Color range determination and alpha matting for color images

by

Zhenyi Luo

A thesis submitted to the University of Ottawa in partial fulfillment of the requirements for the degree of Master of Computer Science

Ottawa-Carleton Institute for Computer Science
School of Electrical Engineering and Computer Science
University of Ottawa

Ottawa, Ontario, Canada

September 2011

Abstract

This thesis proposes a new chroma keying method that can automatically detect background, foreground, and unknown regions. For background color detection, we use K-means clustering in color space to calculate the limited number of clusters of background colors. We use spatial information to clean the background regions and minimize the unknown regions. Our method only needs minimum inputs from user.

For unknown regions, we implement the alpha matte based on Wang's robust matting algorithm, which is considered one of the best algorithms in the literature, if not the best. Wang's algorithm is based on modified random walk. We proposed a better color selection method, which improves matting results in the experiments. In the thesis, a detailed implementation of robust matting is provided.

The experimental results demonstrate that our proposed method can handle images with one background color, images with gridded background, and images with difficult regions such as complex hair stripes and semi-transparent clothes.

Contents

List of Figures

List of Acronyms

BP	Belief Propagation
CG	Conjugate Gradient
CRF	Conditional Random Field
FC	Fuzzy Connectedness
FOV	Field of View
GMM	Guassian Mixture Model
HDR	High Dynamic Range
HVS	Human Visual System
MAP	Maximum A Posteriori
ML	Maximum Likelihood
MRF	Markov Random Field
MSE	Mean Squared Error
PCA	Principal Component Analysis
PDF	Probability Density Function
SAD	Sum of Absolute Differences
SOR	Successive Over Relaxation
SSIM	Structural SIMilarity
UI	User Interface

This thesis is dedicated to my parents.

Acknowledgement

First of all, I would like to deeply thank my supervisor, Professor Jiying Zhao, for bringing the problem of chroma keying to me, and for his valuable directions, delicate guidance and quick feedback during every step of my work. I greatly appreciate his patience and confidence in my research abilities.

Special thanks to Dr. Jue Wang of Adobe Systems Inc. and Dr. Yu Liu of Ross Video Ltd., for their generosity in sharing their knowledge and experience with me.

Great thanks to my colleague and best friend: Wenyi Wang. His knowledge and research interest helps and encourages me a lot.

I would also like to thank the current and former colleagues in Multimedia Communication Research Laboratory (MCRLab) at School of Electrical Engineering and Computer Science in University of Ottawa: Sha Wang, Chengcheng Hao, Yu Zhang, Han Wang, for their great support and generous help in encouraging and improving my research work.

Finally, I am greatly indebted to my best friend: Jing Cao. She has always been the biggest support to my life and study.

Chapter 1

Introduction

1.1 Chroma keying and alpha matting

Chroma keying or compositing is a process of extracting a foreground object with arbitrary shape from an image or video, and compositing this foreground object with a new background. Extracted foreground object is called key. The background can be considered as transparent part, and supposed to be replaced by any other background. This technology is widely used in weather forecast. The news presenter appears to be standing in front of a large weather map during the live broadcasts, but actually he or she is standing in a television studio where the background is blue or green. These blue or green regions are replaced by different weather maps.

Another common application of chroma keying is making special effects, such as replacing the background scene with another image. Larry Butler, who first introduced this technology to film industry and won the Academy Award, used it to create special effects for "The Thief of Bagdad" in 1940. Till now, nearly all modern movies such as "The Lord of the Rings", "Avatar", and "The Matrix" utilize this technology in their production. With the development and great success of 3D and IMAX technologies, chroma keying has great commercial potential. This technology has also been applied to 2D arts and graphics, magazines, and image editing software (e.g., Photoshop).

Earlier chroma keying approaches only segment image or individual video frame into two parts: foreground part and background part, by determining the color range of them. In order to meet the requirements of generating more accurate matte for complex images, advanced geometrical shapes are introduced to calculate the full and partial coverage of the foreground object. But these methods still cannot well deal with the part where detailed information (e.g., the hair stripes of woman) exists. Most chroma keying approaches cannot preserve complete structure of the hair stripes, especially when the hair stripes are long and thin. In addition, traditional chroma keying methods always require that the foreground object be photographed against a constant color background. It cannot calculate the key for natural image that the background can be any color.

In order to resolve these problems, a new technology named "alpha matting" emerged. Its objective is to generate high quality key for foreground object, especially for accurately calculating the extent of partial coverage for hair and semi-transparent part. Compared with traditional chroma keying methods, which have already been applied to industrial field for decades, alpha matting is a novel research topic. Instead of using complex geometrical shapes to determine the color range of foreground and background, they first use more complicated mathematical models (e.g., mixed Gaussian distribution model, Bayesian approach based on maximum a posteriori probability) to estimate the foreground and background color in the semi-transparent or transition area, and then calculate the foreground and background blending factor as "alpha matte".

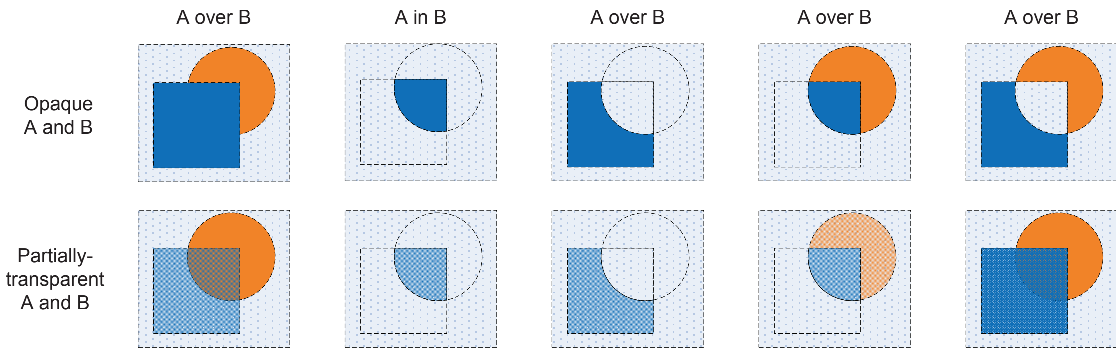


Figure 1.1: Rendering operations [1].

The concept of alpha channel was first introduced by Porter and Duff in 1984 [2]. Interestingly, the authors consider the problem of object overlap in image rendering or compositing field instead of foreground extraction, which are totally two inverse processes. Image rendering can be viewed as an inverse process of foreground extraction, because it generates an image from a model rather than extracts foreground objects from an image. This model is usually called scene file, which may contains detailed information of image, such as shading, texture, viewpoint, lighting and geometry information.

Porter and Duff introduced basic image operations to express image rendering based on alpha channel, as shown in Fig. 1.1. Suppose A and B are two image elements, in order to generate an image element that A appears in the foreground and B appears in the background, a basic operation that A over B is derived. Other common operations are in, out, atop and xor. The compositing of opaque elements is used for earlier chroma keying methods, while partial transparent elements compositing is applied in recent chroma keying methods which considers the problem of partial coverage of foreground object.

In order to composite a foreground object to an arbitrary background, a mixing factor “alpha” is needed to control the linear interpolation of foreground and background colors. Suppose the observed image pixel $I_z(z = (x, y))$ in an input image is a linear combination of foreground image color F_z and background image color B_z, the equation is defined as follows:

$$I_z = \alpha_z F_z + (1 - \alpha_z) B_z \tag{1.1}$$

where α_z can be any value between 0 and 1. If $\alpha_z = 1$, this pixel is pure foreground. If $\alpha_z = 0$, this pixel is pure background. In other cases, the pixel I_z is mixed with foreground and background color.

Besides RGB channel, alpha channel is introduced as the fourth channel in the rendering process [2], which can be used to deal with the object overlap problem for anti-aliasing purpose. Therefore, instead of using triple channel (r, g, b), the researchers use a quadruple channel (r, g, b, α) to render the color of pixels. For example, to display a pixel with RGB color $(250, 143, 33)$ with half partial coverage, we just need to store $(250, 143, 33, 0.5)$ for that pixel.

Researchers borrow the idea of alpha channel from image rendering to chroma keying. It is an inverse process that extracts the foreground from an image rather than renders two image elements. Chroma keying needs to determine the alpha value for the boundary and semi-transparent of foreground in an image. Suppose digital pictures are recorded as three channels X, Y, Z, derived from Equ. (1.1), matting algorithms need to determine the value for "α_z" in composition equation with 10 variables:

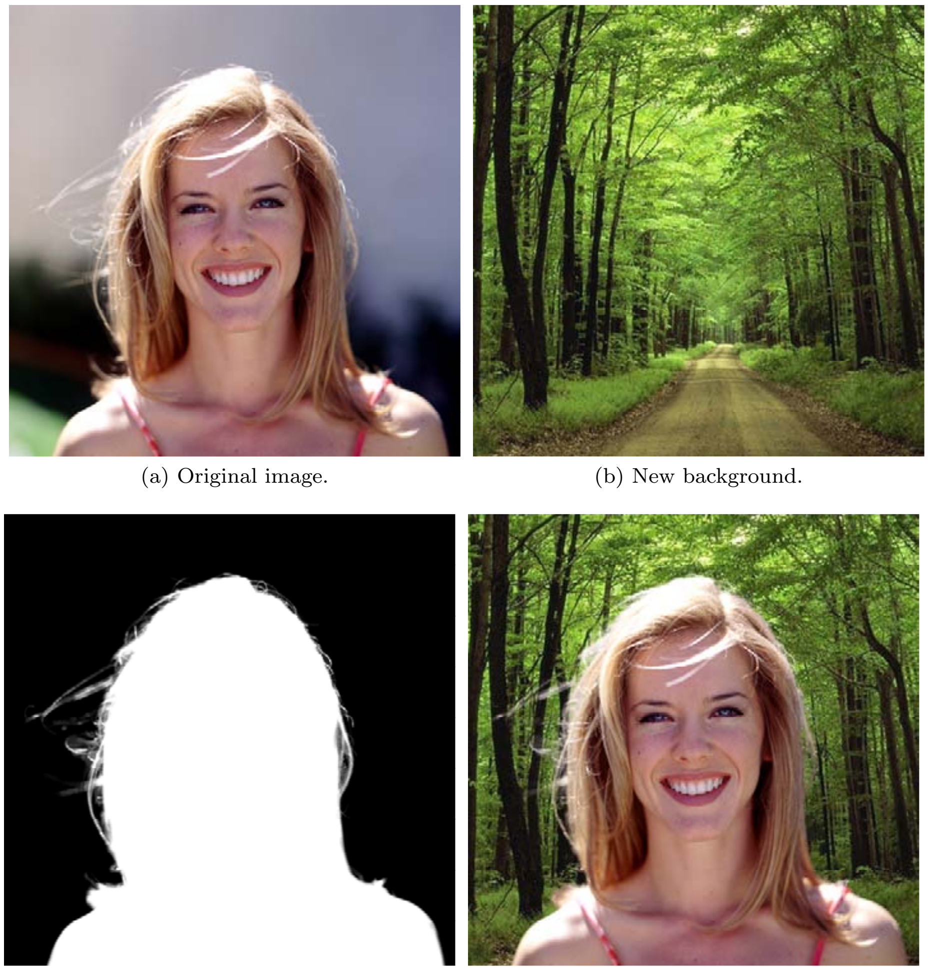
$$\begin{pmatrix} I_{zX} \\ I_{zY} \\ I_{zZ} \end{pmatrix} = \alpha_z \begin{pmatrix} F_{zX} \\ F_{zY} \\ F_{zZ} \end{pmatrix} + (1 - \alpha_z) \begin{pmatrix} B_{zX} \\ B_{zY} \\ B_{zZ} \end{pmatrix} \tag{1.2}$$

An example of alpha matting is shown in Fig. 1.2. In order to composite an original image (a) with new background (b), we need to calculate an alpha matte (c) first. Obtaining the knowledge of partial and full coverage of foreground, final compositing result is generated in (d).

Although a lot of research work has been done in recent years, alpha matte calculation still remains challenging. Refer to Equ. (1.2), for natural background image, there are seven unknown variables: $F_{zX}, F_{zY}, F_{zZ}, B_{zX}, B_{zY}$, and B_{zZ}. By photographing the foreground object against a known constant background color, three variables B_{zX}, B_{zY}, B_{zZ} are known, so the number of unknown variables reduced to four. However, we have only three observation values: I_{zX}, I_{zY}, I_{zZ}. Therefore, the alpha matting problem for both natural and constant color background are under constrained, which means that there is no unique mathematical solution to this problem. Typical existing approaches of chroma keying for color range determination and high quality alpha matting generation are reviewed in Chapter 2.

1.2 User interaction

Chroma keying for color range determination needs the background color information as priori knowledge. For instance, some chroma keying systems require users to specify several background color regions. As shown in Fig. 1.3, a black square is used to mark



(a) Original image. (b) New background.

(c) Extracted alpha matte. (d) Compositing two images together.

Figure 1.2: An example of alpha matting.

the background region, which covers the shadow and different brightness part. Other chroma keying approaches may need users to provide the rough background color in the initialization process.

Instead of providing rough background color range, almost all alpha matting approaches start from having the priori knowledge of image segmentation. It is done manually by users drawing lines or regions on the image. Originally, some matting algorithms ([3] [4] [5] [6]) are based on trimap. Trimap is a rough pre-segmented im-

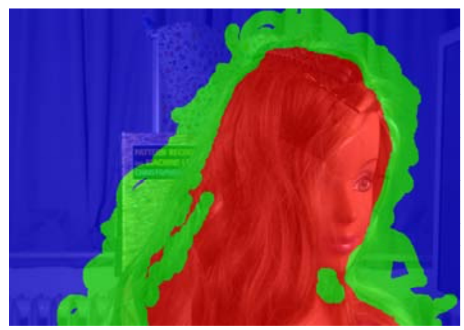
Figure 1.3: An example of input for color range determination [7].

age consisting foreground, background and unknown regions. It should be guaranteed that the foreground contour is located completely in the foreground region, and the background contour is located completely in the background region. A typical trimap example is given in Fig. 1.4. For an original image (a), user can manually draw the trimap (b) to mark the foreground, background and unknown region. As shown in (c), foreground part is drawn with red color, background is filled with blue color, and green color indicates unknown region.

One of the important factors affecting the performance of chroma keying algorithms is the accuracy of the trimap. It is obvious that the unknown region not only covers pixels whose alpha values are actually between 0 and 1, it also covers pure foreground and background pixels. As what has been discussed before, partial coverage values are computed only inside the unknown region. The more unnecessary pixels it covers, the more computation time it requires. In addition, if the unknown region is too thick, which means that the unknown pixels are far from the known region, the estimation will be less accurate. This also indicates that the boundary of the unknown region should be as thin as possible. Accurate trimap makes more pixels fit into the linear

(a) Original image.

(b) User specified trimap.



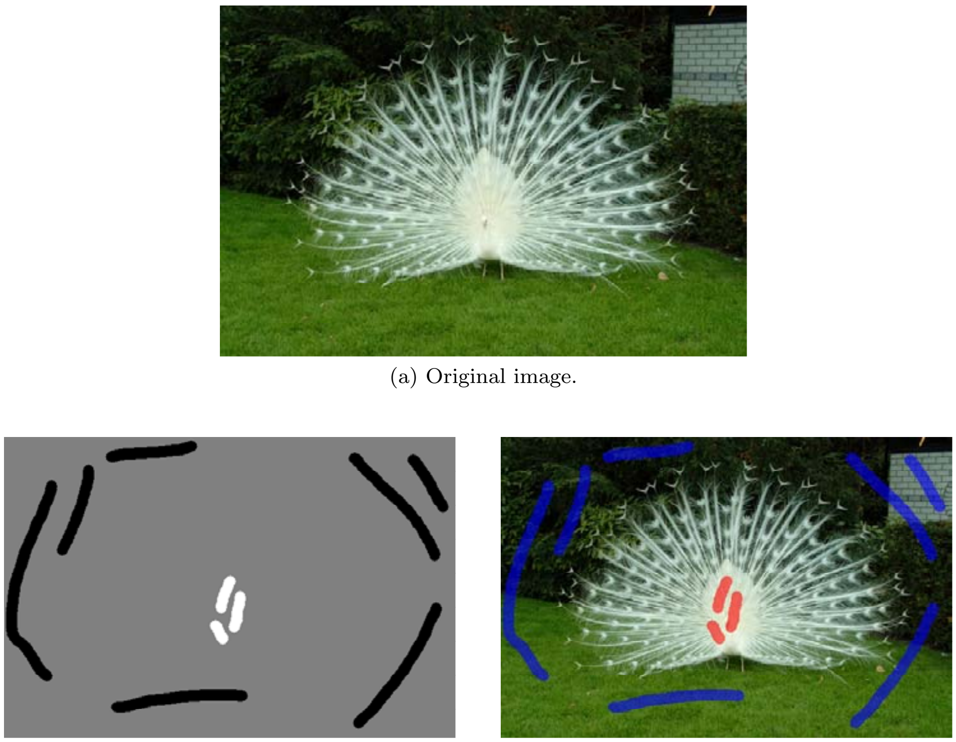
(c) Trimap segmentation on the image.

Figure 1.4: A typical example of trimap for alpha matting [8].

model, and leads to better matting results.

Therefore, the ideal situation is that the unknown region in the trimap only covers the truly mixed pixels. However, this cannot ever happen in reality. No matter how carefully the user pays attention to drawing the trimap, he or she can never create an optimal one. To draw a good trimap, a considerable amount of user interaction is required and it usually takes several minutes to finish the task, especially when the image contains large portion of semi-transparent part. The work to create a trimap is very tedious, as shown in Fig. 1.4 (b). For unskilled users, errors may be introduced if they incautiously mark some boundary parts of unknown area as foreground or background.

Instead of providing a trimap, some recent algorithms ([9] [10] [11] [12] [13]) al-



(a) Original image.

(b) User specified scribbles.

(c) Scribble segmentation on the image.

Figure 1.5: A typical example of scribble for alpha matting [14].

low users to specify a few foreground and background scribbles as input, as shown in Fig. 1.5. For an original image (a), user manually draws scribbles (b) to identify the foreground and background. As shown in (c), red scribbles represent foreground, blue scribbles indicate background. In fact, this approach defines a very coarse trimap by indicating the remaining pixels as the unknown region. Most of these approaches are also compatible with previous ones that use trimap as input. Some of these approaches result in a hard segmentation of the image first, and then a boundary erosion process is used to generate a trimap. Others can compute the alpha matte directly from the paint strokes. These methods are time-saving and user-convenient without providing accurate trimap.

1.3 The contributions of this thesis

This thesis develops a foreground extraction algorithm for constant or even multi background color images. The main contributions are summarized as follows:

1. The thesis adopts mathematical method "K-means" to cluster image. Multi background color pixels are grouped to several clusters. The background region can be determined by only providing a rough background color.

2. A new trimap generation approach is proposed. It releases the users from trivial work that specify the trimap or scribbles of an image. This also results in a stable alpha matting results, because different accuracy of trimaps or user scribbles can lead to different alpha mattes, and our approach does not need them as input.

3. Color range determination and alpha matting are combined together to improve the chroma keying results. The proposed scheme is good at handling hair stripes and semi-transparent part of images, by taking the advantage of confidence evaluation and modified random walk algorithm.

4. A detailed implementation of robust matting system is provided according to Wang's research work [4].

1.4 Thesis structure

This thesis contains six chapters. In Chapter 1, a brief introduction of chroma keying for color range determination and alpha matting for color image is presented. It is followed by introducing basic ideas about user interaction for the two methods. Then, the major contributions of this thesis are summarized.

In Chapter 2, the fundamental theories and techniques of chroma keying for color range determination and alpha matting are reviewed. Several special alpha matting approaches are presented. The motivation of our research work is illustrated at last.

In Chapter 3, the basic idea of proposed chroma keying scheme for color range determination and alpha matting is discussed. This scheme is based on K-means clustering, automatic trimap generation, confidence evaluation and modified random walk. The advanced mathematical models and equations are used to generate good results in the thesis.

In Chapter 4, the implementation process is presented step by step with the intermediate results. The final experimental results are demonstrated and compared with Wang's results, which shows the advantages of our proposed method.

In Chapter 5, conclusions are drawn and the future work for this research is discussed.

Chapter 2

Literature review

Our research work origins from the research of color range determination for chroma keying and then expands to high quality matte generation. In this chapter, we will first introduce color range determination methods, and then talk about related work for alpha matte generation. Finally, the motivation of our research work is stated.

2.1 Color range determination

Methods related to color range determination for chroma keying are introduced first. Compared with the approaches in alpha matting which are mainly implemented in academic field, these methods have already been applied to the industry and preserved as the form of patents. To avoid confusion, we clarify that some of these methods do not adopt the idea of alpha channel, because they only deal with the foreground and background region, and do not consider the semi-transparent part. Novel methods consider the partial coverage of the foreground, and even the shadow region when they are extracting the foreground. We will start from introducing simple histogram-based methods, and then present advanced geometrical shape-based methods.

2.1.1 Histogram based approaches

Histogram based approaches determine the chroma key color range based on histograms of one or more color components. The histogram of the component can be viewed as threshold to decide whether a pixel is foreground or background.

A typical chroma key generation system is invented by Vaughn Iverson [15] for Intel Corporation in 1998. It assumes that the first frame in the video contains only background color, and the background setting does not change during the keying process. This system contains four processing steps.

1. Select the first frame from video stream, which only contains the background color information.

2. Generate histograms for R, G, and B components respectively. Each histogram represents the number of different component values in the image.

3. Clip outlying values in each histogram to identify initial minimum and maximum values for each component. This step can be accomplished by calculating the number of pixels outside the initial range or by dividing the number of pixels in the image through a specified constant.

4. Calculate the final minimum and maximum values that can be used as threshold to define the color range of the foreground and background as follows:

$$Min_final = Min - (Max - Min)/Const \tag{2.1}$$

$$Max_final = Max + (Max - Min)/Const \tag{2.2}$$

The resulting ranges are used as thresholds in determining whether a pixel in the subsequent image is foreground color or background color. If R, G, and B components of a pixel fall within the minimum and maximum range, the pixel belongs background color. Otherwise, it is foreground color.

However, this method has some disadvantages. Firstly, it does not calculate the partial coverage of foreground object, thus image with semi-transparent part cannot be well dealt with. Therefore, it can be categorized as hard or binary segmentation method.

In addition, the system only uses one-dimensional histogram, which means it only encompasses a rectangle space to represent background color in color space. While in the real case, the color distribution of background can be any shape. Due to the limitation of the shape, if we only use a rectangle to encompass background color, foreground color may be included and background color may be excluded. Also, when the assumption that the first frame does not contain foreground object does not hold, the system will mistakenly consider some foreground color as background or setting color.

Uya et al. [16] proposed a similar method. They use one-dimensional histogram as Vaughn Iverson [15] does, so it has the same disadvantage. The difference is that it requires users to define background regions on an image during initialization stage, so it calculates histograms only based on those pixels. The results of the chroma keying signal are not the same with different user input. The optimum range for the key color depends on how users choose the background regions.

2.1.2 2D geometrical shape based methods

Since histogram based methods cannot well solved the background color distribution problems, 2D geometrical shape based methods emerged. They project the three dimensional color space into a chromatic domain, and consider the foreground extraction problem in a two dimensional plane, such as Cr-Cb plane or U-V plane. A hidden assumption is made that the luminance component does not affect the determination of the foreground and background area. Instead of using rectangle, more complex geometrical shapes are used in this plane to identify the foreground and background area.

Miller et al. [17] deals with the keying problem in linear fashion. This method uses a circle to define background color area, and polygon is derived to separate foreground and background regions. "Look up" tables are introduced to store color values corresponding to endpoints from U-V foreground to U-V background. The essential work is to determine whether a pixel is foreground, background or transition pixel. When a transition pixel is identified, the algorithm generates a key signal based on linear transition model. It determines how much the background are subtracted out and replaced by new background. Thus this method can computes the partial coverage of the foreground and deals with semi-transparent part of the image. It is a great improvement compared with previous keying approaches. But this method is not robust to textured image or noise image, due to the limitation of linear model.

To avoid the limitation of Miller's work, Yamamoto et al. [18] proposed an approach to generate key signal by quadratic curve groups in the U-V chroma coordinates instead of conventional straight line fashion. Mathematical representation of coordinate quadratic curve system, ellipse system, and hyperbola system are originally derived.

To sum up, rather than only using rectangle to encompass background pixel in color space as the histogram based methods do, the 2D geometric shape approaches are employed to define more complex shapes such as circles and quadratic curves in a chromatic plane. It intrinsically deals with complex color distribution cases better. However, sometimes, the assumption that the luminance component has no influence on color range determination does not hold. As Lin pointed out in [5], in some cases, luminance component may be a significant influence factor. And weighting parameters should be assigned to luminance component and chromatic component respectively, in order to place emphasis on both of them.

2.1.3 3D geometrical shape based methods

Complex color shape models ([19] [20] [21] [22] [23] [24]) are proposed to encompass foreground and background color distribution, they consider the problem in three dimensional color space instead of two chromatic plane. These methods release the researchers from holding the 2D geometrical shape based methods' hidden assumption that the luminance component does not affect the determination of foreground and background area. An advanced method is introduced that not only solves the problem of color range determination but also considers the partial coverage and shadow region.

Liu [19] proposed a method to generate a clean clipping signal for a chroma key function. Compared with previous methods which consider the image as two components (foreground, background) or three components (foreground, background, and semi-transparent), the great improvement by Liu's method is that it also considers shadow region in color image while previous methods do not.

For the user interface (UI), this system does not need the manually specified location of background, it only needs the information of a rough range of background colors by selecting one color from three principal and three complementary colors as the initial color. They are blue, green, red, cyan, yellow, and magenta, respectively. Users are able to decide which color is the best to match the real background easily, because they are perceptually different from each other a lot.

After the initial color section is set, this method calculates the optimal color range for the background and classifies background and foreground. The author builds two semi-ellipses cylindrical to define the background region. Translucent colors are encompassed in the upper part, and shadow colors are contained in the lower part. Shadow axis and translucent axis are introduced to determine the shadow and foreground object partial coverage and generate the initial alpha matte. Another important aspect of this method is to determine the parameters for the two semi-eclipses cylindrical automatically. The essential work is to calculate the edge of bounding box.

Generally speaking, color range determination methods can generate chroma keying results for video in industry field. But it cannot obtain high quality results with refined part for each frame, because it does not consider some factors such as smooth, connectivity, affinity, and confidence of the image information. On the other hand, alpha matting methods intrinsically consider these factors to derive good mattes. Thus, in order to generate better results, we then refer to the recent research topic "alpha matting". Current alpha matting algorithms are reviewed in the next section.

2.2 Alpha matting

Alpha matting is committed to generate high quality alpha matte for color image. Blue screen matting Mishima's technique [24] or Knockout is not discussed in this part. Based on a good survey paper [25], the matting algorithms is categorized to three categories, they are color sampling based approaches, affinity based approaches, and special alpha matting approaches. The category of optimization by combining sampling and affinity is removed since we consider it as affinity based methods. Another category is added to include some special methods for matting compared to the traditional methods. The color sampling based approaches are reviewed first.

2.2.1 Color sampling based approaches

As what we have discussed in Chapter 1, the matting problem is under constrained. Fortunately, the close correlation between the nearby pixels can be used to help researchers to solve the problem. Thus, priori knowledge such as trimap or scribbles of foreground and background region is required in the initialization stage. Color sampling based approaches solved this problem by sampling nearby known foreground and background pixels as the candidate set to estimate the true foreground and background colors in the unknown region. The implicit assumption is that the foreground and background color of the unknown regions is similar to the color of nearby points which are in the known foreground or background. In this case, we come to the problem of choosing points in the known regions as reference.

Ruzon and Tomasi's method

Ruzon and Tomasi [3] take a probabilistic view to the matting problem. To start with, this algorithm requires a clearly separation of foreground, background and unknown region from user.

Next, the unknown boundary region is partitioned into subregions. They construct a box for each subregion that not only includes the subregion but also contains some of the nearby foreground and background pixels, as shown in Fig. 2.1. An implicit assumption is made that the unknown region is a narrow band along the foreground boundary, and the skeleton of the unknown region can be represented by a chain of centroids in each box.

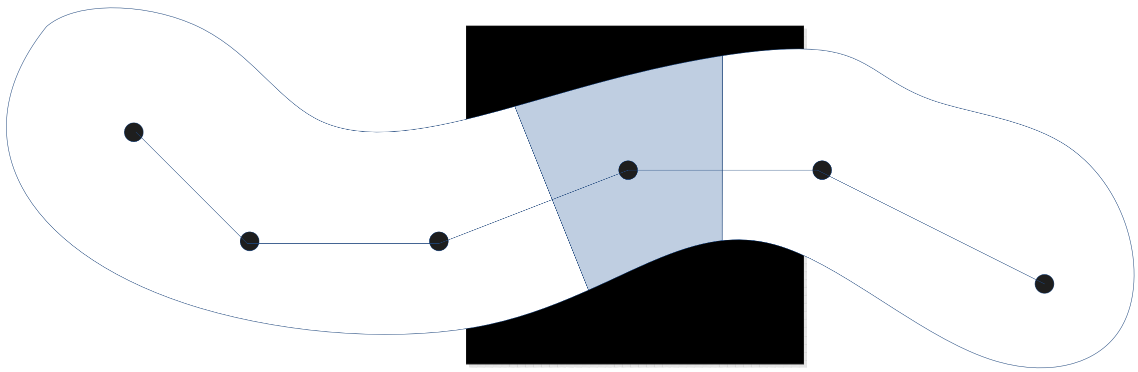


Figure 2.1: Chain of pixels.

In the third step, for each window, the encompassed foreground and background pixels are treated as samples of Gaussian distribution. The foreground pixels are separated into coherent clusters and each of them are considered as isotropic Gaussians. Manifold is constructed for each pair of these distributions. Each of the distributions is represented as a set of point mass. Foreground point mass and background point mass are paired with each other while discarding some wrong connections.

Furthermore, the authors consider the observed color as a combination from an intermediate distribution, which is a sum of Gaussians distributions. Thus, it is linearly interpolated by the foreground and background Gaussian distributions, commonly known as Gaussian mixture model (GMM). The optimal alpha value is generated when an intermediate distribution for the observed color has maximum probability.

Finally, the foreground and background colors are estimated as weighted sums of the mean of foreground and background Gaussian pairs.

This is a method to consider the problem of extracting foreground object from natural image in a probabilistic way. It requires priori knowledge of boundary location to estimate the color distributions. Two alternative schemes are used to provide the trimap in the system because there are some images where one is preferable to the other. One hidden assumption is that the unknown region should be a narrow band which can be centered from dilating a chain of pixels. However, for topology with many boundary regions, the algorithm does not work well. Also, the Gaussian model may generate large fitting errors for textured regions. Since the alpha values are computed independently for unknown pixels, and discontinuity problem occurs when the boundary region is not smoothed in the final alpha matte.

Bayesian matting

In order to improve the performance of Ruzon and Tomasi's method, Chuang [6] introduced a Bayesian approach based on Ruzon and Tomasi's algorithm. Similarly, it models foreground and background colors as mixture of Gaussians, and it makes some improvements. This method calculates the matte parameters, and solves the problem by using the maximum a posteriori (MAP) technique in a Bayesian framework. At the same time, (F, B, and α) are optimized. Then, a sliding window is constructed to encompass the neighborhood computed (F, B, and α) values for color distribution. Furthermore, the scanning direction marches inward from the exterior foreground and background region to the unknown region, as shown in Fig. 2.2.

MAP is closely related to maximum likelihood (ML). It is used to obtain a point estimation of an unobserved quantity on the basis of empirical data. Supposing that we want to estimate an unobserved value of (F, B and α) on the basis of observation of pixel C, and assume P represents the color distribution, the matting problem can be represented as follows:

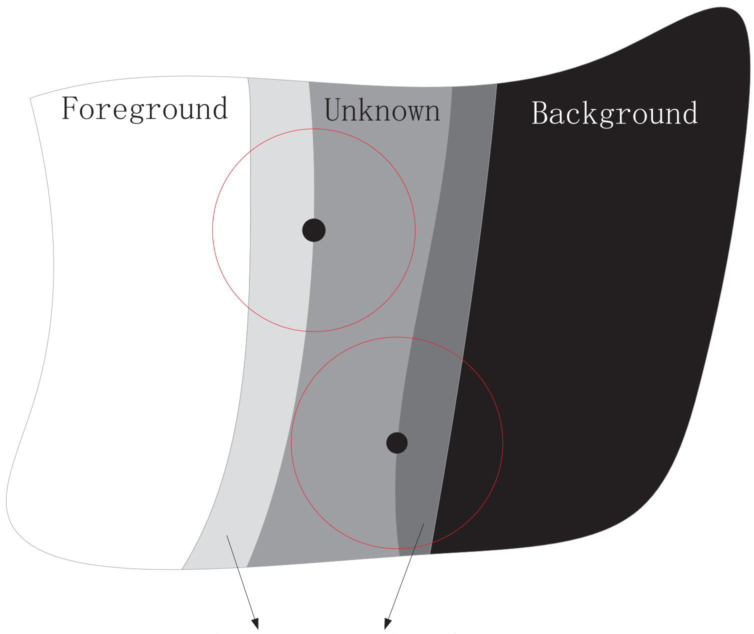


Figure 2.2: Bayesian color sampling.

$$\widehat{(F, B, \alpha)}_{MAP}(C) \quad = \quad arg \max_{F,B,\alpha} P(F, B, \alpha|C) \tag{2.3}$$

Apply Bayes' theorem to Equ. (2.3), we can obtain:

$$\begin{aligned} \widehat{(F, B, \alpha)}_{MAP}(C) &= arg \max_{F,B,\alpha} P(C|F, B, \alpha)P(F)P(B)P(\alpha)/P(C) \\ &= arg \max_{F,B,\alpha} L(C|F, B, \alpha) + L(F) + L(B) + L(\alpha) \end{aligned} \tag{2.4}$$

where $L(\cdot)$ is the log likelihood $L(\cdot) = \log P(\cdot)$, the first term is measured as the difference between the observed color and the predicted color by estimated F, B and α:

$$L(C|F, B, \alpha) = -||C - \alpha F - (1-\alpha)B||^2/\sigma_C^2 \quad (2.5)$$

To estimate the second term $L(F)$, the foreground colors are partitioned into several clusters. And oriented Gaussian distribution is modeled in each cluster, by adding the weight matrix according to the spatial Gaussian fall off. The foreground term is formulated as:

$$L(F) = -(F - \overline{F})^T \sum_F^{-1} (F - \overline{F})/2 \quad (2.6)$$

where $L(B)$ is calculated similarly by using background samples with some changes of weight matrix. $L(\alpha)$ is assumed to be constant in this approach, so it is eliminated from Equ. (2.4).

Bayesian Matting omits the opacity $L(\alpha)$ in the maximization process. An improved definition of $L(\alpha)$ is provided in Equ. (2.4) to generate better mattes. However, this method assumes the trimap is well specified. Once the condition is not satisfied, for instance, if the trimap is coarse, then the correlations between unknown pixels in foreground and background samples are weak, which tends to generate noisy matting results. Furthermore, if the image is highly textured, which means using Gaussians distribution is insufficient to model the high order statistics of color distribution, it also obtains unfavorable results.

The above two methods are both based on probability distribution. Bai [27] proposed a method that uses fast kernel density estimation [26] to generate probability density function (PDF) for foreground and background set.

Fast natural image matting

The above methods all take a probabilistic view on the matting problem, however, they do not differentiate the luminance and chrominance information. According to Lin and Shi [5], this leads to incorrect results in some cases. To avoid this problem, an effective alpha estimating method is proposed for natural image matting.

Firstly, the author builds a practical model to estimate the background and foreground color components of a given pixel in the unknown region. For each pixel in the unknown region, two circles are drawn to encompass the foreground and background sample pixels, as shown in Fig. 2.3. The estimated foreground and background colors are computed as a weighted mean over the candidate set in the circles. A spatial Gaussian fall-off parameter is used as the weight function to indicate the contribution of each candidate,

Next, the author separates chroma and intensity information of color in perceptual color space and treats them differently. Chroma alpha α_{CH} and intensity alpha α_{IN} are calculated differently as:

$$\alpha_{CH} = \frac{(I' - B')(F' - B')}{||F' - B'||^2} \tag{2.7}$$

$$\alpha_{IN} = \frac{L_I - L_B}{L_F - L_B} \tag{2.8}$$

where I', B', F' are the projected plane, and L_I, L_B, L_F are the corresponding intensity values. Then, different weight values for them are derived for both of them.

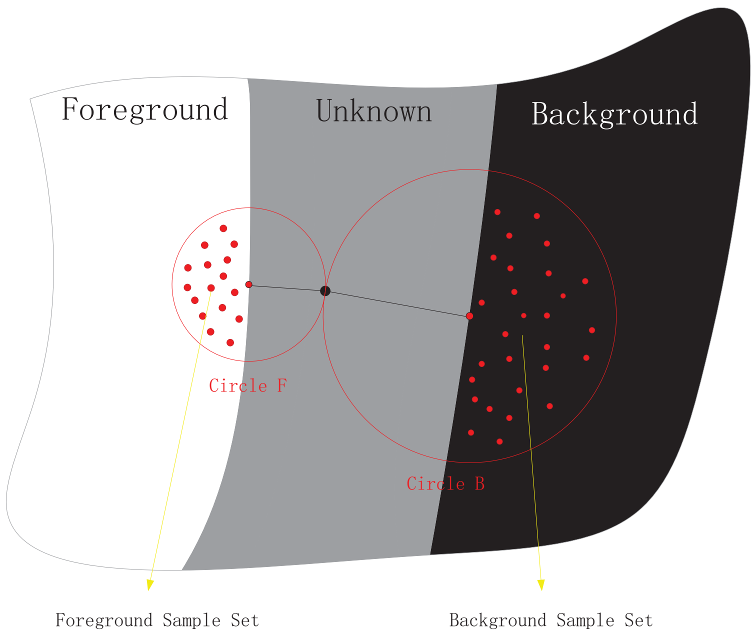


Figure 2.3: Region segmentation and foreground and background color estimation in the unknown region.

Most previous approaches focus on the second step, which build various complex models to estimate foreground and background color, while giving less attention to alpha estimating. This approach differs from previous matting methods in which it concentrates on the third step, which is alpha estimating. Color information is decomposed into chroma and intensity, and alpha is estimated as a weighted mean over chroma alpha and intensity alpha components. Because of this decomposition, significant components are emphasized and insignificant components are ignored. Also, this method solves the problem of natural image matting much faster and marginally better than Chuang's [6] approach.

However, this method faces a problem. As shown in Fig. 2.3. We can see that the radius of the circle is determined by the distance from the pixel in the unknown region to the nearest known foreground or background pixel. If the color of pixels near the boundary of foreground or background changes dramatically, the alpha matting results in the unknown region are not smoothed. Sometimes the results appear like a segmentation line in the matte results. Also, this technology adjusts several parameters, and it does not specify how to adjust the parameters to get a better result, so this method is not favored by general users. Furthermore, the results depend on the accuracy of the trimap a lot. Sometimes, minor change of trimap can lead to big change of matting result. In the experiment, we can try several times to generate the best result. However, in the industry field, trimap based approaches have a lot of limitations. A method does not depend on the accuracy of trimap or it can automatically optimize the trimap is needed. To develop a system to solve this problem contributing to the motivation of our research work.

There are some limitations for color sampling methods. They assume that the image is locally smooth around the boundary region, but for textured image this assumption may be violated. Also these methods compute the alpha values for each individual pixel, which generate image noise.

2.2.2 Affinity based matting approaches

To avoid the limitation of color sampling methods, affinity based matting approaches emerge. They are focused on using different affinity terms between neighboring pixels and they model the matte gradient across the whole image instead of directly compute the alpha value for each individual pixel.

Poisson matting

A typical example is Poisson matting [28], it models the matte gradient across the whole image to avoid matte discontinuities. Poisson matting requires manually specified trimap as user input and models the matte gradient by assuming that intensity changes in the foreground and background are locally smooth. Partial derivatives are applied to both sides of matting equation Equ. (1.1) as:

$$\nabla I_z = (F_z - B_z)\nabla\alpha_z + \alpha_z \nabla F_z + (1-\alpha_z)\nabla B_z \tag{2.9}$$

where ∇ represents the gradient operator, and $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$. Because foreground F_z and background B_z are locally smooth, the two terms $\alpha_z \nabla F_z$ and $(1-\alpha_z)\nabla B_z$ are relatively small with respect to $(F_z - B_z)\nabla\alpha_z$. Then the approximate matte gradient is formulated as:

$$\nabla\alpha_z \approx \frac{1}{F_z - B_z}\nabla I_z \tag{2.10}$$

The final alpha matte is solved by minimizing the variation problem:

$$\alpha^* = arg\min_{\alpha} \int\int_{p\in\Omega} ||\nabla\alpha_p - \frac{1}{F_p - B_p}\nabla I_p||^2 dp \tag{2.11}$$

where Ω is the unknown region in the trimap.

For textured image, the assumption of local smoothness cannot be guaranteed, thus selecting the closet samples to estimate foreground and background color is not accurate. In order to resolve this problem, the user is allowed to refine the matting result by using boosting brush, highpass filtering, diffusion filtering, and clone brush. But it is a time consuming work to generate a good matte in this way.

Closed form matting

When the estimation of foreground and background colors is not accurate, the result of Poisson matting [28] deteriorates a lot. Closed form matting [14] avoids the limitation by introducing a cost function from local smoothness assumptions on foreground and background colors, and shows that in the resulting expression. It is possible to analytically eliminate the foreground and background colors to obtain a quadratic cost function in alpha. Closed form matting only requires a few user specified strokes as the input, it is a time-saving work. The proposed cost function is defined as:

$$J(\alpha, a, b) \quad = \quad \sum_{j\in I}(\sum_{i\in w_j}(\alpha_i - \sum_c a_j^c I_i^c - b_j)^2 + \epsilon \sum_c a_j^{c^2}) \tag{2.12}$$

a and b can be eliminated from the above cost equation and yields:

$$J(\alpha) \quad = \quad \alpha^T L \alpha \tag{2.13}$$

where L is an $N \times N$ matrix, whose (i, j) -th entry is:

$$\sum_{k|(i,j)\in w_k} (\delta_{ij} - \frac{1}{|w_k|}(1 + \frac{1}{\frac{\epsilon}{|w_k|} + \sigma_k^2}(I_i - \mu_k)(I_j - \mu_k))) \tag{2.14}$$

Finally, the optimum alpha matte is solved by computing:

$$\alpha \quad = \quad arg\min \alpha^T L \alpha \tag{2.15}$$

Closed form matting approach derives a novel affinity function and optimizes the global cost function to generate optimum alpha matte. This algorithm is better than others because a novel quadratic cost function is derived. However, when the assumption that local smooth does not hold, the results deteriorate a lot, along with some image noise. This problem can be solved by applying a smoothing operation to optimize the initial alpha matte.

Robust Matting [4] is such an approach that uses random walk to do this operation. It mentions that there are two failure models in previous matting methods. In the two failure models, unrelated pixel samples are used to estimate the foreground color and background color in the unknown region, which introduces some errors. Also, the quality of alpha matting results depends on the accuracy of the trimap a lot. Thus, the matting results are not stable, and different trimaps correspond to different matting results. In general, the more accuracy the trimap is, the better matting results generate. A confidence evaluation method is proposed to make sure good samples are used for color estimation. Sample pairs with low confidence values will not be chosen, only good samples are used to do the estimation process. Also an optimization method named modified random walk is adopted to get better mattes. Furthermore, an iterative optimization method is used in the system. The trimap is optimized during each iteration process, and makes the final results less depended on the accuracy of the original trimap. In other words, the result is more stable and reliable compared with other methods. Since we introduce the idea in our proposed scheme, robust matting will be further discussed in Chapter 3.

To reduce the computation, shared sampling [29] is invented. It is based on the key observation that pixels in a small neighborhood tend to have highly similar values for (a, F, B) triplets. A new objective function for identifying good pairs of background and foreground colors for any given pixel p is proposed. The new function takes into account spatial, photometric, and probabilistic information extracted from the image.

Some methods use probabilistic approach in affinity based matting method, and extensive independence assumptions are required for traditional Markov random fields (MRF) for image segmenting. This limitation has been removed by the new model of conditional random fields (CRF) [30]. Wang [31] applies this new method to affinity based alpha matting and generates good results.

New affinity based matting approaches ([32] [33]) propose fuzzy connectedness (FC) [34] to compute the distance from the unknown pixel to known one. They capture the adjacency and similarity information between two pixels. They are good at dealing with images that have long skinny fuzzy structures.

Other affinity based matting methods ([27] [35] [36]) also use the idea of affinity, and solve the alpha matting problem by minimizing an energy function. Each of them holds some good features. To sum up, affinity based methods deal with textured image better, because the affinities are defined in local window. The disadvantage is that as the alpha matte is estimated in a propagation way, small errors could be propagated and accumulated to big errors.

2.2.3 Special matting methods

Rather than using traditional methods for alpha matting, special matting methods have been proposed. Some of them require advanced devices such as 3D camera, or special initialization to generate additional information, which make the problem easier to solve. With the help of these devices, these methods can do this work better in some aspects. The approaches which can divide the image into multi-layers will be discussed in the following.

Depth Keying

"Depth Keying" [37] uses a novel 3D depth video camera [38]. It is capable to obtain the complete data of the field of view (FOV), which contains both the color data (RGB) and distance data (D) for each pixel. By introducing the 3D camera, depth keying could generates matte in natural background in real time, without user input such as trimap or foreground and background scribbles. Furthermore, instead of segmenting image into only two layers (foreground and background) as traditional matting methods, depth keying generates multi-layered mattes. For each layer with different depth, the basic processing steps of depth keying system are:

1. Generate a depth mask by thresholding the depth map.

2. Foreground objects contour recovery.

3. Define the region to be processed.

4. Color distances between neighboring pixels are calculated. For each pixel $p \in R$ and for each pixel q of its 3×3 neighborhood, a weight is defined as $W_i = \frac{1}{\sqrt[4]{D_i}}$, where D_i is the distance between p and q.

5. For each pixel $p \in R$, the new value of alpha is corresponded to the pixel p as: $\alpha_p = \sum_{q \in N_p} W_I \alpha_q$.

6. Repeat step 5 several times (no more than three), till the mask distribution is similar to the color transition.

However, the depth keying is a hard segmentation algorithm, which means that the alpha matte of the unknown region is not calculated. The results can be improved by dilating the boundary between the foreground and background, applying the alpha matting algorithms (Robust Matting [4], Bayesian matting [6], Poisson matting [28], etc.). Moreover, after integrating the depth keying and 2D chroma keying, the keying system can be done in an automatic way. It generates the multi-layer matte from near to far.

Depth matting approach [39] also improves the results of natural matting algorithms by adding a depth channel. The additional depth information allows the authors to reduce the artifacts that arise from ambiguities. They occur when an object is a similar color to its background. Trimap is generated automatically in three steps:

1. Up-sampling. Assume one picture taken by a digital camera is with high resolution, and the picture taken by a depth camera has a low resolution, the author up-samples the low resolution depth map picture to the high resolution first. Many up-sampling methods such as bi-linear interpolation or nearest neighbor will create depth edges that cross boundaries. The author uses a super resolution method presented by Yang.

2. Thresholding. To do the segmentation, the method requires the user to provide a dividing plane that defines which object lies in the foreground and which object lies in the background.

3. Dilating. Because the two color image are not precise on the edges, which do not take into account the partial alpha values. The authors need to determine the unknown, foreground and background region around the object. To do this, they erode and dilate the foreground. Anything that is modified in either of these steps is considered as unknown region. The exact amount of erosion and dilation is specified by the user and is depended on the "fuzziness" of the object in the foreground. Using these trimaps, they generate an alpha mask for each frame in the video.

Notice that the 3DV Systems's ZCam does not compute partial foreground values and therefore has noticeable artifacts for objects with fuzzy borders such as hair. Traditional alpha matting approaches can be applied to refine the result.

Depth of field matting

Another interesting invention [40] requires input of two images–one where the object is in focus, and one where the background is in focus. Then, this invention automatically produces an alpha matte (binary) indicating which pixels belong to the object. The processing steps are:

1. Image capture. High dynamic range (HDR) image is used to improve the precision rather than less accurate quantized data. Linear and un-quantized images are generated from multiple exposures.

2. Non-linear response compression. Due to the disadvantage of applying all subsequent processing in the logarithmic domain, which introduces arbitrary negative numbers, each image is scaled by its log average luminance.

3. Center-surround analysis. This process is to determine whether a region of pixels is in focus or not. Gaussian blurred function is used to generate local variability, which is further employed to detect the foreground and background pixels.

4. Filling in. Integrating the edge signal into a reconstructed representation of the scene, which is a under constrained process. The authors resort to a diffusion based approach to compute more accurate alpha matte.

5. Clean up. The image is thresholded to create a binary alpha matte (hard segmentation).

It is the first method to use foreground and background in focus images to generate binary alpha matte. A hidden limitation is that two specific HDR images are needed, and tripod should be set in the environment, which is not favorable for many applications. And also it only generates binary alpha matte. More accurate alpha matte may be generated by combining it with other alpha matting methods.

Color filtered aperture

Bando et al. [41] presented a method for automatically extracting a scene depth map and an alpha matte of a foreground object by capturing a scene through RGB color filters placed in the camera lens aperture.

A captured image has depth-dependent color misalignment. The authors proposed a color alignment measure to estimate disparities between the RGB planes for depth reconstruction. They also exploit color misalignment cues in their matting algorithm in order to disambiguate between the foreground and background regions even where their colors are similar.

By dividing the aperture into three regions, where only light in one of the RGB color bands can pass, they acquire three shifted views of a scene in the RGB planes of a captured image in a single exposure.

However, the matting fails when the foreground and background colors are similar with little texture, since they have few color misalignment cues. Also, they use a relatively large window, which means that they cannot recover small features such as hair strands and holes in foreground objects. Once they are missed in the course of optimization. It is also not favorable for general matting approaches because color filtered aperture is not widely used.

2.3 Motivation of our research work

We looked into the detailed patents of several chroma keying for color range determination systems, and found that the chorma keying results can be further improved, especially for some refined parts of image, such as the hair of human. After reviewing the alpha matting methods, we found that robust matting [4] did this work well, it not only considered the confidence of the selected sample pairs, but also optimized the initial alpha mattes by applying modified random walk which guarantees that the method is robust to image noise. The existing problem and discovery motivate us to use advanced alpha matting methods to improve the result of chroma keying for color range. By combining the two kinds of approaches together, our method generates the alpha matte without user specified trimap or scribbles, and it is good at handling multiple background and gridded color image.

Chapter 3

The proposed matting scheme for color images

The proposed matting scheme contains two parts: the color range determination and the robust matting [4].

3.1 Color range determination for chroma keying

Color range determination contains the following five steps: background color initialization, K-means clustering, background color removal, optimization and trimap generation.

3.1.1 Background color initialization

YCbCr is a family of color spaces used in digital photography systems. There are three components in YCbCr space: Y is the luminance component which does not contain chroma information. Cb is the blue difference chroma component and Cr is the red difference chroma component. In the trimap generation step, the luminance component Y is disregarded. We only deal with the chromatic Cb and Cr components, or Cb-Cr plane. For example, the Cb-Cr plane at constant luminance Y=0.5 is shown in Fig. 3.1.

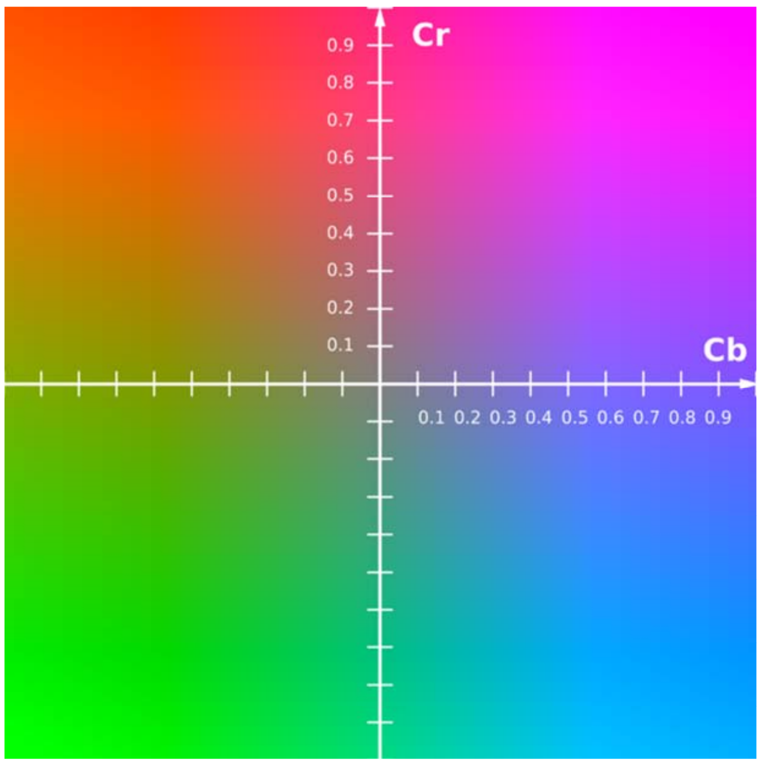


Figure 3.1: The CbCr plane at constant luma Y=0.5 [42]. Cb is on the horizontal x axis going from -1 to 1, and Cr is on the y axis going from -1 to 1 as well.

To simplify the operation for users, instead of specifying the values of Cb and Cr for background, we follow Liu's approach [19] to divide the chromatic plane into six sectors. Each sector has its own centroid color (blue, cyan, green, yellow, red and magenta) and a central angle of 60 degree, which is shown in Fig. 3.2. User can pick one of the six colors as an initial background color, which roughly matches the real background. The centroid of the pre-defined sector is the initial value, which can be considered as the rough background color. This method is also compatible with switcher panel in video

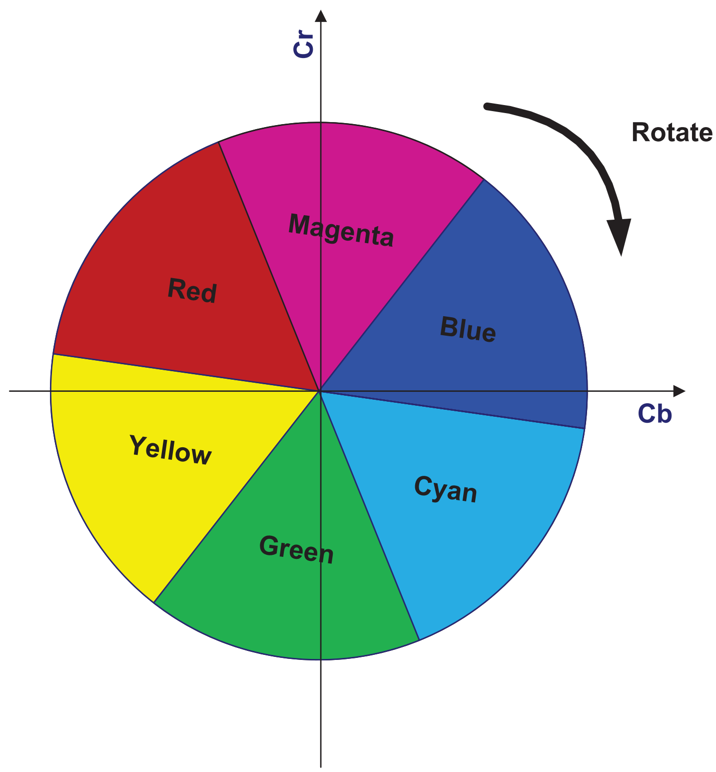


Figure 3.2: Initial color sectors [19].

industry.

In order to process easier, we have to make the background color sector symmetric to the Cb axis. Assume that the user picks the blue color sector. Then the coordinate system is rotated clockwise to make the blue color centroid falls on the Cb axis, as shown in Fig. 3.3

3.1.2 K-means clustering

The initial color sector cannot precisely represent the background color. In our multi background application, there are several main background colors, which can be viewed as several color groups. In this case, a method is needed to partition these groups for

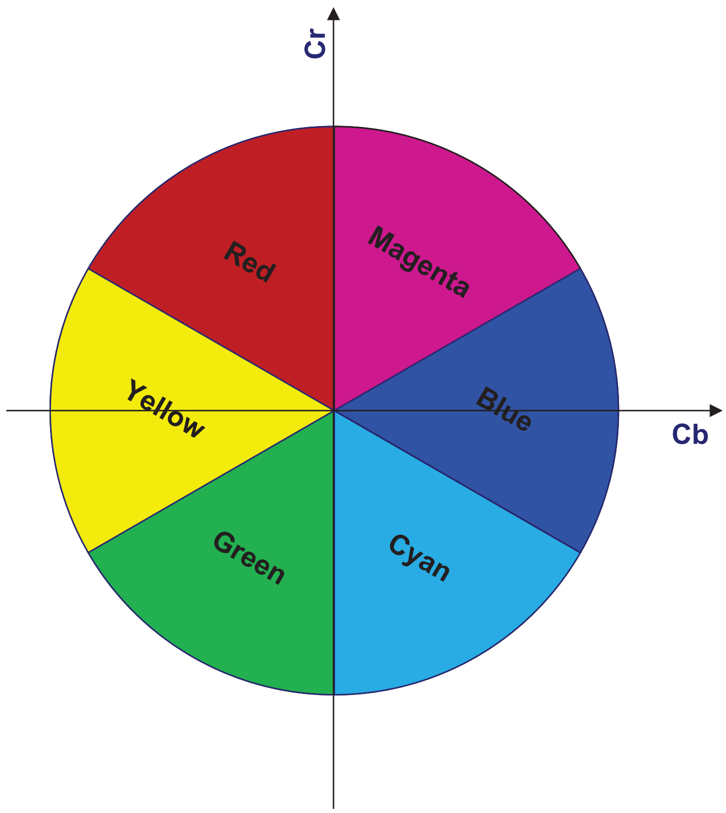


Figure 3.3: Rotated color sectors [19].

further processing.

K-means [43] is a cluster analysis method which aims to partition n observations into k clusters so that the observations in the same clusters have the shortest distance to their mean values. Mathematically, suppose $(x_1, x_2, ..., x_n)$ is an observation set, where each observation is a d-dimensional real vector, the objective of K-means is to partition the n observations into K sets $(K \leq n)S = (S_1, S_2, ..., S_k)$, to minimize the sum of squares in the cluster:

$$arg\min_S \sum_i = 1^k \sum_{x_j \in S_i} ||x_j - \mu_i||^2 \tag{3.1}$$

where μ_i is the mean value of points in S_i.

To solve the K-means problem, a common iterative refinement algorithm is applied: assume the initial set of K-means is $m_1^{(1)}, m_2^{(1)} ..., m_k^{(1)}$, the algorithm proceeds assignment step and update step iteratively:

1. Assignment step: each observation is assigned to the cluster with the closed mean as:

$$S_i^t = x_j : ||x_j - m_i^{(t)}|| \leq ||x_j - m_{i^*}^{(t)}|| forall i^* = 1, ..., k \tag{3.2}$$

2. Update step: the new means are calculated to be the centroid of the observations in the cluster:

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \tag{3.3}$$

After the assignments no longer change, the algorithm converged.

In our system, we use K-means to cluster all the pixels falling in the initial rough color sector. The background pixels are gathered around these centroids for our test images. The default value for k is 4, and it can be adjusted according to different characteristics of background image.

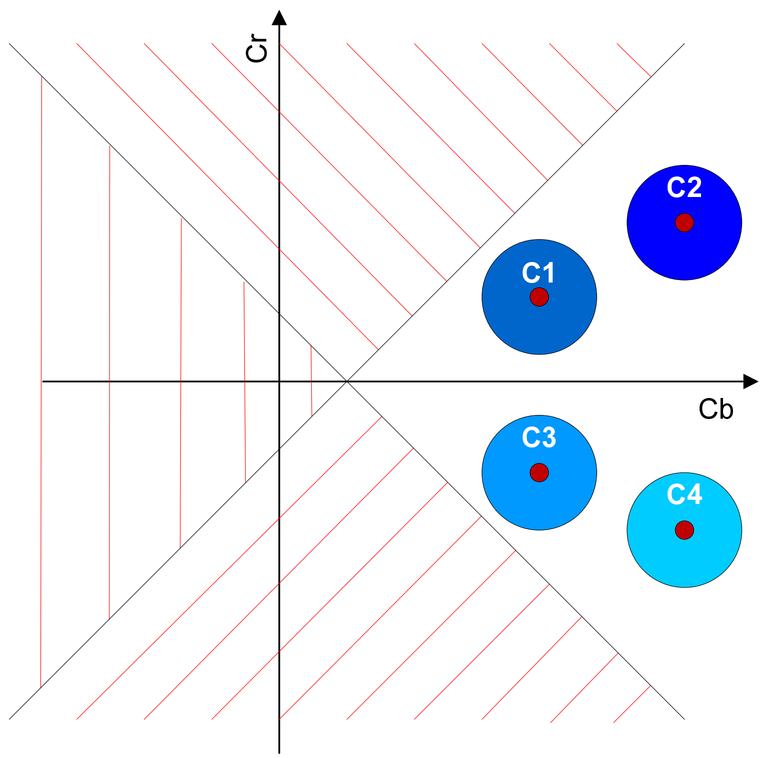


Figure 3.4: K-means centroids.

3.1.3 Background color removal

Considering each centroid of clusters as the center of circle, we draw the circles around them, as shown in Fig. 3.4. Any color of pixels that fall into these circles can be considered as background. The shadow region represents rough foreground.

3.1.4 Optimization

Till now, we can identify the background region in the image. However, the result is not good enough. Some colors not included in the centroid circles but very close to these circles are also background colors. A window searching process is used to optimize the result. We compute the smallest distance to the four centroids for each pixel in the window, and add them together as sum1. Suppose W indicates the window, C

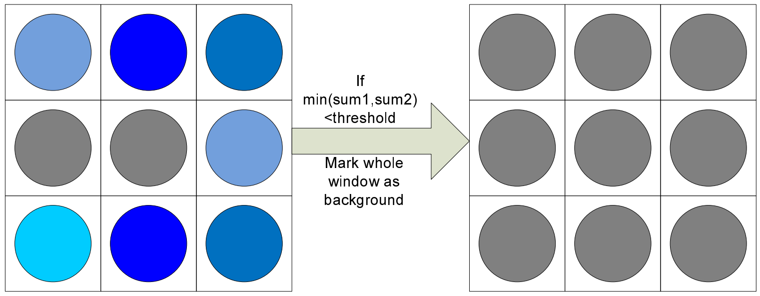


Figure 3.5: K-means centroids optimization.

represents centroids, then for a point sum1 is formulated as:

$$sum1 = \sum_{w_i \in W, c_j \in C} D_{\min}(w_i, c_j) \tag{3.4}$$

In addition, the sum of squares of each pixels' Cb, Cr value distance to the centroids in the window are also calculated, and they are added as sum2:

$$sum2 = \sum_{w_i \in W, c_j \in C} D^2(w_i, c_j) \tag{3.5}$$

If $\min(sum1, sum2) < threshold$, the whole window is marked as background, as shown in Fig 3.5.

3.1.5 Trimap generation

This process contains two steps: unknown region detection and known region expansion.

1. Unknown region detection. An image map is used to record each pixel of the image, depending on whether it is foreground pixel or background pixel. Then a flexible size window is used to detect the whole image. Any region that covers only foreground

pixels is marked as foreground region. And any region that covers only background pixels is marked as background region. Region that covers both foreground and background pixels is marked as unknown region, as shown in Fig. 3.6

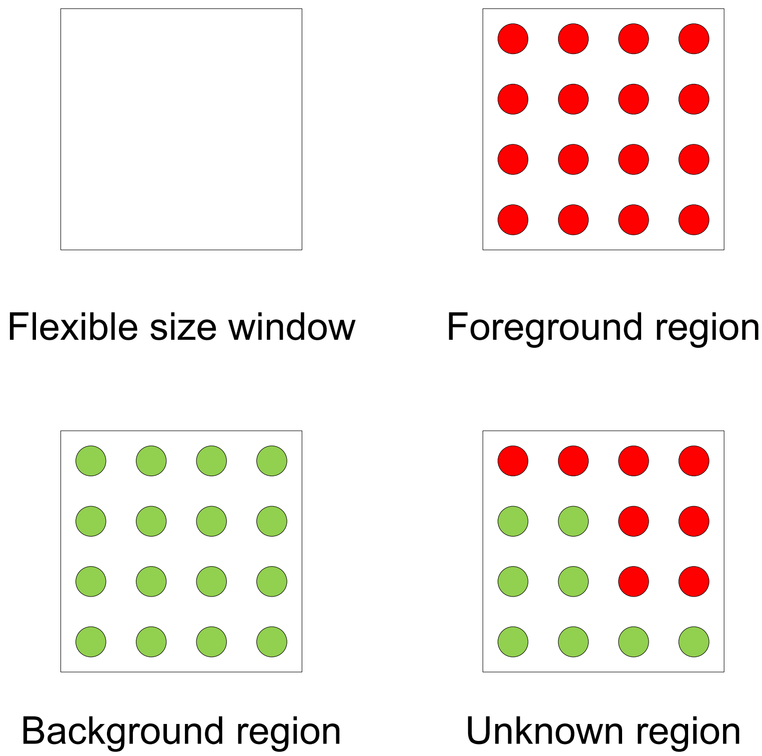


Figure 3.6: Unknown region detection.

2. Known region expansion. This process is derived from Gastal's paper [29]. The objective is to reduce the size of unknown regions by exploiting the affinity of neighboring pixels. Suppose the image-space and color-space distances between two pixels p and q are $D_{image}(p, q)$ and $D_{color}(p, q)$. For each pixel p in the unknown region, it is checked that if there is a pixel q in the known region that $D_{image}(p, q) \leq t_i$, $D_{color}(p, q) \leq t_c$ and $D_{image}(p, q)$ is the minimal for p. If so, the pixel p is labeled as known pixel. If q is in foreground, then p is also in foreground. Otherwise, p is labeled as known background pixel. The threshold t_i equals 10, and t_c is set to be

5/256 according to experimental results.

Till now, definitely foreground, definitely background and unknown regions are known. In other words, the trimap is generated.

3.2 Robust Matting

Robust Matting is proposed by Wang [4], it is able to generate high quality alpha matte. Confidence evaluation is derived to choose good potential samples, and the modified random walk algorithm is used to smooth the alpha matting result. Due to these good characteristics, we use it to optimize our chroma keying result. Since robust matting is commercially preserved, we do not have access to the source code. In order to combine the two approaches together, lots of work is focused on studying and implementing robust matting system. The system works as follows:

3.2.1 Candidate points selection

In order to predict the alpha value in the unknown region, the known foreground and background colors are extrapolated into the unknown region. The following problem is how to choose the foreground and background candidate samples. We have reviewed that Lin [5] used a method to find the nearest point in the foreground and background region, and drawn circles to contain the candidate points. Bayesian [6] and Belief Propagation [31] also choose the samples that have the shortest distances to the target pixel as samples.

All of the above algorithms share a same problem, sometimes they cannot choose good candidate points for color estimation as mentioned by Wang in [4]. In order to fully span the variation in foreground and background colors, Wang chooses candidates along the boundary of the foreground and background, as shown in Fig. 3.7. This method is able to choose samples that span more color variations, and it leads to a better potential candidate set.

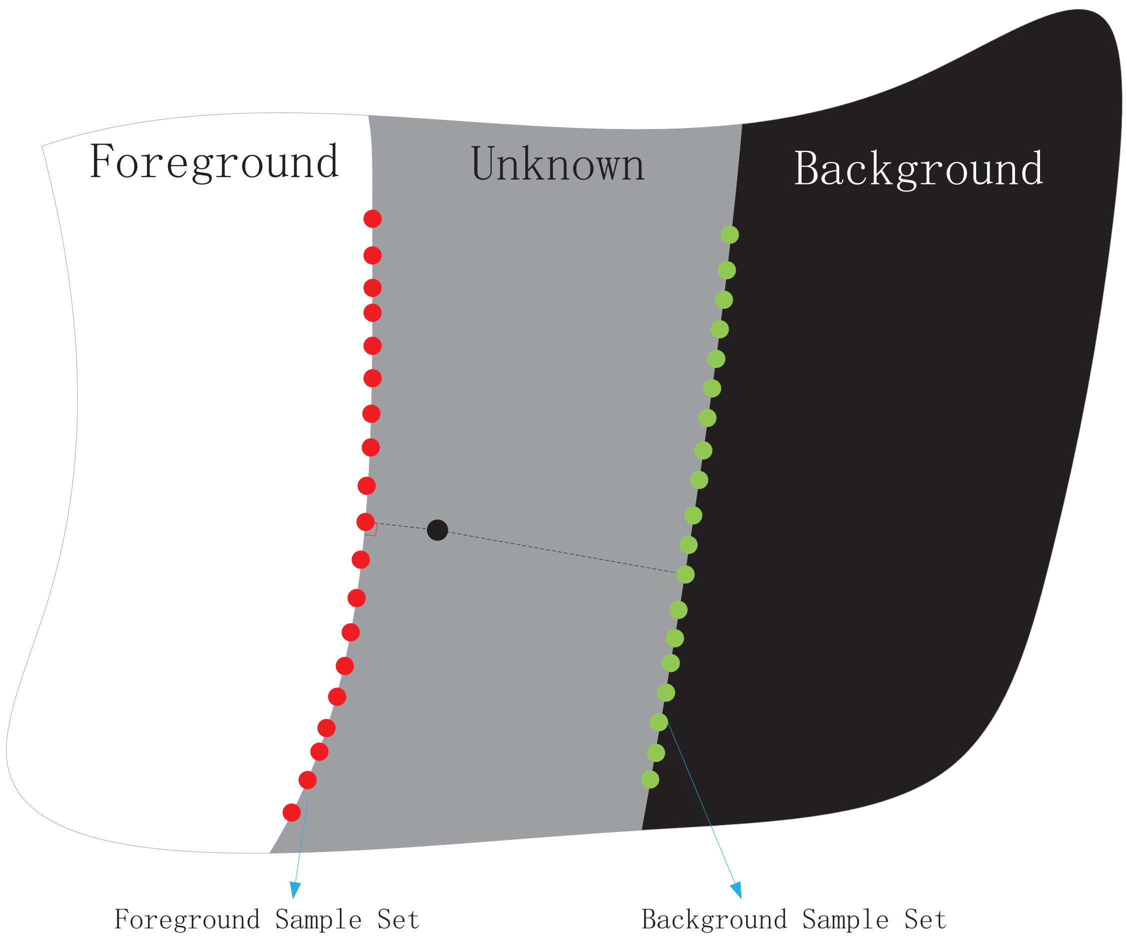


Figure 3.7: Color sampling methods of Wang [4].

The initial alpha value is computed between each pair of foreground and background sample pairs as:

$$\hat{\alpha} = \frac{(I_z - B_z)(F_z - B_z)}{||F_z - B_z||^2} \tag{3.6}$$

3.2.2 Confidence evaluation

Confidence analysis guarantees the color samples chosen by the system has a high confidence value. It is essential to choose good samples from a large candidate set. Good sample pairs should explain mixed foreground and background pixels as linear combinations of the samples. There are two failure models for previous matting approaches, which choose bad samples as candidate set.

One failure model of previous matting approaches is shown in Fig. 3.8. If the line passes through or near the color of the pixel under consideration, it explains the pixel's color as a linear combination more accurately.

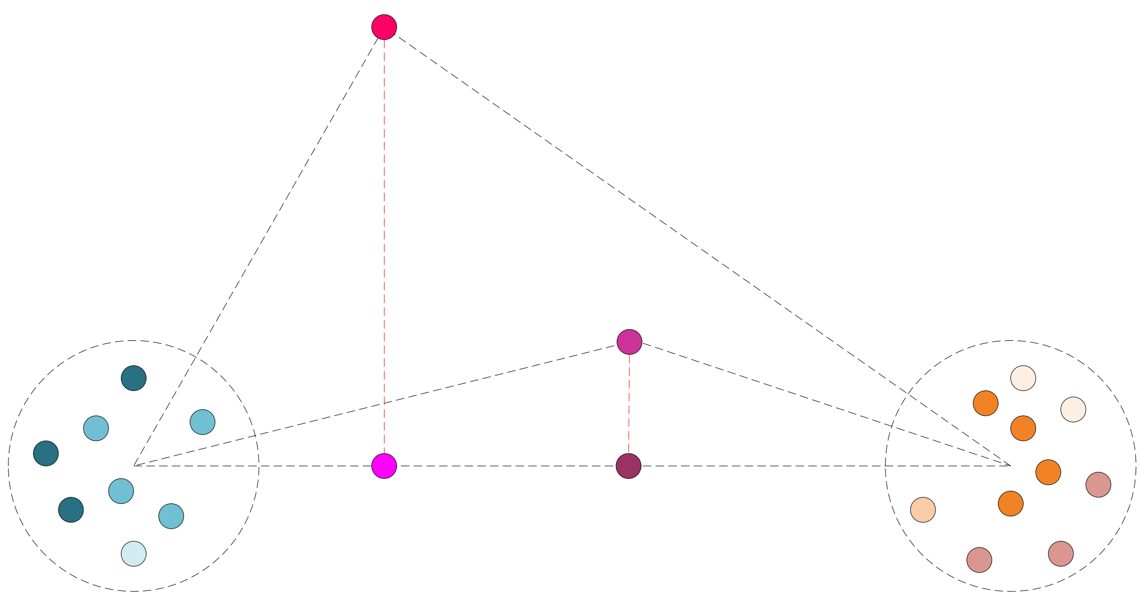


Figure 3.8: Failure model for linear blending of samples.

To avoid this problem, distance ratio is derived to evaluate the sample pairs by examining the ratio of the distances between the pixel color, and the color it would have predicted by the linear model. It is defined as:

$$R_d(F^i, B^j) = \frac{||C - (\hat{\alpha}F^i + (1-\hat{\alpha})B^j)||}{||F^i - B^j||} \tag{3.7}$$

For example, the distance ratio will be much higher for PB than PA. However, the distance ratio may not define the best sampling. In this case, the proximity of samples should also be taken into account, which is another common failure model. For example, as shown in Fig. 3.9, for pixel C, because of proximity in color space, F2 is a better foreground sample even though F1 creates a better fit for the linear blend model.

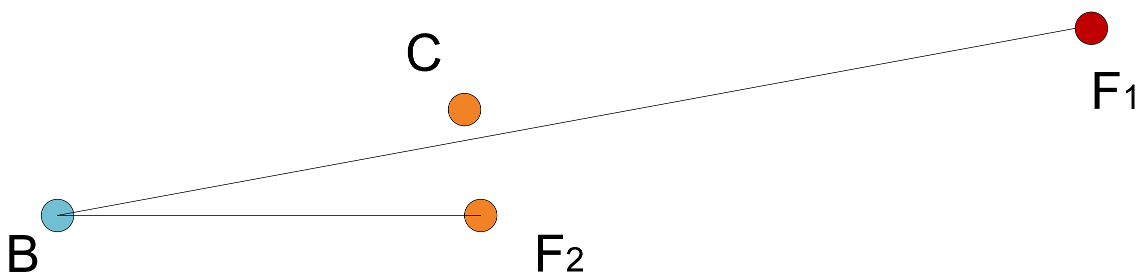


Figure 3.9: Another failure model for matting methods.

Sample weights are defined to solve the proximity problem. Foreground sample weight is defined as:

$$w(F^i) = exp\{-||F^i - C||^2/D_F^2\} \tag{3.8}$$

The sample weight for background is defined as:

$$w(B^i) = exp\{-||B^i - C||^2/D_B^2\} \tag{3.9}$$

where D_F and D_B are the minimum distances between foreground, background samples and the current pixel.

Combining all these factors, the confidence value for a sample pair is defined as:

$$F(F^i, B^j) \quad = \quad exp\{-\frac{R_d(F^i, B^j)^2 \cdot w(F^i) \cdot w(B^j)}{\sigma^2}\} \tag{3.10}$$

where σ is fixed to be 0.1 in the system.

The system chooses 20 foreground and 20 background points as the candidate set. A small number of pairs (3 in the system) with the highest confidence values are selected to evaluate the final alpha value. The result of confidence evaluation is shown in Fig. 3.10.

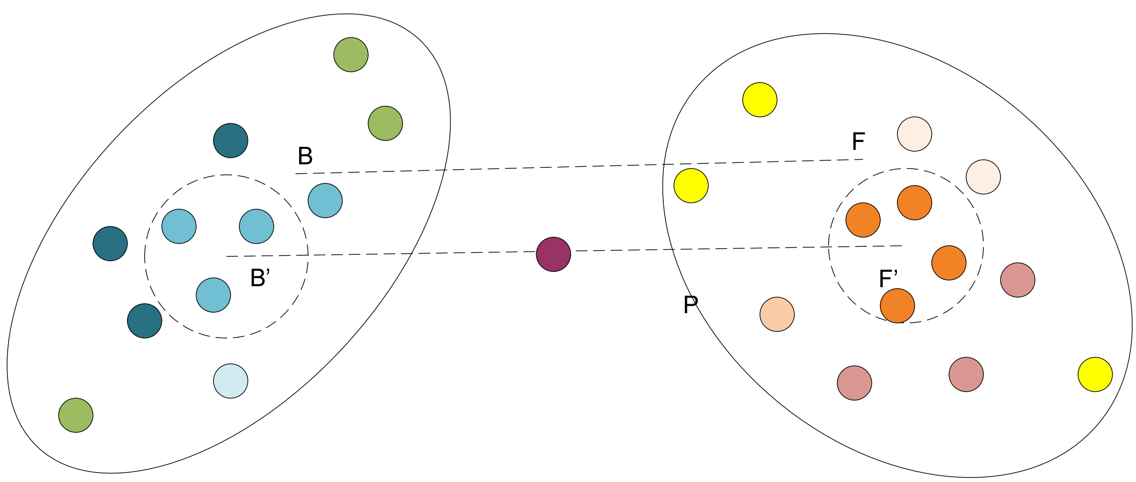


Figure 3.10: A particular subset of samples is valid for alpha value estimation. Only good samples are used.

3.2.3 Random walk and modified random walk

Random walk is defined as follows: given a small number of pixels with user-defined labels, one can determine the probability that a random walker starting at each unlabeled pixel will first reach one of the pre-labeled pixels. Suppose G is a connected graph, for a vertex $x \in K$, and $m(x)$ denotes the number of edges emitting from vertex x. If x and y are connected by an edge, then the probability that a particle moves from x to another vertex $y \in K$ is $1/m(x)$, otherwise it is zero. According to [44], random walk in a square lattice is closely related to the difference analog of the Laplacian:

$$\triangle f(x) = \sum_{y \sim x} f(y) - m(x) f(x) \tag{3.11}$$

where $y \sim x$ indicates x and y are adjacent nodes. Dirichlet integral is defined as:

$$D[u] = \frac{1}{2} \int_{\Omega} |\nabla u|^2 d\Omega \tag{3.12}$$

According to [45], to solve the problem of random walk equals to solve the combinatorial Dirichlet problem, which is to find a harmonic function subject to its boundary values.

To solve this problem, a matrix for the graph is constructed as:

$$L_{ij} = \begin{cases} d_i : & \text{if } i = j, \\ -W_{ij} : & \text{if } i \text{ and } j \text{ are neighbors}, \\ 0 : & \text{otherwise.} \end{cases} \tag{3.13}$$

where $d_i = \sum_j W_{ij}$.

Suppose the harmonic function is x, then the Dirichlet integral can be represented as:

$$D[x] \quad = \quad \frac{1}{2}x^T Lx \tag{3.14}$$

the harmonic function x guarantees the value of D[x] is minimized.

Furthermore, the vertices are separated into two sets to simplify the problem. Assume V_M represents seed nodes and V_U indicates unseeded nodes. Obviously, they satisfy $V_M \bigcap V_U = \emptyset$ and $V_M \bigcup V_U = V$. Suppose the vertices in L and x are ordered and seed vertices appear before unseed vertices. So Equ. (3.14) could be decomposed into:

$$\begin{aligned} D[x_U] &= \frac{1}{2}\begin{pmatrix} x_M^T & x_U^T \end{pmatrix}\begin{pmatrix} L_M & B \\ B^T & L_U \end{pmatrix}\begin{pmatrix} X_M \\ x_U \end{pmatrix} \\ &= \frac{1}{2}(x_M^T L_M x_M + 2x_U^T B^T x_M + x_U^T L_U x_U) \end{aligned} \tag{3.15}$$

where x_B is the potential of the seeded vertices, and x_U corresponds to the potential of unseeded vertices. Differentiating $D[x_U]$ with respect to X_U yields:

$$L_U x_U \quad = \quad -B^T x_M \tag{3.16}$$

where x_U is the unknown potential/alpha we wish to solve. Then L is decomposed into blocks corresponding to unknown nodes P_u, and known nodes P_k including user labeled pixels and virtual nodes as:

$$L = \begin{pmatrix} L_k & R \\ R^T & L_u \end{pmatrix} \tag{3.17}$$

Finally, to calculate the probabilities of unknown pixels belonging to a certain label is the solution to:

$$L_u A_u \quad = \quad -R^T A_k \tag{3.18}$$

where A_u is the vector of unknown alphas we wish to solve for, and A_k is the vector that encodes the boundary conditions.

In order to optimize the initial alpha matte and make it robust to image noise, the classic random walk algorithm is modified as shown in Fig. 3.11. In classic random walk, the seed points are chosen from the image pixels. If we specify two seeds, then one can be viewed as foreground point, and the other is regarded as background point. Since we already have the initial alpha matte, and we know the foreground and background region, so the seeds cannot directly chosen from the image pixels.

Modified random walk defines two virtual points. One represents the virtual foreground, while the other stands for virtual background. Both of them connect to all the pixels in the image. All the weights of the edges need to be specified before the calculation of Dirichlet problem. There are two kinds of weights in Section 3.11. The first one is data weight, which considers the alpha values for each individual pixel especially when the confidence value is high. Data weight for virtual foreground is defined as:

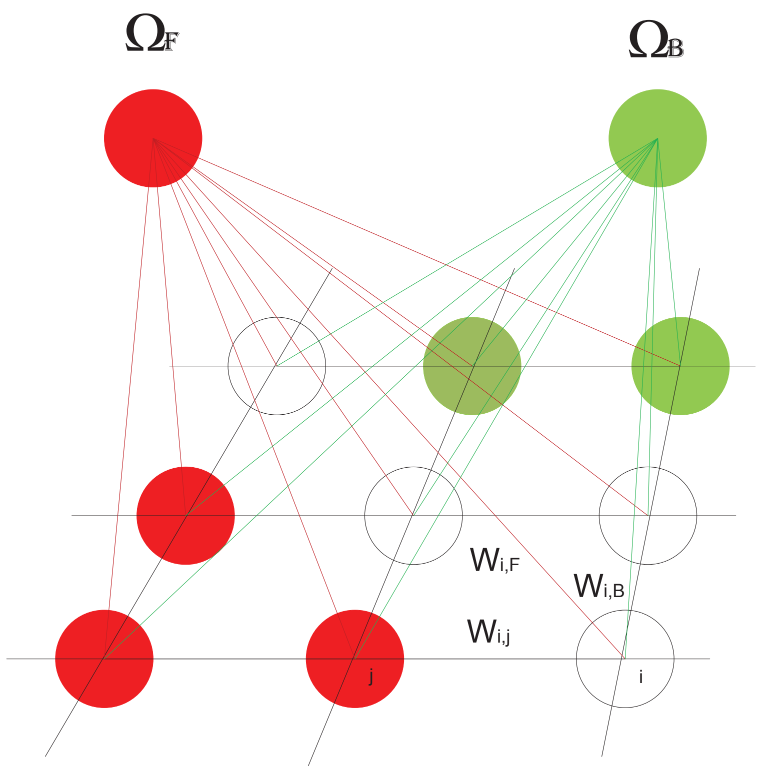


Figure 3.11: Matte estimation by using modified random walk.

$$W(i, F) = \gamma \cdot [\hat{f}_i \hat{\alpha}_i + (1 - \hat{f}_i)\delta(\hat{\alpha}_i > 0.5)] \tag{3.19}$$

Data weight for virtual background is defined as:

$$W(i, B) = \gamma \cdot [\hat{f}_i (1 - \hat{\alpha}_i) + (1 - \hat{f}_i)\delta(\hat{\alpha}_i < 0.5)] \tag{3.20}$$

where $\gamma = 0.1$.

The second kind of weight is edge weight. It considers the neighborhood constraint.

According to [14], edge weight or neighborhood term is defined as:

$$W_{ij} = \sum_{k}^{(i,j)\in w_k} \frac{1}{9}(1 + (C_i - \mu_k)(\sum_k + \frac{\epsilon}{9} I)^{-1}(C_j - \mu_k)) \tag{3.21}$$

3.3 Matte optimization

After defining these weights for the random walk problem, the system can solve the Dirichlet boundary problem and generate the potentials or alpha values for the unseeded points. Finally, the iterative refinement is used, the color sampling and random walk run iteratively. After obtaining the random walk result, alpha values above 0.98 are declared to be known foreground and clamped to 1.0; values below 0.02 are set to known background and clamped to 0. This iteration process helps to get a more accurate trimap after each iteration process, and provides new color samples for mixed pixels in the next iteration. Thus the final result will be more and more accurate.

3.4 Summary

Our proposed scheme is able to extract foreground object from constant or even gridded color image, by exploiting the advantage of K-means clustering. Trimap is also generated automatically by employing unknown region detection and known region expansion. Due to the good feature of confidence evaluation and random walk optimization, and combining it with our color range determination approach, a better alpha matte is generated. Our method estimates the foreground and background color in the unknown region by choosing good samples, and then smoothes the initial alpha matte to generate a better result. This approach does not need user specified trimap or scribbles.

Chapter 4

Implementation and experimental results

4.1 Implementation

For implementation, we consider the proposed scheme as two parts, color range determination and robust matting [4], the basic block diagram is shown in Fig. 4.1. In the first part, the color range determination removes the background colors and outputs trimap. In the second part, the robust matting generates high quality alpha matte based on the trimap and original image. Both parts of the system need user interface (UI), including pre-defined parameters and post-adjusted parameters, which are illustrated in implementation details. We will first discuss the implementation of the color range determination and then the robust matting.

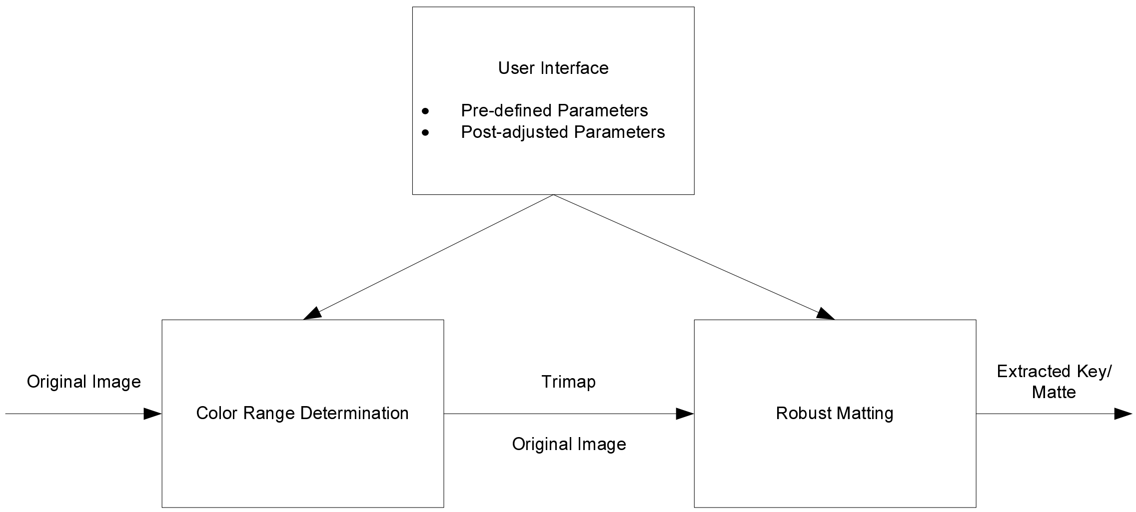


Figure 4.1: Block diagram of proposed scheme.

4.1.1 Color range determination

Color range determination includes five processing steps. The flow chart is shown in Fig. 4.2. They are background color initialization, K-means partition, background color removal, optimization, and trimap generation.

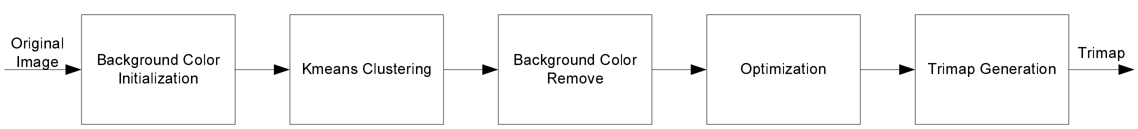


Figure 4.2: Processing blocks of color range determination.

Color initialization

There are three steps in color initialization: color selection, rotation, and definite foreground color removal.

1. Color selection: In video industry, rough color selection is designed for switcher operator to pick one color that is close to real background color from six primitive colors [19]. They are red, magenta, blue, cyan, green, and yellow. In RGB color space, six

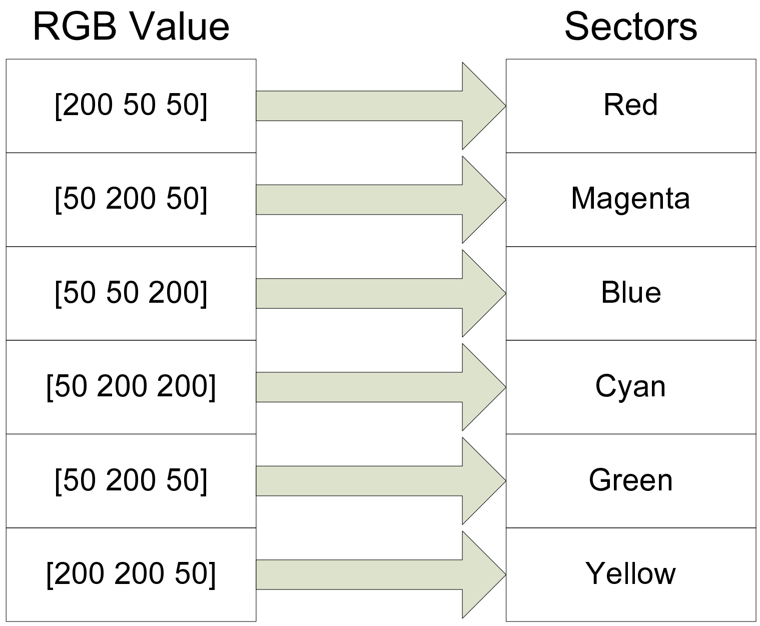


Figure 4.3: Sector values correspondence.

values are corresponded to the six sectors, as shown in Fig. 4.3. After the initial sector is chosen, its corresponding value is converted from RGB color space to YCbCr color space.

2. Rotation: consider the line from origin to the selected color in YCbCr color space, the slope of the tangent is $\frac{Cr}{Cb}$, Thus the corresponding angle is $\theta = \arctan(\frac{Cr}{Cb})$ $(\frac{Cr}{Cb} \geq 0)$or $\theta = \arctan(\frac{Cr}{Cb} + \pi)$ $(\frac{Cr}{Cb} < 0)$. The color section is rotated to make the center point falls on the Cb axis, then the section is symmetric to Cb axis. To rotate point (x, y) counterclockwise with an angle of θ, the rotation formula is as follows:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{4.1}$$

where (x', y') is the point after rotation. Notice that, since the values for Cb and Cr before rotation are in the range of $[0, 255]$. We convert it to the range of $[-128, 127]$ before rotation, and then recover them to $[0, 255]$ after rotation. Since each color section has a fair angle of 60 degree, the target sector is equally divided by the Cb axis after rotation, so the tangent angle is 30 degree.

3. Definite foreground color removal: the colors that are far away from the pre-defined background region are discarded first, because they can be considered as foreground colors directly. To determine whether a color is definitely foreground, the Cb and Cr color range is converted from $[0, 255]$ to $[-128, 127]$, and tangent angle of target color pixel is computed. If the angle is out of $[-30°, 30°]$, then it is believed far away from the pre-defined region and the corresponding colors are removed. At the same time, colors that are near the origin in the selected section are removed too, because they are close to all other color sections, and background colors can not fall in such a region. An adjusted parameter or threshold is given to determine whether the color is near the origin enough. It is set to be 150 as default.

K-means clustering

Currently, we have removed the definite foreground color. However, some colors in the pre-defined color region is still possible to be a foreground color. For constant color image or gridded image with limited color variations, only several groups of color can be viewed as background. In order to remove the background colors, the centroids of the groups should be located first. K-means clustering is employed to cluster these colors. The concept of K-means is discussed in Chapter 3.

The function “K-means” in matlab is used to cluster the colors. Notice that, K-means is a heuristic algorithm, it randomly selects k initial means from the data set, and the results depend on the initial seeds. Thus it cannot be guaranteed that it would converge to a global optimum result. Sometimes, this method generates an empty cluster when the initial defined seed is far away from the aggregate colors. To avoid this problem, extra action is taken when a cluster loses all its member observations, and a new cluster is created to consist of the pixel that furthest from its centroid. The corresponding parameter and value of K-means are ("emptyaction", "singleton").

In addition, the input of K-means can be optimized. For instance, given a high resolution picture (e.g., 1920×1080), there are millions of pixels in the whole image. Therefore, hundreds or even thousands of pixels can have the same Cb-Cr value, which requires a lot of time to cluster them during the K-means' processing. Furthermore, to find the major group centroids, the emphasis should be given to the relative ratio of each group instead of the isolated number on Cb-Cr color space. In order to reduce the computing time, as well as remain the relative ratio information, a quantization step is enforced to reduce the size of the data set. The number of pixels on each Cb-Cr point is quantized with a predefined value, which is another user adjusted parameters.

In the system, the default value for the number of seeds "k" is set to 4, and it works well for constant color images with different light condition and gridded images. Sometimes, this value should be changed to fit some images (e.g., the value is changed to one to fit solid background image).

Background color removal

It is obvious that the background colors are gathered around the K-means centroids. Each centroid forms a circle around them. The radius (default value is 6) is a predefined parameter and can be adjusted, and any color falls in these circles is considered as background. Grey color (128, 128, 128) is used to replace the background color, the result is shown in Fig. 4.4 (1b), (2b), and (3b).

(1a) Original image.

(1b) The result of background color removal.

(1c) The result after optimization.

(2a) Original image.

(2b) The result of background color removal.

(2c) The result after optimization.

(3a) Original image.

(3b) The result of background color removal.

(3c) The result after optimization.

Figure 4.4: The result of background color removal and optimization.

It is obvious that the three images have different features, which are used to test the proposed scheme in different aspects. In (1a), there is a woman photographed against a green backdrop with different light condition. In (2a), two chairs are captured against a blue backdrop with a gridded wall. In (3a), a woman is standing in front of a cyan background with complex hair stripes. We can see that some green background colors have not been removed in (1b), and some blue background colors have not been removed in (2b). Although cyan background color are removed in (3b), the result loses some information of hair details. Thus, the results are not favorable for any of the three images. The work to remove these extra background colors is discussed below.

Optimization

In order to improve the results, especially for (1b) and (2b). We can see that the remaining background colors are near or encompassed by gray color (identified as background), thus this kind of spatial information is used to remove the remaining background colors. For each pixel in a pre-defined window, its distance to the cluster centroids is calculated. If the pixel is already marked as background, the distance is defined as zero. The corresponding smallest values from the four are added together within the window as value "sum1". Also, the squared sum of distance for the pixels in the window to the known centroids is computed, and then adds them together to generate a value "sum2". In the implementation, the default window size is 5×5, and the default number of centroids is 4. Finally, if either sum1 or sum2 is smaller than a given threshold, the whole window is marked as background. Thus, extra background color information can be removed.

The optimized results are shown in 4.4 (1c), (2c), and (3c). It is obvious that the results shown in (1c) and (2c) are much better than (1b) and (2b). Most extra background colors are removed, only with some green light reflection along the boundary of woman and blue light reflection along the boundary of chair. However, the result is not acceptable in (3c), because the hair details are lost. Therefore, high quality alpha matting methods are needed to improve the results, especially for (3c).

Alpha matting methods need trimap or user specified scribbles as input. This requirement is not supported in some applications, such as real time video switcher. It is almost impossible to input trimap or scribbles for each frame. Our solution is to automatically generate trimap, which saves a lot of complex work for user. After obtaining the trimap, the final result can be optimized by using advanced alpha matting algorithm.

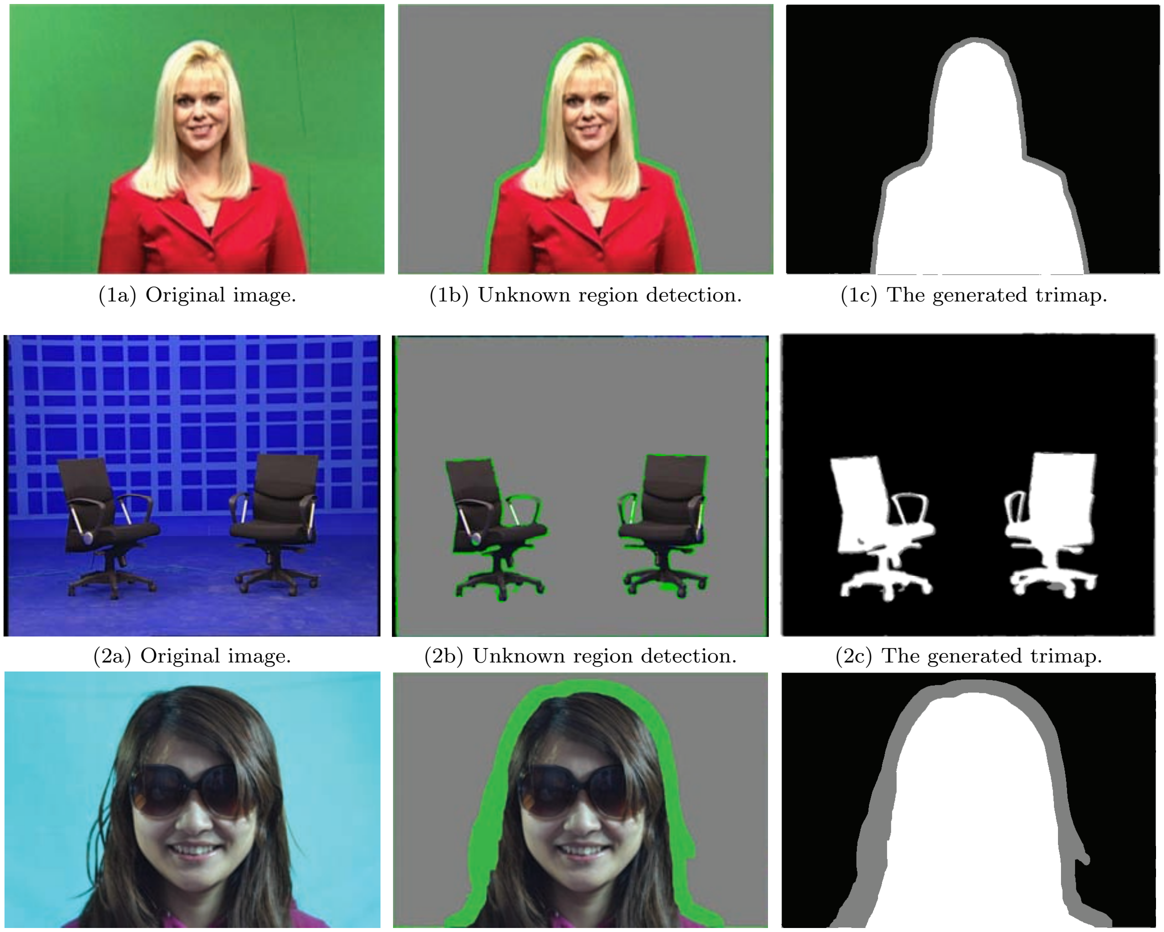
Trimap generation

Till now, we have the knowledge that whether a pixel is foreground or background. But this kind of method cannot generate high quality alpha matte, especially for some images with the semi-transparent part or refined part as hair stripes, because it is a hard segmentation method. In order to generate alpha matte for an image, unknown region should be identified.

Based on the knowledge we already have, a map is used to mark each pixel, no matter whether it is foreground or background. A window is used to scan the whole image, and any window that covers both foreground and background pixels more than a threshold is marked as unknown region. The default threshold is set to the side length of the window.

Next, known region is expanded according to Gastal's known region expansion method [29]. For a target pixel in the unknown region, the nearest foreground and background pixels are found. If it is smaller than a given threshold (e.g., 9 pixels in our system), and distance in the color space is smaller than another threshold (e.g., 6/256 in our system), the pixel is assigned to known foreground or background.

The generated trimaps for two typical images with hair stripes are shown in Fig. 4.5.



(1a) Original image. (1b) Unknown region detection. (1c) The generated trimap.

(2a) Original image. (2b) Unknown region detection. (2c) The generated trimap.

(3a) Original image. (3b) Unknown region detection. (3c) The generated trimap.

Figure 4.5: The result of trimap generation.

4.1.2 Robust Matting

The work flow of robust matting is shown in Fig.4.6, it includes optimized color sampling based on confidence evaluation, matte optimization based on modified random walk and post-refinement.

Optimized color sampling

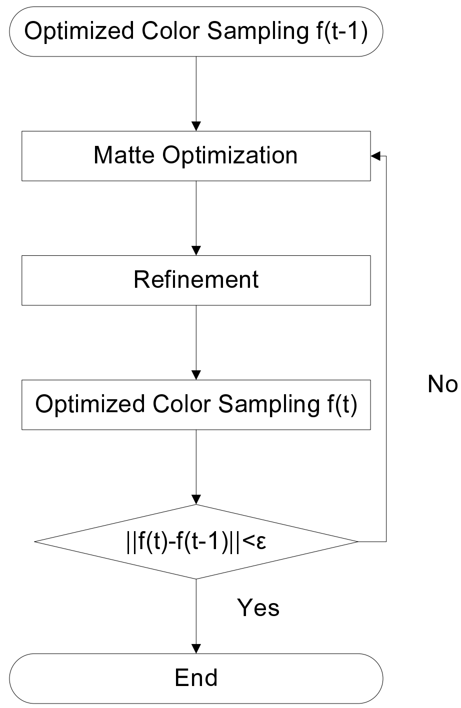


Figure 4.6: System workflow.

The first step for robust matting is optimized color sampling. How to select the foreground and background candidate set is considered first. Since the samples are chosen from the boundary of the known foreground and background, the boundary points should be located first. Our solution is to convert the trimap into two images by marking the unknown region as either known foreground or known background first.

Next, in order to extract the boundary of known foreground and known background, a commonly used Sobel operator is employed to detect the edge. It is a discrete differentiation operator that computes the approximation of the gradient of the image intensity function. Two 3×3 kernels are convolved with the original image to calculate the approximations of the derivatives. Suppose the original image is I, then the horizontal and vertical gradient G_h, G_v are computed as:

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} * I \tag{4.2}$$

$$G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} * I \tag{4.3}$$

The gradient across the whole image G is calculated by combining the two gradients as:

$$G = \sqrt{G_x^2 + G_y^2} \tag{4.4}$$

when the gradient at some points in the image is relatively high enough, it is likely to be an edge. The result of Sobel edge detection is shown in Fig. 4.7 ((1b) (1c) (2b) (2c) (3b) and (3c)).

In order to obtain the nearby samples from the boundary and span the variation of color as much as possible, the distance between the pixel in the unknown region and the pixels in the foreground and background boundary are calculated. The function of "Sortrows" in matlab is used to sort the distance by descending order, and the samples are retrieved from this list according to an interval value to make it span more color variations. The selection process for the samples is illustrated in Fig. 4.8. The foreground and background pixels are arranged by the distance from the pixels in the foreground and background boundary to the target pixel in the unknown region. And

(1a) Generated trimap. (1b) Boundary of known foreground. (1c) Boundary of known background.

(2a) Generated trimap. (2b) Boundary of known foreground. (2c) Boundary of known background.

(3a) Generated trimap. (3b) Boundary of known foreground. (3c) Boundary of known background.

Figure 4.7: Unknown region boundary detection.

each sample is selected from every three samples according to an interval value 2 in our system.

In our system, the default number of foreground samples for each target point in the unknown region is 20, and the number is 10 for the background samples. We choose only 10 points from background instead of 20 in [4], because there are less variations in background color in our application. Therefore, for each target pixel in the unknown region, there are $20 \times 10 = 200$ pairs. The initial alpha and confidence value for each of

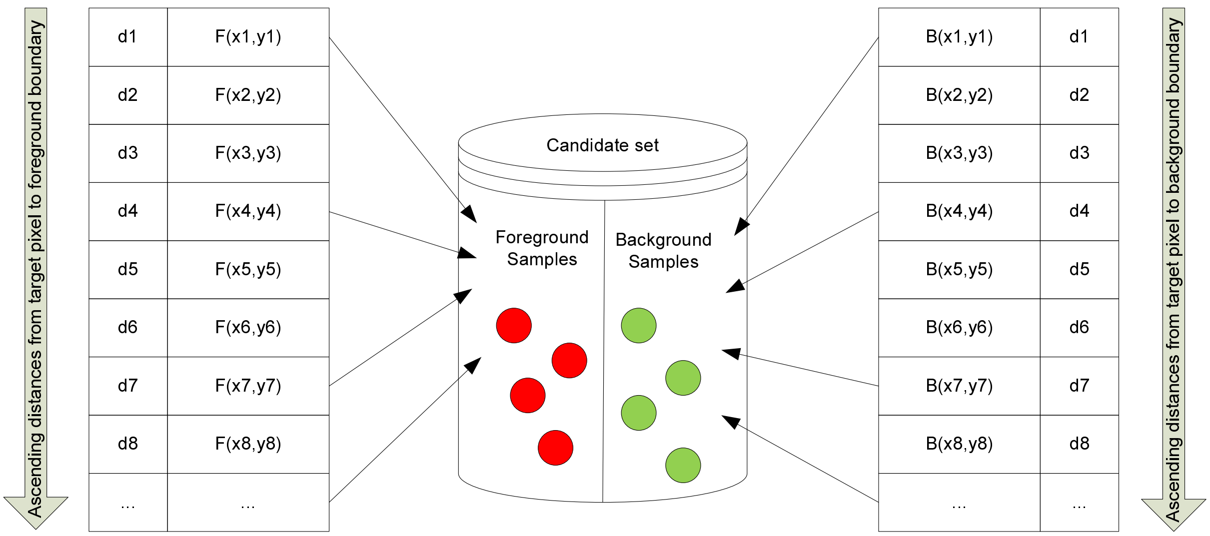


Figure 4.8: Selection criteria for foreground and background samples

them is computed according to Equ. (3.6, 3.7, 3.8, 3.9, 3.10). Three pairs with highest confidence values are chosen from the 200 pairs, and then the confidence and alpha values are calculated by averaging the three values.

Matte optimization

Since the result of initial alpha matte contains some noise, modified random walk algorithm is applied to optimize the matte. Classic random walk will be introduced first, and then the modified random walk is discussed. Leo Grady wrote the open source code for classic random walk algorithm in Matlab [46]. The work flow is shown in Fig. 4.9.

1. Build graph. Generates a 2D four connected Cartesian lattice, all the edge information are stored as the forms of two end nodes.

2. Generate weights. Save 2D image to an array, for a color image with three components, (e.g., RGB image), the three components of the image are saved one by

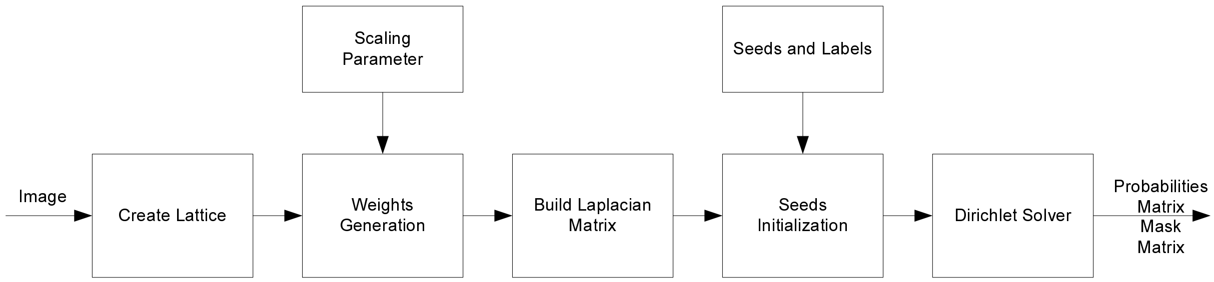


Figure 4.9: Random walk work flow.

one in the array. The weights are assigned to each edge based on the color difference between end points. Then, scaling parameters can be applied to these weights.

3. Generate Laplacian matrix. Sparse Laplacian matrix is build according to Equ. (3.13) we have discussed in the last chapter.

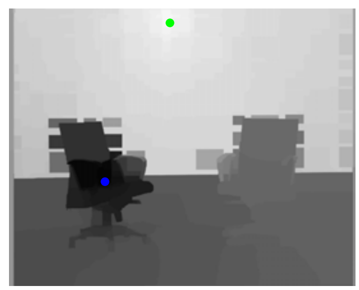
4. Seeds initialization. Manually define initial seeds and assign labels to each seeds. For keying or matting problem, two seeds (foreground seed and background seed) are used to mark the foreground color and background color, as well as location information.

5. Combinatorial Dirichlet solver. Solve the Dirichlet problem according to Equ. (3.18). “Backslash” operation in matlab is used to solve the linear system, similar approaches are lower upper (LU) decomposition solver and Cholesky [47]. But when they face to large dimensional problems, the operation is relatively slow, more advanced linear system solvers (e.g., Conjugate Gradient (CG) [4], successive over relaxation (SOR)[48]) can be applied to improve the performance. Mask and probabilities matrixes are generated finally. Mask matrix that contains a label of each pixel with value 1 to k corresponding to each seed, indicating the group or cluster after segmentation. Probabilities matrix includes the information that the probability that each pixel belongs to each seed. Probabilities matrix contains accurate information that how “far” the pixel to the seed in both color and spatial distance consideration. Applying random walk to several images with two initial seeds, the results are shown in Fig. 4.10

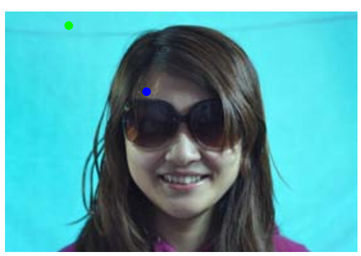
(a) Original image.

(b) Random walk segmented image.

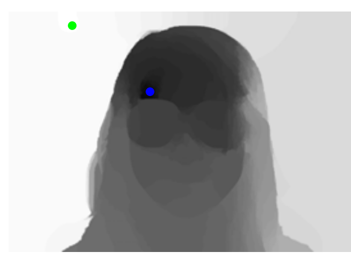
(e) Original image.



(f) Random walk segmented image.



(g) Original image.



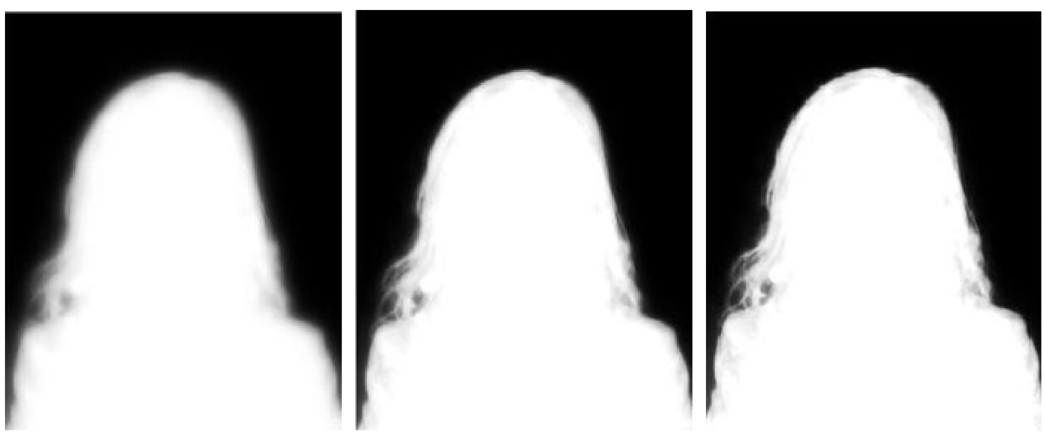
(h) Random walk segmented image.

Figure 4.10: Image segmentation based on random walk.

In order to optimize the initial alpha in last step, the classic random walk algorithm needs to be modified. There are two modifications:

1. In the step of lattice creating, virtual foreground node and virtual background nodes are added. Both of them connected to all the existing nodes. If there are n pixels in the image, then 2n edges are added to the existing lattice.

2. Instead of using neighborhood color difference as the weights, new weights are generated according to Equ. (3.19 3.20 3.21), and the initial alpha and confidence information are used. After combining them, modified random walk is able to smooth and optimize the initial alpha matte. Notice that there is a parameter γ used to balance the data weight and edge weight. On the one hand, setting the weight too high will generate noisy matte; on the other hand, setting it too low will obtain over smoothed matte. The influence of the parameter is illustrated in Fig. 4.11.



(a) $\gamma = 0.1$. (b)$\gamma = 0.5$. (c)$\gamma = 1$.

Figure 4.11: Mattes generated with different parameters.

Refinement

The final step is the refinement and iteration. Pixels are assigned to known foreground set if their alpha values are larger than 0.98. Similarly, some pixels are assigned to known background set if their alpha values are smaller than 0.02. Optimized color sampling and modified random walk is applied to the image with new foreground and background samples again. The default iteration times are three. The trimap is more accurate after each iteration process, as shown in Fig. 4.12

(a) Unknown region before refinement. (b) Unknown region after refinement .

Figure 4.12: Refinement result. Compared to (a), the red unknown region is smaller and more accurate after refinement in (b).

4.2 Experimental results

In order to verify the effectiveness of our research work, the experimental results will be demonstrated in two parts. The first part shows the good performance of our method when dealing with refined regions of image. The second part shows it is easier to use our method.

4.2.1 Part 1

In this part, the experimental results demonstrate the good performance of our methods.

The comparison results are shown in Fig. 4.13 and Fig. 4.14. The left column are the original images; the middle column are the results from color range determination by K-means clustering; the right column of the images are generated by our improved methods.

It can be observed that the results after improvement are much better in three aspects.

1. For the refined transition regions such as hair stripes, color range determination method can not extract the complex structures of hair and only provides a hard (i.e.

binary) segmentation, which means it only considers the foreground and background regions without refined transition regions such as the hair stripes. Compared with the hard segmentation, our improved method can extract the detailed structure of the hair stripes so that a more precise foreground extraction can be achieved. This advantage is shown in Fig. 4.13 (1c) (3c) and Fig. 4.14 (3c).

2. For the simple transition regions: On one hand, a precise foreground extraction is expected to be achieved by color range determination methods. However, for easy structured images such as Fig. 4.14 (1a) and (2a), the background color will be reflected on the edges of foreground object in the results before improvement. On the other hand, the improved method can extract the foreground object without background color reflection. This advantage is shown in Fig. 4.14 (1c) and (2c).

3. For the semi-transparent regions: Color range determination methods cannot extract the partial coverage of foreground object in semi-transparent regions which can be seen in the region under the left forearm of woman in Fig. 4.13 (2b). Our improved method can solve this problem well and the advantage can be seen in Fig. 4.13 (2c).

4.2.2 Part 2

In this part, we show that our proposed scheme is easier to use. While having the same keying quality compared to Wang [4], our method does not require the complex input such as trimap or scribbles from users, which is required by Wang's method. Our method is more convenient for users in applications because user only needs to specify the rough color range. The proposed scheme is also compatible with existing video switchers in video industry.

The comparison results are shown in Fig. 4.15 to Fig. 4.20. The pictures in the left column are the results produced by Wang's algorithm; the pictures in the right

(1a) Original image. (1b) The result without improvement. (1c) Improved result.

(2a) Original image. (2b) The result without improvement. (2c) Improved result.

(3a) Original image. (3b) The result without improvement. (3c) Improved result.

Figure 4.13: Improvement by robust matting (I).

column are the results produced by our proposed scheme. It can be observed that we can generate similar results as Wang's algorithm and the visual difference is difficult to be observed.

To sum up, our proposed system is good at handling gridded color images, as well as constant color images with different light conditions, by employing K-means clustering method. Also, this method is able to generate high quality alpha matte, by exploiting

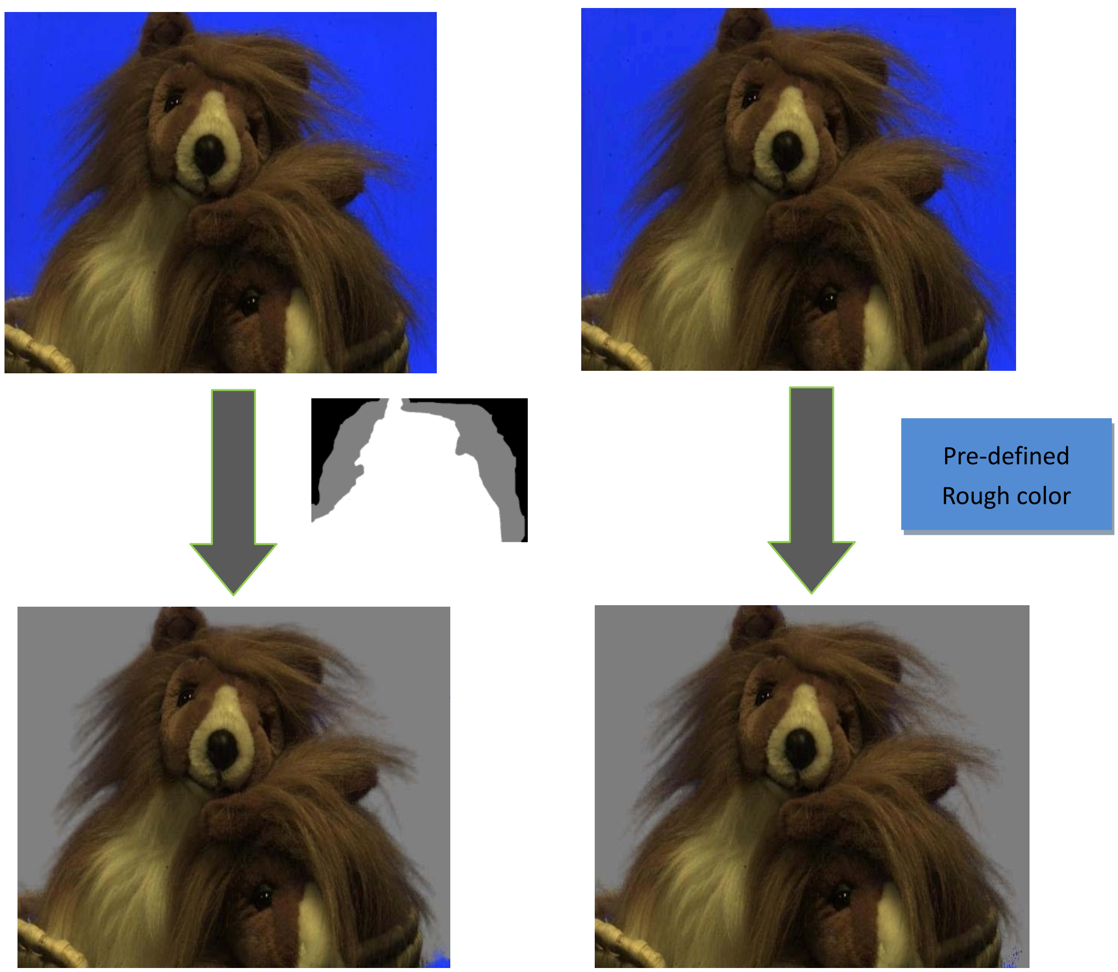
(1a) Original image. (1b) The result without improvement. (1c) Improved result.

(2a) Original image. (2b) The result without improvement. (2c) Improved result.

(3a) Original image. (3b) The result without improvement. (3c) Improved result.

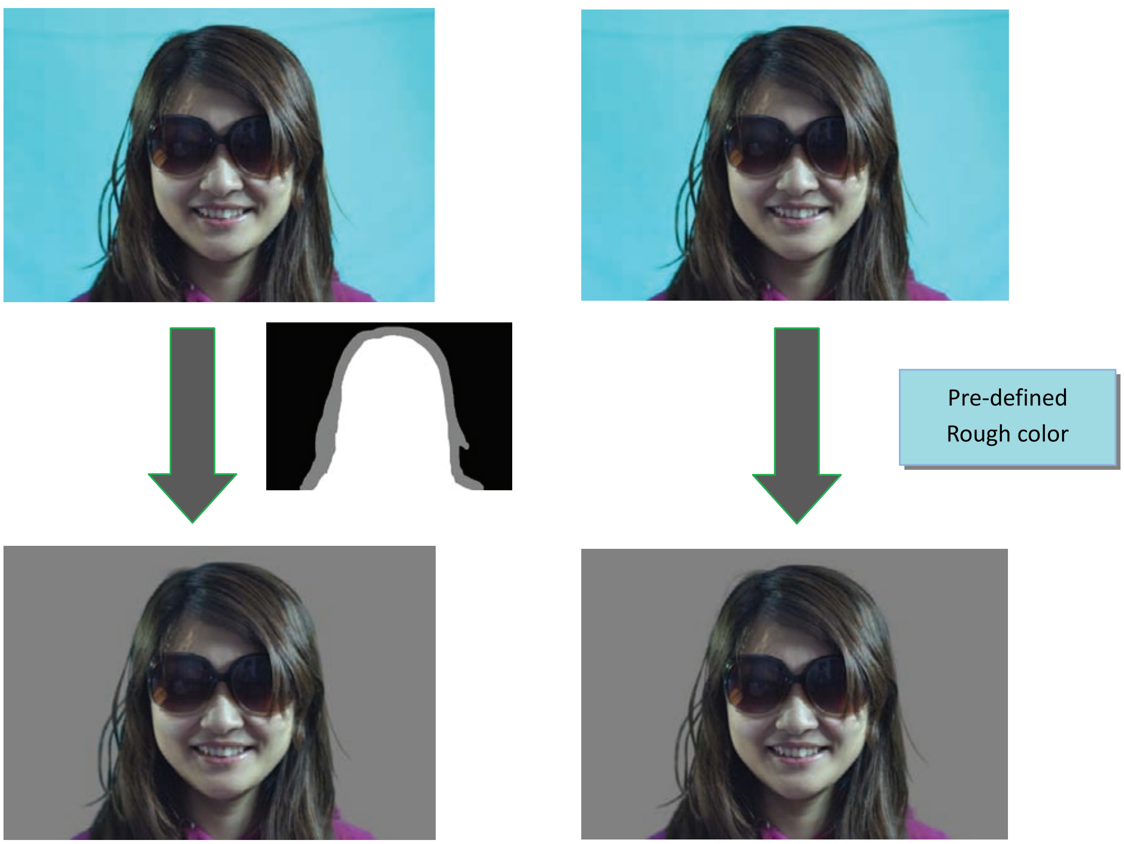
Figure 4.14: Improvement by robust matting (II).

the good characteristic of confidence evaluation and by using modified random walk.



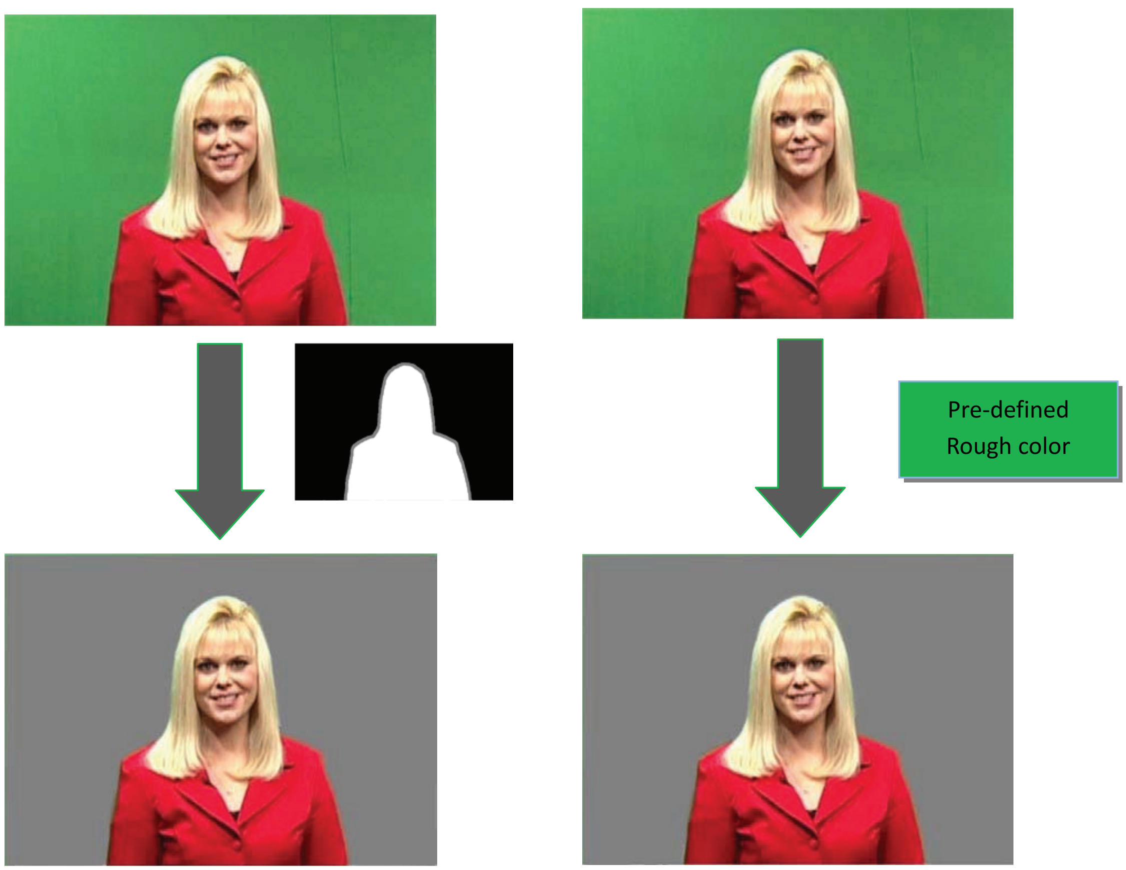
(a) Wang's scheme. (b) Our scheme.

Figure 4.15: Comparison with Wang's method (I).



(a) Wang's scheme. (b) Our scheme.

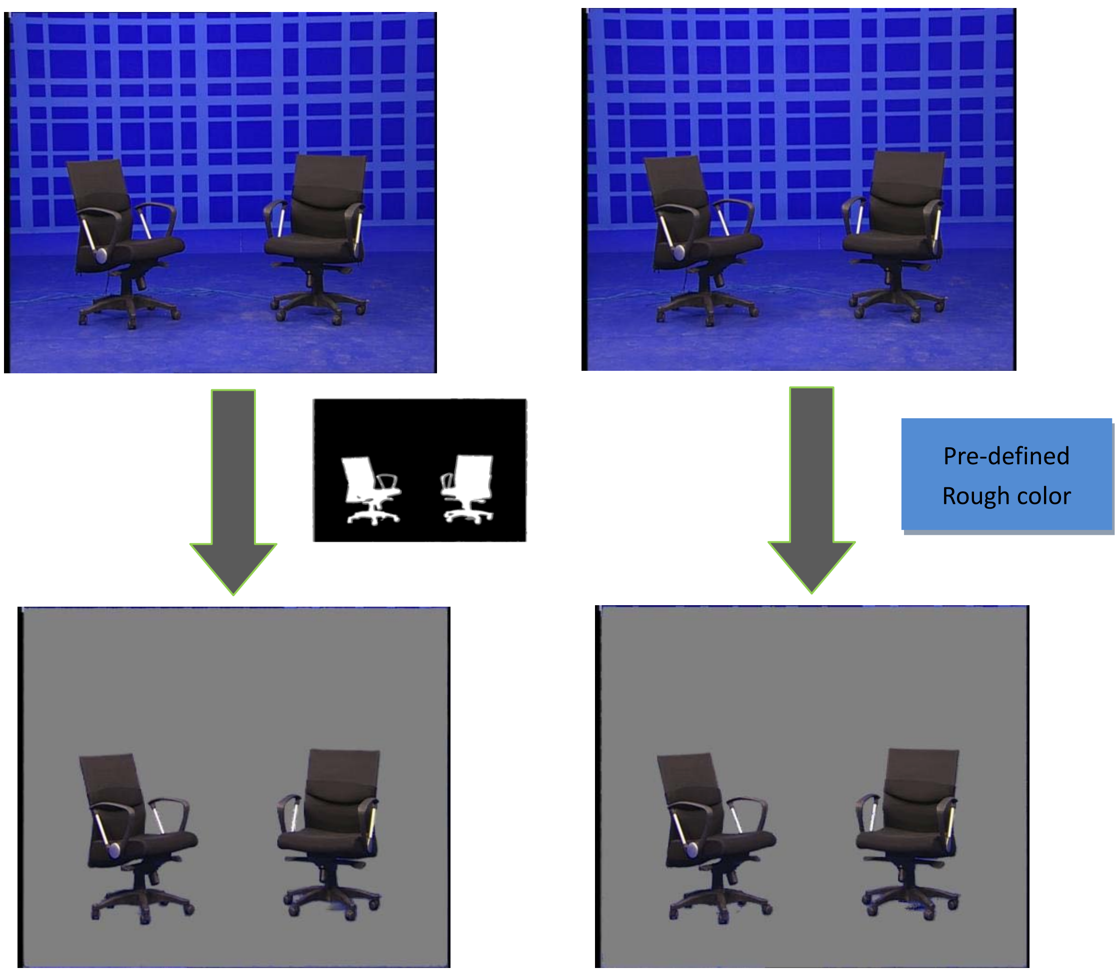
Figure 4.16: Comparison with Wang's method (II).



(a) Wang's scheme.

(b) Our scheme.

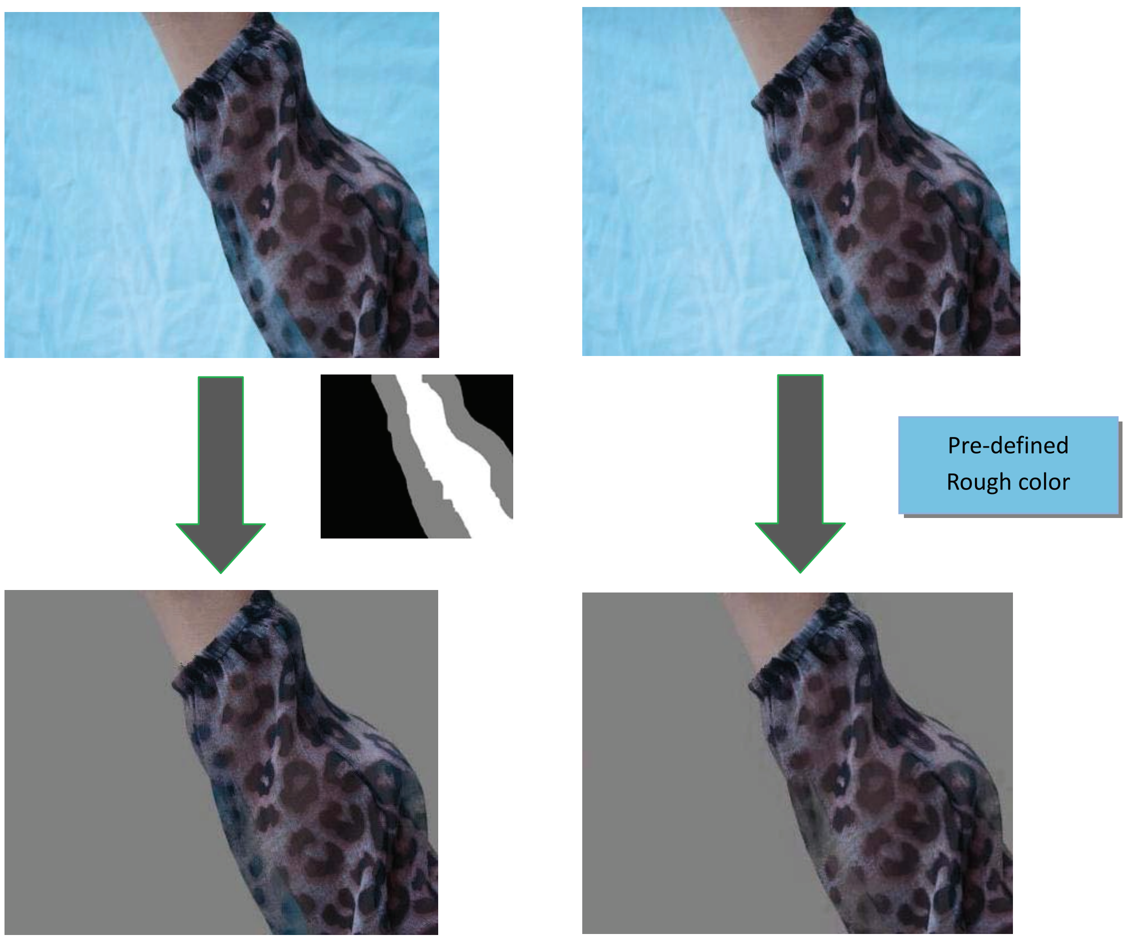
Figure 4.17: Comparison with Wang's method (III).



(a) Wang's scheme.

(b) Our scheme.

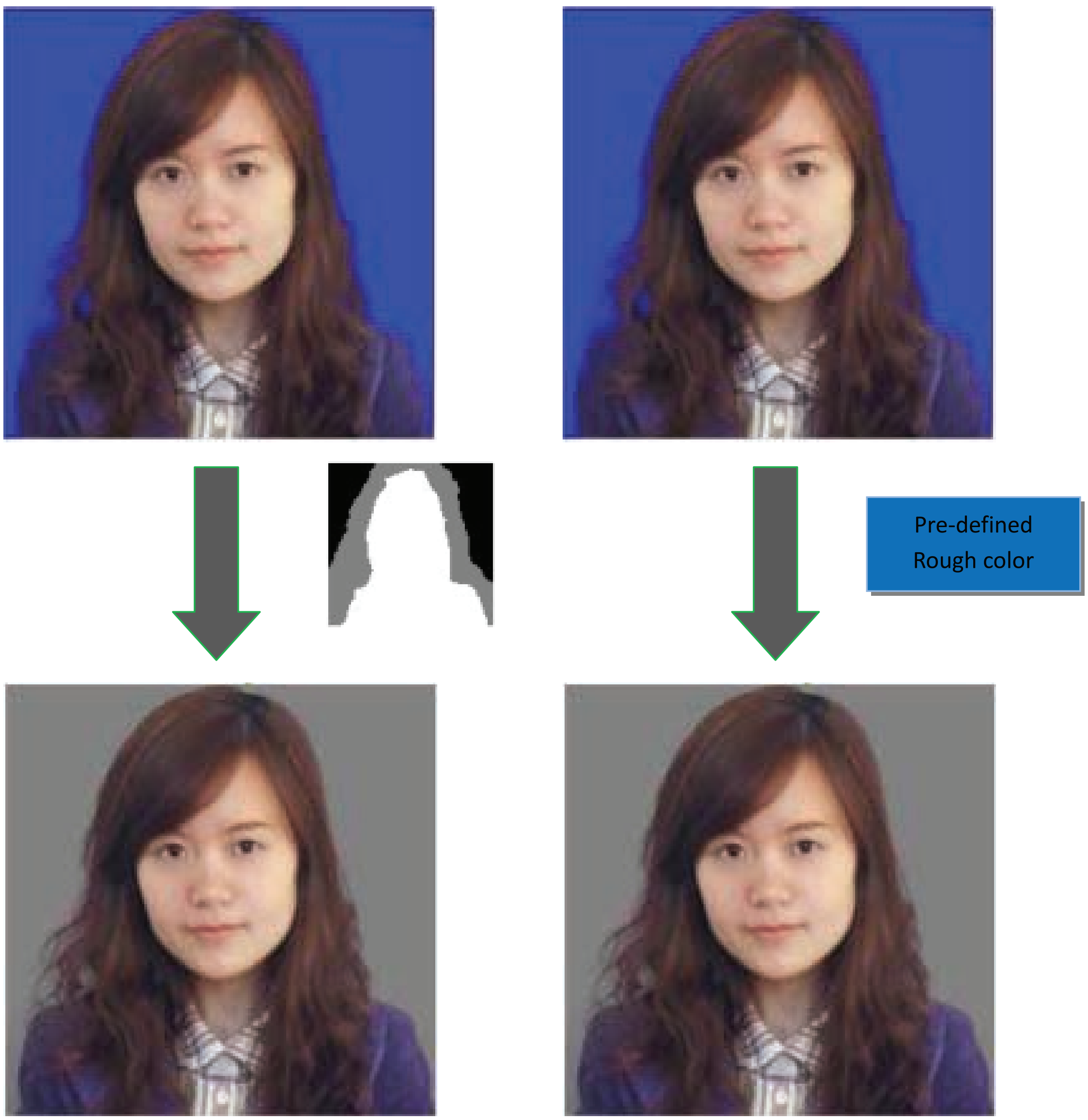
Figure 4.18: Comparison with Wang's method (IV).



(a) Wang's scheme.

(b) Our scheme.

Figure 4.19: Comparison with Wang's method (V).



(a) Wang's scheme.

(b) Our scheme.

Figure 4.20: Comparison with Wang's method (VI).

Chapter 5

Conclusions and future work

5.1 Conclusions

In this thesis, we proposed a method to solve the foreground extraction for color images, especially for multiple color images. On one hand, color range determination method cannot generate high quality keys or matte for complex image (e.g., image with semi-transparent part or hair stripes). On the other hand, advanced alpha matting algorithm requires manually specified trimap as input, which is a complex and time consuming task. In addition, the results are not stable, because it depends on the user input a lot. Unskilled user may generate bad trimap as input, and lead to unfavorable matting results.

Our consideration is to apply the advanced alpha matting algorithm to improve the result of traditional color range determination method. At the same time, the system does not need user specified trimap or paint strokes. It makes the system more robust and stable. Furthermore, the input of the system (e.g., rough color sector selection and adjusted parameter) is compatible with the existing technology, such as video switcher used in video industry, which provides several knobs to adjust the parameter values.

5.2 Future work

The future work is summarized as:

The final matting results can be further improved. To composite the image to a new background, the color spill problem needs to be solved. This is another problem related to image compositing. And this method can be furthered applied to video matting. There are two problems to be considered. Firstly, the requirement for user interaction is tedious. Also, slight differences in the extraction from frame to frame can lead to a lack of temporal coherence, which further resulting in a jittery video. Because human visualization system (HVS) is very sensitive to temporal inconsistencies [49], advanced algorithm is needed for video matting for future research.

References

[1] http://en.wikipedia.org/wiki/Alpha compositing, (Last visited on 9/29/2011).

[2] T. Porter and T. Duff, Compositing digital images, Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques, vol. 18, pp. 253-259, New York, USA, January 1984.

[3] M. A. Ruzon and C. Tomasi, Alpha estimation in natural images, IEEE Conference on Computer Vision and Pattern, vol. 1, pp. 18-25, January 2000.

[4] J. Wang and M. F. Cohen, Optimized color sampling for robust matting, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-8, Minneapolis, USA, June 2007.

[5] S. Lin and J. Shi, Fast natural image matting in perceptual color space, Proceedings of Computers and Graphics, vol. 29, pp. 403-414, 2005.

[6] Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski, A Bayesian approach to digital matting, Proceedings of IEEE International Conference

on Computer Vision and Pattern Recognition (CVPR 2001), vol. 2, pp. 264-271, 2001.

[7] http://cuvideoedit.com/chroma-keying.php, (Last visited on 22/10/2011).

[8] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott, Alpha matting evaluation website, www.alphamatting.com. (Last visited 09/05/2011)

[9] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut" - interactive foreground extraction using iterated graph cuts, ACM Transactions on Graphics (SIGGRAPH), vol. 23, no. 3, pp. 309-314, New York, USA, August 2004.

[10] Y. Boykov and M. P. Jolly, Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images, Proceedings of IEEE International Conference on Computer Vision (ICCV), vol. 1, pp. 105-112, July 2001.

[11] K. Tan and N. Ahuja, Selecting objects with freehand sketches, Proceedings of IEEE International Conference on Computer Vision (ICCV), vol. 1, pp. 337-344, August 2002.

[12] K. Tan and N. Ahuja, Representation for image structure and its application to object selection using freehand sketches, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 667-683, 2001.

[13] Y. Lin, J. Sun, C. Tang, and H. Shum, Lazy snapping, ACM Transactions on Graphics (SIGGRAPH), vol. 23, no. 3, pp. 303-308, August 2004.

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

[15] V. Iverson, Chroma-key color range determination, US Patent 5,774,191, June 1998.

[16] M. Uya, Chroma-key signal generator, US Patent 5,838,310, November 1998.

[17] R. L. Miller, Apparatus and method for compositing video images, US Patent 5,812,214, September 1998.

[18] K. Yamamoto and J. Yonemitsu, Key signal generating apparatus for digital chromakey system, US Patent 4,533,937, August 1985.

[19] Y. Liu, D. A. Ross, and A. J. Fryer, Method, system, and device for automatic determination of nominal backing color and a range thereof, US Patent 7,508,455 B2, March 2009.

[20] D. Pettigrew, Compositing video image data, US Patent 6,445,816, September 2002.

[21] R. Gehrmann, Chromakey method and associated circuit arrangement, US Patent 5,719,640, February 1998.

[22] A. Demay and M. L. Lan, Device and method for determining a key for clipping a subject moving against a colored background, US Patent 5,903,318, May 1999.

[23] A. Demay and M. L. Lan, Device and method for processing a signal with a subject moving against a colored background, US Patent 5,923,381, July 1999.

[24] Y. Mishima, Soft edge chroma-key generation based upon hexoctahedral color space, US Patent 5,353,174, October 1994.

[25] J. Wang and M. Cohen, Image and video matting: a survey, Foundations and trends in computer graphics and vision, vol. 3, no. 2, pp. 97-175, 2007.

[26] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis, Improved fast gauss transform and efficient kernel density estimation, IEEE International Conference on Computer Vision, vol. 1, pp. 464-471, Nice, France, October 2003.

[27] X. Bai and G. Sapiro, A geodesic framework for fast interactive image and video segmentation and matting, IEEE International Conference on Computer Vision, pp. 1-8, Rio de Janeiro, Brazil, October 2007.

[28] J. Sun, J. Jia, C. Tang, and H. Shum, Poisson matting, ACM Transactions on Graphics, vol. 23, no. 3, pp. 315-321, July 2004.

[29] E. S. L. Gastal and M. M. Oliveira, Shared sampling for real-time alpha matting, Proceedings of Eurographics, vol. 29, pp. 575-584, 2010.

[30] J. Weinman, A. Hanson, and A. Mccallum, Sign detection in natural images with conditional random fields, Proceedings of IEEE International Workshop on Machine Learning for Signal Processing, pp. 549-558, Sao Luis, Brazil, September 2004.

[31] J. Wang and M. F. Cohen, An iterative optimization approach for unified image segmentation and matting, IEEE International Conference on Computer Vision (ICCV), vol. 2, pp. 936-943, Beijing, China, October 2005.

[32] M. Hasanzadeh, S. Kasaei, and H. Moheseni, A new fuzzy connectedness relation for image segmentation, International Conference on Information and Communication Technologies: From Theory to Applications, pp. 1-6, Damascus, Syria, April 2008.

[33] Y. Zheng, C. Kambhamettu, J. Yu, T. Bauer, and K. Steiner, FuzzyMatte: a computationally efficient scheme for interactive matting, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-8, Los Alamitos, USA, 2008.

[34] J. K. Udupa and S. Samarasekera, Fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation, Proceedings of Computer Vision, Graphics, and Image Processing (CVGIP): Graphical Models and Image Processing, vol. 58, no. 3, pp. 246-261, May 1996.

[35] Y. Guan, W. Chen, X. Liang, Z. Ding, and Q. Peng, Easy matting - a stroke based approach for continuous image matting, Proceedings of

Eurographics in Journal Computer Graphics Forum, vol. 25, issue 3, pp. 567-576, September 2006.

[36] Y. Zheng and C. Kambhamettu, Learning based digital matting, IEEE International Conference on Computer Vision, pp. 889-896, Kyoto, Japan, September 2009.

[37] R. Gvili, A. Kaplan, E. Ofek, and G. Yahav, Depth keying, Proceedings of SPIE on Stereoscopic Displays and Virtual Reality Systems X, vol. 5006, no. 1, pp. 564-574, Santa Clara, USA, January 2003.

[38] G. Yahav, G. J. Iddan, and D. Mandelboum, 3D imaging camera for gaming application, International Conference on Consumer Electronics (ICCE), pp. 1-2, Las Vegas, USA, January 2007.

[39] O. Wang, J. Finger, Q. Yang, J. Davis, and R. Yang, Automatic natural video matting with depth, Proceedings of Pacific Conference on Computer Graphics and Applications, pp. 469-472, 2007.

[40] E. Reinhard and E. A. Khan, Depth-of-field-based alpha-matte extraction, Proceedings of Applied Perception in Graphics and Visualization (APGV 2005), pp. 95-102, 2005.

[41] Y. Bando, B. Chen, and T. Nishita, Extracting depth and matte using a color-filtered aperture, Proceedings of ACM SIGGRAPH, vol. 27, no. 5, article 134, New York, USA, December 2008.

[42] Wikipedia, http://en.wikipedia.org/wiki/YCbCr, (Last visited on 25/09/2011).

[43] J. B. MacQueen, Some methods for classification and analysis of multivariate observations, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281-297, 1967.

[44] J. Dodziuk, Difference equations isoperimetric inequality and the transience of certain random walks, Transactions of the American Mathematical Society, vol. 284, no. 2, pp. 787-794, August 1984.

[45] L. Grady, Random walks for image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 11, pp. 1768-1783, November 2006.

[46] L. Grady, Random walk open source code, http://cns.bu.edu/ l-grady/software.html, (Last visited on 25/09 2011).

[47] L. Duff, A. Erisman, and J. Reid, Direct methods for sparse matrices, Clarendon Press, Oxford, 1986.

[48] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, Successive overrelaxation(SOR), Numerical Recipes in Fortran: The Art of Scientific Computing, 2nd edition, Cambridge University Press, pp. 866-869, Cambridge, England, 1992.

[49] P. Villegas and X. Marichal, Perceptually-weighted evaluation criteria for segmentation masks in video sequences, IEEE Transactions on Image Processing, vol. 13, no. 8, pp. 1092-1103, 2004.