LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
INSTITUT FÜR INFORMATIK
LEHR- UND FORSCHUNGSEINHEIT FÜR PROGRAMMIER- UND MODELLIERUNGSSPRACHEN
Institut für Informatik

Colour Labelling of Art Images Using Colour Palette Recognition

Matthias Moosburger

Bachelorarbeit

Beginn der Arbeit:	8.12.2016
Abgabe der Arbeit:	17.2.2017
Betreuer:	Prof. Dr. François Bry
	Dr. Clemens Schefels

Erklärung

Hiermit versichere ich, dass ich diese Bachelorarbeit selbständig verfasst habe. Ich habe dazu keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

München, den 17. Februar 2017 Matthias Moosburger

Zusammenfassung

Die Farbkomposition ist eines der auffälligsten Merkmale, die an Kunstwerken wahrgenommen werden kann. Wenig überraschend verweisen Nutzer_innen des ARTigo Bildverschlagwortungs-Spiels sehr häufig auf Farbeindrücke um die visuellen Eigenheiten eines Kunstwerks zu beschreiben. Diese Arbeit widmet sich der Entwicklung eines algorithmischen Ansatzes zur Extrahierung von Farbpaletten aus Bildern von Kunstwerken zur automatischen Verschlagwortung markanter Farben. Den Herausforderungen im Entwicklungsprozess folgend, wird zunächst auf die digitale Darstellung von Farbbildern und verschiedenen Farb-Modellen eingegangen. Anschließend wird eine umfangreiche Analyse und Evaluation aktueller Techniken zur Bildsegmentierung und Farbquantifizierung durchgeführt, mit deren Hilfe die geeignetste Lösung mit Blick auf den Datenbestand von Kunstwerken ausgewählt wird. Weiter wird die gewonnene Farbpalette auf kompositionale Eigenschaften überprüft, unter Rückgriff auf die Farbkompositionstheorie des Kunst- und Farbtheoretikers Johannes Itten. Abschließend werden die so gewonnenen Farbinformationen auf sowohl elementare wie extensive Farbbegriffsmodelle abgebildet, wobei insbesondere auf das RAL Farbspezifikationsmodell zurückgegriffen wird.

Abstract

The colour composition of an art image is one of the most prominent feature being recognized. Unsurprisingly colour impressions are features users of ARTigo image tagging game are very frequently using to characterize visual characteristics of art images. In this thesis an algorithmic approach on extracting a colour palette has been developed, that allows automatic labelling of prominent colours for art images. Following challenges in the development process, first, the computational representation of colour images and different colour representation models are addressed. Then, a comprehensive discussion and evaluation of state-of-art image segmentation techniques and colour quantisation methods is conducted, in order to select the best combination of techniques with regard to art images as objects of analysis. The extracted colour palettes are further evaluated and filtered on compositional importance by consulting colour composition theory from art theorist Johannes Itten. Finally the extracted colours are mapped on basic and extensive colour naming systems using especially the RAL colour specification model.

Acknowledgments

I want to thank Prof. Dr. François Bry and Dr. Clemens Schefels
for constantly inspiring support and feedback.

Contents

1 Introduction **7**

2 Related Work **9**

3 Computational Representation of Colour and Colour Distribution **11**
3.1 Digital Colour Images and Colour Spaces 11
3.1.1 Colour Models and Colour Spaces 12
3.1.1.1 RGB 12
3.1.1.2 HSV and HLS 12
3.1.1.3 CIEL*a*b* 14
3.1.1.4 Other and Related Colour Systems 17
3.1.1.5 Colour Space Selection 17
3.2 Colour Histograms 18

4 Image Processing for Colour Extraction **21**
4.1 Image Segmentation, Superpixel Generation 21
4.1.1 Graph-Based-Segmentation: Felzenswalb 22
4.1.2 Gradient-Ascent-Segmentation: Quick Shift 26
4.1.3 Distance-Based Segmentation: SLIC 29
4.2 Colour Quantisation 31
4.2.1 Median Cut 31
4.2.2 Octree 32
4.2.3 K-Means 33
4.3 Modulation and Implementation for ARTigo Image Pool 34

5 Extraction of Image Specific Colour Palette **40**
5.1 Colour Composition Theory and Semantics 40
5.2 Implementation of Colour Semantics and Colour Palette Extraction 42

6 Automatic Colour Name Tagging **45**
6.1 Basic Colour Terms 45
6.2 Colour Name Sets and Standards 46

7 Implementation and Evaluation of Colour Tagging Algorithm **48**
7.1 Implementation of Colour Extraction and Colour Term Mapping 48
7.2 Evaluation 52
7.2.1 Results 52
7.2.2 Discussion of Results 55

8 Conclusion and Future Work **57**

Literature **58**

1 Introduction

> *A colour which would be 'dirty' if it were the colour of a wall, needn't be so in a painting*
>
> - Ludwig Wittgenstein, *Remarks on Colour [WAMS07, p. 28]*

As the British philosopher Ludwig Wittgenstein stated, the meaning of colours on paintings or artworks, does not necessarily correspond to our ordinary approach on the importance of colour perception. Artworks rather seem to attract special interest and attention on chromatic experiences. This special interest on colours can be seen in the ARTigo[1] art image[2] tagging game, where 5 of the 10 most used terms to describe art images are referring to chromatic features[3]. The interdisciplinary ARTigo research project [WBBL13] was developed by researchers from the departments of art history and computer science at the University of Munich (LMU) by implementing the *games with a purpose* approach (GWAP) [vAD08] on key-wording art works. ARTigo provides a web platform for playing a variety of image tagging games, motivating users through its gamification incentives, to provide rich empirical keyword tagging on art images. Users thereby explore art history on a playful way and can further investigate art history through semantic artwork search engines.

The dominance of colour to characterize images probably does not occur by incidence, but due to the artists' intentional application of colour as an instrument to achieve various semantically enriched effects. Also there are multiple examples and dimensions in art history, how colour was used to barrow semantical weight, like the representation of political power with certain reserved colours, or liturgics' colour usage principles. Therefore, a proper indexing of art images with representative colour terms, that takes into account the specific semantic role of colours in art images, provides a valuable understanding of themselves. Due to the overwhelming amount of art images, an automatic colour analytic technique for art images can support tedious work of spotting relevant colours terms, that provide high recognition and suitable characterisation.

This thesis will therefore explore a technique for automatic colour extraction and indexing, to support the tag based image classification platform ARTigo. Towards this goal, several techniques will be explored, to build a colour index algorithm, that detects semantical relevant parts of a art image, extracts feature colours on key regions, and eventually provides a list of potential appropriate colour terms. This task is faced with several challenges how to represent colour computationally in an appropriate manner (Section 3), how to efficiently generate coherent colour segments (Section 4.1) and how to extract relevant palettes of colours, to match a set of possible anticipated (basic) colour terms (Section 6.1). This thesis will try to discuss selected solutions, suggested in relevant literature and hence describes a implementation of a colour extraction algorithm, written in Python programming language, that suggests and validates colour terms as tags for the ARTigo image database.

[1] http://www.artigo.org

[2] Here and in the following, the term *artwork* will be used to refer to the actual piece of art, while *art image* denotes the visual representation of *art works* (computational in this context).

[3] *black*, *blue*, *red*, *white* and *green* are the five most assigned chromatic tags with *black* even providing the second most used tag on the game

On the side of developing algorithmic colour extraction and labelling, this project also generated lively techniques for colour space exploring, like 3-dimensional colour histograms (see Section 3.2). These and further informative processing interim results have been provided on a web-based interface. Users, interested in computer-assisted image analysis, thereby can explore interactively algorithmic processing to gain semantic information based on colour distribution (Section 7.1).

In order to meet *art* specific challenges on colour usage, colour composition theory has been consulted as approach to discover prominent chromatic regions and improve the selection outcome. Artists often accentuate the importance of specific regions or represented objects by applying complementary colour composition to emphasize semantic weight. The colour composition theory developed by art scientist Johannes Itten (Section 5.2) will be taken into account, by providing a sophisticated framework that can be translated to algorithmic colour analysis (Section 7.1).

2 Related Work

In scientific literature the field of (semantically supported) colour extraction and segmentation is widely discussed and different approaches have been developed and tested. The following overview on selected research projects tries to apprehend the three major research aspects that guided the development in this thesis: Colour extraction from research on image retrieval, comparison of image segmentation and quantification approaches and the choice of appropriate colour naming models

In general, most of colour interested research has been done for image retrieval, i.e. the technique of indexing images for indexed features based queries, like the feature of a dominant colour. A highly referenced work was published by John R. Smith and Shih-Fu Chang from Columbia University, New York in 1995 [SC95], presenting an automatic single colour extraction method for image query. They identified regions in images fitting to a predefined colour set with further determination colour regions' location and size. Thereby they offered a enriched query on chromatic as well as spatial parameters regarding colour regions.

Another interesting approach was presented by Y. Liu, D. Zhang, G. Lu and W-Y. Ma from Monash University, Australia in corporation with Microsoft Research Asia, China in 2004 [LY13]. They also developed a region-based image retrieval system, allowing to retrieve images with high-level semantic colour names by segmenting images using HSV colour space. They argue that HSV colour space (see Section 3.1.1.2) provides a proper colour model to distinguish human perceptional qualities of colour separating hue, saturation and luminance. The image query was done by selecting specific regions on images and retrieving similar images with low distance both, colour location and colour value.

Y. Sirisathitkul, S. Auwatanamongkol and B. Uyyanonvara [SAU04] from Thammasart University in Bangkok, Thailand in 2004 further developed the segmentation tools being used to improve colour constancy on quantisation processing by comparing several earlier and state-of-art segmentation techniques. With few optimisations and pre-selecting colour on variance, they achieve noticeable lower quantisation errors. The major profit of this work thereby lies in its extensive discussion and comparison of "splitting algorithms" and "cluster-based approaches", which have been incorporated as quantisation and segmentation methods in Sections 4.2 and 4.1. Another profound comparison of state-of-art image segmentation algorithms was given by L. Lucchese and S.K. Mitra from University of California, Santa Barbara in 2001 [LM01] providing an in depth analysis on segmentation strategies with respective best-case recommendations.

Many colour analysis approaches took "real-world" images as a basis, with the advantage of getting satisfactory training data for machine learning approaches. J. van de Weijer, C. Schmid and J. Verbeek from University Grenoble, France [vdWSVL09] proposed a learning-based colour naming method that used training data from *ebay* images, which usually show good colour description. Although they could show that colour names learned from real world images outperform colour names based on predefined colour sets, the machine learning approach is not taken into account in this thesis. The main differences to [vdWSVL09] are given with less reliable training data as well as art images as data basis with less clear-cut colours represented than Weijer et al. were given using *ebay* images.

The work of J.M. Corridoni, A. del Bimbo and E. Vicario from University of Florence, Italy in 1997 [CDP99] bears more relevance. Corridoni, del Bimbo and Vicario developed an image retrieval method, exclusively on paintings as data basis. They yet made use of segmentation processing and used visual filters to character chromatic characterisations of images, incorporating spatial features of colour composition into analysis. Most importantly, they introduced the strategy to consult colour semantic theory from art theorists of the 20th century to improve selection strategies for chromatic features. However, they provide less information in implementation compared to the rich discussion on strategies accommodated to analyse art-paintings in particular.

The European research group of K. Ivanova, P. Stanchev and K. Vanhoof from University of Sofia, Bulgaria [ISD08, ISVD12, IS09, Iva11, ISV+12] provides very comprehensive research on colour based analysis of art images. The research group developed a well-engineered retrieval system APICAS (Art Painting Image Colour Aesthetics and Semantics), which allows extraction of several high-level semantic visual features. APICAS provided colour distribution analysis by quantisation of image colour in HSL colour space, a dominant colour selector and colour layout as well as texture analysis. The leading analysis schema has been adopted by MPEG-7 description system. A major knowledge gain of this research project was given by identifying colour palette as powerful tools for creating profiles of art painting images [ISV+12]. A similar approach is presented in Section 5.2 of this thesis.

The third aspect in relevant scientific literature affects the choice of appropriate colour naming models, that allows to close the semantic gap between numerical colour specification and semantical high-level colour naming practise. While many of the above mentioned research approaches are relying on a singular colour naming model, many research projects have been undertaken to obtain empirical approved identification of reliable colour name models.

J. Heer and M. Stone from Stanford University in 2012 [HS12] developed a method for constructing a probabilistic model of colour naming from a large, unconstrained set of human colour name judgements. As importantly they also discussed several colour naming models and palette designs on colour saliency and colour name distance. This analysis allows to select a colour naming model suitable for respective requirement. More generally, most of the colour naming studies have been devoted to the debate, whether the basic colour term survey of B. Berlin and P. Kay [KF09] delivered a reproducible model, by defining 11 to 12 basic colour terms.

The study of V. Setlur and M. Stone [SS16] and D. Conway's experimental comparison of natural language colour naming models [Con92] approved Berlin and Key's findings largely, but also emphasised the need for further expansion of Berlin and Key's limited list of colours to include more colour hues like 'teal' or 'apricot'. See Section 6.1 for further discussion on colour naming models.

3 Computational Representation of Colour and Colour Distribution

The ARTigo database offers a collection of photographs and scans of art images, created by artists using various materials and techniques. To explore the accurate sensation and appeal an art work induces in human perception most accurately, one certainly has to refer to the original material. But the image reproductions in the ARTigo image database also allows to set different art works with various colour impressions in compilation and to compare primary colour impressions. This opportunity promises new knowledge on colours as a central and striking quality of art images and new perceptions on similarities, contrasts or evolutions of colour characteristics. This section addresses first a discussion on how colour is generally represented in digital images. Then, a section on the algorithmic approach on how to process images, extracting primary and important colours follows, including insights of colour modelling.

3.1 Digital Colour Images and Colour Spaces

Colour images are digitally represented in various ways. Depending on the intended application field different techniques of colour image representation have been developed and are still used simultaneously. Colour image representation techniques can be divided into two classes with regard to their *primitives* used: pixel-based and vector-based images. [BB16]

Vector based colour images use assemblages of polygons assigned with style properties. These style definitions allow to assign solid colour values as well as colour flows to polygons. Pixel based colour images are made up by 2-dimensional arrays (corresponding length and height of a picture) to store pixels as primitive elements. Every pixel defines its colour information independently, with equally coloured image areas are assemblies of pixels with equal or slightly varying colour information. Therefore, colours in pixel-based images are distributed independently and any perceived coherence, e.g., solid colour areas or colour flows, are information not stored in the picture primitives themselves but rather are the result of unifying colour perception (see Figure 9).

Most of the pixel based colour images including JPEG [Itu93] and PNG [Duc03] are build up by primary colours red, green and blue (RGB) [SBS99]. These image formats typically take 8 bits for every component and therefore, require $3*8 = 24$ bit for each colour pixel. In professional applications, colour depths of 30, 36 or even 42 bits per pixel are also common. Most of the image formats make usage of 3 colour components per pixel, depending on the used colour space, but there are also image formats common in pre-press application, which represent colours with four or more components based on the subtractive CMYK colour model. Another important technique to store colour information in pixel based images is used in indexed or palette images. These images are restricted in the number of different colours. Individual pixels are assigned to certain colours from the additionally given colour table. This technique allows efficient storage but is less accurate in colour rendering. In an indexed image the number of different colours is usually very restricted. The GIF image standards allows up to 512 distinct colours, while a true colour image in 24 bit encoding allows $2^{24} = 16$ Million distinct colours being represented [Inc90].

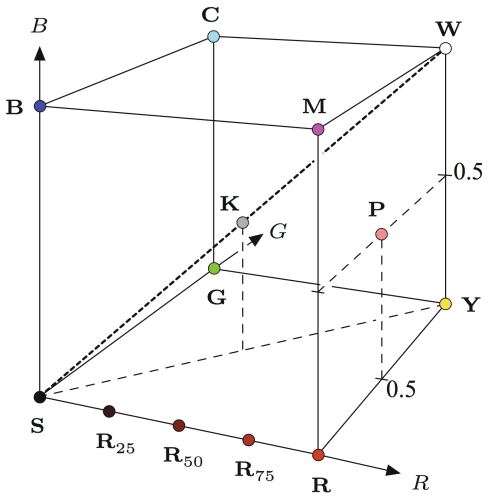
3.1.1 Colour Models and Colour Spaces

Colour spaces contain the possible colour gamut, i.e., the range of definable colours, organised in specific mathematical models. These mathematical models, then called *colour models*, define the way in which colours can be represented as numeric tuples. Usually colour models define triples of numbers (e.g. RGB, HSV, HSB, CIE*L*a*b) or use quadruples (e.g. CMYK). Well known colour spaces are either hardware oriented, such as RGB and CMYK, or user-oriented, such as HLS, HSB, CIEL*a*b* and CIEL*C*h*. While hardware oriented colour spaces are tightly bound to technical reproduction with light sources (RGB) or mixture of pigment colours in printing (CMYK), the user oriented colour spaces are the most suited to support human-like perceptive analysis.[CDP99, p. 176] The determination of a specific colour model as the most suited for an specific application has to consider the specific characteristics and advantages of the different colour models.

3.1.1.1 RGB Probably the most commonly known colour space is spread out by the RGB colour model. The RGB colour model constructs colours as mixtures of three primary light colours red, green and blue with varying intensities. RGB colour system is popular in computer programming [BB16, p. 303], since it is simple to manipulate and maps directly to typical display hardware. Colours in RGB space are represented as triples of numbers corresponding to the three primary colour axes. With RGB colour images often use 24 bit per pixel as colour value, the axis values are positive numbers and lie in the range [0, 255]. Every possible colour c_i therefore corresponds to a point within the RGB colour cube of the form $C_i = (R_i, G_i, B_i)$.

Modification and colour distance measurement reveal some disadvantages of the RGB colour system. As hardware oriented colour system the metric or measured distance within the colour space doesn't correspond proportionally to human perception of colours. Modifying different colours in the RGB space by the same amount can cause very different changes in colour. For example, a slight modification of the red component produces highly perceivable changes. Modifying the green component by the same amount, non-noticeable effects for human perception can be observed. Additionally, the brightness change in this colour system is not perceived as linear. And as a final disadvantage, the RGB colour system represents a hardware related component model which makes it difficult to select a colour from a human perspective, since we usually do not relate to colours as compositions of red, green and blue light, but rather use classifications similar to hue, lightness and saturation.

3.1.1.2 HSV and HLS In order to address these disadvantages of RGB colour space and offer much more perceptional oriented colour models HSV and HLS have been developed. These colour systems have been established in many applications as they provide a more intuitive way of selecting colours [IS09]. HSV and HLS are both user-oriented colour systems that specify colours by three components *hue, saturation* and *luminance* or *value*. Often the HSV system is also called HSB to reference Adobe's renaming of *"value"* to the more descriptive *"brightness"*. The HSV colour space is usually shown as an upside-down pyramid or cone (see Figure 2), where the vertical axes corresponds to the V (value or brightness) axes, the horizontal distance from the S axes as colour saturation and the angle to the H



		RGB values		
Pt.	Color	*R*	*G*	*B*
S	Black	0.00	0.00	0.00
R	Red	1.00	0.00	0.00
Y	Yellow	1.00	1.00	0.00
G	Green	0.00	1.00	0.00
C	Cyan	0.00	1.00	1.00
B	Blue	0.00	0.00	1.00
M	Magenta	1.00	0.00	1.00
W	White	1.00	1.00	1.00
K	50% Gray	0.50	0.50	0.50
$\mathbf{R}_{75}$	75% Red	0.75	0.00	0.00
$\mathbf{R}_{50}$	50% Red	0.50	0.00	0.00
$\mathbf{R}_{25}$	25% Red	0.25	0.00	0.00
P	Pink	1.00	0.50	0.50

Figure 1: Representation of RGB colour model as 3D cube and a table of several colours located on vertices (B, M, W, C, Y, G, S and R) and selected colour shades. All shades of grey with K as an example lie on the diagonal between S and W [BB16, p. 305]

(hue) value. The black point is located at the tip of the cone and the white point in the centre of the layer with maximum of value/brightness. While the colour model in principal allows a cylindric colour space, colours with decreasing saturation values are mapped to the back point at the tip, where saturation value is dispensable.

The HSL colour system defined by components *hue, luminance* and *saturation* provides a slight modification of HSV. While the hue-component in bot systems are identical, HLS corresponds to HSV to the vertical axis and the radius respectively with defining these axis differently. HLS constructs a double pyramid with the white point on one and the black point on the opposing tip (see Figure 3).

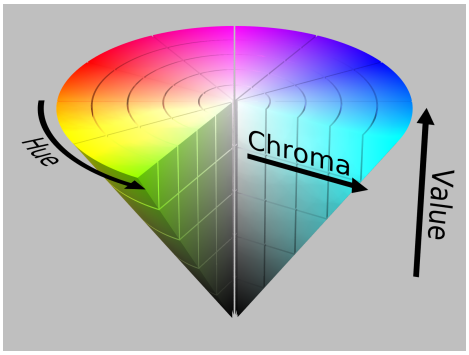


Figure 2: HSV colour space as cone [Shab]

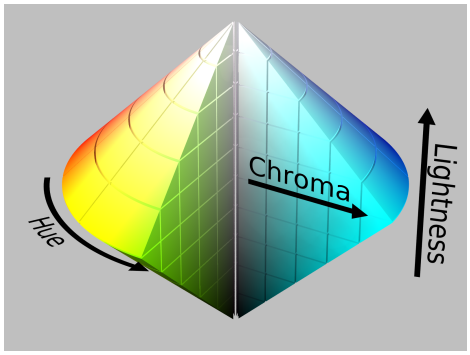


Figure 3: HSL colour space as double cone [Shaa]

To convert RGB colour values to the HSV colour system first the saturation component is calculated as [BB16, p. 308f]

$$S_{HSV} = \begin{cases} \frac{C_{high} - C_{low}}{C_{high}} & C_{high} > 0 \\ 0 & otherwhise \end{cases} \tag{1}$$

with $C_{max} = max(R, G, B)$ and $C_{min} = min(R, G, B)$.

Then brightness (value) component

$$V_{HSV} = \frac{C_{high}}{C_{max}} \tag{2}$$

with C_{max} being the maximum component value (typically 255).

Finally the hue value H_{HSV} is specified as 0 if all the RGB components have the same value ($R = G = B$) and the corresponding colour is achromatic (grey). For $C_\Delta = C_{high} - C_{low} \neq 0$ first each component of RGB is normalised:

$$R' = \frac{C_{high} - R}{C_\Delta}, \quad G' = \frac{C_{high} - B}{C_\Delta}, \quad B' = \frac{C_{high} - B}{C_\Delta} \tag{3}$$

Then depending on which of the three colour components had the maximum value, the hue H' is calculated as

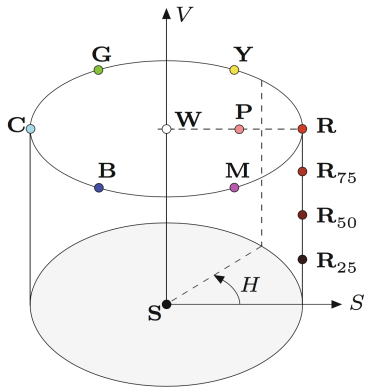
$$H' = \begin{cases} B' - G' & \text{for } R = C_{high} \\ R' - B' + 2 & \text{for } G = C_{high} \\ G' - R' + 4 & \text{for } B = C_{high} \end{cases} \tag{4}$$

And with the resulting value for H' lying in the interval $[-1, 5]$, the final hue value is normalised to the interval $[0, 1]$ with

$$H_{HSV} = \frac{1}{6} \cdot \begin{cases} (H' + 6) & \text{H'} < 0 \\ H' & \text{otherwise} \end{cases} \tag{5}$$

The mapping from RGB to HSV is nonlinear with the RGB colour space is transformed to a cylinder in HSV colour model by spreading out the black point completely across the cylinder's base. Figure 4 shows the mapping of some selected colours from RGB colour space to HSV and compares their location in both spaces.

3.1.1.3 CIEL*a*b* The CIEL*a*b* [BG14] colour model was developed by the Commission Internationale de l'Éclairage (International Commission for Illumination) in 1976 with the goal of linearising colour representation with respect to human colour perception. This way the CIEL*a*b* colour model provides both a intuitive colour system as well as linear colour distribution for human perception. The CIEL*a*b* colour model became a popular and widely used colour system and is used in many professional image processing applications standard model for converting between different colour spaces. To understand CIEL*a*b* and to convert colour from RGB to CIEL*a*b* an understanding of the colorimetric colour spaces is required.



RGB/HSV values

Pt.	Color	R	G	B	H	S	V
S	Black	0.00	0.00	0.00	—	0.00	0.00
R	Red	1.00	0.00	0.00	0	1.00	1.00
Y	Yellow	1.00	1.00	0.00	1/6	1.00	1.00
G	Green	0.00	1.00	0.00	2/6	1.00	1.00
C	Cyan	0.00	1.00	1.00	3/6	1.00	1.00
B	Blue	0.00	0.00	1.00	4/6	1.00	1.00
M	Magenta	1.00	0.00	1.00	5/6	1.00	1.00
W	White	1.00	1.00	1.00	—	0.00	1.00
$\mathbf{R}_{75}$	75% Red	0.75	0.00	0.00	0	1.00	0.75
$\mathbf{R}_{50}$	50% Red	0.50	0.00	0.00	0	1.00	0.50
$\mathbf{R}_{25}$	25% Red	0.25	0.00	0.00	0	1.00	0.25
P	Pink	1.00	0.50	0.50	0	0.5	1.00

Figure 4: HSV colour cylinder as a result of transformation from RGB. The table lists selected colour values with corresponding RGB und HSV values [BB16, p. 309].

All of the colour spaces discussed above are somehow related to physical properties of some media devices, such as the luminance of LEDs on a LED matrix display. To describe colours in a device-independent way which provides reliable colour transformation between different media devices the CIE standardised the XYZ colour system in 1931. This colour system is still the most used colourimentric colour system today [BB16, p. 344] and was developed after extensive measurements of human visual perception. The CIEXYZ colour system is based on three imaginary primary colours X, Y, Z which can be combined with positive values to produce all visible colours. The three imaginary colours thereby spread out a 3D cone-shaped region, with the Y component corresponding to the perceived lightness or luminosity of a colour. The CIEXYZ colour system still is *nonlinear* with respect to human visual perception, but its device-independence justifies its status as reference colour system.

To describe pure colour hues and saturation in a convenient manner, the CIE system also defined three *chromacity* values x, y, z.
[BB16, p. 346]

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}, \quad z = \frac{Z}{X+Y+Z} \tag{6}$$

where $x + y + z = 1$ and thus one of the three component is redundant. Figure 5 shows the resulting CIE chromacity diagram.

To realise device-independent and linear colour representation the CIE further developed the CIEL*a*b* colour model. The CIEL*a*b* colour model specifies colours in three components: $L*$ (luminosity) and $a*$ and $b*$ (specifying hue and saturation along *green-red* and *blue-yellow* axes, respectively). All of those three values are *relative* values and refer to a specified white point $C_{ref} = (X_{ref}, Y_{ref}, Z_{ref})$. Values on colour axes a and b vary in the interval between -150 (green on axis $a*$, yellow on axis $b*$) and 100 (red on axis $a*$, blue

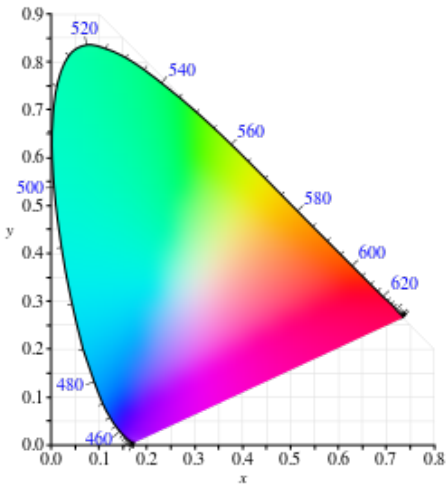


Figure 5: CIEXYZ chromacity [Ben]

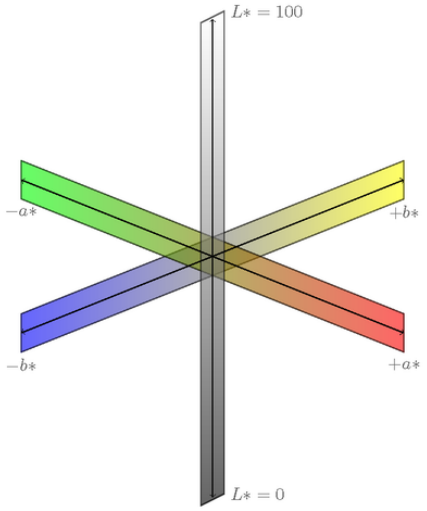


Figure 6: CIEL*a*b* colour space [Vie14]

on axis $b*$), on $L*$ axis between 0 (black) and 100 (white). Figure 6 shows the resulting 3-dimensional sphere.

A major advantage of CIEL*a*b* besides its device independence is the colour distance measuring function $\Delta E*_{ab}$ (also referred as "Delta E function") which provides linear colour distance measurements regarding to human visual perception based on euclidean distance:

$$\Delta E_{p,v} = \sqrt{(L_p^* - L_v^*)^2 + (a_p^* - a_v^*)^2 + (b_p^* - b_v^*)^2} \quad \text{[SWD05]} \tag{7}$$

with several refinements and corrections defined in the L*C*h* colour space with differences in lightness, chroma and hue calculated from L*a*b* coordinates. For a comprehensive description of $\Delta E*_{ab}$ and its modifications to its present version CIEDE2000 see [SWD05].

To transform RGB colour space to CIEL*a*b* first a conversion from RGB to CIEXYZ has to be done and second the CIEXYZ colours are translated to CIEL*a*b* colour space. For transforming RGB colour information to device-independent CIEXYZ colour space, a conversion matrix from sRGB colour space is provided among other. The sRGB standard [Com99] (standard RGB) was developed by Hewlett-Packard and Microsoft to create a precisely specified constitution of the RGB colour space independently from colour capturing media. The sRGB system is the de facto standard for digital cameras and several standard image formats, including EXIF (JPEG) and PNG are based on sRGB colour data. If image data is given in standard image format, like JPEG, the colour information is transformed to CIEXYZ with linear product with a colour matrix

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0,4124 & 0,3576 & 0,1805 \\ 0,2126 & 0,7152 & 0,0722 \\ 0,0193 & 0,1192 & 0,9505 \end{bmatrix} \cdot \begin{bmatrix} R_{sRGB} \\ G_{sRGB} \\ B_{sRGB} \end{bmatrix} \tag{8}$$

[SWD05]

To finally convert CIE-XYZ coordinates to the CIEL*a*b* colour space a white point X_n, Y_n, Z_n with

$$\begin{aligned} L^* &= 116 \cdot \sqrt[3]{\frac{Y}{Y_n}} - 16 \\ a^* &= 500 \cdot \left(\sqrt[3]{\frac{X}{X_n}} - \sqrt[3]{\frac{Y}{Y_n}} \right) \\ b^* &= 200 \cdot \left(\sqrt[3]{\frac{Y}{Y_n}} - \sqrt[3]{\frac{Z}{Z_n}} \right) \end{aligned} \tag{9}$$

[SWD05]

3.1.1.4 Other and Related Colour Systems Next to the colour systems and spaces discussed above, which cover main applications, other colour systems have been established to meet application specific needs. Regarding encoding of television signals the colour spaces YUV, YIQ and YCbCr play an important role as fundamental colour-encoding models for NTSC, PAL and MPEG Video standard. All of these colour spaces have in common the idea of separating the luminance component Y from two chroma components and, instead of directly encoding colours, encoding colour differences. In this way, compatibility with legacy black and white systems is maintained while at the same time the bandwidth of the signal can be optimised by using different transmission bandwidths for the brightness and the colour components. [BB16, p. 317]

Similar to CIEL*a*b* the International Commission on Illumination also defined the CIELUV [Sch16] colour space, which is also constructed by transformation of the CIEXYZ space. CIELUV is also attempted to provide perceptual uniformity and is primarily used for chromatic analysis dealing with coloured lights.

3.1.1.5 Colour Space Selection The colour extraction system of this thesis uses the colour spaces RGB, HSV and CIEL*a*b* and thereby follows multiply recommended practice [BB16, CCL+01, MHS02]. The images of ARTigo platform are provided as scans or photographs and are stored in JPEG files using the (s)RGB colour space. Since segmentation and quantisation processes are done, a major requirement is to ensure, that these processes follow the human perception of colours regarding perceived resemblance and distance comparison. Therefore,, the images are converted to CIEL*a*b* colour space, since CIEL*a*b* offers appropriate perceptual uniformity. With the segmentation algorithms (see Section 4.1) heavily relying on colour distances calculation, it is very important to measure distance oriented on human perception of colour difference. The CIEL*a*b* colour space offers a colour distance function $\Delta E*_{ab}$ which is used in its current standardised version CIEDE2000. Thereby, it is assured that segmentation processing approximates human colour difference perception and judgement

After the first segmentation process, the substitutional colours of image segments are adjusted by gamma correction on their colour saturation and brightness to react to averaging effects (see Section 5.2). For this purpose the colours are intermediately converted

to HSV colour space, providing separate saturation and brightness colour channels. Since CIEL*a*b* allows conversion methods from and to common colour systems, these conversions are without loss, so long as computational accuracy isn't changed.

Perceptional uniform colour distance measurement is also a crucial, when extracted primary colours are mapped to a pre-given colour term set. The $\Delta E*_{ab}$ colour distance function provides a suitable metric for this task. In order to calculate the distances between a given colour and the reference colours the CIEL*a*b* colour model is chosen as fundamental colour space. A minimum in distance between extracted colour and reference colour can therewith be interpreted as perceptional similarity.

3.2 Colour Histograms

A first approach to compare images with regard to their colours is often suggested by using colour histograms. Colour histograms in general are frequency distributions of intensity values occurring on an image. For monochromatic and grey-scale images, histograms provide a useful analysing tool and allows to classify images on their characteristic distribution of intensity values. For colour images usually the distribution of individual colour channels or image intensity (luminance) is meant. Both, luminance and component histograms provide useful information about contrast, lightning and dynamic range relative to individual colour components. Figure 7 shows an image, which was chosen because of its great variety of bright colours, next to its component colour histogram. But it is important to remember, that they do not provide any information on the distribution of *actual* colours, since these histograms are based on individual colour channels that in combination form the colour of an individual pixel. A collection of colour images can have very similar component histograms and still contain fully different colours. For example, an image with prominent yellow areas will show peaks in red and green colour channels, as yellow appears as mixture of red and green values in RGB. This image will show a very similar colour histogram compared with images showing separated green and red spots, even though the appearance of the two images differs greatly. Therefore,, for this thesis *combined* histograms are chosen to analyse colour distribution in order to specify colour characteristics on images oriented on colour appearance. Nevertheless, approaches to compare images and colour distribution based on component histograms have been done in [CCL+01] and [LY13].

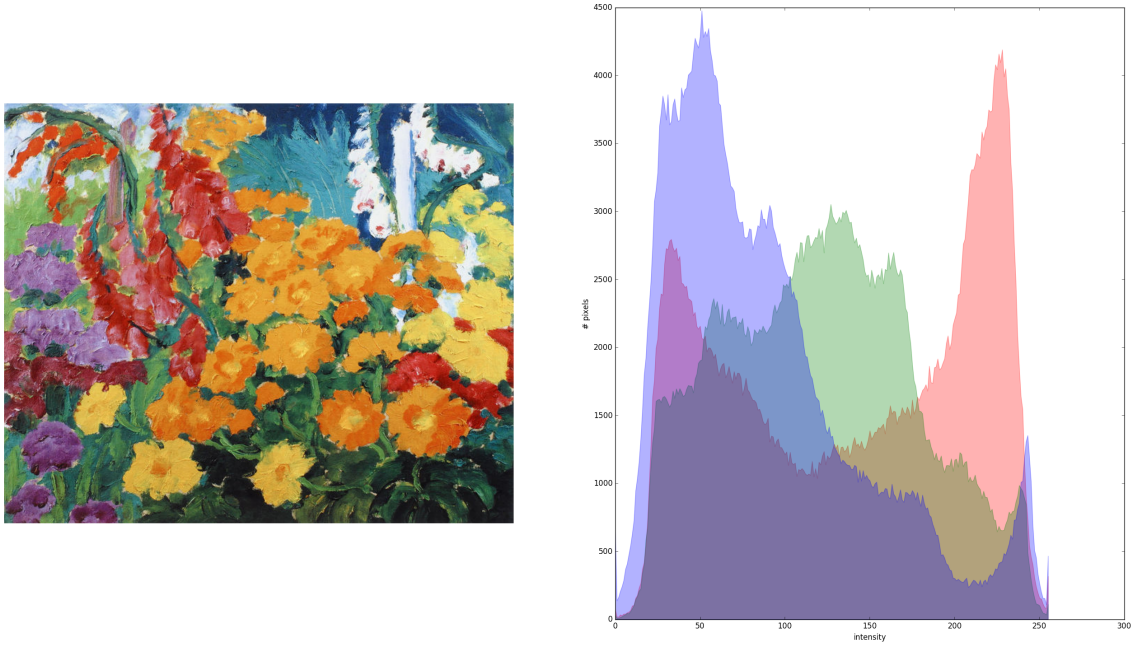


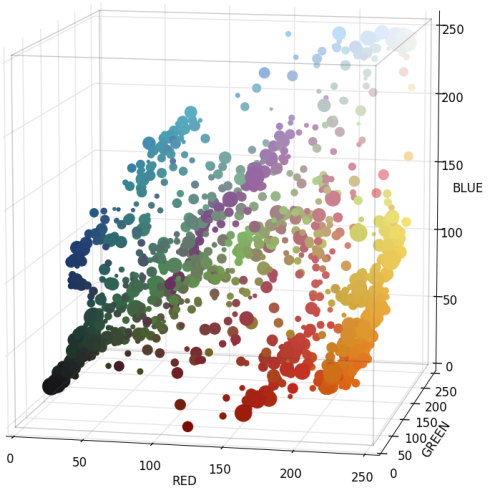
Figure 7: Colour histogram on all three colour channels of RGB image. Image: Emil Nolde: "Flower Garden (Marigold)" ("Blumengarten (Ringelblumen)") 1919.

The colour extraction library developed in this thesis provides creation of combined colour distribution histograms showing frequency of colours in a 3dimensional space in RGB, HSV and CIEL*a*b* colour space. These diagrams accumulate colour values in given arrays and plot spheres filled with respective colour and volume size linearly corresponding to frequency. Figure 8 shows an example image from ARTigo image library after its first segmentation process and its derived colour histograms in different colour systems. These histograms can be explored interactively either with Matplotlib 3D plot application or on the web-interface using plot.ly JS[4] library.

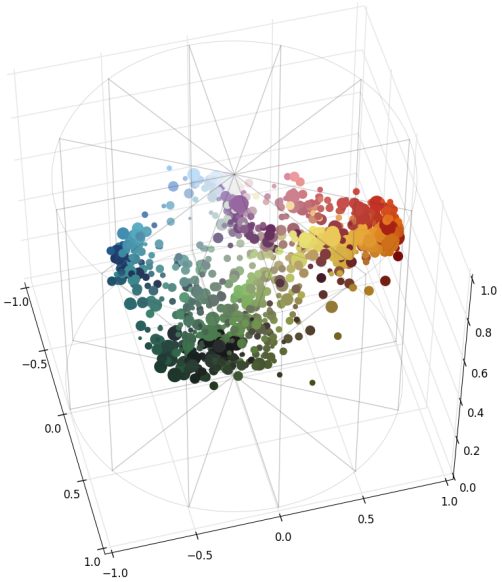
Additionally, this project also provides a web interface allowing to comprehend the segmentation and quantisation processes discussed in Section 4.2 and colour palette extraction discussed in Section 4. Given an colour image from the ARTigo image data set, the original image, the re-evaluated segmented image with its 3D histogram is shown as well as the result of final quantisation and colour palette extracted.

[4] https://plot.ly/javascript/

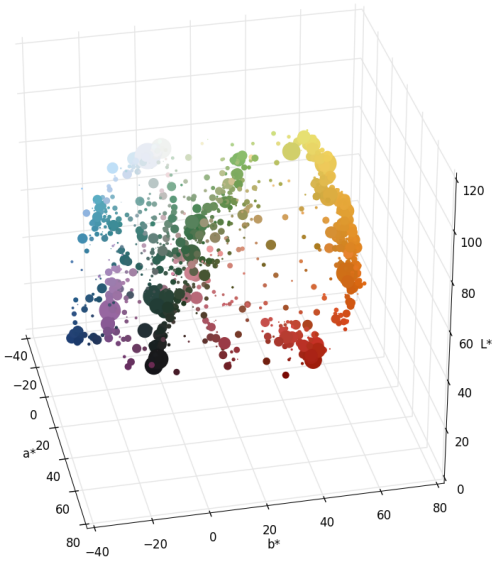
(a) Original image



(b) RGB histogram



(c) HSV histogram



(d) CIEL*a*b* histogram

Figure 8: 3D colour histograms with (a) original image, (b) RGB histogram, (c) HSV histogram and (d) CIEL*a*b* histogram. Image: Emil Nolde: "Flower Garden (Marigold)" ("Blumengarten (Ringelblumen)") 1919.

4 Image Processing for Colour Extraction

The colour images provided in ARTigo are digital JPEG images files. By the term "digital images" we understand a mapping function I mapping location coordinates of pixels (Loc) on colour information values (Val): $I : Loc \rightarrow Val$ with Loc defined as the finite set $Loc = L_1 \times \cdots \times L_d$ and $Val = V_1 \times \cdots \times V_k$. The variable k defines the amount of colour channels, which is set to 3 in (s)RGB JPG images and d specifies the number of dimensions, which is 2 for 2-dimensional images. With $3 * 8 = 24$ Bits for encoding colour information (V_1, V_2, V_3) for every pixel, the JPEG encoding strategy theoretically allows $2^{24} \approx 16.7$ Million distinct colour values to be shown on a single image. Of course this specification of colour doesn't yet describe the impression of colour produced by pixel adjacency. Adjacency in an image $I : Loc \rightarrow Val$ is defined as binary, symmetric and reflexive relation on Loc. Two pixel P, P' are called τ-*adjacent* (similar regarding their values) if $val(P)\ \tau\ val(P')$ applies. [Pri15, p. 66f]

When observing pixel images from a adequate distance those pixels where τ-*adjacent* holds are perceived as distinct and coherent areas. This mathematical definition describes a well known phenomenon of our daily experience on observing digital images. On a high resulted pixel image the human visual perception constantly strives to (re)construct τ-*adjacent* colour regions by aggregating colour impressions of separate pixels. With reducing resolution our capability to define these coherent regions decreases. Figure 9 illustrates this phenomenon by cutting out a region of an image and presenting this extract with reduced resolution by scaling it larger. While we easily spot a coherent bright red colour in the left image, it seems to be much more difficult to abstract this very distinctive colour on the right extraction.

This example illustrates a major challenge in extracting primary or important colours on digital images: The coherency of colours perceived by human visual perception is not given directly as information in a pixel-based digital image. We rather need a sound τ-*adjacency* method, which reconstructs the human visual perception and produces chromatically coherent areas. This section will first discuss several methods that provide these adjacency mappings by presenting superpixel generation algorithms. The second part then discusses further processing to extract a limited set of representative colours. Finally the modulation and implementation of the colour segmentation algorithm for ARTigo image pool is presented.

4.1 Image Segmentation, Superpixel Generation

Superpixel segmentation describes the processing of images into local and coherent segments preserving most of the structure necessary for segmentation at the scale of interest. [RM03] While singular pixels merely provide colour information, superpixels are defined as a set of pixels comprising further (colour) context information by segmenting the image in an essentially smaller set of (super)pixels.

There are several methods and algorithms discussed in literature on how to generate superpixel segmentation (see [LM01] for a comprehensive overview on "state-of-art" segmentation algorithms). Generally these algorithms can be classified into three classes: *graph-based* methods, *gradient-ascent-based* methods and *distance-based* methods [VS08]. Graph-based

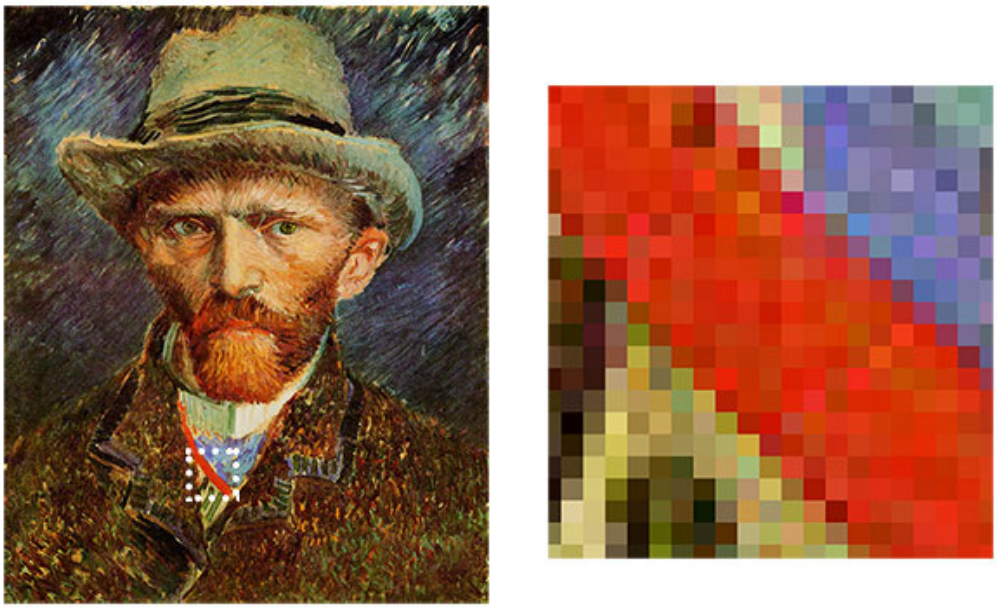


Figure 9: Art image "Self-Portrait With Grey Felt Hat", Vincent van Gogh, 1886/1887 with scaled detail (marked with white dotted square). The scaling reveals a great variety on red colour values corresponding to a perceptually even and similarly red coloured area on the original image.

algorithms partition graphs of all pixels using contour and texture cues, gradient-based algorithms analyse directional variations seeking for maxima in density functions. And distance based algorithms produce superpixels using clustering algorithms that use distances of data points in geometry and colour as segmentation strategy. All three approaches have been proved to produce segmentations that preserve most of the boundaries in the original image [LM01]. These characteristics of superpixel segmentation algorithms let them fit well to the task of getting colour related important regions which reflect the human perception of colour regions from discrete pixels. In the following sections three of the most commonly used algorithms are presented with Section 4.3 discussing modifications necessary for colour extracting in art images.

4.1.1 Graph-Based-Segmentation: Felzenswalb

Felzenszwalb and Huttenlocher published the *Efficient Graph-Based Image Segmentation* [FH04] algorithm developed using a graph-based algorithm. This sort of segmentation algorithms takes up images as a constellation of vertices, forming an undirected graph $G = (V, E)$. The image pixels corresponds to a set of vertices $v_i \in V$ with edges $(v_i, v_j) \in E$. Each edge has a corresponding weight $w(v_i, v_j)$ which describes a non-negative measure of the difference between neighbouring elements or pixels respectively. This measurement defines a distance measure on the colour space, called l_2-Norm. The segmentation algorithm's objective is to extract each connected component or sub-graph from the original Graph G

with every node within a sub-graph is related to each other, but dissimilar to each of the nodes in surrounding sub-graphs

Given a Graph $G = (V, E)$, a partition P of nodes V is searched, whose components $C \in P$ are interconnected. Additionally it is required that edges between two vertices in the same component shall have relatively low weight and and edges between vertices in different component relatively higher weight. To decide on the weight between two components we first have to evaluate the weights of edges within these components. This is done by determining the minimal spanning tree $MST(C, E)$ of a component. The *internal difference* between two components $C \subseteq V$ then is defined as the largest weight in $MST(C, E)$:
[FH04]

$$Int(C) = \max_{e \in MST(C,E)} w(e) \tag{10}$$

The difference between two components C_1 and C_2 consequently is defined as the lowest weight on edges between the components:
[FH04]

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w(v_i, v_j) \tag{11}$$

If there is no edge connecting C_1 and C_2 the difference is set to $Dif(C_1, C_2) = \infty$. The minimum weight is chosen, in favour of other possible quantiles like the median weight, to prevent the problem of finding a good segmentation NP-hard (see Appendix of [FH04]).

To check if there is evidence for a boundary between a pair of components the region comparison predicate evaluates the difference between the components $Dif(C_1, C_2)$ on being relative large to the internal difference within at least one of the components, $Int(C_1)$ and $Int(C_2)$. To control the degree to which the difference between components must be larger than the minimum difference, the pairwise comparison predicate is defined as,
[FH04]

$$D(C_1, C_2) = \begin{cases} true & \text{if } Dif(C_1, C_2) > MInt(C_1, C_2) \\ false & \text{otherwise} \end{cases} \tag{12}$$

with the minimum internal difference $MInt$ defined as,
[FH04]

$$MInt(C_1, C_2) = (min)(Int(C_1) + \tau(C_1), \quad Int(C_2) + \tau(C_2)) \tag{13}$$

The threshold function τ controls to which degree the difference between components is greater than internal differences, for being evidence that there is a boundary between them (D is true). It is based on the size of a component, respectively the amount of $|C|$, since $Int(C)$ isn't a good estimate of local characteristics. For example if in the extreme case $|C| = 0$ then $Int(C) = 0$. Therefore, the threshold function includes $|C|$ and k as constant parameter:
[FH04]

$$\tau(C) = k/|C| \tag{14}$$

with k putting weight on *MInt* (minimal internal difference) and Therefore, k causes a preference for larger components. Larger k factors allows the segmentation to larger components. But the k factor doesn't provide a measurement for minimum component size, since smaller components are still allowed, if there is a large difference between neighbouring components.

The segmentation algorithm them is implemented in the following scheme [FH04, p. 11]:

Input: graph $G = (V, E)$ with n vertices and m edges

Output: segmentation of V into components $S = (C_1, \ldots, C_2)$

1. Sort E into $\pi = (o_1, \ldots, o_m)$ by non-decreasing edge weight
2. Start with segmentation S^0 , with each vertex v_i is its own component
3. Repeat for $q = 1, \ldots, m$
   - Construct S^q with $o_q = (v_i, v_j)$
   - if v_i and v_j in different components C_1, C_2 and $D(C_1, C_2)$ is $true$
   - then merge the two components otherwise do nothing
4. Return $S = S^m$

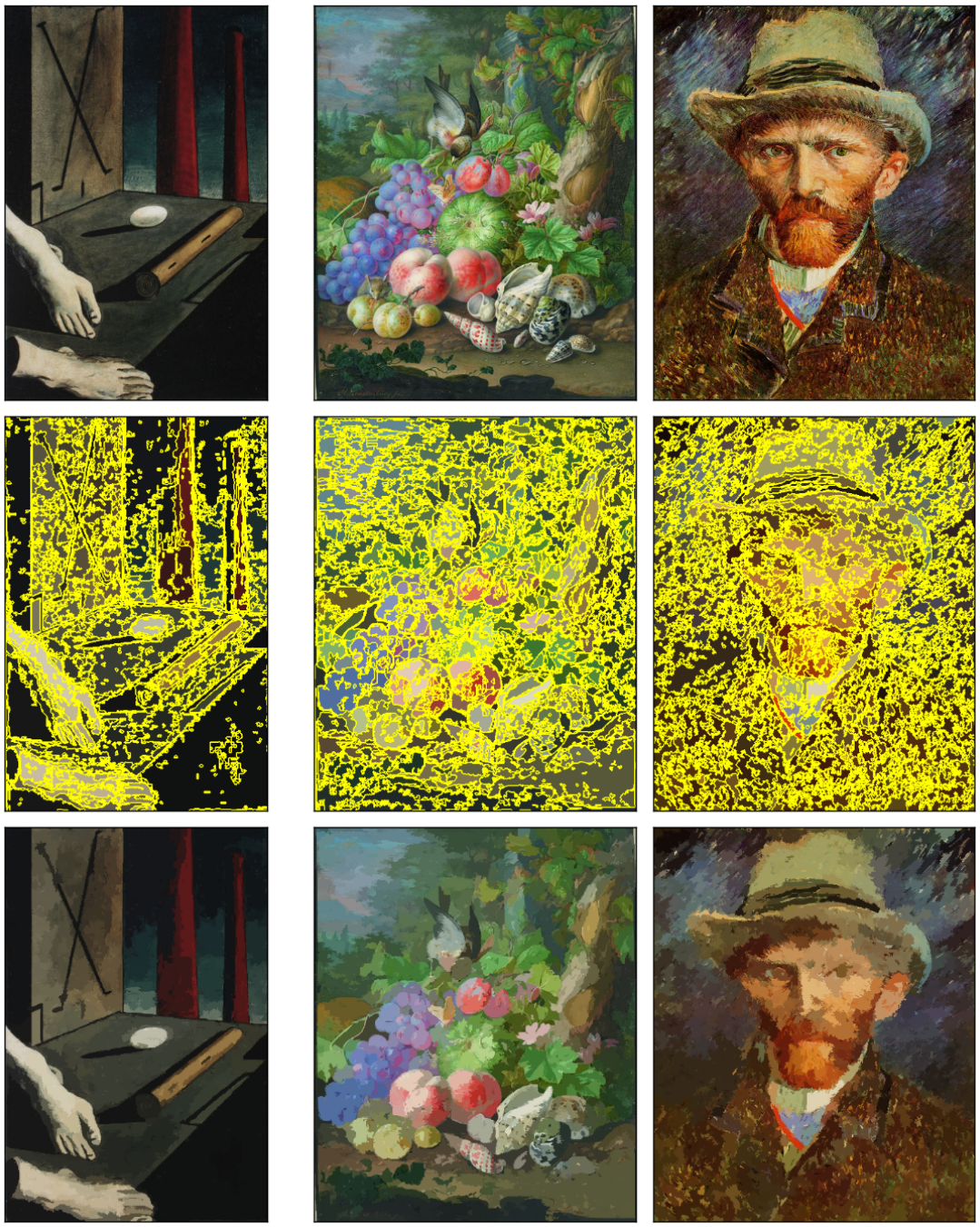


Figure 10: Felzenszwalb superpixel algorithm on three test images from ARTigo data set. Segments' colours are average of pixels in each segment. The images on second row show the segments' boundaries detected on the image.
Images (from left to right): Giorgio de Chirico: "Metaphysical Composition", 1914; Herman Henstenburgh: "Still Life of Fruits With Finch" ("Früchtestilleben mit Fink und maritimen Schneckenhäusern"), around 1700; Vincent van Gogh: "Self-Portrait With Grey Felt Hat", 1886/1887.

4.1.2 Gradient-Ascent-Segmentation: Quick Shift

Another segmentation algorithm that became popular recently in computer vision programming is described by [VS08] as *Quick Shift* segmentation in 2008 at the University of California. The *Quick Shift* segmentation algorithm belongs to gradient-ascent-based methods and its fundamental idea is based on the *Mean Shift* segmentation method, introduced by [Che95] in 1995 at the University of Cincinnati, Ohio. To understand the *Quick Shift* method, first *Mean Shift* will be introduced in the following paragraphs.

Mean Shift Mean Shift algorithm belongs to *mode seeking clustering algorithms*. *Mode* here denotes dense regions in the feature space [CM02] corresponding to local maxima of the probability density function. The *Mean Shift* method takes an image as a set of n data points $(x_1, \ldots, x_n) \in \mathbb{R}^d$. If a set of data points is part of a dense region, this set shall be summarised to a cluster.

Mean Shift builds up on the concept of kernel density estimation. This method takes data as product of probability distribution and tries to estimate the underlying distribution (i.e. probability density function). This approach is done by fist defining a kernel, describing a weighing function. The most popular used kernel is the Gaussian Kernel. By adding up all individual kernels this approach generates a density function or probability surface in $\mathbb{R}^d + 1$. The multivariate kernel density function is then obtained with a kernel $K(x)$ and window radius h: [Che95]

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{15}$$

and allows to estimate the distribution of data points dependent on kernel function K.

The *Mean Shift* method then exploits the kernel density estimation by presuming every point climbs up the hill to the nearest peak on the density surface. This "up-climbing" is achieved by iteratively *shifting* each point up the hill until it reaches the peak.

Given an approximation of density, data points can be clustered with the following procedure: Let $y_i(0)$ the mapping of a data point x_i on its associated point from the kernel density function $f(x)$ at time $t = 0$. The corresponding gradient $\nabla f(x_i)$ is determined and $y_i(0)$ is moved to $y_i(t+1)$ following the gradient (ascent of the hill). The algorithm terminates if the difference between $y_i(t)$ and $y_i(t+1)$ falls below a given threshold value for every data point in its cluster. Finally the data points are labelled by cluster centroids forming a labelled image mask (the segmentation itself does not prescribe the resulting segment colour value). Figure 11 illustrates the shifting procedure of data points on a density function (drawn as grey background contour lines).

Quick Shift *Quick Shift* [VS08] segmentation is based on *Mean Shift* segmentation method. Their main difference however lies in the way how both algorithms update the data points based on the gradients derived from density function.

Quick Shift starts similar to *Mean Shift* by estimating the density distribution of a given set of data point, i.e. pixel values of an image. It uses a more cost and effort reduced estimation

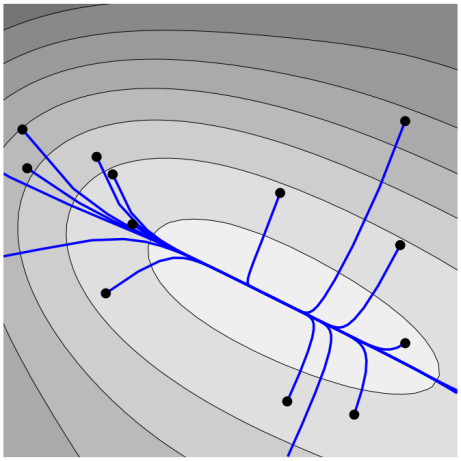


Figure 11: Mean Shift kernel density estimation [VS08, p. 2]

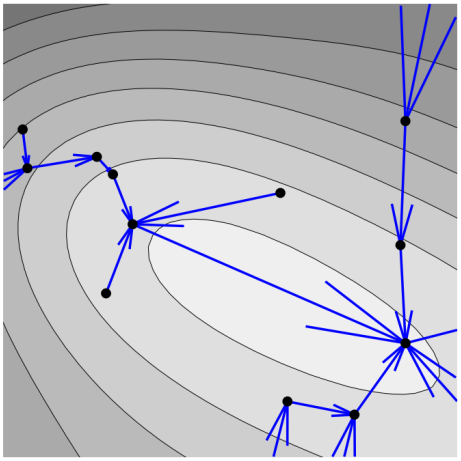


Figure 12: Quick Shift nearest neighbour estimation [VS08, p. 2]

of the probability density by using the density function on N given points [VS08]

$$P_i = \frac{1}{N} \sum_{j=1} N\phi(D_{ij}) \tag{16}$$

with D_{ij} denoting the euclidean Distance between p_i and p_j.

While the *Mean Shift* method the data points then were shifted based on the gradient, *Quick Shift* method shifts the points based on their closest neighbours. If the density P_j of a point x_j is greater than the density P_i of point x_i and x_i is nearest point to x_j, then x_i will be *shifted* towards x_j. This shifting procedure is formally described as index assignment [VS08]

$$y_i(1) = \underset{j:P_j>P_i}{\operatorname{argmin}} D_{ij} \tag{17}$$

Figure 12 illustrates the nearest neighbour shifting method of *Quick Shift* segmentation algorithm. The implementation of *Quick Shift* [5] algorithm in scikit-image library requires three parameters that influence the outcome of segmentation except for the image itself. The parameter τ limits the maximum size of a segment, the parameter σ specifies the window for density estimation and parameter r regulates the ratio on emphasis between geometry or colour of a segment and adjusts the distance function $D_{i,j}$. Higher values for r leads to greater differentiation on colour compared to geometry.

[5] http://scikit-image.org/docs/dev/api/skimage.segmentation.html#skimage.segmentation.Quick Shift

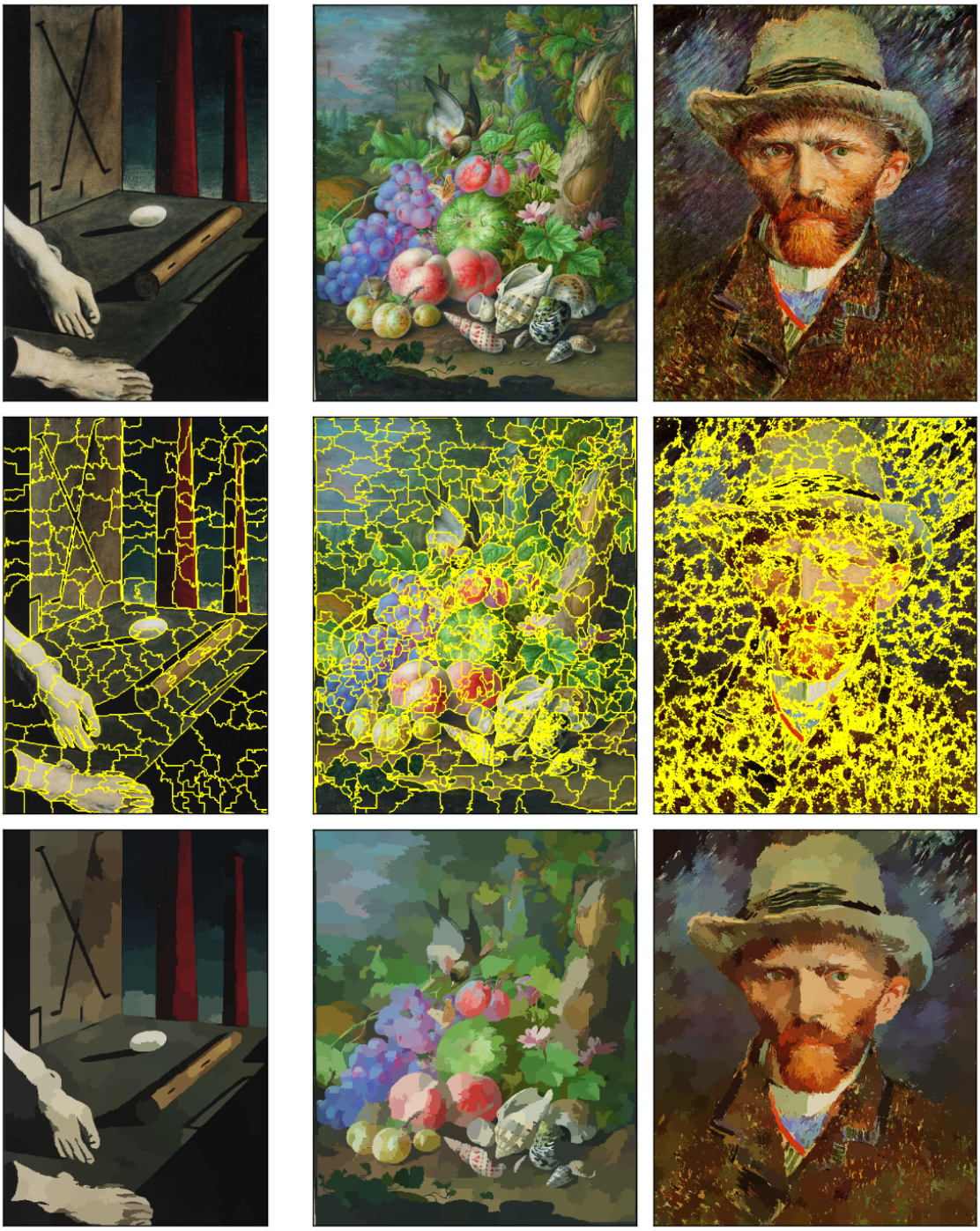


Figure 13: Quick shift superpixel algorithm on three test images from ARTigo data set. Segments' colours are average of pixels in each segment.
Images (from left to right): Giorgio de Chirico: "Metaphysical Composition", 1914; Herman Henstenburgh: "Still Life of Fruits With Finch" ("Früchtestilleben mit Fink und maritimen Schneckenhäusern"), around 1700, Vincent van Gogh: "Self-Portrait With Grey Felt Hat", 1886/1887.

4.1.3 Distance-Based Segmentation: SLIC

The third popular segmentation method widely used in computer vision application is the *simple linear iterative clustering* (SLIC) developed by [ASS+12] in 2010 at École Polytechnique Fédrale de Lausanne, Switzerland. *SLIC* segmentation algorithms belongs to the class of distance-based segmentation algorithms and produces superpixels using distance measurements on both, geometry and colour aspects of images.

SLIC segmentation algorithm is based on a distance measurement method, that implements a two folded comparison proceeding using colour similarity and proximity in the image plane. This distance measuring is therefore undertaken in a five-dimensional space containing three dimensions of colour and two dimensions for pixel position. While the *SLIC* segmentation uses CIEL*a*b* colour space to gain perceptually uniform and linear colour distances, the spatial distance in the xy plane of an image is obviously dependent on the image's size. Therefore, a simple five-dimensional euclidean distance measurement isn't possible and *(SLIC)* introduces a new distance measure D_s distinguishing colour and spatial distance.
[ASS+12]

$$\begin{aligned} d_{lab} &= \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \\ d_{xy} &= \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \\ D_s &= d_{lab} + \frac{m}{S} d_{xy} \end{aligned} \tag{18}$$

where D_s is the sum of the lab distances and the xy plane distance normalised by the grid interval S, which is defined as $S = \sqrt{\frac{N}{K}}$ with N denotes the amount of pixels and K the desired number of approximately equally-sized superpixels. The parameter $m \in (0 \dots 20)$ allows to control the compactness of the superpixels, greater values of m emphasise spatial proximity and Therefore, produce more compact clusters.

The *SLIC* segmentation algorithm generates superpixels by resolving a clustering problem introduces by the five-dimensional data of an image using the distance measure discussed above. Given an Image I and the desired amount of K segments, *SLIC* starts by sampling the image in K regularly spaced clusters. The centres of these clusters are then moved to the lowest gradient position in a 3×3 neighbourhood. With that step the chances of choosing a noisy pixel is reduced and placement at an edge is avoided. The image gradients are computed as:
[ASS+12]

$$G(x, y) = ||I(x + 1, y) - I(x - 1, y)||^2 + ||I(x, y + 1) - I(x, y - 1)||^2 \tag{19}$$

where $I(x, y)$ is the CIEL*a*b* vector at position xy and $||.||$ is the l_2 norm. This gradient definition takes into account both colour and intensity information. Then each pixel is associated to its nearest cluster centre. After all pixels are associated, a new centre is computed by averaging all five-dimensional vectors of pixels belonging to the centre Iteratively the process of pixel to centre association and centre re-computation is repeated until conver-

gence is reached. At a last step in the algorithm connectivity, which is not enforced before, is realised by assigning disjoint segments with labels of the nearest neighbouring cluster.

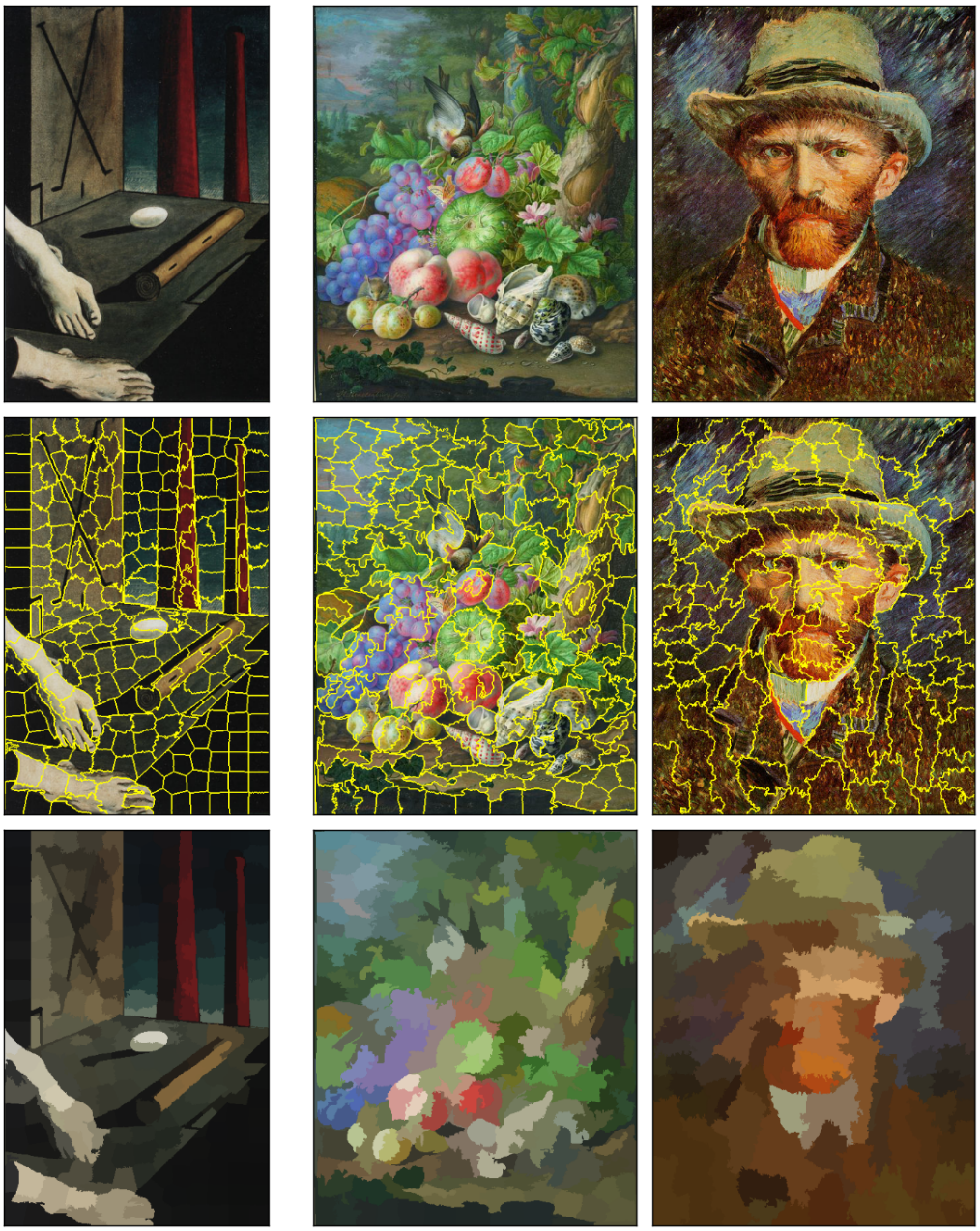


Figure 14: SLIC superpixel algorithm on three test images from ARTigo data set. Segments' colours are average of the pixels in each segment. Images (from left to right): Giorgio de Chirico: "Metaphysical Composition", 1914; Herman Henstenburgh: "Still Life of Fruits With Finch" ("Früchtestilleben mit Fink und maritimen Schneckenhäusern"), around 1700, Vincent van Gogh: "Self-Portrait With Grey Felt Hat", 1886/1887.

4.2 Colour Quantisation

Next to super pixel segmentation, the task of reducing colour variety to a distinct set of colours was primarily done with *colour quantisation* methods. These methods were historically particular relevant when computer displays and graphic processing units weren't capable enough to process the roughly 16 Million colours the (s)RGB specification allows. Therefore, algorithms were needed to reduce the amount of colours on an image to a predefined quantity. Today these colour quantisation algorithms are still used to generate or convert image formats based on indexed colours like the 8-bit GIF [Inc90] and PNG-8 [Duc03] formats, which show limited palettes of up to 256 colours. Since even the cheapest display hardware today is capable of showing at least 24-bit colour-depth, the particular need for colour quantisation for this purpose isn't relevant any more.

A very fast but unsuitable approach of colour quantisation is *scalar quantisation*. [BB16, p. 329f] Scalar quantisation performs a simple linear quantisation $fc_i \rightarrow c'_i$ is applied to each of the colour components c_i in the range $[0, \ldots, m-1]$ to convert them to the new range $[0, \ldots, n-1]$:

$$f : \lfloor c_i \cdot \frac{n}{m} \rfloor \rightarrow c'_i \tag{20}$$

This method multiplies each component with $\frac{n}{m}$, which is equivalent to ignoring a certain amount of lower bits. Therefore, this approach isn't suitable for the tasks of this project, since scalar quantisations do not take into account the distribution of colours in the original image. This technique provides an optimal solution only if image colours are *uniformly* distributed.

Instead of focussing on individual components of colours, the following techniques discussed are treating individual colour vectors separately and are called *vector quantisation* [BB16, p. 330]

4.2.1 Median Cut

The *median cut* algorithm is a widely implemented method, including in popular image processing libraries like ImageJ (Java)[6] and PIL (Python)[7]. This method first computes a colour histogram on the image colours, usually using a reduced number of histogram cells for efficiency. Given the initial histogram volume, it is split along one of the axis so that the two resulting boxes each contain half of the initial volume. This step is recursively repeated by choosing the largest of all sub-boxes and splitting it along its longest axis. As a result, the recursive splitting procedure produces a partitioning of the colour space in a set of disjoint boxes with each box containing an equal amount of colours. Finally a representative vector for all the colour vectors in each box is calculated, usually by computing the median vector. Figure 15 illustrates the recursive splitting process.

[6] https://imagej.nih.gov/ij/
[7] http://www.pythonware.com/products/pil/

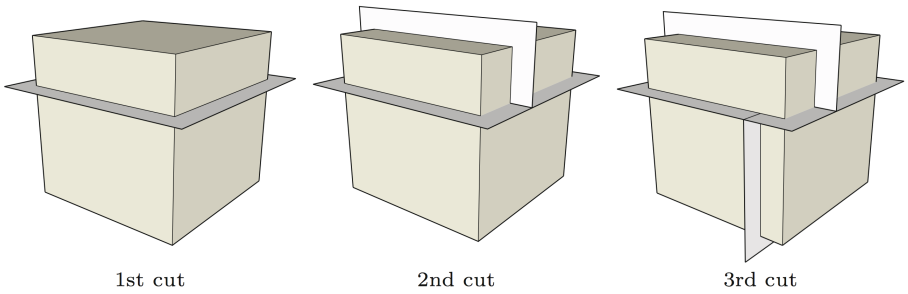


Figure 15: median cut box splitting [BB16, p. 332]

A noticeable characteristic of this technique is that colour regions of high densities are split into smaller cells, with regions of lower density are summarized to larger cells. In consequence *median cut* reduces the quantisation error on these highly represented colours, with only sporadic colour occurrences tend to be averaged.

4.2.2 Octree

Octree [BB16, p. 333] is a colour quantisation method that similarly to *median-cut* partitions the 3-dimensional colour space into cells of varying size. In *octree* algorithm the partitioning is represented by a hierarchical tree structure with each node linking to maximum 8 sub-nodes (octree). Therefore, each node represents a sub-range of the colour space and each subset is reduced to a single representative colour vector. This representative colour is stored with corresponding pixel counts in the leaf nodes of the tree.

The algorithm starts by getting the maximum number of colours as quantisation goal K. The eight-node-spanning tree, which is initially empty, is then filled up sequentially with colours from the original image. Given a RGB colour $C_i = (R_i, G_i, B_i)$ its binary representation of each component R_i, G_i, B_i is written in a table with columns indicating the resulting levels and each colour channel on a single row. Then the octree is followed through a path determined by each 3 bits of every level. Figure 16 shows the colour $(90, 113, 157)$ having $001_2 = 1_{10}$ on first bit level and Therefore, the chosen node on level 1 is 1 and $110_2 = 6_{10}$ for bit-level 2 with node 6 on level 2 etc. [BB16, p. 331]

Since *octree* colour quantisation is limited to a predefined amount of colours K, a reduction procedure is necessary. If a colour shall be inserted into the tree and the amount of already inserted colours is equal to K, then colours on the maximum tree level are merged bz adding their colour values and pixel counts. As soon as all pixel values are processed and the tree is reduced to K maximum leafs the colour palette can be extracted and pixels on the original image are assigned their most similar colour by tracing paths in the octree. A major advantage of *octree* quantisation is its efficiency regarding memory and computation steps, since the number of colour nodes is limited from the start to K and the maximum number of comparisons is also limited to 8.

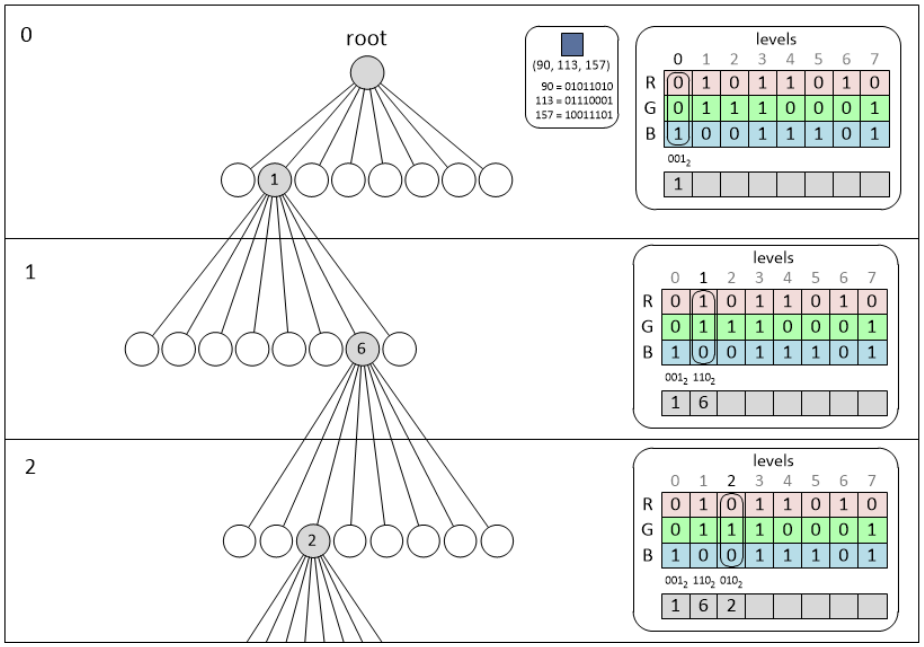


Figure 16: Octree tree building [Del]

4.2.3 K-Means

The third quantisation algorithm to be discussed is the *K-means* algorithm, which follows a different rather statistical approach compared to the previously discussed algorithms *median-cut* and *octree*. *K-means* clustering has already been mentioned in Section 4.1.3 discussing *SLIC* segmentation. In the latter this clustering technique was implemented on a 5-dimensional, representing chromatic *and* spatial values. *K-means* clustering is also used as a non-spatial colour quantisation technique clustering a 3-dimensional colour space [Cel11].

Similarly to *octree* and *median-cut*, *K-means* requires a pre-given amount of clusters K to be calculated. *K-means* [Pie] splits the colour-space into k clusters with each cluster being parametrized by its *mean vector* m^k. Since *K-means* is based on distance measurement between data and cluster centroid vectors, a custom distance function $d : x \times m \to \mathbb{R}$ with mean μ_C can be used. Given a set of colour values $x_p \in \{x_1, \ldots, x_m\}$ the algorithm produces k centroids $m_1, \ldots, m_k$ as follows [Pie]:

1. Initialize random cluster centroids $m_1, \ldots, m_k$
2. Repeat until convergence:
   - Assigment step: assign each x_p to the closest cluster m_i for $1 \leq i \leq k$

$$C_i := \{P \in I | d(x_p, m_i) = \min_{1 \leq j \leq k} d(x_p, m_j)\} \tag{21}$$

   - Update step: for $1 \leq i \leq k$

$$m_i := \mu_{C_i} \tag{22}$$

In Figure 20 a comparison of quantisation methods applied on unsegmented art image is shown with the resulting 3-dimensional colour histograms.

4.3 Modulation and Implementation of a Colour Segmentation Algorithm for ARTigo Image Pool

Extracting representative and outstanding colours of an art image requires an extracting method that fulfils the following needs. First, the extracted colours are not distorted in their appearance but seem to be taken out of the image. Second, tiny details can bear significant colours, therefore, the spatial distribution of colour has to be incorporated. Third, the amount of extracted colours is at a reasonable extend and similar colours are passed over in favour of the more prominent ones. And fourth, the method must be computationally efficient enough.

The two discussed approaches segmentation and quantisation each applied alone does not achieve results fulfilling all of the above criteria. Segmentation algorithms incorporate spatial and chromatic aspects of images and deliver cautious segmentation. But the amount of segments produced has been shown far to big, if segmentation is adjusted to be aware of important details. The average amount of segments varies between around 2000 to 4000 depending on each image's unique colour distribution. Colour quantisation algorithms in contrast indicate good applicability in reducing colour palettes to predictable extends, but seem to fall short in colour consistency when segmenting raw image colour material. Figure 20 shows the visible quantisation errors of these methods. Colours often seem to loose brilliance and distinctiveness through averaging. The colour extraction process introduces in this thesis combines the two approaches by segmenting images in a first step and quantise the segmentation result in a following step.

Segmentation method selection For implementation of colour extraction one of the segmentation methods have been chosen, providing most accurate and reliable segmentation results. *SLIC* segmentation provided usable results, but fell short in segmenting larger coherent areas with uniform colours, as it produces smaller and quadratic segments, where large segments would be expected. Its major disadvantage of requiring a pre-set number of segments as processing goal, finally disqualified *SLIC* as appropriate method for initial chromatic and spatial segmentation, with the great variety of images types in ARTigo image data set prohibiting any assumptions on expected segmentation result.

However the choice between remaining segmentation methods has proved to be less definitive. Even though *Felzenszwalb* segmentation is less performant in computation time, the algorithm accurately traces colour implicated edges and generates sound segmentation results. *Felzenszwalb* segmentation however entails many uncomfortable disadvantages. First, it has been shown highly dependent on pre-adjustments of available parameters. The algorithm allows to set several parameters controlling maximum and minimum component size and width of Gaussian kernel. A stable adjustment that fits to the image variety wasn't found. Figure 17 illustrates the difficulty of a proper adjustment for one image on another. The first image and its segmentation shown on the column, produces good results, while the same input parameters (`scale`, `sigma` and `min_size`) do not fit to the images on sec-

ond and third column. The finding of proper input parameter adjustments seem to require monitoring the segmentation of individual images or a classification of visually similar images. Second, *Felzenszwalb* in comparison to *Quick Shift* showed worse detection of small regions with important colours. Figure 18 shows a comparison of segmentation algorithms for selected images showing tiny but critical colour features.

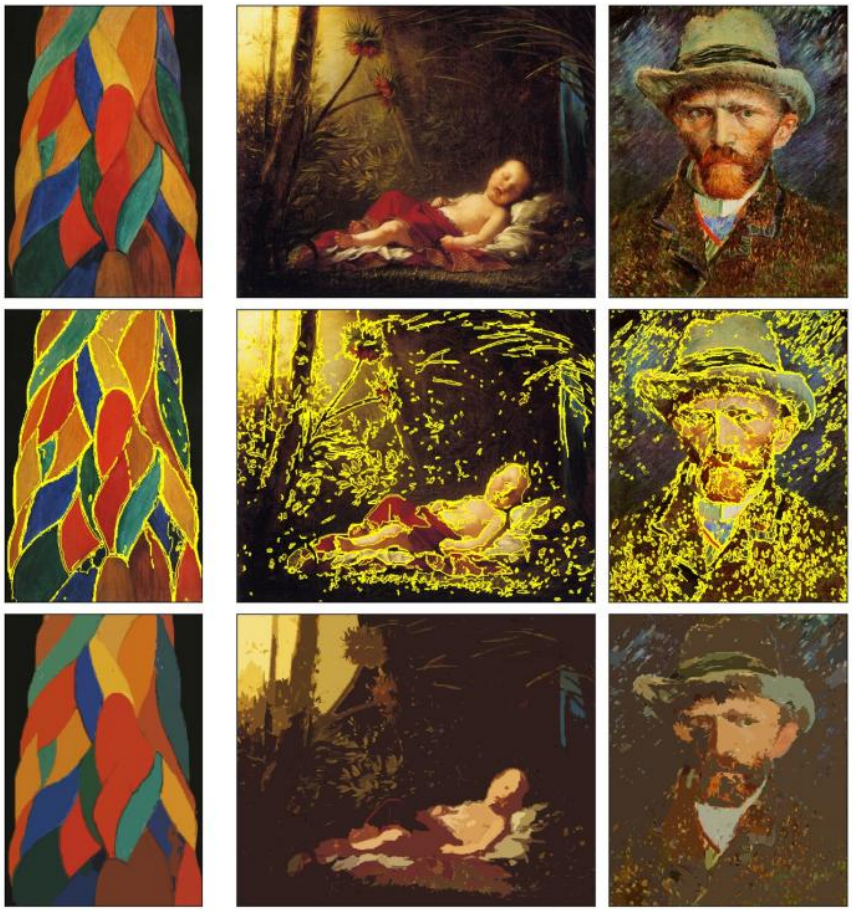


Figure 17: Comparison of three images segmented with Felzenszwalb algorithm with equal input parameters (`scale` = 400, `sigma` = 1 and `min_size` = 10) providing acceptable results for left image while unsatisfactorly segmenting second and third images. This illustrates the difficulty of adjusting *Felzenszwalb* input parameters. Images: Michail Matjuschin: "Movement in Space", 1918; Pierre-Paul Prudhon: "The King of Rome, Sleeping", 1811; Vincent van Gogh: "Self-Portrait With Grey Felt Hat", 1886/1887

Quick Shift was chosen as segmentation algorithm because it was possible to adjust its parameters τ, σ and r to fit the variety of images in ARTigo data set. Quick Shift also delivered best results on images with small details showing important colours (see Figure 18). As discussed in Section 4.1.2 the algorithm is able to recognise colour densities in spatial nearest neighbourhood and Therefore, allows to retain even small but dense regions in the segmentation process. Furthermore *Quick Shift* offered reasonable computing time (about 3 times faster than *Felzenszwalb* and about 2 times slower than *SLIC*). Providing the best set

of suitable segmentation results, reasonable computing time and wide applicability, *Quick Shift* has been implemented as first segmentation step.

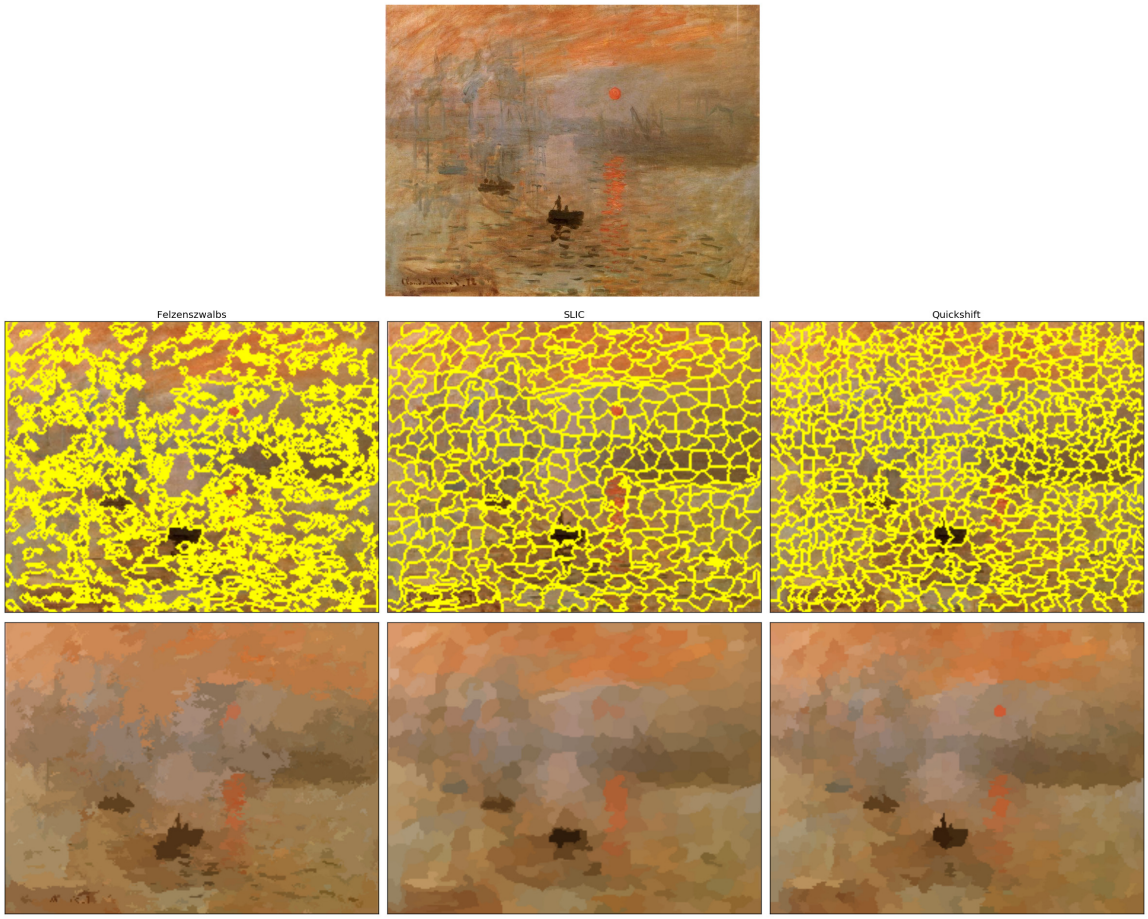


Figure 18: Comparison of image segmentation on an image showing tiny but important colour details. The sun in the upper-right area shows outstanding saturation and colour contrast and is important to capture as distinct colour segment. *Quick Shift* segmentation shows best and most reliable detection of contrasting areas. The algorithms produced around 3000 segments showing around 400 unique and apparently partially similar colours. Image: Claude Monet, "Impression. Rising Sun" ("Impression. Soleil levant"), 1872.

Image Segmentation Expansion The segmentation algorithms discussed in Section 4.1 output a labelled representation of images. To retrieve evenly coloured segments, like seen in third row of the related figures, the set of colour values in each region still has to be further processed. In this step the labelled image is converted into a composition of colour regions by ether choosing the average of colour values, the mode (i.e. the most frequent value) of colour value frequency or the weighted mode. The latter takes n most frequent colours and computes a weighted average on this n-mode-selection. As expected the averaging of pixels delivers the worst outcome and most distorted colours, since the average method includes every noise colour value equally. Choosing mode as a strategy to retrieve the most representative colour bears risk of overrating tiny spots and impurities. This effect can be

seen on Figure 19, where the mode strategy lifts out purple areas erroneously. Therefore, a new strategy of n-weighted mode was developed, consolidating the effectiveness of mode and consistency of average method. From the set of colour values, the n most ($n = 3$ suggested) frequent colours are selected and the weighted average is computed based on frequency. The `scikit-image` library didn't offered a similar method like n-weighted mode, therefore it was newly implemented. The results and comparison can bee seen in Figure 19.

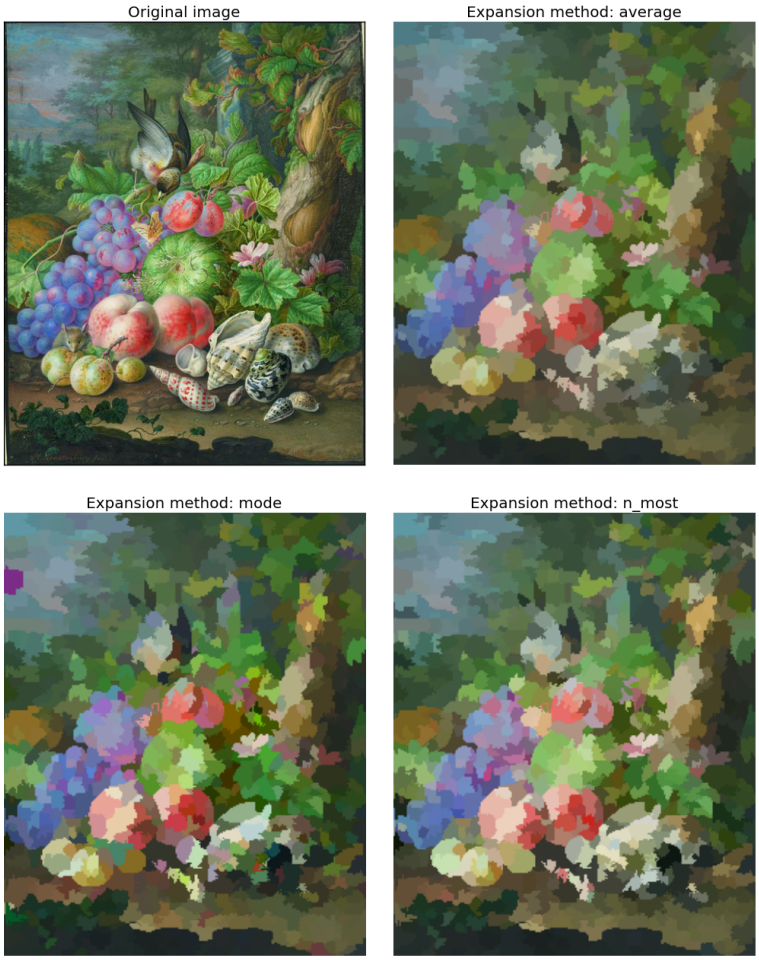


Figure 19: Quick Shift segmentation with label expansion methods *average*, *mode* and *n_most*. As expected the *average* method produces less brilliant and consitent colours, while *mode* method can overrate colours in regions (visible in the upper-left corner). The *N-most* method produces best results in colour consistency and overall appearance
Image: Herman Henstenburgh: "Still Life of Fruits With Finch" ("Früchtestilleben mit Fink und maritimen Schneckenhäusern"), around 1700.

Quantisation method selection As Figure 18 shows, the amount of image segments at this stage is still far to big to directly select the most important and significant colours and to present them in a colour palette. Therefore, a second quantisation step is required that reduces the set of colours to a reasonable amount. While the segmentation step incorpo-

rated the spatial distribution and composition of colours. This aspect isn't required any more and quantisation methods are therefore, applicable. As seen in Figure 20 the quantisation algorithms tend to produce unsatisfactory segmentation by ignoring the spatial colour distribution but they are effective in reducing the palette of pre-segmented images.

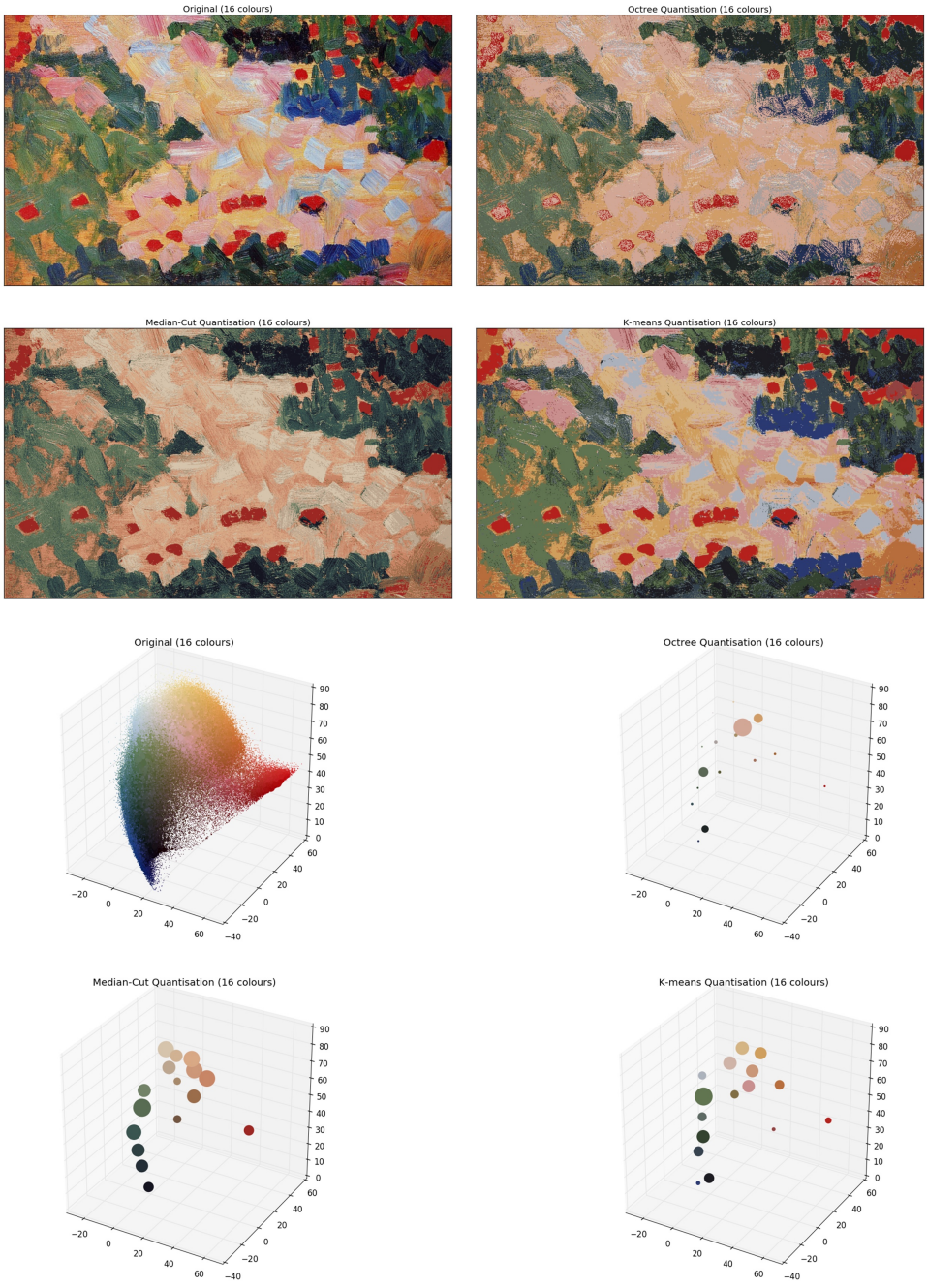


Figure 20: Comparison of quantisation methods *octree*, *median-cut* and *K-means*. The upper four images show the quantisation result, the bottom four images corresponding 3d-histograms. Image: Paul Signac: "Study for Times of Harmony. Red Poppy", 1894.

The selection of best fitting quantisation algorithm was dominated by a particular feature *K-means* quantisation offers exclusively. As discussed in Section 4.2.3 *K-means* calculates colour distances using a distance function $d(x, y)$. This unique feature allows to integrate the CIEL*a*b* colour distance function $\Delta E*_{ab}$ (see 3.1.1.3) as distance measurement and thereby provides colour quantisation based on perceptional rather than numerical (in median-cut and octree) similarities of colours. The *K-means* quantisation consequently produces perceptionally more plausible results, as shown in Figure fig:quantisation. The appropriate adjustment of K as the limiting factor for *K-means* on how many cluster centroids are to be extracted however is still to be defined and will be discussed in the following section on colour composition theory.

5 Extraction of Image Specific Colour Palette

Figure 21: Franz Marc: "The Little Blue Horses" ("Die kleinen blauen Pferde"), 1911.

The colour extraction algorithm presented in this thesis has been developed for extracting colours from artistic images. Consequently the applicable criteria evaluating the success of colour extraction have to be oriented on art specific conditions rather than putting on qualities of non-artistic images. For example, the application of normative conceptions like "realness" or "accuracy" turn out to be highly difficult and complex judgements regarding artworks. As the famous example of Franz Marc (Figure 21) explicitly shows, the *correctness* of colours and their grade of *accordance* with a stated "reality" cannot legitimately be judged on outward-positions on art. In particular, the judgement on Franz Marc's painted, blue horses, as not being painted using "accurate" or "real" colours would fail the specific examination artists undertake with the very concepts of *reality* or *accuracy* [Bir76]. Any evaluation on extracted colours from an art image has to understand at least the difficulties regarding accuracy and importance of colours. But it can also get aid from research to produce art-theory informed decisions as being suggested in this thesis and the following section. The rather cautious approach on carefully segmenting images incorporating their chromatic as well as spatial features, discussed in the previous sections, were also obligated to this idea of informed processing. To enhance the results of selecting significant and representative colours, colour composition theory on visual art will be consulted.

5.1 Colour Composition Theory and Semantics

Colour combinations induce perceptive effects and can enhance the experience of an artwork by using harmonies, disharmonies, brilliance, fades, etc. Artists exploited these effects in the composition of their paintings with most of the effects related to high-level chromatic patterns rather than to physical properties of single points of colours. [CDP99, p. 117] Starting from this or akin observation, multiple scholars obtained different colour composition theories. Probably the most famous examples were developed by Leonardo da Vinci, Isaac Newton, Johann W. von Goethe and Michel E. Chevreul. In addition to these separated theory approaches, Johannes Itten (1888–1967) expanded existing colour composition theories and defined patterns and properties through his research on art and colour composition theory [ISVD12, p. 157]. Itten's approach is chosen in the following discussion

as leading theory, since he successfully combined physical observation with sophisticated examination of art theory. It should be mentioned that Johannes Itten misused his colour harmony concepts sanctifying racial and anti-semitic thought in the 1940s [Wil16]. Nevertheless Itten is today still acknowledged as a renowned colour theorist and respected Bauhaus representative, despite his political attitude.

Figure 22: Itten colour circle with primary colours *red*, *yellow* and *blue* in the inner triangle, corresponding complementary colours *orange*, *purple* and *green* as compositions of primary colours and twelve tertiary colours on outer ring

In his later work "The Art of Colour" [Itt74] Johannes Itten formulated a comprehensive theory on the usage of colour in art and about its inducing semantics. Itten characterizes colours according to *hue*, *luminance* and *saturation* and identifies twelve hues as as fundamental colours, illustrated in Figure 22 on the outer colour circle. These fundamental hues are constructed from three "primary" colours *yellow*, *red*, *blue*. Further three "complementary" colours *orange*, *green*, *purple* and six "tertiary" colours are derived by combining primary and complementary colours. Itten further differentiated five levels of luminance and three levels of saturation, defining 180 distinct colours altogether. While he only specified the abstracted colour circle arrangement (Figure 22), the painter and art scientist Philipp Otto Runge arranged the whole set of Itten's colours to a conventional chromatic sphere — the Itten-Runge sphere (Figure 23) [SGD03]. It shows primary colours along a equatorial ring with varying luminance along meridians and increasing saturation to the outer layers.

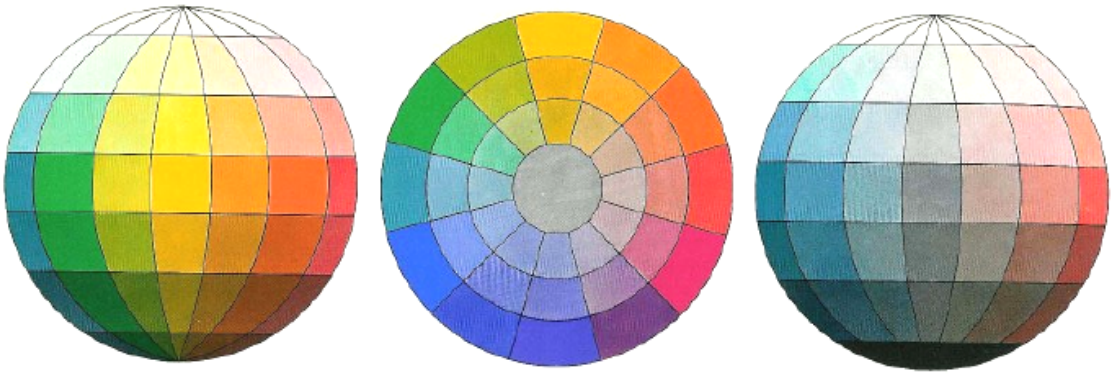


Figure 23: Itten-Runge sphere allocating primary colours and shades to spherical order with primary colours on equatorial ring and varying luminance along meridians, increasing saturation to outer layers of sphere [SGD03]

Johannes Itten's main result of colour analysis is his hue contrasting system. He defines twelve contrasts and other variations with respect to the intensity of the respective hue. These contrasts are effected by deliberately juxtaposing colours with great or even very low distance in the colour model. He defines the following types of contrasts: *contrast of hue*, *light-dark contrast*, *cold-warm contrast*, *complementary contrast*, *simultaneous contrast*, *contrast of saturation*, *contrast of extension*. [Itt74] In the Itten-Runge sphere, the perceptually contrasting colours are located on opposite coordinates.

5.2 Implementation of Colour Semantics and Colour Palette Extraction

The Itten-Runge sphere and Itten's geometrical assembly of colours in opposing primary colours shows much resemblance to HSV and CIEL*a*b* colour space. These colour models organise colours along spherical volumes and place contrasting colours on opposing locations (see Section 3.1.1.3 and 3.1.1.2). Putting Itten's colour composition theory together with the technique of representing colours in the perceptually uniform CIEL*a*b* colour model, we have now obtained a substantiated framework for semantically informed selecting of significant colours. Following the interpretation from Corridoni, Del Bimbo and Pala [CDP99], Ittens's semantically enriched colour contrast theory can be implemented the following way: Colours are arranged in a spherical 3-dimensional space with perceptional opposing colour being located on spational opposition. Following Itten, those colours, who are perceived as most relevant, will arrange at the outer layer of the colour sphere, forming mutual colour contrasts. To retrieve relevant colours based on their contrast qualities, we therefore, need to select those colours on the outer layer, which also span the greatest distance to opposing located colours.

To cover all of Itten's proposed colour contrasts, the quantisation step has to extract as least as 12 colours, Itten identified as the set on which colour contrasts are realised. To obtain even few more possible shades than Itten proposed and to include *white* and *black*, into the list of observed colour contrasts, the quantisation step clusters 16 colours. If – and this is expected to be the common case – not each of the possible colours are used by the artist, a second step step is required to sub-select the most contrasting ones.

Convex Hull The selection of opposed contrast colours is realised by calculating the convex hull on the set of colour centroids. The convex hull is a geometrical concept that defines an envelope enclosing a set of vertices. This approach is chosen, since the resulting colour vertices from quantisation process can be geometrically arranged in a 3-dimensional space spanned by the three axes from CIEL*a*b* colour space. To visualise the concept of convex hull one would imagine to span a tight rubber fabric around vertices in 3-dimensional space. Those vertices, which touch the fabric will be selected and form the convex hull [AJJ+11, p. 10]. Figure 24 shows a processed set of colour vertices and the corresponding convex hull by marking member vertices of the convex hull with black outer lines. The method of calculating convex hull was found as very efficient approach to select contrasting colours, but also to take into account the topological distribution of colours in the perceptional colour space. If a very dense distributed space of low contrasting colours is given, the convex hull is still creatable and even slight colour contrasts can be detected.

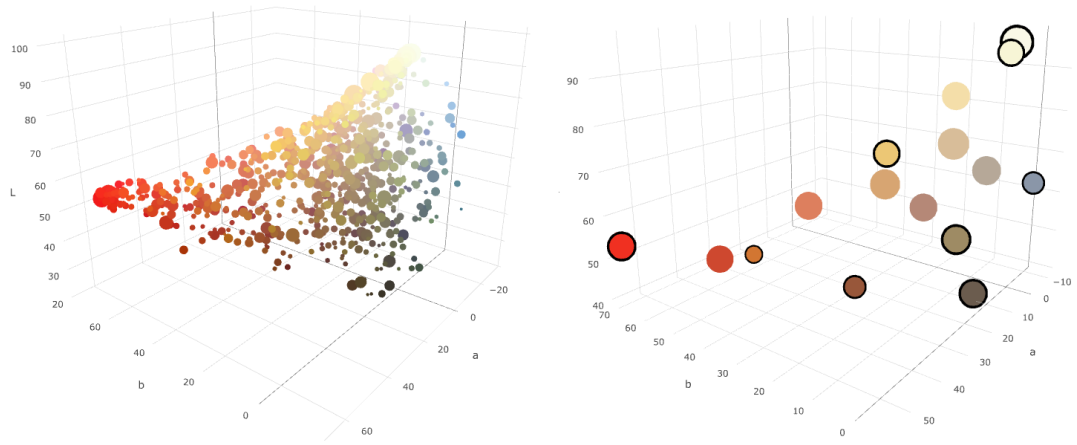


Figure 24: Segmented set of vertices (left) and convex hull of clustered centres (right). Vertices masked by convex hull are marked with strong black outline

A possible objection to this approach can be formulated by pointing out, that grey colours tend to be excluded by spanning the convex hull. Grey colour vertices are located at the centre of the colour sphere and therefore, seem to never sufficient the condition to be selected. This objective is true for those colour arrangements, where contrasting colours dominate the appearance of an image and follow the observation of Johannes Itten. He states that grey colours, as mixtures of primary colour for themselves do not participate in colour contrasts. But Itten also points out that contrast to grey colours is possible in principle, if a solid colour stands in singular opposition to the grey colour [Itt74, p. 37f]. This special case is covered using convex hull, since in this case a grey colour vertex forms a outer limit though located in the very centre of the colour space.

Gamma Correction The first result of the discussed colour selecting process based on Itten's contrast theory unfortunately showed not enough satisfactory results. The colours from clustering procession still lost too much of their brilliance in favour of locating a "mean"

colour as cluster centroid. To gain the colour perception even through clustering process, significant and strong colour values have to get higher emphasis. Remembering the colour representation model of HSV colour space (see Section 3.1.1.2) a gamma correction is applied to saturation (S) and luminance channel (V). Gamma correction is a common technique in digital image processing to compensate the usually linear intensity specification of displaying devices to logarithmic light intensity dispersion. In *gamma correction* the logarithmic-to-linear distortion is compensated by applying the inverted gamma function $f_\gamma^{-1}(a) = a^{\frac{1}{\gamma}}$. $\gamma \in \mathbb{R}$ is called the gamma value and produces "lighter" results for $\gamma < 1$ and "intensified" results for $\gamma > 1$ [BB16, p. 74f]

With gamma correction applied only to the saturation channel (S) and luminance channel (V) while leaving the hue component unaffected, this processing step merely affects the perceived intensity of colours maintaining their overall perceptional quality. Figure 25 shows the effect on this variant of gamma correction with slightly over-saturated appearing results, that will average to more "natural" appearance after the then followed clustering process.

Figure 25: Gamma correction function applied on segmented image. The figure shows the original image (1) with segmentation result (2), the gamma corrected segmentation (3) and the extracted colour palette (4). The slightly over-saturated appearance in (3) prevents faded colour clusters in following quantisation process and produces accordant extraction results (4) compared to original image (1).
Image: Jacques Louis David: "The Funeral of Marat in the Ancient Cordeliers' Church", 1783.

6 Automatic Colour Name Tagging

The colour extraction algorithm presented in this project has been developed to aid the image tagging game of ARTigo by automatically retrieving and validating relevant colour tags. These extracted colour tags can be integrated into the ARTigo game system as a first set of suggested colour terms. This far the algorithm discussed has produced a selection of colour values. The mapping from numerical colours onto a list of colour terms is discussed in the following section. First, the topic of basic colour terms is discussed. This subsection is followed by second, a evaluation of colour name sets and standards followed by third subsection explaining implementation in the colour extraction algorithm.

6.1 Basic Colour Terms

A pre-requirement of assessing the very possibility of tagging of visual perceived colour impressions with a set of terms, is stated by the determination of a reliable set of basic colour terms. A *set of basic colour terms* contains the information on colour impressions conventionally mapped on certain basic colour terms in consideration of cultural, linguistic and time-related factors. Thereby, a *basic colour term* is a colour word that is applicable to a wide class of objects (unlike *blonde*), it is monolexic (unlike *light blue*) and reliable, i.e., used by most native speakers [Har13]. Languages of modern industrial societies have usually developed a wide range of colour terms, but only a few of them meet the forgoing conditions.

In 1969 the American linguists and ethnologists B. Berlin and P. Kay published their influential study on *"Basic Colour Terms"* [KF09] exploring two questions: Is there a (ordered) set of basic colour terms used in most human languages? And are there similarities in the ways that different languages with the same number of basic colour terms carve out the colour space? They used standardised colour stimuli consisting of 329 colour chips and asked 25 monolingual speakers of 110 languages from 45 language families in total [Har13].

Berlin and Kay answered both questions in affirmative ways, but also explored, that languages have developed a different amount of terms. Additionally, an ordered progression in integrating colour terms was found. They proposed a set of basic colour terms (BCT) that vary slightly regarding different languages and also grouped these BCT in categories of sequentially developed terms. For English and most European languages the ordered terms are: *white, black, red, green, yellow, blue, brown, pink, purple, orange, grey* [KF09].

The studies of Berlin and Kay started a still ongoing debate on the linguistic and ethnologist aspects of their research.[HM97] For this and similar projects ([SS16, vdWS07, Lin13, MHS02]) the studies of Berlin and Kay justify the expectation that a decent set of colour terms are used to describe basic colour impression. Even though this assumption does not lower people's capability to understand a much larger set of colour terms in general. The BCT research merely claims that the BCT set meets strong requirements of constancy and verifiability.

6.2 Colour Name Sets and Standards

Human aspiration for categorizing elements on their sensory experience has equipped us with a keen capability to denominate visual perception with an almost unlimited set of colour names. Some colour names refer to a dye or object, like “mouse-grey” and “irish moss”, and from this perspective the amount of colour terms seems to be innumerable or even arbitrary. On the other hand, a lot of empirical linguistic and semantic research is done on the question, whether it is justified to obtain a more or less objective system of colour term classification [HS12].

Research on colour-naming represent either universalist or relativist approaches. Relativists claim, that colour naming practice varies from individual speech groups in idiosyncratic ways, criticising the very idea of developing a common and authoritative set of colour terms. Opposing universalists refer to studies like the colour survey done by Berlin and Kay to emphasise the idea of at least basic accordances across different language groups on colour naming practice. While this academic debate is still ongoing – especially in the linguistics – industry has developed many approaches and guidelines on authoritative colour naming classifications, following the urgent need to offer consistent and applicable colour denomination [Wyl92, p. 13f].

The volume of colours specified in these industrial colour term sets highly varies through the origin from specific areas of application. In computer programming certainly the most known colour term system is provided by the W3C: the “CSS Color Module Level 4” [8], referring to the latest specification for colour values within the Cascading Style Sheets language. This colour term model contains 141 standard colours like “black”, “white”, “red” and shades like “lemon chiffon” or “mint green”. The colour terms are listed with corresponding RGB colour value in 32 bit colour depth. Especially the figurative terms rise suspicion on it systematics and with this on its suitability as a basic colour term reference.

In fact, the CSS colour term list is a product of a multitude of supplementation and retention of colour terms, which can be traced back to the beginning of the 1980s. Originally the colour term list is a product of the X Window System, released in 1984 by MIT. But with many contributions, done by multiple authors adding colours calibrated to their own computer monitors (which in the 1980s and 90s still showed colour distortion), the colour term list became increasingly inconsistent. Nevertheless the colour term set was adopted by the W3C on the specification of CSS language to “codify current practices”, even though developers already adopted the usage of hexadecimal colour specification anyway. [Tve15]

A more promising approach is presented by the RAL classic colour system developed and maintained by the German non-profit RAL institute since 1927 [RAL16]. The RAL (abbreviation for Imperial Commission for Delivery Terms and Quality Assurance, German: Reichsausschuss für Lieferbedingungen) institute developed a colour matching systems defining authoritative colour specifications meeting the industrial needs to replace the former inconvenient practice of interchanging colour samples. RAL began specifying 40 basic colour terms in 1927 and further developed multiple colour classifications incorporating 2.328 colour terms today. With many classifications have been developed to need branch-specific colour specification, the RAL ‘classic’ colour set containing 213 colours defined by a CIEL*a*b*

[8] https://drafts.csswg.org/css-color/

triple value, a four digit *RAL colour number* and corresponding denomination in German, English, French, Spanish, Italian and Dutch language.[9]

The RAL classic colour labelling systems provides two major advantages highly relevant for this project: First, the list of colours and their denomination are released as the product of a still ongoing, sophisticated curation process, consulting industrial, cultural and material scientific perspectives. Second and partly as a consequence, the RAL colour system is structured hierachially defining basic colour classes (corresponding to primary colours). Therefore, it allows to obtain both, detailed colour denomination, as well as more abstract affiliation to basic colour categories. The RAL classic colour systems allows both classification modulation since colours are encoded with a four digit number. The first digit is a system code number (1: yellow, 2: orange, 3: red, 4: violet, 5: blue, 6: green, 7: grey, 8: brown and 9: white and black shades) and the remaining three digits are chosen sequentially. Hence RAL classic colour encoding also serves as a basic colour matching reference [RAL16].

Another, rather unusual approach to match visual colour impression on colour names, was presented by Randall Munroe, the author of popular XKCD web comics [MX10]. Munroe collected data through a advertised campaign on his popular XKCD comic website and gathered over 3.4 million responses. Results from his survey have been publicly available and already used in research ([HS12, Lin13, SS16]). In the survey participants were asked for basic demographic information as well as technical information like display type, temperature and gamma. Mainly participants were shown uniformed samples of colour in sRGB colour space and were asked to name them in free form. Munroe also implemented a per-user spam-score penalizing responses not used by anyone else and high variations in hues. Spam filtered data provides 3,252,134 colour-name pairs and 132,259 unique colour names [HS12]. Obviously the XKCD colour survey lacks on systematics. But as a recent and large-scaled survey, it could provide interesting overvalue to established and curated colour term systems.

The colour term mapping algorithm developed in this project in principle can be conducted with any colour term mapping set (see following Section 7.1 for details). But as stated before, the RAL colour term list is recommended and chosen as further implemented reference model, allowing to retrieve colour terms suitable to compare with data in the ARTigo tagging game (see Section 7.2 for further details).

[9] A comprehensive overview on RAL classic colour can be explored at http://www.ral-farben.de/inhalt/anwendung-hilfe/alle-ral-farbnamen/uebersicht-ral-classic-farben.html.

7 Implementation and Evaluation of Colour Tagging Algorithm

The semantic colour extraction and colour tagging process was implemented using the programming language Python 3.5 and tested with the art image dataset from the ARTigo art exploring project. In the following section the implementation of colour extraction and tagging algorithm is presented by referring to important implementation steps.

7.1 Implementation of Colour Extraction and Colour Term Mapping

The algorithm can be divided into two main application parts: First, the extraction of a semantically enriched colour palette for an individual art image, and second, the mapping of colour values on a given set of colour terms.

Colour Palette Extraction The algorithm was generally implemented as terminal application and requires a given path to a directory containing image files or a comma-separated-values file (CSV) with image paths, file names and ARTigo resource id, which refers to the organisation of the ARTigo database. Image paths and file names are then internally represented by Pandas DataFrames[10] (i.e., image-table), providing potential to execute next calculations row-wise on multiple threads and CPU-cores. Beside the main Pandas DataFrame, global DataFrames are initiated covering errors (e.g., file-not-found exceptions) and recording grey-scale images. These grey-scale images do not feature (s)RGB colour space, using merely a single dimension to represent pixel brightness, and are excluded from chromatic processing accordingly.

If possible, each row on the image-table is further processed on separate execution thread. The method `get_palette_from_image` in `aux_lib/auxlib.py` creates a set of colour values for a given image by conducting the following procedure:

1. Image load from file system using given path and conversion from RGB to CIEL*a*b* colour space (see Section 3.1.1.3).
2. Quick Shift segmentation (see Section 4.1.2) producing a labelled segmentation of the image using Python *scikit-image*[11] library for image processing.
3. Allocation of a representative colour for each segment using n-weighted mode selection strategy (discussed in Section 4.3)
4. Conversion to HSV colour space (see Section 3.1.1.2)
5. Gamma correction on saturation and luminance channels, keeping colour hue channel untreated (as described in Section 5.2)
6. K-Means colour segmentation expecting a maximum of 16 clustering centroids (see Section 4.2.3) using Python *scikit-learn*[12] machine learning library.
7. Selecting most contrasting colours based on convex hull masking, if more than 3 distinct clusters have been extracted (see Section 5.2).

[10] http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html
[11] http://scikit-image.org/
[12] http://scikit-learn.org/

id	artresource_id	frequency	L*	a*	b*
⋮	⋮	⋮	⋮	⋮	⋮
189	15	3893	96.13965860969792	-4.353553461019601	15.247059676125923
190	15	2723	61.804331283316905	-0.2924245431566802	-10.815814917017569
191	16	5160	50.49297369032935	66.36012682514458	54.62556547286384
192	16	7145	20.89180746231898	8.021432088461284	11.219523074764579
193	16	10384	87.33684703253	-3.227628206072107	76.48085339991536
194	16	1953	42.17285940162755	-23.60589657253709	35.905640492163876
195	17	4126	62.129332068358146	6.897007816221198	59.03181388487913
196	17	5269	81.77068702431649	-3.36410330775431	9.236762502271928
⋮	⋮	⋮	⋮	⋮	⋮

Table 1: `color_palette.csv` excerpt as example for output of first colour palette extraction step.

The results of the first palette extraction part is then saved to a CSV file containing unique colours in CIEL*a*b* components with colour frequency for each image. The amount of colour rows per image can vary between 1 and 16. Table 1 displays an excerpt of the produced `palette_images.csv` CSV file. Additionally a metrics table `metrics_images.csv` is exported including the convex hull volume and maximum colour distance for each image. This table allows to compare images on colourfulness, with highly colourful images showing greater values.

Colour Labelling The second part of the algorithm is devoted to mapping accurately extracted colour values from step 1 to a set of predefined reference colours with corresponding colour terms. This part takes the `palette_image` Pandas DataFrame and processes the palette colour to reference colour matching procedure, while duplicate hits have to be reduced to single occurrences.

The crucial part of this procedure lies in the distance calculation between the the colours. As already been discussed in Section 3.1.1.3 the CIEL*a*b* distance function $\Delta E*_{ab}$ provides a perceptually uniform distance measurement. Using CIEL*a*b* colour distance function, a distance matrix is calculated for each row in the colour palette DataFrame, before the minimum distanced reference colour is selected.

The RAL classic colour reference list also provides a reference to the respective *basic* colour for each colour, signifying membership to a basic colour set.

id	artresource_id	RAL colour_id
⋮	⋮	⋮
25897	101671	86
25898	101671	190
25899	101671	183
25900	101671	8
25901	2006886619	10
25902	2006886619	12
25904	2006886619	108
25907	2006886619	1
25909	2006886619	165
25911	2006793372	177
25912	2006793372	200
25913	2006793372	149
⋮	⋮	⋮

Table 2: `image_terms.csv` excerpt as example for image to RAL colour term matching

Table 2 shows an excerpt of the produced CSV table `image_terms.csv` with continuous *id*, *artresource index* and *RAL colour id* and Table 3 a short extraction from RAL basic colour list.

index	L*	a*	b*	german	english	french	...	basic_colour_index
76	44.82	29.08	-18.58	Signalviolett	Signal violet	Violet de sécurité	...	75
77	60.59	10.38	-2.88	Pastellviolett	Pastel violet	Violet pastel	...	75
78	50.39	48.95	-4.24	Telemagenta	Telemagenta	Telemagenta	...	78
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
103	60.5	-9.53	-17.38	Pastellblau	Pastel blue	Bleu pastel	...	96
104	35.93	-11.81	-16.28	Perlenzian	Pearl gentian blue	Gentiane nacré	...	96
105	16	7.84	-29.1	Perlnachtblau	Pearl night blue	Bleu nuit nacré	...	96

Table 3: `ral_colors.csv` excerpt from RAL classic colours list

Web Interface With the image segmentation and colour quantisation process being computed "silently" and with only less illustrative CSV tables are produced as output, a web-interface has also been developed to demonstrate key steps in the colour extracting algorithm. On that account a –json flag can be set, which controls the production of separated JSON representation of image segments and colour clusters as well as visual representation of extracted colours with their corresponding relative frequency. These JSON files can be loaded to the web-interface providing interactive 3-dimensional colour histograms, the original image with its segmentation outcome and the resulting colour palette ordered by colour hue angle. Figure 26 shows a screen-shot of a sample image. The selection of images shown on the web-interface is implemented by retrieving an URL-parameter `file`, expecting the art image resource id, which allows simple integration into ARTigo platform.

Figure 26: Web-Interface showing (from upper-left to lower right) the original image, its segmentation after gamma correction process, a 3-dimensional histogram of segmented image, a similar histogram of colour quantisation with convex hull marked with black outline and the circular hue-ordered extraction result.
Image: Paul Gauguin: “There is the Temple (Parahi te marae)”, 1892.

7.2 Evaluation

The ARTigo image colour extraction and labelling was done on 192.109 art images from the ARTigo database. After the first sorting step, separating 22,878 grey-scale images with only one channel and 528 images with errors on file load, the remaining 168,703 colour art images were further processed generating a colour palette for each image. In average the algorithm produced 3.83897 colour tags per image. As discussed in Section 5.2 a maximum of 16 colours could be extracted as colour palette, if all colours are masked by the convex hull and the colour-to-term matching process hits 16 unique reference colours. In total the algorithm produced 735,037 distinct colour tagging entries on the image dataset (i.e., distinct combinations of colour tag and art image).

For evaluating the accuracy of the presented algorithm compared with predicted colour labelling already done in ARTigo, the empirical dataset of image tagging in ARTigo has to be filtered the following way: First, the tagging results are reduced to those tags that imply colour information, and second, to reliable colour tags being confirmed by users at least n times.

The first task of sub-setting only chromatic tags was implemented with SQL query filtering tags compared with a list of alike colour terms. This list allowed versatile forming of colour tags in case, spelling and also composition (to include tags like "blackwhite" or "lightblue"). In next step this list was further filtered excluding semantical false-positives like "white wine" or "greyhound", i.e., terms that do not precisely imply colour information or have been detected because of showing coincidental permutation of letters only. Since this filtering step required human or human-alike understandings of semantics, the list was worked through manually in order to reduce false-positives to a minimum. As to this point the list of tagged images with colour tags still listed tagging that hasn't been verified by multiple users, the image colour tagging list was further reduced using colour tags being either $n = 5$ or $n = 2$ times verified. The following evaluation has been realised using the filtered German tagging set of ARTigo, since the English and French data sets provided too few validated colour tags covering total colour spectrum.

7.2.1 Results

Regarding colour tags validated at least 5 times, the algorithm tagged 13,310 images in accordance with data in ARTigo. Images in ARTigo were tagged 20,367 times, yielding an accuracy rate of 65.35%. Allowing tags, 2 times being validated at least, the algorithm tagged 69,363 images in accordance with ARTigo, where 118,810 colour-tagged images have been recorded, yielding an accuracy rate of 58.38%.

A detailed chart of image-tag dispersions ordered by colours from the algorithmic results as well as from ARTigo is presented Figure 27 (algorithmic) and Figure 28 (ARTigo). Figure 29 shows Distribution of ARTigo colour tagging in descending order. To evaluate the dependence of accuracy and colour Table 4 compares the frequency of each colour detected by the algorithm next to ARTigo and the colour dependent accuracy rate, which is computed from matching colour taggings on identical images. Therefore the accuracy rate indicates the chance of colour tagging being predicted by the algorithm.

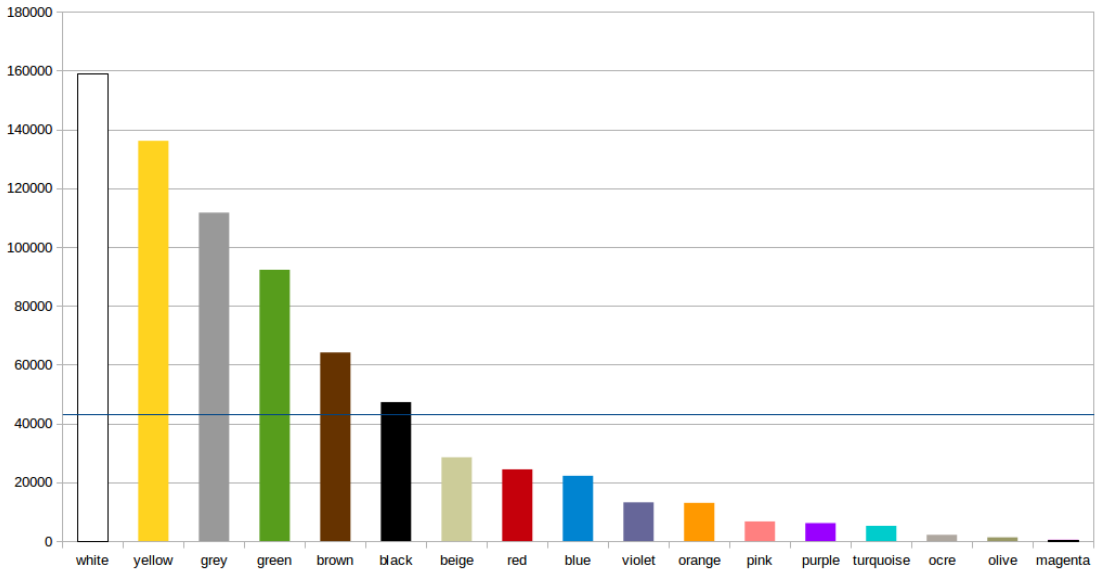


Figure 27: Distribution of colours, tagged on images in the algorithmic results. The blue horizontal line marks the medium of tag frequency. For a discussion on noticeable high quantity of "yellow" and "grey" see Section 7.2.2.

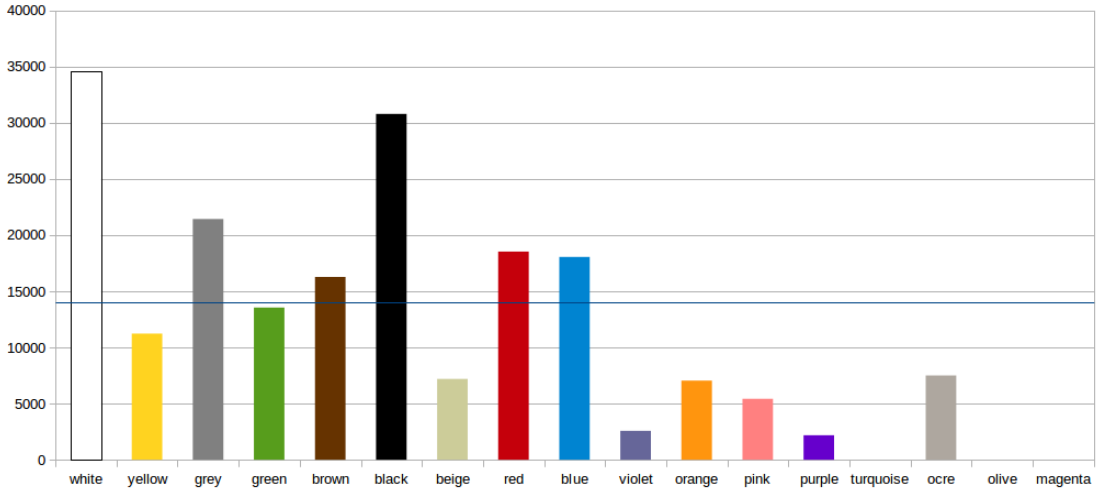


Figure 28: Distribution of colours, tagged on ARTigo images at least 2 times per image (e.g., 18,525 images have been tagged with 'red' twice at least). The blue horizontal line marks the medium. The colours' order corresponds to Figure 27 for comparison.

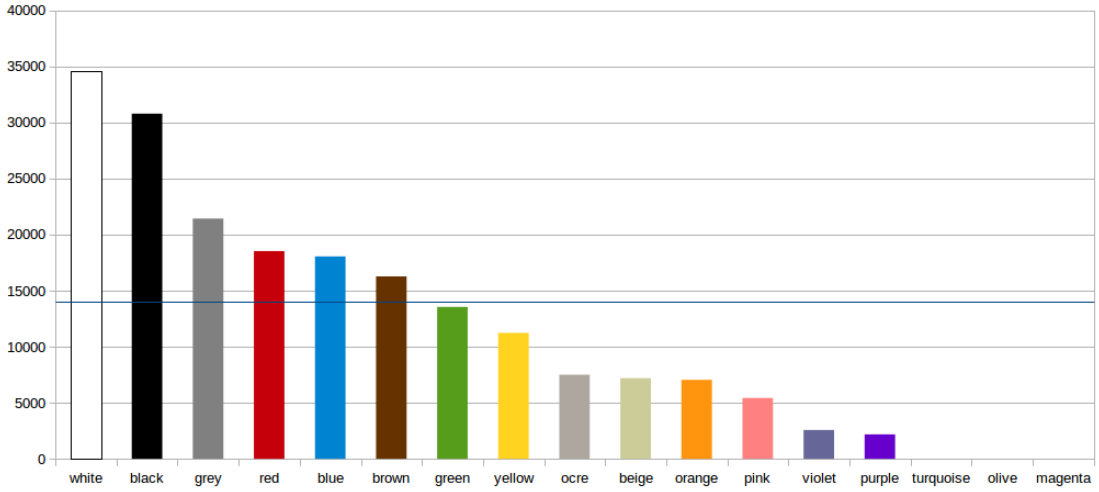


Figure 29: Distribution of colours, tagged on ARTigo images at least twice per image, in descending frequency order. The blue horizontal line marks the medium.

Colour	Algorithm	ARTigo	Accuracy
yellow	136,139	11,224	87.39 %
pink	6,677	5,423	23.27 %
beige	28,489	7,194	50.54 %
turquoise	5,162		
red	24,389	18,525	55.10 %
ocre	2,107	7,494	50.98 %
violet	13,177	2,575	60.23 %
green	92,273	13,546	67.52 %
blue	22,198	18,044	51.00 %
olive	1,260		
orange	12,988	7,044	57.85 %
black	47,258	30,782	40.47 %
grey	111,726	21,426	44.02 %
magenta	455		
brown	64,129	16,263	82.91 %
purple	6,103	2,179	23.01 %
white	159,069	34,587	72.38 %

Table 4: Frequency of images tagged per colour term (at least twice) in comparison of ARTigo and the algorithm's results. Accuracy column is showing accuracy-rate for each colour calculated by comparing matching term prediction and empirical term tagging.

7.2.2 Discussion of Results

The results of algorithmic colour tagging presented above show interesting results, since similarities and differences are both found comparing algorithmic colour tagging with empirical data from ARTigo image tagging game. The following section will briefly mention five major aspects, worth being discussed separately.

First, the average amount of colour tags attributed to an art image by the algorithm is about 4. While Itten does not precisely predict a number of important colours being used in colour composition of art images, the composition models he obtained from art images are limited to a maximum of 5 and are minimum of 2 colours [Itt74]. The results of colour extraction algorithm fits well into Itten's analysis that a alike distinct number of colours constitute a composition based on colour contrasts.

Second, the dataset from ARTigo shows a limited usage of expected basic colour terms. Taking into account the order of basic colour terms proposed by Berlin and Kay (see Section 6.1 the empirical data on colour tagging in ARTigo can confirm Berlin and Kay's basic colour order most widely. *Black*, *white*, *red*, *blue*, *brown*, *green* and *yellow* are obviously predominantly colours tagged and outreach the frequencies of *orange*, *purple* and other secondary developed colours clearly. Although, the image material in ARTigo does not, of course, provide similar standardised specifications as the colour sample set of Berlin and Key's study provided.

Third, the algorithm shows a striking emphasis on tagging the "yellow" colour tag. This effect can be traced back presumably from two reasons. One one hand, it is produced by a big share of drawings, architect's plans and books' pages, which often show yellowish appearance resulting from ageing process.[13] Even if *yellow* is a comprehensible reference to old paper appearance, a more proper describing term would refer to the material ("paper") or the very process of ageing ("yellowed"). But for a second reason the dominance of "yellow" can be traced back to photographic representation of gold material as frequent component of art works in the ARTigo dataset. The colour extracting algorithm is not yet capable of detecting chromatic gradients, as they occur when heavily reflecting materials like gold are photographed. On the contrary the choice on *Quick Shift* as a gradient-based segmenting method even increases the algorithmic affinity to unify such areas – a characteristic of *Quick Shift* algorithm that is used purposely to unify perceptually uniform colour areas, as discussed in Section 4.1.2. An increased analysis capable of detecting reflecting materials like gold and silver, does have to intervene at this very step of segmentation, when gradient information on the images can be still processed and evaluated.

Fourth, the comparison of Figure 27 and Figure 28 can tempt to conclude that the colour extraction algorithm failed to detect, e.g., "red" as a apparently highly relevant colour. But these two figures have to be compared with cautious on their different information scope. While ARTigo colour tagging gives insight to users' assessment of relevance on colours, the colour tagging algorithm in contrast evaluates the colour composition of each image. Therefore the algorithm could at its best only reproduce a small and distinct part of human

[13] English, German and French denote this procedure with verbs referring the yellow colour term — i.e., *yellowing* in English, *Vergilben* in German and *jaunissement* in French.

assessment on colour semantics, since the latter is influence and inspired by many more factors than just the composition of colours.

Finally as fifth conspicuous aspect, the results of automatic colour extraction show a higher frequency of "white" being tagged than expected. Detecting *whiteness* as colour in artworks revealed many problems, since the actual pixel values of areas perceived as "white" show less bright and pure characteristics than our perception purports. In consequence, the colour extracting algorithm shows difficulties in accurately distinguishing slightly grey and tinged white shades. The gamma correction process, discussed in Section 5.2, was therefore tuned to increase brightness of already pale areas to improve the unambiguity of pale white and light gray or tinted colours. Even with overall results in white being frequently tagged, for several art images, the presented algorithmic colour extraction still does not provide sufficient capability in detecting white as perceptional pure colour impression. And as with the case of gold and other reflecting materials, this limitation could be improved by further involvement of gradient analysis like suggested in [NpK04]. The high frequency of "grey" being tagged in contrast with ARTigo's empirical data, prove the need for further refinements. The low proportion of tagging "black" in comparison to ARTigo data could be attributed to grey-scale images which have been excluded in the colour extraction algorithm.

8 Conclusion and Future Work

This thesis presented an algorithmic colour extraction method with automatic colour labelling on art images, incorporating the specifics of art images by consulting colour composition from art theory and analysing idiosyncrasies of state-of-art image segmentation methods. From several segmentation and quantisation methods, *Quick Shift* image segmentation algorithm was selected in combination with a *K-means* colour quantisation with further processing steps selecting significant colours by using features of perceptional uniform colour model CIEL*a*b*. The combination of segmentation and quantisation methods provided a highly adapted approach on art image characteristics, which have been analysed consulting selected colour composition theory based on art images. With colour labelling based on basic colour terms as final step in analysis, the processing steps of chromatic extraction was also provided as an interactive web-interface, allowing colour distribution analysis in demonstrative 3-dimensional colour space presentations. The overall results of algorithmic colour extraction show high accuracy regarding empirical results from ARTigo image tagging game and also reveal starting points for further research and development. Even though this thesis' implementation considered assertively the locally distribution of colours on art images, this region-based approach could be improved by incorporating object detecting, which would increase the semantical information base unprecedentedly and would further approximate human perception and evaluation of colour sensation. The presented solution also has weaknesses in detecting reflecting materials like gold or silver, which play an outstanding role especially in religious art works. A promising approach to meet these weaknesses could be done by further analysis of spatial colour distribution and colour gradients. Altogether the colour analysis can be further improved by incorporating more background data, allowing highly adapted image processing based on art epochs, techniques and artists for example.

References

[AJJ+11] Jörgen Ahlberg, Fredrik Johansson, Ronnie Johansson, Magnus Jändel, Anna Linderhed, Pontus Svenson, and Gustav Tolt. Content-based image retrieval - An introduction to literature and applications. Technical Report FOI-R—3395–SE, FOI, Swedish Defence Research Agency, Stockholm, 2011.

[ASS+12] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[BB16] Wilhelm Burger and Mark J. Burge. *Digital Image Processing*. Texts in Computer Science. Springer London, London, 2016.

[Ben] BenRG. CIE1931xy blank (SVG image). URL: https://de.wikipedia.org/wiki/Datei:CIE1931xy_blank.svg [cited 5.2.2017].

[BG14] Robert R. Buckley and Edward J. Giorgianni. Cielab for color image encoding (cielab, 8-bit; domain and range, uses). In Ronnier Luo, editor, *Encyclopedia of Color Science and Technology*, pages 1–9, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[Bir76] Faber Birren. Color Perception in Art: Beyond the Eye into the Brain. *Leonardo*, 9(2):105, 1976.

[CCL+01] Luigi Cinque, Gianluigi Ciocca, Stefano Levialdi, A. Pellicano, and Raimondo Schettini. Color-based image retrieval using spatial-chromatic histograms. *Image and Vision Computing*, 19(13):979–986, 2001.

[CDP99] Jacopo M. Corridoni, Alberto Del Bimbo, and Pietro Pala. Image retrieval by color semantics. *Multimedia systems*, 7(3):175–183, 1999.

[Cel11] M. Emre Celebi. Improving the performance of k-means for color quantization. *Image and Vision Computing*, 29(4):260–271, 2011.

[Che95] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799, 1995.

[CM02] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[Com99] International Electrotechnical Commission. Multimedia systems and equipment–colour measurement and management – part 2-1: Colour management–default RGB colour space – sRGB. Technical report, IEC 61966-2-1, 1999.

[Con92] Damian Conway. An experimental comparison of three natural language colour naming models. In *Proc. east-west int. conf. on human-computer interaction*, pages 328–339. Citeseer, 1992.

[Del] Delimitri. Octree color add steps [online]. URL: http://delimitry.blogspot.de/2016/02/octree-color-quantizer-in-python.html [cited 5.2.2017].

[Duc03] David Duce. Portable Network Graphics (PNG) Specification (Second Edition). World Wide Web Consortium, Recommendation REC-PNG-20031110, November 2003.

[FH04] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

[Har13] C.L. Hardin. Berlin and Kay Theory. In Ronnier Luo, editor, *Encyclopedia of Color Science and Technology*, pages 1–4. Springer New York, New York, NY, 2013.

[HM97] C.L. Hardin and L. Maffi. *Color Categories in Thought and Language*. Cambridge University Press, 1997.

[HS12] Jeffrey Heer and Maureen Stone. Color naming models for color selection, image editing and palette design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1007–1016. ACM, 2012.

[Inc90] CompuServe Incorporated. Graphics Interchange Format Programming Reference. Technical Report GIF89a, W3C, 1990. URL: https://www.w3.org/Graphics/GIF/spec-gif89a.txt.

[IS09] K. Ivanova and P. Stanchev. Color Harmonies and Contrasts Search in Art Image Collections. In *2009 First International Conference on Advances in Multimedia*, pages 180–187, July 2009.

[ISD08] Krassimira Ivanova, Peter L. Stanchev, and Boyan Dimitrov. Analysis of the distributions of color characteristics in art painting images. *Serdica Journal of Computing*, 2(2):111–136, 2008.

[ISV+12] Krassimira Ivanova, Peter Stanchev, Evgeniya Velikova, Koen Vanhoof, Benoit Depaire, Rajkumar Kannan, Iliya Mitov, and Krassimir Markov. *Features for Art Painting Classification Based on Vector Quantization of MPEG-7 Descriptors*, pages 146–153. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[ISVD12] Krassimira Ivanova, Peter Stanchev, Koen Vanhoof, and Milena Dobreva. *APICAS – Content-Based Image Retrieval in Art Image Collections Utilizing Colour Semantics*, chapter 4, pages 153–201. Plovdiv University Publishing House, 2012.

[Itt74] J. Itten. *The Art of Color: The Subjective Experience and Objective Rationale of Color*. A VNR book. Wiley, 1974.

[Itu93] T. Itu. JPEG Standard, JPEG ISO/IEC 10918-1. Technical report, 1993. URL: http://www.cmpcmm.com/cc/standards.html.

[Iva11] Krassimira Ivanova. Content-based image retrieval in digital libraries of art images utilizing colour semantics. In *International Conference on Theory and Practice of Digital Libraries*, pages 515–518. Springer, 2011.

[KF09] P. Kay and CSLI Publications (Firm). *The World Color Survey*. CSLI Publications. CSLI Publications, 2009.

[Lin13] Albrecht Lindner. *Semantic Awareness for Automatic Image Interpretation*. PhD thesis, École Polytechnique Fédérale de Lausanne, March 2013.

[LM01] L. Luccheseyz and S. K. Mitray. Color image segmentation: A state-of-the-art survey. *Proceedings of the Indian National Science Academy (INSA-A)*, 67(2):207–221, 2001.

[LY13] Guang-Hai Liu and Jing-Yu Yang. Content-based image retrieval using color difference histogram. *Pattern Recognition*, 46(1):188–198, January 2013.

[MHS02] A. Mojsilovic, H. Hu, and E. Soljanin. Extraction of perceptually important colors and similarity measurement for image matching, retrieval and analysis. *IEEE Transactions on Image Processing*, 11(11):1238–1248, November 2002.

[MX10] Randall Munroe and XKCD. Color Survey [online]. 2010. URL: https://blog.xkcd.com/2010/05/03/color-survey-results/ [cited 6.2.2017].

[NpK04] H. Nezamabadi-pour and E. Kabir. Image retrieval using histograms of uni-color and bi-color blocks and directional changes in intensity gradient. *Pattern Recognition Letters*, 25(14):1547–1557, 2004.

[Pie] Chris Piech. K Means - handout. URL: http://stanford.edu/~cpiech/cs221/handouts/kmeans.html [cited 5.2.2017].

[Pri15] Lutz Priese. *Computer Vision*. eXamen.press. Springer, Berlin, Heidelberg, 2015.

[RAL16] Deutsches Institut für Gütesicherung und Kennzeichnung RAL. History of RAL colours. Technical report, RAL, 2016. URL: http://toxid.ral-farben.de/fileadmin/pdf/RAL_History.pdf.

[RM03] Xiaofeng Ren and Jitendra Malik. Learning a Classification Model for Segmentation. In *ICCV*, volume 1, pages 10–17, 2003.

[SAU04] Y. Sirisathitkul, S. Auwatanamongkol, and B. Uyyanonvara. Color image quantization using distances between adjacent colors along the color axis with highest color variance. *Pattern Recognition Letters*, 25(9):1025 – 1043, 2004.

[SBS99] Sabine Süsstrunk, Robert Buckley, and Steve Swen. Standard RGB Color Spaces. In *Proc. Society for Imaging Science and Technology 7th Color Imaging Conference*, volume 7, pages 127–134, 1999.

[SC95] John R. Smith and Shih-Fu Chang. Single color extraction and image query. In *Image processing, 1995. Proceedings., International conference on*, volume 3, pages 528–531. IEEE, 1995.

[Sch16] János Schanda. *CIEuv Uniform Chromaticity Scale Diagram and CIELUV Color Space*, pages 185–188. Springer, New York, NY, 2016.

[SGD03] Peter Stanchev, David Green Jr, and Boyan Dimitrov. High level color similarity retrieval. 2003.

[Shaa] SharkD. Hsl color solid dblcone. URL: https://commons.wikimedia.org/wiki/File:HSL_color_solid_dblcone.png [cited 13.2.2016].

[Shab] SharkD. Hsv color solid cone chroma gray. URL: https://commons.wikimedia.org/wiki/File:HSV_color_solid_cone_chroma_gray.png [cited 13.2.2016].

[SS16] Vidya Setlur and Maureen C Stone. A linguistic approach to categorical color assignment for data visualization. *IEEE transactions on visualization and computer graphics*, 22(1):698–707, 2016.

[SWD05] Gaurav Sharma, Wencheng Wu, and Edul N Dalal. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application*, 30(1):21–30, 2005.

[Tve15] Julianne Tveten. Tomato' versus #FF6347 — the tragicomic history of CSS color names | Ars Technica. *arstechnica.com*, 11 2015. URL: https://arstechnica.com/information-technology/2015/10/tomato-versus-ff6347-the-tragicomic-history-of-css-color-names/.

[vAD08] Luis von Ahn and Laura Dabbish. Designing games with a purpose. *Commun. ACM*, 51(8):58–67, August 2008.

[vdWS07] J. van de Weijer and C. Schmid. Applying color names to image description. In *2007 IEEE International Conference on Image Processing*, volume 3, pages III – 493–III – 496, Sept 2007.

[vdWSVL09] Joost van de Weijer, Cordelia Schmid, Jakob Verbeek, and Diane Larlus. Learning color names for real-world applications. *Trans. Img. Proc.*, 18(7):1512–1523, July 2009.

[Vie14] Vilson Vieira. Example: CIELAB color space [online]. February 2014. URL: http://www.texample.net/tikz/examples/cielab/ [cited 5.2.2017].

[VS08] Andrea Vedaldi and Stefano Soatto. Quick Shift and Kernel Methods for Mode Seeking. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Computer Vision – European Conference on Computer Vision 2008*, volume 5305, pages 705–718. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[WAMS07] L. Wittgenstein, G.E.M. Anscombe, L.L. McAlister, and M. Schattle. *Remarks on Colour/Bemerkungen Uber Die Farben*. University of California Press, 2007.

[WBBL13] Christoph Wieser, François Bry, Alexandre Bérard, and Richard Lagrange. Artigo: building an artwork search engine with games and higher-order latent semantic analysis. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[Wil16] Charly Wilder. On the bauhaus trail in germany. *The New York Times*, August 14th 2016. URL: https://www.nytimes.com/2016/08/14/travel/bauhaus-germany-art-design.html.

[Wyl92] S. Wyler. *Colour and Language: Colour Terms in English*. Tübinger Beiträge zur Linguistik. G. Narr, 1992.