

國立交通大學  
資訊科學與工程研究所  
碩士論文

一種創新的色鉛筆畫作風格轉換技術



A Novel Technique for Generating  
Colored Pencil Style Drawing

研究生：楊明貴

指導教授：施仁忠 教授

中華民國九十五年六月

一種創新的色鉛筆畫作風格轉換技術  
A Novel Technique for Generating  
Colored Pencil Style Drawing

研究生：楊明貴

Student : Ming-Kuei Yang

指導教授：施仁忠

Advisor : Zeng-Chun Shih

國立交通大學  
資訊科學與工程研究所  
碩士論文



Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年六月

# 一種創新的色鉛筆畫作風格轉換技術

研究生：楊明貴

指導教授：施仁忠 教授

國立交通大學資訊科學與工程研究所



色鉛筆的用途相當廣泛且無限制，舉凡插畫、漫畫、海報、風景寫生等，隨處可見以色鉛筆為媒材的藝術作品。市面上大多數軟體所提供的色鉛筆濾鏡功能並不完善，而且轉換後的效果和實際色鉛筆畫家的作品比較，有很明顯的差異存在。本篇論文提出一個嶄新且快速的色鉛筆濾鏡技術，可以將輸入的彩色照片，自動地轉換成為具有色鉛筆風格的畫作輸出。整個演算流程的主要想法與概念，並不是去猜測或是模擬畫家可能的繪畫筆觸走向，而是藉由觀察色鉛筆畫家的實際作畫過程，逐一模擬每個步驟並即時繪出，過程中透過一連串巧妙的影像處理技術，最後呈現出似色鉛筆質感的畫作。由於本論文演算過程並不複雜，相較於過去其他相關的色鉛筆模擬而言，運算速度可說是十分快速。

# A Novel Technique for Generating Colored Pencil Style Drawing

Student: Ming-Kuei Yang

Advisor: Dr. Zen-Chung Shih

Institute of Computer Science and Engineering  
National Chiao Tung University



## ABSTRACT

Colored Pencils have broad applications. For example, iconographs, caricatures, posters, and sketches are the artworks drawn by colored pencils. However, most commercial image processing software does not provide perfect filter for colored pencil style drawing. In addition, there is always a big difference between the final results transformed by computers and the actual colored pencil drawing created by artists. This paper proposes a novel and fast filter for colored pencil style drawing. The filter can automatically transform input color photos into colored pencil style drawings. The main idea and concept of the algorithm is not to guess or to imitate possible stroke directions which artists may draw. It is to observe the actual artists' whole drawing process and then to simulate each drawing step with real-time rendering. Through a series of clever image processes, the system finally presents excellent colored pencil style drawings. Because the proposed algorithm is not complicated, the rendering time is quite short compared to other past related studies.

# Acknowledgement

First of all, I would like to thank to my advisor, Dr. Zen-Chung Shih, for his help and supervision in this work. Also, I thank for all the members of Computer Graphics & Virtual Reality Lab for their comments and instructions. Finally, special thanks go to my family and my dear friends, and the achievement of this work dedicated to them.



# Contents

ABSTRACT (IN CHINESE).....	I
ABSTRACT (IN ENGLISH).....	II
ACKNOWLEDGEMENTS.....	III
CONTENTS.....	IV
LIST OF FIGURES.....	V
CHAPTER 1 INTRODUCTION.....	1
1.1 Motivation.....	1
1.2 Introduction to Colored Pencils.....	2
1.3 System Overview.....	3
CHAPTER 2 RELATED WORKS.....	5
2.1 Drawing Strokes.....	5
2.2 Colored Pencil Drawing.....	5
CHAPTER 3 ALGORITHM OF COLORED PENCIL FILTER.....	7
3.1 Paper Model.....	7
3.2 Outline Drawing.....	9
3.3 Color Transformation.....	10
3.4 Pigment Distribution of Colored Pencils.....	12
CHAPTER 4 IMPLEMENTATION AND RESULTS.....	16
CHAPTER 5 CONCLUSIONS AND FUTURE WORKS.....	25
REFERENCES.....	26

# List of Figures

Figure 1 The proposed framework.....	4
Figure 2 Texture images.....	7
Figure 3 The paper model .....	8
Figure 4 The Sobel Operator.....	9
Figure 5 The HSI color model.....	11
Figure 6 The directions of paper fibers.....	13
Figure 7 The 5x5 pigment mask.....	14
Figure 8 The 7x7 pigment mask.....	14
Figure 9 The enlarged texture example.....	15
Figure 10 The transforming procedure.....	18
Figure 11 The input images for test.....	19
Figure 12 The rendered results.....	20
Figure 13 The comparison with Adobe PhotoShop.....	23
Figure 14 The comparison with Matsui et al. [2].....	24

# Chapter 1

## Introduction

### 1.1 Motivation

Colored pencils are easily-acquired drawing materials. The variety of colored pencils makes it possible to apply broadly to drawing, iconograph, and advertisement. In the past, outstanding artists such as Henri de Toulouse-Lautrec, Pablo Picasso, and Henri Matisse handled and used colored pencils with great skills. Iconographers and commercial designers also could create high quality artworks by using such materials. However, most people ignore colored pencils and regard it as the childish drawing materials without ample understanding of the variety of colored pencils. For the last several years, colored pencil style drawing has been brought to public attention and the interest in such materials has also been growing.

Accordingly, several commercial image editing software provided users with drawing strokes of colored pencil. For those who are not familiar with drawing skills of colored pencils, it is still quite difficult for them to create fine results. Although some software has extra colored pencil filter functions, it transforms input photos into imperfect images. Compared with the actual colored pencil drawings, those output images seem incorrect in some respects.

Furthermore, past researches related to colored pencils often focus their attention on one feature, such as the direction of drawing strokes, the blending of custom colors, or the physical model simulation of drawing papers, etc. Regardless of the complete



picture of actual colored pencil drawings, these studies presented improper output results. Unfortunately, the rendering time cost very much as well.

The purpose of this study is to accomplish a better filter for colored pencil style drawing. In this article, I would like to imitate each drawing step in computer by observing the actual artists' drawing process. Every drawing step will show in real-time and the intuitive user interface is easy for users to adjust the related parameters. The procedure of the algorithm comprises a sequence of clever image processing methods. On the whole, this study contributes short rendering time and persuasive transformation for colored pencil style drawings.

## **1.2 Introduction to Colored Pencils**

It is essential to clearly understand both the character of colored pencil pigments and the drawing style of colored pencil artworks [10] before trying to simulate colored pencils in computer. Owing to the softness of colored pencil pigments, pigments would stick to papers when colored pencils are drawn over the papers. The roughness of papers affects the amount of pigments scraped from colored pencils. The rougher the papers are, the easier the pigments stick to papers. Therefore, different kinds of papers are selected to represent particular style of colored pencil artworks.

Another important skill of colored pencils is “mixing”, that is “blending.” Generally speaking, a box of colored pencils consists of at least twenty-four custom colors. Of course there are other single color ones individually provided for artists' special needs. Nevertheless, artists could still create various kinds of colors with limited colored pencils.

More accurately, “mixing” is a scheme which creates a new color by blending with two (or more) different colors. Despite the same colors, artists could create a

distinct one when the order of use of colored pencils is different. To cite an example, if yellow is the first pigment sticking to papers and red is the second one, the blended color would be reddish orange. On the contrary, if red is the first pigment and yellow is the second one, the blended color would be orange.

Additionally, the pencil pressure is also an important concern when artists using colored pencils. The more pencil pressure artists put on the papers, the deeper the intensity of colors presents.

### **1.3 System Overview**

As noted above, the main idea of our system design is quite intuitive. Based on actual artists' drawing process, all controlling steps in user interface are similar and friendly to follow. Firstly, users have to input a color photo they want to transform. Secondly, users should choose one texture, vertical or horizontal, from two default paper models. Third, our system would automatically create rough outline sketches according to the input photos. What users have to do is adjust several parameters duly. These parameters determine both the complexity and the intensity of the outline sketch. After finishing the outline sketch, artists usually paint light ground colors on it. The painted ground colors will vary depending on the original colors of photos. Certainly users can also decide the thickness of ground colors by adjusting related parameters. At last, system will intelligently sprinkle colored pencil pigments on the processed picture. To output a drawing with smooth or rough quality, or with different brightness, users can again modify few specific parameters. The final colored pencil style drawing hence appears. The following flowchart illustrates the whole framework of our system:

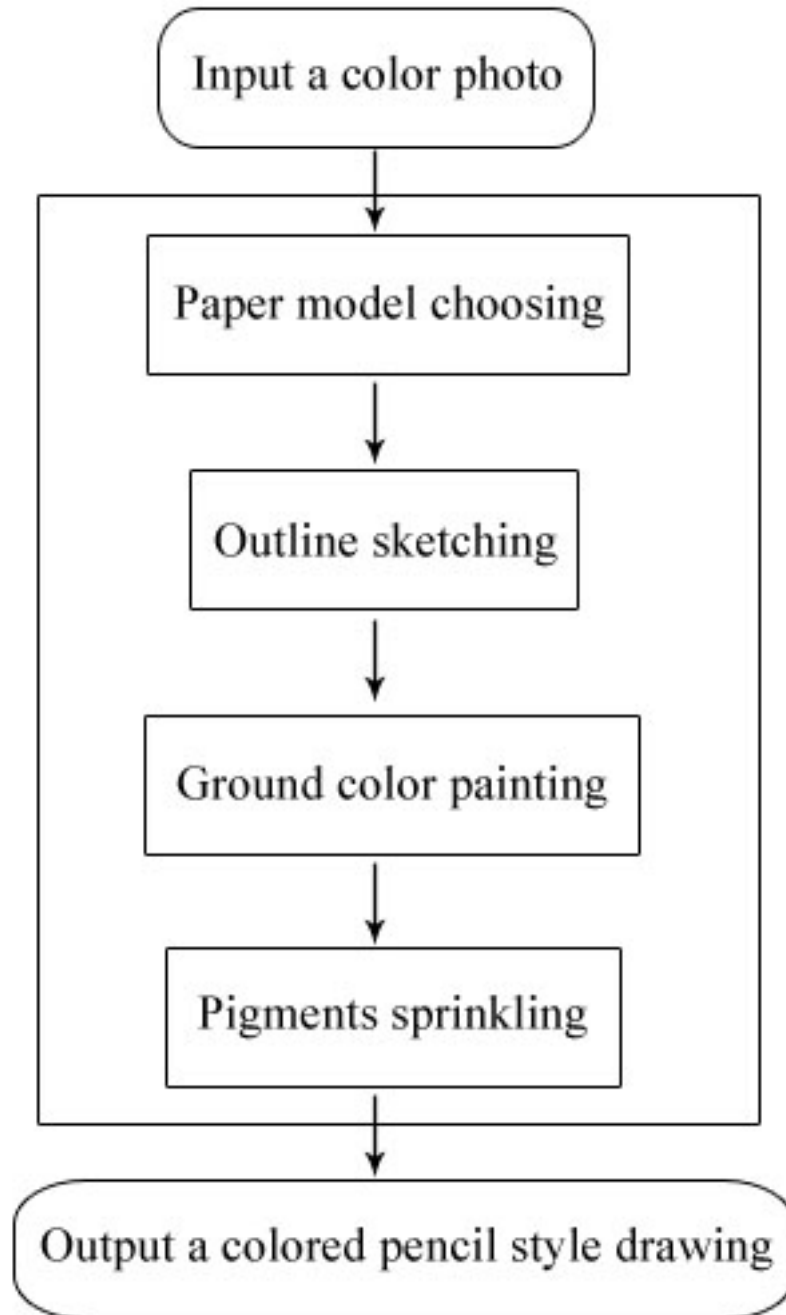


Figure 1 The proposed framework.

In the remainder of this thesis, we first review the related works in Chapter 2. Chapter 3 describes in detail the algorithm of the proposed technique. Chapter 4 shows the implemented results and Chapter 5 concludes this paper with a discussion on the limitation of our approach and possible further extensions.

# Chapter 2

## Related Works

### 2.1 Drawing Strokes

Colored pencil style drawing is entirely different from pencil sketching. Most people are incorrect in thinking that colored pencil style drawing is equivalent to pencil sketching with colors. To put it precisely, pencil sketching is monochromatic. Artists use only the black color to express a wide range of image intensity- from white, gray, to black. Moreover, in order to show different color areas or to emphasize the special texture of objects, artists usually create various kinds of drawing strokes. This drawing stroke is exactly one of the notable features of pencil sketching but not of colored pencil drawing. Researchers [6, 9] have focused on the automatic pencil sketching transformation of 2D resource images on the basis of Line Integral Convolution (LIC) Pencil Filter. Yamamoto et al. [7] further expanded the same LIC technique into the simulation of colored pencil style drawings. However, the generated drawing strokes are unnatural and rigid in direction. Murakami et al. [4] generated nice realistic drawing strokes by using twelve illuminated paper surface samples. Unfortunately, the effect of overlapping strokes seemed imperfect.

### 2.2 Colored Pencil Drawing

Very few attempts have been made at the study of colored pencil drawings. Those researches often tended to concentrate on one individual issue, such as the

blending effect of custom colors [7], the physical model simulation of drawing papers [8], or the direction of drawing strokes [2], without completely considering the picture of actual colored pencil drawings. Although Yamamoto et al. [7] introduced a new idea of color customization, expressing a target color by mixing two colors, the final output result was still deficient in color blending. Based on the physical laws, Takagi et al. [8] performed volumetric renderings which included the different effects of papers, drawing, watering, and erasing. Nevertheless, these rendering effects did not fully resemble the actual colored pencil drawings. In 2005, Matsui et al. [2] combined the concepts stated above and put more emphases on the creation of drawing strokes based on boundaries of regions. This thesis indeed presented much better results compared to those past related researches. However, for those images which contain complex information, the creation of drawing strokes becomes difficult to handle. Beyond that, some commercial software, such as Photoshop [11], also has function which converts images into colored pencil style drawings. Frankly speaking, those who have ever used this software knew that the filter for colored pencil style drawing is incorrect.

# Chapter 3

## Algorithm of Colored Pencil Filter

### 3.1 Paper Model

We will now examine more closely the model of drawing papers. As for actual colored pencil drawings, the texture of drawing papers affects the style of colored pencil artworks. To put it more concretely, pigments are shaved off from the colored pencils by a friction on a convex part of the drawing papers. Therefore, the roughness of drawing papers results in different deposit amount of pigments. The rougher the drawing papers are, the weaker possibility the pigments deposit evenly.

In our simulation method, the drawing paper textures (see Figure 2) were directly downloaded from the internet [12]. Only two textures commonly used by artists are selected in our system. The width and height of these texture images are both 64 pixels. We adjust the size of the texture image according to each input photo. The intensity values of these texture images are also calculated and then used as the deposit efficiency of the pigments. The main concept is to determine the thickness of pigments at each pixel based on the intensities of both the input photo and the texture image. We will later discuss in full detail the pigments distribution in Section 3.4.



Figure 2 Texture images.

Now, let us get down to the issue of paper model. As mentioned above, our system provides two default texture images as paper models for users. Let  $(x_{pi}, y_{pi})$  be the coordinate of a pixel  $i$  on the paper model  $p$ , and  $(x_{ti}, y_{ti})$  be the coordinate of a pixel  $i$  on the texture image  $t$ . We compute each channel of red, green, and blue respectively as follows:

$$R(x_{pi} \% 64, y_{pi} \% 64) = R(x_{ti}, y_{ti}) + Ep \quad (1)$$

$$G(x_{pi} \% 64, y_{pi} \% 64) = G(x_{ti}, y_{ti}) + Ep \quad (2)$$

$$B(x_{pi} \% 64, y_{pi} \% 64) = B(x_{ti}, y_{ti}) + Ep \quad (3)$$

In our experiments, by adding an extra parameter  $Ep = 45$ , we could represent the actual paper color more closely. The example of the final paper model is shown in

Figure 3.

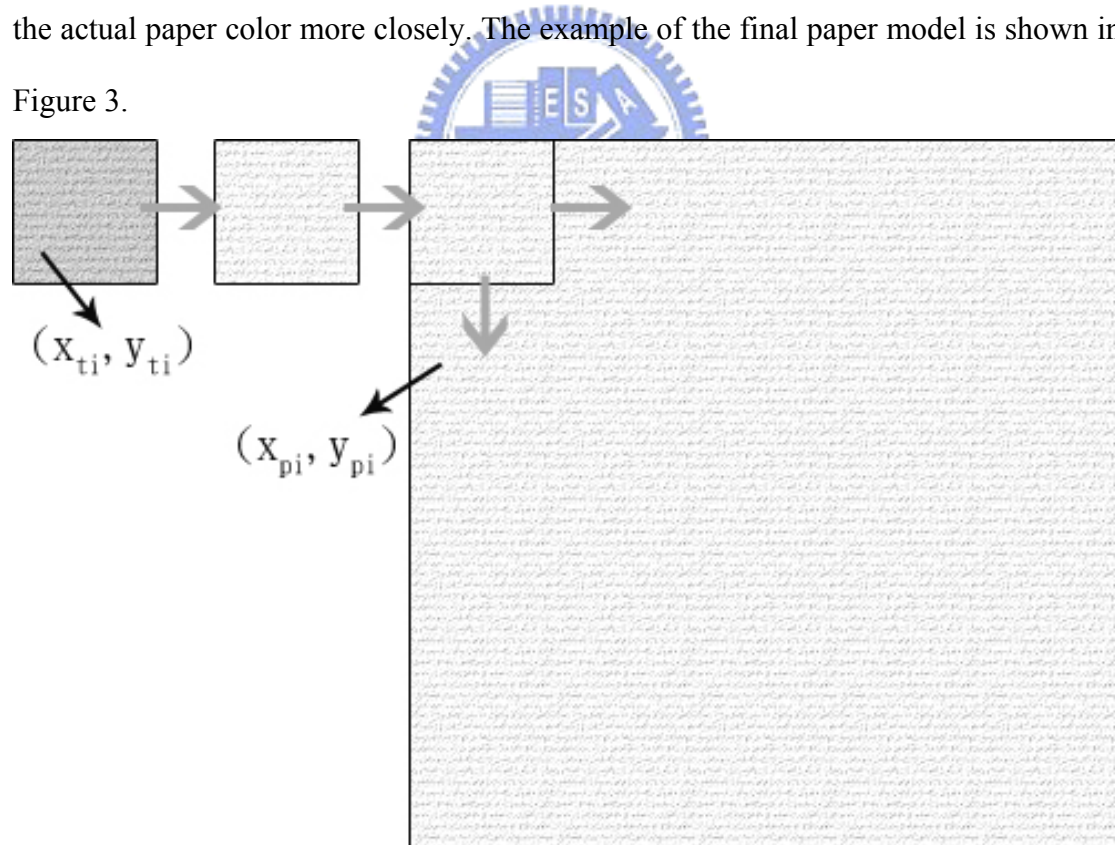


Figure 3 The paper model.

## 3.2 Outline Drawing

The next issue which we have to consider is how to draw outline sketches. Actual artists used to draw rough outline sketches in the beginning of a drawing process. Thus, we introduce the Sobel edge detection technique [1] as our simulation method for outline drawing.

$Z_1$	$Z_2$	$Z_3$
$Z_4$	$Z_5$	$Z_6$
$Z_7$	$Z_8$	$Z_9$

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Figure 4 The Sobel Operator.

The masks shown in Figure 4 are called Sobel operators. These operators are used to compute the gradient value of a 3×3 window at the point labeled  $Z_5$ , as follows:

$$G_x = (z_7 + 2 \times z_8 + z_9) - (z_1 + 2 \times z_2 + z_3) \quad (4)$$

$$G_y = (z_3 + 2 \times z_6 + z_9) - (z_1 + 2 \times z_4 + z_7) \quad (5)$$

$$\nabla f = |G_x| + |G_y| \quad (6)$$

Equation 4 computes the difference of the intensity value between the third and the first rows of a 3×3 window.  $G_x$  then represents an approximation of the derivative in the x-direction. Similarly,  $G_y$  is the difference between the third and the first columns which approximates the derivative in the y-direction.  $\nabla f$  is the sum of both the absolute value of  $G_x$  and the absolute value of  $G_y$ . There is a strong possibility that the point labeled  $Z_5$  is a part of our outline sketches when the  $\nabla f$  value is large enough.



To be precise, a threshold value  $T_e$  is used to indicate the complexity of our final outline sketches. The default value of  $T_e$  is half the maximum intensity of the whole image. By our experiment, the maximum of the intensity value would not exceed 441.67275. Therefore, we set our  $T_e$  value to be 221. In other words, when the calculated  $\nabla f$  value exceeds 221, we take that point as one part of our outline sketches. Let us see the following operation for further explanation. Another parameter  $U_c$  is adopted especially for users. Users can determine the complexity of the final outline sketches by adjusting the value of  $U_c$ .

$$T_e = \{221 + (U_c - 10) \times 10, 0 \leq U_c \leq 20\} \quad (7)$$

Furthermore, we offer an extra calculation (see Equation 8) to control the intensity of the final outline sketches. Let  $I(x_i, y_i)$  be the intensity value of any pixel  $i$  in the image. Once a point  $(x, y)$  has already been detected as a part of the outline sketch, users could again change its intensity value by increasing or decreasing the  $I_p$  value. The smaller the  $I_p$  value is, the darker the intensity of the final outline sketch appears.

$$I(x_i, y_i) = I(x_i, y_i) + I_p \times 8, I_p \in Z \quad (8)$$

### 3.3 Color Transformation

How to transform an original color photo into a colored pencil style image is the main issue of this thesis. Color transformation is one of the key factors that determine whether the final rendering result resembles the actual colored pencil style drawing or not. Compared with the colors of a true color photo, the colors of an actual colored pencil style drawing are limited in hue and the saturation is relatively low.

Therefore, we choose the HSI color model, as shown in Figure 5, which closely matches the experience of human visual perception as our reference model. HSI

stands for hue, saturation, and intensity respectively. Hue is a color attribute that describes different pure colors such as red, yellow, green, and blue, etc. Saturation is the measure of the proportion of one pure color mixing with white color. For example, pink has less saturation than red. Intensity represents the brightness of an image.

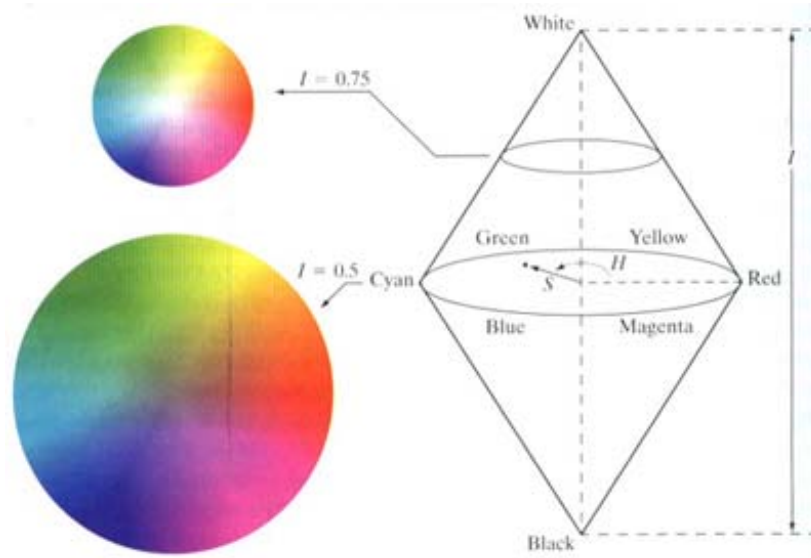


Figure 5 The HSI color model.

After we obtain the RGB values of each pixel from an input color photo, we start to transform these values into the HSI color space. The conversion from the RGB color space to the HSI coordinate system refers to the book of Fundamentals of Color Image Processing [3]. First of all, we rotate the RGB coordinate to form a new coordinate system  $(I, V_1, V_2)$  whose axis is the line  $R=G=B$ . The following linear transformation defines this rotation.

$$\begin{bmatrix} I \\ V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \\ 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{2}{\sqrt{6}} & \frac{-1}{\sqrt{6}} & \frac{-1}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (9)$$

Next, we continue to transform the rectangular coordinates  $(V_1, V_2)$  to the polar coordinates, as shown in Equations 10 and 11. Incidentally, the range of hue

coordinate is from 0 degree to  $2\pi$  degree.

$$H = \tan^{-1}\left(\frac{V_2}{V_1}\right) \quad (10)$$

$$S = \sqrt{V_1^2 + V_2^2} \quad (11)$$

Finally, the following procedure performs the inverse transformation from the HIS color space to the RGB color space.

$$V_1 = S \cos H \quad (12)$$

$$V_2 = S \sin H \quad (13)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{3} & 0 & \frac{2}{\sqrt{6}} \\ \frac{\sqrt{3}}{3} & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{6}} \\ \frac{\sqrt{3}}{3} & \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} I \\ V_1 \\ V_2 \end{bmatrix} \quad (14)$$

### 3.4 Pigment Distribution of Colored Pencils

Now we will discuss the pigment distribution of colored pencils in detail. As we mentioned at the beginning, we do not concern with neither the styles nor the directions of drawing strokes artists may create. Owing to each artist's unique characteristic of drawing, it is unreasonable and not objective to guess the artist's possible drawing strokes. In addition, the drawing stroke is not one of the most important features of colored pencil style drawing. Hence, we concentrate on how pigments adhere to the surface of papers.

First, let us consider the complexity of paper fibers. Instead of carefully simulating the microstructure of papers based on physics, we use a combination of four main directions to represent the complexity of paper fibers. How paper models affect the general deposit efficiency of the pigments has been discussed in Section 3.1.

Here we take a close look at the relationship between the pigments and the paper fibers. The deposit orientations of the pigments would vary because of different directions of paper fibers. Thus, we classify these paper fibers into four main directions including (a) horizontal, (b) up-right diagonal, (c) vertical, and (d) up-left diagonal as shown in Figure 6.

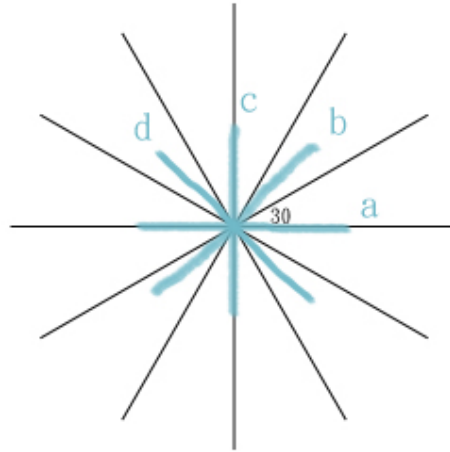


Figure 6 The directions of paper fibers.

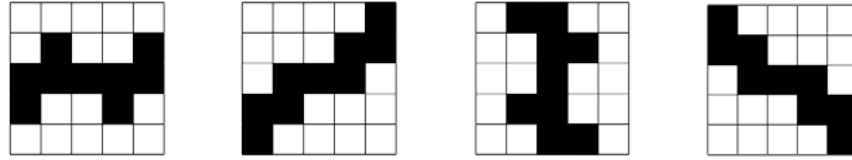
Secondly, let  $P_s$  be a gradient value of a pixel in an image. Equation 15 defines  $P_s$  as the ratio of  $G_y$  and  $G_x$ .  $G_y$  and  $G_x$  are calculated according to the Sobel edge detection technique as mentioned in Section 3.2. Equation 16 further describes how we select the deposit orientation of the pigments.

$$P_s = \frac{G_y}{G_x} \quad (15)$$

$$orientation = \begin{cases} (a) \text{ horizontal} & \text{if } \frac{-1}{\sqrt{3}} \leq P_s < \frac{1}{\sqrt{3}} \\ (b) \text{ up-right diagonal} & \text{if } \frac{1}{\sqrt{3}} \leq P_s < \sqrt{3} \\ (c) \text{ vertical} & \text{if } \sqrt{3} \leq P_s \text{ or } P_s < -\sqrt{3} \\ (d) \text{ up-left diagonal} & \text{if } -\sqrt{3} \leq P_s < \frac{-1}{\sqrt{3}} \end{cases} \quad (16)$$

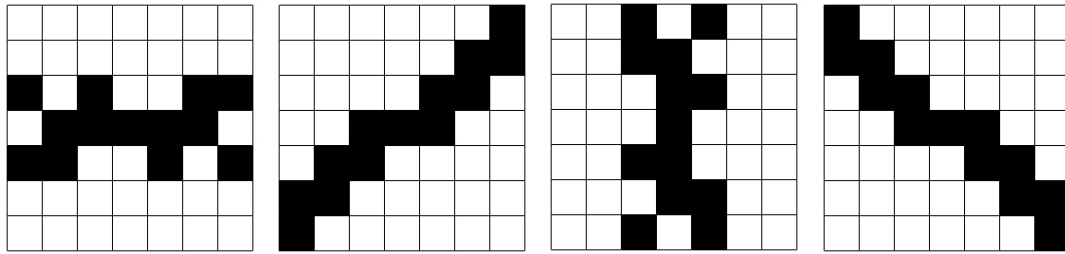
Thirdly, we create two window sizes of pigment masks, 5x5 and 7x7, to sprinkle pigments on each pixel in an image. According to our experiment, these two window

sizes are most suitable for the final rendering results. We then emphasize the intensity values of the pixels included in the dark areas, whereas those intensity values in the white areas are unchanged, as illustrated in Figures 7 and 8.



(a) horizontal (b) up-right diagonal (c) vertical (d) up-left diagonal

Figure 7 The 5×5 pigment mask.



(a) horizontal (b) up-right diagonal (c) vertical (d) up-left diagonal

Figure 8 The 7×7 pigment mask.

So how do we choose a 5×5 or 7×7 pigment mask for a pixel in an image? Again we use those values that are already calculated according to the Sobel edge detection technique. Let  $G_s$  be the sum of both the square value of  $G_x$  and the square value of  $G_y$ , and let  $G_{s\_avg}$  be the average value of the sum of  $G_s$ , as shown in Equations 20 and 21. For any pixel  $(m, n)$ , if the value of  $G_{s(m,n)}$  is larger than the average value  $G_{s\_avg}$ , we apply the 7×7 pigment mask. On the contrary, the 5×5 pigment mask would be selected.

$$G_{s(m,n)} = G_{x(m,n)}^2 + G_{y(m,n)}^2 \quad (20)$$

$$G_{s\_avg} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} G_s(m,n) \quad (21)$$

Finally, in order to stress the same powdered quality as the actual colored pencil

style drawing, we observe several actual artworks of colored pencil style drawing. We cut a piece of image from the artwork and then enlarge this piece, as shown in Figure 9. After carefully examining the distinct intensity shown in each pixel, we make a general conclusion. As a whole, there is a 4% probability that the color of the pixel in an image is close to white. Therefore, for each one pixel included in a  $5 \times 5$  square area of an image, we increase its intensity value twice. Beyond that, we randomly select five pixels included in a  $5 \times 5$  square area and increase their intensity 1.2 or 1.5 times as much as the original values. For the intensity of other remaining pixels, we adjust their values in a limited range at random.

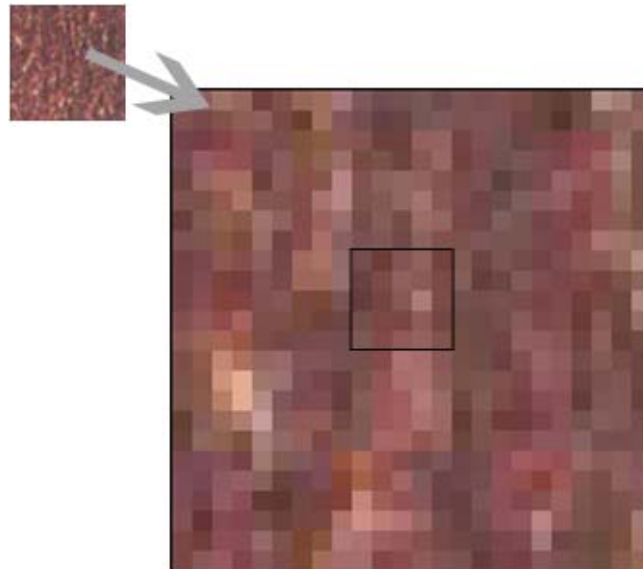


Figure 9 The enlarged texture example.

# Chapter 4

## Implementation and Results

All of our experiments were performed using a Pentium 1.50 GHz, 768 MB RAM in real time. We have implemented all the techniques described above as a Window application using C#. A colored pencil style drawing is automatically generated with a set of default values and other information derived from the input images. Users are allowed to change the final results by interactively adjusting the controlling parameters in each step of the transforming procedure.

Figure 10 illustrates the complete procedure of our transformation for an input image. First, users specify an input photo and then choose a proper paper model. Second, our system creates initial outline sketches like artists' hand-drawings. In this step, we offer users two parameters, both intensity and complexity of sketches, to modify the temporary rendering results. Next, we simulate the step of ground color painting because artists usually draw a layer of light ground color on papers after outline sketching. Again a parameter, thickness of color, is provided for users to determine the darkness or lightness of output images. Finally, our system automatically sprinkles the pigments of colored pencils and hence renders a colored pencil style drawing.

Eventually, let us take a look at our final rendered results. All input images for test, includes general scenic photos and still objects, are shown in Figure 11. Although those scenic photos have very complicated information, we still generate perfect drawings, as shown in Figure 12 (a, b, and c). For those simple images with

single still objects, we also render very good results, as shown in Figure 12 (d and e).

Figure 13 illustrates the comparison with Adobe Photoshop, the famous commercial image editing software. Obviously, those results transformed by Adobe Photoshop have unnecessary fractures and are incorrect in colors. In contrast, our rendered results indeed appear better quality of colored pencil style drawings.

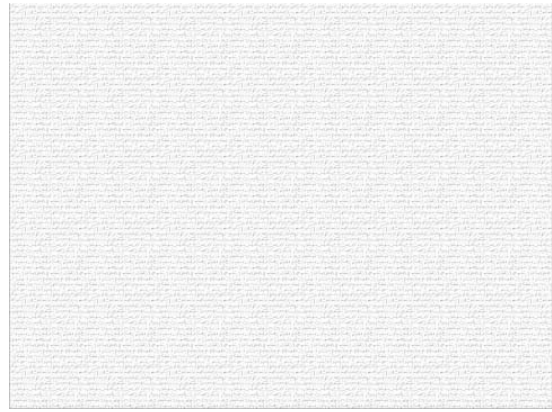
Furthermore, we compare our technique with the method Matsui et al. [2] proposed in 2005. As shown in Figure 14, Matsui et al. [2] create good drawing strokes based on boundaries of regions. However, for complicated objects in images, they fail in generating detailed drawing strokes. As a whole, our rendered results represent a better perception of colored pencil style drawing rather than emphasize the dispensable drawing strokes.



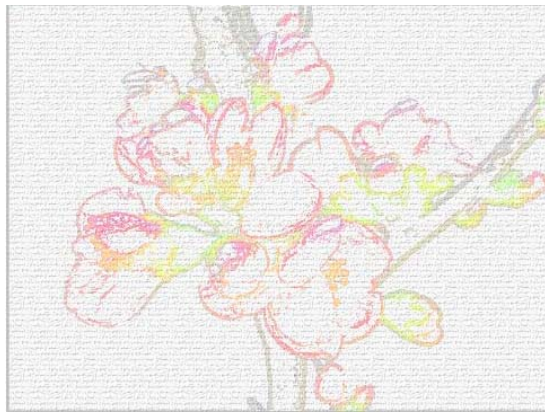




(a)



(b)



(c)



(d)



(e)

Figure 10 The transforming procedure.

(a) input image, (b) paper model, (c) outline sketching, (d) ground coloring, (e) result



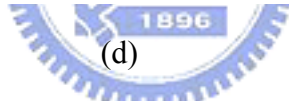
(a)



(b)



(c)



(d)



(e)

Figure 11 The input images for test.

(a)(b)(c) scenic photos, (d)(e) still objects



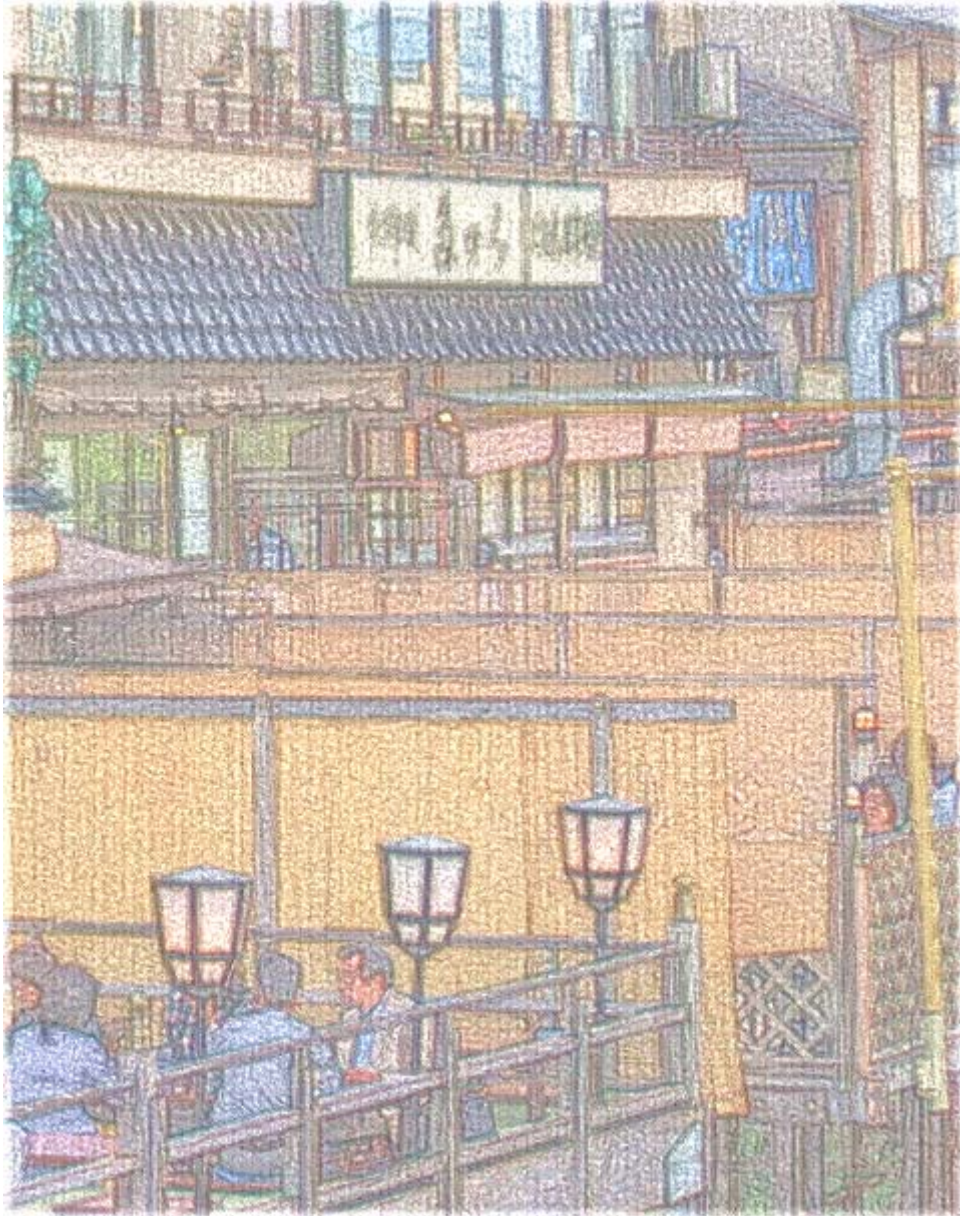


(a)

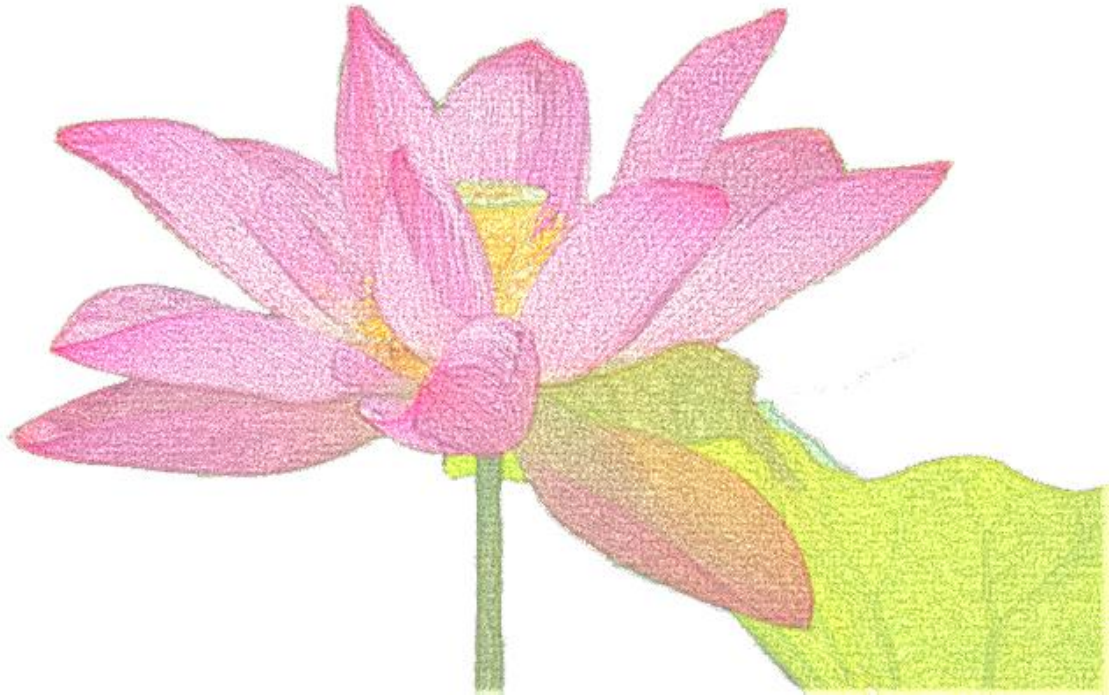


(b)





(c)



(d)



(e)

Figure 12 The rendered results.

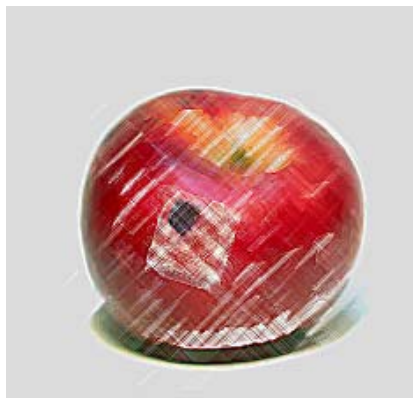




(a)



(b)



(c)



(d)



(e)



(f)

Figure 13 The comparison with Adobe PhotoShop.

(a)(b) input images, (c)(d) Photosop, (e)(f) our proposed method



(a)



(b)



(c)



(d)



(e)



(f)

Figure 14 The comparison with Matsui et al. [2].

(a)(b) input images, (c)(d) Matsui et al. [2], (e)(f) our proposed method

# Chapter 5

## Conclusions and Future Works

On the basis of simulating the actual artist's drawing process, we proposed a novel technique for generating a colored pencil style drawing from an input image. The process of our proposed method consists of paper model choosing, outline sketching, ground color painting, and the final colored pencil pigments sprinkling. Our proposed algorithm is not complicated and the rendering time is almost in real time. Instead of guessing or imitating the possible drawing strokes of different artists, we choose to present the characteristic of the colored pencil pigment itself through a series of clever image processes. To sum up, our final rendering results show a very good quality of colored pencil style drawing, and are even better than the existing commercial image editing software, Adobe Photoshop.

As for the future work, we aim to add an extra interactive function for users to determine the subject drawing strokes in the final rendering results. Moreover, we also want to extend our method for converting a video sequence into a colored pencil style animation owing to our fast rendering speed.



# References

- [1] Gonzalez, R.C., and P. Wintz, Digital Image Processing, Addison-Wesley Publishers, Inc., USA, 1992.
- [2] Hajime Matsui, Henry Johan, Tomoyuki Nishita, “Creating Colored Pencil Style Images by Drawing Strokes Based on Boundaries of Regions”, *Computer Graphics International 2005*, 2005.
- [3] I. Pitas, Fundamentals of Color Image Processing, John Wiley & Sons, Inc., USA, 2000.
- [4] Kyoko Murakami, Reiji Tsuruno, and Etsuo Genda, “Multiple Illuminated Paper Texture for Drawing Strokes”, *Computer Graphics International 2005*, 2005.
- [5] Sousa M. C. and Buchanan J. W., “Observational Models of Graphite Pencil Materials”, *Computer Graphics forum Volume 19* (2000), number 1 pp. 27-49.
- [6] Shigefumi Yamamoto, Xiaoyang Mao, Kenji Tanii, Atsumi Imamiya, “Enhanced LIC Pencil Filter”, *Proceedings of International Conference on Computer Graphics, Imaging and Visualization*, pp.251-256, IEEE, July 2004.
- [7] Shigefumi Yamamoto, Xiaoyang Mao, Atsumi Imamiya, “Colored Pencil Filter with Custom Colors”, *Proceedings of Pacific Graphics 2004*, pp. 329-338, October 2004.
- [8] Takagi S., Fujishiro I., and Nakajima M., “Volumetric Modeling of Colored Pencil Drawing”, *Pacific Graphics '99 conference proceedings*, 1999.
- [9] Xiaoyang Mao, Yoshiyasu Nagasaka and Atsumi Imamiya, "Automatic Generation of Pencil Drawing from 2D Images Using Line Integral Convolution", *Proceedings of the 7th International Conference on Computer Aided Design and Computer Graphics CAD/GRAPHICS2001*, pp.240-248, August 2001.
- [10] 張瓊文，「第一次畫色鉛筆就上手入門篇」，易博士文化，(民國 94 年)。
- [11] Adobe System. Adobe Photoshop.
- [12] [http://fweb.midi.co.jp/~buru\\_nyan/download](http://fweb.midi.co.jp/~buru_nyan/download)