A NOVEL METHOD FOR CHINESE INK PAINTING COLORIZATION

Yun-Wen Wang, Chia-Min Hsu, and Zen-Chung Shih

Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, 30010, R.O.C
E-mail: vincentwant0114@hotmail.com, chiaminhsu.cs95g@nctu.edu.tw, zcshih@cs.nctu.edu.tw

ABSTRACT

The Chinese Ink Painting is an art with long history in Chinese culture. Painters can obtain various kinds of scenery by mixing water and ink properly. These papers provides a colorization technique that can transfer gray scale paintings to color paintings. Various colorization techniques for photorealistic images have good results. But these techniques are uncertainly suitable for Chinese Ink Painting. In our method, users only provide a gray scale Chinese Ink Painting and a similar color Chinese Ink Painting subjectively, system can automatically transfer the color from color painting to gray scale painting. We also provide a method for users to refine the automatically generated result.

Keywords: colorization, Chinese Ink Painting

1. INTRODUCTION

1.1 System Overview

Fig. 1 is the system flowchart. First a user takes a color painting as the reference image. The selected color painting should be similar to the gray painting which we want to colorize. For example, if we want to colorize a landscape painting, we cannot choose a Bird-and-flower painting. Instead we should select a color landscape painting.

Next, we convert original color to YUV because the correlation between the three channels is lower. Therefore, changing one of them will not cause much effect on others. This property is useful for the following steps. After color space transformation, we then partition a color painting into several regions roughly. Each region should be homogeneous in appearance. The image segmentation JSEG [1] is applied in our method.

After segmentation, we then determine which pixels in the gray painting should be colorized according to pixels in color painting. For getting a reasonable result and performance, luminance remapping and clustering are necessary before determining the relation between gray painting pixels and color painting pixels.

Finally, we provide a refinement method to enhance the results. Users can focus on the unsatisfied portion to edit, if the results are unsatisfied.

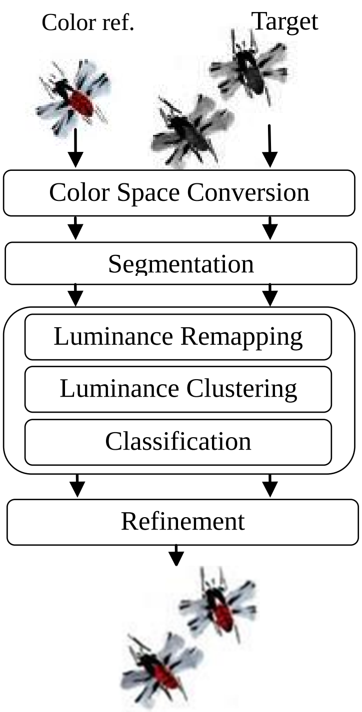

Fig. 1: the system flowchart

2. RELATED WORKS

2.1 Colorization by Reference Images

Reinhard et al. [2] described an automatic system for transferring the color pallet from one color image to another. The user can guide the process by specifying pairs of corresponding swatches in the source and target images.

Welsh et al. [3] colorized a gray image by transferring color from a source, color image, to a destination, gray image. First each image is converted into the $l\alpha\beta$ color space. Next, each pixel in the grayscale image in scan-line order and select the best matching sample in the color image using neighborhood statistics. The chromaticity values (α , β channels) of the best matching pixels are then transferred to the grayscale image to form the final image. Welsh et al.'s method works well on scenes where the image is divided into distinct luminance clusters. However, their current technique does not work very well with similar luminance.

Welsh's method only suits whit simpler images. As regards Chinese Ink Painting, their methods do not work very well because there are much detail objects in Chinese Ink

Painting.

Irony et al. [4] uses a supervised learning technique to better classify feature-space and a voting technique to increase the spatial consistency of the colorized image. This technique assumes that a similar example image is pre-segmented. Otherwise, the task of segmenting the example image itself can be almost as hard as colorizing the input image

Chen et al. [5] combines the grayscale image matting algorithm in Bayesian framework and color transferring techniques. In the beginning, the source grayscale image is separated into different objects using the grayscale image matting algorithm. Then, the objects are colorized using color transferring technique. Finally, the colorized objects are composited using alpha blending to reach the ultimate colorization.

Daniel et al. [6] propose a color-by-example technique which combines image segmentation, patch-based sampling and probabilistic reasoning. This method is able to automate colorization when new color information is applied on the already designed black-and white cartoon.

2.2 Colorization by User-Guided

Levin [7] and Yatziv [8] supposed that luminance is relative to color. Levin et al.'s method is based on a simple premise: neighboring pixels in space-time that have similar intensities should have similar colors. Artist only needs to annotate the image with a few color scribbles, and the indicated colors are automatically propagated in both space and time to produce a colorized image or sequence.

Yatziv et al. also proposed an interactive approach. Compared with Levin, Yatziv et al.'s method is more precise because their method is based on the concept of color blending. This blending is derived from a weighted distance function computed from the luminance channel. The colors are computed by blending many colors defined by users.

Luan et al. [9] presented an interactive system for users to easily colorize the natural image of complex scenes. In their system, colorization procedure is separated into two steps: Color labeling and Color mapping. Pixels that should roughly share similar colors are grouped into coherent regions in the color labeling, and the color mapping stage is then introduced to further fine-tune the colors in each coherent region. Besides, Luan et al.'s system could handle complex texture.

Wang et al. [10] proposed a Chinese Ink Painting colorization algorithm based on fuzzy segmentation. Their method has three main parts: fuzzy segmentation, colorization, over-drawing removal.

There are some researches that is user guided [4][11][12]. The user draws color strokes over the image, and the colors diffuse from the strokes outward across the image.

All above researches work very well on realistic image, but these researches are not suitable for Chinese Ink Painting because Chinese Ink Painting usually contains void and gradualness. "Void" is used in Chinese Ink Painting by Painters, for example, could, waterfalls, brooks and fog are painted by using "Void". "Gradualness" is also used between two objects in Chinese Ink Painting. "Void" and "Gradualness" parts should be preserved in the process of colorization. Existing Researches make color spread on these parts. Chinese Ink Painting has less grayscale information is another reason that existing researches are not suitable because these researches heavily rely on grayscale information.

In this paper, we present a method for Chinese Painting Colorization. Similar to Irony's work, we take one color image as the reference image.

3. SEGMENTATION

We use JSEG [1] in our system because of the color used in Chinese Ink Painting is fewer and simpler than realistic image. Good class-maps conduces the good segmentation results. Besides, JESG can preserve the homogeneity of texture. This property benefits us when we determine which pixels in the gray painting should be colorized according to pixels in color painting because the information we can use not only color but also texture information.

3.1 Criterion for Good Segmentation

Deng et al. [1] proposed an unsupervised segmentation algorithm. This algorithm contains two major parts: color quantization and spatial segmentation.

In color quantization, colors in the image are roughly quantized. The main purpose is to obtain only a few representative colors that can distinguish neighboring regions in an image. A color class is a set of image pixels quantized to the same color. The newly constructed image of labels is called a class-map. Examples of a class-map are shown in Table 1, where labels are represented by three symbols, '+', 'o', and '*'. After quantization, spatial segmentation works according to *J* value in class-map.

Table 1: An example of different class-map and *J* value

+ + + + + o o o o + + + + + o o o o + + + + + o o o o + + + + + o o o o + + + + + o o o o + + + + * * * * * + + + + * * * * * + + + + * * * * * + + + + * * * * *	+ * + * + * + * + o + o + o + o + o + * + * + * + * + o + o + o + o + o + * + * + * + * + o + o + o + o + o + * + * + * + * + o + o + o + o + o + * + * + * + * +	+ + + + + * o * o + + + + + o * o * + + + + + * o * o + + + + + o * o * + + + + + * o * o + + + + * o * o * + + + + o * o * o + + + + * o * o * + + + + o * o * o
Class-map1, *J=1.720*	Class-map2, *J=0*	Class-map 3, *J=0.855*

In class-map, the value of each point is the image pixel position, a 2D vector (x, y). Each point belongs to a color class. We can compute *J* value on these points. Let *Z* be the set of all *N* data points in a class-map.

Let $z=(x,y), z \in Z$, and m be the mean,

$$m = \frac{1}{N}\sum_{z \in Z} z \quad (1)$$

suppose Z is classified into C classes, Z_i, $i=1,...,C$. Let m_i be the mean of the N_i data points of class Z_i,

$$m_i = \frac{1}{N_i}\sum_{z \in Z_i} z \quad (2)$$

Let

$$S_T = \sum_{z \in Z} \|z-m\|^2 \quad (3)$$

and

$$S_w = \sum_{i=1}^{C} S_i = \sum_{i=1}^{C} \sum_{z \in Z_i} \|z-m_i\|^2 \quad (4)$$

Define

$$J = \frac{S_T - S_w}{S_w} \quad (5)$$

Let

$$\bar{J} = \frac{1}{N}\sum_k M_k J_k \quad (6)$$

where J_k is J calculated over region k, M_k is the number of points in region k, N is the total number of points in the class-map.

Deng et al. propose J_k as the criterion to be minimized over all possible ways of segmenting the image given the number of regions. For a fixed number of regions, a better segmentation tends to have a lower value of J_k.

3.2 Algorithm

Fig. 2 and Fig. 3 show the flow chart of JSEG and Spatial Segmentation. Details are described in the following subsections.

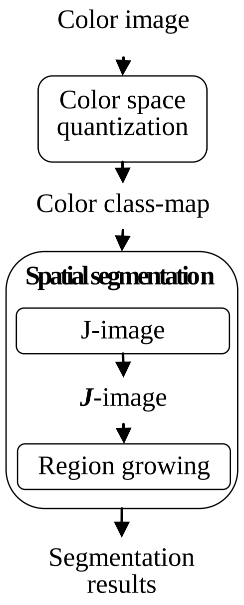

Fig. 2: JSEG

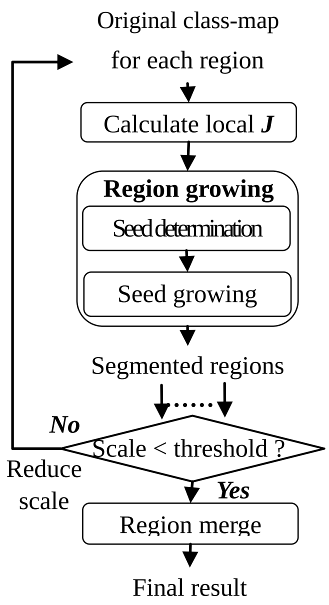

Fig. 3: Segmentation

3.2.1 Seed Determination

1. Calculate the average and the standard deviation of the local J value in the region, denoted by μ_J and σ_J

2. Set a threshold T_J at

   $$T_J = \mu_J + a\sigma_J$$

   where a is chosen from several preset values that will result in the most number of seeds. Pixels with local J values less then T_J are considered as candidate seed points. Connect the candidate seed points based on the 4-connectivity and obtain candidate seed areas.

3. If a candidate seed area has a size larger than the minimum size listed in Table 2 at the corresponding scale, it is determined to be a seed.

Table 2: Window size at different scale

scale	Window (pixels)	sampling (1/pixels)	region size (pixels)	min. seed (pixels)
1	9 x 9	1/ (1 x 1)	64 x 64	32
2	17 x 17	1/ (2 x 2)	128x128	128
3	33 x 33	1/ (4 x 4)	256x256	512
4	65 x 65	1/ (8 x 8)	512x512	2048

3.2.1 Seed Growing

1. Remove holes in the seeds.

2. Average the local J values in the remaining un-segmented part of the region and connect pixels below the average to form growing areas. If a growing area is adjacent to one and only one seed, it is assigned to that seed.

3. Calculate local J values for the remaining pixels at the next smaller scale to more accurately locate the boundaries. Repeat Step 2.

4. Grow the remaining pixels one by one at the smallest scale. Unclassified pixels at the seed boundaries are stored in a buffer. Each time, the pixel with the minimum local J value is assigned to its adjacent seed and the buffer is updated until all the pixels are classified.

3.2.3 Region Merge

1. Distances between the color histograms of any two neighboring regions are calculated and stored in a distance table.
2. The pair of regions with the minimum distance are merged together.
3. The color feature vector for the new region is calculated and the distance table is updated along with the neighboring relationships.
4. Repeat above process until a maximum threshold for the distance is reached.

4. COLORIZATION

4.1.1 YUV Color Space

Our method works in YUV color space, where Y is the luminance channel, U and V are the chrominance channels. We select the YUV color space because it provides a de-correlated achromatic channel for color painting. This allows us to selectively transfer the chromatic channels from the color paintings to gray paintings without cross-channel artifacts.

4.2 Luminance Remapping

In order to transfer chromaticity values from the source to the target, each pixel in the gray image should be matched to a pixel in the color image. The comparison is based on the luminance value and neighborhood statistics of that pixel. In order to account for global differences in luminance between these images we perform luminance remapping to shift and scale the luminance histogram of the source image to fit the histogram of the target image. This helps create a better correspondence in the luminance range between these images but does not alter the luminance values of the target image.

We apply a linear map that matches the means and variances of the luminance. Let $Y(p)$ is the luminance of a pixel *p* in an image A, then we remap it as

$$Y(p) \leftarrow \frac{\sigma_B}{\sigma_A}(Y(p) - \mu_A) + \mu_B \qquad (7)$$

where μ_A and μ_B are the mean luminances, and σ_A and σ_B are the standard deviation of the luminances, both taken with respect to luminance in image A and image B, respectively.

4.3 Luminance Clustering

The best match is determined by using a weighted average of pixel luminance and the neighborhood statistics when we determine each pixel in the gray painting should be matched to a pixel in the color painting. In order to reduce the number of compressions, we use k-means to cluster the luminance channel in the same region in color painting. We only compare the luminance and the neighbor statistics with those pixels which are clustered to a cluster with the most similar in a segmentation region when finding best match.

4.4 Gray Painting Pixels Classification

The statistics are computed over gray painting and color painting and consist of the standard deviation and the average of the luminance values of the pixel neighbor. Gradient is also considered. We have found that a neighborhood size of 3x3 pixels works well for most paintings. Then for each pixel in the gray painting the best match is determined by

$$\frac{1}{3}\left|\mu(p_{gray}) - \mu(p_{color})\right| + \frac{1}{3}\left|\sigma(p_{gray}) - \sigma(p_{color})\right| + \frac{1}{3}\left|G(p_{gray}) - G(p_{color})\right| \qquad (8)$$

where μ is the average of luminance, σ is the standard deviation of luminance, *G* is the sum of gradients at four direction (up, down, left, right) in 3x3 neighborhood. When we find the pixel which has the minimum value computed by (4), the chromaticity values (*u*, *v* channel) of the best match pixel are then transferred to the gray painting.

Unfortunately, there may be some pixels with smaller values computed by (4) which cause unsatisfied result. In order to solve this problem, we use the image pyramid.

The bigger neighbor region is, the more information we can get. It is helpful to colorize by using bigger neighbor. However, the bigger neighbor increases the computational cost. There has to be a trade-off between region size and computational cost. We use image pyramid to solve this problem. If we reduce the width and length to half of original painting, we can use the same neighbor size to get more information and increase not much computational cost. For an image pyramid with four levels, Level 0 is original painting. When we combine four pixels to form one pixel by computing the average of the four pixels, we can get Level 1.

Image pyramid also provides us an easy way to modify the painting. We can get a reference from coarser level. When the difference between Level *k* and Level *k+1* is too much, we can use the information at Level *k+1* to modify Level *k*. For example, the *u* channel value is 100 of one pixel at Level 1 and the corresponding four pixels' *u* channel values are 100, 100, 100, and 0. The average should be 100 ideally. However, the average is 75. We can find and modify this error easily by comparing Level 0 with coarser level. In our implementation, only Level 1, Level 2 and Level 3 are used.

4.5 Refinement

In order to allow user interaction in colorization and to improve results, our method allows users to define corresponding regions in the two paintings. This allows the user to selectively transfer colors between the source and target. This concept is similar to Welsh et al.'s Swatches. When users define corresponding regions in the two paintings, color transferring only works between the corresponding regions

5. IMPLEMENTATION RESULT

The implementation results of our proposed method are presented in this chapter. Our method is written in C# on the PC platform with an AMD Athlon 64 x2 Dual Core Processor 4400+ and 4 G RAM. Fig. 4~Fig. 6 show the results about Landscape, Fig. 7 shows about water fall. Fig. 8 shows the paint. Fig. 9 shows the result about creatures.

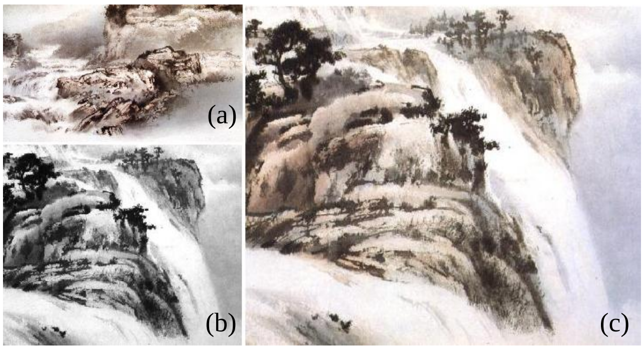

Fig. 4: (a) color reference, “The Clouds over the Clear Mountains”, 680x388 (b) target, “Wang Mountain”, 427x362 (c) the colorized result, 427x362. It takes

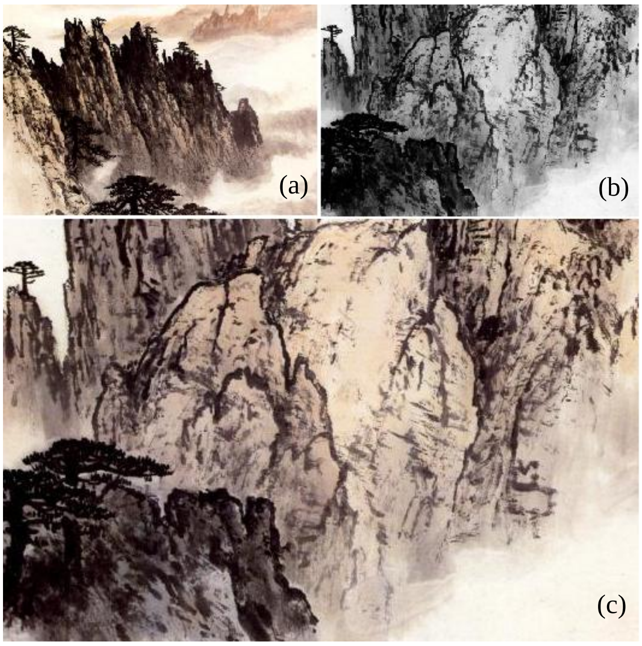

Fig. 5: (a) color reference, “Steep Cliffs”, 396x455 (b) target, “Stalagmite Cistern”, 532x352 (c) the colorized result. The processing time is about 5 minutes.

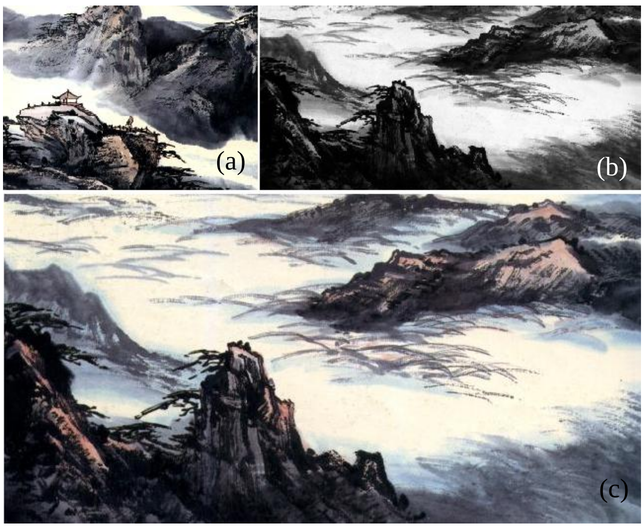

Fig. 6: (a) color reference, 590x460 (b) target, 647x334 (c) the colorized result. About 7 minutes in processing.

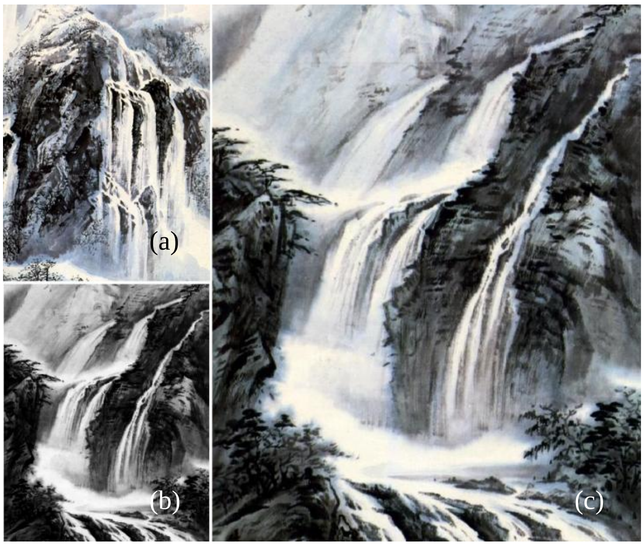

Fig. 7: (a) color reference, “Torrents Rushing by Rock”, 560x749 (b) target, “Cascade”, 555x699 (c) the colorized result. The processing time is about 9 minutes.

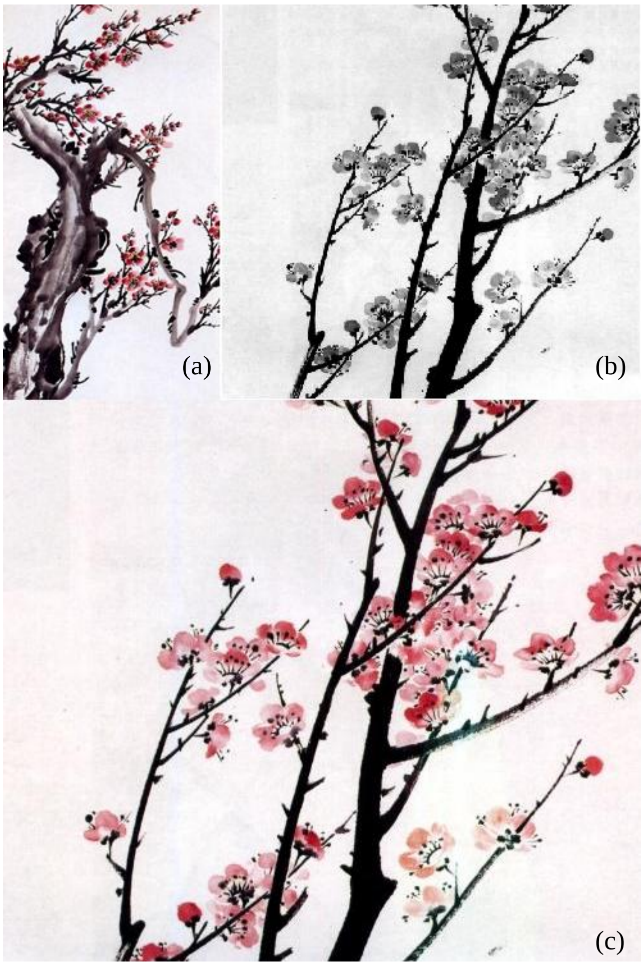

Fig. 8: (a) color reference (b) target (c) the colorized result

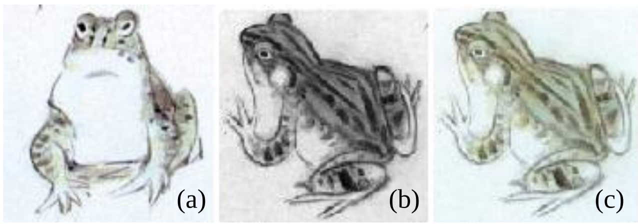

Fig. 9: (a) color reference (b) target (c) the colorized result

6. CONCLUSIONS AND FUTURE WORKS

6.1 Conclusions

In this paper, we propose a method for Chinese Ink Painting colorization. The proposed method is composed of three main steps: a segmentation algorithm, a classification, and interactive refinement.

The classification we proposed can produce good Chinese Ink Painting colorization results. By detecting the best match in a color painting, our method can produce a good result automatically. In order to allow user interaction in colorization and to improve results, we also provide refinement method.

6.2 Future Works

1. Although gray painting pixel classification we proposed can generate good results, but there is a problem cannot be solved: the input paints style should be similar. If the difference between two input paintings is too much, the results quality of results may be awful. In the future, we hope to improve the classification method by proposing a more efficient one to handle different painting style input.

2. The processing time depends on the size of input paintings. Although we use image pyramid and clustering to reduce the time of searching best match, the inputs with large size still take lots of time. This is an unavoidable problem. How to speed up is also a future work.

3. Current refinement step is very simple. We will provide more complex and helpful interface. For example, we can use matting technique when defining the region which users want to refine.

4. Much of colorization technique for realistic image also can use in video. We will try to make our method is also helpful for traditional animation like Japanese style animation.

7. REFERENCES

[1] Deng Y., Manjunath B. S., "Unsupervised Segmentation of Color-Texture Regions in Images and Video," IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 23, NO. 8, 2001

[2] REINHARD E., ASHIKHMIN M., GOOCH B.,SHIRLEY P., "Color transfer between images," IEEE Computer Graphics and Applications, September/ October, pp. 34–40, 2001

[3] WELSH T., ASHIKHMINM.,MUELLER K., "Transferring color to greyscale images," In ASM Transactions on Graphics , pp. 277–280, 2002

[4] IRONY R., COHEN-OR D., LISCHINSKI D., "Colorization by example," Eurographics Symposium on Rendering, pp. 277–280, 2005

[5] Chen T., Wang Y., Schillings V., Meinel C., "Grayscale image matting and colorization," Asian Conference on Computer Vision , 1164–1169, 2004

[6] Daniel S., Jan B., "Unsupervised Colorization of Black-and-White Cartoons," NPAR 2004

[7] LEVIN A., LISHINSKI D., WEISS Y., "Colorization using optimization," ACM Transactions on Graphics, pp. 689–694, 2004

[8] YATZIV L., SAPIRO G., "Fast image and video colorization using chrominance blending," IEEE Transaction on Image Processing, 2006

[9] Luan Q., Wen F., "Nature Image Colorization," Eurographics Symposium on Rendering, 2007

[10] Wang Z. M., You S. A., Huang M. L., "An Efficient Colorization Algorithm to Chinese Ink Paintings Using A Fuzzy Segmentation Technique," NCS, 2005

[11] KONUSHIN V., VEZHNEVETS V., "Interactive image colorization and recoloring based on coupled map lattices," Graphicon, pp. 231–234, 2006

[12] QU Y., WONG T., HENG P., "Manga colorization," ACM Transactions on Graphics , pp. 1214–1220, 2006"