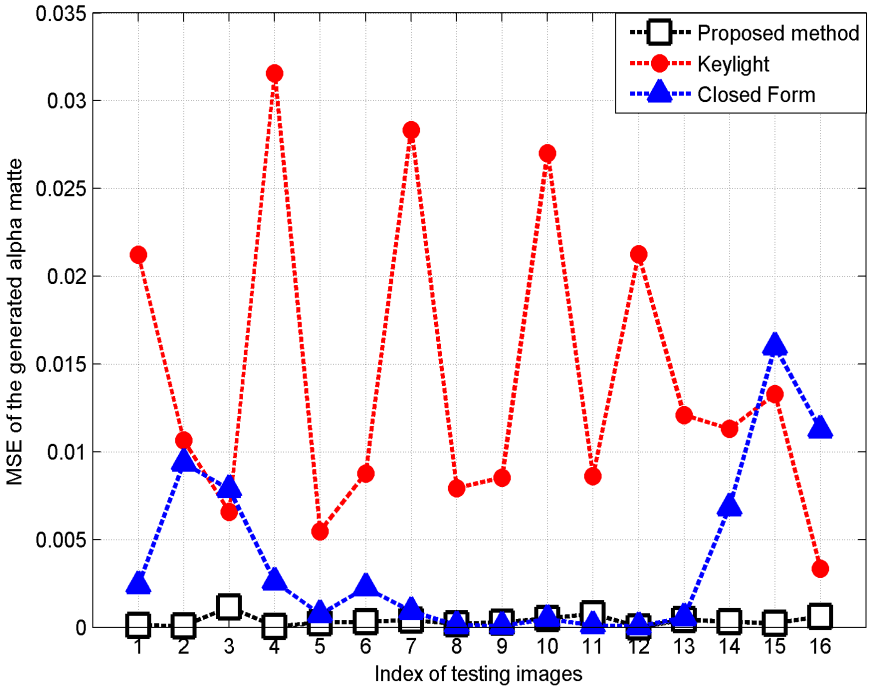
5.5.2 Objective quality comparison

In this part, our proposed chroma-keying method is applied to the test images from online benchmark [19]. Given the groundtruth foreground object in linear RGB color space and groundtruth alpha map from [19], we generate 16 test images by composing foreground objects onto green background in real scene as shown in Fig. 5.26 (a1) - (a16). Our proposed chroma-keying method estimates the alpha maps for the test images as shown in Fig. 5.26 (b1) - (b16). The foreground objects are extracted and composed on a checkerboard background in order to better present the matting results, which are as shown in Fig. 5.26 (c1) - (c16).

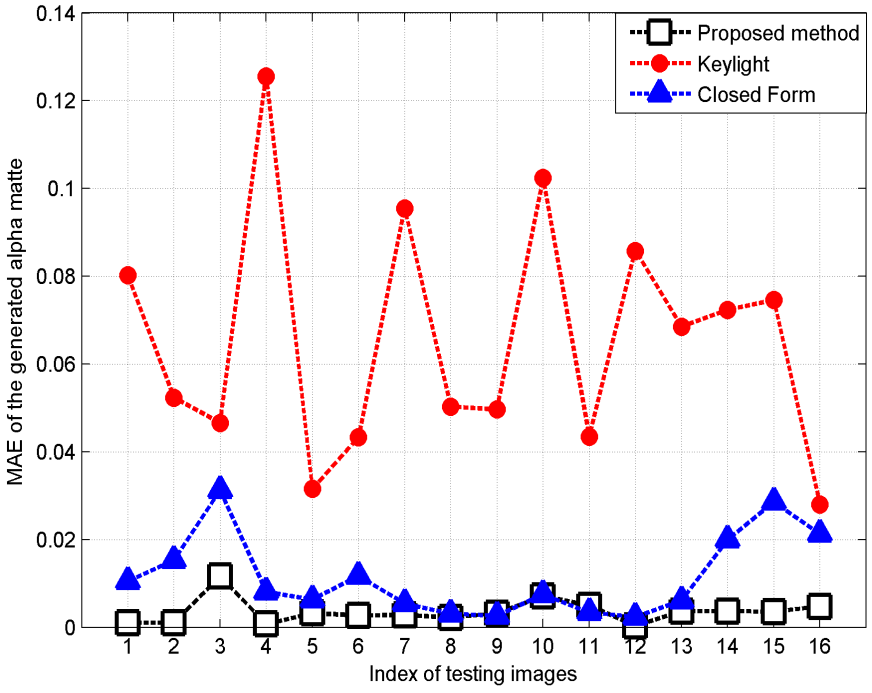
Figure 5.26: Chroma-keying for test images in database [19].

Given the groundtruth alpha maps from [19], MSE (mean squared error) and MAE (mean absolute error) of the generated alpha maps are calculated. The results are shown in Fig. 5.28 (a) and (b). It can be observed that our proposed method performs well for all the test images.

Meanwhile, it is known that the quality of alpha maps can not be always reliably estimated by metrics like MSE or MAE. In this case, we use positive detection ratio as supplementary metrics as shown in Fig. 13 (a) and (b). Here the positive detection means that an estimated foreground/background pixel is also a foreground/background pixel in the groundtruth alpha map. It can be observed that the positive detection ratio for our proposed keying method is very high for both foreground and background (above 90 percentage) regions. For foreground region detection, the matting results from "Keylight" are as good as ours. For background region detection, our method outperforms the other two. This means that our method can reliably extract foreground object while it can also keep the background clean.

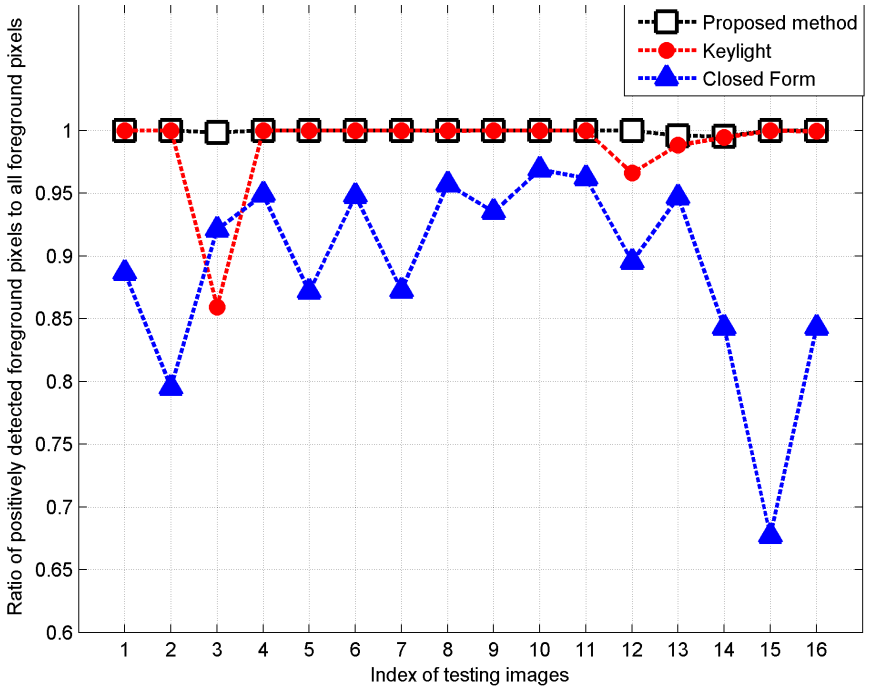


(a) Mean squared error

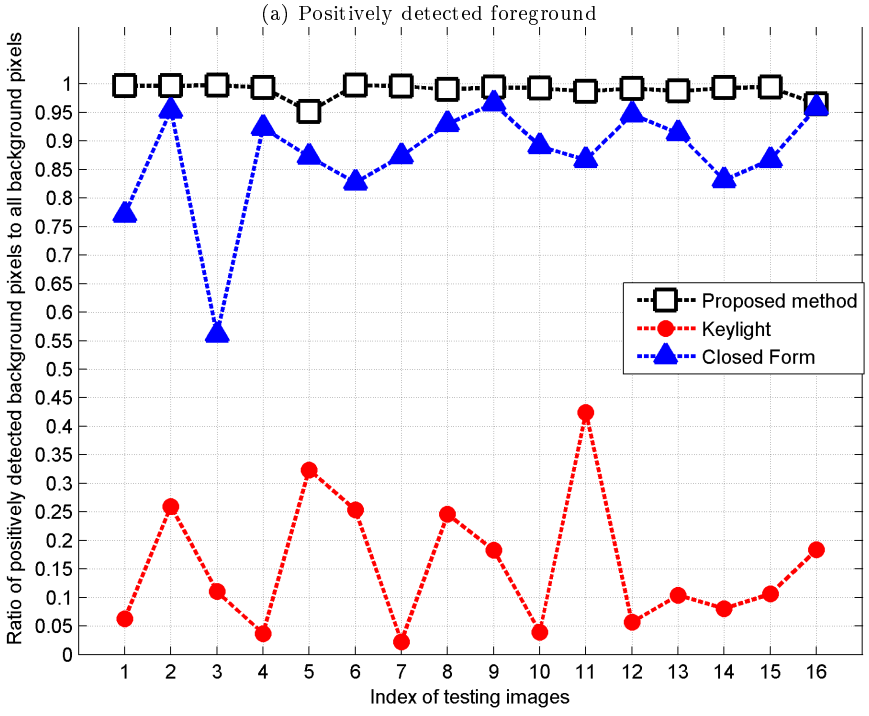


(b) Mean absolute error

Figure 5.27: Matting quality comparison



(a) Positively detected foreground



(b) Positively detected background

Figure 5.28: Positive detection ratio comparison

6

Proposed alpha covert transmission by using reversible watermarking

6.1 Reversible watermarking and covert transmission

In the previous chapter, we proposed the robust chroma keying which can not only be used by professional engineers but also be used by normal users. The wide range of applications can result in widespread multimedia with α information. Meanwhile IP based broadcasting switcher has been compelling since SONY proposed their new IP based switcher technology NXL-IP55. Although the current technology is mainly designed for video sources synchronization and cost saving, this could be a promising start that the mixed video sources can be transmitted and shared over a wider range of the Internet. Since the color video with the alpha channel plays an essential role

for video mixture, it could be helpful in the future that such multimedia content is digital encrypted for copyright protection and access control. Besides, the α information of color image is normally saved and transmitted in a separate channel, which requires extra transmission bandwidth. In order to settle these problems, a covert transmission method is proposed in this chapter to embed the generated α channel into its associated color image/video. By doing this, the copyright can be verified and protected by examining and decoding the hidden α information. At the same time, the transmission bandwidth can also be saved if the α information does not need to be saved and transmitted separately.

Generally, the copyright protection can be made by inserting a digital watermark [89] into the original multimedia content. The concept of digital watermarking is derived from steganography, which is used to convey secure information by embedding it into the cover data. The major difference between steganography and digital watermarking is the requirement of robustness. Normally, the cover information used in steganography is not supposed to be modified. In this case, there is little requirement for the robustness of the embedded information under different distortions. On the other hand, during the storage, transmission, and redistribution, digital multimedia suffers from varies distortions such as scaling, compression, and noise. In this case, the digital watermarking needs to be robust to such distortions to guarantee that the embedded information can be always detected and extracted.

In Fig. 6.1, a typical watermarking system [89] is presented as an example. In this system, watermark W is embedded into the cover information C by using watermarking algorithm with security key K. The watermarked cover information C_w might be distorted into C'_w during the transmission. At the receiver side, the watermark detector should be able to extract or detect the embedded watermark W with the security key K even if the received data C'_w is not exactly the same as the transmitted data C_w.

Besides the robustness, there are two other important properties related to digital

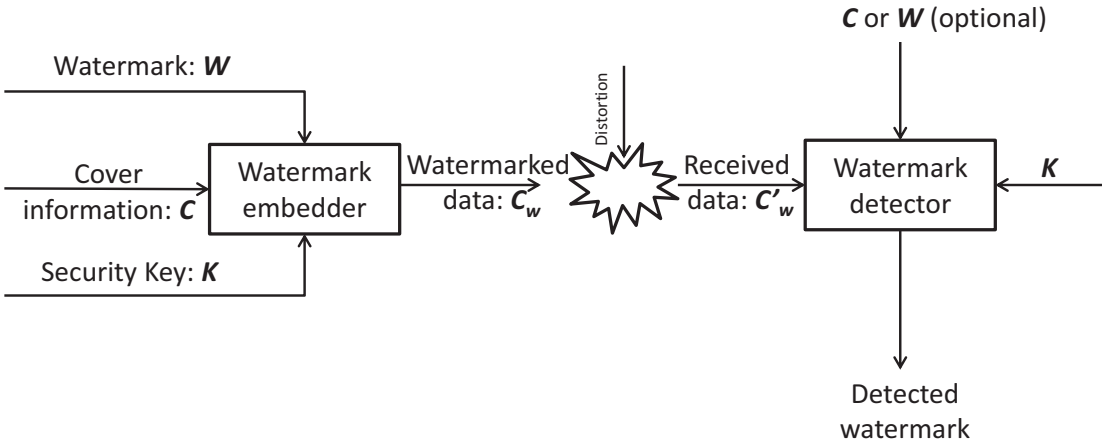


Figure 6.1: A typical digital watermark embedding/extraction scheme.

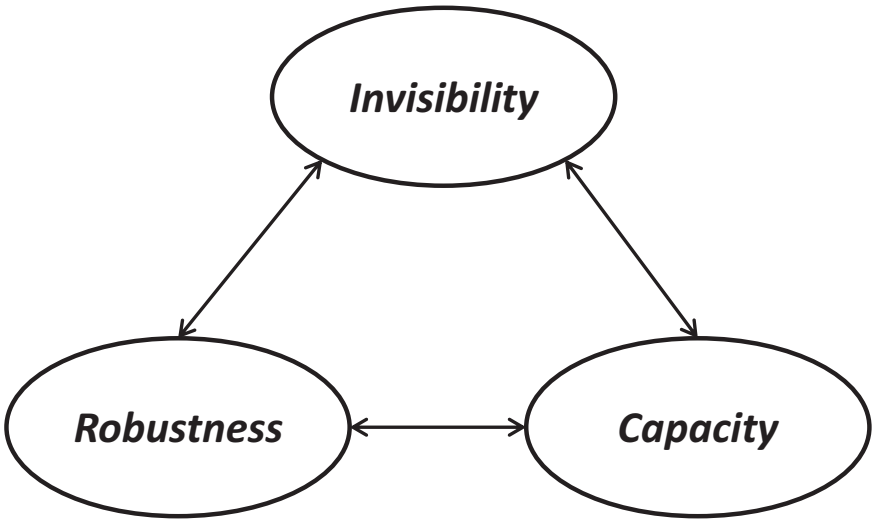


Figure 6.2: The mutual restraints for a robust digital watermarking scheme.

watermarking: invisibility and capacity. These three properties are often used to judge the performance of a digital watermarking algorithm.

- Invisibility: it means that the watermarked image should not be visually different from the original image.
- Robustness: it means that the embedded watermark can still be extracted or detected even if the watermarked image is distorted by filtering, noising, geometric transformation, compression, *etc.*
- Capacity: it represents the maximum amount of retrieval information the embedded watermark can carry.

As shown in Fig. 6.2, these three properties mutually restrain each other. In other word, the other properties often need to be sacrificed if the performance of one property is emphasized. In order to embed the α channel into the original color image/video, the whole α channel is regarded as the watermark. In this case, we need to hide a large amount of information into the cover signal, which means that the requirement for watermarking capacity is very high. We also know that high capacity often introduces the poor performance on robustness and invisibility. On the other hand, the cover information used here is the original color image of the α channel, the quality of which is very important for post-processing such as image compositing. Besides, the image/video are often transmitted and stored in lossy compression format, such as JPEG for image and H.264 for video. In this case, it is very challenging to design such a watermarking system that has good performance on three mutual restraints at the same time.

In the thesis, a reversible watermarking system on quantized DCT domain is proposed to embed the α channel into the color image/video in JPEG/H.264 format. In a reversible watermarking system, the original image/video before watermarking can be completely restored at the receiver side after the watermark is detected and

extracted. In this case, the receiver with authorized decoder cannot only extract the hidden α channel but also remove the distortion in the color image/video introduced by watermarking. The detailed descriptions of our proposed reversible watermarking will be introduced in the following sections.

6.2 Proposed reversible watermarking scheme in quantized DCT domain

6.2.1 Watermarking scheme

For α channel embedding, it is required to hide heavy payload into the host image/video. Considering the requirement for large payload, fragile watermarking algorithms are suitable choice because of their high capacity [90] [91] [92] [93]. However, the requirement for compression robustness makes it not suitable to apply fragile watermarking algorithms in spatial domain. This is because the α channel, which is hidden by fragile watermarking in spatial domain, will be hardly detected and extracted after image/video compression. In order to design a high capacity watermarking method which is also robust to image/video compression, we propose a QDCTE watermarking algorithm that hides the α channel into the quantized DCT (QDCT) coefficients by expansion operation:

$$|Q_{iw}| = 2^L \times |Q_i| + \sum_{k=0}^{L-1} \left(2^k \times b_k\right), \tag{6.1}$$

where $|Q_i|$ is the absolute value of non-zero quantized DCT coefficient, $|Q_{iw}|$ is the absolute value of watermarked coefficient, L is the embedding level, $b_0, b_1, \cdots, b_{L-1}$ is a binary sequence of watermark, which is the compressed α information in this chapter.

The non-zero quantized DCT coefficients (AC component) of JPEG image or H.264 video are expanded and the α channel is bit-wise hidden into the least signifi-

cant bit (LSB) planes of the quantized DCT coefficients. According to basic watermarking principles [89], watermark capacity, robustness and visual quality are three conflictive characteristics. In this case, the visual quality of watermarked image/video can not be guaranteed because high capacity and compression robustness are both required in the application of α channel hiding. In order to ensure high quality image/video at the receiver side, the watermarking method proposed in this chapter is reversible. The reversibility means that any distortion introduced by watermark embedding can be removed and the original cover image/video can be completely restored. This unique property can make up the image/video distortion caused by heavy load watermarking embedding.

6.2.2 Performance on probability distribution preservation

Since α channel is embedded into QDCT coefficients, which are further losslessly entropy coded, the entropy change introduced by watermarking affects compression efficiency and thereby causes data size overhead. With this concern, the occurrence probability distributions of the original and the watermarked QDCT coefficients are derived and analyzed from the perspective of entropy information.

It has been observed that the probability density function (PDF) of DCT coefficients of natural image obeys Laplacian distribution centered at origin [94]:

$$p(x) = \frac{1}{2\sigma} exp\left(-\frac{|x|}{\sigma}\right), \tag{6.2}$$

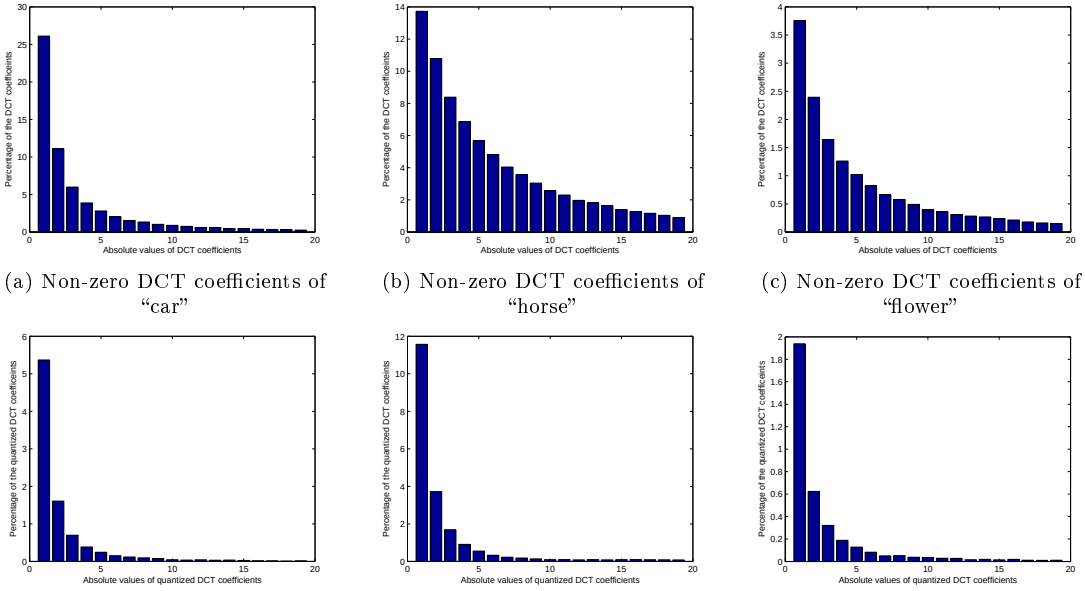
where x is the value of random variable, and σ^2 is the variance of x.

In Fig. 6.4, it gives the occurrence probability distributions of the non-zero DCT coefficients of the test images shown in Fig. 6.3. It can be seen that most of the energy distributes at low values and this phenomenon is well exploited and is regarded as presumption for entropy encoding in image and video compression to enhance the compression ratio [95]. Note that the percentage of zero values are not shown in

Fig. 6.4 because zero values, which are coded by run-length coding, are not used for watermarking in this chapter.

(a) Image "car" (b) Image "horse" (c) Image "flower"

Figure 6.3: The test images for the illustration of DCT coefficients distribution.



(a) Non-zero DCT coefficients of "car" (b) Non-zero DCT coefficients of "horse" (c) Non-zero DCT coefficients of "flower"

(d) Non-zero quantized DCT coefficients of "car" (e) Non-zero quantized DCT coefficients of "horse" (f) Non-zero quantized DCT coefficients of "flower"

Figure 6.4: The Laplacian distribution of image DCT coefficients.

Intuitively, the proposed expansion based watermarking method only shifts and stretches the original PDF other than completely changes its shape. In other word, the Laplacian-shape-like PDF, which benefits the entropy coding ratio, is still preserved after watermarking. Given the continuous Laplacian PDF as shown in Equ.

(6.2), the discrete Laplacian PDF can be derived as follows:

$$\begin{aligned} P(Y=k) &= \frac{p(k)}{\sum_{j=-\infty}^{+\infty} p(j)} = \frac{exp\left(-|k|/\sigma\right)}{\sum_{j=-\infty}^{+\infty} exp\left(-|j|/\sigma\right)} \\ &= \frac{m^{|k|}}{\sum_{j=-\infty}^{+\infty} m^{|j|}} = \frac{m^{|k|}}{1+2\sum_{j=1}^{+\infty} m^{j}} \\ &= \frac{m^{|k|}}{1+2\frac{m}{1-m}} = \frac{1-m}{1+m} m^{|k|}, \quad k,j \in Z\,, \end{aligned} \tag{6.3}$$

where k is the discrete coefficient value, p is the Laplacian PDF in continuous domain, and $m = exp(-1/\sigma)$.

Given the discrete Laplacian PDF, the entropy $H(Y_o)$ can be derived as follows:

$$\begin{aligned} H(Y_o) &= E\left(-log_2 P(Y=k)\right) \\ &= \sum_{k=-\infty}^{+\infty} -\left(\frac{1-m}{1+m} m^{|k|}\right)\left(log_2 \frac{1-m}{1+m} + |k| log_2 m\right) \\ &= \left(log_2 \frac{1-m}{1+m}\right)\left(-\frac{1-m}{1+m}\right)\sum_{k=-\infty}^{+\infty} m^{|k|} + \\ &\quad (2log_2 m)\left(-\frac{1-m}{1+m}\right)\sum_{k=1}^{+\infty}\left(km^{k}\right) \\ &= -log_2 \frac{1-m}{1+m} - 2log_2 m\left(\frac{1-m}{1+m}\right)\frac{m}{(1-m)^2} \\ &= -log_2 \frac{1-m}{1+m} - \frac{2m \cdot log_2 m}{1-m^2}\,, \end{aligned} \tag{6.4}$$

where k is the discrete coefficient value before watermarking, and $m = exp(-1/\sigma)$.

According to Shannon's communication theory [96], the average word length R of any decodable variable-length coding is lower bounded by $H(x)$, which is the entropy of independent and identical distribution (IID) random process. In this case, we approximate the binary distribution of compressed α channel to be independent and identical. Suppose that the proposed watermarking algorithm embeds the IID binary

stream (i.e., compressed α) into the original QDCT coefficients by 1 bit expansion (the derivation for multiple bit expansion can be the same). the original PDF are stretched. The even values and odd values still obey discrete Laplacian distribution respectively. If we denote

$$P^1 = P(Y = 2k + sgn(k)) = \frac{1}{2}\frac{1-m}{1+m}m^{|k|},\ k \in Z\,, \tag{6.5}$$

$$P^2 = P(Y = 2k) = \frac{1}{2}\frac{1-m}{1+m}m^{|k|},\ k \in Z\,, \tag{6.6}$$

where k is the discrete coefficient value before watermarking, $sgn(x) = x/|x|$ is the sign function, $m = exp(-1/\sigma)$.

The entropy $H(Y_w)$ of the watermarked QDCT coefficients then can be derived as follows:

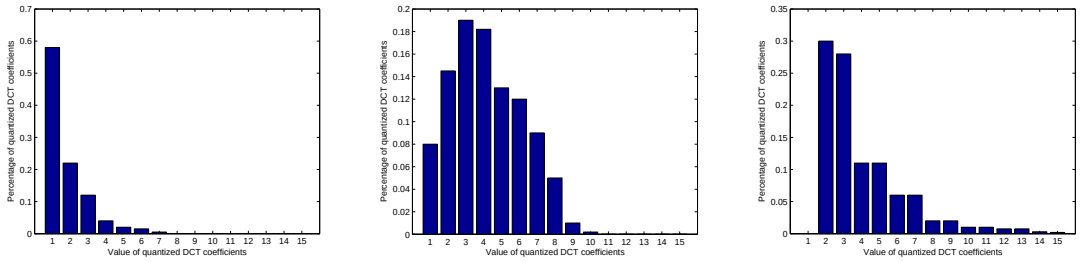
$$\begin{aligned} H(Y_w) &= E(-log_2 P^1) + E(-log_2 P^2) \\ &= 2\sum_{k=-\infty}^{+\infty} -\frac{1}{2}\left(\frac{1-m}{1+m}m^{|k|}\right)\left(-1 + log_2\frac{1-m}{1+m} + |k|log_2 m\right) \\ &= \sum_{k=-\infty}^{+\infty}\frac{1-m}{1+m}m^{|k|} + \\ &\quad \sum_{k=-\infty}^{+\infty} -\left(\frac{1-m}{1+m}m^{|k|}\right)\left(log_2\frac{1-m}{1+m} + |k|log_2 m\right) \\ &= 1 + H(Y_o)\,. \end{aligned} \tag{6.7}$$

According to Equ. (6.7), the entropy increases by 1 bit/symbol after watermarking. On the other hand, we hide one bit α information into each QDCT coefficient (i.e., symbol) in our proposed watermarking algorithm. From the perspective of controlling entropy increase, our proposed watermarking algorithm is the optimal one because it introduces no extra entropy increase.

Besides the mathematical derivation, we also provide the experimental comparison with respect to the preservation of DCT probability distribution. In order to compare

with our proposed QDCTE watermarking, we implemented another reversible expansion based watermarking method, which is called difference expansion (DE) [97], in the quantized DCT domain. The reason that we use DE watermarking here is that this algorithm is also capable of hiding large amount data into the cover signal and it is also reversible.

When the DE watermarking and our QDCTE watermarking are respectively used in the quantized DCT domain of image "Lena", it can be seen from Fig. 6.5 that DE watermarking destroys the Laplacian-shape-like distribution of the quantized DCT coefficients while QDCTE watermarking only shifts and stretches the original distribution.



(a) Original image (b) DE watermarked (c) QDCTE watermarked

Figure 6.5: Distribution of quantized DCT coefficients. The statistics of the quantized DCT coefficients are counted for (a) original image "Lena", (b) watermarked "Lena" by using DE, (c) watermarked "Lena" by using QDCTE.

More specifically, by using our proposed QDCTE watermarking, the probability of watermarked coefficients with four different values (2, 3, 4, 5) is over 80% out of the total number of coefficients while the probability of watermarked coefficients with six different values (2, 3, 4, 5, 6, 7) is about 84%. In this case, our proposed scheme generates watermarked coefficients with better property of energy concentration than DE does. Therefore, the entropy coding performs more efficiently in our proposed watermarking scheme, which leads to a smaller compressed size compared to DE.

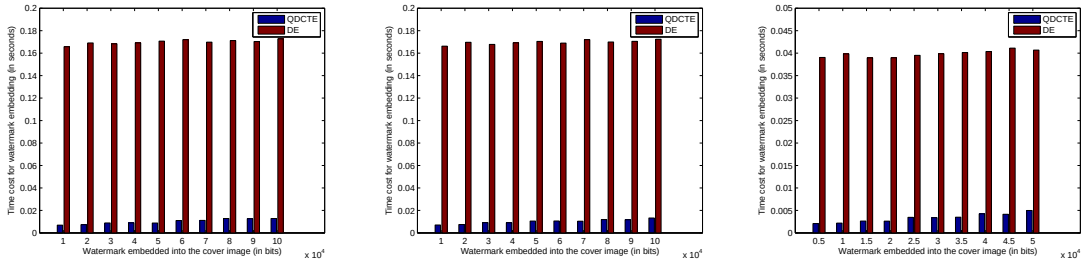
6.2.3 Performance on computing efficiency

As aforementioned, the watermark embedding is performed by bit-wise shift, which results in efficient calculation. In this section, we illustrate the computation efficiency by comparing our proposed watermarking method with DE watermarking method with respect to calculation efficiency. Test images are "Lena", "Plane" and "Baboon", as shown in Fig. 6.6. The test platform is MATLAB running on a ThinkPad laptop with Intel(R) Core(TM) i7-2760QM 2.4GHz, 16 GB RAM, and 64-bits Windows operating system.

(a) "Lena" (b) "Plane" (c) "Baboon"

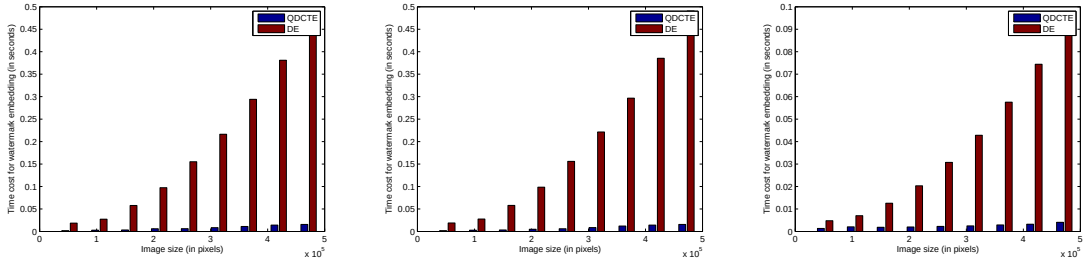
Figure 6.6: Test images

First, each test image is embedded with various watermark payload by using our proposed method (QDCTE) and different expansion (DE). In Fig. 6.7, it can be observed that the calculation time of our proposed method is significantly shorter than DE. Second, each test image is scaled to different size and a watermark with 5000 bits is embedded into each of the scaled images. By doing this, the calculation time for different cover image size is tested and compared. In Fig. 6.8, it is still obvious that our proposed method costs less time to embed watermark into images with different size.



(a) "Lena" (b) "Plane" (c) "Baboon"

Figure 6.7: Calculation time comparison with different watermark payload. The time cost of watermark embedding is compared between our proposed method (QDCTE) and DE method. The watermarking payload varies from 10000 bits to 100000 bits.



(a) "Lena" (b) "Plane" (c) "Baboon"

Figure 6.8: Calculation time comparison with different cover image size. The time cost of watermark embedding is compared between our proposed method and DE method. The horizontal axis presents the image size in pixels after scaling.

6.3 Entropy coding customization

For JPEG image or H.264 video, the quantized DCT coefficients are further encoded by entropy encoding. The probability distribution of these coefficients for natural images are well statistically analyzed and regarded as prior knowledge for entropy encoding in compression standards such as JPEG or H.264. In these compression standards, symbols with small values are coded with shorter codes and large values are coded with longer codes. This assumption performs well for natural images since most of the AC coefficients are small after quantization. Unfortunately, the watermarked DCT coefficients no longer obey this prior assumption of occurrence probability. In our QDCTE watermarking algorithm, the distribution of the watermarked coefficients

is shifted along with the embedding level. In this case, it is necessary that the entropy encoding is adjusted to meet the new probability distribution of the watermarked coefficients. Since the entropy encoding for JPEG image and H.264 video are different, the customization methods will be introduced separately for image and video in the following two sections.

6.3.1 Huffman encoding customization for JPEG images

In JPEG image compression standard, there are two optional entropy encoding methods used for quantized DCT coefficients compression: Huffman encoding and arithmetic encoding. In this section, we utilize and customize Huffman encoding as shown in Fig. 6.9.

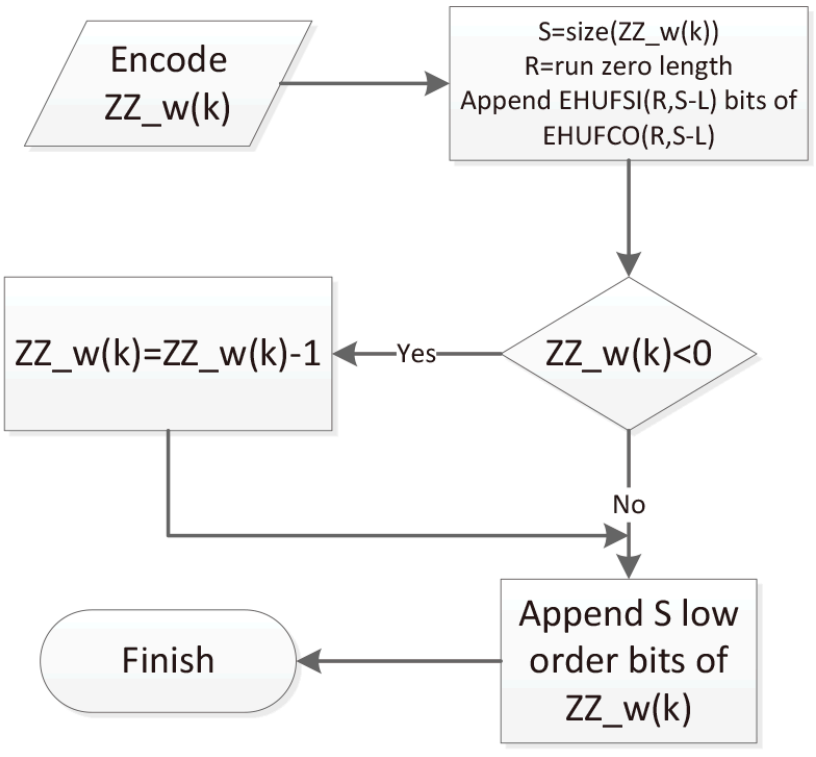


Figure 6.9: Customized Huffman encoding for AC coefficients.

The JPEG Huffman coding for quantized AC coefficients is processed in two steps:

first, an 8×8 quantized DCT block is scanned into a zigzag order sequence ZZ and each nonzero AC coefficient $ZZ(k)$ is paired with the number of successive zeros following the previous nonzero AC coefficient. The number of zeros is called zero-run-length (R). Then the quantized AC coefficient $ZZ(k)$ is assigned a category index S with respect to the following condition:

$$ZZ(k) \in [2^{S-1}, 2^S - 1] \cup [-2^S + 1, -2^{S-1}], \quad \text{where} \quad 1 \leq S \leq 10\,. \tag{6.8}$$

In this case, the quantized AC coefficient is represented by pair (R, S). In the second step, $ZZ(k)$ is coded into a binary sequence by generating a sequence B_1 stored in the entry (R, S) of the Huffman table and an S bits sequence B_2 representing the signed amplitude of AC coefficient $ZZ(k)$. Note that, the code length of B_2 is fixed to S and only the code length of B_1 can be adjusted.

Huffman coding for pair (R, S) generates the binary sequence B_2 from two Huffman tables: EHUFCO (containing the code word for each possible pair) and EHUFSI (containing the length of each code word in bits). These two Huffman tables are pre-defined according to the probability distribution of DCT coefficients for natural images. After watermarking, the values of quantized DCT coefficients are changed and therefore the category index S_w is changed too. In this case, a new mapping between the pair (R_w, S_w) and the probability distribution is necessary. Assume that the probability of occurrence of pair (R, S) is $P(R, S)$, and the value of AC coefficients $ZZ(k)$ corresponding to this pair should satisfy Equ. (6.8). After L-level watermarking as in Equ. (6.1), the value of watermarked AC coefficients $ZZ_w(k)$ will fall into the range as follows:

$$\begin{aligned} ZZ_w(k) &\in [2^{S+L-1}, 2^{S+L} - 1] \cup [-2^{S+L} + 1, -2^{S+L-1}]\,, \\ \Rightarrow ZZ_w(k) &\in [2^{S_w-1}, 2^{S_w} - 1] \cup [-2^{S_w} + 1, -2^{S_w-1}]\,, \end{aligned} \tag{6.9}$$

where

$$S_w = S + L\,. \tag{6.10}$$

Note that the category index S_w of the watermarked AC coefficients $ZZ_w(k)$ is changed as shown in Equ. (6.10). In this case, given a nonzero AC coefficient $ZZ(k)$ and its corresponding index pair (R, S), the index for the watermarked coefficient $ZZ_w(k)$ will be $(R, S+L)$. Since all of the non-zero AC coefficients will be expanded and watermarked, the zero-run-length (R) will not change, and the probability of occurrence of pair (R_w, S_w) is just the same as the probability of occurrence of pair (R, S) before watermarking. This indicates the following probability relation:

$$P_w(R_w, S_w) = P_w(R, S+L) = P(R, S)\,. \tag{6.11}$$

To entropy encode a watermarked AC coefficient, the index pair (R_w, S_w) is first calculated, and Huffman table is searched by index pair $(R_w, S_w - L)$. In this case, the coding length of B_1 can be kept unchanged after watermarking. On the other hand, S_w for the watermarked coefficient is larger than the original S by L, and this is also the code length of B_2. In conclusion, the length of Huffman code word for each nonzero AC coefficient is increased by L bits after L bits watermark information (α channel) is hidden into the JPEG image. By doing this, the α channel can be jointly encoded with the JPEG image without introducing extra size increase.

6.3.2 CAVLC encoding customization for H.264 video

In H.264 video compression standard, there are also two types of entropy encoding methods that can be chosen for quantized DCT coefficients compression: context-based adaptive variable length coding (CAVLC) and context-based adaptive binary arithmetic coding (CABAC). In this section, we utilize and customize CAVLC.

The zig-zag order quantized DCT coefficients from 4×4 (or 2×2) blocks is com-

pressed by entropy coding (CAVLC) which takes advantage of the following characteristics of quantized DCT coefficients:

1. The quantized DCT coefficients within a block are typically sparse. Therefore, CAVLC uses run-level coding as Huffman coding does.

2. The highest non-zero coefficients after zig-zag scan are often ± 1. Therefore, CAVLC encodes the number of high frequency ± 1s in a compact way, which is called "Trailing ones".

3. The number of non-zero coefficients in neighbor blocks is correlated. Therefore, CAVLC encodes the number of non-zero coefficients with different coding table depending on the number of non-zero coefficients in the neighboring blocks.

4. The magnitude of non-zero coefficients tends to be higher at the start of the zig-zag order and lower towards the higher frequency. Therefore, CAVLC chooses from 7 VLC tables (Level-VLC0 to Level-VLC6) to encode the magnitude of non-zero coefficients. Table Level-VLC0 is biased towards low magnitude, table Level-VLC1 is biased towards a little higher magnitude and so on.

After L level QDCTE watermarking, all of the non-zero quantized AC coefficients are increased by 2^L times by using Equ. (6.1). In this case, the value for "Tailings" is no longer ± 1 but $\pm(2^L)$. In the conventional H.264 encoder, the initial VLC table chosen for the last non-zero AC coefficient other than trailings is Table Level-VLC0. In order to adjust the bias of VLC coding table after watermarking, the initial VLC

table is updated according to the embedding level as follows:

$$VLC = \begin{cases} VLC0 & \text{No watermark embedded,} \\ VLC1 & 1 < 2^L \leq 3, \\ VLC2 & 3 < 2^L \leq 6, \\ VLC3 & 6 < 2^L \leq 12, \\ VLC4 & 12 < 2^L \leq 24, \\ VLC5 & 24 < 2^L \leq 48, \\ VLC6 & 2^L > 48, \end{cases} \tag{6.12}$$

where L is the embedding level, VLC0, $\cdots$, VLC6 are 7 possible choices for the VLC encoding table.

6.4 Watermark embedding

In this section, the proposed watermark embedding process along with entropy customization is introduced in details as follows:

1. Partially decode the JPEG image or H.264 video stream and extract the quantized DCT coefficients which are generated from pixel values or prediction errors.

2. Calculate the number of non-zero quantized DCT coefficients N1 in one JPEG image or one frame of the H.264 video and the size N2 for the corresponding α channel. The embedding level L, which decides the number of embedding iterations, is determined by $L = \left\lceil \frac{N_2}{N_1} \right\rceil$.

3. Embed the α channel bit-wise into extracted non-zero quantized DCT coefficients by applying expansion. One watermark bit can be embedded into one quantized DCT coefficient by moving the bit plane of the coefficient to the left

one bit. And the embedding level L determines how many bits should be moved to the left for each DCT coefficient as in Equ. (6.1).

4. Customize the entropy encoding methods as in section 6.3.1 and section 6.3.2.

5. After applying customized entropy encoding to the watermarked DCT coefficients, the encoded coefficients are inserted back into the original JPEG image or H.264 video stream. In this case, the α channel is securely hidden into the compression domain as watermark.

6.5 Watermark extraction and cover signal restoration

On the contrary, the watermark (i.e., the hidden α) is extracted by using the following steps:

1. Extract the bit stream of the quantized DCT values from the watermarked image or video frame.

2. Decode the quantized DCT coefficients using customized entropy decoder with regard to the watermark embedding level L.

3. Extract L least significant bits from each quantized DCT coefficient as the hidden watermark. The original quantized DCT can be recovered by using Equ. (6.13):

$$Q_i = c \left\lfloor Q'_i / 2^L \right\rfloor . \tag{6.13}$$

4. After the restoration of the original DCT coefficients and the extraction of α information, the restored quantized DCT coefficients are encoded by the default entropy encoder and insert back into the JPEG image or H.264 video. In this case, the hidden α information is extracted and the image/frame quality will not suffer from the watermarking process.

6.6 Experimental results

In this section, we will use the proposed watermarking algorithm to hide alpha information in JPEG image and H.264 video. Besides, the watermarked image/video will be compared to the original ones in visual quality and compressed size.

6.6.1 Experimental results for JPEG images

We first hide the alpha map as watermark into the natural image by QDCTE and Huffman coding customization. The cover images are JPEG compressed and the alpha maps are losslessly compressed.

(a) Original "Aloe" (b) Watermarked Aloe

(c) Original "Art" (d) Watermarked "Art"

Figure 6.10: Watermark embedding for testing images "Aloe" and "Art". The watermarked images (b) and (d) are shown before restoration and they can be restored to images (a) and (c) respectively.

Fig. 6.10 shows the experimental results for the test images "Aloe" and "Art". Note that although the quality of the watermarked image is quite low (i.e., 29.7 dB for Aloe, 21.3 dB for Art), the watermarked images can be completely restored to the same as original images. Besides, the hidden alpha maps can also be extracted without any distortion because the embedding happens after DCT quantization.

Table 6.1 shows the size change of the images after watermarking. In this table, "Level" means how many embedding levels have been applied. "Capacity" means the capacity of each embedding level. Note that the product of "Capacity" and "Level", which is the possible maximum load, is almost the same as the size increase ("Add" in Table 6.1) of the JPEG image after watermarking. However, the real load (size of alpha map) can not be always the maximum load for that embedding level and this makes the size of embedded information smaller than the capacity by a little bit. In this case, the size of load is smaller than the size increase a little bit unless the payload is very close to the capacity for that embedding level.

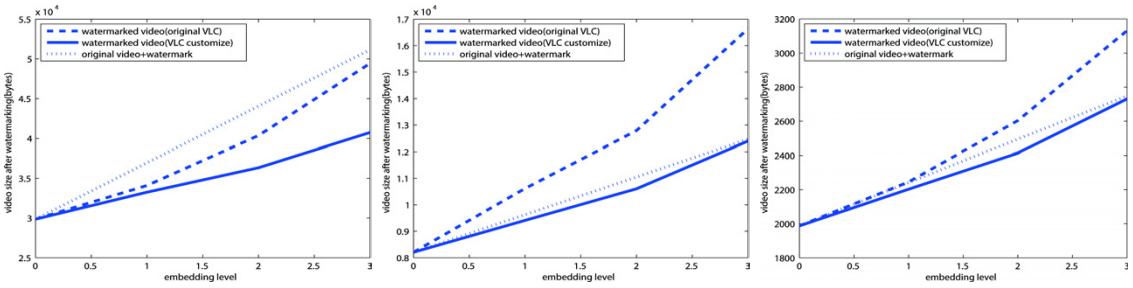
Table 6.1: Image size changes after watermarking

	Aloe (1280×1104)		Art (1384×1104)	
	Original	Watermarked	Original	Watermarked
DC (Mbit)	0.376	0.376	0.376	0.376
AC (Mbit)	3.75	4.45	2.20	2.97
Add (Mbit)	0.7		0.77	
Add (bpp)	0.49		0.51	
Capacity (bpp)	0.49		0.25	
Level	1		2	
Load (bpp)	0.43 (73.5kB)		0.44 (83.2kB)	

6.6.2 Experimental results for H.264 videos

In this section, several videos with various movement and complexity have been used to test our watermarking algorithm. We present the experimental results for three videos: "Horse", "Car" and "Flower". The I- and P-frames of these three video sequences are watermarked with different watermark payload (i.e., different embedding

level). In Fig. 6.11, the size of the watermarked video is compared with the size of the sum of original video and the hidden watermark bytes. "Original VLC" means that the quantized DCT coefficients are encoded by default CAVLC in H.264; "VLC customize" means that the quantized DCT coefficients are encoded by customized CAVLC as discussed in Section 6.3.2. The experimental results (Fig. 6.11) shows that the size of the watermarked video sequences encoded by our customized CAVLC is smaller than the size of the watermarked sequences encoded by the standard CAVLC encoder. Further more, the size of the watermarked video is even smaller than the sum of original video size plus the watermark size. In this case, we can save the transmission bandwidth if the watermark information (i.e., alpha map) needs to be transmitted.

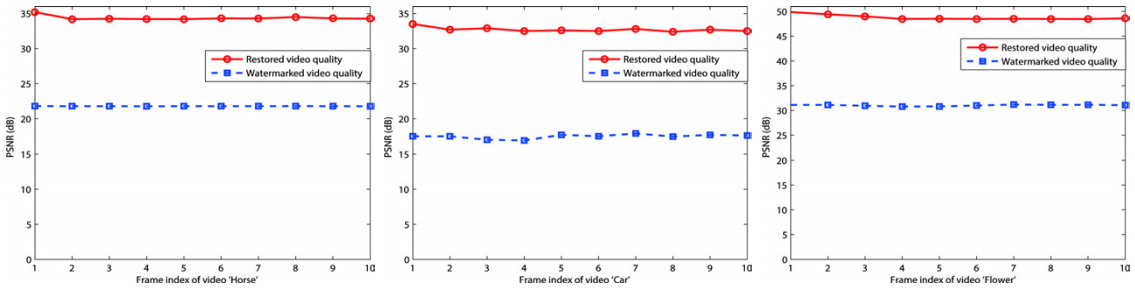


(a) Video "Horse" (b) Video "Car" (c) Video "Flower"

Figure 6.11: Size comparison for watermarked videos.

In Fig. (6.12), the compressed alpha information is hidden into the H.264 video sequence by using our proposed watermarking method. At the receiver side, alpha information can be extracted from the compressed color video sequence without distortion. In order to represent the reversibility of our proposed watermarking algorithm, the watermarked video is also restored after alpha information extraction. In Fig. 6.12, The PSNR values are calculated for watermarked videos and restored videos respectively compared to the raw color video data. The quality loss of restored videos is only caused by H.264 compression and the distortion introduced by watermarking is completely removed after restoration. The PSNR of watermarked

video (blue lines in Fig. 6.12) is quite low because large amount of data (i.e., alpha information) is hidden into the original video. In this case, illegal customers with no restoration algorithm cannot obtain high quality video sequences. On the other hand, high quality video sequences (red lines in Fig. 6.12) can be obtained after the alpha information is extracted and the watermarked videos are restored. The restored videos have the same quality with the ones which are only H.264 compressed without watermarking.



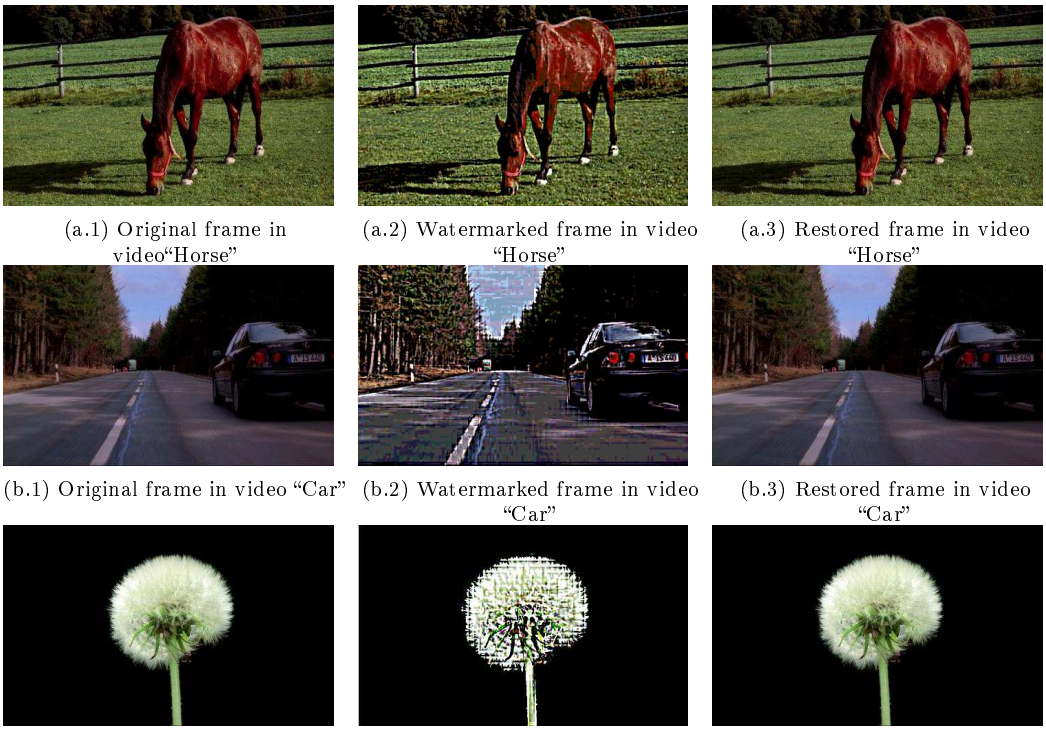
(a) PSNR for 10 frames in video "Horse"

(b) PSNR for 10 frames in video "Car"

(c) PSNR for 10 frames in video "Flower"

Figure 6.12: PSNR comparison before and after restoration.

In Fig. 6.13, one frame from each of the three videos is presented to compare the difference between the original, watermarked and recovered frames. The watermark (alpha information) can be completely hidden into the compressed video sequence and be extracted without any distortion. We can see that the quality of the watermarked videos is quite low if the videos are decoded without knowing the watermark embedding. However the quality of the restored videos is just the same as the original compressed videos. In this case, the access control can be achieved so that only the users with the knowledge of the embedded watermark can restore the original video quality and also extract the hidden alpha information.



(a.1) Original frame in video“Horse”

(a.2) Watermarked frame in video “Horse”

(a.3) Restored frame in video “Horse”

(b.1) Original frame in video “Car”

(b.2) Watermarked frame in video “Car”

(b.3) Restored frame in video “Car”

(c.1) Original frame in video “Flower”

(c.2) Watermarked frame in video “Flower”

(c.3) Restored frame in video “Flower”

Figure 6.13: The original, watermarked, and recovered video frames. alpha information is H.264 compressed and hidden into color frames in (a.1), (b.1) and (c.1) respectively. The watermarked frames are shown in (a.2), (b.2) and (c.2). And these watermarked frames can be restored to high quality frames as shown in (a.3), (b.3) and (c.3). Note that the frames in (a.3), (b.3) and (c.3) are identical to the frames in (a.1), (b.1) and (c.1).

7

Conclusions and future work

7.1 Conclusion

In this thesis, a novel chroma-keying system is proposed to improve the accuracy and reliability of transparency estimation. Given a solid background color image, the color statistics and local lightness variation are respectively analyzed to extract clean background regions. Based on human visual perception, the absolute foreground region and the potential reflective region are also segmented. By using these procedures, an image is exclusively segmented into four regions: foreground, background, reflective and transparent regions. Given the known background region and the color in it, background color is smoothly propagated to the whole image by minimizing an energy function, which is derived from the 2D Laplacian equation. The foreground color estimation is based on GMM color representation model with the concern of

comprehensive color sampling and low computational complexity. Compared to other chroma-keying or alpha matting methods, reflective region is no longer considered as transparency in our proposed method and it is processed differently from transparent region. The proposed method can robustly deal with images with background light variation and can significantly improve matting results when there is reflective part on the foreground object. Furthermore, a reversible watermarking algorithm is developed to insert the alpha channel into its host color image/video. By doing this, the color and alpha channel can be both encrypted so that only authorized users can get access to the original image/video information.

7.2 Discussion of future work

The matting problem is a research of great commercial and scientific value because it is widely used in daily life and it deeply investigates the properties, semantics, and relations of the colors and textures in an image. Since matting problems often arise in image and video editing, the efficiency and automatism are two very important concerns. Considering the efficiency, parallel computing is a suitable choice for this problem since the foreground/background prediction and alpha estimation are often pixel-wise processed. As for the automatism, a good chroma keying system is one that can robustly generate high quality mattes in a period of time with few parameters to be tuned. The last but not the least problem is to distinguish between transparency and reflection. There are few matting methods that can universally settle this problem. In practice, this problem is often avoided by carefully setting the environment lighting. Although our proposed method can distinguish between the reflection and transparency in many situations, it still fails in some cases as illustrated in the previous chapters. We can still try to find the promising solutions from two aspects: (1) physical approaches such as polarized lenses and polarized reflective background; (2) mathematical approaches such as image modeling based on intrinsic layer and environmental lighting layer.

References

[1] Mahanmud, S. *et al. Segmentation of Multiple Salient Closed Contours from Real Images.* IEEE Transactions on Pattern Analysis and Machine Intelligence, **25**(4):433–444 (2003). (Cited on pages x, 4, and 5.)

[2] Cox, D.D. *Do We Understand High-level Vision?* Current Opinion in Neurobiology, **25**:187–193 (2014). (Cited on pages x, 5, and 6.)

[3] Porter, T. and Duff, T. *Compositing Digital Images.* In *Proceedings of ACM SIGGRAPH*, pages 253–259 (1984). (Cited on pages x, 1, 7, 12, 30, 31, and 32.)

[4] Bowmaker, J.K. and Dartnall, H.J. *Visual pigments of rods and cones in a human retina.* The Journal of Physiology, **298**:510–511 (1980). (Cited on pages x, 33, and 34.)

[5] Vlahos, P. *Composite photography utilizing sodium vapor illumination.* U.S. Patent 3,095,304, June 25, 1963. (Cited on pages xi, 43, and 44.)

[6] *Alpha matting benchmark.* http://www.alphamatting.com/index.html. Accessed: 2015-12-1. (Cited on pages xiv, 15, 54, and 135.)

[7] Fu, K.S. and Mui, J.K. *A Survey on Image Segmentation.* Pattern Recognition, **13**(1):3–16 (1981). (Cited on page 3.)

[8] Zhang, Y.J. *A Survey on Evaluation Methods for Image Segmentation.* Pattern Recognition, **29**(8):1335–1346 (1996). (Cited on page 3.)

[9] Zhang, H. *et al. Image Segmentation Evaluation: A Survey of Unsupervised Methods.* Computer Vision and Image Understanding, **110**(2):260–280 (2008). (Cited on page 3.)

[10] Wu, J. *et al. Reverse Image Segmentation: A High-Level Solution to a Low-Level Task.* In *British Machine Vision Conference, BMVC 2014* (2014). (Cited on page 4.)

[11] Martin, D. *et al. A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics.* In *Proceedings of IEEE International Conference on Computer Vision, ICCV 2001*, pages 416–423 (2001). (Cited on pages 4 and 65.)

[12] Canny, J. *A Computational Approach to Edge Detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, **8**(6):679–698 (1986). (Cited on page 5.)

[13] Wang, J. and Cohen, M.F. *Image and Video Matting: A Survey.* Foundations and Trends in Computer Graphics and Vision, **3**(2):97–175 (2007). (Cited on pages 6, 12, and 24.)

[14] Zhu, Q. *et al. Targeting Accurate Object Extraction From an Image: A Comprehensive Study of Natural Image Matting.* IEEE Transactions on Neural Networks and Learning Systems, **26**(2):185–207 (2014). (Cited on pages 6, 12, and 24.)

[15] Juan, O. and Keriven, R. *Trimap segmentation for fast and userfriendly alpha matting.* In *Variational Geometric, and Level Set Methods in Computer Vision*, pages 186–197 (2005). (Cited on page 8.)

[16] Rhemann, C. *et al. High resolution matting via interactive trimap segmentation.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008*, pages 1–8 (2008). (Cited on page 8.)

[17] Guan, Y. *et al. Easy matting: A stroke based approach for continuous image matting.* In *Proceedings of Eurograph*, pages 567–576 (2006). (Cited on page 8.)

[18] Zheng, Y. *et al. Fuzzy Matte: A computationally efficient scheme for interactive matting.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008*, pages 1–8 (2008). (Cited on page 8.)

[19] Rhemann, C. *et al. A Perceptually Motivated Online Benchmark for Image Matting.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009*, pages 1826–1833 (2009). (Cited on page 15.)

[20] *Study on Chroma Key Shooting Formats.* Technical report, Quebéc Film and Television Council. http://www.qftc.ca/uploads/files/projects/bctq-study-on-chroma-key-shooting-formats.pdf. Accessed: 2015-12-1. (Cited on page 15.)

[21] Blinn, J. *What is a pixel?* Computer Graphics and Appications, **25**(5):82–87 (2005). (Cited on page 24.)

[22] Shen, Y. *et al. Image Matting From a Physical Perspective.* In *the Forty-Third Asilomar Conference on Signal Systems and Computers*, pages 841–845 (2009). (Cited on pages 26, 28, and 29.)

[23] Cohen, M.F. *et al. Radiosity and Realistic Image Synthesis.* Academic Press Professional, San Diego, CA, USA (1993). (Cited on page 26.)

[24] Nathans, J. *et al. Molecular genetics of human color vision: the genes encoding blue, green, and red pigments.* Science, **232**(4747):193–202 (1986). (Cited on page 33.)

[25] Curcio, C.A. *et al. Human photoreceptor topography.* The Journal of Comparative Neurology, **292**(4):497–523 (1990). (Cited on page 33.)

[26] Thomas, J.B. and Paris, D.N.S. *CRC Handbook of Fundamental Spectroscopic Correlation Charts.* CRC Press, New York, NY, USA (2005). (Cited on page 33.)

[27] Smith, T. and Guild, J. *The C.I.E colorimetric standards and their use.* Transactions of the Optical Society, **33**(3):73–134 (1931). (Cited on page 35.)

[28] Zhu, Q. *et al. Targeting accurate object extraction from an image: a comprehensive study of natural image matting.* IEEE Transactions on Neural Networks and Learning Systems, **26**(2):185–207 (2015). (Cited on page 40.)

[29] Smith, A.R. and Blinn, J.F. *Blue screen matting.* In *Proceedings of SIGGRAPG*, pages 259–268 (1996). (Cited on page 40.)

[30] Chuang, Y.Y. *et al. Video matting of complex scenes.* In *Proceedings of SIGGRAPG*, pages 243–248 (2002). (Cited on page 40.)

[31] Pham, V.Q. *et al. Real-time video matting based on bilayer segmentation.* In *Proceedings of 9th Asian Conference on Computer Vision*, pages 489–501 (2009). (Cited on page 40.)

[32] Wu, T.P. *et al. Natural shadow matting.* ACM Transactions on Graphics, **26**(2):8 (2007). (Cited on page 40.)

[33] Chuang, Y.Y. *et al. Shadow matting and compositing.* ACM Transactions on Graphics, **22**(3):494–500 (2003). (Cited on page 40.)

[34] Zongker, D.E. *et al. Environment matting and compositing.* In *Proceedings of SIGGRAPG*, pages 205–214 (1999). (Cited on page 40.)

[35] Chuang, Y.Y. *et al. Environment matting extensions: Towards higher accuracy and real-time capture.* In *Proceedings of SIGGRAPG*, pages 121–130 (2000). (Cited on page 40.)

[36] Williams, F.D. *Method of taking motion-pictures.* U.S. Patent 1,273,435, July 23, 1918. (Cited on page 42.)

[37] Dunning, C.D. *Method of producing composite photographs.* U.S. Patent 1,613,163, January 4, 1927. (Cited on page 42.)

[38] Vlahos, P. *Composite color photography.* U.S. Patent 3,158,477, November 24, 1964. (Cited on page 43.)

[39] Vlahos, P. *Electronic composite photography.* U.S. Patent 3,595,987, July 24, 1971. (Cited on page 43.)

[40] Vlahos, P. *Electronic composite photography with color control.* U.S. Patent 4,007,487, February 8, 1977. (Cited on page 43.)

[41] Vlahos, P. *Comprehensive electronic compositing system.* U.S. Patent 4,100,569, July 11, 1978. (Cited on pages 43, 45, and 46.)

[42] Dadourian, A. *Method and apparatus for compositing video images.* U.S. Patent 5,343,252, August 30, 1994. (Cited on page 46.)

[43] Mishima, Y. *Soft edge chroma-key generation based upon hexoctahedral color space.* U.S. Patent 5,355,174, October 11, 1994. (Cited on page 47.)

[44] Liu, Y. *et al. Method, system, and device for automatic determination of nominal backing color and a range thereof.* U.S. Patent 7,508,455, March 24, 2009. (Cited on page 47.)

[45] Berman, A. *et al. Method for removing from an image the background surrounding a selected object.* U.S. Patent 6,134,346, October 17, 2000. (Cited on page 50.)

[46] Berman, A. *et al. Comprehensive method for removing from an image the background surrounding a selected subject.* U.S. Patent 6,134,345, October 17, 2000. (Cited on pages 50 and 51.)

[47] Ruzon, M.A. and Tomasi, C. *Alpha estimation in natural images.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2000*, pages 18–25 (2000). (Cited on pages 50 and 51.)

[48] Chuang, Y.Y. *et al. A Bayesian approach to digital matting.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2001*, pages 264–271 (2001). (Cited on pages 50 and 51.)

[49] Chuang, H. *et al. An iterative Bayesian approach for digital matting.* In *Proceedings of 18th International Conference on Pattern Recognition, (ICPR) 2006*, pages 122–125 (2006). (Cited on page 52.)

[50] Wang, J. and Cohen, M.F. *Optimized color sampling for robust matting.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007*, pages 1–8 (2007). (Cited on pages 53 and 55.)

[51] Gastal, E.S.L. and Oliveira, M.M. *Shared sampling for real time alpha matting.* Computer Graphics Forum, **29**(2):575–584 (2010). (Cited on page 53.)

[52] He, K. *et al. A global sampling method for alpha matting.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011*, pages 2049–2056 (2011). (Cited on pages 53 and 54.)

[53] Shahrian, E. *et al. Improving image matting using comprehensive sampling sets.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, pages 636–643 (2013). (Cited on pages 53 and 54.)

[54] Bai, X. and Sapiro, G. *A geodesic framework for fast interactive image and video segmentation and matting.* In *Proceedings of 11th IEEE International Conference on Computer Vision, ICCV 2007*, pages 1–8 (2007). (Cited on page 53.)

[55] Price, B.L. *et al. Geodesic graph cut for interactive image segmentation.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010*, pages 3161–3168 (2010). (Cited on page 53.)

[56] Barnes, C. *et al. PatchMatch: A randomized correspondence algorithm for structural image editing.* ACM Transactions on Graphics, **28**(3):24:1–24:11 (2009). (Cited on page 54.)

[57] Shahrian, E. and Rajan, D. *Weighted color and texture sample selection for image matting.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*, pages 718–725 (2012). (Cited on page 54.)

[58] Shahrian, E. and Rajan, D. *Weighted color and texture sample selection for image matting.* IEEE Transactions on Image Processing, **22**(11):4260–4270 (2013). (Cited on pages 54 and 127.)

[59] Shahrian, E. and Rajan, D. *Using texture to complement color in image matting.* Image and Vision Computing, **31**(9):658–672 (2013). (Cited on page 54.)

[60] Grady, L. *et al. Random walks for interactive alpha matting.* In *Proceedings of international Conference on Visualization, Imaging, and Image Processing, VIIP 2005*, pages 423–429 (2005). (Cited on pages 57 and 60.)

[61] Rhemann, C. *et al. A spatially varying PSF-based prior for alpha matting.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010*, pages 2149–2156 (2010). (Cited on page 57.)

[62] Rhemann, C. *et al. Improving color modeling for alpha matting.* In *Proceedings of British Machine Vision Conference, BMVC 2008* (2008). (Cited on page 57.)

[63] Sun, J. *et al. Poisson matting.* ACM Transactions on Graphics, **23**(3):315–321 (2004). (Cited on page 57.)

[64] Du, Z. *et al. Oriented Poisson matting.* In *Proceedings of IEEE International Conference on Image Processing, ICIP 2005*, pages 626–629 (2005). (Cited on pages 57 and 58.)

[65] Levin, A. *et al. A closed form solution to natural image matting.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2006*, pages 61–68 (2006). (Cited on pages 58 and 127.)

[66] Levin, A. *et al. Spectral matting.* IEEE Transactions on Pattern Analysis and Machine Intelligence, **30**(10):1699–1712 (2008). (Cited on page 58.)

[67] He, K. *et al. Fast matting using large kernel matting Laplacian matrices.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010*, pages 2165–2172 (2010). (Cited on page 59.)

[68] He, X. and Niyogi, P. *Advances in Neural Information Processing Systems 16.* MIT Press, Cambridge, MA, USA (2003). (Cited on page 60.)

[69] Buades, A. *et al. A non-local algorithm for image denoising.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2005*, pages 60–65 (2005). (Cited on page 60.)

[70] Lee, P. and Wu, Y. *Nonlocal matting.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011*, pages 2193–2200 (2011). (Cited on page 60.)

[71] Chen, Q. *et al. KNN matting.* IEEE Transactions on Pattern Analysis and Machine Intelligence, **35**(9):2175–2188 (2013). (Cited on page 60.)

[72] Chen, Q. *et al. KNN matting.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*, pages 869–876 (2012). (Cited on page 60.)

[73] Rother, C. *et al.* *GrabCut: Interactive foreground extraction using iterated graph cuts.* ACM Transactions on Graphics, **23**(3):309–314 (2004). (Cited on page 61.)

[74] Wang, J. and Cohen, M.F. *An iterative optimization approach for unified image segmentation and matting.* In *Proceedings of 10th IEEE International Conference on Computer Vision, ICCV 2005*, pages 936–943 (2005). (Cited on page 61.)

[75] Weiss, Y. and Freeman, W.T. *On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs.* IEEE Transactions on Information Theory, **47**(2):736–744 (2001). (Cited on page 62.)

[76] Omer, I. and Werman, M. *Color lines: image specific color representation.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2004*, pages 946–953 (2004). (Cited on page 64.)

[77] *Berkeley Segmentation Dataset.* http://https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/. Accessed: 2015-12-1. (Cited on pages 65 and 94.)

[78] Hill, F.S.J. *Computer Graphic.* Macmillna Publishing, London (1990). (Cited on pages 67 and 93.)

[79] Heckbert, P. *Color image quantization for frame buffer display.* ACM SIGGRAPH Computer Graphics, **16**(3):297–307 (1982). (Cited on page 68.)

[80] Wu, X. *Efficient statistical computations for optimal quantization.* Academic Press Professional, San Diego, CA, USA (1991). (Cited on pages 68 and 93.)

[81] Gervautz, M. and Purgathofer, W. *A Simple Method for Color Quantization: Octree Quantization.* Academic Press Professional, San Diego, CA, USA (1990). (Cited on page 68.)

[82] MacQueen, J. *Some methods for classification and analysis of multivariate observations.* In *Proceedings of 5th Symposium on Mathematical Statistics and Probability*, pages 281–297 (1967). (Cited on pages 68 and 93.)

[83] Dunn, J.C. *Well separated clusters and optimal fuzzy partitions.* Journal of Cybernetics, **4**:95–104 (1974). (Cited on pages 68 and 93.)

[84] Liew, A.W.C. and Lau, S.H. *Fuzzy image clustering incorporating spatial continuity.* In *IEEE Proceedings of Vision, Image and Signal Processing*, pages 185–192 (2000). (Cited on page 68.)

[85] Pearson, K. *Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material.* Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, **186**:343–414 (1895). (Cited on page 73.)

[86] Eilers, P.H. and Goeman, J.J. *Enhancing scatterplots with smoothed densities.* Bioinformatics, **20**:623–628 (2004). (Cited on page 76.)

[87] Meyer, F. *Topographic distance and watershed lines.* Mathematical Morphology and its Applications to Signal Processing, **38**(1):113–125 (1994). (Cited on page 78.)

[88] Beucher, S. and Lantuéjoul, C. *Use of watersheds in contour detection.* In *Workshop on Image Processing, CCETT/IRISA*, pages 2.1–2.12 (1979). (Cited on page 78.)

[89] Lloyd, S.P. *Least squares quantization in PCM.* IEEE Transactions on Information Theory, **28**:129–137 (1982). (Cited on page 85.)

[90] Friedmak, J.H. *et al. An algorithm for finding best matches in logarithmic expected time.* ACM Transactions on Mathematical Software, **3**:209–226 (1977). (Cited on page 85.)

[91] *Adobe After Effects*. http://www.adobe.com/products/aftereffects.html. Accessed: 2015-12-1. (Cited on page 127.)

[92] *Primatte Whitepapers*. http://www.primatte.com/content.cfm?n=whitepapers. Accessed: 2015-12-1. (Cited on page 127.)

[93] *Industry Dataset for Chroma Keying*. http://www.hollywoodcamerawork.com/greenscreenplates.html. Accessed: 2015-12-1. (Cited on page 127.)

[94] Cox, I.J. *et al. Digital Watermarking*. Morgan Kaufmann, San Francisco, CA, USA (2001). (Cited on pages 140 and 144.)

[95] Khan, A. *et al. Hiding depth map of an object in its 2D image: Reversible watermarking for 3D cameras*. In *IEEE International Conference on Consumer Electronics*, pages 1–2 (2009). (Cited on page 143.)

[96] Cultuc, D. and Caciula, I. *On stereo embedding by reversible watermarking: Further results*. In *International Symposium on Signals, Circuits and Systems*, pages 121–124 (2009). (Cited on page 143.)

[97] Cultuc, D. *et al. Color stereo embedding by reversible watermarking*. In *International Symposium on Electrical and Electronics Engineering*, pages 256–259 (2010). (Cited on page 143.)

[98] Ellinas, J.N. *Reversible watermarking on stereo image sequences*. Internationl Journal of Signal Processing, **5**(3):210–215 (2009). (Cited on page 143.)

[99] Lam, E.Y. and Goodman, J.W. *A mathematical analysis of the DCT coefficient distributions for images*. IEEE Transactions on Image Processing, **9**(10):1661–1666 (2000). (Cited on page 144.)

[100] Vetro, A. *et al. Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard*. In *Proceedings of IEEE*, pages 626–642 (2011). (Cited on page 144.)

[101] Shannon, C.E. *A mathematical theory of communication.* ACM SIGMOBILE Mobile Computing and Communications Review, **5**(1):3–55 (2001). (Cited on page 146.)

[102] Alattar, A.M. *Reversible watermark using the difference expansion of a generalized integer transform.* IEEE Transactions on Image Processing, **13**(8):1147–1156 (2004). (Cited on page 148.)

[103] Wang, W. and Zhao, Y. *Robust image chroma-keying: a quadmap approach based on global sampling and local affinity.* IEEE Transactions on Broadcasting, **61**:356–366 (2015). (Not cited.)