Finding Theme Color Palettes

Lai Wei
Department of Computing and Software
McMaster University

Supervisor
Jacques Carette

In partial fulfillment of the requirements for the degree of
Master of Engineering
January, 2019

Executive Summary

This project is about finding theme color palettes from images of famous paintings.

A color palette, in the digital world, refers to a full range of colors that can be displayed on a device screen or other interfaces, or in some cases, a collection of colors and tools for use in paint and illustration programs.

We can use the methods we present in this project to get color palettes from images we love which can be an interesting process. In our daily life, the theme color palettes can be used for clothing matching, interior design, and even plate presentations. These applications can let people's aesthetic taste and quality of life improve. Additionally, this project will enable more non-professionals to have more professional color perceptions.

Three different kinds of color quantization algorithms (k-means, median-cut, octree) are implemented to find theme color palettes of reference images in RGB and L*a*b* color spaces. We then use four objective image quality assessments (PSNR, SSIM, VIF, GMSD) to evaluate the fidelity of quantized images with the original images.

According to our experimental results, the color palettes obtained by different color quantization algorithms are different, and the k-means algorithm has the best performance in both RGB and L*a*b* color spaces, and the color space conversion (RGB to L*a*b* color space) improves the visual fidelity of images.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Theme color palettes of world-renowned paintings and beautiful photographic works have always been a vital source of inspiration for artists and designers. Why do professionals prefer these kinds of color combinations? One of the most important reasons is because they are considered as color harmony which plays an essential role in graphic design. In color theory, color harmony refers to the property that certain aesthetically pleasing color combinations have. These combinations can be of complementary colors, split-complementary colors, color triads, or analogous colors and so on.

Theme color palettes not only play a vital role in professional visual design but are also indispensable in our daily lives. If we can apply them to clothing, interior design, and even cooking, then we believe that there will be a significant improvement in the quality and happiness of life.

However, color choice is primarily guided by intuition and qualitative rules. As non-professionals or amateur designers, lack of a sense of aesthetics and color background knowledge will become a stumbling block for us to get theme color

information from the collection of flat artworks. Besides that, a full-color image can have up to 16 billions of different colors. What color combination can represent it? The answer varies from person to person. So it is a challenging but interesting task for most people to choose theme color palettes.

1.2 Overview

In order to enable more people to have a more professional and accurate understanding of color matching, we aim to directly offer harmonious color themes extracted from the targeted painting image. With a limited number of colors, The color themes can represent the color information of the original image to the greatest extent.

Color quantization is the process of reducing the number of distinct colors used in an image, which is aimed to enable the rendering of the image in devices with a limited number of colors, usually with the intention that the quantized image should be as visually similar as possible to the original image. Therefore, we can use this kind of compression to help us find the color combination of the target image.

Through color quantization, we can reconstruct the original image with limited theme colors and obtain quantized images. Through the fidelity computation between the original image and quantized images, we can know whether the color theme can be a good representative of the original image or not.

However, at the same time, a new question comes out: how can we improve the fidelity of quantized images with the reference image? With the development of color theory, the emergence of different color spaces aiming at addressing this issue seems to be a practical solution.

How to prove and perfect our idea: how many theme colors are suitable for this

image? Is the color space conversion effective? These questions can be answered subjectively through the human eye, but they are time-consuming and difficult to control. Another way is through objective image quality assessment (OIQA), which is fast and straightforward.

This project achieves the objective image quality evaluation by three color quantization algorithms to acquire the dominant colors of the specified image and visualize the quantized images and color theme palettes. Through the comparison between subjective and objective analysis, three main issues of the project we should address are

1. to figure out which algorithm enables the target image to have the highest fidelity after color quantization.

2. to verify if the quantized images can have a higher fidelity with the original image by the color space conversion.

3. to explore the influence of lightness information of images in the process of color quantization and OIQAs.

1.3 Related Work

Modeling, Rating and choosing harmonious color themes have been an area of interest in computer graphics for decades. The term color harmony is not, however, a concise, descriptive term; rather, it is one with a wide range of meanings because of any or all of several conditional attributes[2]. Kenneth's study identifies key attributes and related terms consistently used to describe color harmony. A geometric approach to harmonic color palette design was generated by [15]. It relies on the International Commission on Illumination standard for representing colors and evaluates proximity. In [19], they study color compatibility theories

using large datasets and create a new model for choosing colors. The basis of their model is the dataset of online rated color theme palettes with five colors, but we plan to generate color palettes from color quantization algorithms, so the input image and the number of theme colors can be customized. As we mentioned above, we can know that people are trying to address the problem of finding harmonious colors in many different ways.

Color quantization is aimed to design a color palette with limited numbers of colors having almost no noticeably perceived difference between reference and quantized images. This operation is widely used in computer graphics and image processing.

As for traditional quantization algorithms, there are many methods that have been generated to improve existed ones. Different initialization schemes of the k-means algorithm have been implemented in [4], and in [7] they focus on the deterministic initialization of the k-means based on the distribution of image pixels in the color space.

Generic roughness measure for effective segmentation [23] of color images and the pyramid data structure to refine the hierarchical analysis of images [32], and divisive hierarchical clustering [5] to improve the binary splitting strategy, all of these methods provide good support to the existing quantization algorithms.

More and more color quantization algorithms have been created aiming to have better performance. Using the greedy tree growing for image quantization [17], on spatial quantization [20], an ant colony clustering algorithm to group colors into certain clusters [9]. Additionally, the Hopfield [34] and integrated fuzzy neural networks [18] are applied to color quantization which has fast convergent speed and high quantization. In [36], contextual information in images is taken into consideration. In this algorithm, they locate the regions of an image having the greatest need for colors and allocate more quantization levels to them.

Satisfying human vision is one of the most essential research fields in image processing. Human perception based color quantization has developed a lot in these years. In [**28**] and [25], both of they account for the human visual system (HVS). Inverse image frequency is also applied to perception-based color image quantization which can be combined with any standard quantization algorithm [26].

Color space conversion has a very tight connection to the field of image processing. In [21], they propose a new model to simultaneously quantize and halftone color images. In their experiment, they study the role of the chosen color space from RGB to L*a*b* color space and conclude that deficiencies of the color space produce negligible defects compared to the artifacts introduced by the color reduction.

Image quality assessment is an important aspect of color quantization. In [**imageq**], a new image database for the evaluation of image quality metrics over color quantization noise is described and more than 19200 individual human quality judgments were carried out to obtain the final mean opinion scores. They accomplished both objective and subjective image quality assessments. Their methods can be regarded as a good template for this aspect.

According to the related work we obtained, the deeper exploration of color harmony and human vision system in line with color quantization algorithms can be viewed as potential development direction of our project.

Chapter 2

Requirements

Overview

In this chapter, we will describe our requirements of this project.

2.1 Functional Requirements

We will describe functional requirements in three main aspects which are shown below:

2.1.1 Inputs

There are two input requirements of our project:

1. Reference images: the input should be one or a corpus of images in RGB color space and .jpg format. The size of each original image should be under $1000 * 1000$.

2. A range of the number of theme colors: it can be customized by users based on different requirements of theme color palettes.

2.1.2 Operations

To achieve our project goal, there must be three main operations in the requirement shown in Figure 2.1:

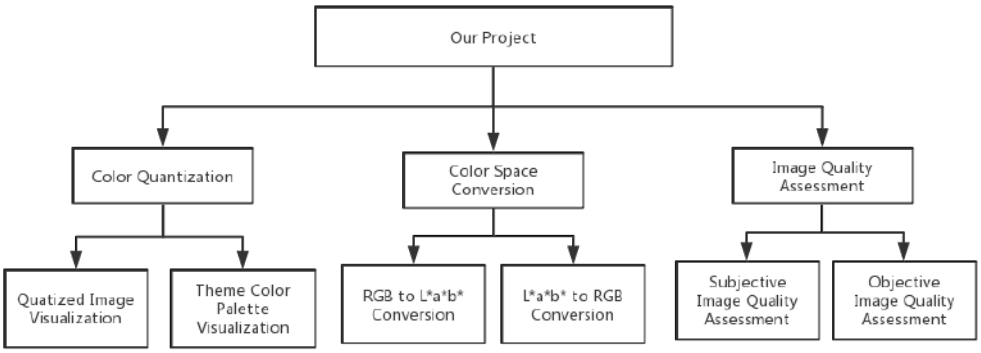


Figure 2.1: The Structure Diagram of operation Requirements

1. Color quantization: the number of colors in the reference image must be reduced to match the input requirement (the range of a number of theme colors).

2. Color space conversion: the original image must be converted into L*a*b* color space to prepare for the verification of the improvement of the visual fidelity of images.

3. Image quality assessment: a reference image and its quantized images must be inputs to accomplish OIQAs to evaluate the fidelity between quantized images and the reference image.

2.1.3 Outputs

The functional requirements of outputs are classified into two aspects:

1. Quantized images and theme color palettes: as for the range of the number of theme colors, there must be same number of quantized images and color palettes as outputs.

2. Results of OIQAs: all experimental results of each reference image should be present as a csv file.

2.2 Non-functional Requirements

Usability The runtime of the proposed methods need to be within a reasonable range.

Efficiency The proposed methods in the project should be usable and operable.

2.3 Constraints

Software Constraints

- Operating system: macOS Majove (version: 10.14.2)
- Compiling environment: python 2.7 IDE (Pycharm community 2017.03)
- Virtual environment and package manager: Anaconda

Hardware Constraints

- CPU: 2.3 GHz Intel Core i5
- Memory: 8GB
- Input/output devices: MacBook Pro (13-inch, 2017, Two Thunderbolt 3 ports)

Chapter 3

Design Details

Overview

Based on the project requirements, through our design we should meet three main functional requirements, inputs, operations, and outputs respectively. Design details in all three aspects are described below.

3.1 Inputs

3.1.1 Reference Images

For our project, reference images in sRGB color space will be selected into the implementation. Images are in sRGB color space by default. sRGB defines the chromaticities of the red, green, and blue primaries, the colors where one of the three channels is nonzero, and the other two are zero. The gamut of chromaticities that can be represented in sRGB is the color triangle defined by these primaries. As with any RGB color space, for non-negative values of R, G, and B it is not possible to represent colors outside this triangle, which is well inside the range of

colors visible to a human with normal trichromatic vision.[12]

3.1.2 The range of the number of theme colors

Users can select the range of a number they want.

3.2 Operations

3.2.1 Color Quantization

Color quantization can be viewed as a subset of the field of vector quantization. Simply stated vector quantization is the problem of selecting K vectors in some N-dimensional space to represent N vectors from that space where $K << N$ and the total error incurred by the quantization is minimized. Color quantization then can be regarded as vector quantization in 3-Dimensional space. Four main stages are consisting of the process of color quantization,[8]

1. Sampling the original image for color statistics.
2. Choosing a color map based on those statistics.
3. Mapping the colors to their representative in the color map.
4. Quantizing and drawing the new image.

The first three stages are connected tightly. In particular, the method used for stages 1 and 2 will determine the best method for accomplishing stage 3.

Although color quantization algorithms have experienced a lot of development, in this project, we will focus on three traditional widely-used color quantization algorithms: k-means, median-cut, and octree to let us have a fundamental and clear understanding of this field. In chapter 4, we will describe the implementation of these three quantization algorithms.

3.2.2 Color Space Conversion

In our design, we plan to implement our requirements in RGB and L*a*b color spaces. Design details in both color spaces are described below:

3.2.2.1 Design in RGB color space

Introduction of RGB color space RGB color space belongs to the additive color system, where all the colors are the mixtures of some colors in the visible light spectrum. Red, green and blue are the most common primary colors used in the additive color system, which is known as the RGB color model. RGB is a method of presenting colours electronically by projecting light rays on to screens, such as TV or computer screens. Each light ray has a unique wavelength that creates a specific color.[30]

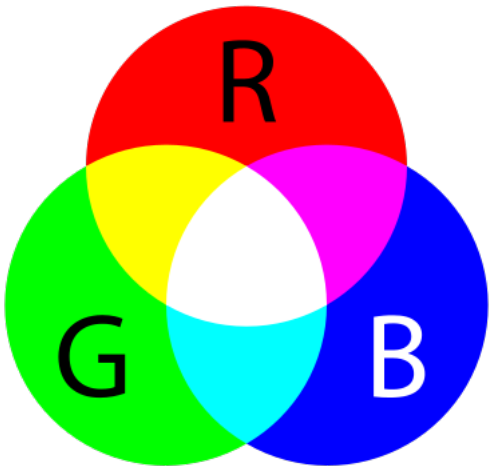


Figure 3.1: RGB color model

As Figure 3.1 shows, any two of standard additive primary colors (red, green and blue) combine into secondary colors, which are yellow, cyan, and magenta. Additive color is a result of the way the eye detects color and is not a property

of light. There is a vast difference between a pure spectral yellow light, with a wavelength of approximately 580 nm, and a mixture of red and green light. However, both similarly stimulate our eyes, so we do not detect that difference, and both are yellow light to the human eye.

3.2.2.2 Design in L*a*b* color space

Introduction of L*a*b* color space Based on our research, L*a*b* color space which was designed to perceptually uniform with respect to human color vision can be viewed as a good fit for our project. L*a*b* color space is the abbreviation for CIE L*a*b* color space defined by International Commission on Illumination (CIE) in 1976. It expresses color as three numerical values, L* for the lightness and a* and b* for the green-red and blue-yellow color components.

These three channel values are usually absolute values, with a pre-defined range. The lightness value, L*, represents the darkest black at L* = 0, and the brightest white at L* = 100. The scaling and limits of the a* and b* axes will depend on the specific implementation, but they often run in the range of 100 or 128 to +127 (signed 8-bit integer).[13]

Besides briefly introducing L*a*b color space, We summarized four key differences between RGB color space and L*a*b* color spaces shown below:

Device-dependent to absolute color space Three primary colors principle is applied in RGB color space, but there is no colormetric definition of red, green and blue, This means that we cannot regard the results of mixing primary colors as specified colors. The mixed colours are described relative to the primary colors[30]. It means that the same RGB combination will present slightly differently on different monitors.

L*a*b* color space is a reference color space. To be more specific, colors are

device-independent with a given white point which provides information about the light available in the scene in this color space. L*a*b* color space is built on a three-dimensional real number space, where we can define infinite numbers of possible representations of colors, and thus L*a*b* values are absolute in this space with a pre-defined range. The pre-defined range depends on a specified white point[13].

Expand the gamut Gamut is the complete subset of colors in color reproduction. The most common usage refers to the subset of colors which can be accurately represented in a given circumstance[16].

Chromaticity is an objective specification of the quality of a color regardless of its luminance. It consists of two independent parameters, often specified as hue (h) and colorfulness (s), where the latter is alternatively called saturation, chroma, intensity, or excitation purity[6].

RGB color space has a chromaticity gamut that is a color triangle, when the amounts of the three primary colors (red, green, and blue) are constrained to be nonnegative. Although red, green, blue are used widely in computer graphics, RGB color space is not sufficient to reproduce all colors. We can say that RGB color space provides a computationally tractable format to store images at the cost of displaying a smaller range of colors than human eyes can perceive[33].

On the contrary, the L*a*b* space is larger than the gamut of computer displays and printers. Its gamut includes both RGB color space and CMYK color model[22].

Fit for a computer to fit for human vision In RGB color space, all three primary colors are in the equal bandwidth (shown in figure 2.2) to produce any desired color, which can simplify the computation. However, human color vision does not fall neatly in a perfect cube, thus if we aim to mimic human color

perception, the bandwidth for each channel should be present differently, and additional information about lighting in the environment should include as well.

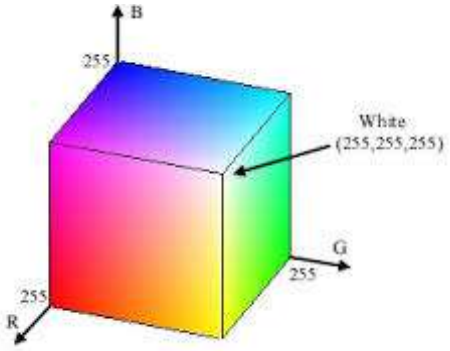


Figure 3.2: RGB cube - all three channels are limited to the same numerical range

Three channels of L*a*b* are L (black to white), a* (green to red), and b* (blue to yellow) are designed to satisfy those requirements mentioned above. This color space is shaped irregularly which is an intuitive way to organize colors for human vision[33].

Perceptually uniform The last but not the least goal of L*a*b* color space is to achieve perceptual uniformity, meaning that uniform changes of components in the L*a*b* color space correspond to the same amount of changes in the perceived color.

MacAdam ellipse is a region on a chromaticity diagram which contains all indistinguishable colors (all the colors in each ellipse are the same color to the average human eye). Therefore the contour of the ellipse represents the just noticeable differences (JND) of chromaticity. We can say that it partly led to this refinement of the CIE L*a*b* color space.

In the MacAdam's color match experiment, observers were asked to adjust color until it matched the test color. As a result, all the matches filled into an ellipse in the CIE 1931 chromaticity diagram, and there were 25 eclipses in total

with the different size and orientation. This experiment confirmed that color difference could be measured with a metric in a chromaticity space.

Although L*a*b* color space did not achieve this purpose completely, it is much less distorted than CIE XYZ color space[29], which means that the MacAdam ellipses become nearly circular in the L*a*b* color space.

3.2.3 Image Quality assessment

Four image quality assessment methods will be implemented in our project. There are PSNR, SSIM, VIF, and GMSD respectively. We will discuss more details of these four evaluation methods in chapter 5.

3.3 Workflows of the Design

In terms of the descriptions of each part of the project design, we decide to design two programs in RGB and L*a*b* color spacses respectively. Two block diagrams are shown below to present the workflows of our program in both color spaces.

3.3.1 Workflow in RGB color space

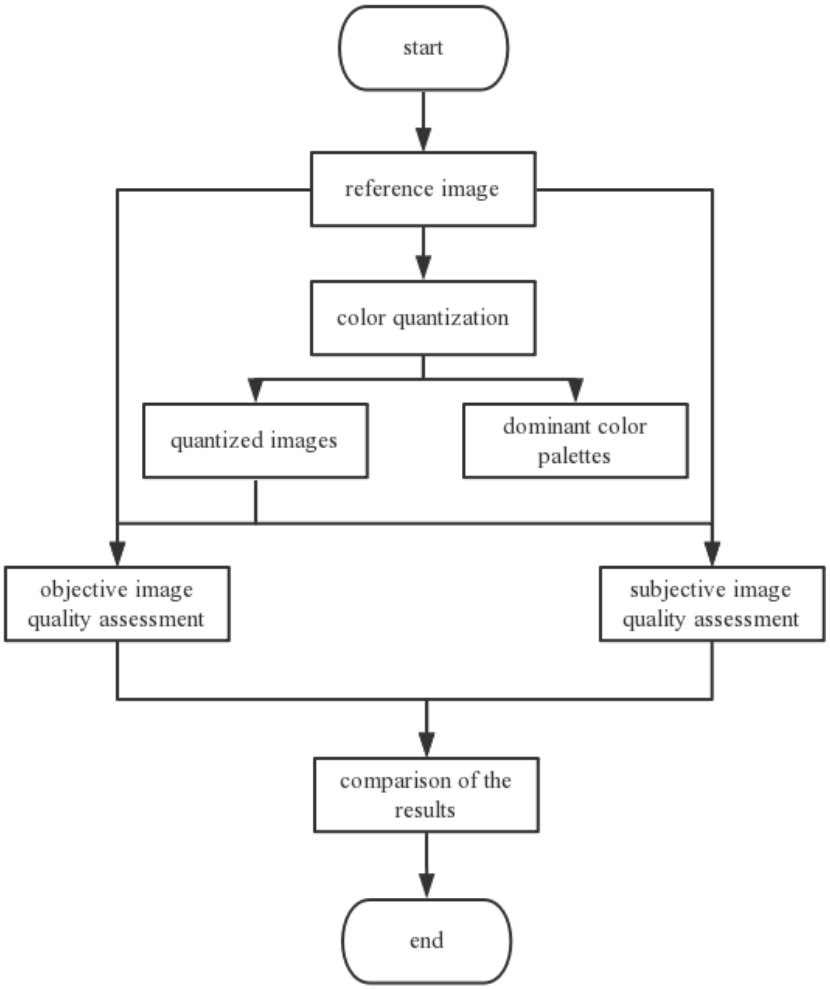


Figure 3.3: Workflow of "color quantization and image quality assessment in RGB color space"

The block diagram above gives an overview of how color quantization and image quality assessment will be implemented in RGB color space.

3.3.2 Workflow in L*a*b* color space

We use Figure3.4 to show the workflow of the program in L*a*b* color space.

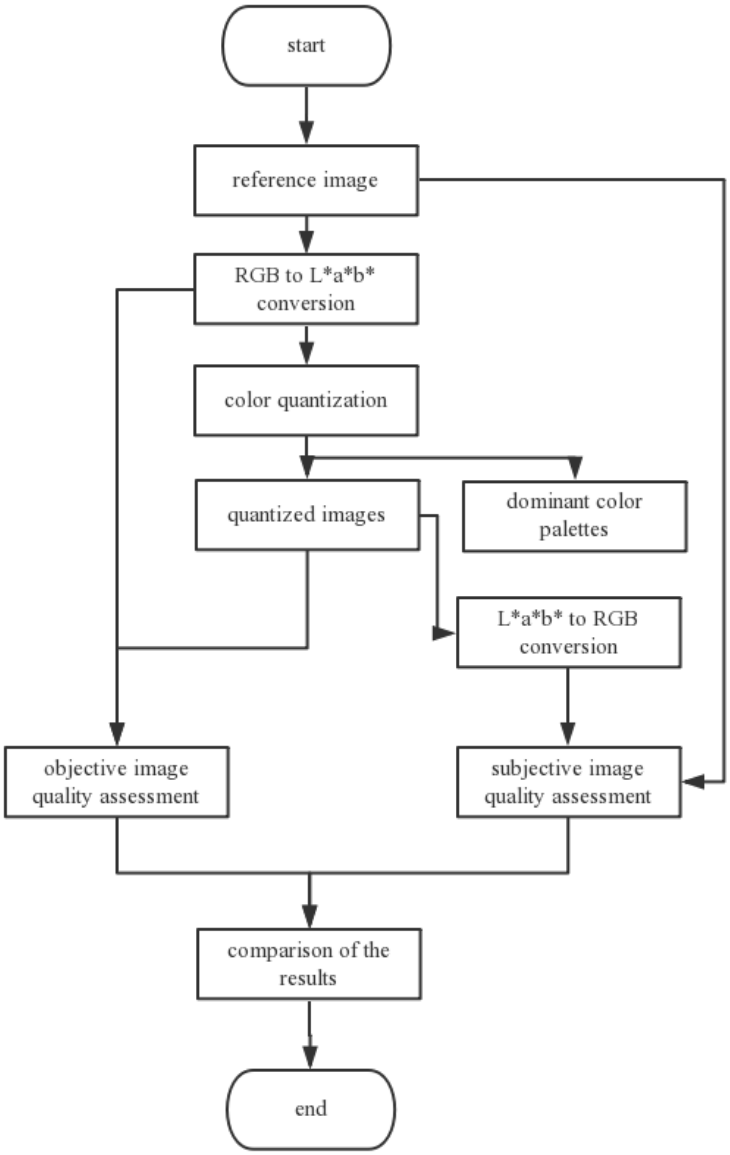


Figure 3.4: Workflow of "color quantization and image quality assessment in L*a*b* color space"

Compared to our design in RGB color space, there are two main differences in the design of L*a*b* color space shown below:

3.3.2.1 The conversion from RGB to L*a*b color space

Before implementing color quantization, we should transform the reference image from sRGB color space to L*a*b color space, and the process of the transformation is shown below:

sRGB color space to XYZ color space sRGB color space is based on the CIE xyY coordinate. The CIE XYZ values must be scaled so that the Y of D65 ("white") is 1.0 $(X, Y, Z = 0.9505, 1.0000, 1.0890)$.

The components of sRGB $(R_{srgb}, G_{srgb}, B_{srgb})$ are in the range 0 to 1. The first step is to transform sRGB to RGB which can be viewed as the non-linear-to-linear transformation.

$$C_{linear} = \begin{cases} \frac{Csrgb}{12.92} & Csrgb \leq 0.0405 \\ (\frac{Csrgb+\alpha}{1+\alpha})^{2.4} & Csrgb > 0.0405 \end{cases}, \text{ where } \alpha = 0.055 \text{ and } C \text{ is } R, G \text{ and } B \tag{3.1}$$

Followed by a matrix multiplication of the linear RGB values to get XYZ:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1875 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{bmatrix} \tag{3.2}$$

Theory of the transformation

The sRGB gamma cannot be represented as a single numerical value, and there is only an approximate median value – 2.2. The transformation is based on this decoding gamma, but with the linear portion near zero to avoid having an infinite

slope at K=0, which can cause numerical problems. The continuity condition for the curve C_{linear}, which is defined above as a piecewise function of C_{srgb}:

$$(\frac{K_0+\alpha}{1+\alpha})^\gamma = \frac{K_0}{\phi} \tag{3.3}$$

$\gamma = 2.4$ and $\phi = 12.92$ yields two K values,$K_0 \approx 0.0381548$ and $K_0 \approx 0.0404482$. The IEC 61966-2-1 standard uses a rounded value $K_0 = 0.0405$.

XYZ color space to L*a*b* color space The last step is to convert XYZ color space to L*a*b* color space. [13]

$$\begin{cases} L^* = 116 f(\frac{Y}{Y_n}) - 16 \\ a^* = 500(f(\frac{X}{X_n}) - f(\frac{Y}{Y_n})) \\ b^* = 200(f(\frac{Y}{Y_n}) - f(\frac{Z}{Z_n})) \end{cases} \quad where \quad f(t) = \begin{cases} \sqrt[3]{t} & \text{if } t > \delta^3 \\ \frac{t}{3\delta^2} + \frac{4}{29} & \text{otherwise} \end{cases}, \delta = \frac{6}{29} \tag{3.4}$$

Here, Xn, Yn and Zn are the CIE XYZ tristimulus values of the reference white point (the subscript n suggests "normalized").

Speaking of the CIE XYZ tristimulus value, as for the human eye with the normal vision, there are three kinds of cone cells that sense light that underlie the human color perception. Weighting a total light power spectrum by the individual spectral sensitivities of the three kinds of cone cells renders three effective values of stimulus; these three values compose a tristimulus specification of the objective color of the light spectrum. The CIE 1931 color space defines the resulting tristimulus values, in which they are denoted by "X", "Y", and "Z".[10]

Theory of the transformation

The division of the domain of the $f(t)$ function into two parts was done to prevent

an infinite slope at $t = 0$. The function $f(t)$ was assumed to be linear below some $t = t_0$, and was assumed to match the $t^{\frac{1}{3}}$ part of the function at t_0 in both value and slope. In other words:

$$\begin{cases} (t_0)^{\frac{1}{3}} = mt_0 + c & \text{(match in value)} \\ \frac{1}{3}(t_0)^{-\frac{2}{3}} = m & \text{(match in slope)} \end{cases} \tag{3.5}$$

The intercept $f(0) = c$ was chosen so that L^* would be 0 for $Y = 0 : c = \frac{16}{116} = \frac{4}{29}$. The above two equations can be solved for m and t_0:

$$\begin{cases} m = \frac{1}{3}\delta^{-2} & = 7.787037... \\ t_0 = \delta^3 & = 0.008856... \end{cases} \quad where \quad \delta = \frac{6}{29} \tag{3.6}$$

3.3.2.2 The conversion from L*a*b* color to RGB color space

After color quantization, in order to visualize quantized images, we should convert them back to RGB color space before the subjective quality image assessment.

3.4 Design Alternative

As we can see, two programs in RGB and L*a*b* color spaces are very similar, so it is durable to combine these two programs together whose workflow is shown in Figure 3.5. But our implementation is based on the original program design due to the time limit.

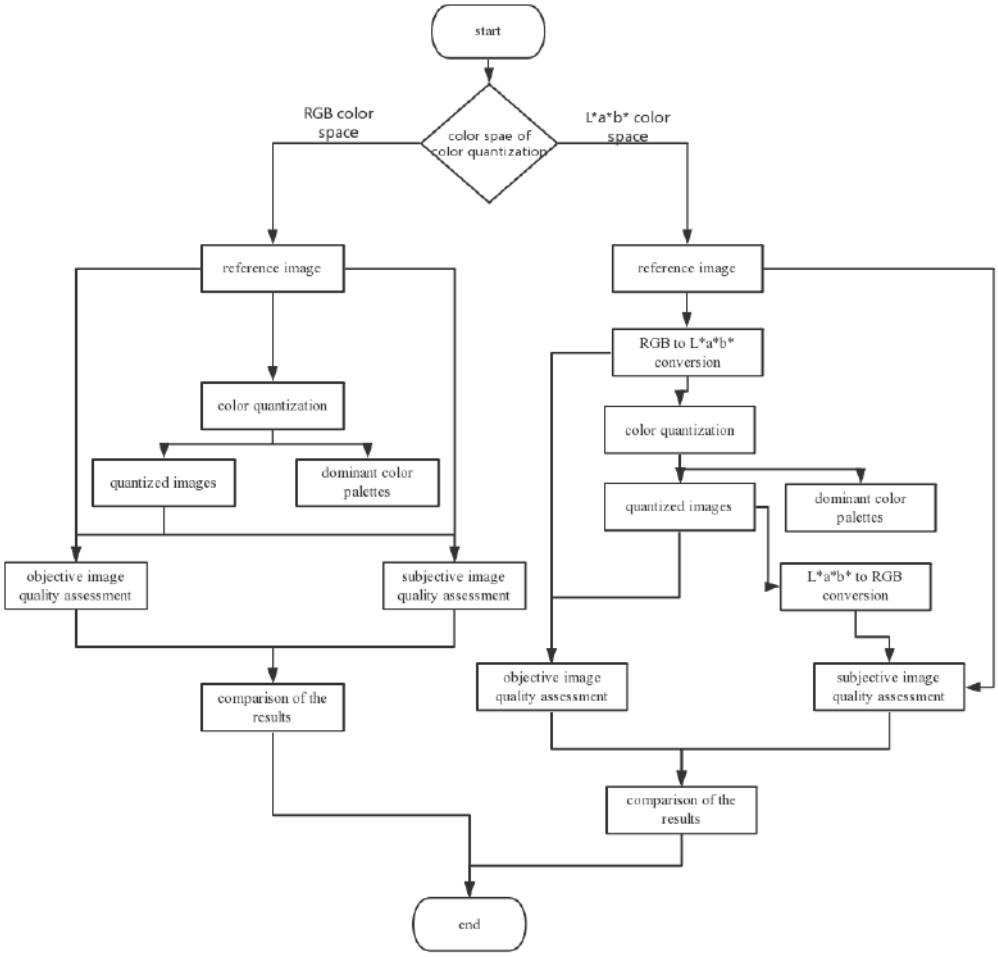


Figure 3.5: Workflow of the alternative program design

Chapter 4

The Implementations of Three Color Quantization Algorithms

Overview

Based on our design description in chapter 3, in this chapter, we will show implementation details of three different kinds of quantization algorithms (k-means, median-cut, and octree) with 11 reference images in both RGB and L*a*b* color spaces.

The 11 reference images we selected are Van Gogh's famous painting "The Starry Night" (Figure 4.1) and 10 visually different images of famous paintings (Figure 4.2). These 11 images include different paining styles and have different color themes, which can represent the variety of the input and lead to general implementing conclusions to a certain extent.

Based on our project design, the range of the number of theme colors can be customized, so we set it to "1 to 20" as an example for these 11 original images. And we also set this range as our experimental standard.

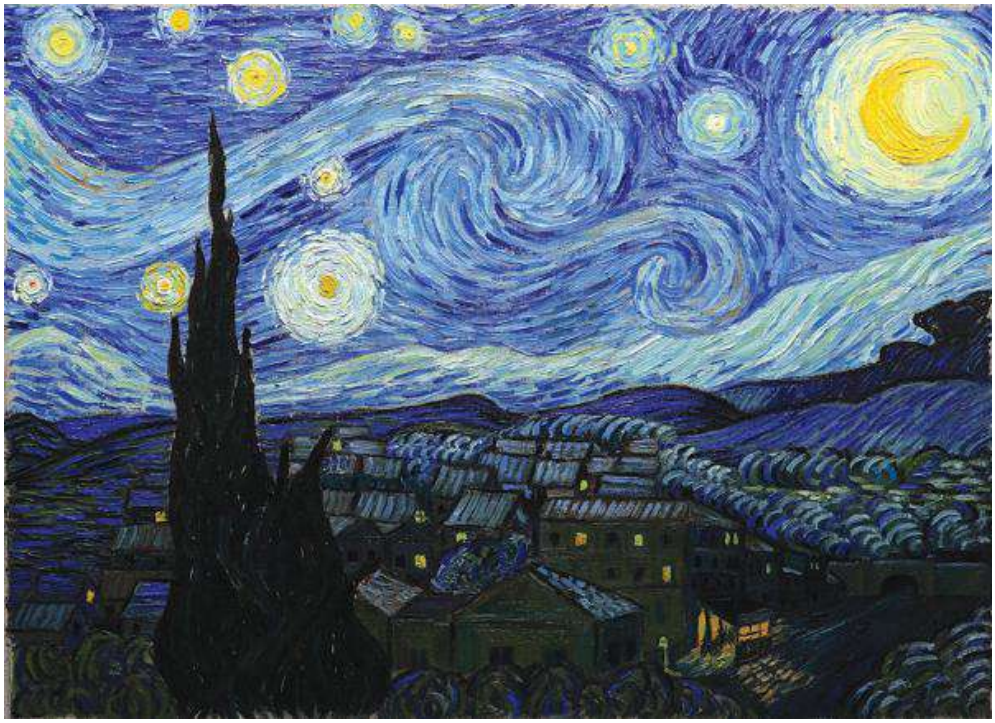


Figure 4.1: "The Starry Night" in RGB color space

Besides that, we also defined "the key image" – the first image that is visually similar to the original image (the shape of the moon shows up), and we also visualized the dominant-color palette for each quantized image.

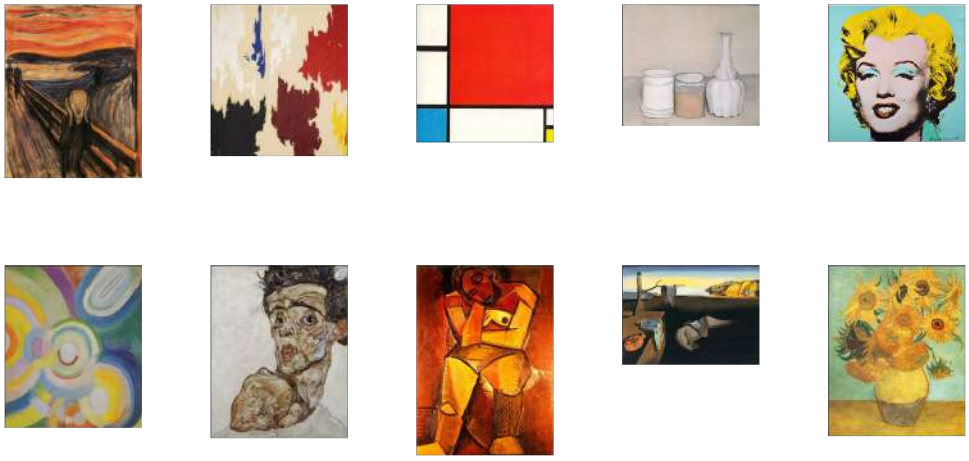


Figure 4.2: 10 original images of paintings in RGB color space

In this chapter, we will discuss the details of each algorithm and display selected quantized images that can summarize the visual performance of all quantized images. We will give a detailed discussion of Figure 4.1 and summarized descriptions of Figure 4.2. Also, we will compare the outputs of the three color quantization algorithms concerning each other in both two color spaces.

4.1 K-means Algorithm

4.1.1 Introduction

One of the most common approaches dealing with color quantization is to use a clustering algorithm. In this case, we chose k-means clustering algorithm that can be easily applied to both RGB and L*a*b* color spaces, because the distance used by the k-means is Euclidean distance which is a natural fit for both. [3]

K-means algorithm consists of four main steps:

1. Randomly select k mean pixel points from the image.

2. K clusters are created by associating each point to the nearest mean point.

3. Compute and place the centroid of each cluster.

4. Repeat step 2 and 3 until all points have been reached, and all centroids of k clusters become the dominant colors of the original image.

In RGB color space, an image is made of 3 channels: red(R), green(G) and blue (B), and there are also three channels in L*a*b* color space. Therefore, in both color spaces, we can think of each pixel of the image as a point in a 3D space and so can apply k-means approach on the same.

4.1.2 Application

4.1.2.1 Implementation in RGB color space

We applied k-means algorithm directly to the RGB original image to get the dominant color palettes. After that, each color in the original image was remapped to the nearest one among those in the palette and then we got the quantized images.

Displayed selected quantized images and their color theme palettes, as Figure 4.3 shows below, we can see the quality of quantized images experienced a dramatic increase from 1 to 6 dominant colors. The key "yellow" color of the moon shows up in the 6th quantized image. Although the visual performance improves with each additional color in the palette, this trend decreases rapidly after reaching the peak (the 6th quantized image). Once the palette contains 15 colors, humans might be unable to tell the difference between subsequent images.

(a) 1 main color (b) 2 main colors (c) 3 main colors

(d) 4 main colors (e) 5 main colors (f) 6 main colors

(g) 7 main colors (h) 8 main colors (i) 9 main colors

(a) 10 main colors (b) 15 main colors (c) 20 main colors

Figure 4.3: selected quantized images with their color palettes

As Figure 4.4 shows below, we can know that k-means has different performances on different reference images. In terms of our observation, the third figure at the left column where there are only five theme colors reaches a very high fidelity with the reference image, while figures at the right column show that quantized images are very similar with the original image when there are over fifteen dominant colors.

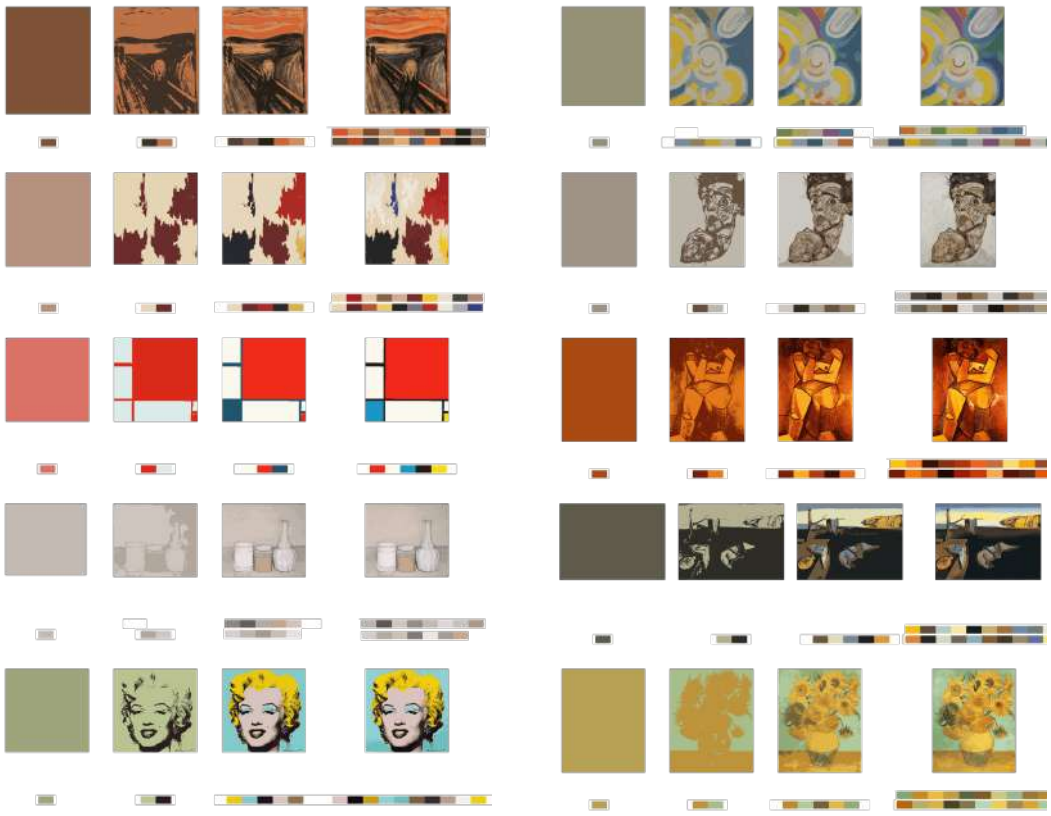


Figure 4.4: Selected quantized image from 10 original images in RGB color space

4.1.2.2 Implementation in L*a*b* color space

As for the implementation in L*a*b* color space, the original image was converted into L*a*b* color space array first, and then it went to the next step: building the k-means model. After the color quantization, we got the L*a*b* quantized

image arrays, but we could not visualize those arrays directly. Instead, they were transformed back into RGB color space in order to match the visual effect of the RGB original image.

As Figure 4.5 In the series of L*a*b* quantized image arrays, the "yellow" color which represents "lightness" in the original image shows up very early (with only four dominant colors) on the sky and even houses. We can see the shape of the moon in the 8th image, and then the image becomes brighter, but there is no significant visual change happening with the number of dominant colors increasing.

(a) 1 main color

(b) 2 main colors

(c) 3 main colors

(d) 4 main colors

(e) 5 main colors

(f) 6 main colors

(g) 7 main colors

(h) 8 main colors

(i) 9 main colors

(j) 10 main colors

(k) 15 main colors

(l) 20 main colors

Figure 4.5: selected quantized images with their color palettes in L*a*b* color space

Compared with quantized images in RGB color space, we can see that images generally become brighter in L*a*b* color space in Figure 4.6, which matches one feature of L*a*b* color space: sensitive to brightness information.

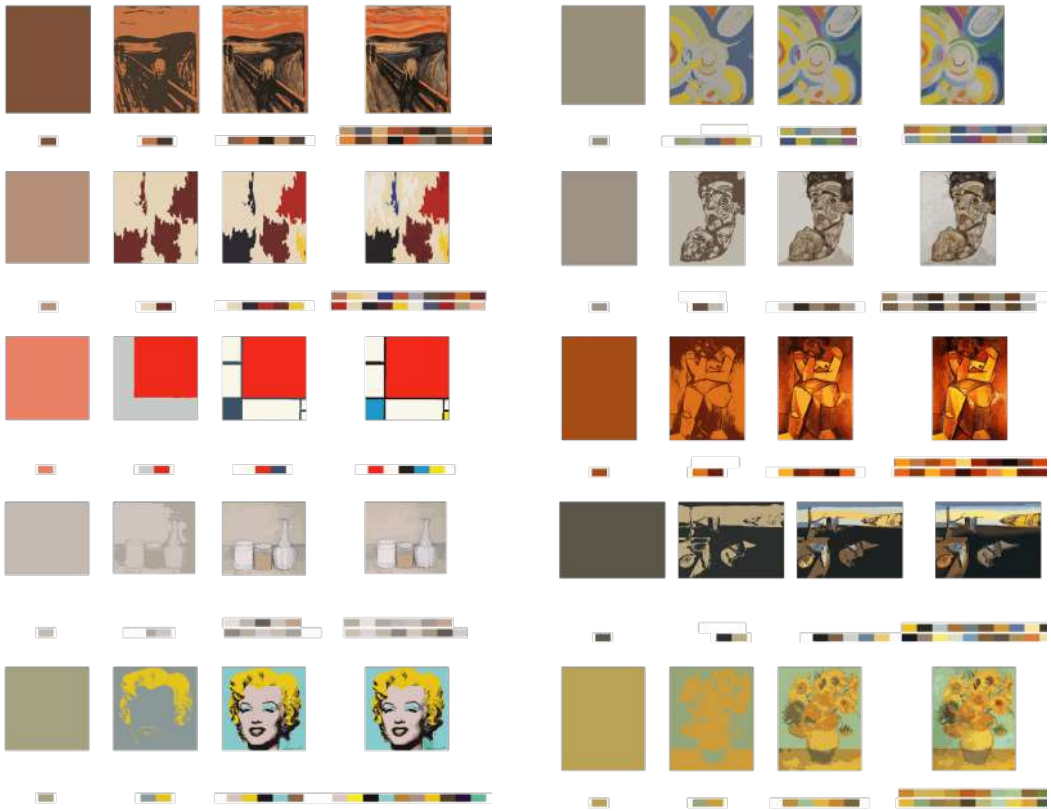


Figure 4.6: Selected quantized image from 10 original images in L*a*b* color space

4.1.2.3 Conclusion

We can have a better visual understanding of these two color spaces by comparing all key images we obtained.

As for RGB color space, although the key image appears earlier than in L*a*b* color space, the key image is much brighter and more precise in L*a*b* color space, which we can say it has a better visual effect.

(a) 6th quantized image in RGB (b) 6th quantized image in L*a*b*

Figure 4.7: Comparison to the 6th quantized images in both color spaces

(a) 8th quantized image in RGB (b) 8th quantized image in L*a*b*

Figure 4.8: Comparison to the 8th quantized images in both color spaces

4.2 Median-cut Algorithm

4.2.1 Introduction

The median-cut algorithm is typically used for color quantization. It is designed to sort data of an arbitrary number of dimensions into a series of sets by recursively cutting each set of data at the median point (or mean point) with the longest dimension. [24]

4.2.2 Application

4.2.2.1 Implementation in RGB color space

There are two main steps for implementing median-cut algorithm:

1. Create a "cube" of the colors in the pixels of an image by using each color component (R, G, and B) as an axis (e.g. x, y, z):
    - Calculate the range of each color component (R, G, and B).
    - For the component with the largest range, C, calculate the mean value, M.
    - Split the "cube" of colors:
        (a) one cube containing the RGB values of all pixels where the C component is greater than M
        (b) one cube containing the RGB values of all pixels where the C component is less than M
    - If the number of cubes is equal to our chosen number of desired colors, exit the loop.
    - For each color cube, calculate the range of each component, choose the cube which contains the largest range, and repeat.
2. For each cube, apply the mean function to the value of each component, and combine into a new RGB value.

As Figure 4.9 shows below, we know that the 7th quantized image can be regarded as the key image. Additionally, the more dominant colors, the more details of the sky displaying in the images, while we only got a rough shape of those dark houses at the bottom of the images. These results explained why bright colors (like yellow, light blue) dominate the color theme palettes we generated.

(a) 1 main color (b) 2 main colors (c) 3 main colors

(d) 4 main colors (e) 5 main colors (f) 6 main colors

(g) 7 main colors (h) 8 main colors (i) 9 main colors

(a) 12 main colors (b) 15 main colors (c) 20 main colors

Figure 4.9: selected quantized images with their color palettes in RGB color space

As we can see in Figure 4.10, quantized images with two or three theme colors

are quite visually different with original images, because we use mean value for each cutting which means some mean values may not exist in the original image. With the number of dominant colors increasing, quantized images become much more similar to the reference image.

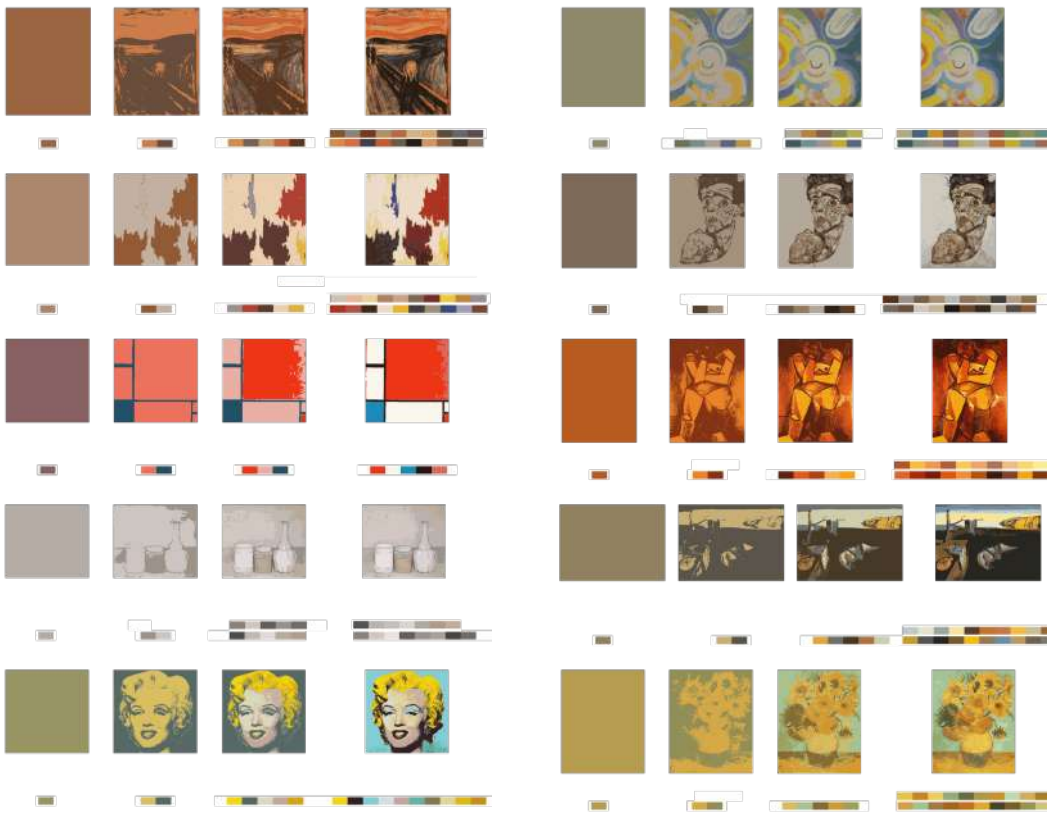


Figure 4.10: Selected quantized images from 10 original images in RGB color space

4.2.2.2 Implementation in L*a*b* color space

We built two different L*a*b* cubes to compare the visual effect of two different levels of lightness in images.

Firstly, We converted the RGB "cube" into L*a*b* cube by changing three channels (R, G, B) to L*, a*, b* channels directly, so as for this cube, each cutting in the median-cut recursion entirely depends on the original range of

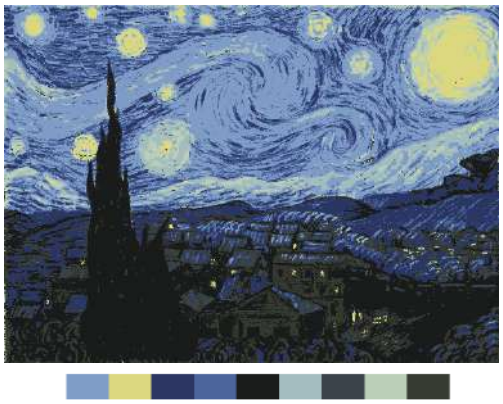
each dimension.

While in the second cube, we increased the length of the L* channel manually. In this case, the new range of the L* channel is 1.5 times longer than the original one, which means that there will be more median cuttings in the L* dimension than before.

(a) 7th quantized image from first cube

(b) 7th quantized image from second cube

(c) 9th quantized image from first cube



(d) 9th quantized image from second cube

(e) 10th quantized image from first cube

(f) 10th quantized image from second cube

(a) 18th quantized image from first cube (b) 18th quantized image from second cube

Figure 4.15: selected quantized images from two L*a*b* color space cubes

Looking at the images generated from the first cube (left column of Figure 4.15), we can see that the bright colors (which represent lightness in the houses, the moon, and stars) appear in the 9th quantized image for the first time, but the visual similarity with the original image wasn't reached until the 18th quantized image showed up.

When focused on the right side of Figure 4.15, it clearly shows that increasing the number of dominant colors from 7 to 9 in color quantization didn't lead to any noticeable visual change of quantized images, but when the number of dominant colors added to 10, the quantized image and the original image are highly similar.

Comparing to images from both L*a*b* color cubes, we can summarize four key points:

1. The first time brightness colors appeared with less dominant colors from the second cube.

2. The high visual similarity reached with less dominant colors from the second cube.

3. Images from second cube have a higher similarity with the original image than those from the first cube with the same number of dominant colors,

although the difference between images become less noticeable from our eyes.

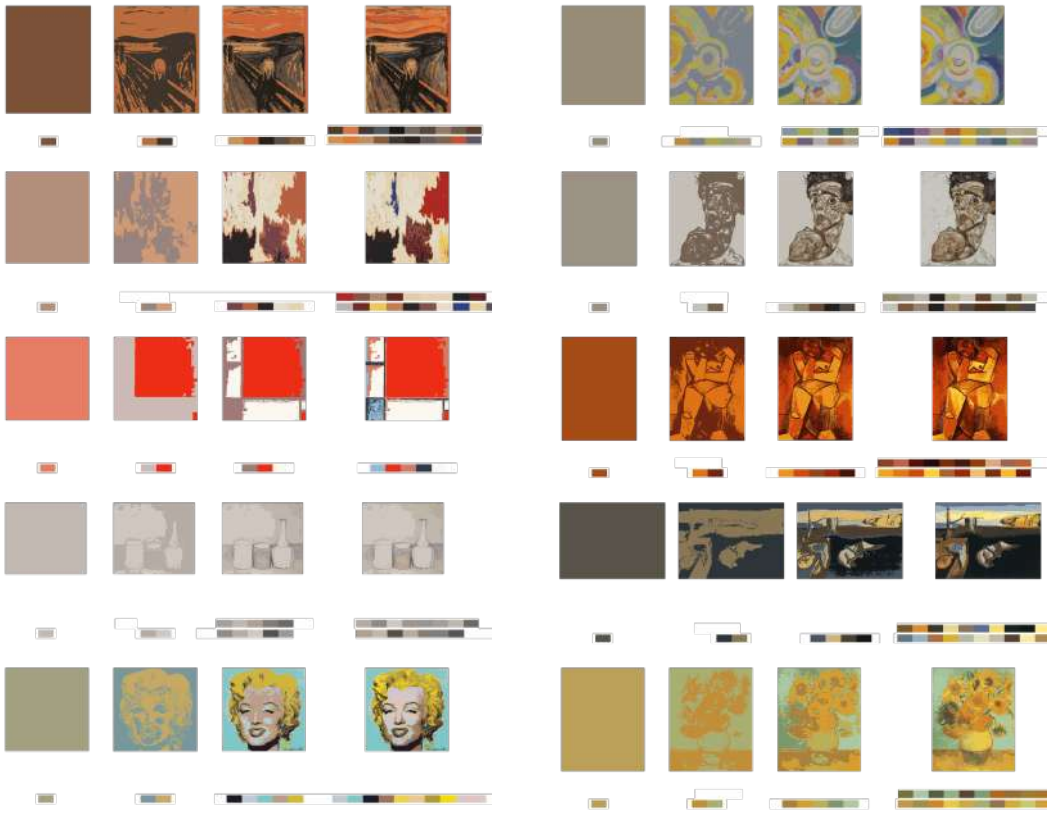


Figure 4.16: Selected quantized images from 10 original images from the first L*a*b* cube

In Figure 4.16, selected quantized images show the visual trend of each original image we chose. With the increase of number of theme colors, the quantized images become clearer and more similar with the original image, but different images achieve high fidelity with the original image in different numbers of theme colors.

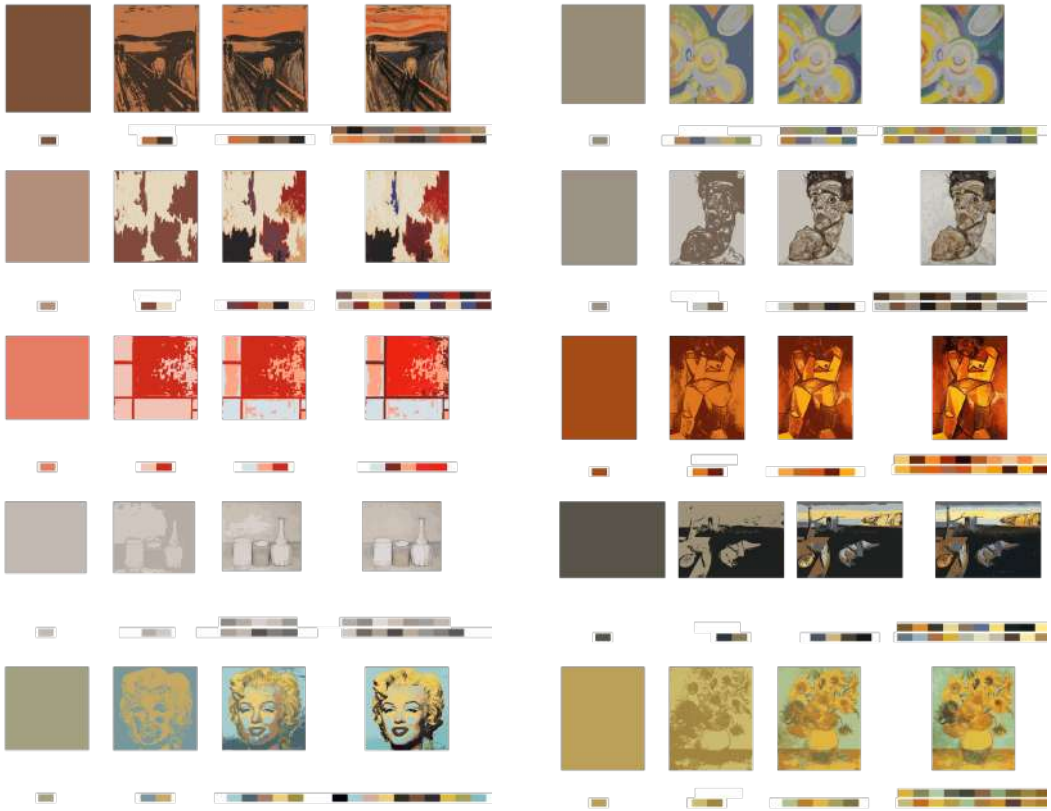


Figure 4.17: Selected quantized image from 10 original images from the second L*a*b* cube

Comparing Figure 4.16 and Figure 4.17, we know that adding L* channel range can generate different theme color palettes of different images. And for the third figure with two theme colors at the left column in Figure 4.17, we think there is a big improvement of fidelity with longer L* channel range.

4.2.2.3 Conclusion

As Figure 4.18 shows, the key images are the 6th, the 18th and the 10th in those three situations respectively. Based on this, we can know that, as for the median-cut approach, it is useful to have a better visual fidelity by manually speeding up the occurrence of "brightness" in the L*a*b* image.

(a) 6 colors in RGB (b) 6 colors in 1st L*a*b* (c) 6 colors in 2nd L*a*b*

(d) 7 colors in RGB (e) 7 colors in 1st L*a*b* (f) 7 colors in 2nd L*a*b*

(g) 10 colors in RGB (h) 10 colors in 1st L*a*b* (i) 10 colors in 2ndL*a*b*

(a) 18 colors in RGB (b) 18 colors in 1st L*a*b* (c) 18 colors in 2ndL*a*b*

Figure 4.18: the key images in three situations

4.3 Octree Algorithm

4.3.1 Introduction

An octree is a data structure with up to 8 nodes as children to store 3-dimensional data of any form, which can be classified as a hierarchical clustering method.

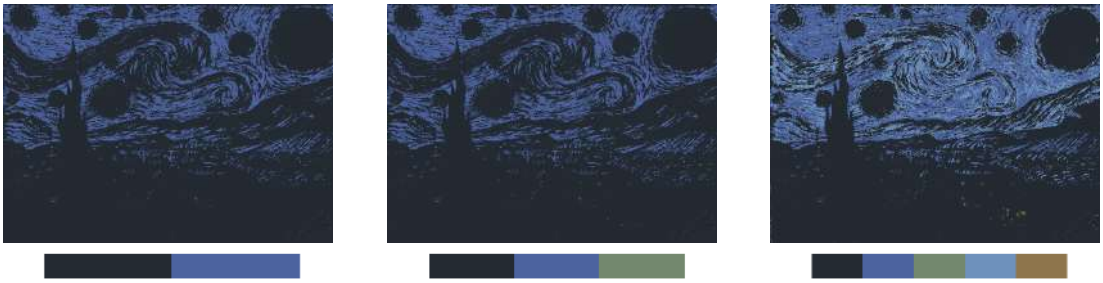
4.3.2 Application

4.3.2.1 Implementation in RGB color space

RGB color space is the premise of the octree algorithm. Each channel of RGB color space has the same numerical range from 0 to 255, so all colors of an image can be represented as an octree with eight levels (8 bits can store color in a binary way).[31]

In RGB color space, the octree method consists of three key stages:

1. Insert all colors of an image into an octree and count the number of pixels
2. Prune nodes in the tree until K leaves remain (K means the maximum number of colors we set in the image) by adding all leaves pixels count and color channels to parent node and making it a leaf.
3. Build the color palette with k colors from the average colors of each leaf by dividing color channels by the number of pixels.
4. Reduce leaves in the palette from the highest level of the tree until N (the number of dominant colors we set) leaves remain.



(a) 2 main colors (b) 3 main colors (c) 5 main colors

(d) 7 main colors (e) 8 main colors (f) 9 main colors

(g) 10 main colors (h) 14 main colors (i) 15 main colors

(a) 16 main colors (b) 17 main colors (c) 20 main colors

Figure 4.19: selected quantized images with their color palettes in RGB color space

As we can see in Figure 4.19, the number of colors in the dominant palettes

has a non-linear relationship with the visual fidelity of quantized images, which means that the visual results did not experience an up or down trend with the increasing number of primary colors.

The reason why this happened is that the octree algorithm only focuses on the distribution of colors in an image, but does not consider the frequency of colors.

In terms of this, in the original image Figure 3.1, we know that the different kinds of yellowish colors can be classified as low-frequency and bluish colors occupy the image, but they are not dominant colors in the color palettes. Additionally, the shape of the moon appears only 3 of all 20 quantized images, 7th, 15th , and 16th respectively. Although the frequency of yellowish colors is very low, it may play an essential role in the visual effect of this image.

The non-linear relationship also appears when we apply the octree method to more images. As Figure 4.20 shows, there is little difference from 1 to 2 dominant colors for most images. Additionally, as for the fifth figure at the left column, the quantized images with 2 and 20 theme colors are almost identical.

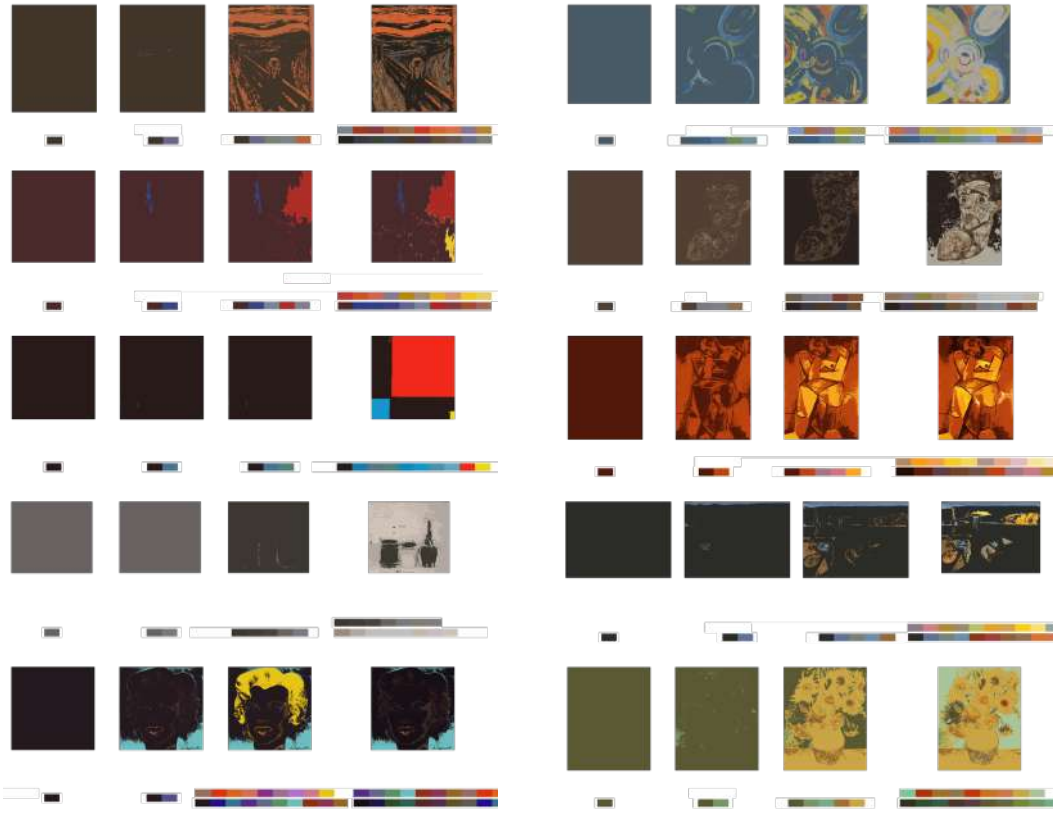


Figure 4.20: Selected quantized images from 10 original images in RGB color space

4.3.2.2 Implementation in L*a*b* color space

Due to the range of each channel in L*a*b* color space, there are two primary deficiencies by using the octree method to implement color quantization:

1. As we described in chapter 2, the range of L* channel is from 1 to 100, and possible values in a* and b* channel are in range [-128,127], which means that it is enough to use 4 bits to represent one channel instead of 8, so the octree method might be a waste using in L*a*b* color space.

2. Negative values may appear in both a* and b* channels, so not all colors of the original image in L*a*b* color space can be stored in the same octree

of RGB color space.

To experience the influence of these two shortcomings more directly, we applied the octree method in L*a*b* color space by using the same octree approach in RGB color space to insert colors directly, and selected quantized images are shown in Figure 3.14.

(a) 2 main colors (b) 3 main colors

(a) 8 main colors (b) 20 main colors

Figure 4.21: selected quantized images with their color palettes in L*a*b* color space

In Figure 4.21, the pixel loss of quantized images (at the left bottom part)

verified the second issue of the octree method in L*a*b* color space - colors in the negative value range cannot be displayed correctly in the quantized images. But the visual fidelity with the quantized images becomes better with more dominant colors which are different in RGB color space.

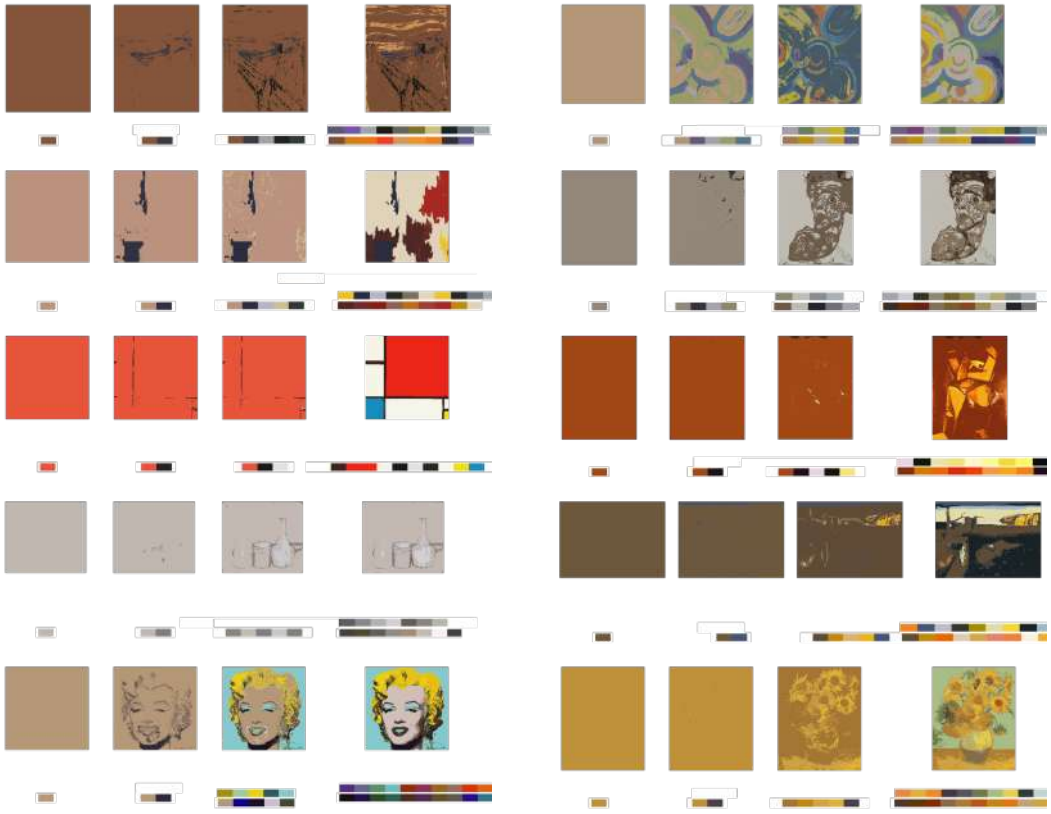


Figure 4.22: Selected quantized images from 10 original images in L*a*b* color space

Although there is still little noticeable visual difference in quantized images from one to two dominant colors for most images in Figure 4.22, the visual fidelity becomes higher and higher with the increase of numbers of theme colors.

4.3.2.3 Conclusions

In Figure 4.28, as for L*a*b* color space, although there is pixel loss in some quantized images,there is an increasing trend of the visual fidelity with the increasing number of dominant colors.

(a) 7th quantized image from RGB

(b) 7th quantized image from L*a*b*

(c) 11th quantized image from RGB

(d) 11th quantized image from L*a*b*

(e) 15th quantized image from RGB

(f) 15th quantized image from L*a*b*

(a) 16th quantized image from RGB

(b) 16th quantized image from L*a*b*

(c) 20th quantized image from RGB

(d) 20th quantized image from L*a*b*

Figure 4.28: selected images from two color spaces

Overall, more quantized images in L*a*b* color space have high fidelity than those in RGB color space, so we can say the octree method performs better in L*a*b* color space.

4.4 Conclusion

We summarize all key images of three color quantization algorithms in Table 4.1.

k-means		median-cut			octree	
RGB	L*a*b*	RGB	1^{st} L*a*b*	2^{nd} L*a*b*	RGB	L*a*b*
# 6	# 8	# 7	# 18	# 10	# 7	# 11

Table 4.1: the key images of three quantization approaches

As Table 4.1 shows, for both RGB and L*a*b* color spaces, the key images appear earliest from k-means algorithm among all three color quantization algorithms.

More quantized images from k-means and median-cut algorithms have high level fidelity with the original image (Figure 3.1) than those from the octree method in both two color spaces.

Speaking of the comparison between RGB and L*a*b* color spaces in each algorithm, we can conclude that:

1. For k-means algorithm: the key image appears earlier in RGB color space, and the overall visual difference of quantized images in these two color spaces is trivial.

2. For median-cut algorithm: the key image appears earlier in RGB color space, and the overall visual difference of quantized images in these two color spaces is trivial.

3. For octree algorithm: the key image appears earlier in RGB color space, and the overall visual fidelity of quantized images in L*a*b* color space is much higher than those in RGB color space.

In conclusion, we can say that:

1. k-means works best for the original image among all three algorithms.

2. L*a*b* color space perform better than RGB color space. It improves the visual effects of quantized images in all three algorithms to varying degrees.

3. the combination of k-means algorithm and L*a*b* color space can generate best visual performing quantized images.

Chapter 5

Objective Image Quality Assessment

Overview

After the color theme extraction, it is an indispensable step to evaluate the quality of those quantized images.

There are three classifications of IQA models [14]:

1. Full-reference (FR) methods assess the quality of a test image by comparing it to a reference image that has perfect quality.

2. Reduced-reference (RR) methods access the quality of a test and reference based on the extracted features from both images

3. No-reference (NR) methods access the quality of a test image without any information from the reference image

The aim of our project is to find color palettes as good representatives of reference images, so we focus on the FR-IQA models, which can include all color

information of original images and are used widely in the evaluation of image processing algorithms. Generally speaking, there are two main approaches to design FR-IQA metrics: top-bottom framework, which aims to model the overall function of the human visual system (HVS) based on some global assumptions, and bottom-top framework which simulates the various processing stages in the visual pathway of HVS. We used four different kinds of FR-IQA metrics to evaluate the quality of those quantized images.

5.1 Peak Signal-to-Noise Ratio (PSNR)

In the physical world, any quantity exhibiting variation in time or variation in space (such as an image) is potentially a signal that might provide information on the status of a physical system, or convey a message between observers, among other possibilities.[28]

Thus, from pixel information, PSNR computes the peak signal-to-noise ratio between two images. The signal in this case is the original image data, and the noise is the error introduced by color quantization. This ratio is used as a quality assessment between the reference image and quantized images.

5.1.1 Introduction

PSNR is defined via mean square error (MSE). Given a noise-free monochrome mn image X (the original image)and its noisy approximation Y(its quantized image), MSE is defined as:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [X(i,j) - Y(i,j)]^2 \tag{5.1}$$

The PSNR (in dB)is defined as:

$$\begin{aligned} PSNR &= 10\log_{10}\frac{MAX_1^2}{MSE} \\ &= 20\log_{10} MAX_1 - 10\log_{10} MSE \end{aligned} \tag{5.2}$$

In the equation above, MAX_1 represents maximum fluctuation in the input image datatype. For example, in grayscale images, MAX_1 is the maximum possible pixel value of the image, and for RGB images the value of MAX_1 is 255 when there are 8 bits in per pixel. To be more general, MAX_1 is $2^B - 1$ where B represents B bits per pixel. [11]

5.1.2 Application

PSNR is designed for grayscale images by default, but many approaches exist for computing PSNR of color images. Since mimicking human vision is the goal of this experiment, PSNR should be as close to the assessment of human eyes as possible.

In addition to mimicking human vision, we must implement a different version of PSNR in RGB and Lab* color spaces to match the features of both color spaces.

5.1.2.1 RGB color space

For color images in RGB color space (size $m \times n$), the range of three channels (red, green, and blue) is identical, so MSE for each channel is defined as:

$$MSE_{(RGB/channel)} = \frac{1}{3mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[X(i,j) - Y(i,j)]^2 \tag{5.3}$$

Based on this, the value of PSNR for each channel is same for a RGB image, and the relationship of PSNR in RGB and Grayscale color spaces is shown below:

$$MSE_{(RGB/channel)} = \frac{1}{3} PSNR_{(Gray)} \tag{5.4}$$

5.1.2.2 L*a*b* color space

As we discussed in chapter 3, fitting for human vision is one of vital design purposes of L*a*b* color space. And human eyes are most sensitive to luma information which represents brightness in an image. With this consideration, in L*a*b* color space we can separate L* channel and only compute PSNR on it, so new MSE is defined as:

$$MSE_{(L^*a^*b^*)} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [X_{(L^*)}(i,j) - Y_{(L^*)}(i,j)]^2 \tag{5.5}$$

When we convert the color space of images from RGB to L*a*b*, the datatype of the image also experiences a transformation, to double-precision floating-point data type, so the value of MAX_1 of PSNR becomes 1 in L*a*b* color space. We can define PSNR in this color space as follows:

$$PSNR_{(L^*a^*b^*)} = 10 \log_{10} \frac{1}{MSE_{(L^*a^*b^*)}} \tag{5.6}$$

5.2 Structural Similarity (SSIM)

SSIM is a perception-based model that considers image degradation as a perceived change in structural information. As it shows above, PSNR focuses on the absolute errors between images, so SSIM is designed to improve this traditional method.

5.2.1 Introduction

SSIM consists of 3 components L for luminance, C for contrast, S for structural information.[1] While still considering luminance and contrast masking terms, the structural information also aims to find the inter-dependencies among pixels, mainly they are spatially close. When we compute SSIM of the original image x and the quantized image y:

$$\begin{cases} L(x,y) = \frac{2\nu_{xy}+C_1}{\nu_x^2+\nu_y^2+C_1} \\ C(x,y) = \frac{2\delta_{xy}+C_2}{\delta_x^2+\delta_y^2+C_2} \\ S(x,y) = \frac{\delta_{xy}+C_3}{\delta_x+\delta_y+C_3} \end{cases} \tag{5.7}$$

where

ν_x = the average of x

δ_x = the variance of x

δ_{xy} = the covariance of xy

$C_1 = k_1L^2, C_2 = k_2L^2, C_3 = \frac{C_2}{2}, k_1, k_2 << 1$ are very small constants.

$k_1 = 0.01, k_2 = 0.03$ by default

L = the dynamic range of the pixel value, typically it is $2^{\#bits \quad per \quad pixel} - 1$.

Then, SSIM can be defined as the combination of the three comparisons:

$$SSIM(x,y) = L(x,y)^\alpha \cdot C(x,y)^\beta \cdot S(x,y)^\gamma \tag{5.8}$$

In order to simplify the formula, we set $\alpha = \beta = \gamma = 1$, so the SSIM formula can be reduced to the form shown below when measuring between the reference image x and the test image y of size $N \times N$ is:

$$SSIM(x,y) = \frac{(2\nu_{xy}+C_1)(2\delta_{xy}+C_2)}{(\nu_x^2+\nu_y^2+C_1)(\delta_x^2+\delta_y^2+C_2)} \tag{5.9}$$

In practice, one usually requires a single overall quality measure of the entire image. Mean SSIM (MSSIM) index is designed to evaluate the overall image quality:

$$MSSIM(x, y) = \frac{1}{M} \sum_{j=1}^{M} SSIM(x_j, y_j) \tag{5.10}$$

A sliding window approach is included in the computation. The window moves pixel-by-pixel across the whole image space. x_j and y_j are the image contents at the $j_(th)$ local window (8×8 square local window by default). and M is the number of local windows of the image.

5.2.2 Application

5.2.2.1 RGB color space

As for RGB color space, MSSIM can be applied directly to the reference image and quantized images.

$$MSSIM_{(RGB)} = MSSIM_{(x,y)} \tag{5.11}$$

5.2.2.2 L*a*b* color space

Since human eyes have high sensitivity to lightness, I think we can use these two alternative methods shown below to refine SSIM in L*a*b* color space.

L* channel computation Inspired from computing PSNR in L*a*b* color space, we can only compute SSIM for the L* channel of images. Since luminance plane consists of significant information in the image the evaluation score obtained

is closer to human vision judgement.

$$\begin{aligned} MSSIM_{(L*a*b*)} &= MSSIM(x_{(L*)}, y_{(L*)}) \\ &= \frac{1}{M}\sum_{j=1}^{M} SSIM(x_{(L*)_j}, y_{(L*)_j}) \\ &= \frac{(2\nu_{x_{(L*)}y_{(L*)}} + C_1)(2\delta_{x_{(L*)}y_{(L*)}} + C_2)}{({\nu_{x_{(L*)}}}^2 + {\nu_{y_{(L*)}}}^2 + C_1)({\delta_{x_{(L*)}}}^2 + {\delta_{y_{(L*)}}}^2 + C_2)} \end{aligned} \tag{5.12}$$

Weighted computation L component in SSIM represents the luminance measurement, thus we can change the relative importance of the three components by setting higher value of α than β,γ in SSIM format shown above.

For instance, we can set $\alpha = 1$, and $\beta = \gamma = 0.1$, so

$$\begin{aligned} SSIM_{(L*a*b*)} &= L(x,y)^1 \cdot C(x,y)^{0.1} \cdot S(x,y)^{0.1} \\ MSSIM_{(L*a*b*)} &= \frac{1}{M}\sum_{j=1}^{M} L(x_{(L*)}, y_{(L*)})^1 \cdot C(x_{(L*)}, y_{(L*)})^{0.1} \cdot S(x_{(L*)}, y_{(L*)})^{0.1} \end{aligned} \tag{5.13}$$

5.3 Visual Information Fidelity (VIF)

VIF index introduces natural scene statistical model in conjunction with a distortion(channel) model to quantify the information shared between the reference and test images. This measurement does not rely on any Human Visual System (HVS) or viewing geometry parameter. It treats HVS as a communication channel and predicts the subjective image quality by computing how much the information within the perceived reference image is preserved in the distorted one.

5.3.1 Introduction

In VIF, an approach called "information-theoretic setting" is included. The reference image and the test image are in two different processes through the measurement.

The reference image is regarded as a natural image source that goes through HSV channel before being passed to the brain. The information comes from the reference image is quantized as being mutual information between input and output of the HSV channel. This kind of information is that the brain could ideally extract from the reference image.

The same measurement is also applied for the test image. There is a distortion channel that distorts the output of the natural image source before it comes to HSV channel, whereby we can get the information that the brain can extract from the test image.

Finally, we can combine those two results from each phase to get the VIF evaluation. The Figure below shows the entire measurement process, where VIF system model consists of three sub-models: source model, distortion model, and HVS model. [27]

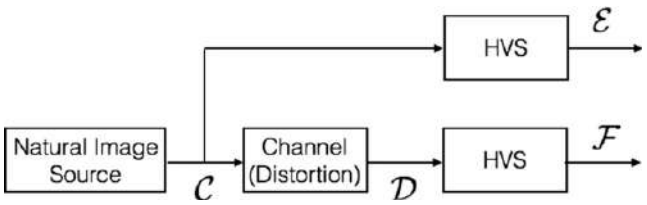


Figure 5.1: VIF measurement process

In Figure 4.1, mutual information between $\mathcal{C}$ and $\mathcal{E}$ quantifies the information that the brain could ideally extract from the reference image, whereas the mutual information between $\mathcal{C}$ and $\mathcal{F}$ quantifies the corresponding information that could be extracted from the test image.[27]

5.3.2 Application

Based on the definition of VIF, the same VIF assessment model can be applied for both RGB color space and L*a*b* color space.

5.4 Gradient Magnitude Similarity Deviation - (GMSD)

In practice, a good FR-IQA method should be not only effective but also efficient. Unfortunately, it is tough to achieve both aspects simultaneously, and those FR-IQAs mentioned before can only perform well in one of these two parts.

GMSD is a two-step framework (Figure 4.2) that tries to fill this. It computes a local quality map (LQM) by comparing the gradient magnitude maps of the test and reference images ,and then a single overall quality score is computed from the LQM via some pooling strategy. After that, GMSD uses standard deviation as the pooling strategy to compute the final quality score. The gradient feature of images can capture image local structures efficiently, to which the HVS is highly sensitive. [35]

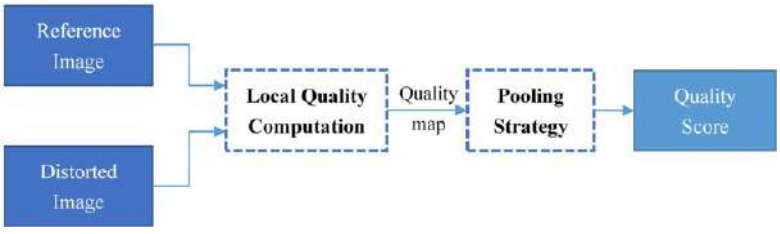


Figure 5.2: The flowchart of a class of the two-step framework

5.4.1 Introduction

5.4.1.1 Gradient Magnitude Similarity

In GMSD, Similarity function called gradient SSIM (G-SSIM) is derivate from SSIM. For digital images, the gradient magnitude is defined as the root mean square of image directional gradients along two orthogonal directions. The gradient is computed by convolving an image with a linear filter -Prewitt filter.

The horizontal (x) and vertical (y) directions of Prewitt filters are defined as:

$$h_x = \begin{bmatrix} \frac{1}{3} & 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{3} & 0 & -\frac{1}{3} \end{bmatrix} h_y = \begin{bmatrix} \frac{1}{3} & 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{3} & 0 & -\frac{1}{3} \end{bmatrix} \tag{5.14}$$

The horizontal and vertical gradient images of r and d are the results of convolving h_x and h_y with the reference and quantized images, and then we can compute the gradient magnitude of r and d at location i, denoted by $m_r(i)$ and $m_d(i)$ as follows:

$$\begin{aligned} m_r(i) &= \sqrt{r \otimes {h_x}^2(i) + r \otimes {h_y}^2(i)} \\ m_d(i) &= \sqrt{d \otimes {h_x}^2(i) + d \otimes {h_y}^2(i)} \end{aligned} \tag{5.15}$$

where $\otimes$ denotes the convolution operation.

Based on the gradient magnitude images m_r and m_d we get, the gradient magnitude similarity (GMS) map then can be computed as follows:

$$GMS(i) = \frac{2m_r(i)m_d(i) + c}{m_r(i)^2 + m_d(i)^2 + c} \tag{5.16}$$

where c is a positive constant to supply numerical stability.

The GMS map serves as the local quality map (LQM) of the quantized image

d. In the GMS map, the brighter the gray level, the higher the similarity, and thus the higher the predicted local quality.

5.4.1.2 Pooling with Standard Deviation

Average pooling is the most common used pooling strategy to compute the final quality score, so it is applied to the GMS map as Gradient Magnitude Similarity Mean (GMSM):

$$GMSM = \frac{1}{N}\sum_{i=1}^{N} GMS(i) \tag{5.17}$$

In estimating the overall image quality, the main issue about average pooling is that the importance of each pixel is same, which means that it ignores the fact that the global variation of image local quality degradation can reflect its overall quality.

In order to solve this problem, the standard deviation of GMS map is computed and taken as the final IQA index, namely Gradient Magnitude Similarity Deviation (GMSD):

$$GMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(GMS(i) - GMSM)^2} \tag{5.18}$$

With the value of GMSD, the range of distortion severities shows the reflection in an image.

5.4.2 Application

Based on the definition of GMSD, the identical GMSD model can be fit into both RGB and L*a*b* color spaces.

Chapter 6

Experimental Results

Overview

In this chapter, we will detail all the results of four IQA methods discussed in chapter 5 applied to quantized images of the reference image 4.1 we got from three color quantization approaches, and then we can have a direct comparison between IQA models and human-eye subjective evaluation. As for other ten original images, we will give summarized results.

6.1 Human-eye subjective evaluation

In chapter 4, We already made a roughly subjective evaluation of quantized images from those three color quantization algorithms and summarized those "key images" in Table 4.1.

Additionally, in order to have a better comparison between subjective and objective images quality assessments, from each color quantization approach, we (the author) as a observer also select those sequential quantized images which are visually different observed by our eyes. The results are shown in Table 6.1.

	k-means		median-cut			octree	
	RGB	L*a*b*	RGB	1st L*a*b*	2nd L*a*b*	RGB	L*a*b*
visually different pairs	(1st,2nd)	(1st,2nd)	(1st,2nd)	(1st,2nd)	(1st,2nd)	(1st,2nd)	(1st,2nd)
	(2nd,3rd)	(2nd,3rd)	(3rd,4th)	(2nd,3rd)	(2nd,3rd)	(3rd,4th)	(2nd,3rd)
	(3rd,4th)	(3rd,4th)	(5th,6th)	(4th,5th)	(3rd,4th)	(6th,7th)	(3rd,4th)
	(4th,5th)	(4th,5th)		(5th,6th)	(4th,5th)	(7th,8th)	(5th,6th)
	(5th,6th)	(7th,8th)		(8th,9th)	(5th,6th)	(14th,15th)	(7th,8th)
					(6th,7th)	(16th,17th)	(8th,9th)
					(9th,10th)		(10th,11th)

Table 6.1: subjective visual different image pairs of three quantization approaches

6.2 Objective image quality assessment results

In terms of the definitions of IQA methods, we know that we can evaluate the quality of quantized images by comparing it to the original image, so tables below show the resulting score of comparing the original image and quantized images for different quantization methods.

6.2.1 PSNR results

The higher the PSNR, the better quality of the quantized image.

6.2.1.1 k-means results

In Table 6.2, we know that the PSNR scores experience a dramatic increase from 11.867 to 22.617 in the color number range [1,6].

k-means			
number of colors	RGB	L*a*b*	L*
1	11.867	16.126	11.764
2	17.587	21.339	17.965
3	20.172	23.264	19.504
4	21.643	24.416	20.431
5	22.673	25.44	21.45
6	23.617	26.459	22.624
7	24.351	27.155	23.324
8	24.918	27.509	23.608
9	25.33	27.752	23.733
10	25.769	28.27	24.305
11	26.149	28.697	24.751
12	26.528	29.251	25.419
13	26.861	29.452	25.584
14	27.151	29.606	25.689
15	27.415	29.898	25.996
16	27.656	30.146	26.254
17	27.871	30.424	26.56
18	28.07	30.63	26.774
19	28.27	30.893	27.074
20	28.466	31.089	27.277

Table 6.2: PSNR score of k-means method

As for L*a*b* color space, PSNR also has a good performance: the trend of the results match all the pairs in table 6.1 except one pair –(7th,8th).In table 6.2, the scores of these two quantized images are 27.155 and 27.509 respectively, which can be regarded as very similar, but our human eyes think they are very different.

(a) 7 main colors (b) 8 main colors

Figure 6.1: 7th and 8th quantized images from k-means in L*a*b* color space

In Figure 6.1, we can focus on the color theme palettes at first. Honestly, these two palettes are very similar but with one difference the number of yellowish colors, one and two respectively. As we mentioned before, the frequency of yellowish colors in this image is very low, but this kind of colors plays an important visual role representing the bright objects. For our eyes, we can see the shape of the moon at the right image but not in the left one. Quantized images can also have a significant visual difference even though they have similar dominant colors. For PSNR, these two images have the roughly same similarity with the original image, but our human eyes can recognize the representative of those color shapes. The key images are the 6th and 7th in RGB and L*a*b* color spaces which both match the PSNR results (the resulting scores experience a massive increase before reaching these two images).

Besides that, the growing trend of those scores of the L* channel computation is entirely the same as that of L*a*b* color space. We can say that PSNR has a good match with our human eyes.

6.2.1.2 PSNR score of median-cut method

median-cut					
number of colors	RGB	1st L*a*b*	1st L*	2nd L*a*b*	2nd L*
1	11.135	16.124	11.763	16.124	11.763
2	15.487	16.843	12.27	16.843	12.27
3	15.67	18.816	14.348	18.816	14.348
4	19.423	18.911	14.395	20.328	15.954
5	19.669	20.463	16.024	24.6	20.899
6	20.064	21.076	16.626	24.895	21.005
7	20.192	22.55	18.232	25.044	21.023
8	20.234	23.404	19.189	26.376	22.637
9	20.643	23.993	19.852	26.441	22.674
10	21.045	24.043	19.852	26.535	22.678
11	21.74	24.188	19.98	26.759	22.956
12	21.884	24.652	20.512	26.88	23.106
13	21.912	24.715	20.554	26.918	23.155
14	21.985	25.107	21.008	28.013	24.51
15	22.123	25.235	21.156	28.052	24.561
16	24.638	25.412	21.166	28.181	24.573
17	24.695	25.44	21.194	28.199	24.579
18	24.725	25.48	21.196	28.372	24.804
19	24.763	25.545	21.233	28.85	25.446
20	24.952	25.549	21.236	28.96	25.597

Table 6.3: PSNR score of median-cut method

Generally speaking, PSNR scores of L*a*b* color space are higher than those of RGB color space and also more similar to the subjective evaluation results. Besides that, the results of both L* channel computation have same growing trends corresponding to their L*a*b* quantized images.

In terms of PSNR results of these two kinds of L*a*b* color spaces, with the

same number of dominant colors, it shows that more brightness in color palettes leads to higher scores.

As for the similarity with the key images from subjective evaluation results, in RGB color space, the resulting score still grows steadily, up to 24.638 of the 16th image. While in L*a*b* color space, the growth rate decreases dramatically, which illustrates that PSNR results of L*a*b* color space for median-cut have a better match with the subjective evaluation than those of RGB color space.

6.2.1.3 Octree results

octree			
number of colors	RGB	L*a*b*	L*
1	9.268	14.153	9.816
2	9.868	14.94	10.373
3	9.872	13.441	9.017
4	10.98	14.417	9.821
5	10.984	14.129	9.541
6	10.984	14.605	10.017
7	11.234	14.606	10.017
8	11.011	16.005	11.467
9	11.011	16.243	11.681
10	11.011	16.604	12.029
11	11.011	20.947	16.595
12	11.017	20.947	16.595
13	11.024	20.947	16.595
14	11.029	20.947	16.595
15	11.127	20.947	16.595
16	11.244	20.947	16.595
17	10.596	20.947	16.595
18	10.596	21.018	16.671
19	10.603	21.018	16.671
20	10.609	21.018	16.671

Table 6.4: PSNR score of octree method

The scores shown in Table 6.4 in L*a*b* color space match more different pairs shown in Table 6.1 than those in RGB color space do. The key images, the 7th and the 11th quantized images in two color spaces respectively, have very high scores among all results, so we can say that PSNR results have a high similarity with the subjective evaluation in both color spaces.

6.2.2 SSIM results

The higher score represents a higher similarity of the reference and test images. In the experiment, MSSIM assessment is applied to compare the reference image and test images after k-means quantization in RGB, L*a*b* color spaces, and L* channel as well.

6.2.2.1 k-means results

In terms of the resulting scores of SSIM shown in table 6.5, it is evident that the trend of SSIM scores in RGB color space has a better match to the subjective evaluation result than those in L*a*b* color space do.

In L*a*b* color space, the most significant difference appears in the first and second quantized images which matches human eyes' evaluation, but in L* channel computation, the difference is over three times as high as that in L*a*b* color space while both situations show a good match with the subjective results.

The scores increase slightly after the 6th image in RGB color space and the 8th image in L*a*b* color space, so SSIM has a high similarity with the individual key images.

Overall, SSIM performs well in RGB color space and L* channel.

k-means			
number of colors	RGB	L*a*b*	L* channel
1	0.185	0.696	0.31
2	0.472	0.774	0.564
3	0.654	0.796	0.6
4	0.74	0.824	0.65
5	0.77	0.851	0.706
6	0.791	0.873	0.761
7	0.828	0.884	0.785
8	0.857	0.891	0.794
9	0.872	0.895	0.794
10	0.873	0.905	0.821
11	0.874	0.912	0.836
12	0.889	0.919	0.855
13	0.894	0.923	0.86
14	0.897	0.926	0.862
15	0.905	0.929	0.87
16	0.909	0.931	0.872
17	0.916	0.934	0.879
18	0.919	0.936	0.883
19	0.917	0.939	0.89
20	0.925	0.941	0.894

Table 6.5: SSIM score of k-means method

6.2.2.2 median-cut results

median-cut					
number of colors	RGB	1st L*a*b*	1st L*	2nd L*a*b*	2nd L*
1	0.166	0.696	0.357	0.696	0.357
2	0.499	0.7	0.326	0.7	0.326
3	0.528	0.743	0.455	0.743	0.455
4	0.621	0.744	0.45	0.8	0.582
5	0.622	0.803	0.58	0.844	0.684
6	0.622	0.809	0.58	0.849	0.685
7	0.623	0.829	0.633	0.853	0.686
8	0.623	0.843	0.671	0.872	0.744
9	0.653	0.845	0.673	0.874	0.746
10	0.661	0.846	0.673	0.875	0.746
11	0.707	0.849	0.674	0.88	0.76
12	0.711	0.854	0.688	0.882	0.766
13	0.711	0.856	0.689	0.883	0.768
14	0.715	0.861	0.7	0.9	0.82
15	0.725	0.862	0.704	0.901	0.822
16	0.832	0.87	0.704	0.903	0.822
17	0.834	0.871	0.706	0.903	0.822
18	0.834	0.872	0.706	0.905	0.827
19	0.834	0.873	0.705	0.911	0.844
20	0.837	0.873	0.705	0.912	0.848

Table 6.6: SSIM score of median-cut method

As we know that, from the median-cut algorithm, there are three pairs of quantized images regarded as visually different in RGB color space, (1st,2nd), (3rd,4th) and (5th,6th) respectively.

When we look at the Table 6.6, the resulting scores of images from 1 to 4 match the first two subjective pairs we have, but the 5th and 6th images have

the same SSIM scores.

(a) 5 main colors (b) 6 main colors

Figure 6.2: 5th and 6th quantized images from k-means in RGB color space

There is only one different color in the palettes of these two quantized images – yellowish color of the moon, which can easily attract human eyes, so we think that the 6th image is much similar to the original image.

For other two kinds of images from L*a*b* color space, the difference of SSIM results from L* channel computation are much similar to those pairs of the subjective evaluation than from entire L*a*b* color space.

Although the RGB results show higher similarity with the subjective evaluation, SSIM scores in L*a*b* color space are higher than those in RGB color space, especially at the beginning.

6.2.2.3 Octree results

We get surprising results of using SSIM for the octree method. In all three conditions, the first quantized images with only one color have higher scores than the second ones with two dominant colors.

L* channel computation has the highest similarity with the subjective evaluation results, while SSIM scores of L*a*b* color space are much higher than those

of RGB color space.

octree			
number of colors	RGB	L*a*b*	L*
1	0.227	0.7	0.375
2	0.113	0.657	0.272
3	0.115	0.698	0.368
4	0.209	0.651	0.256
5	0.21	0.63	0.218
6	0.21	0.681	0.318
7	0.214	0.681	0.319
8	0.213	0.692	0.344
9	0.213	0.698	0.351
10	0.213	0.704	0.338
11	0.213	0.78	0.522
12	0.213	0.78	0.522
13	0.213	0.78	0.522
14	0.212	0.78	0.522
15	0.212	0.78	0.522
16	0.215	0.78	0.522
17	0.313	0.78	0.522
18	0.313	0.782	0.526
19	0.313	0.782	0.526
20	0.313	0.782	0.526

Table 6.7: SSIM score of octree method

6.2.3 VIF Results

The higher the VIF score, the better the quality of the quantized image. In Table 6.8, it illustrates that k-means has the highest score among all three algorithms, and VIF scores of L*a*b* are much higher than those of RGB color space and are more similar to the subjective evaluation.

	k-means		median-cut			octree	
	RGB	L*a*b*	RGB	1st L*a*b*	2nd L*a*b*	RGB	L*a*b*
1	0.058	0.026	0.057	0.029	0.029	0.053	0.380
2	0.110	0.576	0.114	0.136	0.136	0.021	0.256
3	0.194	0.696	0.127	0.384	0.384	0.022	0.512
4	0.246	0.744	0.179	0.387	0.542	0.040	0.329
5	0.276	0.790	0.187	0.544	0.784	0.041	0.346
6	0.302	0.832	0.193	0.577	0.791	0.041	0.361
7	0.331	0.854	0.196	0.681	0.795	0.046	0.362
8	0.356	0.863	0.197	0.734	0.842	0.044	0.401
9	0.374	0.870	0.219	0.755	0.845	0.044	0.418
10	0.385	0.884	0.234	0.757	0.848	0.044	0.345
11	0.396	0.894	0.274	0.762	0.855	0.044	0.495
12	0.413	0.906	0.281	0.783	0.856	0.044	0.495
13	0.423	0.911	0.282	0.784	0.858	0.044	0.495
14	0.433	0.913	0.286	0.801	0.886	0.044	0.495
15	0.444	0.919	0.295	0.806	0.887	0.045	0.495
16	0.452	0.922	0.341	0.811	0.889	0.046	0.495
17	0.462	0.926	0.342	0.812	0.889	0.104	0.495
18	0.470	0.930	0.343	0.813	0.893	0.104	0.502
19	0.472	0.934	0.344	0.817	0.905	0.104	0.502
20	0.481	0.936	0.351	0.817	0.907	0.105	0.502

Table 6.8: VIF results of three quantization approaches

We can know that the score difference becomes smaller with the number of colors in the quantized images increasing.

There is an essential feature of VIF assessment that distinct from other IQAs: a linear contrast enhancement of the reference image that does not add noise will result in a VIF value more significant than unity, thereby signifying that the

enhanced image has a superior visual quality than the reference image. It is common that contrast enhancement of images has a better perceptual performance after color quantization.

In theory, contrast enhancement results in a higher signal-to-noise ratio at the output of the human visual system neurons, therefore enabling the brain to have a greater ability to discriminate objects present in the visual signal. VIF can capture this kind of improvement.[27]

As mentioned in chapter 3, fitting for human vision is one of the most important purposes of the conversion from RGB to L*a*b* color space, so we can notice that the bright objects in L*a*b* quantized images become brighter than those in RGB color space by our observation. This kind of brightness enhancement can be viewed as a subset of contrast enhancement, which probably can explain why VIF performs much better in L*a*b* color space than RGB color space.

6.2.4 GMSD Results

As for GMSD, the lower the score, the better the quality of the quantized image.

In terms of results we have, in Table 6.9, we know that scores of k-means and median-cut are much higher than those of octree, overall. Additionally, in all three quantization methods, GMSD scores are also much lower in L*a*b* color space than RGB color space, which means that L*a*b* color space has better performance than RGB color space does.

Speaking of the comparison with the subjective evaluation in RGB color space, there are only a few similar matches in all quantization methods, but much more image pairs match the score differences in L*a*b* color space, especially those from the median-cut algorithm.

	k-means		median-cut			octree	
	RGB	L*a*b*	RGB	1st L*a*b*	2nd L*a*b*	RGB	L*a*b*
1	0.217	0.229	0.214	0.218	0.218	0.298	0.233
2	0.176	0.127	0.210	0.186	0.186	0.257	0.183
3	0.168	0.128	0.197	0.239	0.239	0.254	0.252
4	0.151	0.095	0.164	0.231	0.207	0.224	0.186
5	0.127	0.096	0.149	0.188	0.090	0.222	0.171
6	0.122	0.081	0.146	0.159	0.085	0.222	0.170
7	0.105	0.062	0.144	0.108	0.083	0.211	0.169
8	0.093	0.060	0.145	0.095	0.066	0.218	0.163
9	0.080	0.055	0.142	0.093	0.065	0.218	0.160
10	0.079	0.048	0.141	0.092	0.064	0.218	0.157
11	0.073	0.042	0.136	0.089	0.062	0.218	0.146
12	0.063	0.038	0.134	0.083	0.061	0.217	0.146
13	0.060	0.037	0.134	0.083	0.061	0.216	0.146
14	0.058	0.036	0.133	0.078	0.054	0.216	0.146
15	0.055	0.034	0.132	0.077	0.053	0.212	0.146
16	0.053	0.032	0.093	0.077	0.052	0.211	0.146
17	0.051	0.030	0.093	0.077	0.051	0.205	0.146
18	0.050	0.029	0.093	0.077	0.048	0.205	0.145
19	0.050	0.028	0.093	0.074	0.043	0.204	0.145
20	0.046	0.027	0.092	0.074	0.042	0.202	0.145

Table 6.9: GMSD results of three quantization approaches

In general, there is a decreasing trend of the score difference in both k-means and median-cut algorithms, while the pattern of octree values fluctuate in RGB color space but decline in L*a*b*b color space.

6.3 Summarized Results of 10 Original Images

Based on the image quality assessment process of Figure 4.1, we also implemented the same process to those ten original images shown in Figure 4.2.

6.3.1 Color Space Comparison

Table 6.10 summarizes all experimental results we obtained for those ten reference images in Figure 4.2. We compare all four IQA results for each color quantization algorithms in both RGB and L*a*b* color spaces. The number in the table shows the number of original images whose quantized images match that "comparison" resutls.

The column "comparison" represents three different comparison results of RGB and L*a*b* color spaces. To be more specific, if result scores from over two IQAs of an original image are higher in L*a*b* color space, we will count one in the third row (RGB < L*a*b*); if scores from only one IQA are higher in L*a*b* color space, we will count one in the first row (RGB > L*a*b*); other situations are counted in the second row (RGB = L*a*b*).

color space comparison by 4 IQA results			
comparison	k-means	median-cut	octree
RGB > L*a*b*	-	1	-
RGB = L*a*b*	-	2	1
RGB < L*a*b*	10	7	9

Table 6.10: 4 IQA results of Figure 4.2

As we can see, for the k-means algorithms, quantized images in L*a*b* color space have higher fidelity with all ten original images. And for median-cut and octree methods, most quantized images perform better in L*a*b* color space.

6.3.2 L* Channel Extension in Median-cut

Same as Figure 4.1, we also implemented those two different kinds of median-cut algorithms mentioned above to Figure 4.2. We manually added the range of the L* channel of the original images to explore if this operation can increase the visual fidelity of quantized images.

Table 6.11 shows the summarized results.We use "first Lab" to represent the score results of the original median-cut method and "second Lab" to represent the score results of the L* channel extension median-cut method.

comparison of L* channel extension in median-cut	
comparison	number of images
"first Lab" > "second Lab"	2
"first Lab" = "second Lab"	1
"first Lab" < "second Lab"	7

Table 6.11: comparison of L* channel extension in median-cut of Figure 4.2

There are seven original images that have better visual fidelity after increasing the range of L* channel, so we can say that more lightness in the L*a*b* quantized images can have better visual fidelity with the original images.

6.3.3 L* Channel Compuation in PSNR and SSIM

As for PSNR and SSIM image quality assessments, we also applied L* channel compuation for each assessment. The results show that the score trends of L* channel are almost identical with those of L*a*b* color space, so we can say that L* channel computation can be viewed as a good representative of L*a*b* color space compuatation.

6.4 Conclusion

By observing quantized images with subjective human eyes and four objective IQA results, we understand that:

1. Quantitative images obtained through k-means have the highest scores in all four objective IQAs, and the visual effect of human eyes is also excellent.

2. Among the four kinds of objective IQAs, the scores of L*a*b* and RGB color space are different, and the scores of L*a*b* are higher than RGB in general, which shows that the conversion of color space can indeed improve the quality of images. On the other hand, the four kinds of objective IQAs also follow the concept of color space conversion.

Chapter 7

Testing

Overview

In this chapter, we will discuss our test results for both functionality and non-functionality. Our porject code is available at `https://gitlab.cas.mcmaster.ca/G-ScalE/Lai_Project`

7.1 Functional Test

7.1.1 Quantitative Data Test

We choose quantitative data test to verify our conclusions we obtained through the color quantization experiment.

7.1.1.1 Test Preparation and Test Process

Preparation We randomly select 50 publicly accessible images in RGB color space as test input to implement this experiment.

Process We follow the program operating instructions shown in appendix A to run this test.

7.1.1.2 Test Results

We summarize our test results in four aspects shown below:

Color Space Comparison In Table 7.1, we compare all four IQA results for each color quantization algorithms in both RGB and L*a*b* color spaces. We use the same kind of Table 6.10 in chapter 5 to show our test results. The number in the table shows the number of test images whose quantized images match that "comparison" resutls.

color space comparison by 4 IQA results			
comparison	k-means	median-cut	octree
RGB > L*a*b*	-	3	1
RGB = L*a*b*	-	8	1
RGB < L*a*b*	50	39	48

Table 7.1: 4 IQA results of 50 test images

Based on Table 7.1, the number of test images whose quantized images have higher fidelity with the original image in L*a*b* color space are 50, 39, and 48 respectively. This test results verifies our conclusion: color space conversion (from RGB to L*a*b* color space) can improve the visual fidelity of quantized images.

Best Performance Algorithm Through our experimental results, the k-means algorithm performs best among all three quantization algorithms. Quantized images of all 50 test images from the k-means algorithm have best scores of all four IQAs.

L* Channel Extension in Median-cut In order to justify our conclusion: L* channel extension in median-cut can increase the visual fidelity of quantized images, we implement this operation to 50 test images.

Table 7.2 are the similar with Table 6.11 where "first Lab" represents the score results of the original median-cut method and "second Lab" represents the score results of the L* channel extension median-cut method.

comparison of L* channel extension in median-cut	
comparison	number of images
"first Lab" > "second Lab"	-
"first Lab" = "second Lab"	2
"first Lab" < "second Lab"	48

Table 7.2: comparison of L* channel extension in median-cut of 50 test images

There are 48 test images that have better visual fidelity after increasing the range of L* channel, and only 2 images did not experience the visual fidelity increasing. Thus, L* channel extension in the median-cut algorithm can help quantized images L*a*b* color space improve visual fidelity with the original images.

L* Channel Compuation in PSNR and SSIM To verify the conclusion: L* channel computation can be viewed as a good representative of L*a*b* color space compuatation, by PSNR and SSIM, we compute L* channel of quantized images of 50 test images in L* channel and compare the results with L*a*b* computation. The comparative results show that the change of scores of L* channel and L*a*b* color space are synchronous, so our conclusion are verified.

7.1.2 Qualitative Data Test

The variety of input (the reference image) can affect the performance of output (quantized images), especially the frequency and the distribution of color information.

7.1.2.1 Test Preparation and Test Process

Preparation We tested four different reference images with different features of the color frequency and distribution.

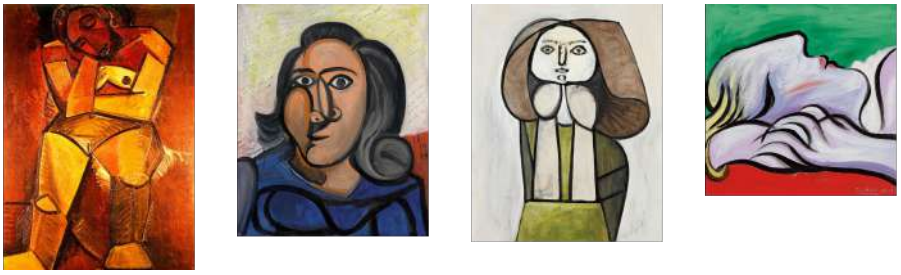


Figure 7.1: Four Reference Images

As Figure 7.1 shows, we chose four famous paintings from Pablo Picasso as our reference images. They are Seated woman, Buste de femme (Dora Maar), Femme la robe verte and Lerepos from left to right. The frequency and distribution of color information in all four images shown in Table 7.3 are different and typical.

Color Information		
Name	Frequency	Distribution
Seated woman	even	even
Buste de femme (Dora Maar)	even	not even
Femme la robe verte	not even	even
Lerepos	not even	not even

Table 7.3: Color Information for Four Reference Images

Addtionally, overall features of each reference image are summarized below:

1. Seated woman: most are warm colors;
2. Buste de femme (Dora Maar): large difference in chromaticity;
3. Femme la robe verte: most are brownish and greenish colors, and black color is low-frequency,but widely-distributed;
4. Lerepos: white, green, and red colors are dominant, and black color is low-frequency,but widely-distributed

Process We follow the program operating instructions shown in appendix A to run this test for four times.

7.1.2.2 Test Results

Three figures below show quantized images with five theme colors from all three color quantization algorithms,k-means, median-cut, and octree respectively.

Figure 7.2: Four Quantized Images from K-means

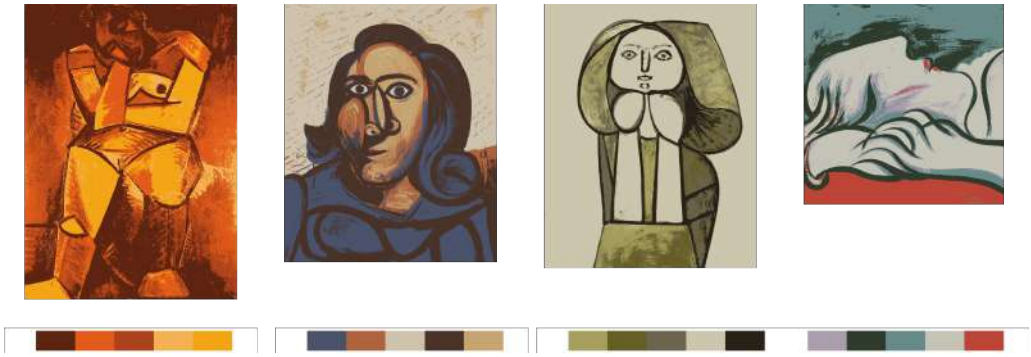


Figure 7.3: Four Quantized Images from Median-cut

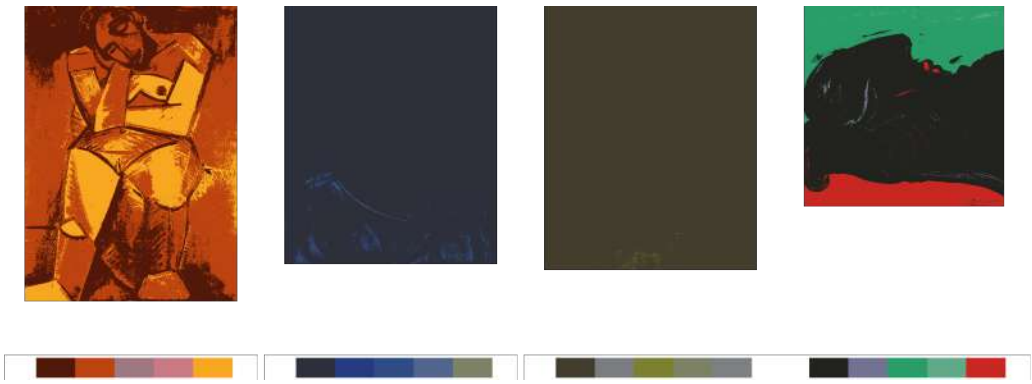


Figure 7.4: Four Quantized Images from Octree

Based on the definition of k-means, we know that the first step of this algorithm is to randomly select a number of centroids that is equal to the number of colors we want the palette to be composed of, so the frequency and the distribution of the reference image have little influence on the performance of quantized images.

As for the median-cut algorithm, the first step of each loop is to find the median value in the longest range of three dimensions in both RGB and L*a*b* color spaces. In the X dimension, the longest range equals to the difference between maximum and minimum values, which means that median-cut gives priority to the dimension where there is the most widely distributed color information.

In general, the median-cut algorithm focuses more on the distribution of color information than the frequency of them.

As we discussed in chapter 3, the octree algorithm only pays attention to the distribution of color information, not the frequency of them. So except the first quantized image, other images have poor fidelity with the original images.

Overall, k-means performs best based on all four situations, while quantized images from median-cut also have high fidelity with the original images. Octree has the worst performance among all three algorithms. In terms of this test, we should know that targeting to different algorithms, the selection of images play an essential role. In other words, reference images are supposed to be categorized based on the color frequency and distribution before color quantization.

7.2 Non-functional Test

7.2.1 Performance Test

We test the runtime of the process of color quantization and palette visualization for all three algorithms based on RGB and L*a*b* color spaces to evaluate the performance of our program.

7.2.1.1 Test Preparation and Test Process

Preparation We add code that can count runtime of the program in each color quantization algorithm.

Process We rerun our program to get runtime of Figure 4.1 and Figure 4.2 – 11 original images in total.

7.2.1.2 Test Results

Table 7.4 shows the experimental results of Figure 4.1 and Figure 4.2.

	k-means		median-cut			octree	
	RGB	L*a*b*	RGB	1st L*a*b*	2nd L*a*b*	RGB	L*a*b*
runtime (min:sec)	04:50	05:13	01:20	04:58	04:49	06:39	02:57
	08:29	02:30	01:46	09:16	05:56	10:20	05:37
	03:52	04:04	01:01	01:00	05:40	03:22	05:21
	00:14	00:30	00:08	02:53	00:26	00:23	00:29
	00:23	00:56	00:08	00:28	00:26	00:22	00:30
	03:41	03:44	00:46	02:53	03:00	02:35	03:00
	10:39	06:19	01:38	09:56	10:13	05:31	05:13
	04:51	05:25	00:43	04:59	04:15	02:20	01:40
	15:34	14:31	01:52	09:55	10:05	05:20	05:07
	01:02	01:39	00:30	01:49	01:50	01:34	00:55
	01:34	01:30	00:22	00:54	01:36	01:07	01:03

Table 7.4: Runtime of Four Color Quantization Algorithms

As Table 7.4 shows above, we can see that the median-cut algorithm completed the run in the shortest amount of time in RGB color space, while in L*a*b* color space the octree algorithm ran fastest. In both color spaces, the runtime of the k-means is relatively long.

Chapter 8

Conclusions

8.1 Project Conclusions

Our project accomplish our project purpose, and we conclude all experimental results shown below:

1. K-means algorithm enables the target image to have the highest fidelity after color quantization.

2. The specified image can have a better visual fidelity by the color space conversion from RGB to L*a*b* color space.

3. We explored the effect the brightness of the image in the visual fidelity and found it plays an essential role for this aspect.

4. The number of thematic colors in key images shown in Table 4.1 obtained by different quantization algorithms is different, among which the number of k-means approach is the least.

5. The colors of the palette with the same number of dominant colors obtained by different quantization algorithms are also different.

8.2 Main Innovative Features

The main innovative features of the project are :

1. In the aspect of color quantization, the order of bright color appearing in the image is advanced artificially in the median-cut algorithm. Also, the visual effect of the image has been improved significantly in both subjective and objective evaluation methods.

2. L* channel computation is added in the PSNR and SSIM assessment methods. The experimental results also show that this kind of computation and the entire L*a*b* computations show high synchronization.

Through these two features, we can conclude that brightness has an essential impact on the visual fidelity of pictures.

8.3 Future Work

8.3.1 Object Correspondence

The four OIQA methods used in this project are also always close to the human vision, from imitating the absorption of light by human eyes, the whole process of perceiving colors, to adding the information channel to simulate the brain to collect image information. However, the actual meaning of color and shape (e.g., the yellow semicircle is the moon) cannot be well reflected in the current mainstream objective IQAs.

If objects in the images can form a one-to-one correspondence with objects in real life in color quantization and objective IQAs, can the visual effect of quantized images have a higher degree of matching with human vision?

8.3.2 Choice of Color Quantization Algorithms

The color distribution of different pictures must be different. Different color quantization methods also have emphasis. Which kind of color quantization algorithms can achieve the best visual effect of the original image? It is worth exploring in future research.

8.3.3 Color Palette Application

In this project, we use a famous painting as an example to quantify the color. The resulting palettes are considered to be excellent color matching in line with the theory of color harmony. if we can extract the color palette from an image, can we change the color palette and reapply it to the image? Can we apply these palettes to daily clothing matching, interior design and so on? The objective and precise theme color palettes, together with the artistic attainments of the designers and artists themselves, can make the past works of art show greater artistic and commercial value.

Appendix A

Operating Instructions of the Project

Preparation

1. Operating environment: python 2.7 IDE
2. Python library installation: numpy, pandas, PIL, skimage, matplotlib, sklearn.cluster, scipy.misc

Operating Steps

1. Color quantization and visualization of quantized images and theme color palettes:
    - Open *color_quantization* folder
    - Add reference images in *img* folder
    - Three color quantization methods are in three different folders (*k-means, median-cut, octree*); Choose one folder and continue
    - Open *rgb_main.py* if color quantization in RGB color space; open *lab_main.py* if color quantization in L*a*b* color space

- Insert the relative path of the reference image, the range of numbers of theme colors, and save paths of quantized images and theme color palettes
- Run the program

2. Four objective image quality assessments

   - Open *img_quality_assessment (IQA) / allIQAs* folder under each quantization folder
   - Open *rgb_allIQAs / rgb_test_main.py* if color quantization in RGB color space; open *lab_allIQAs / lab_test_main.py* if color quantization in L*a*b* color space
   - Insert the relative path of the reference image and save paths of the csv file of assessment results
   - Run the program

References

[1] A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (Apr. 2004), pp. 600–612. ISSN: 1057-7149. DOI: 10.1109/TIP.2003.819861.

[2] Kenneth E. Burchett. "Color harmony". In: *Color Research & Application* 27.1 (2002), pp. 28–31. DOI: 10.1002/col.10004. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/col.10004. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/col.10004.

[3] Lou Marvin Caraig. *color quantization using k-means*. URL: https://lmcaraig.com/color-quantization-using-k-means/.

[4] M. E. Celebi. "Effective initialization of k-means for color quantization". In: (Nov. 2009), pp. 1649–1652. ISSN: 1522-4880. DOI: 10.1109/ICIP.2009.5413743.

[5] M. E. Celebi and Q. Wen. "VARIANCE-CUT: A fast color quantization method based on hierarchical clustering". In: (Nov. 2013), pp. 103–106. DOI: 10.1109/ICECCO.2013.6718239.

[6] *Chromaticity Coordinate*. URL: https://www.sciencedirect.com/topics/engineering/chromaticity-coordinate.

[7] M. Frackiewicz and H. Palus. "Outlier-based initialisation of K-means in colour image quantisation". In: (Sept. 2013), pp. 36–41. DOI: 10.1109/ICoIA.2013.6650226.

[8] Paul Heckbert. "Color Image Quantization for Frame Buffer Display". In: *SIGGRAPH Comput. Graph.* 16.3 (July 1982), pp. 297–307. ISSN: 0097-8930. DOI: 10.1145/965145.801294. URL: http://doi.acm.org/10.1145/965145.801294.

[9] Xinrong Hu, Tianzhen Wang, and Dehua Li. "A new approach of color quantization based on ant colony clustering algorithm". In: 1 (Apr. 2005), 102–108 Vol. 1. DOI: 10.1109/ITCC.2005.18.

[10] R. W. Hunt. *Measuring Colour*. England: Fountain Press, 1998.

[11] Q. Huynh-Thu and M. Ghanbari. "Scope of validity of PSNR in image/video quality assessment". In: *Electronics Letters* 44.13 (June 2008), pp. 800–801. ISSN: 0013-5194. DOI: 10.1049/el:20080522.

[12] *IEC 61966-2-1:1999*. URL: https://webstore.iec.ch/publication/6169.

[13] "Image technology colour management fffdfffdfffd Architecture, profile format, and data structure". In: (Oct. 2004).

[14] Brian W. Keelan. *Handbook of image quality : characterization and prediction*. 2002.

[15] Lara-Alvarez et al. "A geometric approach to harmonic color palette design". In: *Color Research & Application* 44.1 (2019), pp. 106–114. DOI: 10.1002/col.22292. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/col.22292. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/col.22292.

[16] Hsien-Che Lee. *18.7: Theoretical color gamut*. 2005, p. 468.

[17] Tsann-Shyong Liu and Long-Wen Chang. "Greedy tree growing for color image quantization". In: v (Apr. 1994), V/97–V100 vol.5. ISSN: 1520-6149. DOI: 10.1109/ICASSP.1994.389554.

[18] S. Mitra et al. "Efficient color image compression using integrated fuzzy neural networks for vector quantization". In: 1 (Oct. 1997), 184–188 vol.1. ISSN: 1062-922X. DOI: 10.1109/ICSMC.1997.625746.

[19] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. "Color Compatibility from Large Datasets". In: *ACM Trans. Graph.* 30.4 (July 2011), 63:1–63:12. ISSN: 0730-0301. DOI: 10.1145/2010324.1964958. URL: http://doi.acm.org/10.1145/2010324.1964958.

[20] J. Puzicha et al. "On spatial quantization of color images". In: *IEEE Transactions on Image Processing* 9.4 (Apr. 2000), pp. 666–682. ISSN: 1057-7149. DOI: 10.1109/83.841942.

[21] J. Puzicha et al. "On spatial quantization of color images". In: *IEEE Transactions on Image Processing* 9.4 (Apr. 2000), pp. 666–682. ISSN: 1057-7149. DOI: 10.1109/83.841942.

[22] *RGB versus CMYK*. URL: http://www.printernational.org/rgb-versus-cmyk.php.

[23] V. Sathya, P. Niraimathy, and K. Bhoopathy Bagan. "Quantization of color image using generic roughness measure". In: (Mar. 2015), pp. 1–5. DOI: 10.1109/ICSCN.2015.7219927.

[24] Steven Segenchuk. *An Overview of Color Quantization Techniques*. URL: https://web.cs.wpi.edu/~matt/courses/cs563/talks/color_quant/CQindex.html.

[25] B. Shah, P. Dhatric, and V. Raghavan. "Using inverse image frequency for perception-based color image quantization". In: (Mar. 2004), pp. 71–75. DOI: 10.1109/IAI.2004.1300947.

[26] B. Shah, P. Dhatric, and V. Raghavan. "Using inverse image frequency for perception-based color image quantization". In: (Mar. 2004), pp. 71–75. DOI: 10.1109/IAI.2004.1300947.

[27] H. R. Sheikh and A. C. Bovik. "Image information and visual quality". In: *IEEE Transactions on Image Processing* 15.2 (Feb. 2006), pp. 430–444. ISSN: 1057-7149. DOI: 10.1109/TIP.2005.859378.

[28] Priyabrata Sinha. Nov. 2008, p. 9.

[29] T Smith and J Guild. "The C.I.E. colorimetric standards and their use". In: *Transactions of the Optical Society* 33.3 (Jan. 1931), pp. 73–134. DOI: 10.1088/1475-4878/33/3/301. URL: https://doi.org/10.1088%2F1475-4878%2F33%2F3%2F301.

[30] Sabine Sfffdfffdsstrunk, Robert Buckley, and Steve Swen. "Standard RGB Color Spaces". In: (1999), pp. 127–134.

[31] TwinklingStar. *Octree quantization*. URL: http://www.twinklingstar.cn/2013/491/octree-quantization/.

[32] X. Wang, Y. Song, and Y. Zhang. "A spatial FCM color quantization algorithm with pyramid data structure". In: (Sept. 2011), pp. 1–4. DOI: 10.1109/ICSPCC.2011.6061696.

[33] Hannah Weller. *Color Spaces*. URL: https://cran.r-project.org/web/packages/colordistance/vignettes/color-spaces.html.

[34] Yingqian Wu, Cuiping Yang, and Tianzhen Wang. "A new approach of color quantization of image based on neural network". In: 2 (July 2001), 973–977 vol.2. ISSN: 1098-7576. DOI: 10.1109/IJCNN.2001.939492.

[35] W. Xue et al. "Gradient Magnitude Similarity Deviation: A Highly Efficient Perceptual Image Quality Index". In: *IEEE Transactions on Image Processing* 23.2 (Feb. 2014), pp. 684–695. ISSN: 1057-7149. DOI: 10.1109/TIP.2013.2293423.

[36] M. P. Yu and K. C. Lo. "Contextual color quantization algorithm". In: (Sept. 2001), pp. 596–601. DOI: 10.1109/ICIAP.2001.957075.