Optimization of Sampling Structure Conversion Methods for Color Mosaic Displays

Xiang Zheng

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, Ontario, Canada
June 2014

A thesis submitted to the University of Ottawa in partial fulfilment of the requirements of the degree of Master of Applied Science in Electrical and Computer Engineering.

© Xiang Zheng, Ottawa, Canada, 2014

Abstract

Although many devices can be used to capture images of high resolution, there is still a need to show these images on displays with low resolution. Existing methods of subpixel-based down-sampling are reviewed in this thesis and their limitations are described. A new approach to optimizing sampling structure conversion for color mosaic displays is developed. Full color images are filtered by a set of optimal filters before down-sampling, resulting in better image quality according to the SCIELAB measure, a spatial extension of the CIELAB metric measuring perceptual color difference. The typical RGB stripe display pattern is tested to get the optimal filters using least-squares filter design. The new approach is also implemented on a widely used two-dimensional display pattern, the Pentile RGBG. Clear images are produced and color fringing artifacts are reduced. Quality of down-sampled images are compared using SCIELAB and by visual inspection.

Index terms: Subpixel rendering, Image down-sampling, Least-squares filter design, SCIELAB.

Acknowledgement

I would like to express my special appreciation and thanks to my supervisor, Professor Eric Dubois, whose expertise, understanding and patience added considerably to my graduate experience. I would like to thank you for encouraging my research, giving me useful remarks and offering me help in the research of this master thesis.

Thank Professor Oscar Au's group: Jin Zeng, Ketan Tang and Lu Fang. I would like to thank you for kindly sharing the software of your subpixel rendering methods. It is extremely useful for this research.

Also, I would like to thank the staff of the School of Electrical Engineering and Computer Science. You provided me advice and assistance during my studies and research in University of Ottawa.

I would like to thank my loved ones, who have supported me throughout the entire process to strive towards my goal. I will be grateful forever for your love.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Display of Color Images

1.1.1 Color Mosaic Display

Color mosaic displays are most frequently used to display color images. A mosaic of primary colors forms the structure of most color image display devices, such as cathode ray tube (CRT) display and plasma display [1]. Color mosaic displays consist of a number of pixels and have a repeating pattern of pixels. Only one color is displayed at each spatial location in color mosaic displays.

Most color mosaic displays have a subpixel structure for the pixels. In a pixel, there are a certain number of subpixels. Each subpixel is typically a rectangular area. Other shapes such as a triangle or a hexagon are also used in some displays. Each subpixel displays one primary color. Most displays use three colors as the basic tiles of the mosaic. Some color mosaic displays use more primary colors [6]. The primary color of a subpixel is produced over the entire subpixel area, so color images displayed on a color mosaic display are continuous-domain images.

The RGB stripe display pattern is a typical color mosaic display pattern, which is shown in Figure 1.1. The RGB stripe display pattern has three stripe subpixels in a pixel. Red, green and blue are used as the primary colors in the RGB stripe display pattern. The white lines in the figure are shown only to help visualize how subpixels are grouped into a pixel structure.

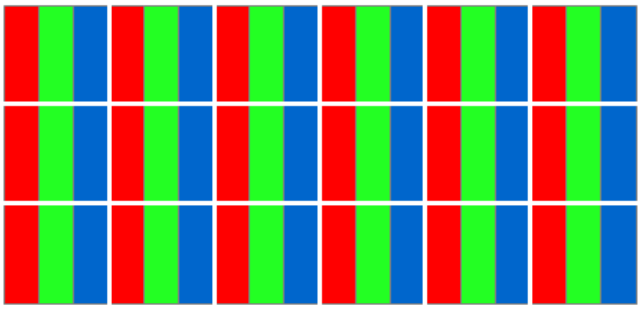


Figure 1.1: The RGB stripe display pattern

1.1.2 Color Images

Devices such as cameras and phones are used to capture color images. Original images on the camera sensor are continuous in both spatial dimensions, the vertical direction and the horizontal direction. Color images are sampled on the sampling structure of the camera color mosaic to obtain the full color images for displaying on color mosaic displays. In this research, we assume a color image has RGB values at each point of a sampling structure. Color images are processed by software and displays before being displayed on color mosaic displays. Displayed images on color mosaic displays are also continuous color images to human observers.

1.2 Problem

1.2.1 Resolution mismatch

Nowadays many portable devices are capable of capturing images with high resolution. For example, a typical camera can capture 10 mega-pixels or more. However, there is still a need to show high-resolution images on displays with lower resolution. For example, a display of size 1680 by 1050, which currently is considered high resolution, can only display images with two mega-pixels or less. Screens on mobile phones and similar devices have

much lower resolution. Thus, high-resolution images and videos must be down-sampled before being displayed.

Using a liquid crystal display (LCD) screen with higher resolution is not a suitable way to solve this problem, because the battery power supply is limited in portable devices. It is expensive for these systems to use a large LCD display, which is unacceptable for the consumers. For that reason, using a software method to obtain images with higher apparent resolution is a solution.

1.2.2 Subpixel Image

In the process of subpixel-based image down-sampling, full color images with high resolution are re-sampled on subpixel structures to obtain subpixel images used to display on lower resolution displays. The human visual system is a low-pass filter, which makes high-frequency patterns invisible at a certain viewing distance. Due to the spatial frequency response of the human visual system, subpixels are not resolved at a normal viewing distance, and displays viewed by human observers behave additively.

1.2.3 Subpixel Rendering

The main problem is to convert input color images to a color mosaic image according to the sampling structure of displays. However, in traditional methods based on the pixels, the anti-aliasing filter often leads to excessive blurring. The subpixel structure of a display can be used to increase the apparent resolution.

Subpixel rendering technologies take advantage of the subpixel display structure and can be used to improve display quality of text and images. Subpixel rendering is a novel way to improve the apparent resolution of down-sampled images. In subpixel rendering, the color of each subpixel is optimized instead of optimizing the color of each entire pixel, in order to enhance the spatial resolution, as shown in Figure 1.2. Full color images are compared with subpixel images in order to produce displayed images appearing to be similar to the original images for the human visual system. Microsoft ClearType [7] is an

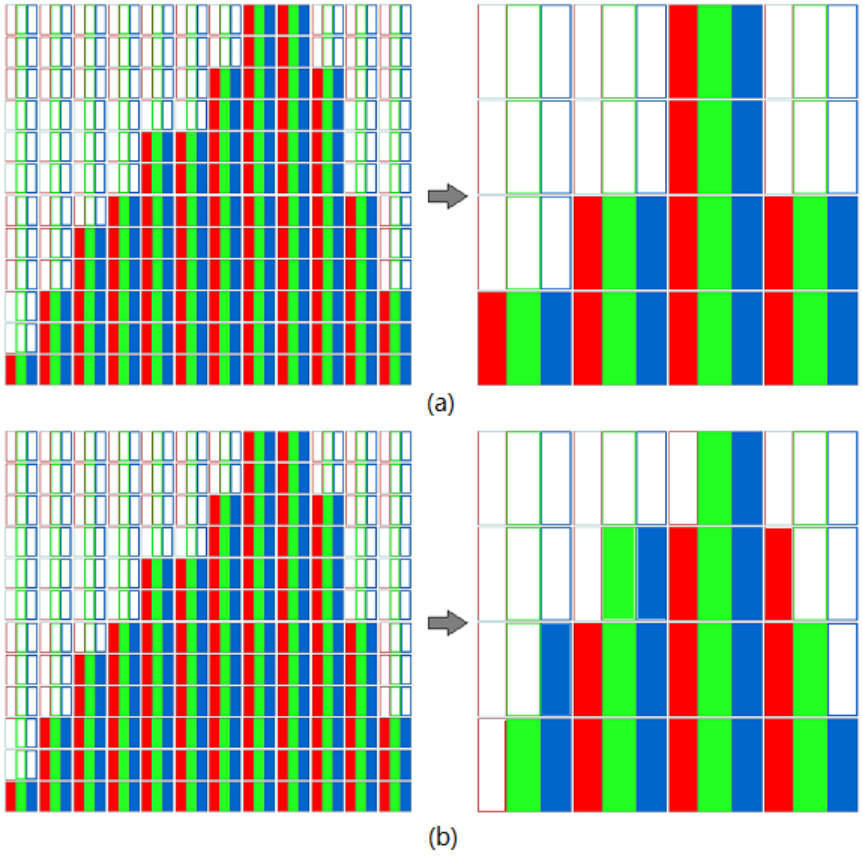


Figure 1.2: Subpixel rendering (a) Down-sampling based on the pixel structure (b) Down-sampling based on the subpixel structure

example of an optimization technique for generating optimal subpixel values making use of the subpixel structure of color mosaic displays.

Subpixel arrangement and subpixel rendering algorithm are both important for increasing quality of displayed images for a human observer. For a certain display pattern, subpixel scaling is a key process in image processing before images are displayed on screens. Without subpixel rendering, the problem of imbalance of local color will lead to color distortion or color fringing when re-sampling images using the subpixel structure. Subpixel rendering generates subpixel images by calculating the color of each subpixel using colors of this subpixel and neighboring subpixels.

Subpixel rendering is a good way to increase the perceptual resolution based on the subpixel structure of color mosaic displays [8][9]. Some methods relieve color fringing at the price of image blurring. The blurring effect will be brought by using anti-aliasing filters in the process of image down-sampling, so the balance between perceived resolution and color distortion is significant in the technology of subpixel rendering [10].

1.2.4 Image Quality Metrics

Among existing image quality metrics [11][12][13] , we choose the SCIELAB color metric [3] in this research to measure the color difference between original images and displayed images. The SCIELAB color metric is a spatial extension to the CIELAB color metric and better measures the perceived color difference. Pattern-color separability of the human visual system is introduced in [14]. This property of human vision is derived from experimental data with different methods. The SCIELAB metric makes use of the pattern-color separability of the human vision system to increase computation efficiency of the error calculation. The SCIELAB color metric models the human visual system using pattern-color separability and three spatial filters, from data of psychophysical experiments, acting as the low-pass filtering effect of human vision.

1.3 State of the Art

There are several state-of-the-art methods in previous work by other researchers in [4][5][15][16] [17], which take advantage of the subpixel structure and the blurring effect of the human visual system to convert sampling structures of full color images to the RGB stripe display pattern and minimize different reproduction errors.

There are some conventional basic schemes available for subpixel rendering. The direct subpixel-based down-sampling (DPD) is an example of a basic scheme of subpixel rendering. Sloping edges are not well reproduced using DPD, which is shown in Figure 1.2(a). Other examples of conventional basic schemes are the Direct Subpixel-based Down-sampling (DSD) in Figure 1.2 (b) and Diagonal Direct Subpixel-based Down-sampling

(a) (b)

Figure 1.3: Color fringing artifacts (a) An original image (b) A down-sampled image with color fringing artifacts on the frame of the window

(DDSD). During the down-sampling process, using no anti-aliasing filters may lead to color fringing artifacts, which is shown in Figure 1.3.

The optimal filtering in the paper by Platt [18] generates a set of filters to minimize the color error of fonts displayed on the RGB stripe displays. Platt's method is used in Microsoft ClearType to enhance the resolution and clarity of fonts displayed on color mosaic displays. The same filters as Platt's method are used in [19]. Displaced box filters are used in the optimization to suppress the color fringing of sampled text graphics on subpixel displays.

In [5], a method of subpixel-based image down-sampling optimizing a minimum mean square error is implemented to generate clearer images compared with basic schemes of subpixel-based down-sampling. Another method of subpixel-based down-sampling minimizing the min-max directional error is introduced in [4]. The problem is relaxed in two ways to reduce computational complexity.

Most conventional subpixel rendering methods are based on the typical RGB stripe display pattern and the good color metric SCIELAB is not used in these conventional methods to reduce the color distortion. With more and more new two-dimensional (2D) display patterns being used in display devices, a new approach to subpixel rendering for general pixel geometries is necessary to improve image quality under the SCIELAB metric.

1.4 A New approach

1.4.1 System Structure

The general process of the new approach is shown in Figure 1.4. This is the general form of a linear shift-invariant system for three-dimensional (3D) color signals as described in [1].

Each color channel of the original image is prefiltered by a set of filters in the spatial domain. For example, RGB color channels are filtered by a set of nine filters to obtain prefiltered color channels for typical RGB stripe displays, as shown in Figure 1.5. Filtered color images are down-sampled using one of basic subpixel down-sampling schemes, as shown in Figure 1.6.

Original full color images and subpixel images rendered by the spatial filters are compared using the SCIELAB metric. Before calculating SCIELAB errors in the nonlinear LAB color space, original full color images and subpixel images are linearly converted into a opponent color space and then filtered using the low-pass filters used in SCIELAB. The frequency weighted errors between original images and subpixel images in the opponent color space are minimized in the new approach in order to attempt to minimize the SCIELAB errors in the nonlinear color space.

Let $\mathbf{f}$ and $\mathbf{g}$ be the original image and subpixel image and let $\mathscr{M}$ the subspace of subpixel images in the vector space of images. We measure the difference of two color images by $\| \mathbf{f} - \mathbf{g} \|_O$ in the SCIELAB opponent space weighted by weighting filters in opponent channels.

Given the original image $\mathbf{f}$, we seek the optimal subpixel image $\mathbf{f}_s$ in $\mathscr{M}$:

$$\mathbf{f}_s = \underset{\mathbf{g} \in \mathscr{M}}{argmin} \| \mathbf{f} - \mathbf{g} \|_O \tag{1.1}$$

The optimal subpixel image $\mathbf{f}_s$ can be obtained by the projection of $\mathbf{f}$ on $\mathscr{M}$ according to this norm. Least-squares filter design using a training set is used in this research to optimize the frequency weighted error.

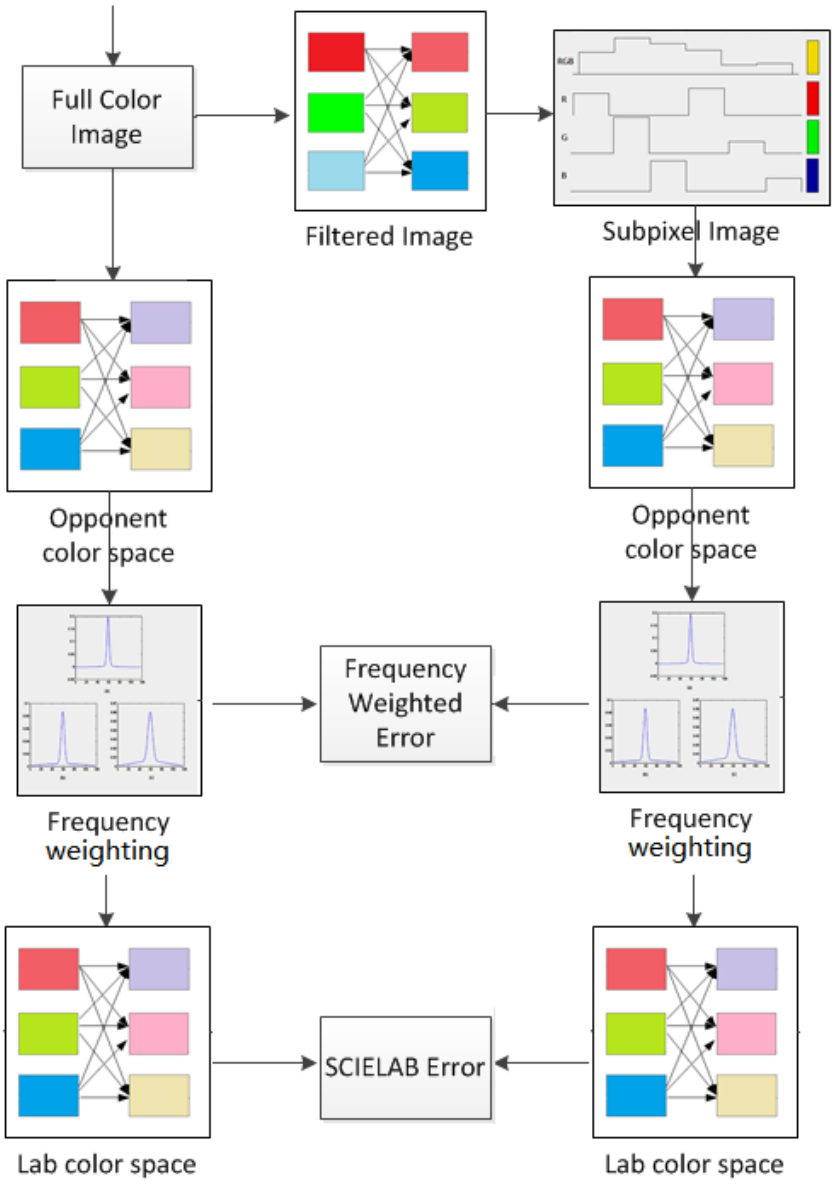


Figure 1.4: General process of the new approach

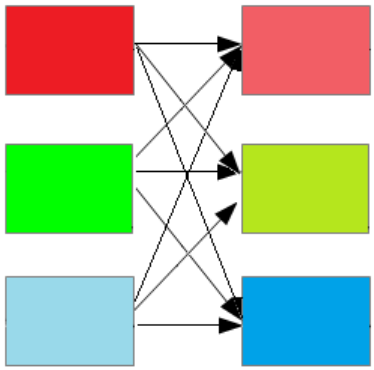


Figure 1.5: Prefiltering RGB images by nine filters

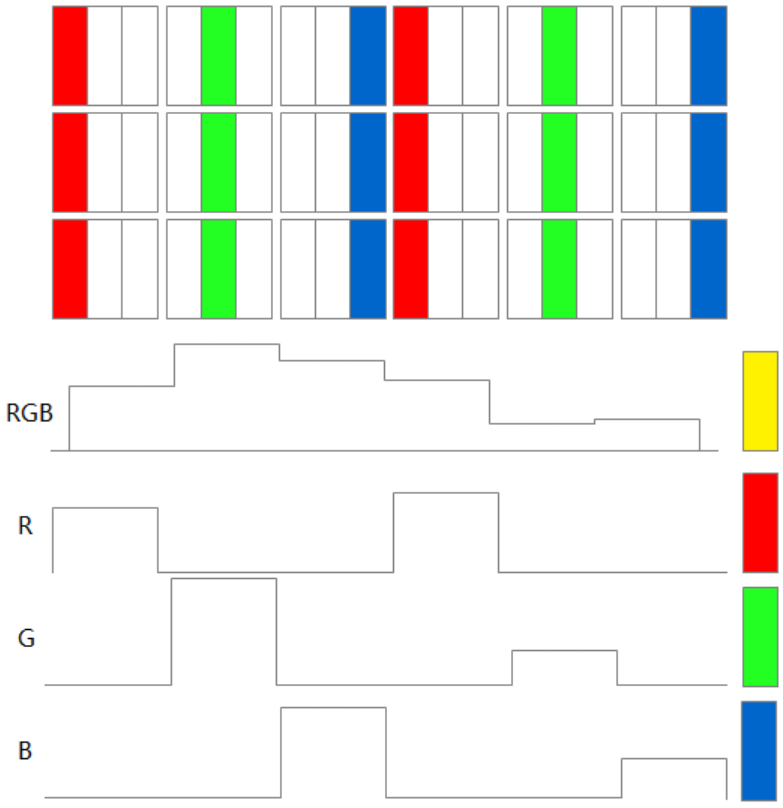


Figure 1.6: Sampled RGB signals

1.4.2 Results

Two analytical models, the RGB stripe display pattern and the Pentile RGBG display pattern, are tested in this research using the new approach and existing subpixel rendering methods. Photographic images and text images are tested for both display patterns. Both the frequency weighted errors and SCIELAB errors are minimized using the new approach, compared with existing methods. The new approach to subpixel rendering enhances the quality of down-sampled images by suppressing color differences and keeping details of images.

1.4.3 Contribution

There exists no methods in the literature that present a correct linear shift-invariant prefiltering in 3D color space, down-sampling to the subpixel mosaic, with filters optimized to minimizing a perceptual error such as frequency weighted mean square error of SCIELAB. This thesis will fill this gap.

Theoretical results and two analytical models are developed for improving image quality of color images after subpixel-based down-sampling on 2D display patterns of color mosaic displays. The new approach minimizes both frequency weighted errors and perceptual color errors more compared with conventional subpixel rendering methods.

1.5 Thesis overview

Chapter 2 introduces the background of this research. Definitions necessary for formulating and discussing the problem of sampling structure conversion are stated in this chapter. The RGB stripe display pattern is used as a specific pattern to introduce definitions and formulation in Chapter 2. Previous work by other researchers in this area is also introduced and reformulated according to the book [1], in order to compare my results with theirs.

By giving a mathematical solution, Chapter 3 introduces the new approach to improve frequency weighted mean square errors and errors under SCIELAB color metric by optimizing filters used to prefilter full color images, under the formulation described in Chapter

2. The mathematical solutions to the problem of sampling structure conversion for the RGB and Pentile RGBG display patterns are derived in order to implement the optimization.

In Chapter 4, the optimization of nine filters for the RGB stripe display is implemented, which can be done in just one dimensional (horizontal). The results of the RGB stripe display pattern are shown and discussed in this chapter. Two kinds of errors are calculated to verify that the optimization works correctly for the RGB stripe display pattern.

In Chapter 5, the Pentile RGBG display pattern is used as a model to implement the optimization in both horizontal and vertical directions of the display pattern. The frequency weighted mean square errors and SCIELAB errors are calculated for images reproduced on the Pentile RGBG display pattern. The results of error calculation verify that the new sampling structure conversion method can be used for 2D display patterns, including the current widely-used display patterns and other developing 2D display pattern in the future.

Chapter 6 summarizes the thesis and highlights the contributions of this research. Lastly, future directions to carry this work further are discussed.

Chapter 2

Background

In this chapter, definitions and formulation of sampling structures, color spaces and display patterns are introduced, in order to formulate subpixel-based down-sampling systems. The SCIELAB color metric is also introduced in this chapter, which will be used as the objective in optimization of down-sampling systems. Previous work by other researchers in subpixel rendering are introduced and formulated using consistent notations.

2.1 Definitions

2.1.1 RGB Stripe Display Pattern

The RGB stripe display is a most widely used pixel layout on displays. Each pixel on the RGB stripe display contains one red, green and blue subpixel. The layout of the RGB stripe display is shown in Figure 1.1. Each subpixel is of the same size. Three subpixels are in a row with a stripe shape. Each pixel is square with both the width and height d. Each subpixel has the same height as pixels and has one-third the width of the pixel , $\frac{d}{3}$.

Because of the blurring effect of human eyes, the three subpixels appear to be one single color. If we use a magnifying glass, we can see subpixels without blurring. Both LCD and organic light-emitting diode (OLED) displays have subpixel structures. Subpixel rendering increases apparent resolution based on subpixel structures.

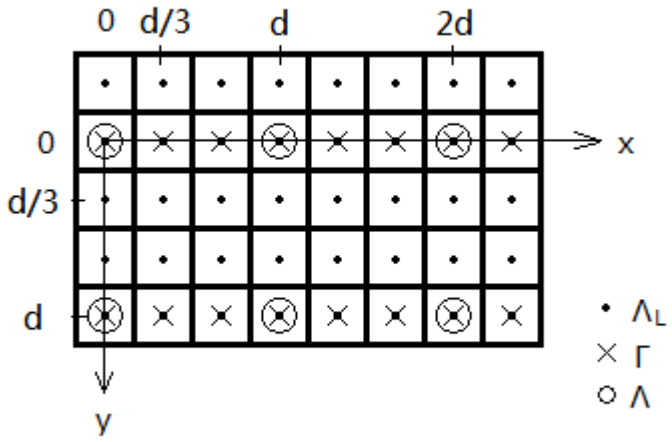


Figure 2.1: The sampling structure of the RGB stripe display pattern

2.1.2 Sampling Lattice

According to the structure of the RGB stripe display pattern, we can define a sampling lattice on the RGB stripe display pattern to have our discrete 2D signals described. The introduction of lattices can be found in the appendix of [1], which is not repeated in this thesis. For each pixel, we take a sample at its center, then we have the sampling lattice Λ_L, which is shown in Figure 2.1.

$$\Lambda_L = \mathrm{LAT}(\begin{bmatrix} \frac{d}{3} & 0 \\ 0 & \frac{d}{3} \end{bmatrix}) \tag{2.1}$$

After we have the sampling lattice Λ_L, full RGB images can be defined on this sampling lattice. Full RGB images have all three red, green, and blue components defined at each lattice point. In both horizontal and vertical directions, the distance between two samples is $\frac{d}{3}$.

Thus, if we use the notation $\mathbf{f}(\mathbf{x})$ to describe the full RGB images defined on lattice Λ_L, we have:

$$\mathbf{f}(\mathbf{x}) = [f_R(\mathbf{x}), f_G(\mathbf{x}), f_B(\mathbf{x})], \mathbf{x} \in \Lambda_L \tag{2.2}$$

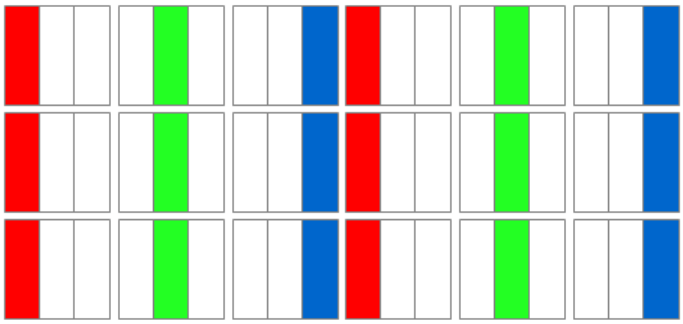


Figure 2.2: The sampling strategy for the RGB stripe display pattern

2.1.3 Subpixel Image

Subpixel images are defined on the sampling lattice Λ_L. However, for each primary color, values of samples are zero, if samples are not on the sampling lattice for this primary color. The sampling strategy for the RGB stripe display pattern is shown in Figure 2.2.

We can define sampling lattice for subpixels of each primary color. Then, we have three sampling structures $\Lambda_R, \Lambda_G,$ and Λ_B for red, green and blue subpixels individually.

If we define

$$\Lambda_R = \Lambda = \mathrm{LAT}(\begin{bmatrix} d & 0 \\ 0 & d \end{bmatrix}), \tag{2.3}$$

the sampling structure for green subpixels can also be seen as the sampling lattice for red subpixels shifted to the right by $\frac{d}{3}$. The sampling structure for blue subpixels is the sampling lattice for red subpixels shifted to the right by $\frac{2d}{3}$. Then,

$$\Lambda_G = \Lambda + \begin{bmatrix} \frac{d}{3} \\ 0 \end{bmatrix} \tag{2.4}$$

$$\Lambda_B = \Lambda + \begin{bmatrix} \frac{2d}{3} \\ 0 \end{bmatrix} \tag{2.5}$$

Thus, we can define subpixel images on the sampling structure as:

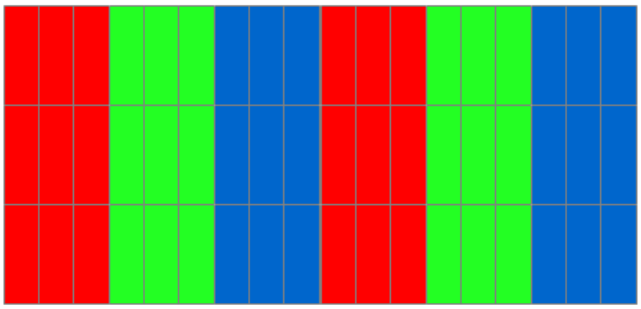


Figure 2.3: Subpixel images of the RGB stripe display pattern

$$
\begin{aligned}
\mathbf{f}_S(\mathbf{x}) &= [f_{RS}(\mathbf{x}), f_{GS}(\mathbf{x}), f_{BS}(\mathbf{x})], \mathbf{x} \in \Lambda_L \qquad (2.6)\\
&= \begin{cases}
(f_R(\mathbf{x}),0,0) & \mathbf{x} \in \Lambda_R \\
(f_R(\mathbf{x}-(0,\frac{d}{3})),0,0) & \mathbf{x} \in \Lambda_R + (0,\frac{d}{3}) \\
(f_R(\mathbf{x}-(0,\frac{2d}{3})),0,0) & \mathbf{x} \in \Lambda_R + (0,\frac{2d}{3}) \\
(0,f_G(\mathbf{x}),0) & \mathbf{x} \in \Lambda_G \\
(0,f_G(\mathbf{x}-(0,\frac{d}{3})),0) & \mathbf{x} \in \Lambda_G + (0,\frac{d}{3}) \\
(0,f_G(\mathbf{x}-(0,\frac{2d}{3})),0) & \mathbf{x} \in \Lambda_G + (0,\frac{2d}{3}) \\
(0,0,f_B(\mathbf{x})) & \mathbf{x} \in \Lambda_B \\
(0,0,f_B(\mathbf{x}-(0,\frac{d}{3}))) & \mathbf{x} \in \Lambda_B + (0,\frac{d}{3}) \\
(0,0,f_B(\mathbf{x}-(0,\frac{2d}{3}))) & \mathbf{x} \in \Lambda_B + (0,\frac{2d}{3})
\end{cases}
\end{aligned}
$$

The subpixel image of the RGB stripe display pattern is shown in Figure 2.3.

Our goal is to choose values of the subpixel image to make it appear as similar as possible to the full color image to a human viewer at a given distance. To do this, we need to specify spatial distance in degrees observed at the eye.

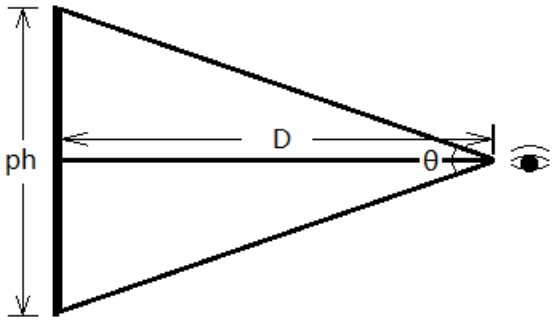


Figure 2.4: Viewing conditions

2.1.4 Sampling Density

2.1.4.1 Sample Per Degree

Sample per degree is a unit used to describe sampling density. This unit can be used for filters processing 2D signals defined on the sampling structures of display patterns. Degrees of the visual angle are used as the unit of distance. The viewing distance used to describe viewing conditions needs to be converted to the distance measure of displays such as pixels or picture height.

Assume a screen has width n_1 pixels (columns) and height n_2 pixels (rows). The picture height is ph and the picture width is pw. Images on the screen are viewed at a distance of D, as shown in Figure 2.4. For the vertical direction, the viewing angle is:

$$\theta = 2\arctan\frac{ph}{2D} \text{ (degrees)} \tag{2.7}$$

So the sampling density of pixels in the unit of samples per degree is:

$$SDP = \frac{n_2}{\theta} = \frac{n_2}{2\arctan\frac{ph}{2D}} \tag{2.8}$$

For the RGB stripe display, one pixel contains three subpixels, so the sampling density for subpixels is:

$$SPD = 3 \times \frac{n_1}{2 \arctan \frac{pw}{2D}} \tag{2.9}$$

2.1.5 Sampling Structure Conversion

When we need to down-sample full RGB images, sampling structure conversion is necessary [1].

2.1.5.1 Sublattices

In order to introduce sampling structure conversion, we need the definition of a sublattice. If both lattices Γ_1 and Γ_2 are in $\mathbb{R}^D$, and all the points of Γ_1 are also points in lattice Γ_2, then Γ_1 is a sublattice of Γ_2, and Γ_2 is a superlattice of Γ_1.

For example, the lattice Λ_R or Λ for the red color component is the sublattice of the lattice of the pixels in Λ_L. Λ_L is the superlattice of the lattice for red color component.

2.1.5.2 Down-sampling

Because we want to down-sample images taking advantage of subpixel structures, we need to convert sampling structures from pixel structures to subpixel structures. It is a process of down-sampling.

If f is the signal defined on the lattice Γ_2, and $\Gamma_1 \subset \Gamma_2$, then the down-sampled image of f is:

$$g[\mathbf{x}] = f[\mathbf{x}], \mathbf{x} \in \Gamma_1 \tag{2.10}$$

The values on lattice $\Gamma_2 \backslash \Gamma_1$ are removed, which will introduce extra spectral replicas in the frequency domain. So we need to filter the signal $f[\mathbf{x}], \mathbf{x} \in \Gamma_2$ before we down-sample it.

The ideal anti-aliasing prefilter has frequency response:

$$H(\mathbf{u}) = \begin{cases} 1 & \mathbf{u} \in \mathscr{P}_{\Gamma_1^\star} \\ 0 & \mathbf{u} \in \mathscr{P}_{\Gamma_2^\star \setminus \Gamma_1^\star} \end{cases} \tag{2.11}$$

where $\Gamma_1^\star$ and $\Gamma_2^\star$ are the reciprocal lattice of Γ_1 and Γ_2, and $\mathscr{P}$ is the unit cell of a lattice.

2.1.6 Color Coordinate Conversion

The color space used in this thesis is CIE 1931 color space and different bases are used, such as XYZ and sRGB [20]. In the process of image reproduction, color coordinate conversion is necessary for processing images in different color coordinate systems. Some color coordinate systems [1] used in this research are introduced in this subsection.

2.1.6.1 XYZ

XYZ is the standard coordinate system used to represent colors and all other coordinate systems are referred to XYZ.

2.1.6.2 sRGB

The sRGB primaries are also called ITU-R Rec. 709 Primaries, which is used by most digital cameras. The tristimulus values of RGB primaries are defined according to their XYZ tristimulus and the reference white, which is typical of daylight D_{65}. The primaries are denoted as $[R_{709}], [G_{709}]$ and $[B_{709}]$. They satisfy the constraints:

$$\begin{aligned} D_X &= \frac{d_X}{d_Y} = 0.9505 \\ D_Y &= 1 \\ D_Z &= \frac{d_Z}{d_Y} = 1.0888 \end{aligned} \tag{2.12}$$

and

$$[D_{65}] = [R_{709}] + [G_{709}] + [B_{709}] \tag{2.13}$$

Thus, sRGB tristimulus values can be obtained by the following conversion from the XYZ tristimulus values:

$$\begin{bmatrix} Q_R \\ Q_G \\ Q_B \end{bmatrix} = \begin{bmatrix} 1.9591 & 0.9293 & -0.3013 \\ -0.5942 & 1.1343 & 0.1990 \\ 0.1143 & -0.1234 & -1.0835 \end{bmatrix} \begin{bmatrix} Q_X \\ Q_Y \\ Q_Z \end{bmatrix} \tag{2.14}$$

2.1.6.3 CIELAB

The CIELAB color space is a nonlinear color space, which is a widely used measuring system for color differences as an international standard. It gives a reasonable perceptual uniformity for measuring errors in color reproduction. We can obtain the color components for CIELAB color space by converting a color from the XYZ color space. Values of the components depend on the choice of the reference white $[W]$.

The brightness component in CIELAB space is:

$$Q_{L^\star} = \begin{cases} 903.3\frac{Q_Y}{W_Y} & \frac{Q_Y}{W_Y} \leq 0.008856 \\ 116(\frac{Q_Y}{W_Y})^{\frac{1}{3}} - 16 & \frac{Q_Y}{W_Y} > 0.008856 \end{cases} \tag{2.15}$$

where $0 \leq Q_Y \leq W_Y$, so that the range of the value $Q_{L^\star}$ is from 0 to 100.

The other two color components of the CIELAB system can be described by the XYZ tristimulus values too:

$$Q_{a^\star} = 500[q(\frac{Q_X}{W_X}) - q(\frac{Q_Y}{W_Y})] \tag{2.16}$$

$$Q_{b^\star} = 200[q(\frac{Q_Y}{W_Y}) - q(\frac{Q_Z}{W_Z})] \tag{2.17}$$

where the function $q(t)$ is

$$q(t) = \begin{cases} t^{\frac{1}{3}} & \text{if } t > (\frac{6}{29})^3 \\ \frac{1}{3}(\frac{29}{6})^2 t + \frac{4}{29} & \text{if } t \leq (\frac{6}{29})^3 \end{cases} \tag{2.18}$$

2.1.6.4 Opponent Color Space

The opponent color space is used to approximate the first processing of the human visual system in the SCIELAB color metric, which is a spatial CIELAB model defined by Zhang and Wandell [3].

It is a linear system for color images, so it can be linearly converted from other linear color space. For example, if the color image is defined by the CIE 1931 XYZ tristimulus values, the linear transformation from XYZ color space to opponent color space is:

$$\begin{aligned} \begin{bmatrix} Q_{O_1} \\ Q_{O_2} \\ Q_{O_3} \end{bmatrix} &= \begin{bmatrix} 0.6266 & 1.3699 & 1.5057 \\ -1.8672 & 0.9348 & 1.4213 \\ -0.1532 & 0.4362 & 2.5360 \end{bmatrix} \begin{bmatrix} Q_X \\ Q_Y \\ Q_Z \end{bmatrix} \\ &= \mathbf{A}_{XYZ \to OPP} \begin{bmatrix} Q_X \\ Q_Y \\ Q_Z \end{bmatrix} \end{aligned} \tag{2.19}$$

2.1.7 Gamma Correction

Gamma correction, shown in Figure 2.5, is required for CRT displays. The relation between the light output I of the cathode ray tube and the voltage V is approximately a power law [1]:

$$I = V^{\gamma} \tag{2.20}$$

or a better expression:

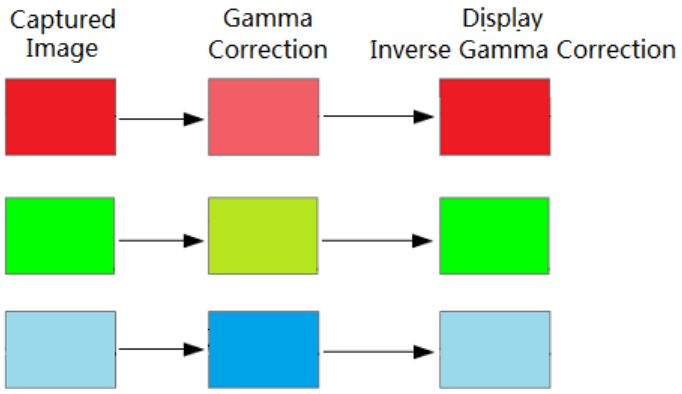


Figure 2.5: Gamma correction

$$I = (V + \varepsilon)^{\gamma} \tag{2.21}$$

Gamma correction is also introduced into other displays like LCD, so a gamma-corrected image can be displayed correctly on these displays.

The gamma-corrected RGB color space is more perceptually uniform. Let the tristimulus values of linear RGB color space be Q_R, Q_G and Q_B and the values of gamma-corrected RGB color space be $Q_R^{'}, Q_G^{'}$ and $Q_B^{'}$. The standard form of gamma correction is :

$$Q_i^{'} = \begin{cases} 4.5Q_i & Q_i \leq 0.018 \\ 1.099Q_i^{0.45} - 0.099 & 0.018 < Q_i \leq 1.0 \end{cases} \tag{2.22}$$

where $i \in \{R, G, B\}$.

The inverse gamma correction is

$$Q_i = \begin{cases} \frac{2}{9}Q_i^{'} & Q_i^{'} \leq 0.081 \\ (\frac{Q_i^{'}+0.099}{1.099})^{\frac{20}{9}} & 0.081 < Q_i^{'} \leq 1.0 \end{cases} \tag{2.23}$$

where$i \in \{R, G, B\}$.

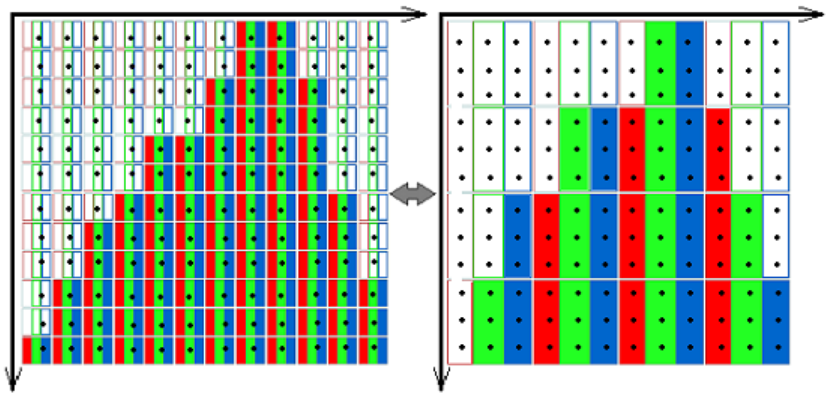


Figure 2.6: A comparison lattice

2.1.8 Comparison Lattice

After reproduction, reproduced images are to be compared with original images. The original images captured by devices capable to capture high-resolution images are continuous. Reproduced images displayed on a color mosaic screen are also continuous. The value on a point of subpixel images is chosen to be the color of the entire area of the subpixel. Displayed images are the images we want to compare with original images, so we need a comparison lattice to compare original images and displayed images.

For the RGB stripe display, full RGB images are the continuous original image sampled on the image capture lattice Λ_L. The subpixel images used to display on a low resolution screen are also defined on Λ_L. The comparison lattice Λ_c can be the lattice of full RGB images and subpixel images Λ_L, and can also be a sublattice of Λ_L. Samples on the comparison lattice are compared point by point to obtain the color difference between original and displayed images, shown in Figure 2.6.

2.1.9 Other Display Patterns

In other display patterns, the number of subpixels in one pixel varies, which can be four, five or more. Subpixels in modern displays can have different shapes such as triangles and

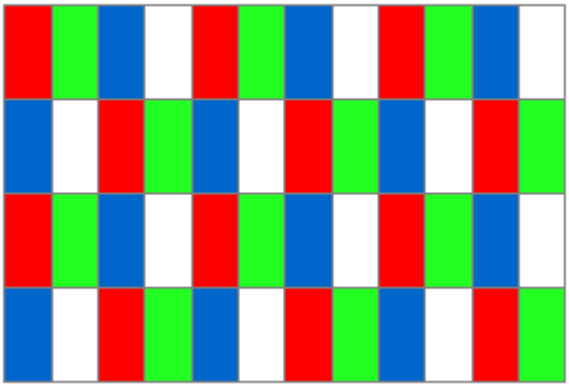


Figure 2.7: The Pentile RGBW display pattern

hexagons, for example, the display patterns in Figure 2.8 and 2.9. Modern displays also have different primary colors, not only the typical ones, but also others such as yellow and white. Input data of full color images needs to be converted into the displayed subpixel images on colored subpixel structures for other display patterns [21].

For example, the Pentile RGBW display pattern uses red, green, blue and white as its primary colors, shown in Figure 2.7. There are two subpixels in a pixel of the RGBW display pattern. In the horizontal direction, every two pixels contains four primary colors as subpixels. There is a shift to the right by one pixel location between two consecutive rows.

The Pentile RGBW display pattern is used in Motorola Atrix 4G phone [22]. It has better power efficiency and less energy cost by comparison with the typical RGB stripe display.

2.1.9.1 Pentile RGBG Display Pattern

The Pentile RGBG display pattern, shown in Figure 2.10, uses red, green and blue as its primary colors. In a row of Pentile RGBG display pattern, green subpixels are placed between red and blue subpixels. Green subpixels have half the size of red or blue ones. There is a shift of one green and one red subpixel locations to the right between every two

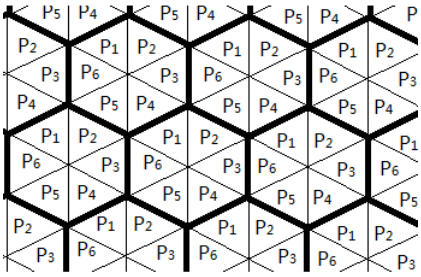


Figure 2.8: A display pattern with six subpixels in one pixel based on [2]

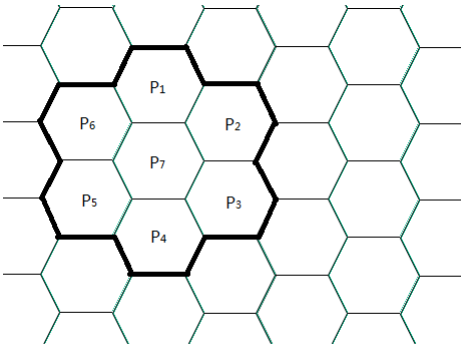


Figure 2.9: A display pattern with seven subpixels in one pixel based on [2]

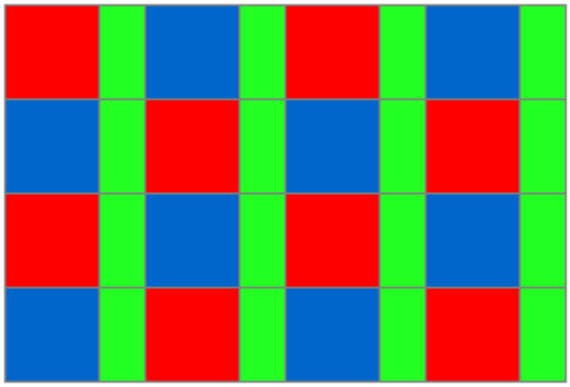


Figure 2.10: The Pentile RGBG display pattern

rows. Considering that the human eye has most sensitivity to green, the number of green subpixels is twice that of red or blue subpixels in Pentile RGBG display pattern.

By comparison with typical RGB stripe display pattern, the Pentile RGBG is better, considering the cost performance and power efficiency [22]. A wide varieties of phones use the Pentile RGBG as their display pattern, such as the Nokia Lumia 925, HTC One S, and Galaxy S 4 phones.

2.2 SCIELAB

2.2.1 Introduction

Zhang and Wandell introduced the SCIELAB color metric in [3]. We use the SCIELAB color metric to measure the color distortion in this research. The SCIELAB is an image metric measuring the perceptual color difference. It measures how accurate the color pattern of a processed image is to the original one for human observers. It is a spatial extension to the CIELAB color metric. In order to consider the spatial-color sensitivity of human eyes, SCIELAB adds a spatial pre-processing step to the standard CIELAB metric.

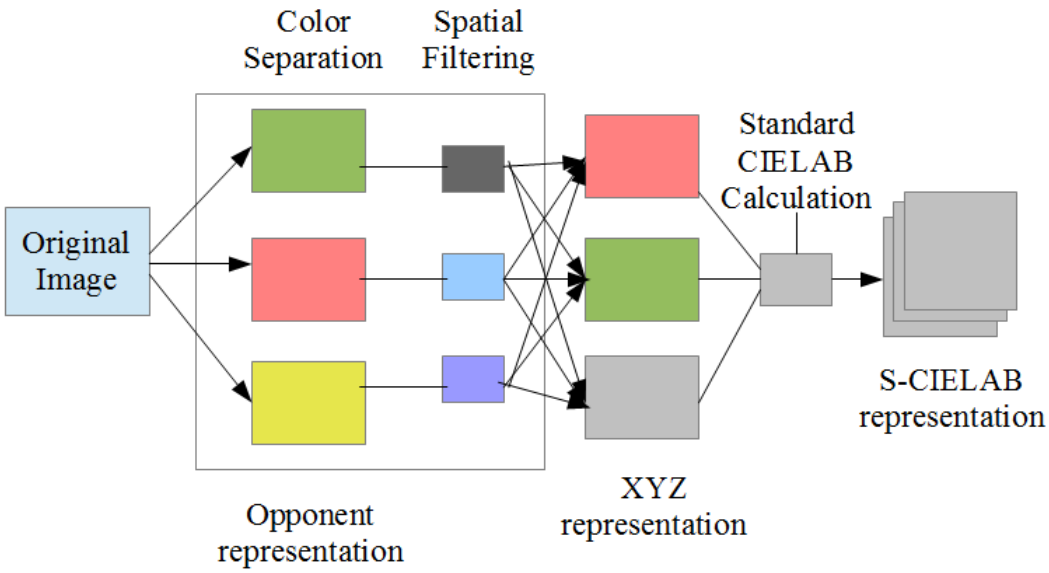


Figure 2.11: The SCIELAB model based on [3]

In this way, SCIELAB color metric removes the high-frequency pattern invisible to the human visual system, which better measures the perceptual color difference.

The SCIELAB calculation is illustrated in Figure 2.11. Key components of the calculation are color transformation and spatial filtering steps before the standard CIELAB calculation.

The first step of error calculation is to convert color images into an opponent color space. Then, it filters color channels of images using three filters, which simulate the spatial blurring effect of the human visual system. The shape of the filters is determined by the sensitivity of the human visual system to the color components. At last, the CIELAB color metric is used to calculate the error between the original and reproduced images in CIELAB representation.

To implement the SCIELAB color metric, we need the opponent color transformation to transfer the RGB images into three planes in the opponent color space. Using the opponent color space transformation, we need to convert the RGB image into the XYZ color space first, then do the opponent color space transformation as Formula 2.19.

Then, three spatial filters are necessary to filter the three planes in the opponent color space. In SCIELAB, the data of the transformation and spatial filters are obtained from

component i	term l	ω_{il}	σ_{il}	k_{il}
1	1	1.0033	0.0283	397.4
	2	0.1144	0.1330	17.99
	3	-0.1176	4.336	0.0169
2	1	0.6167	0.0392	207.1
	2	0.3833	0.4940	1.304
3	1	0.5681	0.0536	110.8
	2	0.4319	0.3860	2.136

Table 2.1: Parameters of the three SCIELAB filters [1]

human psychophysical measurements. The parameters of the filters used in the SCIELAB model are shown in Table 2.1.

The impulse response of three spatial filters acting as the low-pass filtering effect of human vision in one-dimensional case can be expressed in the equations below and are shown in Figure 2.12.

$$h_{ll}(x) = \sum_{i=1}^{I_l} \omega_{il} k_{il} e^{-\frac{x^2}{\sigma_{il}^2}},\ l = 1,2,3 \tag{2.24}$$

2.2.2 Advantages

Compared with the CIELAB color metric, the SCIELAB color metric can be used for photographic images, which are mostly made up of small fields, instead of large uniform fields. Using SCIELAB to calculate local color reproduction errors gives better results than using the CIELAB color metric.

The human visual system is considered pattern-color separable, so the transformation in SCIELAB is pattern-color separable. Using the SCIELAB opponent color space improves the computation efficiency.

Parameters used in the SCIELAB model are obtained from the psychophysical measurements. The SCIELAB metric well models the human visual system, so it is better

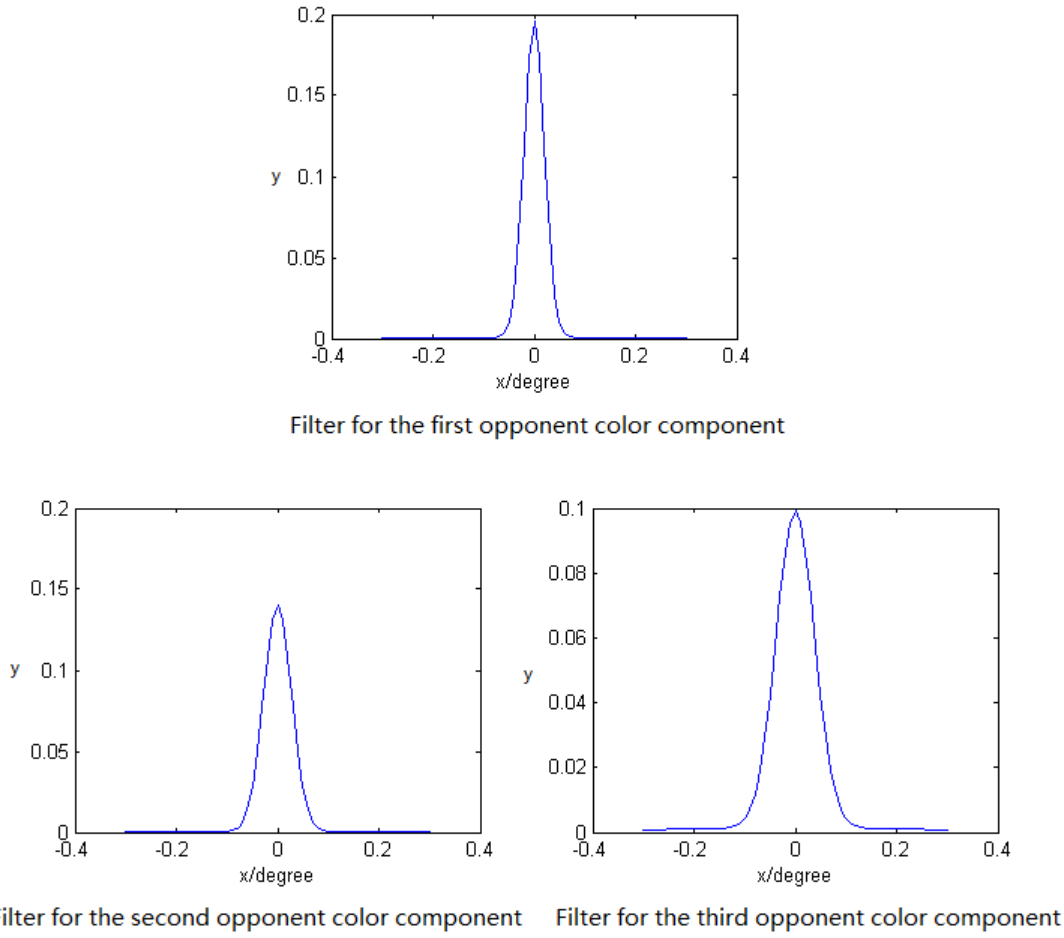


Figure 2.12: Three filters in the opponent color space used by the SCIELAB model [3]

to use SCIELAB to measure the perceptual color difference between original images and reproduced images.

2.3 Previous Work

2.3.1 Platt's Method

2.3.1.1 Introduction

Platt defines an error metric in the frequency domain and describes optimal filtering to obtain subpixel values from input images with high resolution [18]. The metric measures perceptual visual color errors. By minimizing the error metric, we can have a linear system of equations.

2.3.1.2 Formulation

Let $\mathbf{f}_L[x] = (f_{LR}[\mathbf{x}], f_{LG}[\mathbf{x}], f_{LB}[\mathbf{x}])$, $\mathbf{x} \in \Lambda_L$ be an image with high resolution and $\mathbf{f}_S^{platt}[\mathbf{x}]$, $\mathbf{x} \in \Lambda$ be the down-sampled image under the down-sampling ratio 3 : 1.

Define an error ε:

$$\varepsilon[\mathbf{x}] = \mathbf{M}(\mathbf{f}_S^{platt}[\mathbf{x}])^T - \sum_{j=1,2,3} \mathbf{C}(\mathbf{f}_L[\mathbf{x}+\mathbf{d}_j])^T, \mathbf{x} \in \Lambda \tag{2.25}$$

where $\mathbf{C} = \mathbf{A}_{RGB \to OPP}$, $\mathbf{M}_{kl} = 3\mathbf{C}_{k,lmod3}$ and

$$\mathbf{d}_j = \begin{cases} (0,0) & j = 1 \\ (\frac{d}{3},0) & j = 2 \\ (\frac{2d}{3},0) & j = 3 \end{cases} \tag{2.26}$$

Then,

$$\varepsilon[\mathbf{u}] = \mathscr{F}(\varepsilon[\mathbf{x}]) \tag{2.27}$$

$$\varepsilon[\mathbf{u}] = \sum_{\mathbf{u}\in\Lambda^*}\sum_{i} W_i[\mathbf{u}]\varepsilon_i[\mathbf{u}]\varepsilon_i^*[\mathbf{u}], i \in \{O_1, O_2, O_3\} \tag{2.28}$$

where $\mathscr{F}$ is the Fourier transform, W_i are frequency-weighting filters and $\mathbf{u} = (u, v)$. It is a frequency weighted mean square error.

The one-dimensional (1D) filters W_i for the RGB stripe display pattern used by Platt are

$$\begin{cases} W_1(u) = \min\{1, (8/u)^8\} \\ W_2(u) = \min\{1, (5/u)^4, W_1(u)\} \\ W_3(u) = \min\{1, (3/u)^4, W_1(u)\} \end{cases} \tag{2.29}$$

which are based on and different from the three filters used in SCIELAB (Figure 2.12).

So, the desired down-sampled image is:

$$\mathbf{f}_s^{platt}[\mathbf{x}] = argmin_{\mathbf{f}_s[\mathbf{x}]}\varepsilon[\mathbf{u}], \mathbf{x} \in \Lambda. \tag{2.30}$$

Platt's method is designed for improve quality of down-sampled fonts and used in Microsoft ClearType.

2.3.1.3 Network of Spatial Filters

The solution to the linear system in Platt's method is a set of nine filters, so we can use nine prefilters to process the input images before down-sampling, in order to get a better quality of reproduced images under a good color metric.

Platt treats the human visual system as a low-pass filter and use three filters in the frequency domain, which is based on psychophysical experiments. This method is only designed for the RGB stripe display pattern. Compared with Platt method, we want to use SCIELAB as the color metric and design a down-sampling system for general display patterns. In SCIELAB model, three spatial filters in the opponent color space are used to model the human visual system. Platt method inspires us to design a network of filters to prefilter color components before down-sampling to a color mosaic display and obtain better image quality under the SCIELAB model for photographic images as well as text images.

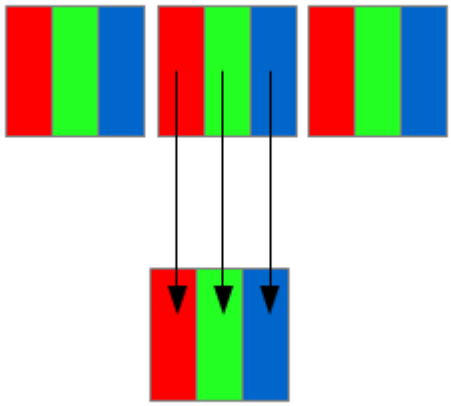


Figure 2.13: DPD based on [4]

2.3.2 DPD, DSD and DDSD

2.3.2.1 DPD

The Direct Pixel-based Down-sampling (DPD) [4] chooses one pixel out of every nine pixels, as shown in Figure 2.13. Also there are other two basic subpixel-based down-sampling scheme, the Direct Subpixel-based Down-sampling (DSD) [4] and Diagonal Direct Subpixel-based Down-sampling (DDSD) [5], which are illustrated in Figure 2.14 and Figure 2.15. If we input a high-resolution image $\mathbf{f}_L[\mathbf{x}]$ with $3M \times 3N$ pixels and down sample it to a low-resolution image $\mathbf{f}_S[\mathbf{x}]$ with $M \times N$ pixels, the component of the $(X,Y)th$ pixel is taken directly from the $(3X-2,3Y-2)th$ pixel of image $\mathbf{f}_L[\mathbf{x}]$.

Without anti-alias filters, color components of a subpixel image are:

$$\mathbf{f}_{SR}^{DPD}[\mathbf{x}] = \begin{cases} (f_{LR}[\mathbf{x}],0,0) & \mathbf{x} \in \Lambda_R \\ 0 & \mathbf{x} \in \Lambda_L \setminus \Lambda_R \end{cases} \tag{2.31}$$

$$\mathbf{f}_{SG}^{DPD}[\mathbf{x}] = \begin{cases} (0, f_{LG}[\mathbf{x} - \begin{bmatrix} d/3 \\ 0 \end{bmatrix}],0) & \mathbf{x} \in \Lambda_G \\ 0 & \mathbf{x} \in \Lambda_L \setminus \Lambda_G \end{cases} \tag{2.32}$$

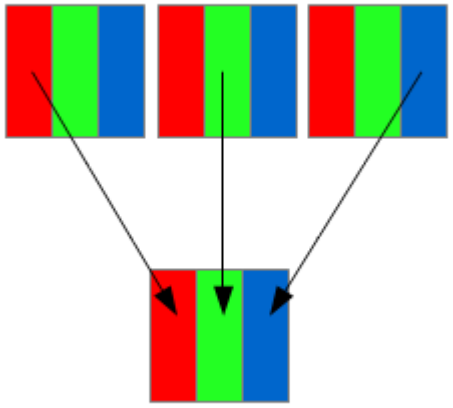


Figure 2.14: DSD based on [4]

$$\mathbf{f}_{SB}^{DPD}[\mathbf{x}] = \begin{cases} (0,0,f_{LB}[\mathbf{x} - \begin{bmatrix} 2d/3 \\ 0 \end{bmatrix}]) & \mathbf{x} \in \Lambda_B \\ 0 & \mathbf{x} \in \Lambda_L \setminus \Lambda_B \end{cases} \tag{2.33}$$

$$\mathbf{f}_{S}^{DPD}[\mathbf{x}] = \mathbf{f}_{SR}^{DPD}[\mathbf{x}] + \mathbf{f}_{SG}^{DPD}[\mathbf{x}] + \mathbf{f}_{SB}^{DPD}[\mathbf{x}], \mathbf{x} \in \Lambda_L \tag{2.34}$$

During the down-sampling process, no anti-aliasing filter is applied, which can lead to aliasing artifacts in the part of image with high spatial frequency.

2.3.2.2 DSD

DSD uses red, green and blue subpixels of different pixels in image $\mathbf{f}_L[\mathbf{x}]$ to generate a pixel in image $\mathbf{f}_S[\mathbf{x}]$. If $\mathbf{f}_S[X,Y]$ is the $(X,Y)th$ pixel in image $\mathbf{f}_S[\mathbf{x}]$ and $\mathbf{f}_L[X,Y]$ is the (X,Y) th pixel in image $\mathbf{f}_L[\mathbf{x}]$, then $f_{SR}[X,Y] = f_{LR}[3X-2,3Y-2], f_{SG}[X,Y] = f_{LG}[3X-2,3Y-1]$, and $f_{SB}[X,Y] = f_{LB}[3X-2,3Y]$. This method takes use of the subpixel structure of display, however, only the horizontal direction is considered.

Without anti-alias filter, color components of a subpixel image are:

$$\mathbf{f}_{SR}^{DSD}[\mathbf{x}] = \begin{cases} (f_{LR}[\mathbf{x}],0,0) & \mathbf{x} \in \Lambda_R \\ 0 & \mathbf{x} \in \Lambda_L \setminus \Lambda_R \end{cases} \tag{2.35}$$

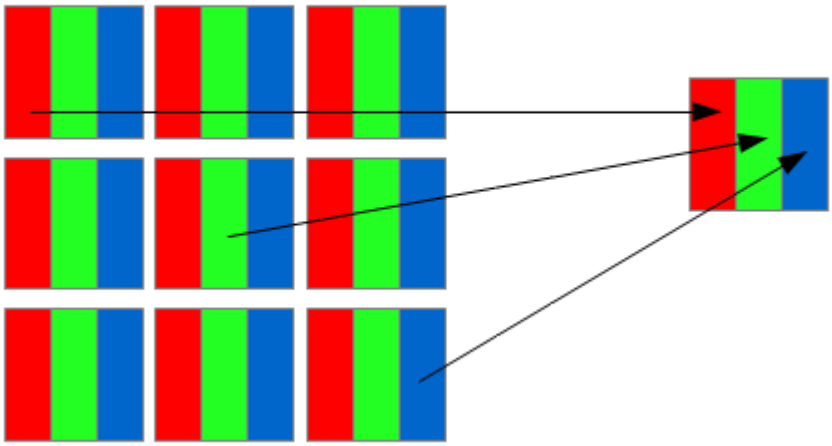


Figure 2.15: DDSD based on [5]

$$\mathbf{f}_{SG}^{DSD}[\mathbf{x}] = \begin{cases} (0, f_{LG}[\mathbf{x}], 0) & \mathbf{x} \in \Lambda_G \\ 0 & \mathbf{x} \in \Lambda_L \setminus \Lambda_G \end{cases} \tag{2.36}$$

$$\mathbf{f}_{SB}^{DSD}[\mathbf{x}] = \begin{cases} (0, 0, f_{LB}[\mathbf{x}]) & \mathbf{x} \in \Lambda_B \\ 0 & \mathbf{x} \in \Lambda_L \setminus \Lambda_B \end{cases} \tag{2.37}$$

$$\mathbf{f}_{S}^{DSD}[\mathbf{x}] = \mathbf{f}_{SR}^{DSD}[\mathbf{x}] + \mathbf{f}_{SG}^{DSD}[\mathbf{x}] + \mathbf{f}_{SB}^{DSD}[\mathbf{x}], \mathbf{x} \in \Lambda_L \tag{2.38}$$

Because of more reconstruction points used, DSD could keep more details than DPD. DSD does not consider the vertical direction, so quality of down-sampled images with horizontal edges is not improved.

2.3.2.3 DDSD

DDSD selects subpixels of RGB primary colors from pixels in the diagonal direction. For the (X,Y)th pixel in $\mathbf{f}_S$, the red, green and blue component of the pixel is $f_{SR}[X,Y] = f_{LR}[3X-2, 3Y-2], f_{SG}[X,Y] = f_{LG}[3X-1, 3Y-1]$, and $f_{SB}[X,Y] = f_{LB}[3X, 3Y]$ [5].

For DPD, DSD and DDSD methods, the filters for prefiltering in spatial domain can be considered as:

$$h_{RG}[\mathbf{x}] = h_{BG}[\mathbf{x}] = h_{GR}[\mathbf{x}] = \mathbf{0}$$
$$h_{RB}[\mathbf{x}] = h_{GB}[\mathbf{x}] = h_{BR}[\mathbf{x}] = \mathbf{0}$$
$$h_{RR}[\mathbf{x}] = h_{BB}[\mathbf{x}] = h_{GG}[\mathbf{x}] = \mathbf{1} \tag{2.39}$$

2.3.3 MMSE-SD and MMDE

2.3.3.1 MMSE-SD

MMSE Subpixel-based Down-sampling (MMSE-SD) method [5] formulates the problem by minimizing a mean square error between original images and down-sampled images. The solution is to filter the result of DDSD by a 2D linear filter.

MMSE-SD method treats the problem as minimizing a mean square error. For example, for red component, the problem is find $\min_{f_{SR}[\mathbf{x}]} \parallel f_{LR}[\mathbf{x}] - f'_{LR}[\mathbf{x}] \parallel_2^2$, so that $f'_{LR}[\mathbf{x}] = q_r(f_{SR}[\mathbf{x}])$, where q_r is a reconstruction function. Differentiate $\parallel f_{LR}[\mathbf{x}] - f'_{LR}[\mathbf{x}] \parallel_2^2$ with respect to $f_{SR}[\mathbf{x}]$ and set it to zero, we can have $M \times N$ equations as matrix form: $\mathbf{H}_{R1} \cdot f_{SR}[\mathbf{x}] = \mathbf{H}_{R2} \cdot f_{LR}[\mathbf{x}]$, where $\mathbf{H}_{R1}$ is a $MN \times MN$ matrix and $\mathbf{H}_{R2}$ is a $MN \times 9MN$ matrix. The solution is $f_{SR}[\mathbf{x}] = (\mathbf{H}_{\mathrm{R1}}{}^{-\mathbf{1}}\mathbf{H}_{\mathrm{R2}}) f_{LR}[\mathbf{x}]$.

Each color plane is independently optimized to minimize MSE between the down-sampled image $\mathbf{f}'_L[\mathbf{x}]$ and the original image $\mathbf{f}_L[\mathbf{x}]$.

$$f_{Sk}^{MMSE}[\mathbf{x}] = \underset{f_{Sk}[\mathbf{x}]}{argmin} \sum_{\mathbf{x} \in \Lambda_L} (f_{Lk}[\mathbf{x}] - f'_{Lk}[\mathbf{x}])^2 \tag{2.40}$$

$$k \in \{R, G, B\}$$

The method can be formulated as using nine prefilters:

$$h_{RG}[\mathbf{x}] = h_{BG}[\mathbf{x}] = h_{GR}[\mathbf{x}] = \mathbf{0}$$
$$h_{RB}[\mathbf{x}] = h_{GB}[\mathbf{x}] = h_{BR}[\mathbf{x}] = \mathbf{0}$$
$$h_{kk}[\mathbf{x}] = h_{kMMSE}[\mathbf{x}], k \in \{R, G, B\} \tag{2.41}$$

where $h_{kMMSE}[\mathbf{x}]$ is the $\frac{MN+1}{2}$th row of the matrix $\mathbf{H}_{\mathrm{k1}}^{-1}\mathbf{H}_{\mathrm{k2}}$.

2.3.3.2 MMDE

Down-Sampling by minimizing a Min-Max Directional Error (MMDE) is introduced in [4]. The minimizing problem can be convexified and further simplified to solve efficiently. Two relaxation methods are performed during the down-sampling process, the MMDE with Visual Relaxation and the MMDE with Edge Relaxation.

In this approach, $\mathbf{f}_S^{MMDE}[\mathbf{x}]$ is chosen to minimize a certain error function. It is assumed that $\mathbf{f}_S^{MMDE}[\mathbf{x}]$ is interpolated to Λ_L using a simple averaging filter on each of the three components. Then, squared errors in four directions are computed and the maximum error is minimized. This is done separately on three components.

The interpolated image is $\mathbf{f}_L^{'}[\mathbf{x}]$, and it is obtained by a separate filtering of $\mathbf{f}_S[\mathbf{x}]$, assumed to be defined on Λ_L.

$$\mathbf{f}_{Su}[\mathbf{x}] = \begin{cases} \mathbf{f}_S[\mathbf{x}] & \mathbf{x} \in \Lambda \\ 0 & \mathbf{x} \in \Lambda_L \setminus \Lambda \end{cases} \tag{2.42}$$

$$\mathbf{f}_L^{'}[\mathbf{x}] = \mathbf{f}_{Su}[\mathbf{x}] * h_3[\mathbf{x}] \tag{2.43}$$

where, h_3 is a scalar filter, applied independently to each color plane.

$$h_3[\mathbf{x}] = \begin{bmatrix} \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{2}{3} & \frac{2}{3} & \frac{2}{3} & 0 \\ 0 & \frac{2}{3} & 1 & \frac{2}{3} & \frac{1}{3} \\ 0 & \frac{2}{3} & \frac{2}{3} & \frac{2}{3} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} \end{bmatrix}$$

Each sample of $\mathbf{f}_S[\mathbf{x}]$ contributes to five samples in $\mathbf{f}_L^{'}[\mathbf{x}]$ in each of four principle directions. To define the directional measure, 4 directional subsets of Λ_L are defined:

Horizontal:

$$D_1 = \{ \begin{bmatrix} -\frac{2d}{3} \\ 0 \end{bmatrix} \begin{bmatrix} -\frac{d}{3} \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} \frac{d}{3} \\ 0 \end{bmatrix} \begin{bmatrix} \frac{2d}{3} \\ 0 \end{bmatrix} \} \tag{2.44}$$

Vertical:

$$D_2 = \{ \begin{bmatrix} 0 \\ -\frac{2d}{3} \end{bmatrix} \begin{bmatrix} 0 \\ -\frac{d}{3} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ \frac{d}{3} \end{bmatrix} \begin{bmatrix} 0 \\ \frac{2d}{3} \end{bmatrix} \} \tag{2.45}$$

Diagonal:

$$D_3 = \{ \begin{bmatrix} -\frac{2d}{3} \\ -\frac{2d}{3} \end{bmatrix} \begin{bmatrix} -\frac{d}{3} \\ -\frac{d}{3} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} \frac{d}{3} \\ \frac{d}{3} \end{bmatrix} \begin{bmatrix} \frac{2d}{3} \\ \frac{2d}{3} \end{bmatrix} \} \tag{2.46}$$

Anti-diagonal:

$$D_4 = \{ \begin{bmatrix} -\frac{2d}{3} \\ \frac{2d}{3} \end{bmatrix} \begin{bmatrix} -\frac{d}{3} \\ \frac{d}{3} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} \frac{d}{3} \\ -\frac{d}{3} \end{bmatrix} \begin{bmatrix} \frac{2d}{3} \\ -\frac{2d}{3} \end{bmatrix} \} \tag{2.47}$$

Then, four squared-error terms are:

$$E_k^i[\mathbf{x}] = \sum_{\mathbf{s} \in D_i} (f_{Lk}[\mathbf{x}-\mathbf{s}] - f_{Lk}^{'}[\mathbf{x}-\mathbf{s}])^2, \mathbf{x} \in \Lambda_k \tag{2.48}$$
$$k \in \{R, G, B\}$$

The proposed optimal $\mathbf{f}_{Sk}[\mathbf{x}]$ is:

$$f_{Sk}^{MMDE}[\mathbf{x}] = \underset{f_{Sk}[\mathbf{x}]}{argmin} \sum_{\mathbf{x} \in \Lambda_k} \max_i (E_k^i[\mathbf{x}]) \tag{2.49}$$
$$k \in \{R, G, B\}$$

After relaxation, the method can be formulated using nine prefilters:

$$\begin{aligned} h_{RG}[\mathbf{x}] = h_{BG}[\mathbf{x}] = h_{GR}[\mathbf{x}] = \mathbf{0} \\ h_{RB}[\mathbf{x}] = h_{GB}[\mathbf{x}] = h_{BR}[\mathbf{x}] = \mathbf{0} \\ h_{RR}[\mathbf{x}] = h_{BB}[\mathbf{x}] = h_{GG}[\mathbf{x}] = h_{MMDE}[\mathbf{x}] \end{aligned} \tag{2.50}$$

where

$$\begin{aligned} h_{MMDE}[\mathbf{x}] = &(0.0017, 0.0017, -0.0056, -0.0129, -0.0202, 0.0580, 0.1361, \\ &0.2143, 0.1361, 0.0580, -0.0202, -0.0129, -0.0056, 0.0017, 0.0017) \end{aligned} \tag{2.51}$$

2.3.4 Gibson's Method

The width of images are tripled in the Gibson's method. Then a five-tap filter $h_{Gilbson}[\mathbf{x}]$ are used to convolve with the image to spread the color energy. Then, the image are down sample at the ratio 3 : 1 [4].

$$\mathbf{f}_L[\mathbf{x}] = (f_{LR}[\mathbf{x}], f_{LG}[\mathbf{x}], f_{LB}[\mathbf{x}]), \mathbf{x} \in \Lambda_L, \tag{2.52}$$

Down-sample $\mathbf{f}_L[\mathbf{x}]$ with a basic down sampling scheme such as DPD and DSD. Then, we have $\mathbf{f}_{s0}[\mathbf{x}], \mathbf{x} \in \Lambda_L$.

The down-sampled image is filtered by:

$$h_{Gilbson}[\mathbf{x}] = (\frac{1}{9}, \frac{2}{9}, \frac{3}{9}, \frac{2}{9}, \frac{1}{9}) \tag{2.53}$$

Then, the desired subpixel image is:

$$\mathbf{f}_s^{Gibson}[\mathbf{x}] = \mathbf{f}_{s0}[\mathbf{x}] * h_{Gilbson}[\mathbf{x}], \mathbf{x} \in \Lambda_L. \tag{2.54}$$

The filters used in Gibson's method can be formulated as nine filters between each color channel:

$$\begin{aligned} h_{RG}[\mathbf{x}] = h_{BG}[\mathbf{x}] = h_{GR}[\mathbf{x}] = \mathbf{0} \\ h_{RB}[\mathbf{x}] = h_{GB}[\mathbf{x}] = h_{BR}[\mathbf{x}] = \mathbf{0} \\ h_{RR}[\mathbf{x}] = h_{BB}[\mathbf{x}] = h_{GG}[\mathbf{x}] = h_{Gilbson}[\mathbf{x}] \end{aligned} \tag{2.55}$$

2.4 Summary

In this chapter, definitions and the SCIELAB metric for formulating subpixel-based down-sampling problems are introduced. Previous work by other researchers in subpixel rendering is introduced and can be reformulated using a set of simple prefilters. Most existing methods are only designed for the RGB stripe display pattern and do not use the SCIELAB metric to measure color differences. The prefiltering before down-sampling of most existing methods is on RGB values using only three filters $h_{RR}[\mathbf{x}], h_{BB}[\mathbf{x}]$ and $h_{GG}[\mathbf{x}]$,

and the other six filters are $\mathbf{0}$. Thus, a set of optimized filters needs to be obtained for general display patterns through the down-sampling system design.

Chapter 3

Down-sampling System Design

Subpixel rendering methods can be formulated using a set of prefilters in the spatial domain, for example, a set of nine filters in conventional subpixel-based down-sampling methods. The good color metric, SCIELAB, is not used to measure color differences in most existing methods. Thus, a set of optimized spatial filters needs to be designed in order to minimize SCIELAB errors. The system structure and mathematical solutions of the new approach to subpixel-based down-sampling will be described in this chapter.

3.1 Optimization System Structure

The general process of the new approach is shown in Figure 3.1. Color channel of original images are prefiltered by a set of filters and then are down-sampled using one of the basic subpixel-based down-sampling schemes, for example, DSD. Original full color images and subpixel images are compared using the SCIELAB metric. Projection theory is use to minimize frequency weighted errors between original images and subpixel images in the opponent color space in order to attempt to minimize the SCIELAB errors.

Experimental data indicates that the relation between frequency weighted errors and SCIELAB errors are approximately a linear relation. Figure 3.2 shows a plot of frequency weighted errors and SCIELAB errors of twenty testing images.

3.1.1 Inner Product and Norm

Let $\mathbf{f}$ and $\mathbf{g}$ be two color images, defined on a space lattice Λ_L, with the unit of distance measured in visual angle. Let the vector space of such images be $\mathscr{I}$. Assume that the two

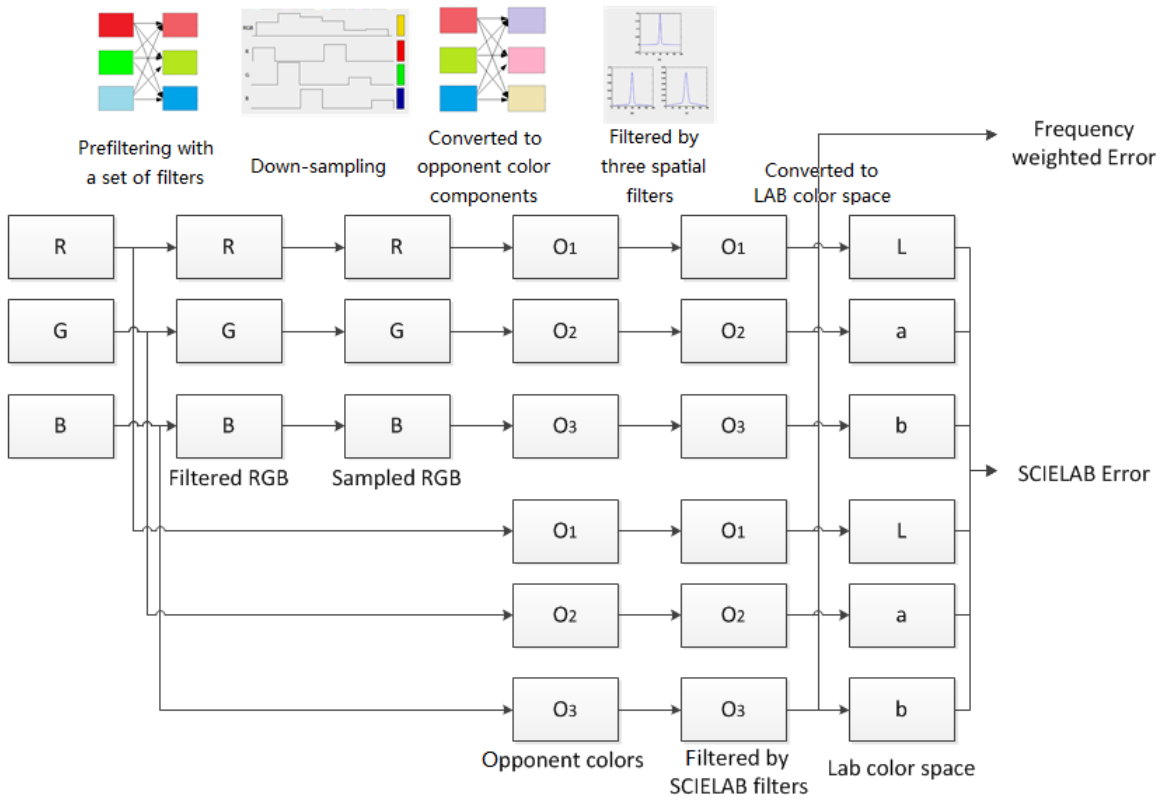


Figure 3.1: The process of down-sampling system design

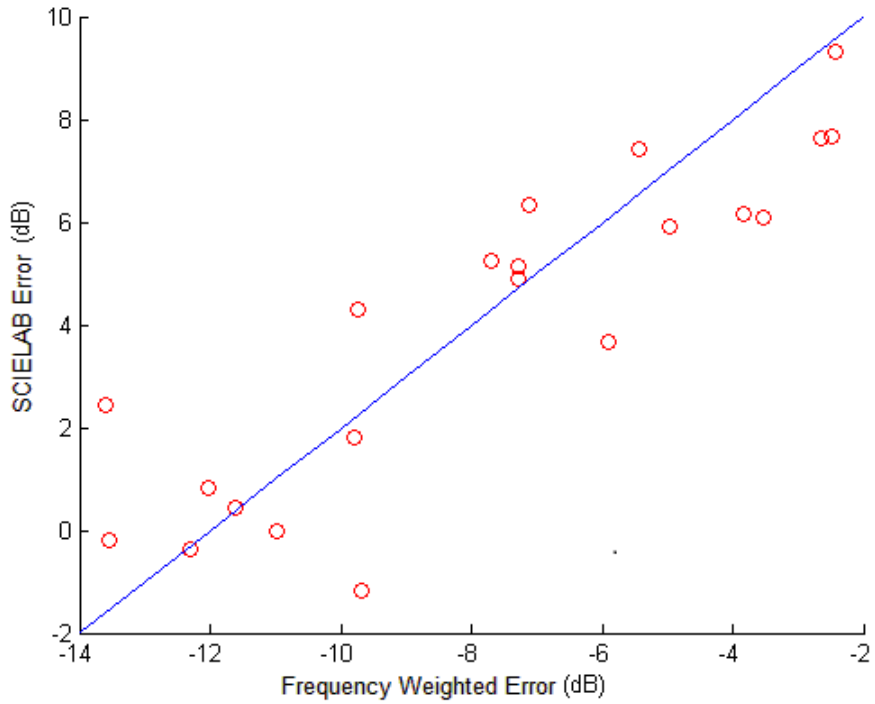


Figure 3.2: The relation between frequency weighted errors and SCIELAB errors

images are represented in the SCIELAB opponent space $B_O = \{[O_1], [O_2], [O_3]\}$. Let the components be $f_{O_1}, f_{O_2}, f_{O_3}$ and $g_{O_1}, g_{O_2}, g_{O_3}$. We define an inner product for color images defined on Λ_L:

$$< \mathbf{f} \mid \mathbf{g} >_O = \sum_{k=1}^{3} \sum_{\mathbf{x}=\Lambda_L} (f_{Ok} * h_{Ok})[\mathbf{x}](g_{Ok} * h_{Ok})[\mathbf{x}] \tag{3.1}$$

$h_{Ok}, k = 1,2,3$ are the weighting filters in the three opponent channels, which are defined in terms of spatial units of visual angle.

The corresponding norm is

$$\| \mathbf{f} \|_O^2 = \sum_{k=1}^{3} \sum_{\mathbf{x}=\Lambda_L} ((f_{Ok} * h_{Ok})[\mathbf{x}])^2 \tag{3.2}$$

This satisfies all the properties of an inner product and norm. Thus, we measure the difference of two color images by $\| \mathbf{f} - \mathbf{g} \|_O$

3.1.2 Optimization

Consider an RGB subpixel display, with equal numbers of R,G,B subpixels. Thus, let Λ be a sublattice of Λ_L of index 3. Let $\mathbf{d}_1 = \mathbf{0}, \mathbf{d}_2, \mathbf{d}_3$ be coset representatives for the three cosets of Λ in Λ_L. The three cosets correspond to R,G,B subpixels.

We define a subspace $\mathscr{M}$ of $\mathscr{I}$ of subpixel images, i.e.

$$f_G[\mathbf{x}] = f_B[\mathbf{x}] = 0, \text{ for } \mathbf{x} \in \Lambda + \mathbf{d}_1$$

$$f_R[\mathbf{x}] = f_B[\mathbf{x}] = 0, \text{ for } \mathbf{x} \in \Lambda + \mathbf{d}_2$$

$$f_R[\mathbf{x}] = f_G[\mathbf{x}] = 0, \text{ for } \mathbf{x} \in \Lambda + \mathbf{d}_3 \tag{3.3}$$

Given $\mathbf{f} \in \mathscr{I}$, we seek the element in $\mathscr{M}$ to best approximate $\mathbf{f}$:

$$\mathbf{f}_s = \underset{\mathbf{g} \in \mathscr{M}}{argmin} \| \mathbf{f} - \mathbf{g} \|_O \tag{3.4}$$

This is given by the projection of $\mathbf{f}$ on $\mathscr{M}$, and it satisfies the orthogonality principle:

$$\mathbf{f} - \mathbf{f}_s \perp \mathscr{M} \tag{3.5}$$

This yields the normal equations to be solved.

3.2 Filter Design

3.2.1 Conversion of Full RGB Images

The RGB stripe display pattern is used as an example to mathematically describe the process of down-sampling system design. After undoing the gamma correction, we have the original full color RGB image $\mathbf{f}_L[\mathbf{x}] = (f_{RL}[\mathbf{x}], f_{GL}[\mathbf{x}], f_{BL}[\mathbf{x}]), \mathbf{x} \in \Lambda_L$, where Λ_L is the same subpixel lattice of the RGB stripe display pattern as we described in Chapter 2.

First, we convert color coordinates of the full RGB color image into the opponent color space used in the SCIELAB color metric:

$$\sum_i a_{ij} f_{iL}[\mathbf{x}] = f_{jL}[\mathbf{x}] \tag{3.6}$$

$$i \in \{R, G, B\}, j \in \{O_1, O_2, O_3\}$$

where values of a_{ij} are from the color coordinate conversion matrix converting colors from RGB color space to the opponent color space:

$$\mathbf{A}^T_{RGB \to O_1O_2O_3} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{3.7}$$

$$\begin{bmatrix} f'_{O_1L}[\mathbf{x}] \\ f'_{O_2L}[\mathbf{x}] \\ f'_{O_3L}[\mathbf{x}] \end{bmatrix} = \mathbf{A}_{RGB \to O_1O_2O_3} \cdot \begin{bmatrix} f_{RL}[\mathbf{x}] \\ f_{GL}[\mathbf{x}] \\ f_{BL}[\mathbf{x}] \end{bmatrix} \tag{3.8}$$

After the color coordinate conversion, we have components of the full RGB color image in the SCIELAB opponent color space:

$$\mathbf{f}_L^{'}[\mathbf{x}] = (f_{O_1L}[\mathbf{x}], f_{O_2L}[\mathbf{x}], f_{O_3L}[\mathbf{x}]). \tag{3.9}$$

Let $h_j[\mathbf{x}], j = O_1, O_2, O_3$ be the spatial filters for three opponent color channels in SCIELAB. Then filter each color component of $\mathbf{f}_L^{'}[\mathbf{x}]$ with these three filters. We have the frequency-weighted opponent color components $\mathbf{f}_{LH}^{'}[\mathbf{x}]$. Opponent color components are defined as $f_{O_1LH}[\mathbf{x}]$, $f_{O_2LH}[\mathbf{x}]$ and $f_{O_3LH}[\mathbf{x}]$ individually.

$$\begin{aligned} \mathbf{f}_{LH}^{'}[\mathbf{x}] &= (f_{O_1L}[\mathbf{x}] * h_{O_1}[\mathbf{x}], f_{O_2L}[\mathbf{x}] * h_{O_2}[\mathbf{x}], f_{O_3L}[\mathbf{x}] * h_{O_3}[\mathbf{x}]) \\ &= (f_{O_1LH}[\mathbf{x}], f_{O_2LH}[\mathbf{x}], f_{O_3LH}[\mathbf{x}]). \end{aligned} \tag{3.10}$$

3.2.2 Displayed Subpixel Images

After obtaining frequency-weighted opponent color components $\mathbf{f}_{LH}^{'}[\mathbf{x}]$, a displayed subpixel image is created in the same sampling structure in order to compare with the original full color image $\mathbf{f}_L[\mathbf{x}], \mathbf{x} \in \Lambda_L$. Let $h_{ki}[\mathbf{x}], k, i = R, G, B$ be nine spatial filters we want to optimize. Then, filter RGB color components of the original full color image $\mathbf{f}_L[\mathbf{x}]$. We have:

$$f_{iF}[\mathbf{x}] = \sum_k h_{ki}[\mathbf{x}] * f_{kL}[\mathbf{x}] = \sum_k \sum_{\mathbf{l} \in \Lambda_L} h_{ki}[\mathbf{l}] f_{kL}[\mathbf{x} - \mathbf{l}] \tag{3.11}$$

$$i, k \in \{R, G, B\}$$

where $f_{RF}[\mathbf{x}]$, $f_{GF}[\mathbf{x}]$ and $f_{BF}[\mathbf{x}]$ are RGB color components of the prefiltered original full color image. The prefiltered image is:

$$\mathbf{f}_F[\mathbf{x}] = (f_{RF}[\mathbf{x}], f_{GF}[\mathbf{x}], f_{BF}[\mathbf{x}]), \mathbf{x} \in \Lambda_L \tag{3.12}$$

After filtering the original RGB color image, a basic down-sampling method, for example, DSD is used to down-sample the prefiltered full color image. Let $f_D[\mathbf{x}]$ be the down-sampled image. The down-sampled image is defined on Λ_L:

$$\mathbf{f}_D[\mathbf{x}] = \begin{cases} (f_{RF}(\mathbf{x}),0,0) & \mathbf{x}\in\Lambda_R \\ (f_{RF}(\mathbf{x}-(0,\frac{d}{3})),0,0) & \mathbf{x}\in\Lambda_R+(0,\frac{d}{3}) \\ (f_{RF}(\mathbf{x}-(0,\frac{2d}{3})),0,0) & \mathbf{x}\in\Lambda_R+(0,\frac{2d}{3}) \\ (0,f_{GF}(\mathbf{x}),0) & \mathbf{x}\in\Lambda_G \\ (0,f_{GF}(\mathbf{x}-(0,\frac{d}{3})),0) & \mathbf{x}\in\Lambda_G+(0,\frac{d}{3}) \\ (0,f_{GF}(\mathbf{x}-(0,\frac{2d}{3})),0) & \mathbf{x}\in\Lambda_G+(0,\frac{2d}{3}) \\ (0,0,f_{BF}(\mathbf{x})) & \mathbf{x}\in\Lambda_B \\ (0,0,f_{BF}(\mathbf{x}-(0,\frac{d}{3}))) & \mathbf{x}\in\Lambda_B+(0,\frac{d}{3}) \\ (0,0,f_{BF}(\mathbf{x}-(0,\frac{2d}{3}))) & \mathbf{x}\in\Lambda_B+(0,\frac{2d}{3}) \end{cases}$$

The down-sampled image $\mathbf{f}_D[\mathbf{x}]$ is the subpixel image we want to display on low-resolution display devices. The RGB color components $\mathbf{f}_D[\mathbf{x}]$ of are denoted as $f_{RD}[\mathbf{x}]$, $f_{GD}[\mathbf{x}]$ and $f_{BD}[\mathbf{x}]$.

Then, the down-sampled image $\mathbf{f}_D[\mathbf{x}]$ is converted to the same opponent color space on Λ_L as the original full color image, and then filtered with the spatial filters, $h_{O1}[\mathbf{x}], h_{O2}[\mathbf{x}]$ and $h_{O3}[\mathbf{x}]$, in the SCIELAB opponent color space on Λ_L, in order to compare the images in the frequency-weighted opponent color space. We convert color coordinates of the filtered image $\mathbf{f}_D[\mathbf{x}]$ to the opponent color space:

$$\sum_i a_{ij} f_{iD}[\mathbf{x}] = f'_{jD}[\mathbf{x}], \tag{3.13}$$

$$i \in \{R,G,B\}, j \in \{O_1,O_2,O_3\}$$

So, we have the subpixel image in the opponent color space:

$$\mathbf{f}'_D[\mathbf{x}] = (f_{O_1D}[\mathbf{x}], f_{O_2D}[\mathbf{x}], f_{O_3D}[\mathbf{x}]). \tag{3.14}$$

Filter each opponent color component of $\mathbf{f}'_D[\mathbf{x}]$ with three SCIELAB filters.

$$\mathbf{f}'_{DH}[\mathbf{x}] = (f_{O_1DH}[\mathbf{x}] * h_{O_1}[\mathbf{x}], f_{O_2DH}[\mathbf{x}] * h_{O_2}[\mathbf{x}], f_{O_3DH}[\mathbf{x}] * h_{O_3}[\mathbf{x}]). \tag{3.15}$$

Then, we have the frequency-weighted opponent color components $\mathbf{f}'_{DH}[\mathbf{x}]$. Each opponent color components are denoted as $f_{O_1DH}[\mathbf{x}]$, $f_{O_2DH}[\mathbf{x}]$ and $f_{O_3DH}[\mathbf{x}]$ individually.

3.2.3 Minimization of the Squared Error

After we have both original full color images and subpixel images in the frequency-weighted opponent color space, we can minimize the squared error between $\mathbf{f}'_{DH}[\mathbf{x}]$ and $\mathbf{f}'_{LH}[\mathbf{x}]$ to optimize the nine filters we want.

Assume we have M images in a training set, so the optimal filters on this training set are:

$$[h^{opt}_{RR}, h^{opt}_{GR}, ..., h^{opt}_{BB}] = argmin_{h_{ki}} \sum_{\mathbf{x}\in L_m} \| \mathbf{f}'^{(m)}_{DH}[\mathbf{x}] - \mathbf{f}'^{(m)}_{LH}[\mathbf{x}] \|_2 \tag{3.16}$$

$$m = 1, ..., M.$$

where L_m is the subpixel sampling structure of mth image.

3.3 Mathematical Solution

3.3.1 General Display Patterns

A color mosaic display may have multiple primary colors $P_1, P_2, ...P_n$, so a set of $3 \times n$ filters are required to prefilter the original full color image $\mathbf{f}_L[\mathbf{x}]$, if the original image is an RGB image. Then, we have color components of each primary colors on the color mosaic display as:

$$\begin{aligned} f_{iF}[\mathbf{x}] &= \sum_{k\in\{R,G,B\}} h_{ki}[\mathbf{x}] * f_{kL}[\mathbf{x}], i \in \{P_1, P_2, ...P_N\} \\ &= \sum_{k\in\{R,G,B\}} \sum_{\mathbf{l}\in\Lambda_L} h_{ki}[\mathbf{l}] f_{kL}[\mathbf{x}-\mathbf{l}], \mathbf{x} \in \Lambda_L. \end{aligned} \tag{3.17}$$

Let $\delta_{P_n}[\mathbf{x}]$ be the sampling structure for primary color P_n:

$$\delta_{P_n}[\mathbf{x}] = \begin{cases} 1 & \mathbf{x} \in \Lambda_{P_n} \\ 0 & \mathbf{x} \in \Lambda_L \setminus \Lambda_{P_n} \end{cases} \tag{3.18}$$

Down-sample the $\mathbf{f}_F[\mathbf{x}]$:

$$\begin{aligned} f_{iD}[\mathbf{x}] &= f_{iF}[\mathbf{x}]\delta_i[\mathbf{x}] \\ &= \delta_i[\mathbf{x}] \sum_{k\in\{R,G,B\}} \sum_{\mathbf{l}\in\Lambda_L} h_{ki}[\mathbf{l}] f_{kL}[\mathbf{x}-\mathbf{l}] \end{aligned} \tag{3.19}$$

$$i \in \{P_1, P_2, ...P_N\}.$$

After down-sampling, converting the down-sampled image into the SCIELAB opponent color space:

$$f'_{jD}[\mathbf{x}] = \sum_{i\in\{P_1,P_2,...P_N\}} a_{ij}\delta_i[\mathbf{x}] \sum_{k\in\{R,G,B\}} \sum_{\mathbf{l}\in\Lambda_L} h_{ki}[\mathbf{l}] f_{kL}[\mathbf{x}-\mathbf{l}], j \in \{O_1, O_2, O_3\} \tag{3.20}$$

Filter color components of $\mathbf{f}'_D[\mathbf{x}]$ using the three spatial filters in SCIELAB:

$$\begin{aligned} f'_{jDH}[\mathbf{x}] &= f'_{jD}[\mathbf{x}] * h_j[\mathbf{x}] \\ &= \sum_{\mathbf{t}\in\Lambda_L} h_j[\mathbf{t}] f'_{jD}[\mathbf{x}-\mathbf{t}] \\ &= \sum_{\mathbf{t}\in\Lambda_L} h_j[\mathbf{t}] \sum_i a_{ij}\delta_i[\mathbf{x}-\mathbf{t}] \sum_k \sum_{\mathbf{l}\in\Lambda_L} h_{ki}[\mathbf{l}] f_{kL}[\mathbf{x}-\mathbf{t}-\mathbf{l}]. \end{aligned} \tag{3.21}$$

$$i \in \{P_1, P_2, ...P_N\}, k \in \{R, G, B\}$$

Comparing original images and the displayed images in the frequency-weighted opponent color space, we can have the color difference we want to minimize for a color mosaic display pattern:

$$\begin{aligned}\mathscr{E}(\mathbf{h}) &= \sum_{m=1}^{M}\sum_{\mathbf{x}\in L_m}\sum_{j\in\{O_1,O_2,O_3\}}(f'^{(m)}_{jDH}[\mathbf{x}]-\frac{1}{c}f'^{(m)}_{jLH}[\mathbf{x}])^2 \\ &= \sum_{m=1}^{M}\sum_{\mathbf{x}\in L_m}\sum_{j\in\{O_1,O_2,O_3\}}(\sum_{\mathbf{t}\in\Lambda_L}h_j[\mathbf{t}]\sum_{i\in\{P_1,P_2,\ldots P_N\}}a_{ij}\delta_i[\mathbf{x}-\mathbf{t}](\\ &\quad \sum_{k\in\{R,G,B\}}\sum_{\mathbf{l}\in\Lambda_L}h_{ki}[\mathbf{l}]f^{(m)}_{kL}[\mathbf{x}-\mathbf{t}-\mathbf{l}]-\frac{1}{c}f'^{(m)}_{jLH}[\mathbf{x}]))^2. \end{aligned} \tag{3.22}$$

where c is the down-sampling ratio. It is 3 for the RGB and Pentile RGBG display pattern.

3.3.2 RGB Stripe Display Pattern

The RGB stripe display pattern is used as a model to analyze the new approach. The solution of least-squares filter design for the RGB stripe display pattern is given in this subsection.

Filter the original full RGB image $\mathbf{f}_L[\mathbf{x}]$ using nine filters we want to design for the RGB stripe display pattern:

$$\begin{aligned} f_{iF}[\mathbf{x}] &= \sum_{k\in\{R,G,B\}}h_{ki}[\mathbf{x}]*f_{kL}[\mathbf{x}], i\in\{R,G,B\} \\ &= \sum_{k\in\{R,G,B\}}\sum_{\mathbf{l}\in\Lambda_L}h_{ki}[\mathbf{l}]f_{kL}[\mathbf{x}-\mathbf{l}], \mathbf{x}\in\Lambda_L. \end{aligned} \tag{3.23}$$

Let

$$\delta_\Lambda[\mathbf{x}]=\begin{cases}1 & \mathbf{x}\in\Lambda \\ 0 & \mathbf{x}\in\Lambda_L\setminus\Lambda\end{cases} \tag{3.24}$$

Then, each color component of the down-sampled image $\mathbf{f}_D[\mathbf{x}]$ is:

$$f_{RD}[\mathbf{x}]=f_{RF}[\mathbf{x}]\delta_\Lambda[\mathbf{x}] \tag{3.25}$$

$$f_{GD}[\mathbf{x}]=f_{GF}[\mathbf{x}]\delta_\Lambda[\mathbf{x}-(\frac{d}{3},0)] \tag{3.26}$$

$$f_{BD}[\mathbf{x}] = f_{BF}[\mathbf{x}]\delta_{\Lambda}[\mathbf{x} - (\frac{2d}{3}, 0)] \tag{3.27}$$

Or we can define the sampling structure for each color component using $\delta_{\Lambda}[\mathbf{x}]$:

$$\delta_R[\mathbf{x}] = \delta_{\Lambda}[\mathbf{x}] \tag{3.28}$$

$$\delta_G[\mathbf{x}] = \delta_{\Lambda}[\mathbf{x} - (\frac{d}{3}, 0)] \tag{3.29}$$

$$\delta_B[\mathbf{x}] = \delta_{\Lambda}[\mathbf{x} - (\frac{2d}{3}, 0)] \tag{3.30}$$

Then, each color component of the down-sampled image $\mathbf{f}_D[\mathbf{x}]$ is:

$$f_{iD}[\mathbf{x}] = \delta_i[\mathbf{x}] \sum_{k \in \{R,G,B\}} \sum_{\mathbf{l} \in \Lambda_L} h_{ki}[\mathbf{l}] f_{kL}[\mathbf{x} - \mathbf{l}], i \in \{R,G,B\}. \tag{3.31}$$

Convert the down-sampled image into the SCIELAB opponent color space:

$$f'_{jD}[\mathbf{x}] = \sum_i a_{ij}\delta_i[\mathbf{x}] \sum_k \sum_{\mathbf{l} \in \Lambda_L} h_{ki}[\mathbf{l}] f_{kL}[\mathbf{x} - \mathbf{l}] \tag{3.32}$$

$$i, k \in \{R,G,B\}, j \in \{O_1, O_2, O_3\}$$

Filter color components of $\mathbf{f}'_D[\mathbf{x}]$ using the three spatial filters in SCIELAB:

$$\begin{aligned} f'_{jDH}[\mathbf{x}] &= f'_{jD}[\mathbf{x}] * h_j[\mathbf{x}] \\ &= \sum_{\mathbf{t} \in \Lambda_L} h_j[\mathbf{t}] f'_{jD}[\mathbf{x} - \mathbf{t}] \\ &= \sum_{\mathbf{t} \in \Lambda_L} h_j[\mathbf{t}] \sum_i a_{ij}\delta_i[\mathbf{x} - \mathbf{t}] \sum_k \sum_{\mathbf{l} \in \Lambda_L} h_{ki}[\mathbf{l}] f_{kL}[\mathbf{x} - \mathbf{t} - \mathbf{l}]. \end{aligned} \tag{3.33}$$

$\mathbf{f}'^{(m)}_{DH}[\mathbf{x}]$ depends linearly on coefficients of $h_{ki}[\mathbf{x}]$. Then, we have the color difference we want to minimize for the RGB stripe display pattern:

$$\begin{aligned}\mathscr{E}(\mathbf{h}) &= \sum_{m=1}^{M}\sum_{\mathbf{x}\in L_m}\sum_{j}(f_{jDH}^{'(m)}[\mathbf{x}]-\frac{1}{3}f_{jLH}^{'(m)}[\mathbf{x}])^2 \\ &= \sum_{m=1}^{M}\sum_{\mathbf{x}\in L_m}\sum_{j}(\sum_{\mathbf{t}\in\Lambda_L}h_j[\mathbf{t}]\sum_{i}a_{ij}\delta_i[\mathbf{x}-\mathbf{t}](\\ &\quad \sum_{k}\sum_{\mathbf{l}\in\Lambda_L}h_{ki}[\mathbf{l}]f_{kL}^{(m)}[\mathbf{x}-\mathbf{t}-\mathbf{l}]-\frac{1}{3}f_{jLH}^{'(m)}[\mathbf{x}]))^2. \end{aligned} \tag{3.34}$$

Because the subpixel structure of the RGB stripe display pattern only enhances the resolution of display in the horizontal direction, the optimization of nine filters for the RGB stripe display pattern can be done in one direction. We can partition the training set into a collection of 1D blocks $\mathbf{f}_{LH}^{'(m)}[\mathbf{x}]$ and partition the output into a collection of 1D blocks $\mathbf{f}_{DH}^{'(m)}[\mathbf{x}]$ in the same way.

Note that the sums over Λ_L are limited to the support of $h_j[\mathbf{x}]$ and $h_{ki}[\mathbf{x}]$ as appropriate. To simplify expressions, let $i,j,k\in\{1,2,3\}$,where $1=R,2=G,$ and $3=B$ for i,k, and $1=O_1,2=O_2,$ and $3=O_3$ for j.

Assume $\Lambda_L=\mathbb{Z}$, where Λ_L is the sampling structure of 1D images, so $d=3$. The size of training image is $h_m=1,...,Q_t$. The support of three SCIELAB filters is $-Q_{Hj},...,Q_{Hj}, j=1,2,3$. The support of nine filters we need to design is $-P_{ki},...,P_{ki}$. So the frequency weighted error can be expressed as

$$\begin{aligned}\mathscr{E}(\mathbf{h}) &= \sum_{m=1}^{M}\sum_{x=1}^{Q_t}\sum_{j=1}^{3}(\sum_{t=-Q_{Hj}}^{Q_{Hj}}h_j[t]\sum_{i=1}^{3}a_{ij}\delta_i[x-t] \\ &\quad \sum_{k=1}^{3}\sum_{l=-P_{ki}}^{P_{ki}}h_{ki}[l]f_{kL}^{(m)}[x-t-l]-\frac{1}{3}f_{jLH}^{'(m)}[x])^2 \end{aligned} \tag{3.35}$$

where unknowns are $h_{ki}[l],i,k\in\{1,2,3\},l\in\{-P_{ki},...,P_{ki}\}$.

The quadratic function has a unique minimum. We can take derivatives to find it:

$$\begin{aligned}\frac{\partial \mathscr{E}(\mathbf{h})}{\partial h_{\xi\eta}[\gamma]} &= 2\sum_{m=1}^{M}\sum_{x=1}^{Q_t}\sum_{j=1}^{3}\Big(\sum_{t=-Q_{Hj}}^{Q_{Hj}} h_j[t]\sum_{i=1}^{3} a_{ij}\delta_i[x-t] \\ &\quad \sum_{k=1}^{3}\sum_{l=-P_{ki}}^{P_{ki}} h_{ki}[l]f_{kL}^{(m)}[x-t-l] - \frac{1}{3}f_{jLH}^{\prime(m)}[x]\Big)\cdot \\ &\quad \sum_{t=-Q_{Hj}}^{Q_{Hj}} h_j[t]a_{\eta j}\delta_\eta[x-t]f_{\xi L}^{(m)}[x-t-\gamma] \qquad (3.36) \\ &= 0 \qquad (3.37)\end{aligned}$$

There are $P_T = \sum_{k=1}^{3}\sum_{i=1}^{3}(2P_{ki}+1)$ equations in the unknowns. The $(\xi,\eta,\gamma)^{th}$ equation is:

$$\begin{aligned}&\sum_{i=1}^{3}\sum_{k=1}^{3}\sum_{l=-P_{ki}}^{P_{ki}}\Big(\sum_{m=1}^{M}\sum_{x=1}^{Q_t}\sum_{j=1}^{3}\Big(\sum_{t=-Q_{Hj}}^{Q_{Hj}} h_j[t]a_{ij}\delta_i[x-t] \\ &f_{kL}^{(m)}[x-t-l]\sum_{s=-Q_{Hj}}^{Q_{Hj}} h_j[s]a_{\eta j}\delta_\eta[x-s]f_{\xi L}^{(m)}[x-s-\gamma]\Big)h_{ki}[l] \\ &= \frac{1}{3}\sum_{m=1}^{M}\sum_{x=1}^{Q_t}\sum_{j=1}^{3} f_{LH}^{\prime(m)}[x]\sum_{s=-Q_{Hj}}^{Q_{Hj}} h_j[s]a_{\eta j}\delta_\eta[x-s]f_{\xi L}^{(m)}[x-s-\gamma] \qquad (3.38)\end{aligned}$$

Thus, by mapping $(i,k,l)\mapsto 1,..,P_T$, we can write the set of linear equations as $\mathbf{Ah}=\mathbf{b}$.

$$\text{eg. } z(i,k,l) = \begin{cases}\sum_{a=1}^{k}(2P_{ki}+1)+(P_{ki}-l) & i=1 \\ \sum_{b=1}^{i-1}\sum_{a=1}^{3}(2P_{ki}+1)+\sum_{a=1}^{k}(2P_{ki}+1)+(P_{ki}-l) & i=2,3\end{cases} \qquad (3.39)$$

So

$$\mathbf{A}_{z(\xi,\eta,\gamma),z(i,k,l)} = \sum_{m=1}^{M}\sum_{x=1}^{Q_t}\sum_{j=1}^{3}\Big(\sum_{t=-Q_{Hj}}^{Q_{Hj}} h_j[t]a_{ij}\delta_i[x-t]f_{kL}^{(m)}[x-t-l] \sum_{s=-Q_{Hj}}^{Q_{Hj}} h_j[s]a_{\eta j}\delta_\eta[x-s]f_{\xi L}^{(m)}[x-s-\gamma]\Big) \tag{3.40}$$

$$\mathbf{b}_{z(\xi,\eta,\gamma)} = \frac{1}{3}\sum_{m=1}^{M}\sum_{x=1}^{Q_t}\sum_{j=1}^{3} f_{jLH}^{\prime(m)}[x] \sum_{s=-Q_{Hj}}^{Q_{Hj}} h_j[s]a_{\eta j}\delta_\eta[x-s]f_{\xi L}^{(m)}[x-s-\gamma] \tag{3.41}$$

3.3.3 RGBG Display Pattern

3.3.3.1 Sampling Structure

The Pentile RGBG display pattern is used as another model to analyze the new approach. Full color images $\mathbf{f}(\mathbf{x})$ are defined on lattice Λ_L of the RGBG display pattern, which is shown in Figure 3.3.

$$\mathbf{f}(\mathbf{x}) = [f_R(\mathbf{x}), f_G(\mathbf{x}), f_B(\mathbf{x})], \mathbf{x} \in \Lambda_L \tag{3.42}$$

$$\Lambda_L = \mathrm{LAT}(\begin{bmatrix} \frac{d}{3} & 0 \\ 0 & \frac{d}{3} \end{bmatrix}) \tag{3.43}$$

The sampling structures for the three color components on the RGBG display pattern are $\Lambda_R, \Lambda_G,$ and Λ_B.

3.3.3.2 Subpixel Image

The subpixel image for the RGBG display pattern shown in Figure 3.6 is

$$\mathbf{f}_S(\mathbf{x}) = [f_{RS}(\mathbf{x}), f_{GS}(\mathbf{x}), f_{BS}(\mathbf{x})], \mathbf{x} \in \Lambda_L \tag{3.44}$$

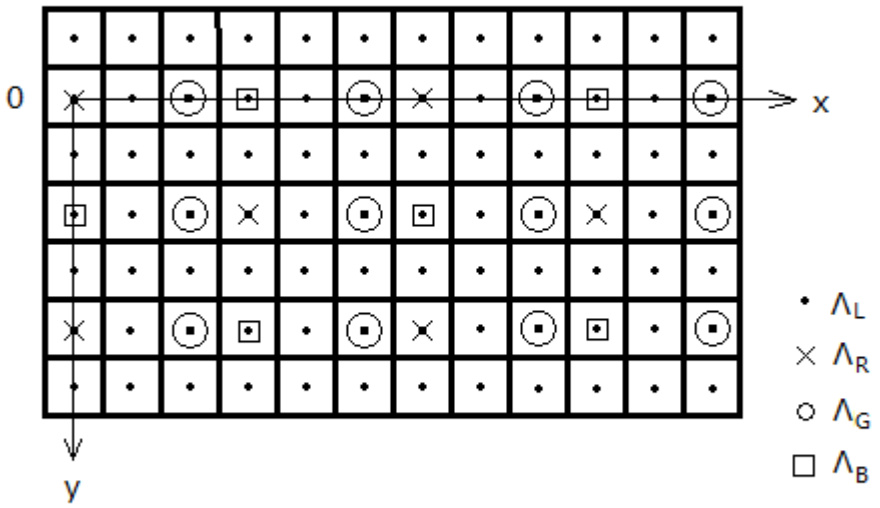


Figure 3.3: The sampling structure of the RGBG display pattern

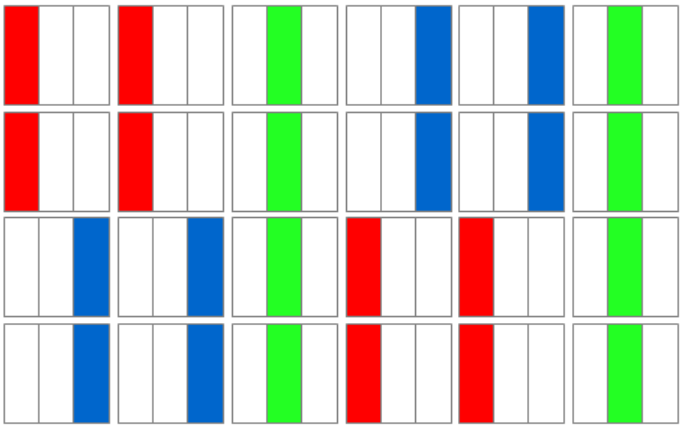


Figure 3.4: The sampling strategy for the RGBG display pattern

where

$$f_{iS}(\mathbf{x}) = \begin{cases} f_i(\mathbf{x}) & \mathbf{x} \in \Lambda_i \\ f_i(\mathbf{x}-\mathbf{b}_{\mathrm{i1}}) & \mathbf{x} \in \Lambda_i + \mathbf{b}_{i1} \\ f_i(\mathbf{x}-\mathbf{b}_{\mathrm{i2}}) & \mathbf{x} \in \Lambda_i + \mathbf{b}_{i2} \quad \text{for } i = R, B \\ f_i(\mathbf{x}-\mathbf{b}_{\mathrm{i3}}) & \mathbf{x} \in \Lambda_i + \mathbf{b}_{i3} \\ 0 & \text{otherwise} \end{cases} \tag{3.45}$$

and

$$f_{GS}(\mathbf{x}) = \begin{cases} f_G(\mathbf{x}) & \mathbf{x} \in \Lambda_G \\ f_G(\mathbf{x}-\mathbf{b}_{\mathrm{G1}}) & \mathbf{x} \in \Lambda_G + \mathbf{b}_{G1} \\ 0 & \text{otherwise} \end{cases} \tag{3.46}$$

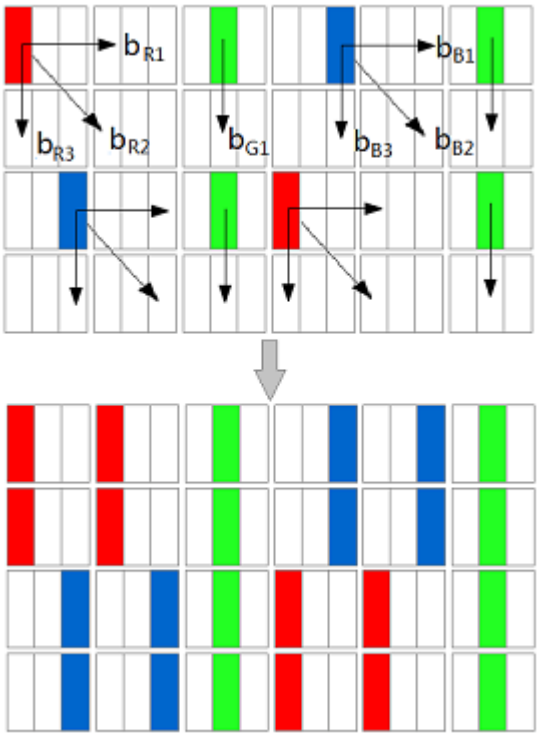


Figure 3.5: Shift of sampled images on the Pentile RGBG display pattern

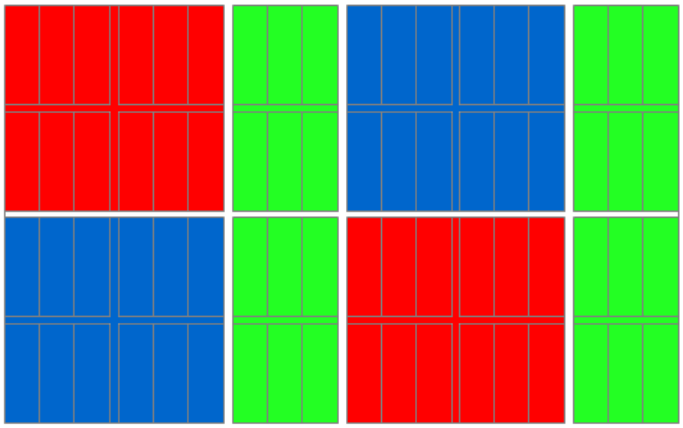


Figure 3.6: Subpixel images of the RGBG display pattern

The shift of sampled image are shown in Figure 3.4 and the vectors $\mathbf{b}_{\mathrm{R1}}, \mathbf{b}_{\mathrm{R2}}, ..., \mathbf{b}_{\mathrm{G1}}$ are shown in Figure 3.5.

3.3.3.3 Least-squares Filter Design

Filter the original full RGB image $\mathbf{f}_L[\mathbf{x}]$ using nine filters we want to design for the Pentile RGBG display pattern:

$$\begin{aligned} f_{iF}[\mathbf{x}] &= \sum_{k\in R,G,B} h_{ki}[\mathbf{x}] * f_{kL}[\mathbf{x}], i \in \{R,G,B\} \\ &= \sum_{k\in R,G,B} \sum_{\mathbf{l}\in\Lambda_L} h_{ki}[\mathbf{l}] f_{kL}[\mathbf{x}-\mathbf{l}], \mathbf{x} \in \Lambda_L. \end{aligned} \tag{3.47}$$

Let

$$\delta_i[\mathbf{x}] = \begin{cases} 1 & \mathbf{x} \in \Lambda_i \\ 0 & \mathbf{x} \in \Lambda_L \setminus \Lambda_i \end{cases} \quad i \in \{R,G,B\} \tag{3.48}$$

where Λ_i is the sampling structure for each color component before the shift, shown in Figure 3.5.

In order to make each subpixel of the low-resolution RGBG display have the same color value in the entire subpixel area, shift the sampled image using vector $\mathbf{b}_{\mathrm{i}}$ for each color component to obtain the displayed subpixel image $\mathbf{f}_D[\mathbf{x}]$:

$$\begin{aligned} f_{iD}[\mathbf{x}] &= \sum_{\mathbf{b}_{\mathrm{i}}} f_{iF}[\mathbf{x}-\mathbf{b}_{\mathrm{i}}] \delta_{\Lambda_i}[\mathbf{x}-\mathbf{b}_{\mathrm{i}}] \\ &= \sum_{\mathbf{b}_{\mathrm{i}}} \delta_i[\mathbf{x}-\mathbf{b}_{\mathrm{i}}] \sum_k \sum_{\mathbf{l}\in\Lambda_L} h_{ki}[\mathbf{l}] f_{kL}[\mathbf{x}-\mathbf{b}_{\mathrm{i}}-\mathbf{l}] \end{aligned} \tag{3.49}$$

where $\mathbf{b}_{\mathrm{i}} \in \{\mathbf{0}, \mathbf{b}_{\mathrm{i}1}, \mathbf{b}_{\mathrm{i}2}, \mathbf{b}_{\mathrm{i}3}\}$ for $i = R, B$, and $\mathbf{b}_{\mathrm{i}} \in \{\mathbf{0}, \mathbf{b}_{\mathrm{i}1}\}$ for $i = G$.

Convert the down-sampled image into the opponent color space in SCIELAB:

$$f'_{jD}[\mathbf{x}] = \sum_i a_{ij} \sum_{\mathbf{b}_{\mathrm{i}}} \delta_i[\mathbf{x}-\mathbf{b}_{\mathrm{i}}] \sum_k \sum_{\mathbf{l}\in\Lambda_L} h_{ki}[\mathbf{l}] f_{kL}[\mathbf{x}-\mathbf{b}_{\mathrm{i}}-\mathbf{l}] \tag{3.50}$$

$$j \in \{O_1, O_2, O_3\}$$

Filter color components of $\mathbf{f}_D^{'}[\mathbf{x}]$ using the three 2D spatial filters in SCIELAB for the RGBG display pattern:

$$\begin{aligned} f_{jDH}^{'}[\mathbf{x}] &= f_{jD}^{'}[\mathbf{x}] * h_j[\mathbf{x}] \\ &= \sum_{\mathbf{t}\in\Lambda_L} h_j[\mathbf{t}] f_{jD}^{'}[\mathbf{x}-\mathbf{t}] \\ &= \sum_{\mathbf{t}\in\Lambda_L} h_j[\mathbf{t}] \sum_{i\in\{R,G,B\}} a_{ij} \sum_{\mathbf{b_i}} \delta_i[\mathbf{x}-\mathbf{t}-\mathbf{b_i}] \\ &\quad \sum_{k\in\{R,G,B\}} \sum_{\mathbf{l}\in\Lambda_L} h_{ki}[\mathbf{l}] f_{kL}[\mathbf{x}-\mathbf{t}-\mathbf{b_i}-\mathbf{l}] \end{aligned} \tag{3.51}$$

The image $\mathbf{f}_{DH}^{'(m)}[\mathbf{x}]$ depends linearly on the coefficients of $h_{ki}[\mathbf{x}]$. Then, we have the color difference we want to minimize for the RGBG display pattern:

$$\begin{aligned} \mathscr{E}(\mathbf{h}) &= \sum_{m=1}^{M} \sum_{\mathbf{x}\in L_m} \sum_{j\in\{O_1,O_2,O_3\}} (f_{jDH}^{'(m)}[\mathbf{x}] - \frac{1}{3} f_{jLH}^{'(m)}[\mathbf{x}])^2 \\ &= \sum_{m=1}^{M} \sum_{\mathbf{x}\in L_m} \sum_{j\in\{O_1,O_2,O_3\}} (\sum_{\mathbf{t}\in\Lambda_L} h_j[\mathbf{t}] \sum_{i\in\{R,G,B\}} a_{ij} \sum_{\mathbf{b_i}} \delta_i[\mathbf{x}-\mathbf{t}-\mathbf{b_i}] \\ &\quad \sum_{k\in\{R,G,B\}} \sum_{\mathbf{l}\in\Lambda_L} h_{ki}[\mathbf{l}] f_{kL}^{(m)}[\mathbf{x}-\mathbf{t}-\mathbf{b_i}-\mathbf{l}] - \frac{1}{3} f_{jLH}^{'(m)}[\mathbf{x}])^2 \end{aligned} \tag{3.52}$$

Also, to simplify expressions, let $i,j,k \in \{1,2,3\}$, where $1=R, 2=G, 3=B$ for i,k and $1=O_1, 2=O_2, 3=O_3$ for j. Unknowns are $h_{ki}[\mathbf{l}]$, $k,i \in \{1,2,3\}, l \in \Lambda_L$. The quadratic function has a unique minimum. Take derivatives:

$$\begin{aligned} \frac{\partial \mathscr{E}(\mathbf{h})}{\partial h_{\xi\eta}[\gamma]} &= 2 \sum_{m=1}^{M} \sum_{\mathbf{x}\in L_m} \sum_{j} (\sum_{\mathbf{t}\in\Lambda_L} h_j[\mathbf{t}] \sum_{i} a_{ij} \sum_{\mathbf{b_i}} \delta_i[\mathbf{x}-\mathbf{t}-\mathbf{b_i}] \\ &\quad \sum_{k} \sum_{\mathbf{l}\in\Lambda_L} h_{ki}[\mathbf{l}] f_{kL}^{(m)}[\mathbf{x}-\mathbf{b_i}-\mathbf{t}-\mathbf{l}] - \frac{1}{3} f_{jLH}^{'(m)}[\mathbf{x}]) \cdot \\ &\quad \sum_{\mathbf{t}\in\Lambda_L} h_j[\mathbf{t}] a_{\eta j} \sum_{\mathbf{b}_\eta} \delta_\eta[\mathbf{x}-\mathbf{t}-\mathbf{b}_\eta] f_{\xi L}^{(m)}[\mathbf{x}-\mathbf{b}_\eta-\mathbf{t}-\gamma] \end{aligned} \tag{3.53}$$

$$= 0 \tag{3.54}$$

There are P_T equations in the unknowns. By mapping $(i,k,l) \mapsto 1,..,P_T$, we can write the set of linear equations as $\mathbf{Ah} = \mathbf{b}$:

$$\begin{aligned}\mathbf{A}_{z(\xi,\eta,\gamma),z(i,k,\mathbf{l})} &= \sum_{m=1}^{M} \sum_{\mathbf{x}\in L_m} \sum_{j\in\{O_1,O_2,O_3\}} \\ &\quad \sum_{\mathbf{t}\in\Lambda_L} h_j[\mathbf{t}] a_{ij} \sum_{\mathbf{b}_\mathrm{i}} \delta_i[\mathbf{x}-\mathbf{t}-\mathbf{b}_\mathrm{i}] f_{kL}^{(m)}[\mathbf{x}-\mathbf{b}_\mathrm{i}-\mathbf{t}-\mathbf{l}] \\ &\quad \sum_{\mathbf{s}\in\Lambda_L} h_j[\mathbf{s}] a_{\eta j} \sum_{\mathbf{b}_\eta} \delta_\eta[\mathbf{x}-\mathbf{s}-\mathbf{b}_\eta] f_{\xi L}^{(m)}[\mathbf{x}-\mathbf{b}_\eta-\mathbf{s}-\gamma], \end{aligned} \tag{3.55}$$

$$\begin{aligned}\mathbf{b}_{z(\xi,\eta,\gamma)} &= \frac{1}{3} \sum_{m=1}^{M} \sum_{\mathbf{x}\in L_m} \sum_{j=1}^{3} f_{jLH}^{\prime(m)}[\mathbf{x}] \\ &\quad \sum_{\mathbf{s}\in\Lambda_L} h_j[\mathbf{s}] a_{ij} \sum_{\mathbf{b}_\eta} \delta_\eta[\mathbf{x}-\mathbf{s}-\mathbf{b}_\eta] f_{\xi L}^{(m)}[\mathbf{x}-\mathbf{b}_\eta-\mathbf{s}-\gamma] \end{aligned}$$

3.4 Solution for Implementation

3.4.1 RGB Stripe Display Pattern

We can simplify above results by using convolutions. Let

$$p_{\eta\xi\gamma}^{(m)}[x-s] = \delta_\eta[x-s] f_{\xi L}^{(m)}[x-s-\gamma] \tag{3.56}$$

Then, the following equation can be expressed as a convolution:

$$\begin{aligned}&\sum_{s=-Q_{Hj}}^{Q_{Hj}} h_j[s] \delta_\eta[x-s] f_{\xi L}^{(m)}[x-s-\gamma]) \\ &\qquad = \sum_{s=-Q_{Hj}}^{Q_{Hj}} h_j[s] p_{\eta\xi\gamma}^{(m)}[x-s] \\ &\qquad = h_j[x] * p_{\eta\xi\gamma}^{(m)}[x] \end{aligned} \tag{3.57}$$

This is used in the expression of both $\mathbf{A}_{z(\eta,\xi,\gamma),z(i,k,l)}$ and $\mathbf{b}_{z(\xi,\eta,\gamma)}$. So, we can express the solution as:

$$\mathbf{A}_{z(\eta,\xi,\gamma),z(i,k,l)} = \sum_{m=1}^{M}\sum_{j=1}^{3} a_{ij}a_{\eta j}\sum_{x=1}^{Q_t}(h_j * p^{(m)}_{\eta\xi\gamma})[x]\cdot(h_j * p^{(m)}_{ikl})[x]$$

$$\mathbf{b}_{z(\xi,\eta,\gamma)} = \frac{1}{3}\sum_{m=1}^{M}\sum_{j=1}^{3} a_{\eta j}\sum_{x=1}^{Q_t} f'^{(m)}_{jLH}\cdot[x](h_j * p^{(m)}_{\eta\xi\gamma})[x]$$

The optimal filters for the RGB stripe display pattern are:

$$\mathbf{h} = \mathbf{A}^{-1}\mathbf{b} = \begin{bmatrix} \mathbf{h}_{11} \\ \mathbf{h}_{21} \\ \mathbf{h}_{31} \\ \vdots \\ \mathbf{h}_{33} \end{bmatrix} \tag{3.58}$$

3.4.2 RGBG Display Pattern

Similar with the solution of the RGB stripe display pattern, we have

$$\begin{aligned}\mathbf{A}_{z(\xi,\eta,\gamma),z(i,k,\mathbf{l})} = &\sum_{m=1}^{M}\sum_{j=1}^{3} a_{ij}a_{\eta j}\sum_{\mathbf{x}\in L_m} \\ &h_j[\mathbf{x}] * (\sum_{\mathbf{b_i}}\delta_i[\mathbf{x}-\mathbf{b}_\mathbf{i}]f^{(m)}_{kL}[\mathbf{x}-\mathbf{b}_\mathbf{i}-\mathbf{l}]) \\ &h_j[\mathbf{x}] * (\sum_{\mathbf{b}_\eta}\delta_\eta[\mathbf{x}-\mathbf{b}_\eta]f^{(m)}_{\xi L}[\mathbf{x}-\mathbf{b}_\eta-\gamma])\end{aligned} \tag{3.59}$$

$$\begin{aligned}\mathbf{b}_{z(\xi,\eta,\gamma)} = &\frac{1}{3}\sum_{m=1}^{M}\sum_{j=1}^{3} a_{\eta j}\sum_{\mathbf{x}\in L_m} f'^{(m)}_{jLH}[\mathbf{x}]\cdot \\ &h_j[\mathbf{x}] * (\sum_{\mathbf{b}_\eta}\delta_\eta[\mathbf{x}-\mathbf{b}_\eta]f^{(m)}_{\xi L}[\mathbf{x}-\mathbf{b}_\eta-\gamma])\end{aligned} \tag{3.60}$$

Optimal filters for the RGBG display pattern are

$$\mathbf{h} = \mathbf{A}^{-1}\mathbf{b} = \begin{bmatrix} \mathbf{h}_{11} \\ \mathbf{h}_{21} \\ \mathbf{h}_{31} \\ \vdots \\ \mathbf{h}_{33} \end{bmatrix} \tag{3.61}$$

3.5 Summary

In this chapter, the system structure of the new approach are described. A mathematical solution for a general 2D display pattern is introduced. Solutions for implementing the RGB stripe display pattern and the Pentile RGBG display pattern are given for testing the two display patterns in the following chapters.

Chapter 4

RGB Stripe Display Pattern

In this chapter, the subpixel-based down-sampling on the RGB stripe display pattern is implemented using existing methods and the new approach. Frequency weighted errors and SCIELAB errors are attempted to optimize. Down-sampled images of existing methods and the new approach are compared by the two errors and visual inspection.

4.1 Optimization of Frequency Weighted Errors

According to the down-sampling system design, optimization is performed in the opponent color space, so errors between original images and reproduced images using different kinds of structure conversion methods are calculated to verify that the optimization works correctly.

For existing methods described in Chapter 2 and the new approach, the frequency weighted errors for the training image (TI) and ten testing images calculated using Formula 3.35 are shown in Table 4.1. Nine filters trained by one training image, Figure 4.1, are shown in Figure 4.2. We choose filter orders of the nine filters as nine, which makes the training time in experiments reasonable and gives good quality of reproduced images. Testing Images are shown in Figure 4.3. The coefficients of the nine filters are shown in Table 4.2. The designed filters are not linear-phase, because they must include a shift component.

The training image and testing images are from the Kodak Photo CD collection, which is often used for image testing. The size of the Kodak images is 3072×2048 pixels.

From the implementation results, it can be verified that the optimization of frequency weighted errors for RGB stripe display patterns works correctly. The frequency weighted

	DPD	DSD	DDSD	DDSD-FA [16]	DSD-MMSE [5]	DDSD-MMSE [5]	The new approach
1	0.9306	0.8892	1.0397	1.0326	0.7570	0.9241	0.5687
2	0.6336	0.6132	0.6702	0.6530	0.5243	0.5996	0.4131
3	0.1413	0.1286	0.1672	0.2391	0.1818	0.2311	0.1069
4	0.1283	0.1167	0.1333	0.1493	0.1218	0.1430	0.0794
5	0.3100	0.2848	0.3199	0.3260	0.2554	0.3013	0.1877
6	0.9379	0.8694	0.9405	0.8812	0.7301	0.8348	0.5428
7	0.1241	0.1048	0.1198	0.1184	0.1000	0.1170	0.0631
8	0.5008	0.4775	0.5506	0.5294	0.4200	0.4941	0.3187
9	0.0855	0.0735	0.0917	0.1452	0.1069	0.1287	0.0599
10	0.4535	0.4160	0.4566	0.4242	0.3594	0.4077	0.2550
TI	1.1654	1.0460	1.1683	1.0355	0.7732	0.9422	0.5491

Table 4.1: Frequency weighted errors of ten testing images and the training image using existing methods and the new approach

Figure 4.1: The training image for the RGB stripe display pattern

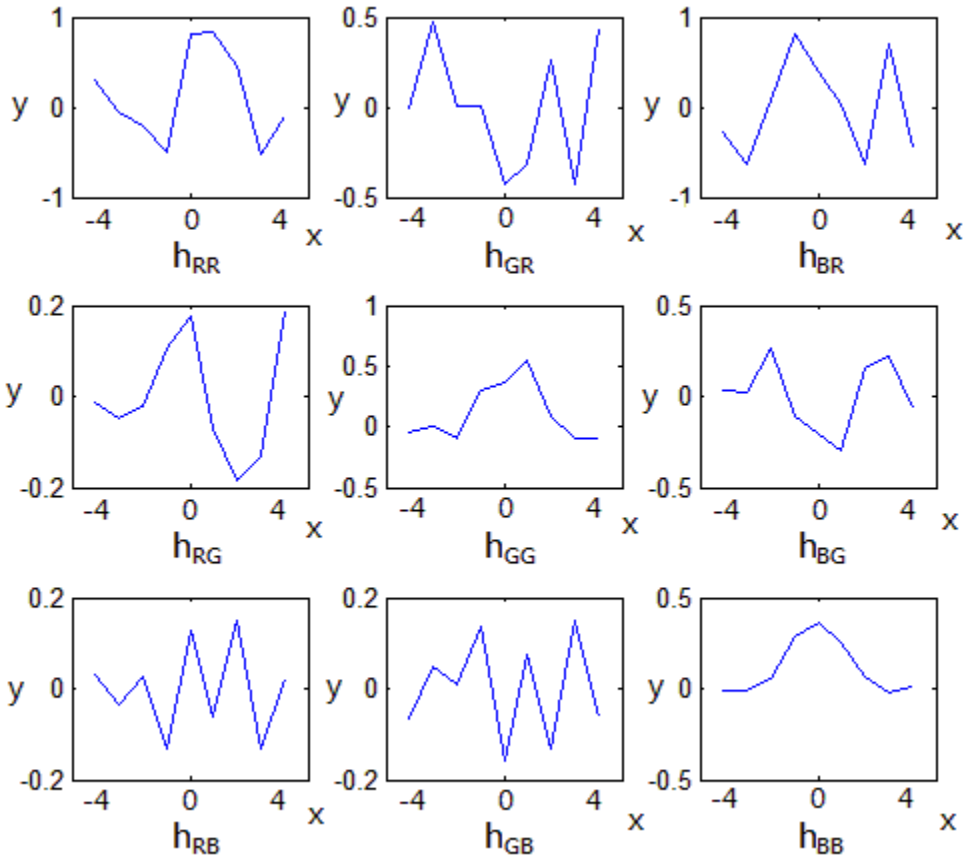
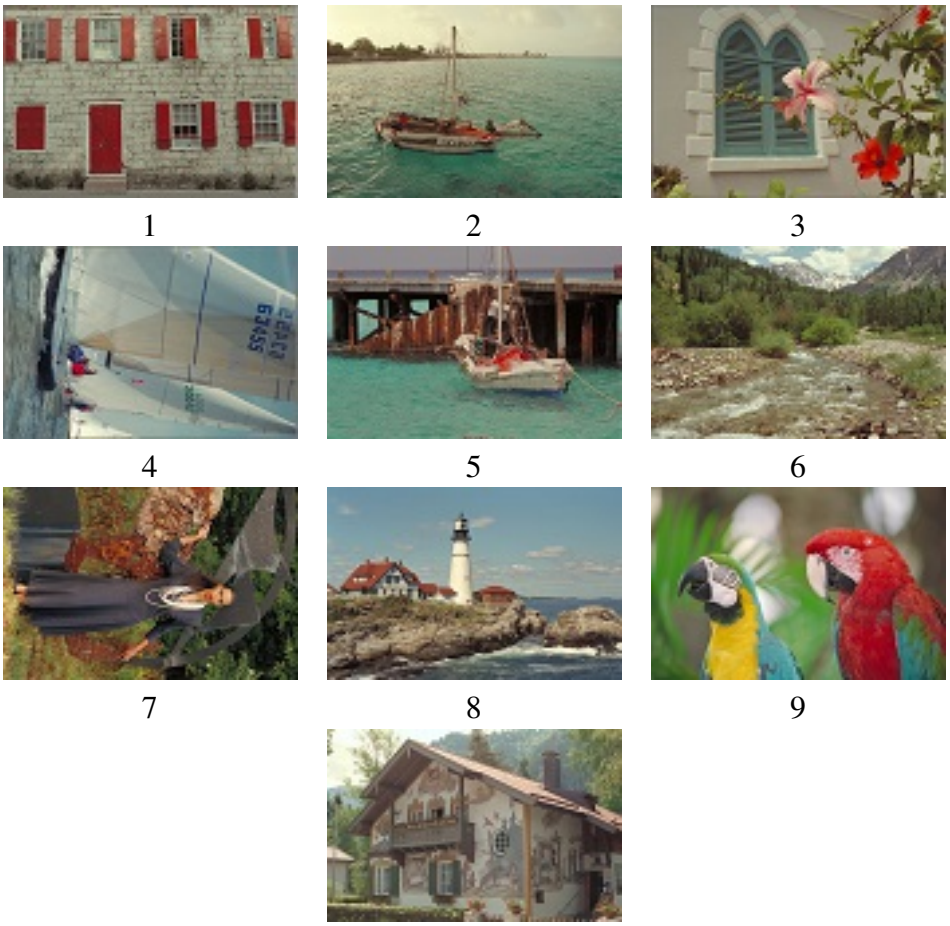


Figure 4.2: Nine filters of the new approach for the RGB stripe display pattern

	-4	-3	-2	-1	0	1	2	3	4
h_{RR}	0.2893	-0.0652	-0.2052	-0.4969	0.8010	0.8230	0.4604	-0.5208	-0.0982
h_{GR}	-0.0047	0.4691	0.0060	0.0025	-0.4234	-0.3197	0.2663	-0.4245	0.4434
h_{BR}	-0.2676	-0.6375	0.0718	0.8090	0.3804	0.0293	-0.6305	0.7075	-0.4562
h_{RG}	-0.0127	-0.0489	-0.0239	0.1074	0.1770	-0.0722	-0.1861	-0.1316	0.1910
h_{GG}	-0.0403	0.0079	-0.1004	0.2988	0.3724	0.5488	0.0918	-0.0888	-0.0924
h_{BG}	0.0340	0.0176	0.2612	-0.1071	-0.2106	-0.2957	0.1560	0.2162	-0.0695
h_{RB}	0.0312	-0.0331	0.0278	-0.1314	0.1268	-0.0612	0.1476	-0.1322	0.0205
h_{GB}	-0.0673	0.0466	0.0066	0.1371	-0.1574	0.0728	-0.1298	0.1481	-0.0603
h_{BB}	-0.0072	-0.0122	0.0580	0.2898	0.3640	0.2575	0.0650	-0.0171	0.0127

Table 4.2: Coefficients of the nine filters for the RGB stripe display pattern



1 2 3

4 5 6

7 8 9

10

Figure 4.3: Testing images 1-10

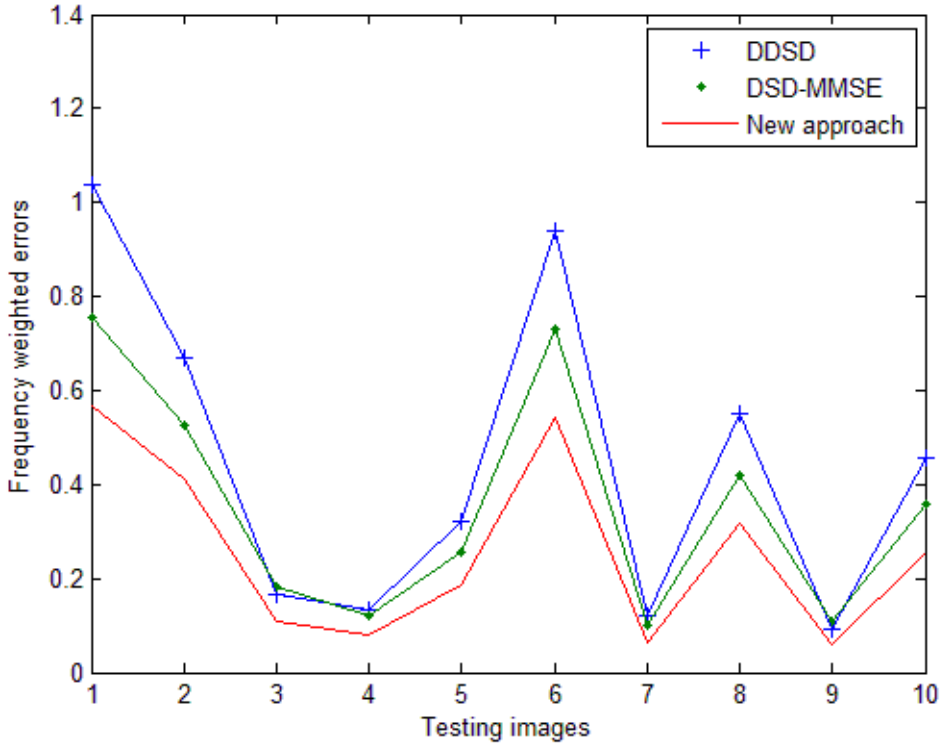


Figure 4.4: Plots of frequency weighted errors of the new approach and two existing methods for the ten testing images

errors of the new approach are smaller than those of all the tested existing methods for the training image and all the testing images, which is shown as plots in Figure 4.4.

4.2 SCIELAB Error Calculation

The SCIELAB errors between the original images and reproduced images are calculated. The viewing distance used in the SCIELAB is twice the display height.

For the existing methods and the new approach, the SCIELAB errors for the training image and the testing images are shown in Table 4.3.

From the implementation results, it can be verified that the optimization of the frequency weighted errors for the RGB stripe display pattern also works correctly when we use the SCIELAB color metric measuring the color difference. The SCIELAB errors of

	DPD	DSD	DDSD	DDSD-FA	DSD-MMSE	DDSD-MMSE	The new approach
1	13.6053	16.9674	48.9036	35.6297	9.9644	28.8660	8.5120
2	6.2826	7.5336	19.2786	16.0009	5.1364	13.3556	4.1527
3	3.7605	3.7027	9.9742	13.3802	4.0251	12.6935	2.8112
4	1.7003	2.1548	3.6602	3.3170	1.4063	3.0131	1.0292
5	5.4200	8.0078	15.6935	12.1613	3.9467	10.4249	3.1797
6	10.5944	17.0744	29.7027	20.5376	7.6472	18.0167	5.8423
7	2.5070	3.8684	6.2099	4.6295	1.7634	4.3486	1.2579
8	6.2070	8.2318	19.7035	15.3364	5.0096	13.2704	3.9312
9	1.6797	1.4308	2.7658	3.4588	1.3769	3.0464	1.1534
10	4.1620	5.8473	10.6833	8.4525	3.3326	7.6193	2.3497
TI	11.7075	20.7638	34.7581	22.4954	7.5061	18.6549	5.7748

Table 4.3: SCIELAB errors of ten testing images and the training image using existing methods and the new approach

new approach are smaller than those of all the tested existing methods for the training image and all the testing images, which is shown as plots in Figure 4.5, even though it is frequency weighted mean square error that has been minimized.

4.2.1 Relation Between Two Errors

Using the frequency weighted errors and SCIELAB errors we obtained for the ten testing images, the relation between two kinds of errors can be shown by plotting them in one figure. In Figure 4.6, the normalized frequency weighted errors and SCIELAB errors for the testing images are shown. The relation between the two kinds of errors is approximately linear.

The ranking of errors is also plotted in Figure 4.7 to verify the approximate linear relation between the frequency weighted errors and SCIELAB errors.

For the purpose of comparison, PSNR of the RGB values between the original images and reproduced testing images are calculated to plot in one figure with the SCIELAB errors in dB. From Figure 4.8, the nonlinear relation between the PSNR and the SCIELAB errors

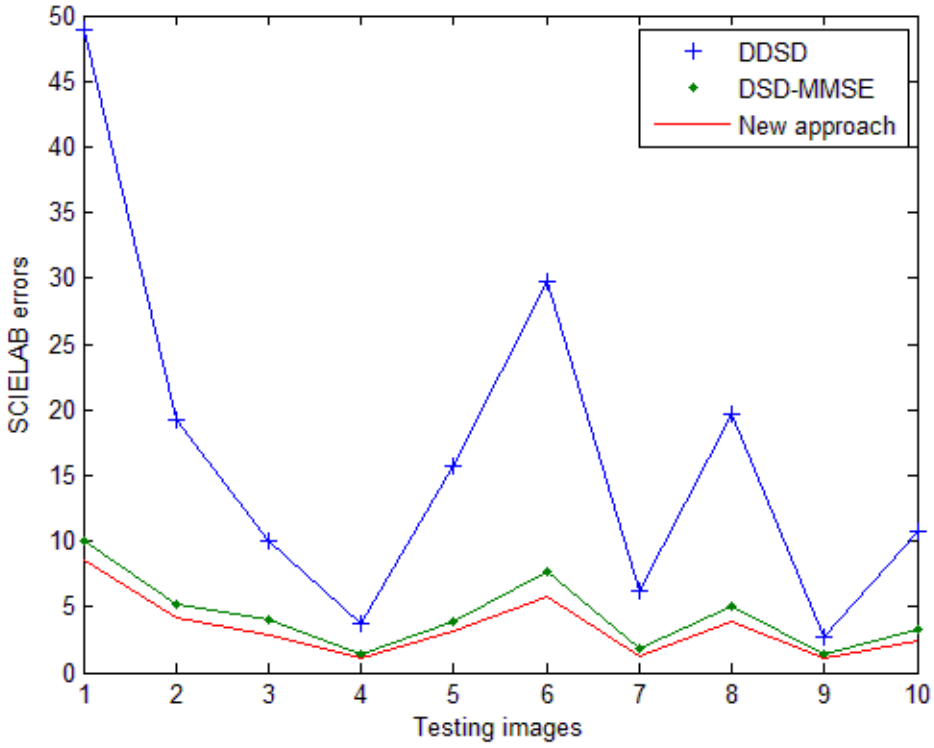


Figure 4.5: Plots of SCIELAB errors of the new approach and two existing methods for the ten testing images

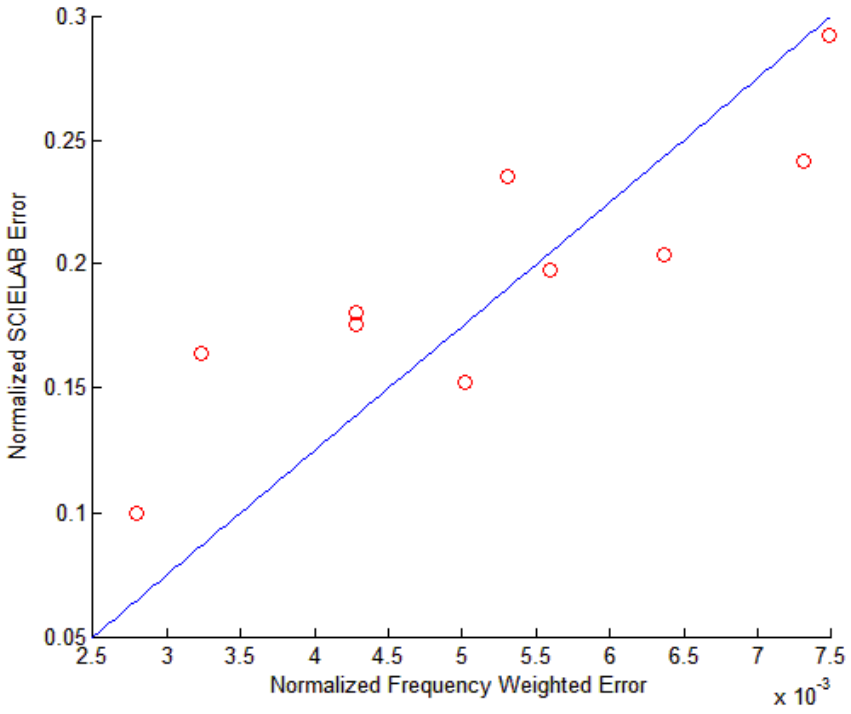


Figure 4.6: The plot of normalized frequency weighted and SCIELAB errors for the ten testing images using the new approach

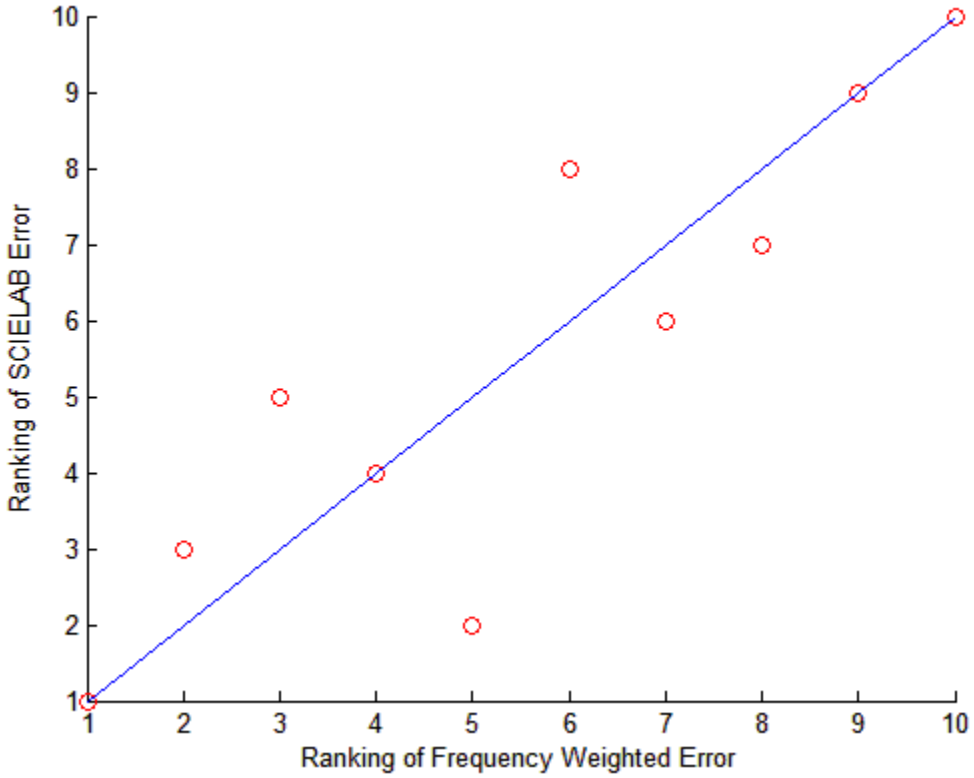


Figure 4.7: Rankings of frequency weighted and SCIELAB errors for the ten testing images

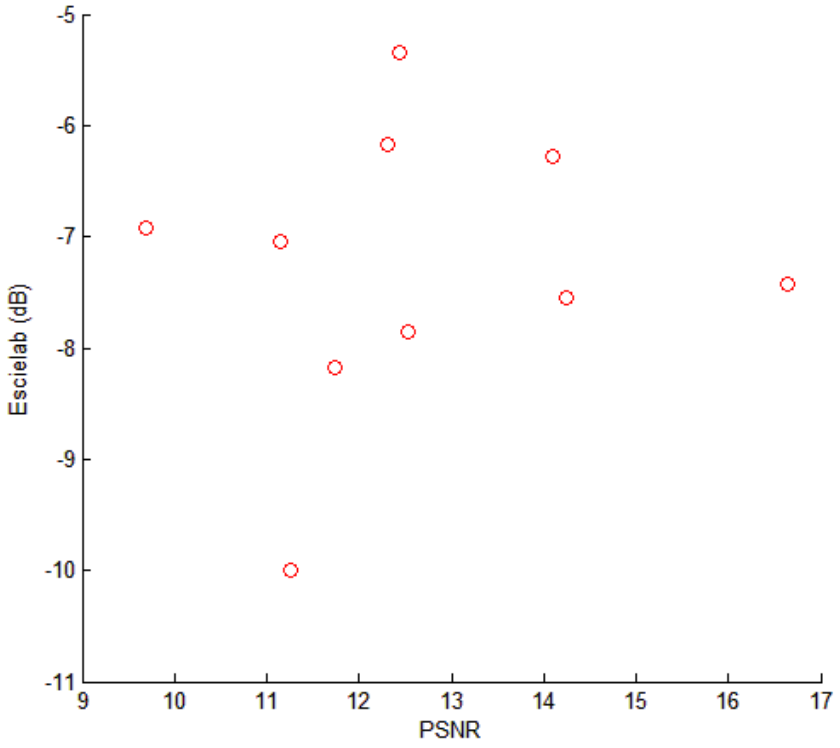


Figure 4.8: The plot of PSNR and SCIELAB errors for the ten testing images

can be seen. It shows that the optimization of the PSNR cannot minimize the perceived color difference measured by the SCIELAB color metric.

4.3 Visual Inspection

4.3.1 Photographic Images

Subpixel images of reproduced images and original images of the training image and some testing images are shown in Figure 4.9, 4.10, 4.11 and 4.12. Subpixel images are a zoomed version of down-sampled images which would appear on the subpixel display. The subpixel structure can be seen on subpixel images at a normal viewing distance without using a magnifying glass. The subpixel images should be viewed on a computer screen from an appropriate distance.

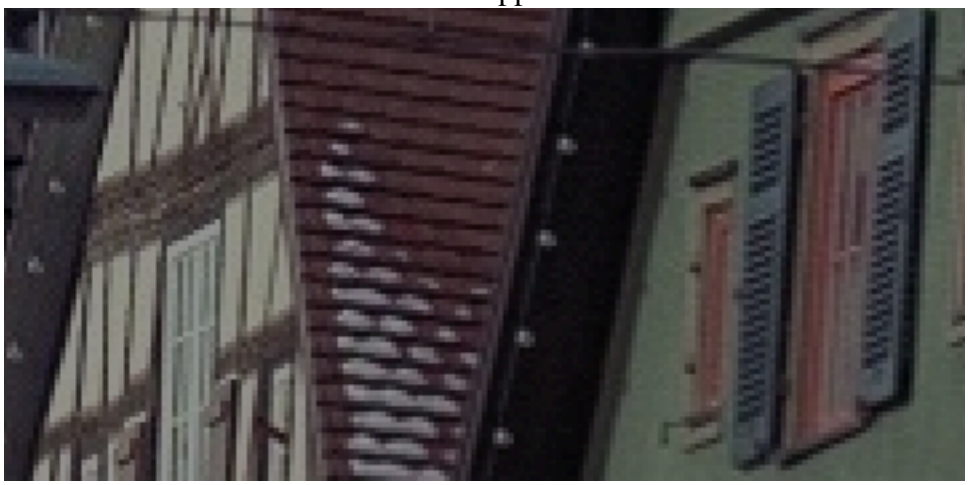
In Figure 4.9, subpixel images of the training image are displayed at 200%, so the images are normally viewed at a distance of four times the display height. At this viewing distance, we can see the color distortion of the subpixel image is suppressed more using the new approach, compared with an existing subpixel-based down-sampling method. The color fringing on the window and roof disappears on the subpixel image obtained from the new approach, when observers move back to view it at a distance of four times the display height. The color fringing is still visible on the subpixel image of the existing method, DDSD-MMSE, at the same viewing distance. If observers view the subpixel images in Figure 4.9 in a printed thesis, the viewing distance would be about 40 inches for seeing this effect. If observers view the subpixel images further away, the color differences between subpixel images become smaller.

In Figure 4.10 and 4.11 (b), subpixel images are displayed at 100%, so the normal viewing distance is twice the display height. Subpixel images in Figure 4.11 (a) are displayed at 200%. The normal viewing distance is four times the display height. We can see that color distortion on the fence and window is less using the new approach, compared with existing methods.

In Figure 4.12, subpixel images are displayed at 100%. The color of flower edges is enhanced using the new approach. Details of branches and leaves are well preserved. In the

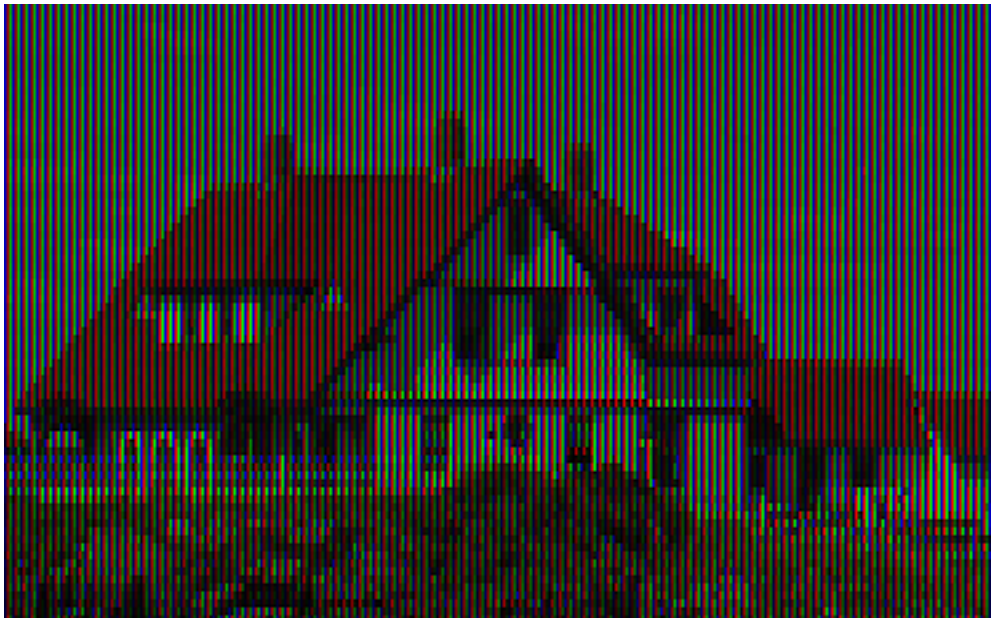
DDSD-MMSE

The new approach

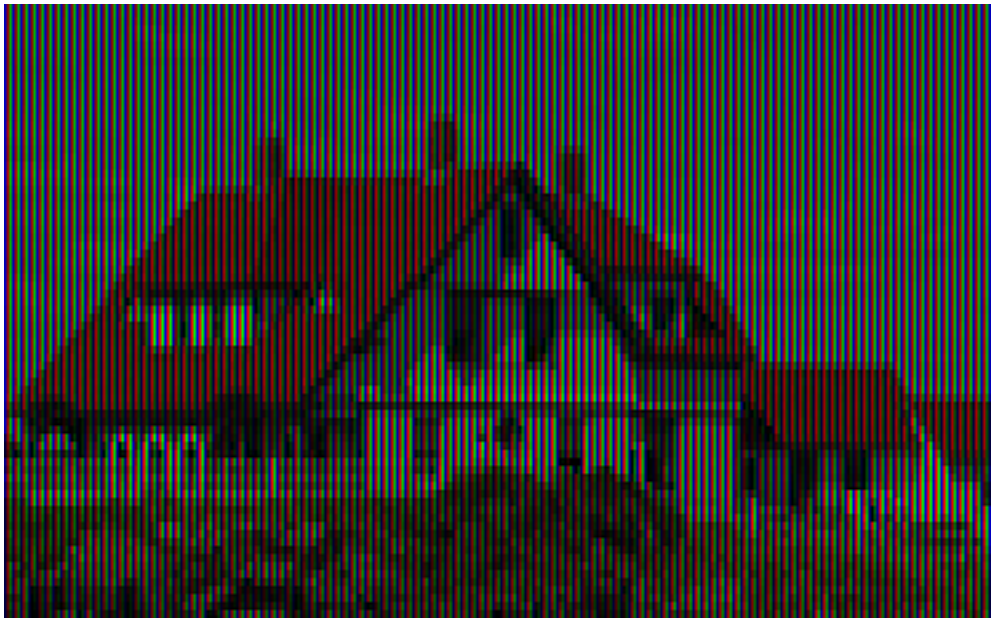


Original

Figure 4.9: Displayed subpixel images of the training image



DDSD



The new approach

Figure 4.10: Displayed subpixel images of Image 8

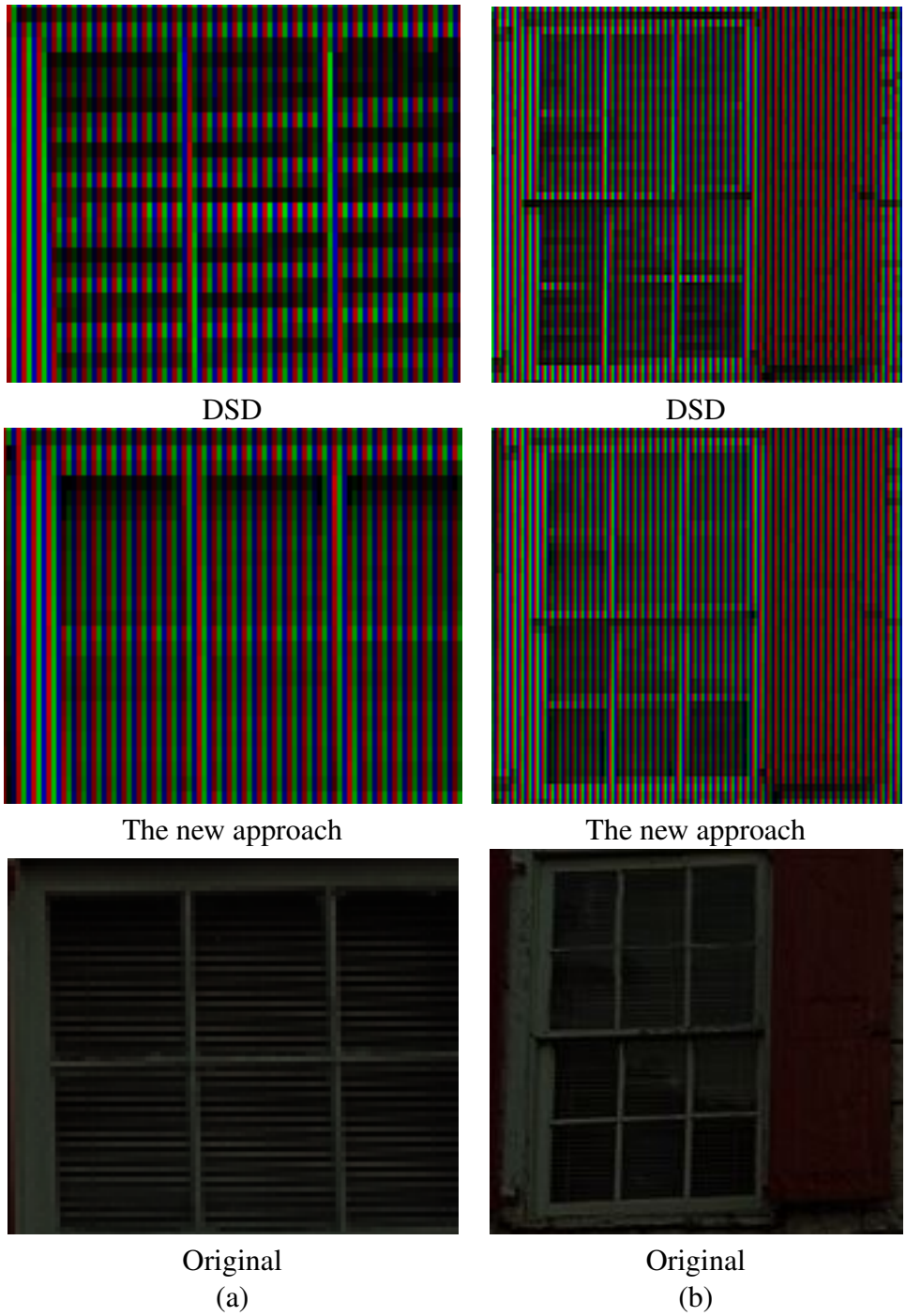


Figure 4.11: Displayed subpixel images of Image 1

DDSD-MMSE

The new approach

Figure 4.12: Displayed subpixel images of Image 7

subpixel image of an existing method, DDSD-MMSE, there are more bright and colored edges around the pedals, leaves and branches.

4.3.2 Text

Original images and subpixel images of a text image are shown in Figure 4.13. Subpixel images are displayed at 200% in Figure 4.13 (a) and at 100% in Figure 4.13 (b).

More color fringing effect is suppressed in the images using the new approach, in comparison of an existing subpixel rendering method, DDSD-FA. Besides, the edges of letters in the text are still maintained as well as those obtained by using the existing method.

4.3.3 Zone Plate

Subpixel images of a reproduced zone plate by the new approach and an existing method, DDSD-MMSE, are displayed 200% in Figure 4.14. Compared with the existing method, the new approach keeps more details of high frequency and shows less color distortion on the edges of the black and white stripes.

4.4 Summary

In this chapter, we used the RGB stripe display pattern to test the the new approach. The frequency weighted errors and SCIELAB errors are minimized using the new approach, in comparison of existing subpixel rendering methods. More sloping edge effect and color fringing effect are suppressed and details of testing images are well preserved.

DDSD-FA DDSD-FA

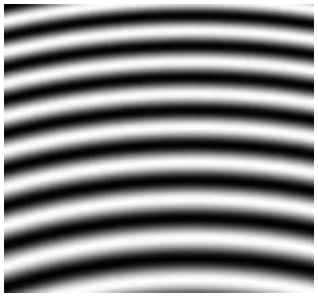
The new approach The new approach

Original Original

(a) (b)

Figure 4.13: Subpixel images of text images

DDSD-MMSE New approach



Original

Figure 4.14: Original and subpixel images of a zone plate

Chapter 5

PenTile RGBG Display Pattern

In this chapter, the 2D display pattern, Pentile RGBG, is tested to compare the new approach with existing subpixel rendering methods. Because there is a shift in the vertical direction, images for the Pentile RGBG display pattern cannot be prefiltered only in one direction as those for the RGB stripe display pattern. Nine 2D filters are required to optimize for the 2D display pattern, in order to prefilter color components in both dimensions. The frequency weighted errors and SCIELAB errors are attempted to be minimized using nine 2D filters for the Pentile RGBG display pattern.

5.1 Optimization of Frequency Weighted Errors

According to the down-sampling system design, optimization is performed in the opponent color space, so frequency weighted errors between original images and reproduced images using existing subpixel rendering methods and the new approach are calculated, to verify that the optimization works correctly. DSD and a subpixel rendering scheme for the Pentile RGBG display pattern using tent filters are used to compare with the new approach. The latter existing subpixel rendering scheme uses the tent filter:

$$h_t = \begin{bmatrix} 0 & \frac{1}{8} & 0 \\ \frac{1}{8} & \frac{1}{2} & \frac{1}{8} \\ 0 & \frac{1}{8} & 0 \end{bmatrix}$$

to do subpixel rendering, which uses values of a central and four neighboring subpixels to generate a subpixel in down-sampled images [23].

Figure 5.1: The training image for the Pentile RGBG display pattern

Row	1	1	1	2	2	2	3	3	3
Col.	1	2	3	1	2	3	1	2	3
h_{RR}	0.0611	-0.8174	0.5856	-0.0475	-0.1406	0.1158	0.0499	-0.2101	0.1024
h_{GR}	0.0762	0.6633	-0.4042	-0.0606	0.1769	-0.0799	-0.0271	0.0811	0.0424
h_{BR}	0.1927	-0.2350	0.2384	0.0116	-0.1909	0.0946	0.0172	-0.1144	0.1629
h_{RG}	-0.0384	-0.0345	0.1050	0.0374	0.2712	-0.0882	-0.0335	0.0663	0.0741
h_{GG}	0.1310	0.3445	-0.4185	0.0495	0.2972	-0.1413	0.0410	0.0289	0.0077
h_{BG}	-0.0420	-0.0892	0.1834	0.0202	0.2698	-0.0741	-0.0720	0.0844	0.1713
h_{RB}	0.1605	-0.6056	0.3324	-0.0407	0.1319	0.0676	0.0177	-0.0478	0.1659
h_{GB}	0.0869	0.8831	-0.5847	0.0598	0.2978	-0.1383	-0.0012	-0.0424	0.1640
h_{BB}	0.3733	-0.0956	-0.0448	-0.0298	-0.1079	0.2401	0.0082	0.1579	0.1074

Table 5.1: Coefficients of the nine filters for the Pentile RGBG display pattern

Nine filters for the Pentile RGBG display pattern are trained by the same training image as the one used for RGB stripe display pattern, Figure 5.1. The coefficients of the nine filters are shown in Table 5.1. Ten testing images used for the Pentile RGBG display

	1	2	3	4	5	6
DSD	0.0700	0.0353	0.0149	0.0120	0.0257	0.0649
Tent	0.0300	0.0183	0.0124	0.0075	0.0148	0.0324
New	0.0055	0.0031	0.0013	0.0017	0.0021	0.0065
	7	8	9	10	the training image	
DSD	0.0227	0.0431	0.0088	0.0326	0.1805	
Tent	0.0140	0.0236	0.0067	0.0183	0.0865	
New	0.0021	0.0041	0.0016	0.0028	0.0080	

Table 5.2: Frequency weighted errors of ten testing images and the training image using existing methods and the new approach

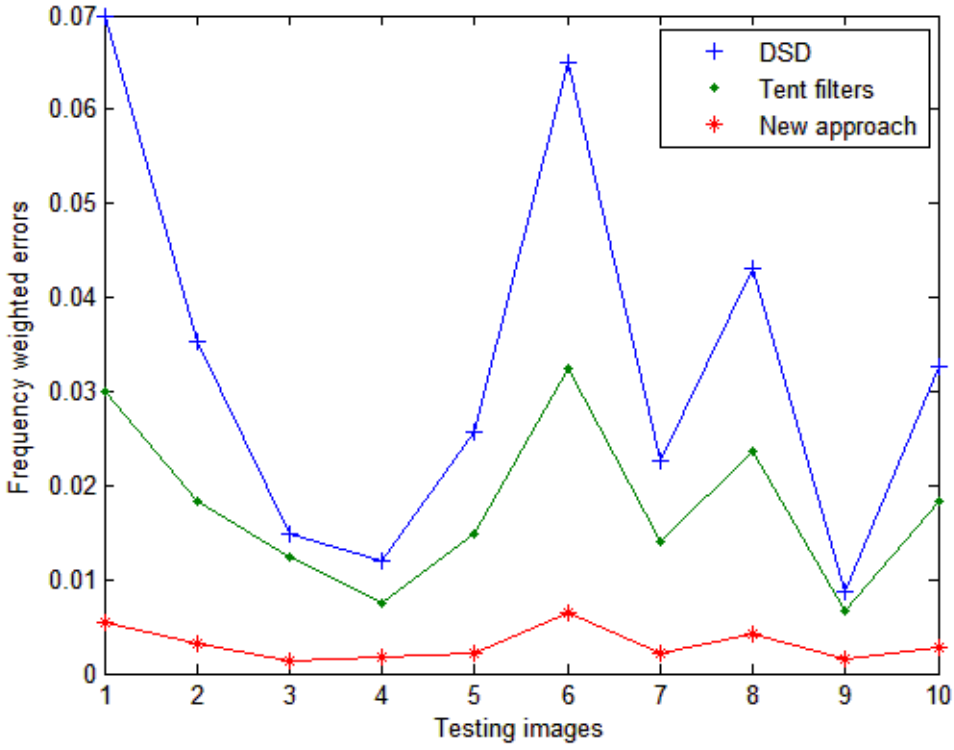


Figure 5.2: Plots of frequency weighted errors for the ten testing images

	1	2	3	4	5	6
DSD	0.7846	0.2510	0.1538	0.1216	0.4493	0.7069
Tent	0.3283	0.1180	0.1139	0.0667	0.2199	0.2866
New	0.1805	0.0682	0.0469	0.0252	0.0990	0.1939

	7	8	9	10	the training image
DSD	0.3551	0.3773	0.0896	0.2930	1.6744
Tent	0.1653	0.1756	0.0641	0.1449	0.7223
New	0.0851	0.0866	0.0478	0.0635	0.2559

Table 5.3: SCIELAB errors of ten testing images and the training image using existing methods and the new approach

pattern are the same as the ten testing images for the RGB stripe display pattern, shown in Figure 4.3.

The frequency weighted errors of existing subpixel rendering schemes and the new approach for the training and testing images are shown in Table 5.2 and Figure 5.2. The frequency weighted errors are calculated using Formula 3.52.

From the results of frequency weighted errors, we can see that the new approach yields smaller frequency weighted errors than DSD and the existing rendering scheme using tent filters for all the testing images. Thus, the optimization in the frequency weighted color space works correctly.

5.2 SCIELAB Error Calculation

The SCIELAB errors between the original images and reproduced images are calculated as well. The viewing distance used in the SCIELAB is twice the display height.

Table 5.3 and Figure 5.3 show the SCIELAB errors of existing subpixel rendering schemes and the new approach.

The SCIELAB errors of the new approach are smaller than the existing subpixel rendering schemes for all the ten testing images. The results of SCIELAB errors verify that the optimization of frequency weighted errors for RGBG display patterns also works correctly for the SCIELAB color metric. The approximate linear relation between frequency weighted errors and SCIELAB errors can also be seen from the results.

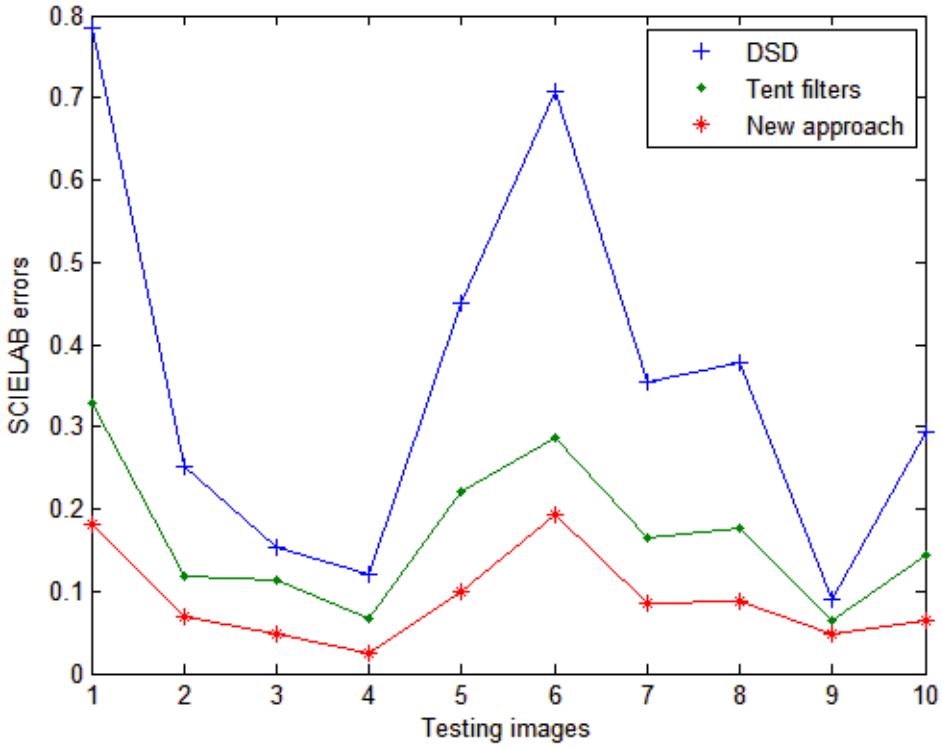


Figure 5.3: Plots of SCIELAB errors for the ten testing images

5.3 Visual Inspection

5.3.1 Photographic Images

Subpixel images of reproduced images using the new approach and the two existing methods are shown in the following figures.

In Figure 5.4, 5.5 and 5.6, subpixel images of the training images are displayed at 100%. The normal viewing distance is twice the display height. We can see that the subpixel images of the training image using the new approach has the least color fringing effect, which can be seen from the windows of the house and the snow on the roof. The existing subpixel rendering scheme using tent filters has less color distortion than the DSD method.

Figure 5.7, 5.8 and 5.9 are the subpixel images of Image 4 displayed at 200%, so the viewing distance is four times the display height. The color fringing is suppressed for the diagonal and anti-diagonal white bars on the boat using the new approach. The white bars are more colorful in reproduced images of the two existing methods.

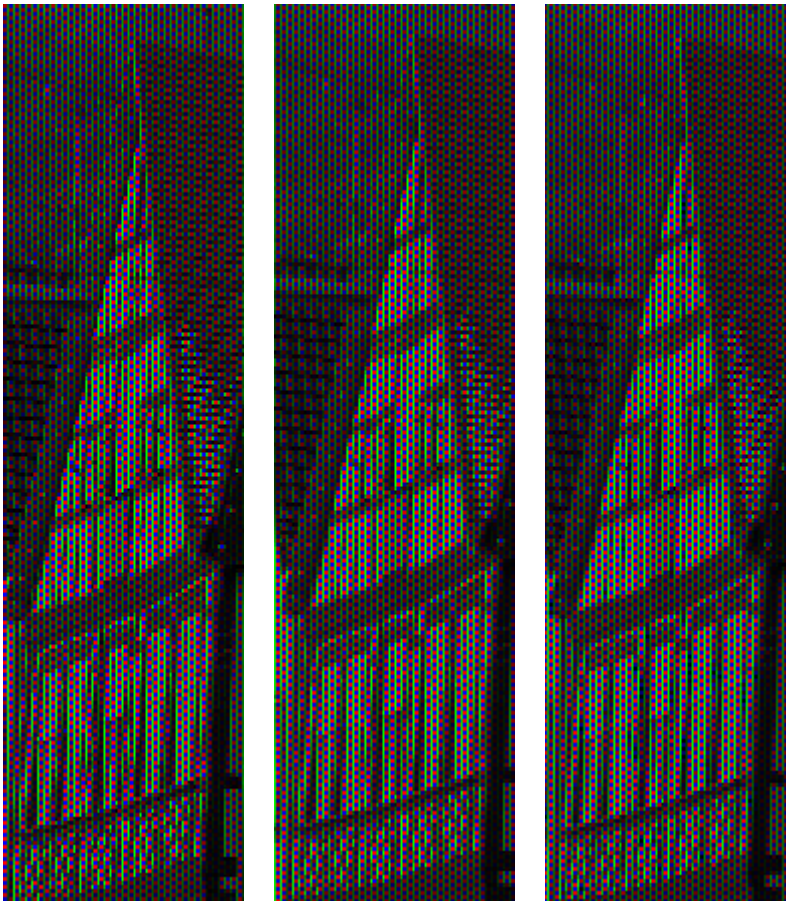
Another example is shown in Figure 5.10, which shows subpixel images displayed at 200%. We can see that there are less color fringing effect and more smooth edges of the necklaces in the down-sampled image obtained by using the new approach, compared with existing methods, at a viewing distance of four times the display height.

5.3.2 Text

Subpixel images of reproduced images of a text image are shown in Figure 5.11 at 300%.

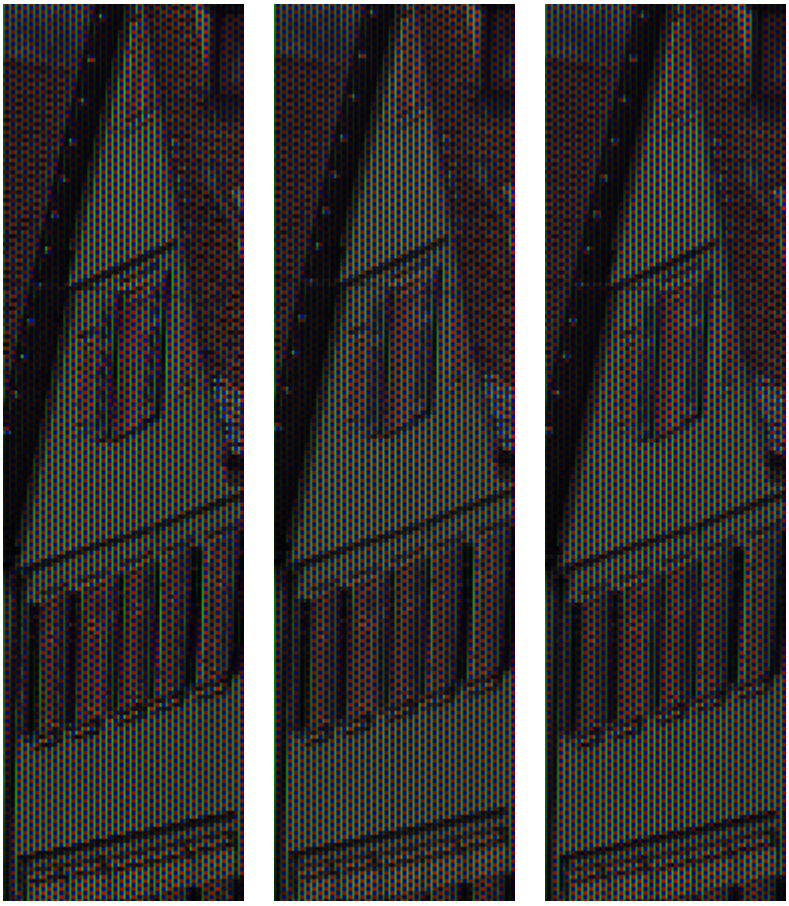
The subpixel images need to be viewed at a distance of six times the display height. When observers start moving back from a closer distance to the distance of six times the display height, the color fringing disappears in the subpixel image of the new approach, but still visible in other two subpixel images of existing methods.

We can see that more color fringing effect is reduced by the new approach than using the two existing methods. The numbers in the reproduced image of the new method keep more original black color, which makes the down-sampled text easier to recognize by human viewers.



DSD Tent filters The new approach

Figure 5.4: The subpixel images of the training image



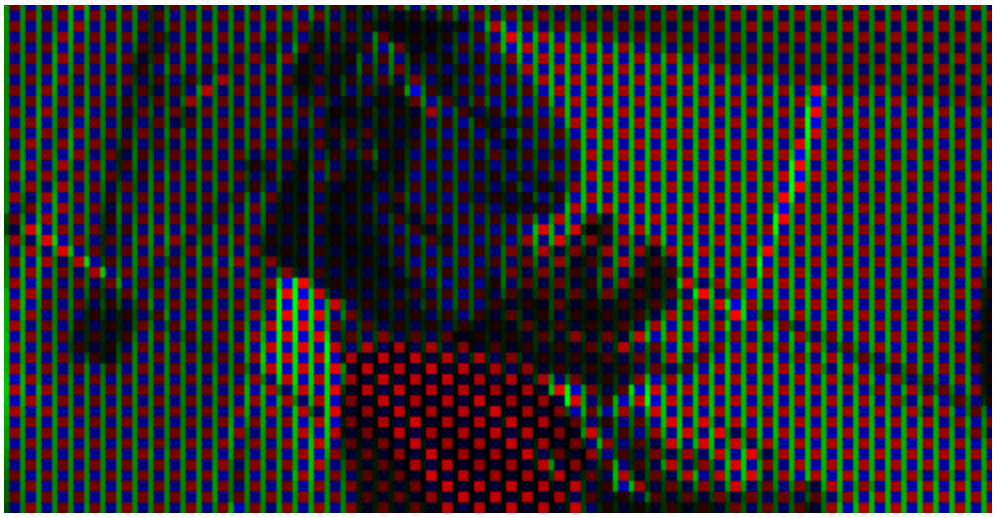
DSD Tent filters The new approach

Figure 5.5: The subpixel images of the training image

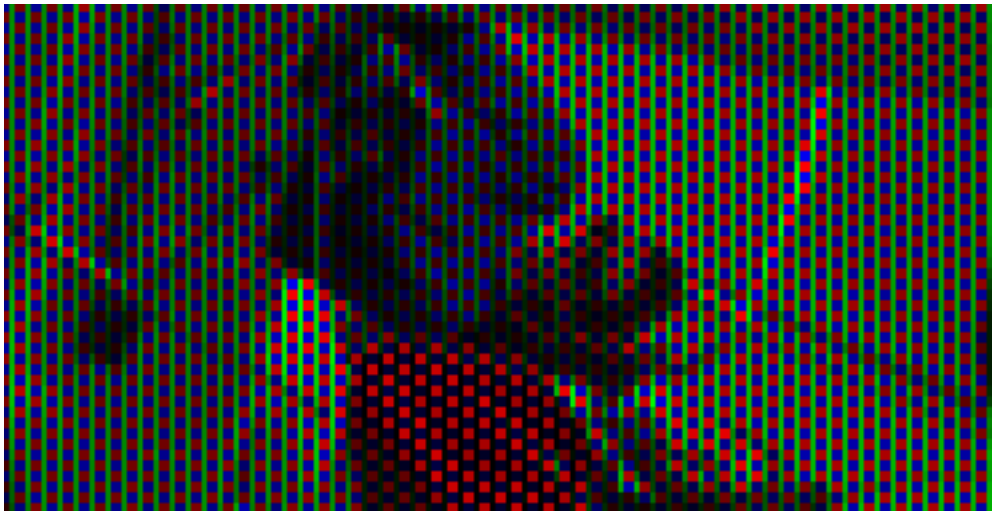
DSD Tent filters The new approach

Figure 5.6: The subpixel images of the training image

(a)



(b)



(c)

Figure 5.7: The subpixel images of Image 4 (a) DSD (b) Using tent filters (c) The new approach

(a)

(b)

(c)

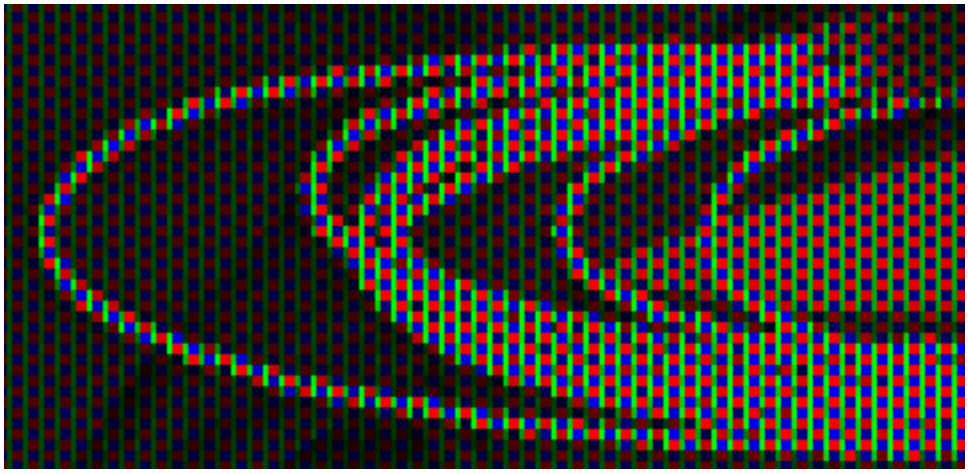
Figure 5.8: The subpixel images of Image 4 (a) DSD (b) Using tent filters (c) The new approach

(a)

(b)

(c)

Figure 5.9: The subpixel images of Image 4 (a) DSD (b) Using tent filters (c) The new approach

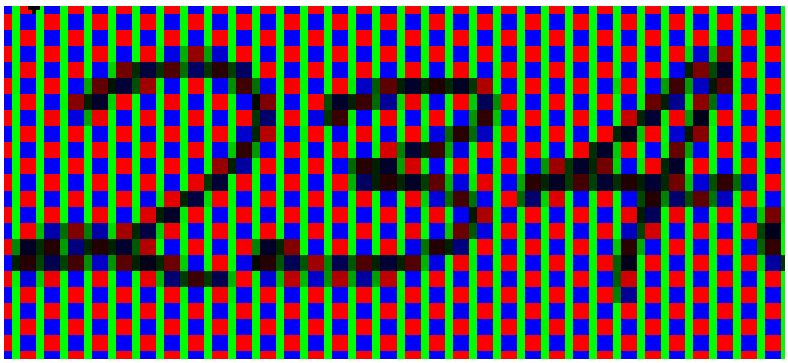


DSD

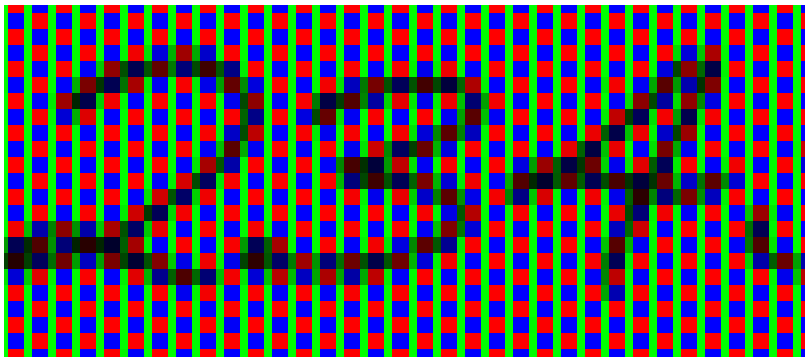
Tent filters

The new approach

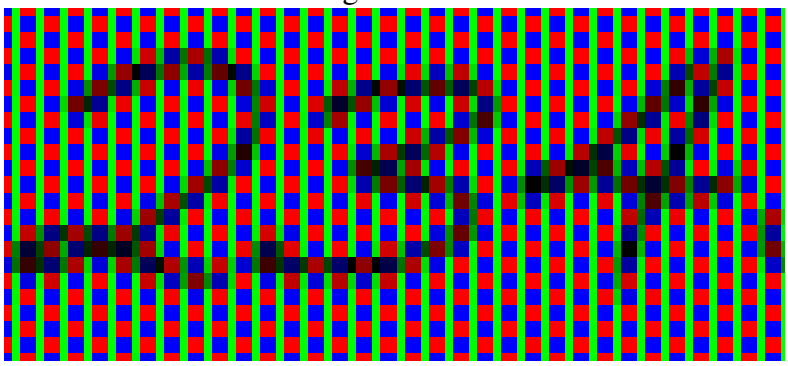
Figure 5.10: Displayed subpixel images of Image 7



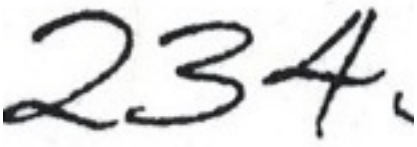
DSD



Rendering with tent filters



The new approach



Original

Figure 5.11: Displayed subpixel and original images of a text image

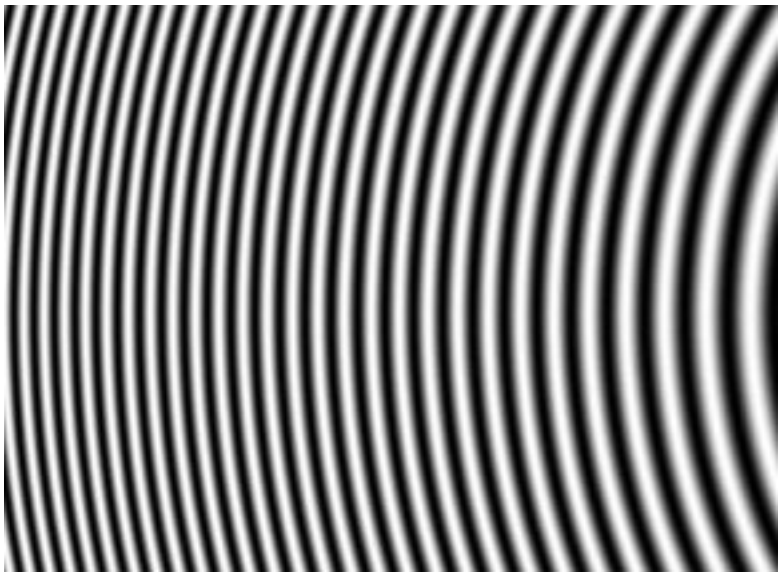


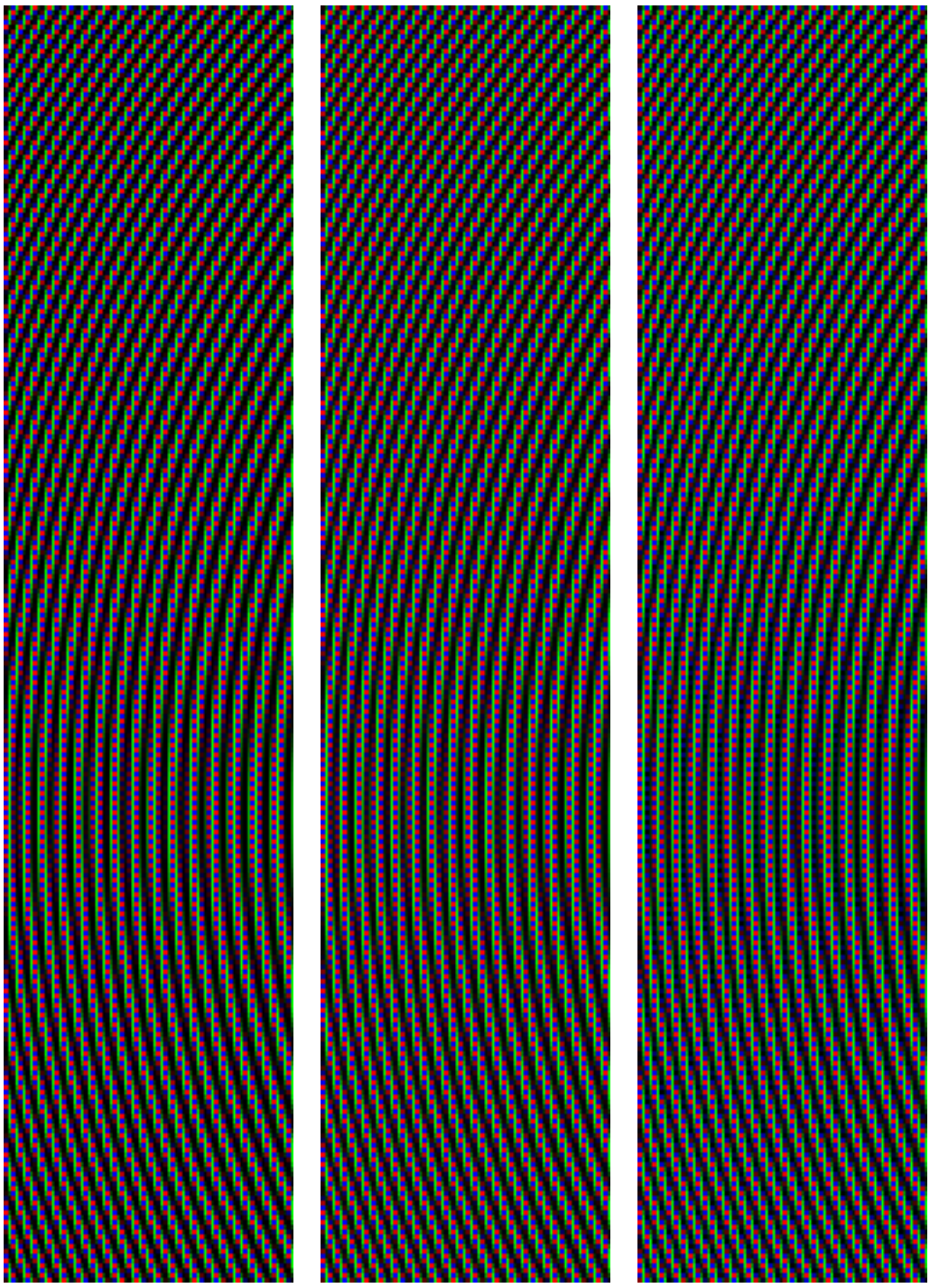
Figure 5.12: The original image of a zone plate

5.3.3 Zone Plate

Subpixel images of a reproduced zone plate, Figure 5.12, are shown in Figure 5.13. The subpixel images are displayed at 100%, so the normal viewing distance is twice the display height. Compared with the existing methods, the new approach suppress more aliasing effect of high frequency.

5.4 Summary

In this chapter, the Pentile RGBG display pattern is used to test the new approach. Comparing with existing methods, the frequency weighted errors and SCIELAB errors are minimized using the new approach. Reproduced images, including photographic images and text images, by the new approach for the Pentile RGBG display pattern have less color fringing effect and smaller color differences from original images. Thus, besides the RGB stripe display pattern, the new approach is also better for subpixel rendering on the Pentile RGBG display pattern.



DSD Tent filters The new approach

Figure 5.13: Displayed subpixel images of a zone plate

Chapter 6

Conclusions

6.1 Summary

In this thesis, a new approach to subpixel-based down-sampling to optimize the frequency weighted mean square errors in order to minimize errors under the SCIELAB measuring is developed. The system structure and theoretical results are presented for the new approach. In the down-sampling system, each color channel of original images is prefiltered by a set of optimized spatial filters and then down-sampled on the subpixel sampling structure of color mosaic displays. Mathematical solutions for the RGB stripe display pattern and the Pentile RGBG display pattern are given as two case studies.

Existing subpixel-based down-sampling methods are used to compare with the new approach. The two analytical models are implemented. The results show that the perceived color difference between original images and reproduced images is reduced using the new approach to subpixel rendering. The frequency weighted errors and SCIELAB errors are minimized. Color differences of images are suppressed, while details in images are still maintained well.

6.2 Thesis Contributions

6.2.1 System Structure

Theoretical system structures have been developed in this thesis to minimize the perceptual color difference of subpixel-based down-sampled images, such as the frequency

weighted mean square error and the SCIELAB error. No methods in the literature presents the down-sampling system with a correct linear shift-invariant prefiltering for color mosaic displays with subpixel structures. This research presents the down-sampling system with the linear shift-invariant prefiltering in 3D color space. Using this down-sampling system, the perceptual color difference is relieved. We can obtain down-sampled images with better quality for the human visual system.

6.2.2 Different Color Mosaic Displays

This new approach to image sampling structure conversion can be used not only for typical the RGB stripe display pattern, but also for other 2D display patterns. This method is a general sampling structure conversion method to improve the quality of reproduced images displayed on any color mosaic display pattern. It can be applied on existing display patterns and any other new 2D display pattern in the future.

6.2.3 Filters with Less Coefficients

Filter orders of the filter array used in the new approach are less than ten. Better image quality is obtained using the new approach, comparing with existing methods. With smaller filters, the speed of processing images will be increased, which is an advantage in fabricating color mosaic displays and image communications.

6.3 Future Work

6.3.1 Optimization of SCIELAB Errors

In this research, the target was originally to minimize SCIELAB error. As a proxy, we optimize frequency weighted mean square errors in SCIELAB opponent color space using SCIELAB filters in order to minimize SCIELAB errors. Future work could be to better study this approximation and to see if better results are obtained by minimizing SCIELAB error, using iterative optimization.

6.3.2 Other 2D Display Patterns

In my research, the typical RGB stripe display pattern and the widely used 2D display pattern, Pentile RGBG, were tested in reproducing images using the new approach to image sampling structure conversion. Other 2D display patterns can also be tested using this approach, for example, those display patterns using a triangle or a hexagon as the shape of

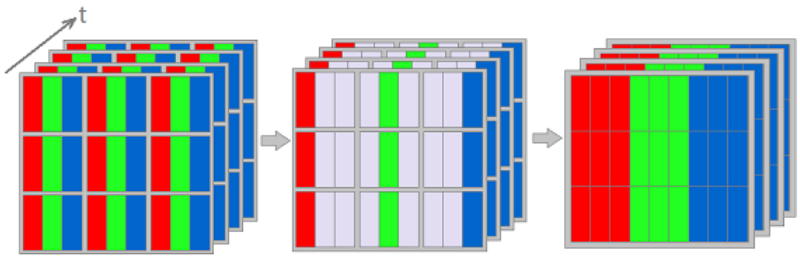


Figure 6.1: Video down-sampling

subpixels, instead of using a stripe or a square, and those display patterns which have more than three primary colors. Down-sampling models could be developed for these display patterns. Thus, the new approach can be used to reproduce images for them.

6.3.3 Video Processing

The new approach could also be used in video processing. When reproducing videos, each frame in videos is reproduced by changing the sampling structure, shown in Figure 6.1. We can use the new approach to convert the sampling structure of video frames for different kinds of color mosaic display patterns to obtain better video quality for the human visual system.

References

[1] E. Dubois, "The structure and properties of color spaces and the representation of color images," *Synthesis Lectures on Image, Video, and Multimedia Processing*, vol. 4, no. 1, pp. 1–129, 2009.

[2] G. Kutas, H.-K. Choh, P. Bodrogi, L. Czuni, and Y. Kwak, "Subpixel arrangements and color image rendering methods for multiprimary displays," *Journal of Electronic Imaging*, vol. 15, no. 2, pp. 023 002–023 002, 2006.

[3] X. Zhang and B. A. Wandell, "A spatial extension of CIELAB for digital color-image reproduction," *Journal of the Society for Information Display*, vol. 5, no. 1, pp. 61–63, 1997.

[4] L. Fang and O. C. Au, "Subpixel-based image down-sampling with min-max directional error for stripe display," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 2, pp. 240–251, 2011.

[5] L. Fang, O. C. Au, K. Tang, X. Wen, and H. Wang, "Novel 2-D MMSE subpixel-based image down-sampling," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 5, pp. 740–753, 2012.

[6] M. Kanazawa, "Color reproduction with a various number of sub-pixels," *Journal of the Society for Information Display*, 2013.

[7] J. Xu, J. Farrell, T. Matskewich, and B. A. Wandell, "Prediction of preferred ClearType filters using the S-CIELAB metric," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 361–364.

[8] M. A. Klompenhouwer and G. Haan, "Subpixel image scaling for color-matrix displays," *Journal of the Society for Information Display*, vol. 11, no. 1, pp. 99–108, 2003.

[9] D. S. Messing and S. Daly, "Improved display resolution of subsampled colour images using subpixel addressing," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 1. IEEE, 2002, pp. I–625.

[10] L. Fang, O. C. Au, K. Tang, and X. Wen, "Increasing image resolution on portable displays by subpixel rendering–a systematic overview," *APSIPA Transactions on Signal and Information Processing*, vol. 1, p. e1, 2012.

[11] X. Zhang, D. A. Silverstein, J. E. Farrell, and B. A. Wandell, "Color image quality metric S-CIELAB and its application on halftone texture visibility," in *Compcon'97. Proceedings, IEEE*. IEEE, 1997, pp. 44–48.

[12] J. Farrell, G. Ng, X. Ding, K. Larson, and B. A. Wandell, "A display simulation toolbox for image quality evaluation," *Journal of Display Technology*, vol. 4, no. 2, pp. 262–270, 2008.

[13] U. Rajashekar, Z. Wang, and E. P. Simoncelli, "Perceptual quality assessment of color images using adaptive signal representation," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2010, pp. 75 271L–75 271L.

[14] A. B. Poirson and B. A. Wandell, "Appearance of colored patterns: patterncolor separability," *JOSA A*, vol. 10, no. 12, pp. 2458–2470, 1993.

[15] S. A. Martucci, "Image resizing in the discrete cosine transform domain," in *Image Processing, 1995. Proceedings., International Conference on*, vol. 2. IEEE, 1995, pp. 244–247.

[16] L. Fang, O. C. Au, K. Tang, and A. K. Katsaggelos, "Antialiasing filter design for subpixel downsampling via frequency-domain analysis," *Image Processing, IEEE Transactions on*, vol. 21, no. 3, pp. 1391–1405, 2012.

[17] K. Hirakawa and P. J. Wolfe, "Fourier domain display color filter array design," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 3. IEEE, 2007, pp. III–429.

[18] J. C. Platt, "Optimal filtering for patterned displays," *Signal Processing Letters, IEEE*, vol. 7, no. 7, pp. 179–181, 2000.

[19] C. Betrisey, J. F. Blinn, B. Dresevic, B. Hill, G. Hitchcock, B. Keely, D. P. Mitchell, J. C. Platt, and T. Whitted, "20.4: Displaced filtering for patterned displays," in *SID Symposium Digest of Technical Papers*, vol. 31, no. 1. Wiley Online Library, 2000, pp. 296–299.

[20] R. W. G. Hunt, *The reproduction of colour*. John Wiley & Sons, 2005.

[21] C. Can and I. Underwood, "Compact and efficient RGB to RGBW data conversion method and its application in OLED microdisplays," *Journal of the Society for Information Display*, vol. 21, no. 3, pp. 109–119, 2013.

[22] L. Fang, O. C. Au, and N.-M. Cheung, "Subpixel Rendering: From Font Rendering to Image Subsampling [Applications Corner]," *Signal Processing Magazine, IEEE*, vol. 30, no. 3, pp. 177–189, 2013.

[23] C. Elliott, T. Credelle, S. Han, M. Im, M. Higgins, and P. Higgins, "Development of the PenTile matrix^{TM} color AMLCD subpixel architecture and rendering algorithms," *Journal of the Society for Information Display*, vol. 11, no. 1, pp. 89–98, 2003.