University of Zurich[UZH]

Department of Informatics

University of Zürich
Department of Informatics
Binzmühlestrasse 14
CH-8050 Zürich
Phone +41 44 635 43 11
Fax +41 44 635 68 09
www.ifi.uzh.ch

UZH, Department of Informatics, Binzmühlestr. 14, CH-8050 Zürich

Prof. Dr. Renato Pajarola
Visualization and MultiMedia Lab
Phone +41 44 635 43 70
Fax +41 44 635 68 09
pajarola@ifi.uzh.ch
vmml.ifi.uzh.ch

Master Thesis (30 ECTS)

Color Palettes: Pattern Recognition and Classification of Images

Name: Linda Samsinger

Student ID:

From: 1 March 2020 To: 31 August 2020

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.

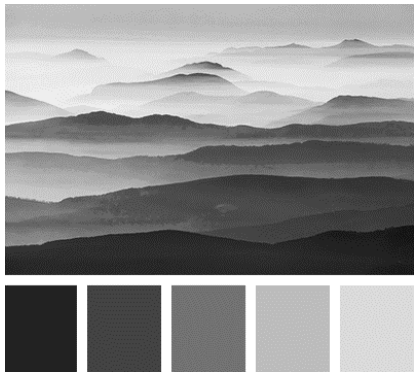

Figure 1: An example of an Image and a Color Palette

Description

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 human-tolerant, and color corrected representation of an image.

The project consists of the following 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.

2. 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). 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.

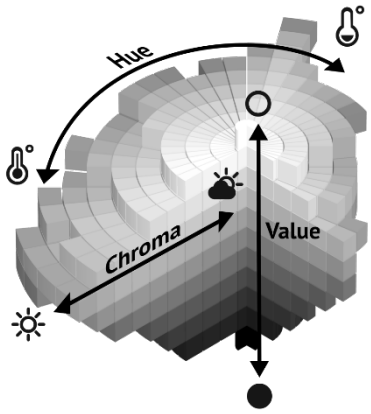

Figure 2: Munsell color system using HSV-color axes

3. Given a color palette, determine the 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].

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)

Remarks

In addition to the above described software, the student also has to write a report/thesis (according the the IFI rules) and defend it. This defense includes a live demonstration or video of the results. The code, a demo video as well as the report are part of the deliverables of the thesis. This thesis will be supervised by Prof. Dr. Renato Pajarola.

All source code written as a part of this thesis should be released under suitable open-source licenses. The typical rules of academic work must be followed.

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.