CHAPTER III

COLOR IMAGE GENERATION

A color image imported via a b/w image input device can be generated by using either color mixing or pseudo coloring methods. These two color image generation methods will be used when the source images are either color or black and white, respectively, and they will be described in details in this chapter.

Color Image Generation by Color Mixing

When a source color picture is captured with a b/w image input device, the output gray scale image will contain different intensities of brightness ranging from darkest (close to black) to lightest (close to white) with different shades of gray in between. If this procedure is performed three times and each time a red, green, or blue filter is applied, three images of different shades of gray will be obtained. As red, green, and blue of different intensities can be mixed to yield a wide range of colors, mixing these three gray scale images interpreted as intensities of the three primary colors will result in an output color image. However, this concept of color mixing can be put into practice only when a number of basic procedures are designed and determined. Such procedures include:

- choosing appropriate color filters and detecting their light reduction characteristics,
- calibrating gray levels into the brightness of the three primary colors of the same bases,
- approximating the intensities of each color in the source color picture,
- building up a color look-up table (LUT), and
- choosing the closest colors for the output color image.

The procedure required to generate an output color image is illustrated in Fig. 3.1 and summarized as follows.

1. Capture four images of the source color picture: one without any filter, and three applied with each of RGB filters.* The automatic exposure feature of the image input software, if available, must be exploited each time the picture is captured. Jot down the brightness and contrast values applied to each image.
2. Capture four images of calibration card: one without any filter, and three applied with each of the RGB filters and exploiting the same brightness and contrast values from (1).
3. Put the four images of calibration card from (2) into the *Filter Characteristic Detecting Process* and obtain the *Filter Characteristic File.*
4. Put the three images from (1) and the Filter Characteristic File from (3) into the Color Mixing Process and obtain the output color image.

It should be noted that the exposure should be adjusted to a proper setting. From the experiments of 20 examples, if the automatic exposure feature is not available, the exposure should be set to high brightness and high contrast. Such exposure setting usually results in a highly efficient use of the gray level range while maintaining the highlights and shadows of the picture.

---

* To get the best result, special care must be taken to ensure the image is captured exactly in the same position and resolution each time the filter is changed.

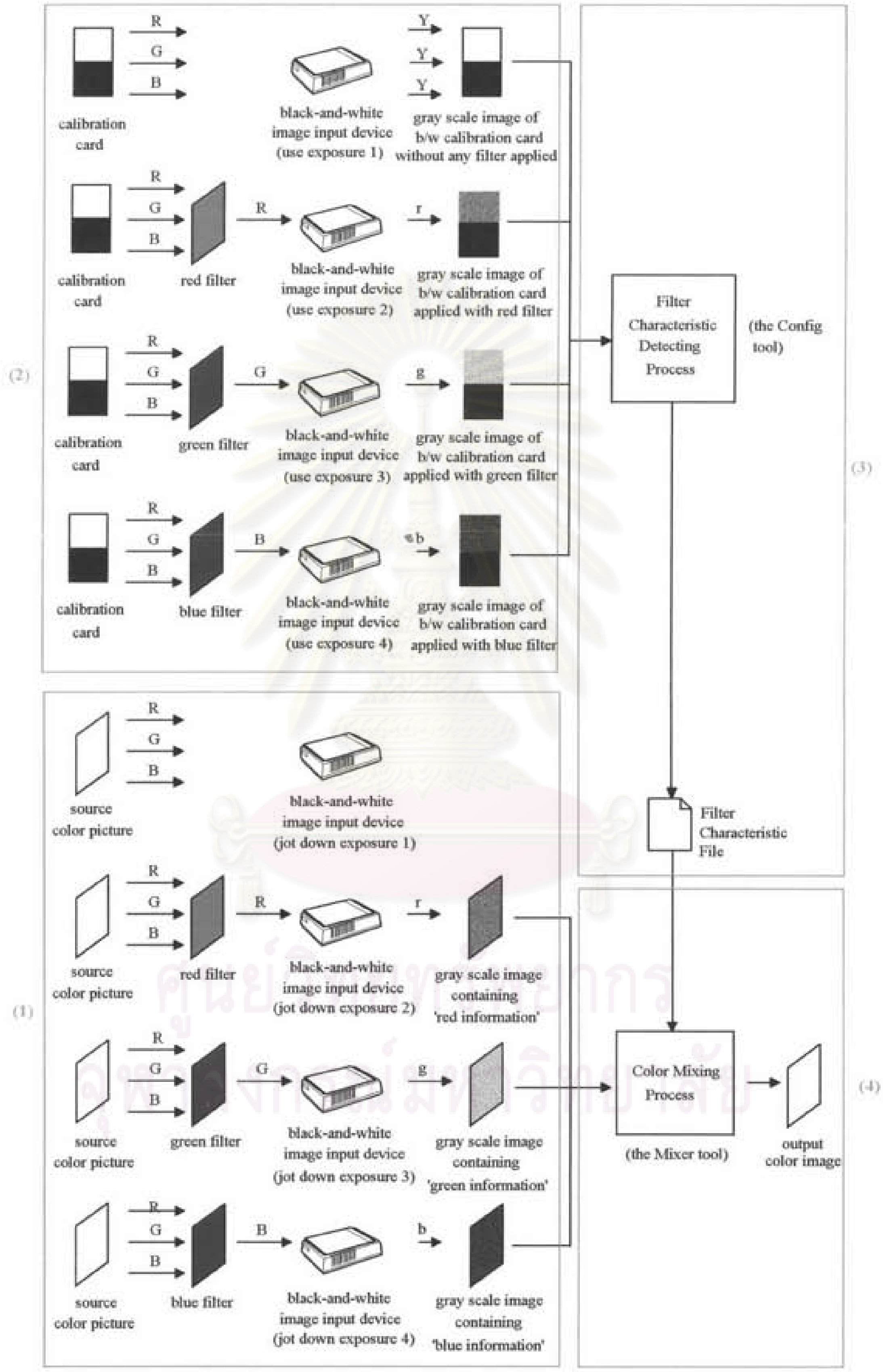


Fig. 3.1 Procedures to get a color image by color mixing

The details of the method designed for obtaining a color image by color mixing are described as follows.

1. The Use of RGB Filters

As mentioned earlier, colors that humans see are actually the mixtures of three primary colors: red, green, and blue. Applying RGB filters during image acquisition allows only, or mostly, the red, green, or blue light, respectively, to pass to the image input device. The purity of the intensities of each color captured by the device depends on the quality of the RGB filters and, thus, filter selection is another important point of this research.

1.1 Filter Selection

The RGB filters in use have much influence on the quality of the input data to the image input device, and consequently on the generated output color image. The ideal RGB filters are expected to allow only the primary color light to pass through. As a result, the ideal captured images will contain only different shades of the primary color light. In this case, the function to transform these gray levels to the intensities of a primary color on the source picture can be derived directly. However, real filters have at least two imperfections: impurity and intensity reduction. Firstly, they allow lights other than the color of the filter to pass through to some extent, and this causes some degrees of color distortion on the final color image. In the case that the error caused by this impurity is ignored, the degree of color distortion on the output image will depend on the amount of other colors that pass through the filters. Secondly, as all filters are translucent, the amount of light that passes through is reduced.

In this research, filters were selected according to the lower impurity and lower intensity reduction. Cellophane and acrylic glass were chosen to be the appropriate candidates for filters. Filters for cameras were not considered as they are too small to be used with a flatbed scanner.

The selection process was done on a color scanner by scanning an image of white paper first, then with each of RGB filters. The results, as shown in Table 3.1, indicated that the acrylic glass filters are better for red and green as they allow more intensity values of their colors, but fewer of other colors, to pass through. Even though the blue acrylic glass filter gave more blue light to pass through than the blue cellophane filter did, the green acrylic glass allowed very little green light to pass through. Furthermore, according to the Y function of the YIQ color model (Equation 2.1), the green light has most influence on the perceived brightness. Therefore, there will be a high error rate when the green filter has high imperfection. As a result, the cellophane filters are more suitable than the acrylic glass filters. Hence, the cellophane filters were chosen for use in this research.

Table 3.1 The result from the filter selection process

Filter		Average intensity value		
Type	Color	R	G	B
None	White	255	255	255
Cellophane	R	207.17	0	0
	G	24.75	109.26	2.28
	B	0	0	106.36
Acrylic glass	R	170.85	0	0
	G	0.26	15.16	0.26
	B	0	0	166.24

2. The Use of a Calibration Card

2.1 Why Using a Calibration Card?

In this research, a calibration card is used to form the relation between the brightness of each primary color on the source picture and the brightness on the image obtained when a filter of consistent color is applied. This relation is then used to compensate for the effect that a filter has made to an image captured with a b/w image input device. The detailed discussion of this topic will be presented in Section 2.3, Using a Calibration Card for Filter Characteristic Detection.

2.2 Making a Calibration Card

Users of the BW2COLOR software can make calibration cards easily by themselves. The calibration card of this research comprises at least one white band and one or more gray bands of different gray levels. If it comprises multiple gray bands, they should be laid in the sequence of decreasing brightness as illustrated in Fig. 3.2. These gray bands should be made of paper pigmented with dense gray color. In other words, paper with gray color made by *halftoning* (or *dithering*)* should not be used. In this research, a color chart of book printing and four color charts of car enamel were tested. More than 100 color bands were scanned to select the best gray bands for use on the calibration card. As a result, four gray bands from a TOA High Gloss Enamel chart were selected because of their purity of grayness. To make a calibration card, these bands must be glued on a piece of paper. The size of each band should be around 0.5 inch × 1 inch. All the bands must be placed next to each other, sorting from white to the darkest gray, with no space in between adjacent bands. Each band must have distinct gray level.

The number of gray bands on a calibration card is limited by the ability of the BW2COLOR software to distinguish different gray bands. The ideal maximum number of gray bands can be obtained by dividing the number of gray levels available (256) by the sensitivity value of the BW2COLOR software. Since the sensitivity value is 10, the ideal maximum number of gray bands is 25. However, the practical maximum number of gray bands is much less than 25 because the gray bands will become less different when the amount of light is reduced by a filter. If the BW2COLOR software detects that two adjacent bands are not much different, it will inform the user to remove some of those bands.

It should be noted that the sensitivity value of 10 gray levels was obtained from the experiment of a calibration card applied with a set of cellophane filters. Ten sample pictures of varying exposures were scanned to measure the automatic exposures. Each of these exposures was then applied to scanning the calibration card applied with each of the filters. As the minimum difference of the gray bands on the calibration card was about 20 gray levels, the sensitivity value was firstly set to 20 gray levels. However, the detected number of gray bands

---

* Halftoning is a technique to approximate gray levels by printing dot patterns of different density, also called*dithering*.

at this sensitivity was less than the actual number. The sensitivity value was then decreased gradually. It was found that the detected number of gray bands was correct when the sensitivity value was around 10 gray levels. When the sensitivity value was set to lower than 10, the software detected more gray bands than it actually existed. Thus, the sensitivity value of the BW2COLOR software in distinguishing different gray bands was set to 10. The experiment led to the condition for making a calibration card that the adjacent bands should be different for more than 20 gray levels.

Thus, the results suggested that at the sensitivity value of 10 gray levels, the adjacent bands should be different for more than 20 gray levels to provide double number of gray levels against the intensity reduction property of filters.

An example of a calibration card is illustrated in Fig. 3.2. When in use, the calibration card must be placed in the direction as shown in Fig. 3.2, with the white band at the top and the darkest gray band at the bottom.

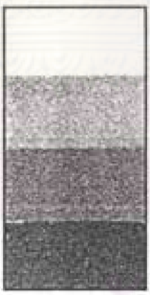
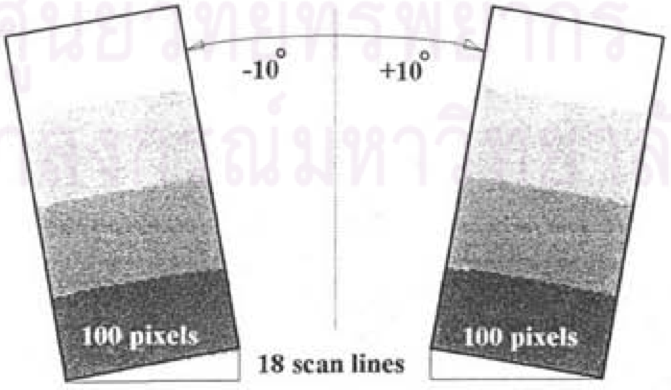


Fig. 3.2 A calibration card

2.3 Using a Calibration Card for Filter Characteristic Detection

As already mentioned, real filters have at least two imperfections: impurity and intensity reduction. If the degree of impurity is low, it can be ignored without causing much color distortion. Thus, there may be two approaches for filter characteristic detection, either by ignoring or recognizing the impurity. On the other hand, if the degree of impurity is high, a specific color calibration card might be needed. As the cellophane filters used in this research have low impurity, the effects of their impurity are ignored. This results in reduction of the detection time. Moreover, being able to use calibration cards that can be made easily by the users results in a higher portability.

A filter characteristic detection scheme was designed to detect the specific characteristics of filters used for gray scale image calibration. Firstly, the gray level of each gray band on a calibration card, which is the average gray level of the pixels on each individual gray band, was identified by the BW2COLOR software. To find this average value, the software must be able to recognize the junctions between these gray bands. It was assumed that a calibration card might be captured at any alignment within 10 degrees of inclination from the vertical axis (any position between the figures shown in Fig. 3.3(a) and 3.3(b)). As the resolution of an image captured for full screen display is usually 100 dots per inch (dpi), averaging the gray levels from six lines of pixels at a time should be adequate to distinguish the junction between bands. When a junction was detected, 3 six-lined ranges, giving a total of 18 lines, were excluded from averaging. This exclusion would cover the junction of 10° inclination on the assumption that the calibration card was approximately 100 pixels (1 inch) wide as illustrated in Fig. 3.3. The areas remained for averaging were illustrated in Fig. 3.4.



(a) (b)

Fig. 3.3 The images of a calibration card with 10° inclination

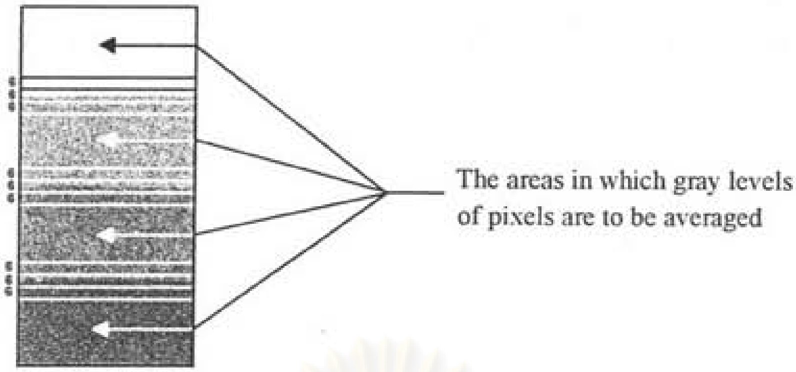


Fig. 3.4 The areas on a band of a calibration card of which gray level are to be calculated

At higher degree of inclination, the number of lines to be excluded from averaging will be too much and the areas that remain for averaging will be too small. As an example, the gray bands of the calibration card used in this research were obtained from a color chart of enamel. They are about half an inch wide. Thus, the number of lines that remained would be only 32 lines of pixels when the degree of inclination was 10°. Furthermore, the inclination of 10° is obvious enough to be observed, and users should adjust the direction of the calibration card to be less inclined.

By comparing the intensities of gray scale images of a calibration card applied with each of RGB filters to those without any filter, transformation functions from the gray levels obtained from the image to the intensities of each primary color on the source color picture can be derived using linear interpolation. Then the functions obtained can be used to perform *histogram modification* to yield the original information on the source color picture.

For example, a calibration card with two bands of different gray level[*] N1 and N2 would give the gray levels n1 and n2 when applied with a filter. When a range of

---

[1] Pradit Pinyopasakul and Nongluk Covavisaruch, "Getting color images using a black-and-white image input device," Paper presented at the 1st National Computer Symposium, Bangkok, 1 April 1994.

[*] One of the two gray bands must be white, as already mentioned, because the lightest band is assumed to be the white band for the BW2COLOR software.

all gray levels available is presumed to be 0 to 255, the transformation functions can be derived in three different intervals, i.e., from 0 to N1, from N1 to N2, and from N2 to 255, as illustrated in Fig. 3.5. These three functions are shown by the equations (3.1).

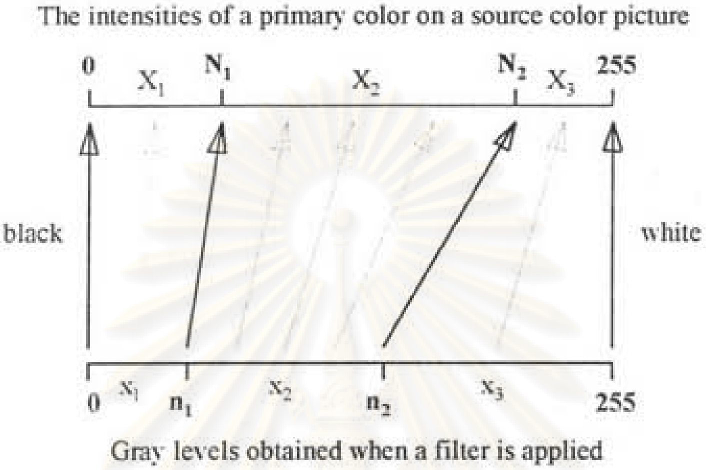


Fig. 3.5 Linear interpolation to transform gray levels backward to the intensities of each primary color on a source color picture

$$
\begin{aligned}
X_1 &= x_1 \times \left(\frac{N_1}{n_1}\right) \\
X_2 &= N_1 + (x_2 - n_1) \times \left(\frac{N_2 - N_1}{n_2 - n_1}\right) \\
X_3 &= N_2 + (x3 - n_2) \times \left(\frac{255 - N_2}{255 - n_2}\right)
\end{aligned}
\qquad (3.1)
$$

Applying linear interpolation to this transformation is proper since the response function of the real input/output device is usually a curve, not a straight line [2] as it was expected to be. For this reason, the more intervals lead to the less error in approximating a curve to a straight line, as illustrated in Fig. 3.6. However, not all errors can be eradicated due to intensity reduction characteristics of filters and the proportional effect of each color to

---

[2] Tim Wegner, Image Lab (California: Waite Group Press, 1992), pp. 79 - 81.

its perceived brightness as indicated by the Y function of YIQ color model, which cause multiple intensities to be mapped into the same gray levels.

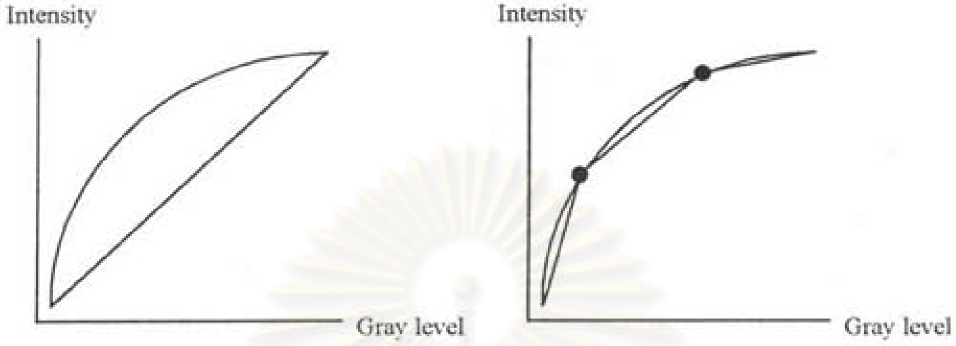


Fig. 3.6 Reduction of error in approximation of curve to straight line when the curve is divided into more intervals

However, having too many gray bands on a calibration card may not reduce errors but may even cause greater errors being due to the inability to recognize all the different bands. In an experiment, the error was also caused by the limitation in the contrast calculation of the DeskScan II Version 1.51,[3] the image input software which comes with the HP ScanJet IIc color scanner. The error occurred when high contrast was needed on the very dark image of a calibration card applied with a blue filter. Fig. 3.7 illustrated the error in an image of a calibration card showing that the second band appeared darker than the third band which was contrary to the actual sequence.

---

[3] "DeskScan II version 1.51", Computer software produced by Hewlett-Packard, California, 1992.

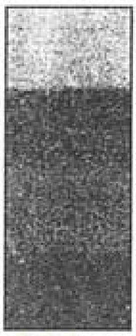


Fig. 3.7 The error in an image of calibration card applied with a blue filter which was caused by the limitation of DeskScan II Version 1.51

Apart from ignoring the impurity of filters which was applied to this research, another method of filter characteristic detection by recognizing the impurity of filters was also proposed. Though it was found that this method is incorrect, it is noted here as for reference. The underlying idea of this method was to recognize the effects of the other primary colors which could pass through a filter by the use of a color calibration card. The hypothesis was that a filter was assumed to have three characteristics: a_r, a_g, and a_b, which were the reduction coefficients of red, green, and blue primary colors, respectively. These three characteristic values ranged from 0 to 1. To find them, three equations were required. The proposed strategy to obtain the three equations was to use a color calibration card with three bands of different mixed colors. A color image input device was used to acquire the reference RGB values of each band. If the reference RGB values of the three bands were (R_1, G_1, B_1), (R_2, G_2, B_2), and (R_3, G_3, B_3), respectively, and the gray values of the three bands when a filter was applied were Y_1, Y_2, and Y_3, respectively, the three equations could be formulated as follows:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} R_1 & G_1 & B_1 \\ R_2 & G_2 & B_2 \\ R_3 & G_3 & B_3 \end{bmatrix} \begin{bmatrix} a_r \\ a_g \\ a_b \end{bmatrix} \tag{3.2}$$

Gaussian Elimination was then applied to solve the Equation (3.2).

The model of the filter characteristics above was tested by applying it to two different calibration cards using the same set of filters. In an experiment, a set of acrylic glass was used as filters. The HP ScanJet IIc color scanner was used to acquire the color coordinates of each color band. It was also used as an emulated black-and-white scanner. The first calibration card comprised 3 color bands whose RGB coordinates were (255, 0, 0), (102, 204, 0), and (153, 255, 255), respectively. The RGB coordinates of the 3 color bands on the second calibration card were (255, 204, 0), (0, 153, 153), and (0, 255, 153), respectively. Each calibration card was scanned using a fixed exposure for each filter. The experiments yielded the results as shown in Table 3.2 and Table 3.3.

Table 3.2 The result from the test of the first color calibration card

Color of filter	Gray level of color bands (Y)		
	(R, G, B) = (255, 0, 0)	(R, G, B) = (102, 204, 0)	(R, G, B) = (153, 255, 255)
Red	181	22	84
Green	69	136	188
Blue	52	188	231

Table 3.3 The result from the test of the second color calibration card

Color of filter	Gray level of color bands (Y)		
	(R, G, B) = (255, 204, 0)	(R, G, B) = (0, 153, 153)	(R, G, B) = (0, 255, 153)
Red	199	71	48
Green	177	85	117
Blue	193	137	219

The Equation (3.2) was applied to the value in Table 3.2 and Table 3.3 and obtained the $(a_r, a_g\ a_b)$ characteristic values as shown in Table 3.4.

Table 3.4 The resulting characteristic values of a set of acrylic glass filters

Color of filter	(a_r, a_g, a_b) when applied with …	
	First calibration card	Second calibration card
Red	(0.7098, -0.2471, 0.1506)	(0.4392, -0.5882, 1.0719)
Green	(0.2706, 0.5314, 0.0435)	(0.6745, -0.5882, 0.4392)
Blue	(0.2039, 0.8196, -0.0361)	(0.6196, 0.2843, 0.9641)

The results from the experiments apparently gave two dissimilar sets of the three coefficients, i.e., the a_r a_g, and a_b significantly changed when the calibration card was changed. These inconsistent values of coefficients indicated that the proposed filter characteristic model was not correct. Hence, the proposed filter characteristic detection scheme cannot be used. The reason why the two sets of coefficients were inconsistent might be that each of the characteristics was not a constant.

As a suggestion for further study, a color calibration card with a pile of color bands of known RGB values may be used. A software to verify each band on an image of the calibration card must be developed for such a calibration card. By verifying the change of intensity of each color band after applied with each filter, a set of color mapping can be obtained. This information may then be used to compensate the impurity effects. However, such a color calibration card would be so specific that the software would lose its portability. Moreover, the verification of all the color bands requires a large number of calculations resulting in a longer execution time. Thus, if a set of filters with high purity, as cellophane, is in use, the effect of other primary colors that can pass through a primary color filter can be ignored, and the color calibration card approach will not be necessary.

3. Image Acquisition

To acquire the RGB information of a source color picture, three gray scale images of that picture with each of the RGB filters applied must be captured. Due to light intensity reduction characteristics of all real filters, the information obtained from those three gray scale images must be transformed to get closer to the actual intensities on the source picture. Such a transformation function for each primary color light can be achieved by mapping the gray level of each gray band on calibration card before a filter is applied (N_1 and N_2 in Fig. 3.5) to that after applying a filter (n_1 and n_2 in Fig. 3.5), as already described in the previous section.

As illustrated by Fig. 3.1, the automatic exposure feature of the image input device in use, if available, should be exploited on capturing each gray scale image of the source picture. The same exposure setting for each filter must also be used in capturing a calibration card image. All the three images of the source picture must be of the same size and position so that each pixel on each image represents the same spot of the original picture.

For the strategy above, seven gray scale images must be captured to obtain a color image. Another capture of the source color picture must also be done to measure the automatic exposure. Comparing with using a color image input device, it is considered to be a very complicated and clumsy task. Hence, to reduce complexity of the image acquisition process, users can capture and keep the exposure settings of the calibration card 4 times and then use it as a common set of exposure values to any color source pictures. Any source color picture can then be captured by using these common exposure settings. However, capturing the source picture first to get the automatic exposure of the picture when a filter is applied always yields the best setting of brightness and contrast to make the use of the gray level range most efficient while maintaining the highlights and shadows of the picture.

4. Color Look-Up Table Generation for Output Images

A color look-up table (LUT) is a collection of colors to be used on an image. It comprises a number of RGB coordinates indexed by numbers starting from 0. The RGB coordinates comprise three 8-bit values, each of which represents the intensity level of red, green, and blue primary color, respectively. When in use, these RGB data are stored into the color palette of the video adapter. For VGA/SVGA adapter, the color palette is implemented as an array of 256×3 Digital-to-Analog Converter (DAC) registers, each of the six-bit size. Thus, the intensity data from the LUT must be scaled from 256 levels (8 bits) to 64 levels (6 bits), by right shifting for 2 bits, before they are stored in DAC. Then an index number can be stored into a proper location in the display memory to display a color pixel on the screen. The digital intensity data will then be converted to analog by DAC before they are sent to VGA/SVGA monitor, as illustrated in Fig. 3.8. For more details on 256-color display on VGA/SVGA monitors, please refer to Appendix B.

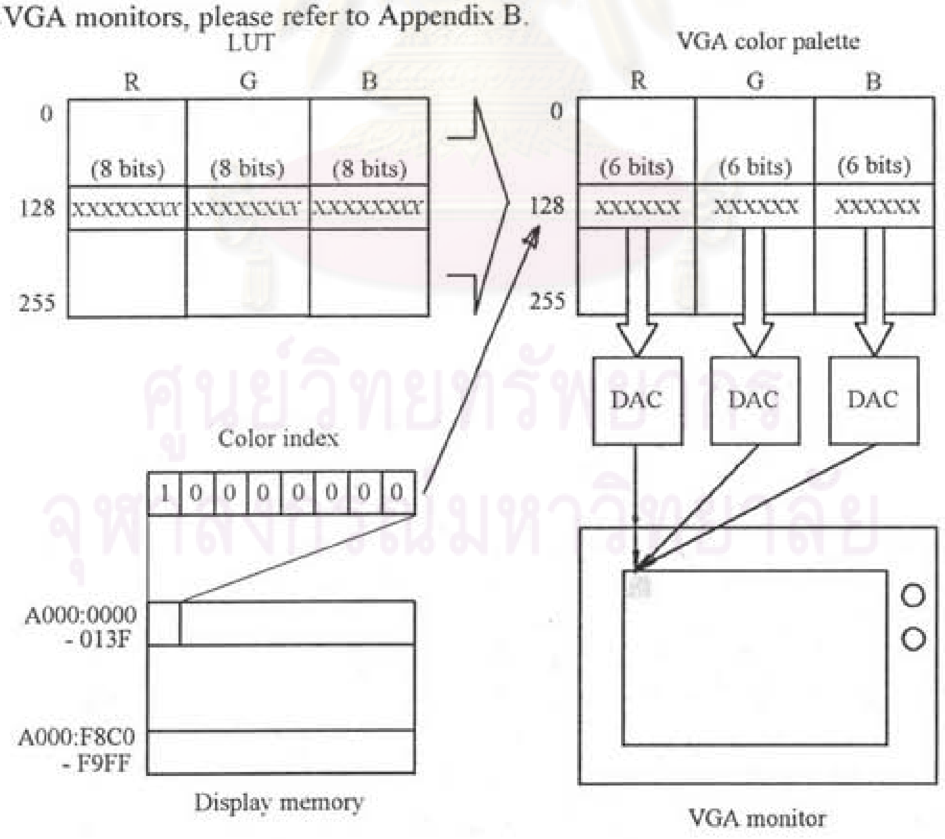


Fig. 3.8 The use of an LUT in 256-color mode of a VGA monitor

LUT is another important factor that affects the quality of the output images. A good LUT should be able to cover the color space actually used in an image, in other words, it should be able to represent most colors of the image. In this section, discussion of how to use the LUT effectively will be presented.

No matter how accurate the procedure to calculate the intensities on the source picture is, if the color table applied is inept, there will be much of color approximation in which the nearest colors are used instead, and there will be high color distortion on the output images. In this research, generating an LUT for an output image can be performed in two ways, either by using a fixed LUT or generating an unfixed LUT according to the characteristic of the colors of the source picture.

4.1 A Fixed Color Look-Up Table

A fixed LUT is the predefined LUT to be applied to any output image. The idea used to design a fixed LUT in this research is to quantize possible intensities of each primary color light into a number of intervals. Each step value of each color is then mixed to all other step values of the other two primaries by means of permutation. As the size of the color table of the output images is 256 colors, the number of step values of each color is 6 and the actual number of colors in use is 216 (6^3), leaving 40 colors unused, as shown in Table 3.5. To make the use of this fixed LUT more efficient, unequal intervals may be applied, as suggested by Zhigang Xiang and Gregory Joy.[4] For instance, as the green primary has the most consequence to the perceived brightness, the number of step values for green may be 7 so that the number of colors in use will be 252, leaving only 4 colors unused. These remaining colors may be used on user interfaces so that the colors of menu will not be changed when the colors on the image being displayed are changed.

---

[4] Zhigang Xiang and Gregory Joy, "Color image quantization by agglomerative clustering," IEEE Computer Graphics and Applications 14 (May 1994): 45.

The 216 colors used in the fixed LUT of the BW2COLOR software look like those illustrated in table 3.5. As the number of steps for each primary is 6, the step size of each interval is 51 $\left(\text{obtained from } \frac{256}{6-1}\right)$. The function to fill up the fixed LUT is shown in Algorithm 3.1.

Table 3.5 The fixed LUT used in the BW2COLOR Software

Index	R	G	B
0	0	0	0
1	0	0	51
2	0	0	102
⋮	⋮	⋮	⋮
6	0	51	0
7	0	51	51
⋮	⋮	⋮	⋮
36	51	0	0
37	51	0	51
⋮	⋮	⋮	⋮
215	255	255	255
216	unused	unused	unused
⋮	⋮	⋮	⋮
255	unused	unused	unused

Algorithm 3.1 Filling up a fixed LUT

Variables I : color index
R, G, B : red, green, and blue values
SR, SG, SB : the step numbers of red, green, and blue values
LUT : 256-color look-up table
LUT[I][0] : red value of the color I
LUT[I][1] : green value of the color I
LUT[I][2] : blue value of the color I

```
BEGIN
   I ← 0;
   FOR SR = 0 TO 5 DO
   BEGIN
      R ← SR × 51;
      FOR SG = 0 TO 5 DO
      BEGIN
         G
         FOR SB = 0 TO 5 DO
         BEGIN
            B ← SB × 51;
            LUT[I][0] ← R;
            LUT[I][1] ← G;
            LUT[I][2] ← b;
         END;
      END;
   END;
END;
```

As the colors in this type of LUT are fixed, the index of the color in this fixed LUT that is closest to a given RGB coordinates can be calculated by the function as shown in Algorithm 3.2.

Algorithm 3.2 ColorIndex(R, G, B, I) Calculating the closest fixed color

Arguments R, G, B : desired red, green, and blue values
I : returning value of color index

```
BEGIN
    I ← 36 × ROUND( R/51.0 ) +
         6 × ROUND( G/51.0 ) +
             ROUND( B/51.0 );
END;
```

4.2 Unfixed Color Look-Up Tables

An unfixed LUT is a specific LUT generated according to the characteristic of the colors on the source picture. In this research, an unfixed LUT is generated by constructing a temporary image of more than 256 colors first. The LUT of the temporary image was constructed as a fixed LUT of, as used in this research, 4096 colors in the same manner as that in Table 3.5. Then the 256 colors which are most frequently used in the temporary image were selected from the first 256 colors of highest bars on the histogram of the temporary image. Finally, the 256 colors selected are stored into the LUT of the output image.

A math-coprocessor is strongly needed when the unfixed LUT is applied, otherwise, the processor will consume a very long time. This is because the processes of constructing a temporary image and selecting the 256 colors to be applied to the output image require a lot of floating point calculations. Furthermore, the color assigned to each pixel on the output image with the unfixed LUT must be obtained by searching the closest color, instead of direct calculation. Hence, in the condition with no math-coprocessor available, a fixed LUT is recommended to lessen execution time.

The number of colors to be used with the temporary image can be up to 16,777,216 (256^3, or 16 M). However, to employ such a large temporary image for a full-screen 1024x768 pixel resolution requires 100,663,296 bytes (or 96 MB) of memory, with 6 bytes for each color of 3-byte index and 3-byte pixel counter. The sizes of the index and the counter may even be 4 bytes when implemented with double-word storage. Such a large number of colors is too extravagant and may yield a color image of lower quality.

Furthermore, the time for generating such a large temporary image and manipulating such a large number of colors is too long. Thus, the number of colors which is only a little larger than 256, such as 512, or 4096, should be used. The selection of the number of colors to be used in the temporary image was done by gradually increasing the power (n) of 2^n, starting from 8 ($2^8 = 256$). It was found that, when increasing from 4096 colors to 8192 colors, five sample images were only slightly better but the processes consumed a significantly longer time. Thus, the number of colors for use in the temporary image in this research was chosen to be 4096, with 24,576 bytes (24 KB) of memory required. The temporary image is easily implemented by an array. For those programmers who want to go beyond 64 KB, the maximum size of an array on DOS, a program library like the *Virtual and Huge Arrays* [5] may be needed.

The function to search for the closest color in an unfixed LUT is shown in Algorithm 3.3. Please note that the unfixed LUT must have been sorted according to (R, G, B) coordinates before applying this function. An example of unfixed LUT is shown in Table 3.5.

Algorithm 3.3 UnfixedColorIndex(R, G, B, I) Searching for the closest color in an unfixed LUT

Arguments R, G, B : desired red, green, and blue values
I : returning value of color index
L : lower limit of search
U : upper limit of search
LUT : 256-color look-up table
LUT[I][0] : red value of the color I
LUT[I][1] : green value of the color I
LUT[I][2] : blue value the color I

---

[5] Graham Robertson, "Virtual and Huge Arrays version 2.03," Computer program library produced by British Software Licensing, Glasglow, England, 1992.

```
BEGIN
    I ← 0;
    L ← 0;
    U ← 255;
    search for r
    WHILE ( I < U ) AND ( ABS( R - LUT[I+1][0] )
            <= ABS( R - LUT[I][0] ) ) DO
       I ← I + 1;
    U ← I;
    search for g
    WHILE ( I > L ) AND ( ABS( R - LUT[I-1][0] )
          <= ABS( R - LUT[I][0] ) )
          AND ( ABS( G - LUT[I-1][1] ) <= ABS( G - LUT[I][1] ) )
       I ← I-1;
    search for b
    WHILE ( I < U ) AND ( ABS( R - LUT[I-1][0] )
          <= ABS( R - LUT[I][0] ) )
          AND ( ABS( G - LUT[I+1][1] ) <= ABS( G - LUT[I][1] ) )
          AND ( ABS( B - LUT[I+1][2] ) <= ABS( B - LUT[I+1][2] ) )
       I ← I + 1;
    RETURN;
END;
```

5. Color Mixing Process

After the three images of the source picture applying RGB filters have been acquired, the transformation function can be applied to each of these images to yield the intensity information of each primary color on the source picture. The resulting intensity of each primary color at each pixel will then be assigned as the intensity of the (R, G, B) coordinates respectively as illustrated in Fig. 3.9, to yield the desired color for the output image. The color and color index of each pixel are acquired according to the selected color LUT's, i.e. fixed or unfixed, as explained earlier in Section 4.1 and Section 4.2.

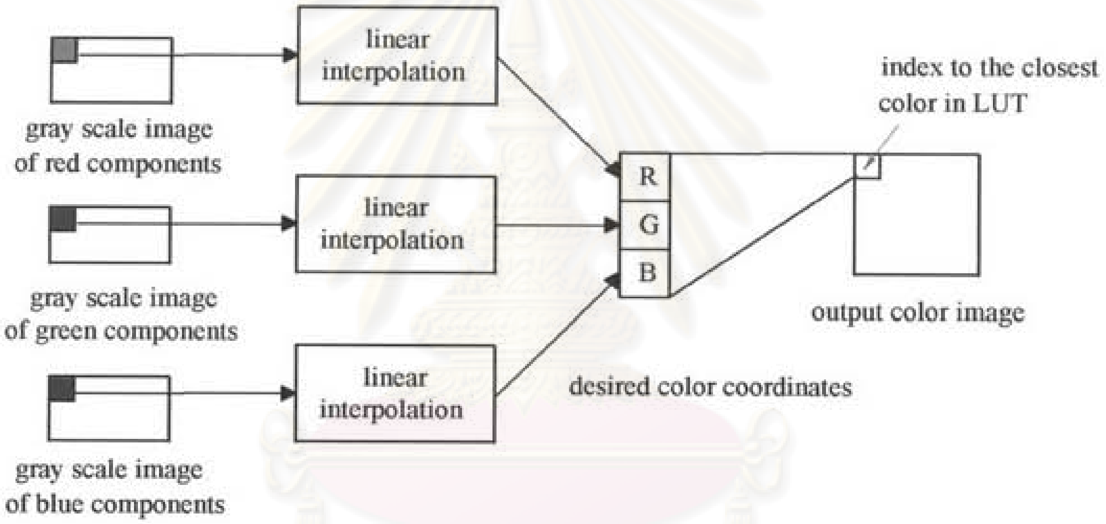


Fig. 3.9 Color mixing process

Color Image Generation by Pseudo Coloring

There is no rule of using pseudo coloring for image enhancement, especially in artistic fantasy. However, a number of guidelines can be formed for a domain of application. The most fundamental guideline is to use colors selected according to some methods. The method exploited in this research is to traverse a smooth path in a color model.[6] A number of heuristics were applied to this research to construct guidelines for

---

[6] James D. Foley and Andries van Dam, Fundamentals of Interactive Computer Graphics (Massachusetts: Addison-Wesley), 1982.

coloring gray scale images by the so-called *gray level to color transformation* technique for the purpose of maintaining continuity of colors or *color harmony* and perceived brightness of colors on the output color images.

A gray-level-to-color transformation function can be applied to a gray scale image easily by changing only its LUT. Thus, it is usually selected for implementation. However, to retain continuity and perceived brightness of colors, the following two heuristic restrictions were used.

1. To preserve continuity of colors, the colors assigned to adjacent gray levels must be harmonious, in other words, hue must be changed only slightly.
2. To keep the perceived brightness unchanged, the color assigned to a gray level must be of the equal perceived brightness.

The perceived brightness can be calculated by the Y function of the YIQ color model, already shown as Equation (2.1):

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

Undoubtedly, it can be seen from the Y function that many different colors can be mapped to the same gray level. Thus, the above two heuristics can be implemented in many ways. However, a random selection of colors usually appear quite garish. Thus, additional programming heuristics, for example, the three presented transformation functions, are needed.

As the values of red, green, and blue components of a color range from 0 to 255, three computer programs are written to ease the designing of the three gray level to color transformation functions as follows:

1. Transformation Function 1

Approach: Sorting the additive primary colors according to their perceived brightness and interpolating the colors in between these primary colors.

Comparing the three additive colors according to the Y function, blue color has least influence to perceived brightness, while red has more influence to perceived brightness than blue, and green has the most influence to perceived brightness. It is reasonable to assign the following colors to gray levels respectively:

black → blue → (magenta) → red → (yellow) → green → cyan → white

The colors in brackets are those which are resulted from the interpolation between the adjacent primary colors. The ones without brackets are fully saturated colors. To cover the whole range of color spectrum, it was designed that, when changing from green to white, blue is added before red which resulted in cyan between green and white.

The perceived brightness of the above colors were calculated by Equation 2.1. Since gray levels are integer number, the resulting numbers were rounded to be the same as those shown in Table 3.6. The intensities of red, green, and blue colors to be assigned to each gray level can be calculated by means of linear interpolation between each pair of two adjacent pole colors as shown in Table 3.7. The graph of the transformation function 1 is illustrated in Fig. 3.10.

Table 3.6 The colors used in the transformation function 1

Gray Level	0	29	→	76	→	150	179	255
Color	black	blue	(magenta)	red	(yellow)	green	cyan	white

Table 3.7 R, G, and B functions of the transformation function 1

Gray Level (i)	Intensity			Note
	R	G	B	
0-29	0	0	$\frac{i}{0.114}$	
30-76	$\frac{255\times(i-29)}{d_1}$	0	$\frac{255\times(76-i)}{d_1}$	$d_1=\frac{29.07\times(76-i)+76.245\times(i-29)}{i}$
77-149	$\frac{255\times(149-i)}{d_2}$	$\frac{255\times(i-76)}{d_2}$	0	$d_2=\frac{76.245\times(149-i)+149.685\times(i-76)}{i}$
150-179	0	255	$\frac{255\times(i-149.685)}{29.07}$	
180-255	$\frac{255\times(i-178.755)}{76.245}$	255	255	

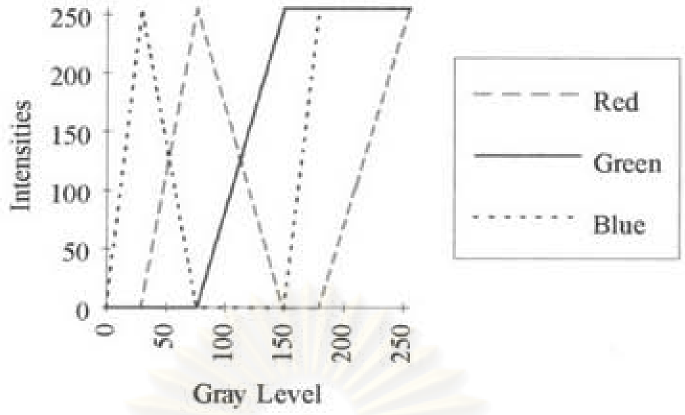


Fig. 3.10 Graph of the transformation function 1

As an example, the (R, G, B) coordinates of the colors between blue and red can be interpolated like this:

According to Equation 2.1; $i = 0.299 \times R + 0.587 \times G + 0.114 \times B$

G = 0; $i = 0.299 \times R + 0.114 \times B \qquad (1)$

For fully saturated blue,

(R, G, B) = (0, 0, 255); $i = 0.114 \times 255 = 29.07$

$\cong 29$ (rounded)

For fully saturated red,

(R, G, B) = (255, 0, 0); $i = 0.299 \times 255 = 76.245$

$\cong 76$ (rounded)

Between blue (i=29) and red (i=76), colors will be gradually changed in this manner:

R increases from 0 (i=29) to 255 (i=76);

$$R = \frac{255 \times (i - 29)}{d_1}$$

B decreases from 255 (i=29) to 0 (i=76);

$$B = \frac{255 \times (76 - i)}{d_1}$$

From (1);
$$i = 0.299 \times \frac{255 \times (i-29)}{d_1} + 0.114 \times \frac{255 \times (76-i)}{d_1}$$

$$= \frac{76.245 \times (i-29) + 29.07 \times (76-i)}{d_1}$$

Thus
$$d_1 = \frac{76.245 \times (i-29) + 29.07 \times (76-i)}{i}$$

After the transformation function 1 was tested with five gray scale images, including the X-ray image shown in Fig. 3.13(a), it was found that:

Pros: The colors assigned have the same brightness as the original gray levels, resulting in harmonious change of colors.

Cons: The colors assigned to near-white range is inept. As the human eye is less sensitive to saturation changes in bright light,[7] only the saturation reduction is not enough for the range from cyan (179) to white (255). Thus, the addition of hue change is needed. An example of this effect is shown in Fig. 3.13(b) where the details inside the bottom sections of the backbone are not cleared.

2. Transformation Function 2

Approach: Inserting subtractive primary colors into function 1 in order to change hue as often as possible.

Transformation function 2 was designed to insert the fully saturated additive colors and the fully saturated subtractive colors into the black to white range, sorting by their perceived brightness so that the following order of colors is applied:

black → blue → (magenta) → red → magenta → green → cyan → yellow → white

The above colors are assigned to the gray levels as shown in Table 3.8 and the intensity functions of red, green, and blue colors are calculated by means of linear

---

[7] James D. Foley et al., Computer Graphics Principles and Practice, 2nd ed. (Massachusetts: Addison-Wesley, 1990), pp. 575-578.

interpolation between each pair of two adjacent pole colors as shown in Table 3.9. The graph of the transformation function 2 is shown in Fig. 3.11.

Table 3.8 The colors applied to the transformation function 2

Gray Level	0	29	→	76	105	150	179	226	255
Color	black	blue	(magenta)	red	magenta	green	cyan	yellow	white

Table 3.9 R, G, and B functions of the transformation function 2

Gray Level (i)	Intensity			Note
	R	G	B	
0-29	0	0	$\frac{i}{0.114}$	
30-76	$\frac{255\times(i-29)}{d_1}$	0	$\frac{255\times(76-i)}{d_1}$	$d_1=\frac{29.07\times(76-i)+76.245\times(i-29)}{i}$
77-105	255	0	$\frac{255\times(i-76.245)}{29.07}$	
106-149	$\frac{1496.85-9.97\times i}{1.77}$	$\frac{255\times(i-105)}{45}$	$\frac{1496.85-9.97\times i}{1.77}$	
150-179	0	255	$\frac{255\times(i-149.685)}{29.07}$	
180-226	$\frac{255\times(i-179)}{d_2}$	255	$\frac{255\times(226-i)}{d_2}$	$d_2=\frac{76.245\times(i-179)+29.07\times(226-i)}{i-249.685}$
227-255	255	255	$\frac{i-225.93}{0.114}$	

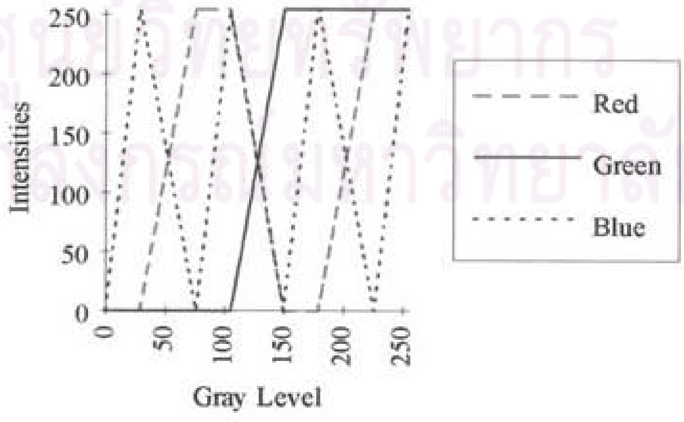


Fig. 3.11 Graph of transformation the function 2

After the transformation function 2 was tested on five gray scale images, it was found that:

Pros: The change of hue from cyan to yellow before fading to white can help solving the problem of transformation function 1. An example of this benefit is shown in Fig. 3.13(c) where the details inside the bottom sections of the backbone are more obvious than those in Fig. 3.13(b).

Cons: The depth dimensions of some of those images were lost. The endeavor to change hue as often as possible caused the images to look unnatural, especially in the range from blue to dark magenta, to red, and then to fully saturated magenta (see Table 3.8). It can be observed that the problem of the transformation function 2 is caused by the very close range between dark magenta and fully saturated magenta which is interleaved by red color only. An example of this effect is shown in Fig. 3.13(c) where the edges of the backbone are blurred.

3. Transformation Function 3

Approach: Solving the problem of the function 1 by the benefit of the function 2. The transformation function 3 is of the following color order:

black → blue → (magenta) → red → (yellow) → green → cyan → yellow → white

The colors above are assigned to the gray levels as shown in Table 3.10 and the intensity functions of red, green, and blue colors are calculated by linear interpolation between each pair of two adjacent pole colors as shown in Table 3.11. The graph of the transformation function 3 is illustrated in Fig. 3.12.

Table 3.10 The colors applied to the transformation function 3

Gray Level	0	29	→	76	→	150	179	226	255
Color	black	blue	(magenta)	red	(yellow)	green	cyan	yellow	white

Table 3.11 R, G, and B functions of the transformation function 3

Gray Level (i)	Intensity			Note
	R	G	B	
0-29	0	0	$\frac{i}{0.114}$	
30-76	$\frac{255\times(i-29)}{d_1}$	0	$\frac{255\times(76-i)}{d_1}$	$d_1=\frac{29.07\times(76-i)+76.245\times(i-29)}{i}$
77-149	$\frac{255\times(149-i)}{d_2}$	$\frac{255\times(i-76)}{d_2}$	0	$d_2=\frac{76.245\times(149-i)+149.685\times(i-76)}{i}$
150-179	0	255	$\frac{255\times(i-149.685)}{29.07}$	
180-226	$\frac{255\times(i-179)}{d_3}$	255	$\frac{255\times(226-i)}{d_3}$	$d_3=\frac{76.245\times(i-179)+29.07\times(226-i)}{i-149.685}$
227-255	255	255	$\frac{i-225.93}{0.114}$	

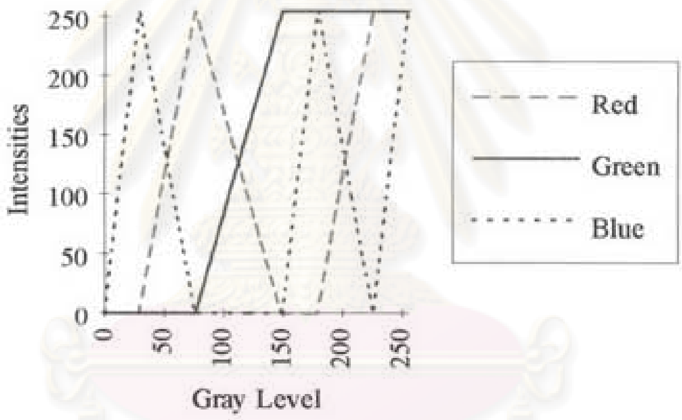


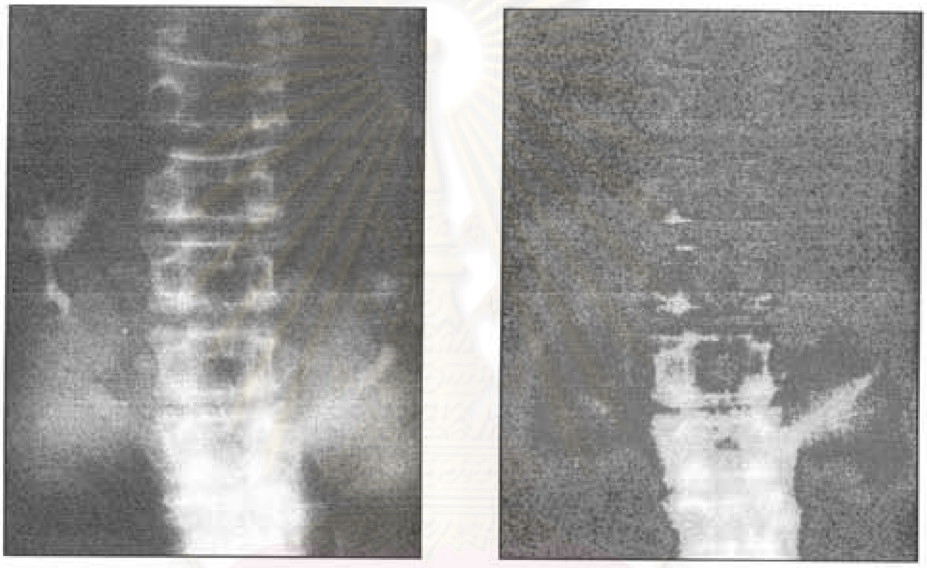
Fig. 3.12 Graph of the transformation function 3

After the transformation function 3 was tested on five gray scale images, it was found that:

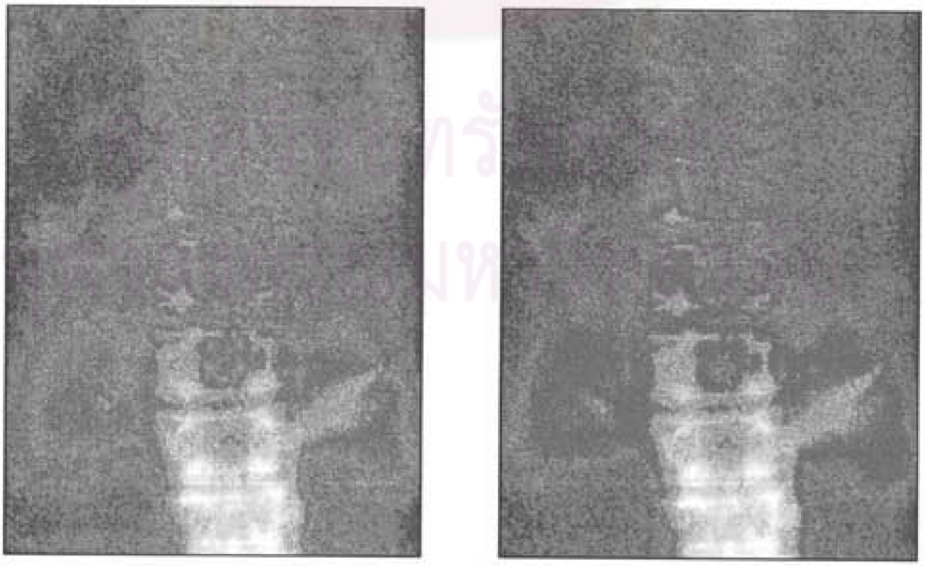
Pros: The problem of function 1 can be solved by this function without the effect of function 2. An example of the application of this function is shown in Fig. 3.13(d).

4. Example Applications of the Three Proposed Transformation Functions

Two sets of gray scale images before and after applied with the three transformation function are shown below to be examples of applying the three functions of color to gray level transformation techniques described above. In general, all the functions are good for applications on the images where colors have no relationship to any object in human sense, such as the X-ray images in Fig.3.13 and paint images in Fig. 3.14.

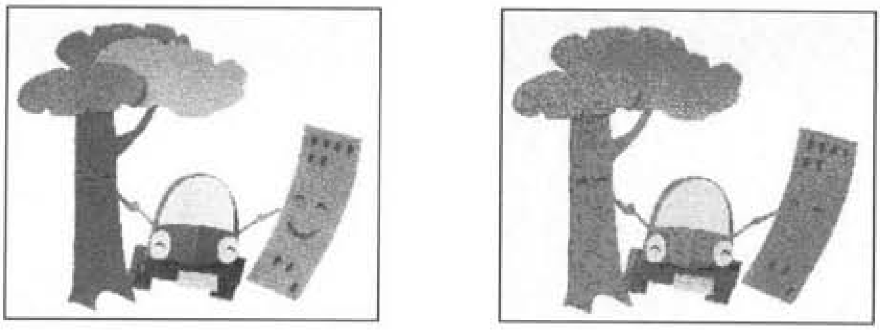


(a) Original X-ray image (b) Colored by function 1



(c) Colored by function 2 (d) Colored by function 3

Fig. 3.13 Sample applications on an X-ray image



(a) Original paint image

(b) Colored by the function 1

(c) Colored by the function 2

(d) Colored by the function 3

Fig. 3.14[*] Sample applications on a paint image

It can be seen that the backbones on the pseudo-colored X-ray images are more obvious, especially the one generated by the transformation function 3, and that the cartoon images applied with pseudo coloring are much more attractive than the original gray scale one.

It should be noted that this is only a case study of pseudo coloring. To get good results, proper transformation functions should be designed for the specific case. Pseudo coloring should also be interactive. For instance, a user should be allowed to select a hue for a range of gray levels and the software should automatically assign a proper saturation for each gray level in that range.

---

[*] The original image was from the outer cover of Smart Drive, a handbook produced by Highway Police Constabulary and Bangchak Petroleum Co., Ltd.