

Department of Informatics

**Color Palettes: Pattern Recognition and Classification of Images**

Master Thesis Description

**Introduction**

In visual multimedia documents such as images (or sequence of images such as gif animations and videos), a commonly used method for the specification of color distribution are **color palettes**. Using a color palette, a user can efficiently determine all matching colors of an image. While there exist tools in computer vision to extract a color palette from an image, using predefined color palettes for **image classification** could lead to new insights about the use of colors in different domains such as web design, architecture, and online marketing. To gauge and minimize the discrepancy between the human visual system and the computationally determined color combinations both data visualization algorithms and insights from perceptions of a human visual system are leveraged.

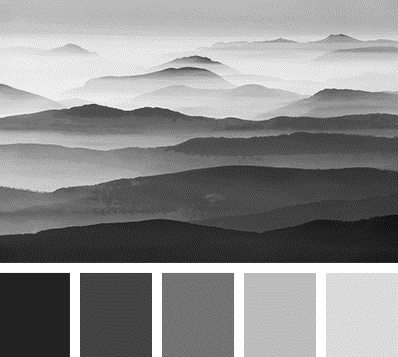


Figure 1: An example of an Image and a Color Palette

**Assignment**

The goal of this thesis is to optimize the **color distribution** of an image. The centerpiece of this procedure will be to use **color palettes** – an important tool for color image analysis. The colors of a color palette are plotted in an optimized color space model backed by color theoretical considerations. Further image processing hinges on categories of color palettes. Possible **color combinations** are determined and passed through a human visual filter system, which will help create a color corrected representation of an image.

**Task Description**

The thesis can be split into three main parts:

1. Extend a given color palette to all possible palette colors

From a predefined color palette, determine its closure: for each color patch in the color palette, plot its numerical representation into the most suitable color space. Specify the most suitable **color space** (CIELAB, Munsell color cone) and **color model** (HSV, RGB, CMYK, HEX or Adobe Photoshop models) for encoding the color of a pixel in the patch. Then determine a **pattern** - area or (non-)linear function - in color space that connects the color dots or clusters them together. From there, extract all possible color patches and add them to the original color palette.

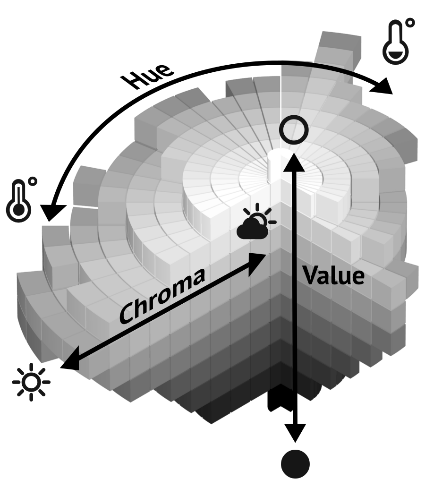


Figure 2: Munsell color system using HSV-color axes

1. Classify an image into a set of given color palettes

Given an image, classify it into one of k predefined color palettes. Determine the best count (with margin) of **superpixels** and transform the image into a set of s superpixels by aggregating the pixels from the image for each color hue into a higher-level compound (image segmentation).[[1]](#footnote-1) Locating objects and boundaries in in the image is key to determining superpixels [1]. Then minimize the distance between superpixel colors and all color patches belonging to a color palette (optimization). Identify the nearest color palette for the **image for classification**. An alternative approach is to use **machine learning**: learn a classification model from a set of given images and labeled color palettes, then predict the color palette of a new image.

1. Given a color palette, determine best color combinations

For all colors in a color palette, find matching colors by extracting all possible **combinations of colors** for a subset of up to four colors (pairwise, triples…). Experiment with neutrals (black, white, grey) or semi-neutrals (earth tones) that do not appear on the color wheel [2], textures (wood, brass, metals such as gold and silver, skin) and varying proportions (50-50 … 10-90) in unit space. Then, poll to rank or true-false categorize all color combinations according to a **test group** individual’s taste. Given an image, do image conversion by replacing the derived superpixels with the best-matching colors – the results of the survey - for a before-and-after effect. Compare results with findings from **color theory** [3].

1. Given a color palette, find all related images

Devise a search functionality where images can be searched according to their most salient color distribution for a given color palette.

**Requirements**

The implementation will be in **Python** or **R**. Familiarity with linear algebra is a must. For each task, a python script will be written using suitable image processing libraries such as OpenCV, Scikit-image, Matplotlib, PIL/Pillow (NumPy, SciPy, Mahotas, SimpleITK).

**Work Load**

60% Theory (1. Literature Review, 2. Data Collection, 3. Concepts and Model Design, 4. Methodology)

30% Implementation (5. Prototype, 6. Optimization, 7. Assessment)

10% Test (8. Survey evaluation, 9. Discussion, 10. Results)

**General Thesis Guidelines**

The standard rules of academic work must be followed. At the end of the thesis, a final report has to be written. The report should be clearly organized and follows the usual academic report structure.

**Tutor**

Barbara Flückiger

**Signatures**

Linda Samsinger Prof. Renato Pajarola, Ph.D.

**References**

1. L. Mouselimis. *Image segmentation based on Superpixels and Clustering*, 2019.
2. Theresa-Marie Rhyne. *Applying color theory to digital media and visualization*, 2012.
3. S. Bleicher. *Contemporary Color: Theory and Use*, 2012.

1. See project on Forground/Background Color Palette Annotation for Film <https://www.ifi.uzh.ch/dam/jcr:976a322f-4d01-4fd8-8e8d-e66832958d93/VIAN.pdf> [↑](#footnote-ref-1)