Color Palette Swapping using K-Means and Basic Image Processing

Jaya Mangalo Soegeng Rahardjo - 13520015
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: jayamangalo1@gmail.com

***Abstract*—Colors are important elements of a design and color palettes represent the combination of colors that can make or break a design. The importance of experimenting with numerous color palettes can be done using a K-Means approach. An image can have its color palette extracted using K-Means and using basic image processing, have its color transformed or swapped into a new color, creating a new alternate version of a design. The explored implementation of K-Means explores manages to swap palettes of simpler designs, but have difficulties for much more complex designs.**

***Keywords*—color theory; color palette; palette swapping; image processing; k-means;**

I. Introduction

In various design fields, whether its UI design, logo design, animated character design, video game design, colors are often critical to said design. Colors are a crucial aspect in visual design as they convey hidden meaning and emotions as well as shaping how individuals perceive a design. The role of colors extends beyond aesthetics; it influences user experiences, evokes specific responses, and contributes to the overall visual language of the design.

By extension, color palettes are integral in the design process. Selected colors not only is important to the aesthetic of a design, but also carries inherent meaning through symbolisms as well as interact with each other in harmony or disharmony. Therefore, experimenting with alternative color palettes and combinations can improve a design significantly.

This paper explores an approach using K-Means clustering as well as image processing to enable designer in their creative process of experimenting with swapping color palettes of an already created design to create alternative designs with different color palettes.

II. Theoretical Foundation

A. Color Theory

Color Theory is a collection of multiple definitions, concepts, and applications with the purpose of creating a logical structure and understanding colors as well the relations of a color with another. In 1666, Sir Isaac Newton developed the first color wheel, a structural representation of colors which define their relationship with one another.

Colors are separated into three categories: 1) Primary Colors, consisting of red, blue, and yellow. 2) Secondary colors, consisting of colors formed by mixing two primary colors, green, orange, and purple. 3) Tertiary colors, formed by mixing a secondary color and a primary color. In the color wheel, colors are separated so that primary colors are located opposite of their complementary colors.

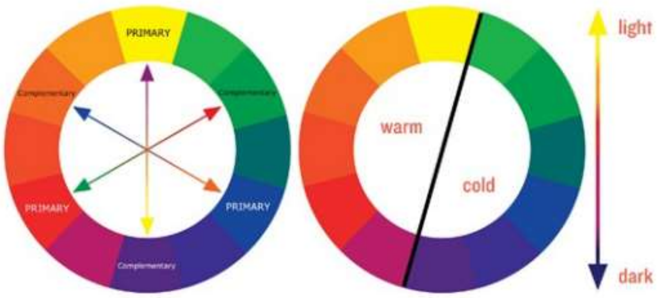

Figure 1. Color Wheels

Another way to look at the color wheel is to split the wheel by temperature into a cool half, consisting of cold colors such as blue, purple, and green as well as a warm half, consisting of colors such as orange, red, and yellow.

A use of the color wheel is to explain color harmony, a few concepts include 1) Analogous color scheme, where similar colors that three colors which are located side by side with each other are used. 2) Complementary color scheme, where colors opposite of each other are used together to provide a sharp contrast.

Another use of the color wheel named color context, it is a study of how colors are perceived when used next in different combinations. An example is the warm orange is more noticeable when used with the opposite color cold blue rather than when used together with closely located colors such as warm red.

B. Color Symbolism and Psychology

In design, Colors provide meaning and communicate and influence emotions in a primal and subconscious level. Hence, color symbolism is the study of the different effects of different colors have to the human psyche.

Color conveys meaning through two methods – natural associations and psychological symbolism. Natural associations refer to the association of colors with objects of nature, such as green for life and vegetation, blue for the sky and water. Psychological associations, also called cultural association comes from the mind and cultural influences. Cultural influences vary with each culture, such as green is associated as heaven (Muslims) and luck (USA and Ireland).

Example of colors and its meanings include – 1) Red for passion, love, anger. 2) Blue depicting calm, responsibility, sadness. 3) Black meaning mystery, elegance, evil. These are just examples of three colors and other colors have their own meaning as well. When designing, it is important to understand these symbols and what other people might associate with it.

C. *Color Palette*

Color Palettes are used in design to limit the number and range of colors that may be used in any design. The limitation of colors will ground the design and create a foundation for a brand, character, or design to maintain consistency. An example is McDonald's, which logo is widely associated with the distinctive colors of orange and yellow. Or the superhero character spiderman, which uses the palette of mainly red and blue supported by white and black.

When selecting a color palette to use, it is important to understand what a color symbolizes, as well as its relationship or harmony with each other. Will the color blue enhance the color red, will it take attention away, and so forth.

D. *Image Processing*

Image Processing is study of the processing of an image into a new image with a certain purpose, such as improving the image quality or the transformation of an image into a new image.

Image Processing has many methods and techniques each used for a different objectives. Such as using removing blur, removing noise, increasing contrast all to improve an image quality, or segmenting an image to extract valuable information to be used for further processing or gathering insights.

E. *Image Segmentation*

Image segmentation is a part of image processing which focuses on partitioning an image into many related regions. The purpose of image segmentation is to create regions which can separate objects with the background as well as objects with other objects.

Image segmentation has many real world applications such as medical imaging, autonomous vehicles, object recognition and scene understanding. Object segmentation also has many approaches typically categorized as discontinuity, partitioning objects based on differences such as edges, and similarity, partitioning objects based on similarities of a property.

F. *K-Means Clustering*

K-Means is a clustering algorithm which separates data and create groups of data called clusters by minimizing a criteria named within-cluster sum-of squares, the distance of each data in a cluster to its center (centroid).

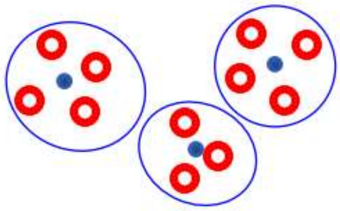

Figure 2. Example of K-Means Clustering

K-Means can be used as an approach to image segmentation categorized as similarity. K-Means will try to cluster an image by its cluster to find and object and separate it from its background.

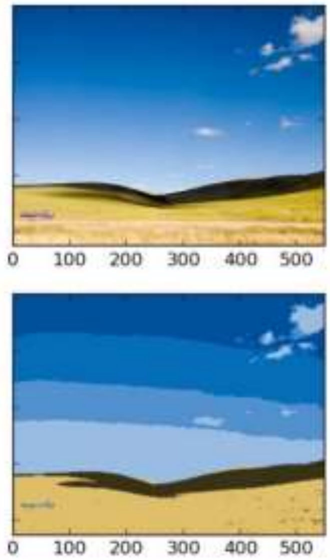

Figure 3. Image Segmentation using K-Means

K-Means does have some weakness however, K-Means is sensitive to the number of clusters used, anisotropically distributed data, unequal variance, as well as unevenly sized data. When facing such issues or images with such characterstics, other clustering methods or other image segmentation methods might be preferred.

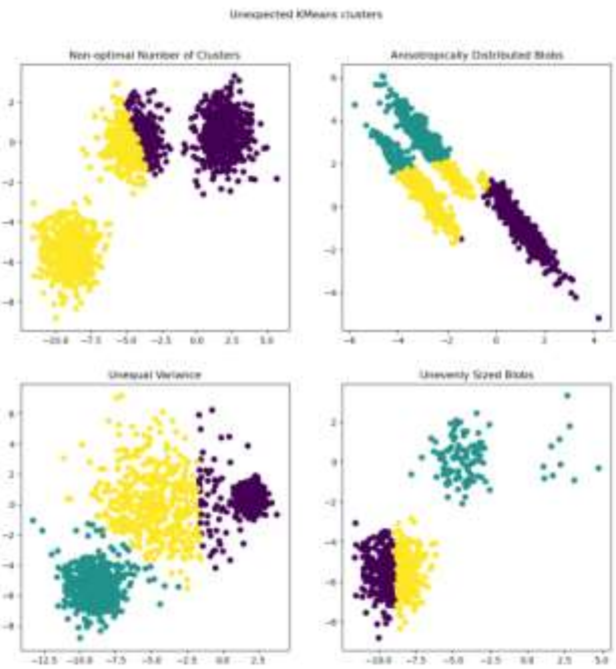

Figure 4. K-Means Issues

III. Implementation and Evaluation

The implementation of the application used to swap palettes is written using Matlab for its functions as well as Matlab GUI for its interface. The application created follows the following procedure:

1. User provides a base image as input.
2. User may experiment and enter the number of clusters used by K-Means to segment the image.
3. Program will segment with K-Means, display the segment image as well as its color palette.
4. User may experiment by selecting a color in the color palette, as well as selecting a color to transform the color into.
5. Program will take the target color, output color, and display an image with the target color from the original image replaced by the output color.

A. Matlab GUI

The application interface is developed using Matlab GUI and provides the following basic functionalities:

1. Uploading an image and downloading the transformed image.
2. Various buttons, drop down, and input boxes used for the user to interact with the application by segmenting and choosing a color from the palette.
3. Input Boxes as a stand in for color picking, as Matlab does not have a built-in color picker.

B. K-Means Segmentation

User will provide the number of clusters to be used in K-Means (default is 3). Using the kmeans function provided in the Statistics and Machine Learning Toolbox Add-on in Matlab, the function **kmeans_segmentation** will segment the image into clusters, and then return the segmented image as well as the color palette clustered by the K-Means.

Below is the code for **kmeans_segmentation** function.

```
function [segmented_image, color_palette] =
kmeans_segmentation(image, n_clusters)
    % Convert the image to double
    image = im2double(image);

    % Reshape the image into a 2D matrix
    [height, width, num_channels] = size(image);
    reshaped_image = reshape(image, height *
width, num_channels);

    % K-means clustering
    [idx, centroids] = kmeans(reshaped_image,
n_clusters);

    % Assign each pixel to its corresponding
cluster centroid
    segmented_image = zeros(size(reshaped_image));
    for i = 1:n_clusters
        cluster_pixels = reshaped_image(idx == i,
:);
        segmented_image(idx == i, :) =
repmat(centroids(i, :), size(cluster_pixels, 1),
1);
    end

    % Reshape the segmented and return
    segmented_image = reshape(segmented_image,
height, width, num_channels);

    % Extract color palette from the centroids and
return
    color_palette = centroids;
end
```

C. Color Masking

After the color palette has been defined and created, user may select a color and the program will use the function **createColorMask** to create a mask based on the selected color with a deviance threshold of 0.1 and display the section of image with said color.

Below is the code for **createColorMask** function.

```
function binaryMask = createColorMask(inputImage,
targetColor, threshold)

    % Convert the input image to double
    inputImage = im2double(inputImage);

    % Reshape targetColor to image size
    targetColorMatrix = ones(size(inputImage)) .*
reshape(targetColor, [1, 1, 3]);
```

```
    % Calculate the distance of color and image
    colorDistances = sqrt(sum((inputImage - targetColorMatrix).^2, 3));

    % Mask with threshold
    binaryMask = colorDistances <= threshold;
end
```

D. Color Swapping

After selecting a color, user may enter a new color (outputColor) to replace the previous color (targetColor). The program will use the function **colorManipulation**, it will create a mask of an image using createColorMask to define the sections of the image to be transformed, then using basic image processing, transform the targetColor into outputColor while maintaining small deviations in the form of shading or small gradients

Below is the code for **colorManipulation** function.

```
function outputImage = colorManipulation(inputImage, targetColor, threshold, outputColor)
    % Convert the input image to double
    inputImage = im2double(inputImage);

    % Create a binary mask based on color similarity
    binaryMask = createColorMask(inputImage, targetColor, threshold);

    % Convert targetColor and outputColor to double for consistent data types
    targetColorMatrix = ones(size(inputImage)) .* reshape(targetColor, [1, 1, 3]);
    outputColorMatrix = ones(size(inputImage)) .* reshape(outputColor, [1, 1, 3]);

    % Calculate deviation of input and targetColor
    deviationMatrix = targetColorMatrix - inputImage;

    % Calculate the output color transformation
    targetToOutput = outputColorMatrix - targetColorMatrix;

    % Calculate new image
    outputImage = inputImage + bsxfun(@times, deviationMatrix, binaryMask) + bsxfun(@times, targetToOutput, binaryMask);

    % Normalize pixels
    outputImage = max(0, min(1, outputImage));
end
```

E. Examples

Below is an example of 2 images used to test the application following the procedure for palette swapping. The first image is a basic logo without any shading or deviations, while the second is a more complex image with shading.

1. Input Image

Figure 5. Input Images of Word Logo

Figure 6. Input Images of Lena Image

2. K-Means Segmentation

Figure 7. Segmented Word Logo using K-Means with 2 clusters

Figure 8. Segmented Lena Image using K-Means with 7 clusters

3. Color Palette Results

Figure 9. Word Logo Color Palette

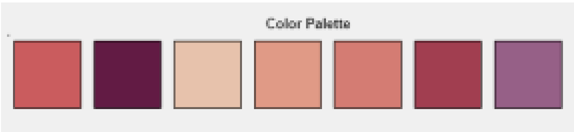

Figure 10. Lena Image Color Palette

4. Color Palette Selection

Figure 11. Word Logo Blue Color Mask

Figure 12. Lena Image Dark Purple Color Mask

5. Color Transformation

Figure 13. Word Logo Transformed Color Mask

Figure 14. Lena Image Transformed Color Mask

6. Transformed Image

Figure 15. Transformed Word Logo

Figure 16. Transformed Lena Image

The application is easily able to swap the palette of the first image due to its simple undetailed nature. The Application however isnt as accurate in segmenting and swapping the color if the second more detailed image. Still, it is able to give a basic idea of how the image would look like if it were to be swapped to a deeper purple and thus serving its main purpose, experimenting with color palettes.

IV. Conclustion and Suggestions

The concept of palette swapping has been implemented using the application by using K-Means Clustering and basic image processing. The dominant colors represented by the color palette can be replaced with a new color allowing users to experiment to create designs with different color palettes.

However, the implementation of the palette swapping is very basic and fails to recolor detailed images. Experimentation and research with other segmentation methods is recommended. Other functionalities such as custom thresholds and deeper control of the image processing as well as quality of life upgrades is preferred to improve the user experience.

Github Link

https://github.com/JayaMangalo/palette-swapping-kmeans

Acknoledgement

The writer expresses deep gratitude to God for providing blessings throughout the research process. Special thanks are extended to Professor Dr. Ir. Rinaldi Munir, M.T. the instructor for IF4073 Image Processing, for invaluable insights and knowledge. The writer also acknowledges the contributions of previous researchers whose work serves as a foundation for this study.

References

[1] https://www.smashingmagazine.com/2010/01/color-theory-for-designers-part-1-the-meaning-of-color/ accessed on 19 December 2023.

[2] https://www.colormatters.com/color-and-design/basic-color-theory accessed on 19 December 2023.

[3] https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2023-2024/22-Segmentasi-Citra-Bagian1-2023.pdf accessed on 19 December 2023.

[4] Solarski, C. (2012). Drawing Basics and Video Game Art. Watson-Guptill, New York. 69 Tillman, B. (2011). Creative Character Design. Waltham, MA: Elsevier Inc.

[5] https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2023-2024/23-Segmentasi-Citra-Bagian2-2023.pdf accessed on 19 December 2023.

[6] https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2023-2024/16-Warna-bagian1-2023.pdf accessed on 19 December 2023.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 December 2023

Jaya Mangalo Soegeng Rahardjo - 13520015