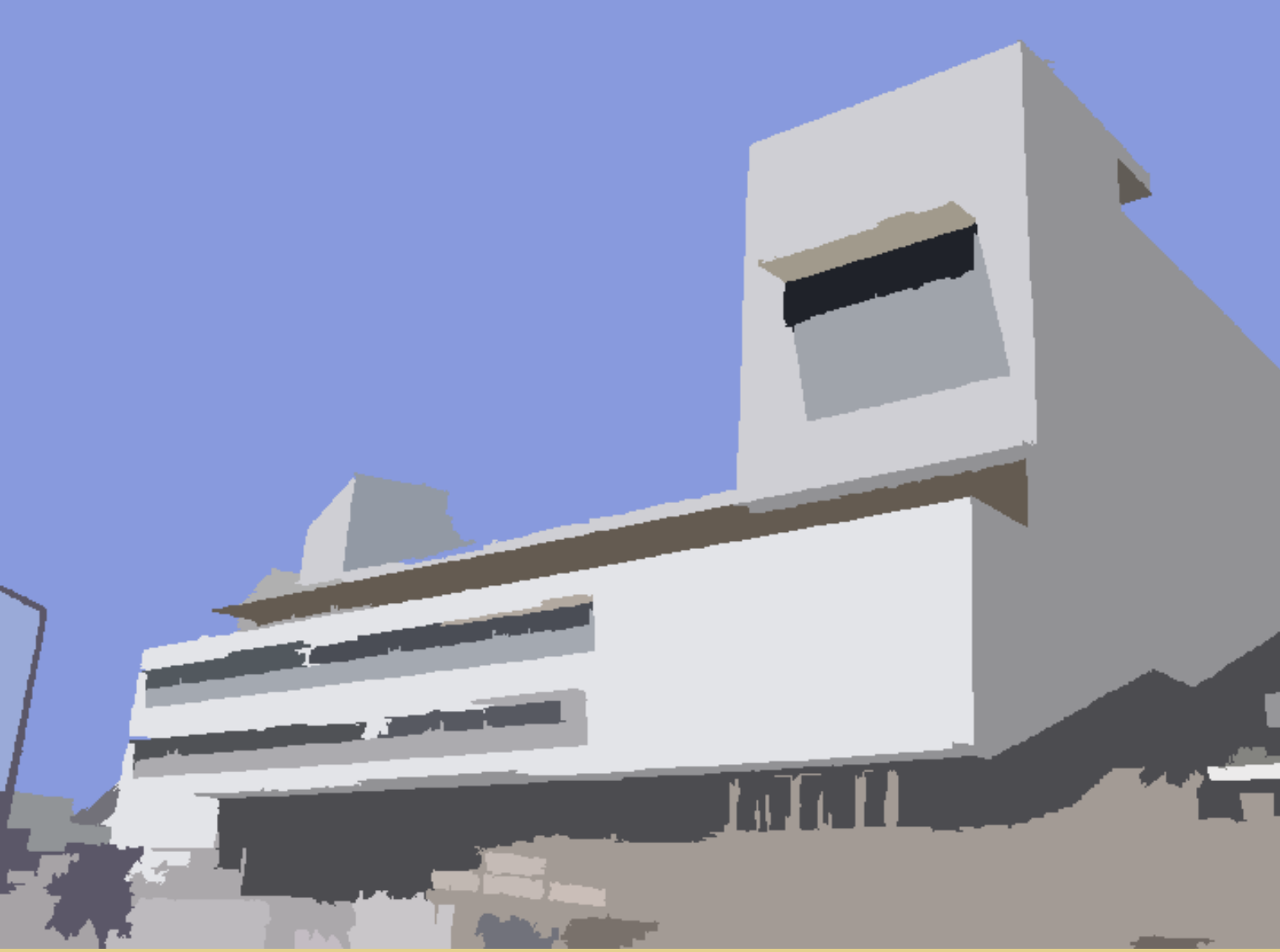


Jorge Monteiro

Human-Inspired Object Discrimination by Color for Artificial Attention

Dissertação de Mestrado em Engenharia Eletrotécnica e de Computadores
09/2017

· U C ·

Universidade de Coimbra

UNIVERSIDADE DE COIMBRA
FACULDADE DE CIÊNCIAS E TECNOLOGIAS
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

Human-Inspired Object Discrimination by Color for Artificial Attention

Jorge Miguel de Carvalho Monteiro

Supervisor: Prof. Doctor João Filipe de Castro Cardoso Ferreira

Jury:
Prof. Doctor Luís Alberto da Silva Cruz
Doctor Cristiano Premebida
Prof. Doctor João Filipe de Castro Cardoso Ferreira

Dissertation submitted to the Electrical and Computer Engineering Department of the Faculty of Science and Technology of the University of Coimbra in partial fulfilment of the requirements for the Degree of Master of Science.

Coimbra, September of 2017

I would like to dedicate this thesis to my loving parents, Jorge and Tina, my beloved girlfriend Ana Luísa Reis, my close relatives, Isa, grandma Constança, Kico, Vera and Rafaela, my college family, Eva, Bruno, Inês, Migalhas and Noira, my squad, Eduardo, Gertrudes, Maurício, "my best friend" Franca, Catarina, Daniela, JT, Freire, Alexandre, Inês Costa, Matias, "my cousin" Ramires, Zé Pedro, Vasco, Martins, Castanheira, Cavaleiro, Rui, Afonso, Lima, Chichorro, Abreu, Gui, Abegão, Fraga, Weasel, Pepi, Ivo, and Steve, the chaps at home, Blue and Girão, my great friends in my home town, Paulo, Filipa, Catarina, Sónia, Rute, Tânia, Badjola, Rita, Afonsito, Roque, Pedro and Filipe, the guys at the lab that created a fantastic environment to work in, Gonçalo, Mira, Hugo, Fernando, Nuno, Bruno, Farzan, Francisco and João Pedro. Finally to all the professors that accompanied our daily lives in the lab, professor Rui Rocha, professor Paulo Menezes, professor Luís Santos, professor Jorge Lobo and finally, the one that followed the work from beginning to end, professor João Filipe Ferreira.

Acknowledgements

I would like to thank my supervisor, professor João Filipe Ferreira for the help and support provided during the whole time I spent in the lab, especially during the writing of my master's degree thesis.

Abstract

Human beings are very efficient in detecting target objects among distractors using color as a search feature. Different aspects on a scene drive the attention of the observer, which does not equally process all this information, and color is one of the basic features that influences attentional capture. Additionally, humans are particularly impressive at appropriately discriminating colors. They are able to successfully identify them under different degrees of illumination and congruently classify them in abstract terms. This has a high impact in human performance in object search by color. The goal of the presented work is to create a biologically plausible solution that is able to regulate attentional capture according to a visual search objective that depends on a color that can be abstractly defined. To attain this goal, first a set of behavioral experiments was conducted to select the approach that best reflected human performance. Results showed that the RGB color model for computing similarity to a reference color and the mode for computing the dominant color of an image patch were the methods most consistent with human behavior. Additionally, it was proven that the effect of abstract color classification had a marginal effect in memorizing a reference color, and that Retinex-based color constancy algorithms used in preprocessing did not substantially improve overall processing performance. Next, the final processing pipeline was designed. Different algorithmic alternatives for proto-object segmentation were tested, and the proposed solution was applied to video frames taken from a typical scenario. The proposed solution was found to have satisfying performance in terms of replicating human behavior. Future work includes a final optimized, real-time implementation of the pipeline in a artificial attention system developed in previous research performed at the Institute of Systems and Robotics.

Keywords: **color**, **visual search**, **attentional capture**, **similarity**, **saliency**, **abstract color classification**

Resumo

Os seres humanos são muito eficientes a detectar objetos-alvo entre distrações usando a cor como meio de procura. Diferentes aspetos na cena captam a atenção do observador, que não processa toda a informação da mesma maneira, e a cor é uma das propriedades básicas que influencia a captura por atenção. Adicionalmente, os humanos são particularmente impressionantes a discriminar cores de uma forma apropriada. São capazes de as identificar com sucesso em diferentes graus de luminosidade e classificá-las apropriadamente usando termos abstratos. Tudo isto tem um grande impacto na procura de objetos pela cor. O objectivo deste trabalho é criar uma solução biologicamente plausível que seja capaz de regular a captura por atenção de acordo com um objectivo obtido por procura visual que dependa de uma cor que possa ser abstractamente definida. Para atingir este objectivo, foram primeiramente executadas uma série de experiências para analisar qual a melhor abordagem que reflita o comportamento humano. Os resultados demonstram que o modelo de cor RGB usado para calcular a semelhança de cores relativamente a uma cor de referência e a moda para obter a cor dominante de uma imagem foram os métodos mais consistentes comparativamente ao comportamento humano. Adicionalmente, foi provado que o efeito de classificação abstrata de cor tinha um efeito marginal na memorização de uma cor de referência, e que algoritmos baseados no método Retinex usados em pré-processamento não melhoravam a performance do processamento. De seguida, uma pipeline de processamento foi desenhada. Foram testados vários algoritmos alternativos para a segmentação em proto-objetos, e a solução proposta por eles foi aplicada a frames de um vídeo captado num cenário quotidiano. A solução proposta aparentou ter resultados satisfatórios em termos de replicação do comportamento humano. Trabalhos futuros includem uma versão em tempo-real e optimizada do pipline num sistema de atenção artificial desenvolvido em investigações prévias no Instituto de Sistemas e Robótica.

Palavras-chave: **cor**, **procura visual**, **captura por atenção**, **semelhança**, **saliência**, **classificação abstrata de cor**

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Motivation

Color is a crucial discriminative feature in human visual perception. In fact, visual perception in the human brain includes color from its earliest processing stages [1]. Additionally, humans and other primates evolved beyond the more common dichromatic color vision of most mammals towards trichomatric color vision, a relatively new genetic development, allowing for improved discrimination between various shades of red and green [2, 3]. This evolution is believed to be a survival skill for effectively discriminating the color of tropical fruits, resulting from the development of specialized tuning to long wavelength colors (red, orange and yellow), which often signal the most nutritious fruits at hand [3].

Human beings therefore became very efficient in detecting target objects among distractors using color as a search feature [4], a process named *visual search by color*. The ability to discriminate color helps us to detect objects that might otherwise be confused with their surroundings [1]. As a species, we are particularly impressive at appropriately discriminating colors, successfully identifying them under different degrees of illumination [1] and coherently abstractly classifying them [5][1].

To improve sensory and computational resource usage, the human brain does not equally process all the sensory data from a scene – in fact, the brain anticipates its needs and selects

[1] These issues will be discussed in detail in Chapter 2.

Fig. 1.1 Real life example of color contrast. The orange/blue disparity is often used in movie posters to convey sensations of both coolness and enthusiasm. It performs well because orange and blue are complementary colors (©Abheetsingh / Wikimedia Commons / CC BY-SA 4.0)

which resources to use and what parts of the scene should be scanned by the senses in a process called *attention* [6]. Attention is volitionally directed by the goals of the observer, but it is also heavily driven by the inherent relevance of the sensory stimuli themselves, which is unconsciously determined by the brain – in fact, it is believed that the current goal of the observer modulates stimulus-driven salience processes by means of an *attentional set* [7, 6]. Color, as a basic attentional feature, is a particularly important driver of stimulus salience. This results in particularly strong pop-out effects, in which the salience of a feature captures attention.

Color pop-out effects have been usually taken as being mainly driven by local color contrast, i.e. a striking difference in color between a particular object or part of an object and its surroundings[2]. In fact, color contrast is frequently used in advertising, most commonly in movie posters, in order to convey different feelings (Fig. 1.1). However, *color priming*, the preparation for detecting a target by focusing on its color, is known to take on an important role in visual search by color because it increases search effectiveness by decreasing detection times through the pop-out effect. In fact, Becker et al. [8] recently showed that relative color of the target in opposition to non-targets might be more important than contrast *per se*. For instance, consider a red ball on a grassy field, and a green ball on a red carpet. The contrast

[2]To be introduced in detail in Section 2.1.2.

between object and background is the same in both cases, but the time it takes to find the ball by presenting the latter scenario after the former is not influenced, whilst when using the same target color consecutively detection time decreases significantly [8]. This appears to mean that the *reference target color* is at least as important as (and probably cumulative with) color contrast – in fact, it is possible that the observer's goal of visual search by a reference target color (e.g. "find the red ball") is modulating the salience of that particular color and thus increasing its pop-out effect.

In summary, the main motivation for the work presented in this thesis rests on the working hypothesis that visual search by color in artificial perception might also benefit from these types of mechanisms.

1.2 Related Work

The basis for attentional prioritization in our work relates to seminal research by Itti et al. [9]. Their studies resulted in a computational solution for stimulus-driven attention inspired on the behavior and neuronal architecture of the early primate visual system. The system breaks down the problem by selecting conspicuous locations to be analyzed in detail. For this purpose, a salience map is proposed, to "represent the conspicuity at every location in the visual field by a scalar quantity and to guide the selection of attended locations, based on the spatial distributions of saliency" [9][3]. In accordance to what was said in the previous section, one of the features driving this solution, and most of its derivative follow-ups, is color contrast, which is commonly used as a conspicuity channel in salience computation.

Gelasca et al. [10] propose the only solution, to our knowledge, that directly explores color relativity instead of contrast as salience; however, their solution relies on a static *a priori* prioritization, as would be expected when performing free viewing, i.e. no specific attentional goal, as opposed to visual search based on a reference color. More specifically, they performed a free viewing experiment in which they demonstrated that human subjects shared a common prioritization ranking of colors (red in this case would be the most important

[3] See also Section 2.2.1.

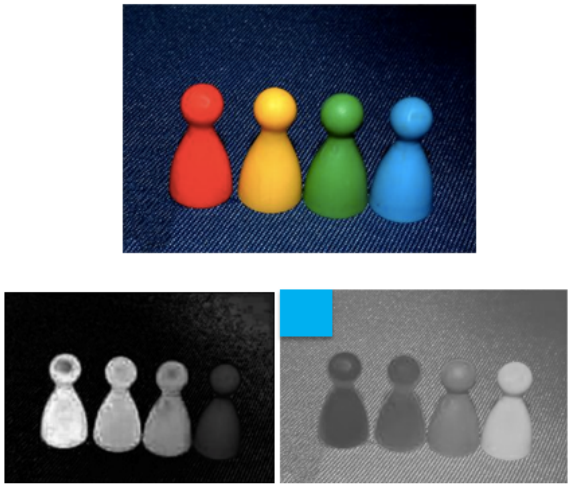


Fig. 1.2 Differences between attentional capture by color. On the top we have the original picture. On the bottom left we have a salience map using color contrast, and on the bottom right using similarity to reference color (obtained using a modified version of the solution proposed in Section 4.3), which is displayed on its top left corner. In both maps, lighter pixels have been classified as more salient, while darker pixels as less so – attention will consequently be captured by the lightest pixel in each case. Original image and capture by contrast kindly provided by Dr. Radhakrishna Achanta, author of [12].

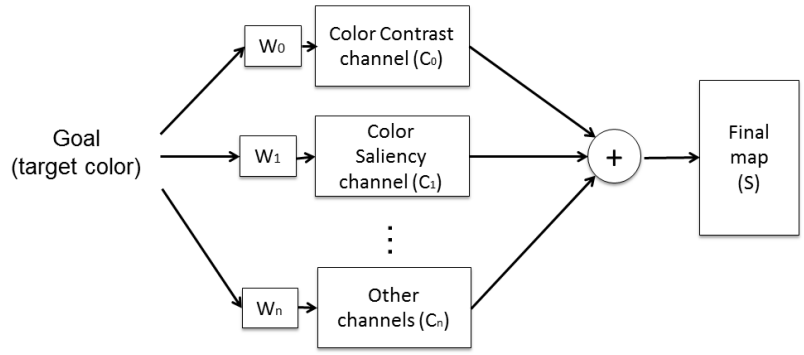
color from a set of 12, followed by yellow and green), and then proceeded to implementing this prioritization as a saliency model.

1.3 Goals and Contributions

As mentioned in Section 1.2, we have no knowledge of any work that studies color salience in artificial systems in a systematic fashion, in particular involving a reference target color. Additionally, the majority of the proposed solutions in the literature use salience channels for color contrast, with the sole exception of [10].

We therefore intend to contribute with a biologically-plausible solution that takes the similarity to a reference color that we wish to find within the given scene as a measure of saliency (Fig. 1.2). This mechanism is to be integrated into an attentional system developed during the CASIR FCT-funded project (Coordinated Attention for Social Interaction with Robots – FCT Contract PTDC/EEI-AUT/3010/2012) [11] – see Fig. 1.3[4].

[4] And also http://mrl.isr.uc.pt/projects/casir/ for more information on the project.



$$S = W_0 C_0 + W_1 C_1 + \ldots + W_n C_n$$

Fig. 1.3 Simplified conceptual diagram of part of the CASIR-IMPEP attentional middleware. The attentional goal of the observer (which, if engaged in a visual search, will include a target color) will influence the weights (W_0, W_1, ..., W_n) of the saliency channels(C_0, C_1, ..., C_n) resulting in the final map (S). This map is obtained by calculating a weighted sum of all the channels.

1.4 Structure of Dissertation

This dissertation is structured as follows:

- **Chapter 1** explains the motivation for the reported work, surveys related research, and states the main goal and expected contributions to the state of the art.
- **Chapter 2** introduces and defines relevant and presents the specific implementation algorithms and methods used for implementation of the proposed solution.
- **Chapter 3** reports on the behavioral studies conducted to ensure the biological plausibility of the proposed solution and respective findings.
- **Chapter 4** describes the testing and selection of the algorithms used in the implementation of the final processing pipeline and reports results of its application.
- **Chapter 5** concludes the dissertation by drawing conclusions and proposing follow-up work.

Chapter 2

Background and Methods

2.1 Background and definitions

2.1.1 Color model / color space

A *color model* can be described as a mathematical model consisting of tuples of numbers used to represent colors. These tuples can be represented in a delimited $n-$dimensional space, called *color space*.

The most common color spaces used on this work were *RGB* (red, green, blue) , *Lab* (lightness, green-red ratio, blue-yellow ratio), *HSV* (hue, saturation, value), *YUV* (luminance, chrominance 1, chrominance 2), *HCL* (hue, chroma, luminance) [4]. These color spaces were selected based on the different ways they represent color, and, therefore, the different color distance methods associated with them.

Using the 3 *RGB* coordinates as a starting point, we can easily obtain the coordinates for the remaining color spaces. The *RGB* color space faces some limitations such as the impossibility to represent some colors by superposing the three values and difficulty in detecting the presence (or absence) of a specific color [4].

The *HSV* (Hue, Saturation, Value) represents color in a way that can be compared to the process of selecting colors from a wheel or palette. One can experience the usage of the *HSV* color space during the process of selecting paint on a hardware store, as the shades of

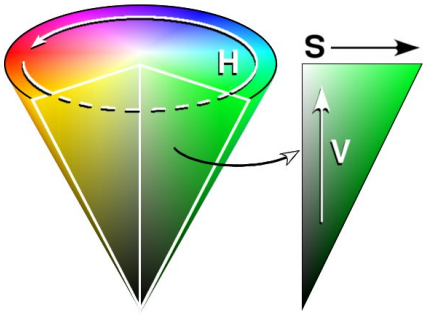


Fig. 2.1 Depiction of the HSV color space. (©Eric Pierce / Wikimedia commons / CC BY-SA 3.0)

colors are represented using this color space, because it corresponds better to how people experience color in detriment do the *RGB* color space [13]. The Hue component (H) specifies a dominant color as it is perceived by a human. Saturation (S) refers the amount of white light that gets mixed up with the previous component, therefore creating brighter shades of the same color [4]. The smaller the value of saturation, the most the color resembles a shade of gray. Finally, Value (V) represents brightness, and the smaller it is the more it appears as a darker shade of gray. The limitation of this color space resides on the fact that an high amount of white present in a color has a small variation of luminosity compared to a fully saturated color.

Contrary to the RGB color space, YUV's chrominance has two dimensions (U and V), and the intensity is coded on the third (Y) [14]. Due to this characteristic, YUV is normally associated to a color image encoding similar to human perception [14].

Lab (also referred as L*a*b* in some literature) is a color space defined by the CIE (Comission Internatuinale de l'Eclairage). The main problem associated to the usage of this color space is the inadequacy to represent shades of blue [4].

The HCL (hue, chroma, luminance) color space was also used. Assuming that both chrominance and hue of a specific color can be obtained as a blend of red, green or blue, therefore we can obtain the coordinates for the HCL color space from RGB tuples [4]. The calculations used to obtain the HCL coordinates take into account several human traits, such as the reaction to color intensity. Therefore, one can argue that HCL seems to be the best suit

to simulate human behavior at color perception, although the results shown on 3.2.5 prove otherwise.

Besides these more standard color models, two more color models were used on this work, which were proposed by [15]: normalized RGB (nRGB) and $L_1L_2L_3$, due to their invariance to changes in viewing direction, illumination and object geometry [15]. The first of the two is a color model that takes information from the main diagonal axis on the cube formed in the 3 dimensional space created by the R, G and B axes. By connecting the "black" and "white" corners we can define Intensity, which consists on an important factor that allows the transformation into nRGB coordinates. It is simply obtained by dividing the value of each pixel by the sum of the values of the remaining channels of the same pixel. According to the work previously cited, this color model seems to be the most suited when there are no highlights and under the constraint of white illumination. The $L_1L_2L_3$ color model works in a similar way, as it is also obtained by calculations involving all the channels of a specific pixel, but produces an entirely different result. This new model represents color in a way that is only dependent on the sensors and the surface albedo. More information on these color models can be found on Appendix B.

2.1.2 Color similarity, difference and contrast

Color similarity can be defined as the subjective evaluation of how identical two color samples are. The example on Fig. 2.2 depicts the situation better. The perception of color similarity varies from individual to individual: the way one perceives and discriminates colors is different for each person (e.g. some people tend to find the same tone of green closer to yellow and another group of people might find them closer to cyan, as depicted on Fig. 2.2).

Color difference refers to the distance (for this reason sometimes also called “color distance”) between two colors in a particular color space. In this work it is used as a measure for color (dis)similarity.

There are several standard color difference metrics, for example Euclidean distance, which is given by

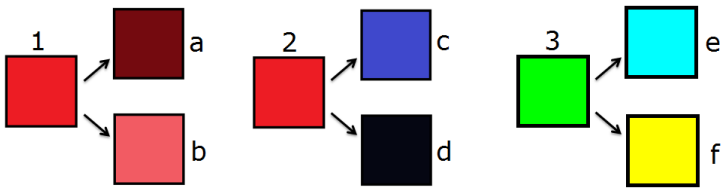


Fig. 2.2 Three different sets of colors used to define color similarity. On the left, it is easy to infer how human subjects will grade the similarity of the two colors a and b to reference color 1. Conversely, for reference colors 2 and 3 it is not trivial to infer how subjects will react when trying to classify their similarity to color-pairs c,d and e,f, respectively.

$$D_E = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}, \tag{2.1}$$

and also weighted Euclidean, which is given by

$$D_{WE} = \sqrt{W_1(x_1 - x_2)^2 + W_2(y_1 - y_2)^2 + W_3(z_1 - z_2)^2}, \tag{2.2}$$

where W_1, W_2 and W_3 are different values (weights).

For the *RGB*, *HSV*, *nRGB* and $L_1L_2L_3$ color spaces, we compared both of these metrics.

The Lab color space has a particular color distance measure called ΔE^*_{ab}, which is essentially an Euclidean distance. The *HCL* color space has a distinct color similarity measure associated, as it was adapted to better simulate human behavior. Consequently, this color difference (Section 2.1.2) metric takes into account human color perception [4], and is given by

$$D_{HCL} = \sqrt{(A_L \Delta L)^2 + A_H(C_1^2 + C_2^2 - 2C_1C_2\cos(\Delta H))}, \tag{2.3}$$

where A_H is a parameter which helps to reduce the distance between colors having a same hue as the hue in the target (reference) color, and A_L is a constant of linearization for luminance from the conic color model to the cylindric model [4]. Considering two colors $A(H_1, C_1, L_1)$ and $B(H_2, C_2, L_2)$, ΔH refers to the difference in hue $(H_1 - H_2)$, and ΔL represents the difference in luminance$(L_1 - L_2)$.

Color contrast can be defined as the difference between the color of a region in space and the color of its surroundings. As described in Section 1.2, most of the state-of-the-art attentional capture algorithms use color contrast as a saliency channel.

2.1.3 Color constancy

Color constancy consists on the capability that human beings have which allows them to successfully identify similar colors under varying illumination conditions [1]. This is a useful skill when it comes to identifying colors regardless of shading and illumination. The Retinex theory (Section 2.2.5) attempts to computationally model this human trait.

2.1.4 Pre-attentive segmentation – proto-objects / superpixels

Proto-objects consist of pre-attentive structures with feature coherence that limit a specific area on an image, forming volatile perceptual units that are believed to be formed in early stages of visual processing in the brain. They are a partial byproduct of the simultaneous effect of the perception of color discrimination and constancy, among other features. This rapid pre-attentive segmentation of the image allows for the conjunction of image locations, features and objects as a unified *unit of attention* [6], making it arguably preferable to single abstract pixels.

There are several methods for segmenting proto-objects — in this work, superpixel clustering was used.

It is imperative that the proto-object segmentation produces an acceptable superpixel outlining with a proper dominant color attributed to it. The desired properties of superpixel segmentation are mentioned in [16]:

- Every superpixel may only overlap with a single object.
- The boundaries of a set of superpixels should constitute a superset of object boundaries.
- The performance of the application should not be jeopardized by including superpixel segmentation.

- The final result should be obtained with the minimum number of superpixels.

Therefore, by having a robust superpixel segmentation method, one can obtain proto-objects that resemble the original objects on the scene, which will facilitate the analysis of certain attributes in the original picture.

2.2 Methods and algorithms for implementing an artificial version of attention capture by color

2.2.1 Spatial prioritization of the visual field using color saliency

Distinctive sensory stimuli (the measure for "distinctiveness" was dubbed by Itti, Koch et al. [9] as *saliency*) attract attention more effectively when they are relevant to the task at hand. Attentional capture, resulting from the so-called *pop-out stimulus effect* (see Section 1.1) has been identified as the fast, automatic, *pre-attentive* evaluation of the relevance of the incoming stimuli according to *basic features*, which is then encoded into sensory-centred spatially organized maps [6]. The particular encoding divised by Itti et al., named *salience map* [9], attributes priority in attentional capture (typically as a integer or floating point grayscale value ranging from 0 to 255 or 0 to 1, respectively) to regions in an image, more specifically to pixels or groups of pixels, according to stimulus relevance – examples of this type of visual map were presented in Fig. 1.2. This map is then usually subjected to a winner-take-all process (i.e. its maximum is computed) to determine the next focus of attention. As also mentioned in Section 1.1, involuntary attentional mechanisms, while apparently fundamentally stimulus-driven, are additionally modulated by goal-directed influences through the so-called *attentional sets* that impose *task relevance* as a prioritising measure [6]. In other words, the sensory (bottom-up) distinctiveness of a feature interacts with the ongoing cognitive (top-down) goal.

As described in Section 1.3, we will follow up on previous work described in [11] and base our solution on this concept in order to perform spatial prioritization of entities in images according to similarity (Section 2.1.2) to a reference color.

2.2.2 Abstract color classification

Abstract color classification refers to the capability human beings have to associate an abstract class (represented by a name) to a group of colors. People with normal color vision generally tend to agree on abstract classifications, and can also restrict a specific class by adding a proper classifier (i.e. an adjective). For instance, an individual might classify a color as “red". Although this serves as a proper abstract class for that color, one could restrict the class the specific color belongs to even further by saying it is “dark red" or even “maroon". Both of the previous labels are acceptable to classify the color since they all fit the classification of “shades of red". The previous example shows that the way people classify color varies from one individual to the other, but there are some rules that apply. The influence of abstract classification in color memory was investigated in our work, and the results of this study are presented in Chapter 3.

2.2.3 Methods for proto-object segmentation

There are several methods that can be employed to segment an image into proto-objecs by creating superpixels, however, the algorithms used within this work focused on just two of them: ERS and SLIC. ERS (Entropy Rate Superpixel) is a superpixel segmentation method that is based on the "entropy rate of a random walk on a graph and a balancing term" [16]. The entropy rate was used in order to promote the creation of compact and homogeneous clusters and the balancing term ensured that these clusters were similar in terms of their size . Its creators propose a novel graph construction for images and prove that it induces a matroid (combinatorial structure that generalizes the concept of linear independence in vector spaces) [16]. The segmentation itself is obtained through the "graph topology that maximizes the objective function through the matroid constraint" [16]. On the other hand, SLIC (Simple Linear Iterative Clustering) is a faster method and is more memory efficient. It is based of k-means for superpixel generation, with some differing factors such as limiting the search space to a region proportional to the superpixel size, which causes the number of distance calculations to be reduced, and control over compactness and size of the superpixels

by computing a weighted distance measure that combines color with spatial proximity [17]. Unfortunately, the off-the-shelf implementation of the SLIC method only works on the Windows operating system, forcing us to mostly use the ERS method, given our time restrictions.

The Proto-Object segmentation methods used for the tests on this work (Section 4.2.1 and Section 4.3) were implemented using an adaptation of a series of off-the-shelf MATLAB scripts developed by Yu and coworkers [18]. These adaptations included the removal of unnecessary calculations (for our purpose) as well as adding some parameters and arguments to the functions in order to adapt them to fit our analysis.

This algorithm takes an image and segments it into Proto-Objects based on a few parameters. According to the author of the paper, the code was suited to work with images with an 800×600 resolution, using its default parameters. Empirical tests were performed to find the best set of parameter values for smaller resolutions, such as our target resolutions of 640×480 and 320×240.

The `ProtoObjectSegmentation` function returns the processed image and accepts 6 input parameters:

imageFile

As the name suggests, it contains the path to the image file we want to segment. It accepts images in every format, but works better with .png files.

cs

This parameter refers to the color space used to preform cluster analysis in order to obtain superpixels based on neighboring pixels that share similar features. The color spaces available are the same as in the original algorithm: *RGB*, *HSV* and *LAB*.

numSeg

This parameter determines the maximum number of proto-objects obtained in the final result. A smaller value of `numSeg` will require a shorter period of time to produce a poorly segmented

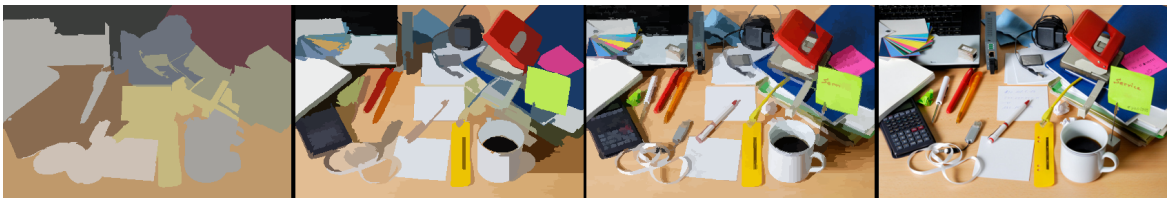


Fig. 2.3 Example of some results using the proto-object segmentation algorithm with varying number of superpixels. From left to right we have a segmentation done with 10 maximum proto-objects, followed by 100, followed by 1000. The image on the far right represents the original picture used. All the parameters, except for numSeg, were the same for all segmentations.

image, contrary to a higher value of `numSeg`, which will require a larger amount of time to produce an extremely segmented image. One should find the balance and choose the value that better suits their needs. Examples of segmentation using different values of `numSeg` can be found on Fig. 2.3.

segType

This variable determines the method for superpixel segmentation. It can either be 'SLIC' or 'ERS'. As was mentioned on Section 2.2.3 , the 'SLIC' method could only be used on the Windows operative system since it required .exe files to run, so, in order to have a multi-platform algorithm, the use of 'ERS' is advisable.

meanMed

`meanMed` determines what color to be attributed to a specific superpixel. It can take the form of 'MEAN', which attributes the mean color of all pixels contained in the superpixel, or 'MED', which does the same but used the median color instead.

BW

Referring to band width, this final parameter determines the search area for the shift directions [18].

Fig. 2.4 Dominant colors of real life objects. How does the human brain decide what is the dominant color in an object? In this picture, someone with normal color vision would probably classify the cookie box (1) as being orange, although it has numerous other colors in its composition. An easier choice can be made regarding the ketchup bottle (2), which one would agree is red, although it has some shades of white and yellow in its label. So, how do we make that choice?

2.2.4 Dominant color computation

Determining the dominant color of an object in a biologically plausible fashion proved to be challenging, since the way humans perform this task is still mostly unknown. Consequently, we decided to tackle the problem using simple statistical measures: the mean, the median and the mode. Due to time restrictions, we then followed a purely empirical strategy, and chose the method that produced the most visually convincing results (see Fig. 2.4). The examples shown in Figs. 2.5 and 2.6 portray two additional results obtained during the preliminary empirical tests. On all these examples the mode operand proved to be a valid option to determine the dominant color of an object. However, the example depicted in Fig. 2.7 represents a situation where the mode does not perform as adequately as the mean or the median.

The effect of each of these methods can be indirectly verified in the results presented in Chapter 3.

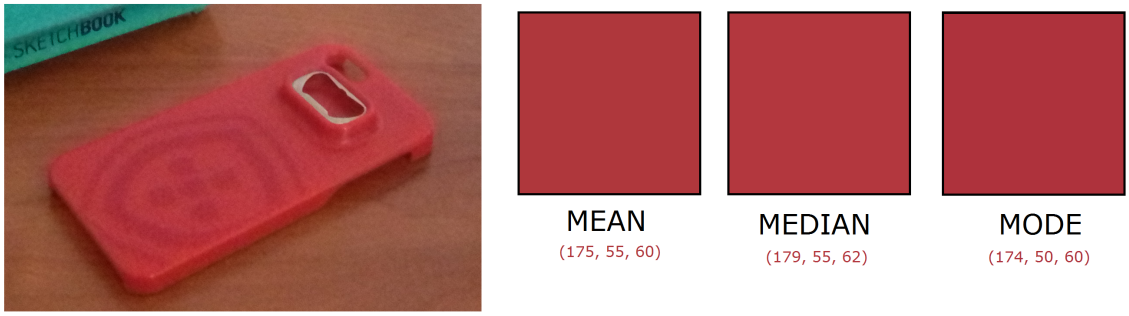


Fig. 2.5 Dominant color computing using mean, median and mode - example 1. In this example, the crop shown on the left contains an object. The following 3 colored squares represent the output of the algorithm as well as the RGB coordinates for each color; As it is perceivable, the 3 colors are very similar, and any of them could be easily described as the object's dominant color.

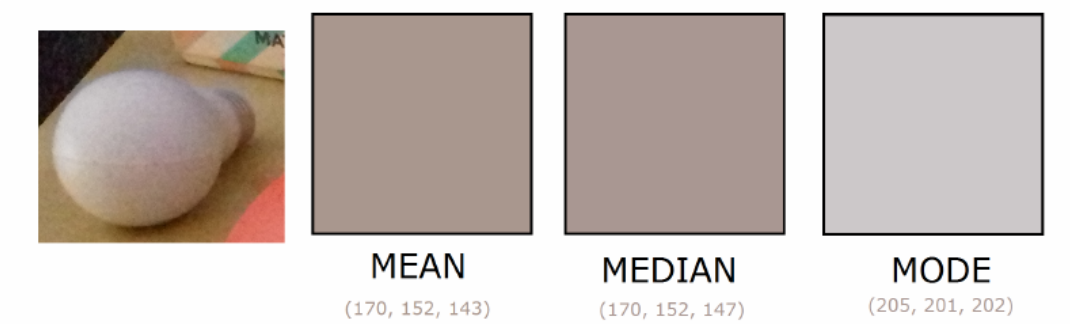


Fig. 2.6 Dominant color computing using mean, median and mode - example 2. The crop shown on the left contains an object. The following 3 colored squares represent the output of the algorithm as well as the RGB coordinates for each color; In this case, the results differ a bit more (compared to the previous example). The opinions on which of the 3 colors could be ascribed as the dominant one are, on this case, more divided.

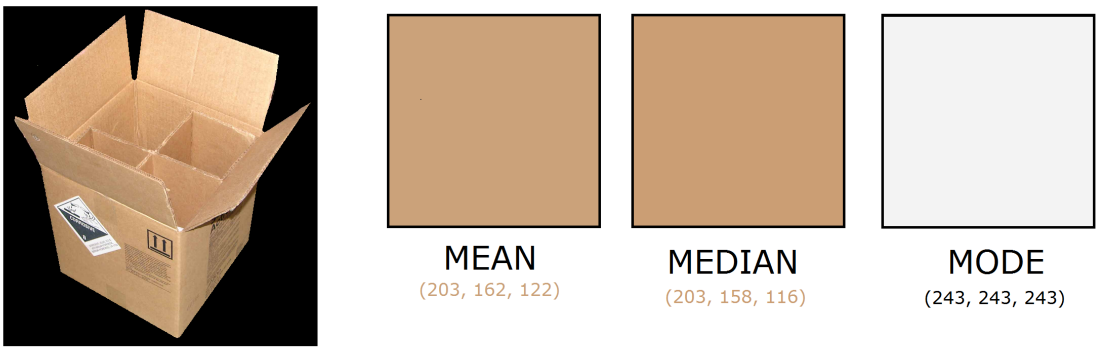


Fig. 2.7 Dominant color computing using mean, median and mode - example 3. Taking into account the pixels that compose the cardboard box, we can see that the mode operand, in this case, is the worst of all three. This is due to the fact that the sticker on the box had a lot of gray (243,243,243) pixels (probably caused by jpeg compression), which influenced the calculation. The remaining methods provide acceptable results. Cardboard box image: ©Rlsheehan / Wikimedia Commons / Public Domain.

Fig. 2.8 Example of Retinex pre-processing. Retinex algorithm input (left) and output (right). Output generated using the demo from [20]

2.2.5 Color constancy - the Retinex theory

As was mentioned in Section 2.1.3, human beings can easily identify similar colors under different degrees of illumination. In order to replicate this phenomena, the Retinex theory was created. It attempts to replicate and illustrate how the human visual system perceives a scene, most specifically, its color [19].

This theory establishes that human beings perceive relative lightness instead of absolute lightness. Relative lightness can be interpreted as a variation of lightness in local image regions. Therefore, the Retinex algorithms result in a shadow removal effect [20], creating the illusion depicted in Fig. 2.8. Retinex methods are therefore potentially useful, and were tested as a pre-processing method in Section 2.2.4.

Chapter 3

Color Salience as a Computational Model of Human Performance

3.1 Introduction

As a first step, a set of experiments were conducted in order to help frame the biological plausibility of our approach, which will be described in this chapter. With these experiments, we set out to attempt to answer the following research questions:

- How does the human brain classify, through a similarity measure, a color compared to a reference, both in an abstract and contextual situation? (Experiment 1, and Experiments 2 and 3 respectively.)
- How do humans determine the dominant color of a specific object on a scene? (Experiments 2 and 3)
- Does the lighting on a scene influence color vision in humans? If so, how do humans react to that change? (Experiment 3)
- How do humans abstractly classify similar colors? (Experiment 4.)
- Does attributing an abstract class to a color change a human subject's memory of the original color used as reference through time? (Experiment 4.)

- Do humans share a consistent set of color classification criteria? (All experiments.)

The color similarity scores provided by the subjects on the first 3 experiments were given as integers from 0 to 100 (which can be construed as a “degree of similarity" represented as a percentage), an unorthodox range of values for use in behavioral studies, which usually use psychometric scales such as the Likert-type scales [21], which typically use no more than 10 different values. This was done so as to achieve a more direct correspondence with the scale resolution used by the saliency methods described in Section 2.2.1.

3.2 Behavioral studies of color difference

3.2.1 Experiment 1 – decontextualized study of color difference

This experiment was conducted in order to study how humans rate the similarity between two colors in an abstract, non-contextualized setting.

A total of 15 subjects were inquired on this experiment. 13 of them were male and the remaining 2 were female, ranging from 19 to 65 years old. The average age of the participants was approximately 23 years old. A total of 14 were engineering students and one had no college degree whatsoever. All the subjects were inquired, before the start of the experiment, about their vision. The vast majority reported that they had normal or corrected-to-normal vision and all reported to have normal color vision. All the subjects were naïve to the experiment.

The experiment was conducted using MATLAB. An ASUS VW199 monitor was used to show the subjects the different images through out the experiment. The monitor measured $44 \times 29cm$ and worked with a resolution of 1400×900, with its brightness level at 100, contrast at 74, saturation at 50, using the settings of Color Temp. as "User Mode" and Skin Tone as "Natural".

Stimuli

In this experiment there were a total of 11 numbered comparison colors per reference color from a total of 15 for each subject. Both of these sets of colors were the same for each subject, although their order was randomly changed for each participant. The time limitation of ten minutes imposed by experimental design, chosen so as to bring participant discomfort to a minimum, determined the ratio of reference colors, comparison colors and time expended per scoring used – considering a mean response time of 4 seconds per comparison, we obtain a total of 660 seconds per trial, representing one minute over the 10 minute limit. The comparison colors were selected based on a subjective classification found on [22]. This classification ensured that we had at least one shade of every color present in the visible light spectrum. The selection of reference colors was performed using the same process, but since their number was greater, a few shades of the same colors, as well as tones of gray, black and white were added. An example can be found on Fig. 3.1.

Procedure

Subjects remained seated straight during the whole extent of the experiment, whilst looking a the computer monitor at a distance of roughly 48cm. Their chin was properly placed in a chin rest that allowed some horizontal head tilting but prevented vertical head swings. They were asked to keep their hands on their knees and to avoid moving their heads during the whole extent of the experiment, in order to maintain their head on the initial position. The computer monitor was always kept at the same angle and at the same distance from all the subjects; by doing so, all the subjects were able to keep the same eye level throughout the procedure.

The objective of this experiment was to compare the human subjects abstract scoring with scorings provided by an algorithm. The MATLAB script generated the order of the comparison colors and a specific reference color. A window similar to Fig. 3.1 was generated and the subject was asked to then score, on a scale from 0 to 100 (0 meaning colors were the exact opposite, and 100 that the colors were precisely the same), starting from comparison color 1 all the way to 11. This scoring measured the similarity between each comparison-

reference color pair. This value would then be called out to the technician, who would validate and introduce them via keyboard. When all the 11 scores were input, a new reference color would appear on the screen, as well as a new numerical order for the comparison colors. After this, a new batch of classification values would be annotated. The process was then repeated 13 additional times, in order to obtain the results for all the reference colors.

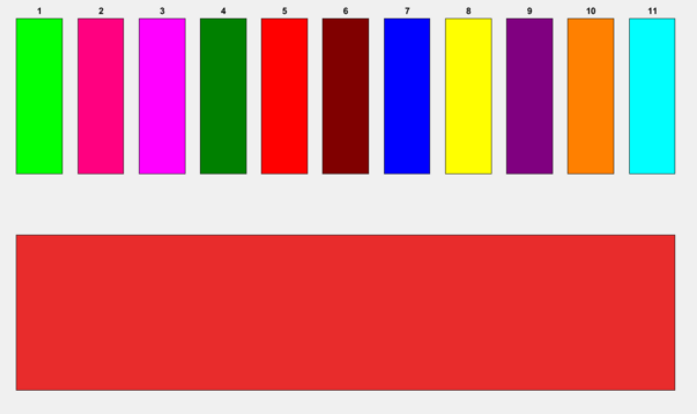


Fig. 3.1 Experiment 1 window example. The comparison colors are displayed in the topmost part and the reference color at the bottom.

3.2.2 Experiment 2 – contextualized study of color difference under regular lighting conditions

This experiment was created in order to investigate how human beings rate the similarity between two colors in a contextualized setting (i.e. viewing a photo of a real-life scenario) under regular lighting conditions. These results also provided some elucidation on the way human beings determine the dominant color of an object.

A total of 15 subjects were tested in this experiment. 4 of these subjects were female and the remaining 11 were male. The subjects had ages ranging from 23 to 65 years. The average age of the participants was 31 years old. From the total of 15 participants, 10 of them were engineering students, 2 were investigators at Instituto de Sistemas e Robótica, and 3 had no college degree whatsoever. As in the previous experiment, all of them were inquired, before the beginning of the experiment, about their vision. The vast majority reported that they had normal or corrected-to-normal vision and all reported to have normal color vision. Again, all the subjects were naïve to the experiment.

The materials used were the same as in experiment 1.

Stimuli

In this experiment, there were a total of 15 reference colors and 3 different images containing 5 highlighted objects each. Although the reference colors were shown to the participants, the comparison colors were not directly shown as in the previous experiment. Instead, they had to classify the dominant color of every highlighted object and use the result of their assessment as the comparison color. The reference colors were chosen using the following criteria:

1. 3 colors that were very similar to the dominant color of 3 different objects (calculated using euclidian distance);
2. 1 color that was almost the opposite of a forth object among the highlighted ones (calculated using euclidian distance);
3. A pure tone of *RGB* (Red 255, Green 255 or Blue 255);

In this case, there were 5 reference colors for each set of 5 objects, so a total of $5 \times 5 \times 3 = 75$ values that should be input. With an estimated comparison time of 5 seconds, the experiment should last approximately 7 minutes.

Procedure

Just like in the previous experiment, the subjects remained seated and maintained their positions following the same initial procedures described on Experiment 1.

The objective of this experiment is to compare contextual scoring given by human subjects with the ones provided by an algorithm, calculated using different methods. It also served as a case study on how humans determine what is the dominant color of an object on a scene.

The first window shown to the subjects (Fig. 3.2a) contained the first reference color. The participants should look at it, memorize it, and verbally signal the technician when they were ready to begin. After the verbal input, the window containing the reference color was

(a) Experiment 2 window 1 example.

(b) Experiment 2 window 1 example.

Fig. 3.2 Example of two windows used on experiment 2.

closed and substituted by a window containing the picture surrounded by the five numbered crops, each one with the corresponding highlighted objects (Fig. 3.2b). The subjects would then compare each of the objects dominant color with the reference color they had previously memorized using the same method as in Experiment 1. This process was repeated five times for each reference color. After inputing the last score associated with the object comparing to the fifth reference color, the technician would warn the subjects about changing the image, which lead to a new batch of results. This process was repeated a total of two more times, since there were three different images.

3.2.3 Experiment 3 – contextualized study of color difference using varying lighting conditions

The third experiment allowed us to better understand how humans rate the similarity of two colors also in a contextualized setting, but this time under varying lighting conditions. As before, it also provided insight concerning the way human beings determine the dominant color of an object on a scene.

A total of 15 subjects were tested in this experiment. 5 of these subjects were female and the remaining 10 were male. The subjects had ages ranging from 21 to 26 years. The average age of the participants on this experiment was of approximately 23 years old. All of them were either engineering students or investigators at Instituto de Sistemas e Robótica. As before, they were individually inquired, before the start of the experiment, about their

vision. The vast majority reported that they had normal or corrected-to-normal vision and all reported to have normal color vision. All the subjects were naïve to the experiment.

The third experiment had a similar procedure to the second one, where a reference color was first shown followed by the crops with highlighted objects on them. More specifically, 3 sets of 3 photos were taken, where each set had 3 pictures of the same scene captured with different degrees of illumination. The 3 sets contained several objects placed on multiple environments (All the pictures can be seen on Fig. 3.3). The objects would always differ between sets. The angle of the camera and the position of the objects were not changed during the photo shoot for each set, so that the only changing factor between pictures was the illumination conditions.

Stimuli

This experiment was mostly similar to the previous one, as it relied on the subject's ability to determine the dominant color on a specific object within a scene. In this case, there were 15 different reference colors divided into 3 sets of 5 colors each. There were also 3 sets of 3 images each, where each image belonging to the same set had 3 highlighted objects. The reference colors were chosen based on the criteria described on the previous experiment (Section 3.2.2), and the comparison colors were obtained through the same MATLAB script used for experiment 2. The subjects had to compare 5 reference colors with 3 highlighted objects on 9 pictures, which totaled 135 scoring values. With an estimated response time of 5 seconds, the experiment's duration was approximately 12 minutes.

Procedure

The subjects began by sitting comfortably on a chair and remained seated maintaining their positions by following the same initial procedures described on Experiment 1.

This experiment consisted of several steps, each step containing two windows. The first window (similar to 3.2a) shown to the subjects contained the reference color. The participants should take a look at it, memorize it, and verbally signal the technician when they were ready to begin the next step. Next, the first window was closed and replaced by the second

Fig. 3.3 All the pictures used for experiment 3. Each line consists on one of the three sets, with each picture taken with different lighting conditions: normal lighting (normal), artificial lighting(artificial) and dim lighting(dim). In the first set we have, from left to right, normal, artificial, dim, on the second one it is normal, dim, artificial, and on the third line we have artificial, normal, dim.

one (similar to 3.2b), containing the original picture and the 3 numbered crops, each one with the corresponding highlighted objects. The subjects would then compare each of the dominant color of the objects with the reference color they had previously memorized using the same scoring method explained in the previous experiments. This scoring would then be verbally signaled to the technician. This process was repeated three times for each reference color, one for each highlighted object. These actions constitute a step. There was a change of picture after the input of the third score corresponding to the fifth reference color for that specific image. By signaling the last score to the technician, he would then inform the subject about the change of pictures, and the process was repeated again. This procedure was repeated a total of 9 times, one for each picture of each set (Fig. 3.3).

3.2.4 Data collection and processing

In order to analyze and compare the data obtained from all the experiments some precautions had to be taken. The data was all stored in .txt files, one for each subject, that contained all the information needed for further analysis. In experiments 1 and 2 the .txt files had the following structure, divided into 7 separate lines:

- *RGB* coordinates (0-255) of the reference color, separated by commas (integers);

- *RGB* coordinates (0-255) of the comparison color, separated by commas (integers);
- Scoring attributed to the pair by the subject (0-100) (integers);
- Euclidian distance (0-100) in *RGB* (double);
- Euclidian distance (0-100) in YUV (double);
- Euclidian distance (0-100) in Lab (double);
- Euclidian distance (0-100) in HSV (double);

This was repeated as many times as was necessary to ensure that data regarding every pair of reference-comparison colors was collected. The format made it easy to develop the scripts to analyze all the data, since only a few parameters (like the number of reference and comparison colors) had to be changed from one script to the other.

Experiment 3 had a similar structure, but instead of 7 lines, the data was distributed into 9 lines. The first 7 contained the same information displayed previously, whereas the 8th and 9th contained:

- Image number (integer);
- Crop number (integer);

The different format can be justified by the fact that Experiment 3 had 3 sets of 3 similar pictures. Due to the similarity on some pictures within the same set, a reference to the picture and, subsequently, to the crop, was needed.

Regarding experiments 2 and 3, when the data was collected initially, it only contained placeholder values for the coordinates of the comparison colors. This was due to the fact that several methods for obtaining these coordinates were used. Because of this, a different set of scripts were created in order to substitute these placeholder values with coordinates regarding a specific method. Since the comparison color coordinates were changed, lines 4 to 7 also had to change, in order to display the newly calculated distances between reference and comparison colors. All of the new files were saved using a different name to prevent the overwriting of the old ones.

Table 3.1 Data regarding the subject-subject comparison. This values were obtained after following the procedure described in Section 3.2.5

Experiment	mean ρ	maximum p-value
1	0.747554	5.55E-15
2	0.735112	5.28E-09
3	0.715723	2.80E-07

Methods

As was mentioned before, experiments 2 and 3 used real life scenes as pictures, therefore, the *RGB* coordinates related to each object's dominant color had to be calculated using different methods mentioned in Section 2.2.4.

In order to do this, a MATLAB script was created that calculated the *RGB* coordinates of every pixel containing the desired object. The remaining pixels on the crop were set to an *RGB* triplet that was not contained in the object. After this, the statistical measures described in Section 2.2.4 were used to calculate a set of 3 *RGB* coordinates, that could be interpreted as the dominant color of that object.

The color differences between the two colors were obtained through distinct dissimilarity measures, namely the Euclidean distance and the weighted Euclidean distance or a color space-specific measure (Section 2.1.2).

3.2.5 Results and discussion

For every experiment, the results obtained from every subject were first sorted using a pre-determined order. This sorting process gave priority to the scores attributed and, in case of ties, the order was stipulated by a file containing all the pairs of colors. There was a different file for every experiment. The Spearman's rank correlation (see Appendix A) was then calculated for every subject-subject pair – Table 3.1. Results yielded mean ρ values over 0.71 for significance levels substantially under 1% – the high correlation value shows that subjects ranked color similarity in a coherent fashion, meaning that an aggregate ranking representing them as a group could be built in order to compare with each similarity computation method.

For the purpose of building this aggregate representation, a ranking corresponding to the median of the rankings of all subjects was determined, which we will call from now onwards, for simplicity sake, as the “median participant". An incremental value ranging from 1 to **total number of comparisons** was attributed to each line of the resulting sorted table of scores, hereby designated as “initial rank", and the fractional ranks method was applied to scores with the same value (see Appendix A for an explanation of this method).

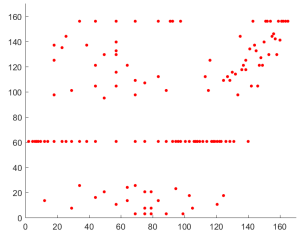
The rank for the scores given by the algorithms were calculated the same way as the rank of the median participant according to the similarity scores each method produced. These results were then compared to the median participant and placed into a file following the order obtained from the median participant, allowing the computation of the results that can be seen on tables 3.2, 3.3, and 3.4. The methods were then considered to be more suitable for final selection according to the following hierarchy of criteria:

1. Rank correlation – the correlation between how the method and the participants order the degree of similarity is the most important criteria in our perspective.

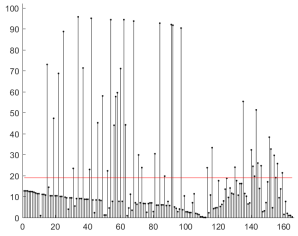
2. Mean absolute score error – in case rank correlations achieved by two methods are similar, this would serve as a tie-breaker.

3. Standard deviation of absolute score error – in case the two previous criteria are similar for two methods, the lowest standard deviation (which would correspond to a more stable algorithm) would serve as a tie-breaker.

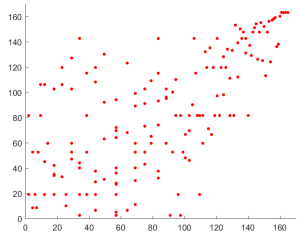
The data reveals that the *RGB* color space had the highest rank correlation in all the experiments (Tables 3.2, 3.3, 3.4). Concerning mean error, the results demonstrate that the $L_1L_2L_3$ color space scores the lowest in all the experiments; however, it performs rather badly in terms of ranking (i.e. low ρ values). This could be caused by the way the model itself was built (Appendix B). The instabilities associated with this method are caused by the computation of photometric invariants, which involve non-linear transformations, therefore this approach may be impractical for some applications [23]. These conclusions can be verified by analyzing Fig. 3.4.



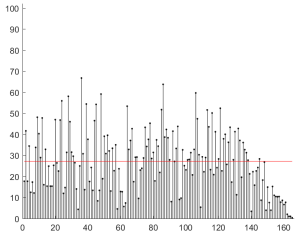
(a) $L_1L_2L_3$ rank comparisons



(b) $L_1L_2L_3$ score errors



(c) *RGB* rank comparisons



(d) *RGB* score errors

Fig. 3.4 Plots of the data obtained from the comparison between median participant and different methods in experiment 1. The graphs portrayed in (a) and (c) correspond to the plotting of the rank comparisons according to the process described in the beginning of Section 3.2.5. The graphs portrayed in (a) and (c) correspond to the plotting of the comparison of the $L_1L_2L_3$ and the *RGB* rankings (X-axis) with the "median participant" ranking (Y-axis) according to the process described in the beginning of Section 3.2.5.The plots on (b) and (d) display the absolute error between the scores obtained from the median participant and the scores attributed by the method. The red horizontal line represents the mean error.

The usage of *Retinex* pre-processing proved to have a residual effect in a few cases, as it scored the lowest mean error in Tables 3.3 and 3.4 (represented in cyan); however it significantly decreases the value of ρ on every situation. Due to the increase of processing time caused by the addition of *Retinex* pre-processing, the data obtained using the same method of determining the dominant color (without *Retinex*) was highlighted in red on table 3.3 and table 3.4. A close analysis shows that the values regarding the mean error are not that far apart, revealing that the methods highlighted in red can also be considered acceptable as results.

In regard to the best method to determine the dominant color of an object in an image, the mean and mode operand turned out to be the most accurate, despite the somewhat contradictory results shown on tables 3.3 and 3.4. On the first case, the highest Spearman's ρ was associated with a method using the mean operand and the lowest mean error to a method using the mode operand. The complete opposite can be seen on 3.4. By analyzing the results

Table 3.2 Summarized table for experiment 1. It comprises the best results obtained on experiment 1. The highlighted lines represent the method and metric with highest ρ (in green), and the lowest mean error (in cyan). The full content of this table can be seen on appendix C (table C.2).

Method	metric	ρ	Min Error	Max Error	Mean Error	stdev	Median Error	p-value
RGB	E	0.6818	0.33	66.69	27.12	14.75	27.07	3.34E-24
	WE	0.6855	0.33	63.5	27.47	14.93	26.80	1.56E-24
YUV	E	0.6748	0.33	75.35	44.89	16.51	47.15	1.43E-23
L1L2L3	E	0.3945	0.06	95.67	18.99	24.07	10.06	7.88E-08
	WE	0.3354	0.10	95.67	22.27	23.77	15.16	5.31E-06

Table 3.3 Summarized table for experiment 2. It comprises the best results obtained on experiment 2, namely the highest Spearman's correlation factor and lowest mean errors. The highlighted lines represent the method and metric with highest ρ (in green), the lowest mean error (in cyan), and the lowest mean error without using pre-processing (in red). The full content of this table can be seen on appendix C (table C.3).

Method	metric	ρ	Min Error	Max Error	Mean Error	stdev	Median Error	p-value
NR_Mean_*RGB*	E	0.6853	2.59	63.67	27.02	15.91	27.13	5.92E-12
	WE	0.6731	0.31	62.31	26.20	16.19	25.26	1.85E-11
NR_Median_*RGB*	E	0.6789	2.04	66.47	27.82	15.95	27.55	1.09E-11
	WE	0.6625	0.20	65.07	26.49	16.22	24.63	4.79E-11
NR_Mode_L1L2L3	E	0.4576	0.2	86.58	17.48	20.54	7.99	1.83E-05
	WE	0.4832	0.5	87.7	19.11	20.38	12.72	5.63E-06
WR_Mean_*RGB*	E	0.6821	0.17	64.08	26.77	16.03	27.85	8.03E-12
	WE	0.6668	0.42	62.09	26.22	16.26	25.84	3.26E-11
WR_Mode_L1L2L3	E	0.4573	0.49	83.55	16.81	19.40	9.06	1.86E-05
	WE	0.5029	0.11	84.97	18.35	19.19	12.61	2.13E-06

on both tables, one can conclude that either one of these operands can be used to determine the dominant color of an object.

3.3 Behavioral study of color classification

3.3.1 Experiment

A total of 26 subjects were inquired on this experiment. 15 of them were male and the remaining 11 were female, with ages ranging from 19 to 65 years old. The average age of the participants was 27.73 years old. A total of 20 were engineering students, 2 of them were law students, and 4 had no college degree whatsoever. All the subjects were inquired before the start of the experiment about any anomalies related to their vision and color vision. The vast majority reported that they had normal or corrected-to-normal vision and all reported to

Table 3.4 Summarized table for experiment 3. It comprises the best results obtained on experiment 3, namely the highest Spearman's correlation factor and lowest mean errors. The highlighted lines represent the method and metric with highest ρ (in green), the lowest mean error (in cyan), and the lowest mean error without using *Retinex* pre-processing (in red). The full content of this table can be seen on appendix C (table C.4).

Method	metric	ρ	Min Error	Max Error	Mean Error	stdev	Median Error	p-value
NR_Mean_L1L2L3	E	0.3172	0.08	78.27	20.62	21.46	12.00	8.89E-05
	WE	0.3521	0.02	79.97	21.99	21.74	14.85	1.41E-05
NR_Median_*RGB*	E	0.5636	0.02	68.16	31.46	17.35	32.96	5.56E-13
	WE	0.5496	0.07	69.98	31.99	17.39	34.03	2.54E-12
NR_Mode_*RGB*	E	0.5798	0.03	79.92	29.85	17.57	31.03	8.67E-14
	WE	0.5657	0.07	79.90	30.56	17.50	31.94	4.37E-13
WR_Mean_L1L2L3	E	0.3369	0.03	78.21	20.60	21.59	11.97	3.23E-05
	WE	0.3519	0.02	79.48	22.02	21.82	14.86	1.43E-05
WR_Median_*RGB*	E	0.5694	0.24	67.94	31.34	17.29	32.80	2.87E-13
	WE	0.5562	0.14	69.80	31.89	17.35	33.90	1.25E-12

have normal color vision. In spite of this one result had to be discarded due to non-diagnosed color blindness detected on a subject. All the subjects were naive to the experiment.

The equipment used on this experiment was the same used on the previous experiments.

Stimuli

In this experiment there were a total of nine reference colors and sixty four comparison colors. These were the same for all the subjects, although the order on which every comparison color was displayed on the screen varied from subject to subject. The order on which the reference colors appeared was always the same. To select the 9 reference colors, we fixed five values for each RGB component (0, 32, 64, 128 and 255), and then selected nine *RGB* triplets that supposedly would contain, all together, almost all of the comparison colors available. These comparison colors were generated by dividing the *RGB* spectrum into 3 different sections, which gave us four values to work with: 0, 85, 170 and 255. Since *RGB* colors work in triplets and each component could have one of four values, we had a total of sixty four different colors: $4 \times 4 \times 4 = 64$. By avoiding to choose the triplets $(0,0,0)$ and $(255,255,255)$, we had reference colors that were never the same as the comparison colors.

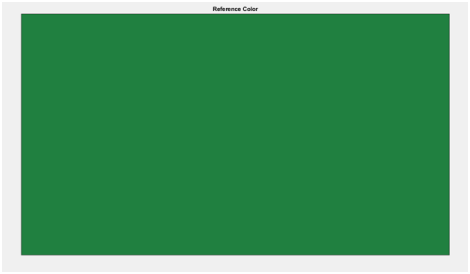
Procedure

The subjects followed the same preparation procedure described in the previous experiments. The duration of this experiment was approximately twelve minutes.

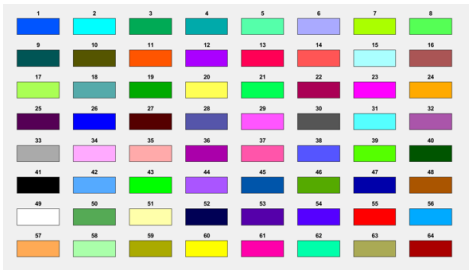
The goal of this experiment is to compare the classification (class attribution) given to the same colors by different human subjects. The experiment was composed of 9 steps (one for each reference color), each one having associated three different windows (Fig. 3.5). All the steps started by showing a window (Fig. 3.5a) on the screen with a squared filled with the first reference color. The subject had an undetermined amount of time to memorize it and to classify it (attribute an abstract class to it). After this process, the previous window was closed and the screen showed now a new window with a numbered 8×8 grid with the sixty four comparison colors (Fig. 3.5b). The subjects should then verbally identify the numbers corresponding to colors that fitted the abstract class previously named by them. The results were input in the computer by the hand of a specialized technician. When the subjects finished identifying the numbers of the colors, they would verbally warn the technician of this choice, which would lead to the closing of the comparison colors window and to the opening of the third one (Fig. 3.5c). In this final window, the reference color and all the colors that they had selected in the previous window (Fig.3.5b) were displayed through numbered vertical filled rectangles. The order on which they were represented was random. The subjects should then verbally identify the number associated to the reference color, according to their memory, whose value was input in the computer by the technician, and therefore starting the next step. This whole process was repeated eight more times.

3.3.2 Data collection and processing

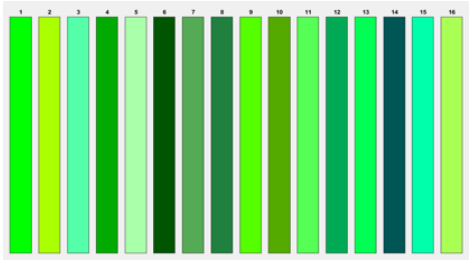
Since experiment 4 was completely different from the remaining ones, the way the data was stored had to be done using a different approach. Instead of having a single file for each subject, the data was separated in 28 different .txt files for each participant, three for each reference color (forming a total of 9 trios) and another one simply containing all the information to ease the analysis in some cases. Each trio corresponded to a step on the experiment. The first file contained the name (abstract class) chosen by the subject for that specific color (first step). The second file contained the *RGB* coordinates of all the colors chosen by the subject during the second step, and the third file contained the color chosen by the subject on the final step. This division can be justified because of the use of different



(a) Example of the first window of each step.



(b) Example of the second window of each step.



(c) Example of the third window of each step.

Fig. 3.5 Example windows of 1 of the 9 steps for experiment 4. The subject had to classify and memorize the color in (a), select colors that could be classified using the same abstract class as in (a) in (b), and identify the reference color (a) in (c).

data types, where the first file of the trio we are using **strings** and for the remaining two we use **integers**, as well as files with different sizes due to the fact that, contrary to the previous experiments where each reference color got paired with a comparison color an a score was associated with it, on this one a reference color could be paired with several (or even none) comparison colors, leading to files from different subjects referring to the same reference colors having disparate values in distinct quantities.

In this experiment, the subjects provided us some color names that had to be compared to some sort of reference, so, it was decided that the colors on [22] would serve this purpose, since it was the most complete color list available online. Although containing a huge sample of color coordinates, it did not contain a label for every single *RGB* triplet. In order to cover all possible combinations, a simple script was created that received an *RGB* coordinate and attributed it a name, by calculating the minimum euclidian distance to any color that had a label associated with them.

Due to all the differences previously mentioned, a different script had to be created in order to analyze the data.

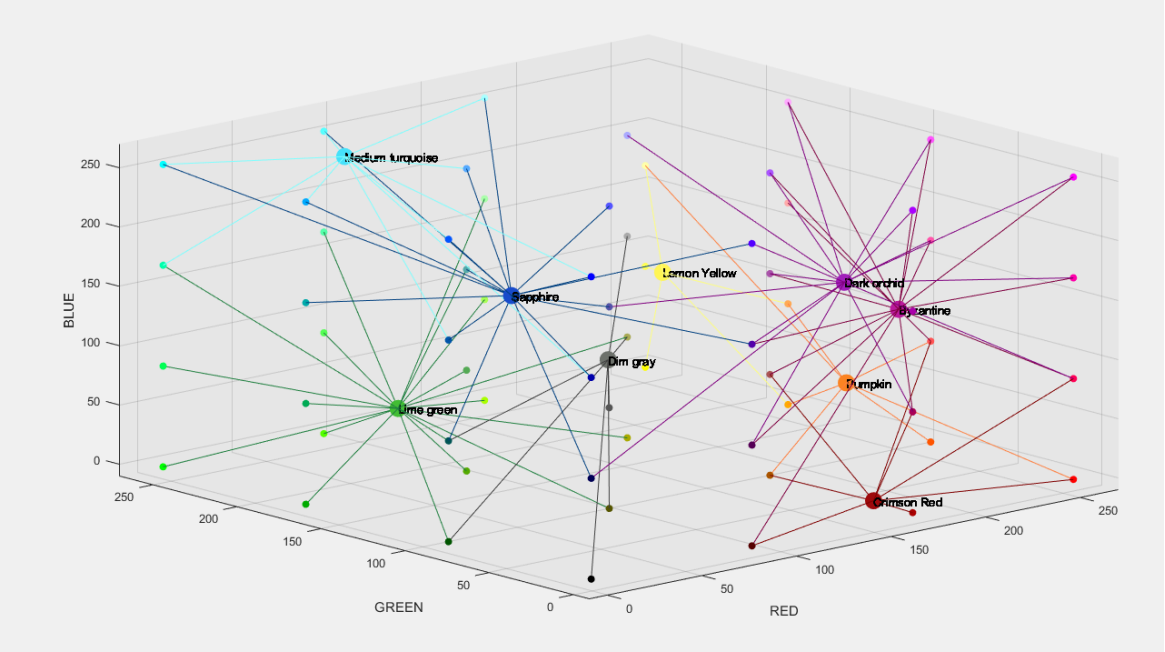


Fig. 3.6 All the colors obtained on experiment 4 plotted in the *RGB* space. Each small dot corresponds to a color picked by the subjects to fit in a specific color class. The bigger dots correspond to the mean color obtained from the colors connected to it by a line.

3.3.3 Results and discussion

The results of experiment 4 can be examined on Fig. 3.6 and Fig. 3.7. The color names that can be read there were attributed according to [22]. On Fig. 3.7, the first two columns refer to the reference color, which the subjects had to memorize in the beginning of each step. The first of these two columns has the *RGB* coordinates for each color name represented on the second one. The third and forth columns represent the color from memory, obtained from the last part of each step. In this case, the colors and coordinates represented refer to the ones chosen more often by the subjects. It is important to mention that all of the colors on these two columns were featured on the first or second part of each step of the experiment. The two final columns represent the mean color, obtained from the selection made by every subject for each step, along with the label obtained from the script mentioned on 3.3.2. The colors on the background of each cell containing a name represent the color obtained by the *RGB* triplets on the cells by their left.

The most common abstract classes attributed by the subjects can be seen on Fig. 3.8a, along with the percentage of subjects that impute that label.

Reference Color		Color from Memory		Average Color	
RGB coordinates	Name	RGB coordinates	Name	RGB coordinates	Name
32, 128, 64	Dark Spring Green	32, 128, 64	Dark Spring Green	66, 188, 52	Lime Green
128, 0, 64	Tyrian purple	170, 0 85	Jazzberry Jam	186, 27, 155	Byzantine
0, 64, 128	Dark Cerulean	0, 64, 128	Dark Cerulean	25, 72, 203	Sapphire
255, 128, 64	Mango Tango	255, 85, 0	International Orange	251, 130, 34	Pumpkin
255, 255, 128	Pastel Yellow	255, 255, 128	Pastel Yellow	251, 245, 86	Lemon Yellow
128, 0, 128	Patriarch	170, 0, 170	Dark Magenta	160, 33, 185	Dark orchid
128, 255, 255	Electric blue	128, 255, 255	Electric blue	74, 217, 250	Medium turquoise
128, 0, 0	Maroon	128, 0, 0	Maroon	160, 10, 10	Crimson red
64, 64, 64	Very Dark Gray	85, 85, 85	Davy Gray	106, 114, 109	Dim Gray

Fig. 3.7 Table with all the results from experiment 4. The columns with numbers represent the *RGB* coordinates of every color named to their immediate right. The background of each cell containing a name represents the color obtained by the coordinates on its immediate left.

(a)

Names	%votes
Green	59.30%
Dark pink	22.20%
Blue/Dark Blue	40.70%
Orange	88.90%
Yellow	51.90%
Purple	48.10%
Light Blue	44.40%
Dark Red	59.30%
Gray	55.60%

Reference	Memory	Average
100	←	84.15
89.37	←	82.63
100	←	82.01
82.54	→	87.2
100	←	90.18
86.55	→	91.49
100	←	85..01
100	←	92.08
91.76	←	90.24

(b)

Fig. 3.8 (a) shows the most common abstract classes attributed by the subjects on experiment 4. (b) depicts a side-by-side comparison between reference, memory and average colors. The values contained in the boxes represent the color similarity value. The column in the middle shows what value is greater.

The results show that abstractly classifying a color influences its memory (Fig. 3.7). On cases where the selected abstract class is more specific (for instance "Dark red"), the majority of the subjects correctly identified the reference color. But on cases where the abstract class was more broad ("Orange") the majority chose a color different from the reference one.

The mean colors, represented on the rightmost two columns on Fig. 3.7, are empirically consistent to the reference colors (also visible on Fig. 3.8b). One can easily notice that although having a different shade, the colors fit the same "label" when comparing both reference and average color. The presence of the *RGB* coordinates on Fig. 3.7 enables one to have better understanding on how close or far apart the coordinates are in the *RGB* plane. The color similarity, in he *RGB* color space, obtained through an euclidian distance and

normalized to fit a scale from 0 to 100 is also portrayed in Fig. 3.8b. In this case, the closer the values are to 100, the more similar the two colors are. By looking at the numbers on Fig. 3.8b one can notice that all of the pairs have quite high similarity values, which can be empirically verified by looking at the colors side by side.

3.4 General discussion and conclusions

In summary, it seems that the *RGB* and $L_1L_2L_3$ color models were the ones that depict color differences closer to the way humans do, although a lot more work can be done in this field. The data also shows that most humans define the most common color on an object as the dominant color. In terms of color classification, it seems that abstractly classifying a color affects color memory, specifically when the “label" is not very specific, but, more often than not, humans recall colors correctly based on an abstract class imposed by them.

The conclusions we reached regarding the pre-processing of an image with a *Retinex* algorithm reveal that it has little to no influence on the final result, despite the positive results obtained on tables 3.3 and 3.4. This pre-processing also delays the acquisition of the final results, since the execution times have to take into account the application of the method, which, on the tests preformed on an ASUS K56C laptop, would increase the run time by roughly 1.5 seconds. This could easily be bypassed by a using a computer with better hardware, but the application of the method would still increase the run time. Despite being overlooked on this situation, we could still use *Retinex* pre-processing before preforming proto-object segmentation (Section 4.2.1).

Chapter 4

Implementation

4.1 Introduction

In this Chapter we describe the implementation of the algorithm, using the data obtained from the experiments on chapter 3. It also includes a performance study to determine if the proto-object segmentation provides acceptable results.

4.2 Choice of appropriate processing sequence and algorithms

4.2.1 Testing processing pipeline

In Section 3.2.5 we mentioned that using the Retinex method had little to no influence on the process of determining the dominant color of a specific object in a biologically plausible fashion. In spite of this, it does not completely remove the method from our investigation. To test its usability during proto-object segmentation we designed the pipeline shown on Fig. 4.1.

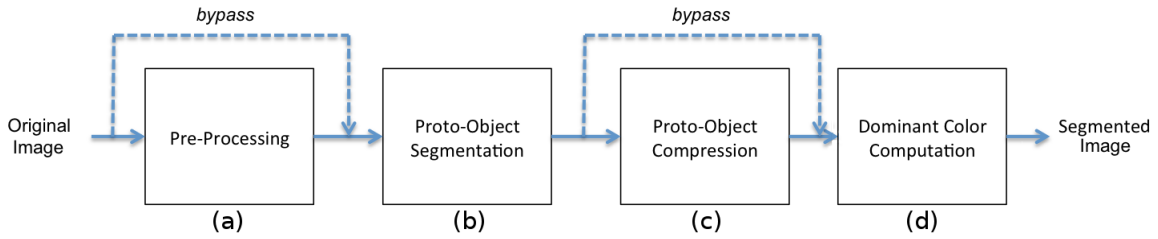


Fig. 4.1 Image segmentation pipeline. The original image could be either pre-processed (a) using the PDE-Retinex algorithm [19] or the Multiscale Retinex algorithm [20]. It was also possible to leave the image in its original state. The next step (b) used the algorithm on [18]. The output from the last step could suffer, or not, proto-object compression (c) using a method described in Section 4.2.1, leading to dominant color computation (d) which used a similar method to the one described on Section 2.2.4.

On the first step of the pipeline (Fig. 4.1-(a)) we used 2 retinex algorithms on the original images: the PDE-Retinex [19] and Multiscale Retinex [20]. Due to the presence of an online demo, all the manipulation was done using the demo from [19] and the demo from [20]. This pre-processing was utilized in order to remove the shading effect on the image, which in theory would prevent the superpixel segmentation from detecting an object and its shade as two different clusters, which is a major problem when using the proto-object segmentation algorithm.

The PDE-Retinex method only had a parameter that could be changed, the contrast threshold [19]. This variable (t) is used to eliminate the small intensity variations caused by the shading. The value of t could not be too large, which would cause a major loss in significant details, but it could also not be too low, which resulted on no clear changes relative to the original picture [19]. Therefore, it was decided that the value 4 was the optimal one to use for these tests. It is important to mention that even though the image is modified in order to better simulate how humans perceive it, it maintains the same mean and variance, which causes the original and processed image to have the same global contrast [19]. Contrary to the PDE-Retinex method, the Multiscale Retinex allowed substantially more freedom in terms of processing the final image, causing it to have 5 parameters: "three scales, and the percentage of saturation on each side of the histogram" [20]. This method is significantly slower comparing to PDE-Retinex, which should be taken into account when analyzing the results. The parameters were set to default since they provided the best results.

As for proto-object segmentation (Fig. 4.1-(b)), it uses the same algorithm described in Section 2.2.3, but limited to the ERS method (the reason for this choice can be seen on Section 2.2.3). In order to further eradicate the effect of false positives cause by the shading on objects, two consecutive proto-object segmentations were preformed on this step. By doing so, adjacent superpixels that were attributed similar colors due to being part of the same object could be merged on a singe one, therefore creating a more realistic proto-object. In order to do this, the algorithm on Section 2.2.3 was run using 'rgb' as **cs**, 'ERS' as the **segType**, 'MEAN' as **meanMed** and a **BW** of '3'. The tests were made with two different initial values for **numSeg**, '40' and '400'. These values are described as "initial" due to the fact that there were two consecutive segmentations, were the second one was preformed with the same parameters as described above except the **numSeg**, which was half of the initial value. This step was limited to the RGB color space due to the results obtained on Section 3.2.5. The same data also fixated the choice on 'mean' for the **meanMed**.

In order to further remove the illusion of multiple proto-objects associated to the same object, the step "Proto-Object Compression" (Fig. 4.1-(c)) was introduced in this pipeline. It consisted on a MATLAB script that took the segmented image obtained from the previous step and calculated which pair of colors had color distances under a specific threshold. The adjacent superpixels with similar colors fused into one, with its new color obtained by the application of the mean operand to the *RGB* coordinates of the pair. The downside of this step, besides the added processing time, seems to be the loss in contrast through out the whole image.

The final step (Fig. 4.1-(d)) consisted on dominant color computation. The segmented image obtained from the previous steps had the areas where the proto-objects were located analyzed. By calculating the mean color of every pixel contained inside the area previously mentioned, we obtained a new *RGB* coordinate, which was compared with the coordinate attributed by the previous methods. The dominant color was computed using the mean of the two colors. This step was avoided due to the time it took to analyze each image (approximately 30 minutes per image). Although useful, it still lacks a lot of optimization. The areas were

Table 4.1 Summarized agreement table. The methods featured on the column "Method" use the following template: [Pre-Processing]_[Compression]_[Number of proto-objects]. Pre-processing can replaced by “NR" (No Retinex), “WR" (Retinex), or “MSR" (Multi scale retinex). "Compression" can be replaced by “nc" (no proto-object compression), or “wc" (proto-object compression). Number of proto-objects can either be 40 or 400. The method also contains a reference for its specific table on Appendix C. The "Method" column refers to the table from which the values were taken. "JM" and "JFF" refer to the initials of the human experts whose analysis was compared to the method.

Method	JM % agreement	JFF %agreement	mean %agreement
NR_nc_40 (Table C.5)	6.96%	14.28%	10.62%
NR_nc_400 (Table C.6)	6.99%	36.37%	21.68%
NR_wc_400 (Table C.7)	9.61%	27.50%	18.55%
WR_nc_40 (Table C.8)	5.98%	4.93%	5.46%
WR_nc_400 (Table C.9)	7.72%	33.30%	20.51%
WR_wc_400 (Table C.10)	6.18%	21.26%	13.72%
MSR_nc_400 (Table C.11)	6.88%	31.27%	19.07%

filled with the newly obtained color and compared with the empirical segmentation, whose results can be seen in Section 4.2.2.

4.2.2 Preliminary testing results

The data on table 4.1 reveals that using a segmentation with a larger number of proto-objects is more in agreement with human experts. The mean %agreement was higher for an image with a large number of proto-objects, either using Retinex pre-processing or not. In fact, the highest value for % agreement corresponds to a method that used no pre-processing of any sort. Another important detail is the fact that proto-object compression did not improve the concordance, which reveals that this step can be overlooked, although it is important to mention was not properly optimized, which could justify the not so good results.

Finally, in these tests it was found that superpixel computation/proto-object segmentation decreased processing times significantly when comparing to applying the full pipeline to the image considering all individual pixels, even taking into account segmentation processing time, which represents an additional advantage of using proto-objects.

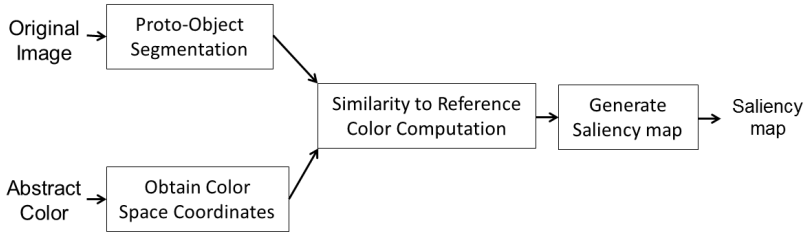


Fig. 4.2 Algorithm pipeline for the implementation. The "original image" and "abstract color" are input by the user. The algorithm segments the image into proto-objects and attributes them a dominant color. The abstract color (reference color) is analyzed and given *RGB* coordinates. The similarity to reference color is computed cy comparing the dominant color of each proto-object to the reference. The saliency map is then generated as an output.

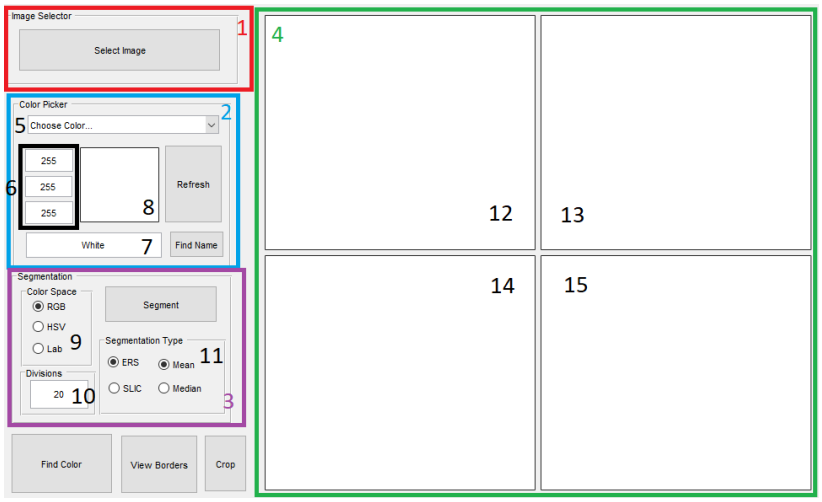


Fig. 4.3 GUI main window with colored shapes to indicate different functionalities. The red square (1) represents the image selection step, followed by the blue area (2) which allows the selection of the reference color. Next, the purple rectangle (3) allows the proto-object segmentation and finally, the green area (4), displays the results through out all the previous steps. The remaining numbers identify specific functions of each area: (5) to (7) allow the user to choose the reference color, which will appear on (8), (9) to (11) determine the proto-object segmentation parameters, and (12) to (15) display the results of the different steps of the process.

4.3 Definitive algorithm pipeline and respective implementation

Fig. 4.2 depicts the pipeline for this implementation. The main window of the application can be seen on Fig. 4.3. The application was designed based on the results from 3.2.5. The purpose of this script was to simulate all the processes that allows the system to successfully identify an object based solely on a reference color input by the user.

The red area (Fig. 4.3-(1)) represents the button where the user would select the image they want analyzed. Every image file is accepted, but it only works with small resolution

pictures in the *.png* and *.jpg* formats. The maximum image resolution for this build is 640×480, which was set to this value due to issues associated to proto-object segmentation, making it slower and, sometimes, even crashing MATLAB.

The blue rectangle is the reference color selection area (Fig. 4.3-(2)) where the user can choose from one of three methods to select the desired color. The first consists of a list that can be accessed through the pop-up menu (Fig. 4.3-(5)). This list contains 746 different colors, whose names and values were based on the ones found in [22]. The selected color can be previewed on the white square on Fig. 4.3-(8). The user may opt to input the *RGB* coordinates of the color, which can be done by filling in the proper text boxes visible in Fig. 4.3-(6). By inputing the values and clicking the "Refresh" button the user will be able to see a preview of the color on Fig. 4.3-(8). In order to prevent the user from inserting disparate values, a simple exception handling method was implemented, where each value input outside of the range of 0 to 255 was treated as 255. Finally, the user may decide to manually write the color name, which can be done through the window on Fig. 4.3-(7). Although, it is important to mention that the latest version will only accept colors whose name is featured on the list aforementioned. Just like before, the user will be able to see a preview of the selected color on the window in Fig. 4.3-(8).

Moving to the purple rectangle (Fig. 4.3-(3)), it consists on the proto-object segmentation part of the process (Section 2.2.3), whose algorithm was described on Section 4.2. On Fig. 4.3-(3) one can see that this area has 3 separate bounding boxes that can be filled in any order. The "Color Space" bounding box (Fig. 4.3-(9)) lets the user the selection of the color difference method that allows superpixel clustering. The text box on Fig. 4.3-(10) recognizes the maximum number of proto-objects to be formed in the segmentation process. It has no maximum value that can be input, but one should remember that the higher the value, the more time spent on this particular step. Finally, the "Segmentation Type" bounding box (Fig. 4.3-(11)) allows the user to chose between 'ERS' and 'SLIC' as segmentation methods (one again emphasizing that 'SLIC' only works on Windows OS) and the operand used for superpixel coloring (Mean or Median). To prevent errors, the application had default values

for all these parameters. Also, the buttons on each bounding box are mutually exclusive to avoid the selection of multiple values for the same parameter.

Lastly, the area indicated by a green rectangle (Fig. 4.3-(4)) displays the different results from the original image to the final result that can either be the original image with the desired color highlighted, or a crop containing the objects whose dominant color resembles the reference one. The first window on Fig. 4.3-(12) displays the original image chosen from Fig. 4.3-(1). The second window (Fig. 4.3-(13)) illustrates the result of proto-object segmentation preformed after clicking "segment" in Fig. 4.3-(3). After that, we immediately obtain the result of saliency mapping the reference color (chosen in Fig. 4.3-(2)) on Fig. 4.3-(14). In this case, the lighter the shade of gray filling an area, the more similar the dominant color of that proto-object is to the reference color. An analogous reasoning can be applied to darker shades of gray, that represent colors very different from the reference one. The original image with highlighted objects consists on the final result, represented on Fig. 4.3-(15). In order for the highlight to be salient, it is always filled in with the opposite color compared to the reference one. This way, the desired objects can be easily identified.

It is important to mention that, for debugging purposes, the application always identified a color on the image that was the most similar to the reference one, even if there were no colors that resemble it on the picture. For instance, if one chose "Dark blue" as the reference color for a picture containing only shades of red, the application would highlight a shade of red as being similar to "Dark blue", which, even though it is the closest according to our color dissimilarity measure, it is not an acceptable result due to the the colors belonging to clearly distinct abstract classes.

Some concrete examples of results obtained through the application can be seen on Section 4.4

4.4 Results on image sequence processing

The human experts were asked to manually segment a set of images into proto-objects, with a limit of 50 proto-objects per picture. These segmentations were compared to the results

obtained from the pipeline on Section 4.2.1. Using the proto-objects segmented by each human expert as reference, the percentage of agreement was calculated, by comparing them to the segmentations provided by the methods. These results were depicted on Table 4.1.

The 4 images on the first column of Fig. 4.4 represent 4 frames from an 8 second video. The images were then analyzed using the pipeline described in Section 4.3. The target color was 'Green', which the application associated with the *RGB* coordinates $(0, 255, 0)$. The results of the search can be seen on Fig. 4.4, column 4. It is important to mention that the algorithm only highlights the objects whose dominant color is the closest to the reference color, in this case, 'Green'. The images were individually analyzed using an ASUS K56C laptop running the unoptimized methods using MATLAB, resulting in an average processing time of 1.207 seconds. The results depicted on row (d) are acceptable, although one of them has an odd shape. This can be justified by looking at the second row of (b) and verifying that the proto-object segmentation algorithm considered the plushie as being two proto-objects instead of one. This effect was caused by its shadow on the left, causing a color difference high enough for the superpixel cluster to consider two separate superpixels. Due to the fact that the algorithm only considers the one proto-object whose color is the closest to the reference color, only one area in the image was selected. The results regarding the other frames seem empirically acceptable.

4.5 General Discussion

The image segmentation pipeline revealed that Retinex pre-processing did not provide better results when applied before proto-object segmentation. Instead, a solution with a great number of proto-objects and no Retinex pre-processing of any sort was considered the closer to human analysis. The implementation of the algorithm also provided acceptable results when used to identify 'Green' objects on several images.

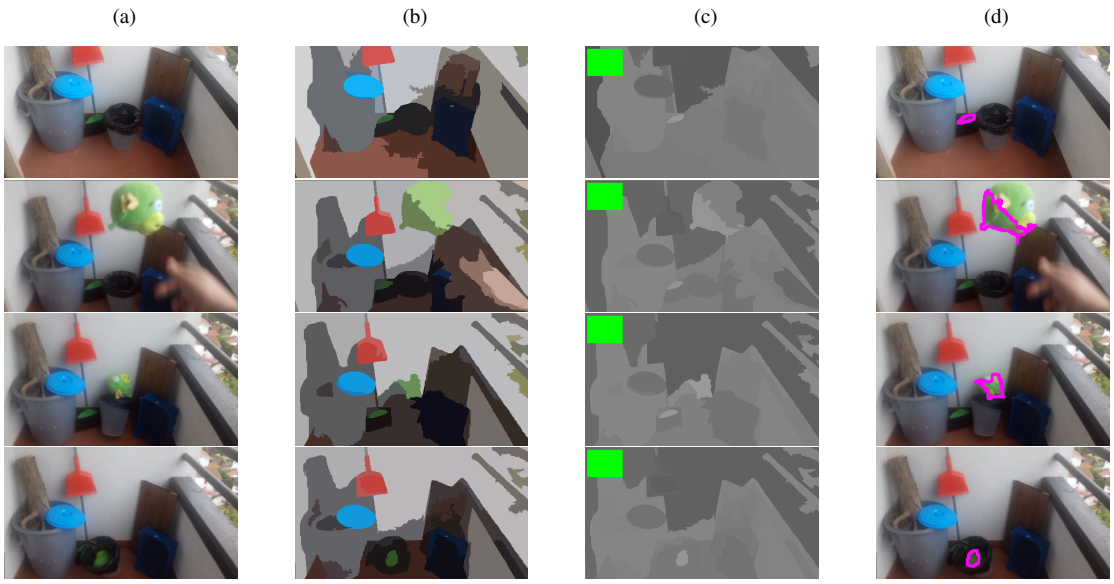


Fig. 4.4 Steps from frame analysis using the algorithm. This analysis was preformed using 'Green' as the reference color. The first column (a) represents the original frames. The second column (b) shows the result of proto-object segmentation using 200 maximum proto-objects. Column (c) depicts the saliency map (and the reference color in the top left color), with areas filled with a lighter shade of gray representing proto-objects whose color is most similar to 'Green'. The final column (d) represents the area on the original picture where the color 'Green' is more salient.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this work we carried out a series of experiments that allowed the gathering of data regarding how humans discriminate color, how they describe it and how they compare it with other sets of colors. The results revealed that the *RGB* color model proved to be the most concordant with the subjects, although some more data is required to assess the best color model to do so. The implementation of the algorithm used the data previously mentioned to provide the more human-like results, which were found to exhibit a satisfactory performance. The image segmentation pipeline revealed that subjects were more concordant when the number of proto-objects segmented was higher, without pre-processing the image using the Retinex method. The algorithm implementation was not optimized, which preclude real-time performance as is. However, an implementation using C in the future is expected to substantially improve processing times.

5.2 Future Work

In terms of future work there are several new tasks that can be implemented, namely the optimization of the algorithm for color search, the usage of the $L_1L_2L_3$ color space instead of the *RGB* to represent color in the algorithm, the usage of the angular distance as a

color dissimilarity measure on the data collected from the first 3 experiments, and finally, implement the mechanisms on section 1.3 on the CASIR-IMPEP attentional middleware.

References

[1] E Bruce Goldstein and James Brockmole. *Sensation and perception*. Cengage Learning, 2016.

[2] Gerald H Jacobs. The distribution and nature of colour vision among the mammals. *Biological Reviews*, 68(3):413–471, 1993.

[3] David E. Lewis, Joel Pearson, and Sieu K. Khuu. The Color 'Fruit': Object Memories Defined by Color. *PLoS ONE*, (8), 2013.

[4] M Sarifuddin and Rokia Missaoui. A new perceptually uniform color space with associated color similarity measure for content-based image and video retrieval. In *Proc. of ACM SIGIR 2005 workshop on multimedia information retrieval (MMIR 2005)*, pages 1–8, 2005.

[5] Paul Kay and Terry Regier. Resolving the question of color naming universals. *Proceedings of the National Academy of Sciences*, 100(15):9085–9089, 2003.

[6] Joao Filipe Ferreira and Jorge Dias. Attentional mechanisms for socially interactive robots–a survey. *IEEE Transactions on Autonomous Mental Development*, 6(2):110–125, 2014.

[7] Maurizio Corbetta and Gordon L. Shulman. Control of goal-directed and stimulus-driven attention in the brain. *Nature Reviews Neuroscience*, 3:201–215, March 2002.

[8] Stefanie I Becker, Christian Valuch, and Ulrich Ansorge. Color priming in pop-out search depends on the relative color of the target. *Frontiers in psychology*, 5, 2014.

[9] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.

[10] Elisa Drelie Gelasca, Danko Tomasic, and Touradj Ebrahimi. Which colors best catch your eyes: a subjective study of color saliency. In *Fisrt International Workshop on Video Processing and Quality Metrics for Consumer Electronics*. Citeseer, 2005.

[11] Pablo Lanillos, João Filipe Ferreira, and Jorge Dias. Designing an Artificial Attention System for Social Robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4171–4178, Hamburg, Germany, 2015.

[12] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on*, pages 1597–1604. IEEE, 2009.

[13] Convert from hsv to rgb color space. https://www.mathworks.com/help/images/convert-from-hsv-to-rgb-color-space.html. Date accessed: 06-08-2017.

[14] James Bruce, Tucker Balch, and Manuela Veloso. Fast and inexpensive color image segmentation for interactive robots. In *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 3, pages 2061–2066. IEEE, 2000.

[15] Theo Gevers and Arnold WM Smeulders. Color-based object recognition. *Pattern recognition*, 32(3):453–464, 1999.

[16] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. Entropy rate superpixel segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2097–2104. IEEE, 2011.

[17] 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.

[18] Chen-Ping Yu, Dimitris Samaras, and Gregory J Zelinsky. Modeling visual clutter perception using proto-object segmentation. *Journal of vision*, 14(7):4–4, 2014.

[19] Nicolas Limare, Ana Belén Petro, Catalina Sbert, and Jean-Michel Morel. Retinex poisson equation: a model for color perception. *Image Processing On Line*, 1:39–50, 2011.

[20] Ana Belén Petro, Catalina Sbert, and Jean-Michel Morel. Multiscale Retinex. *Image Processing On Line*, pages 71–88, 2014.

[21] Rensis Likert. A Technique for the Measurement of Attitudes. *Archives of Psychology*, pages 1–55, 1932.

[22] ColorHexa. ColorHexa color encyclopedia, 2012.

[23] Joost Van de Weijer, Theo Gevers, and J-M Geusebroek. Edge and corner detection by photometric quasi-invariants. *IEEE transactions on pattern analysis and machine intelligence*, 27(4):625–630, 2005.

[24] Investopedia - p-value. http://www.investopedia.com/terms/p/p-value.asp. Date accessed: 10-09-2017.

Appendix A

Statistical Analysis of Experimental Data

The Spearman's rank correlation coefficient ρ is a statistical measure that calculates the statistical dependence between the ranking of two variables (rank correlation). It is a measure that determines how consistent two variables can be. The value of ρ ranges from -1 to 1, where the first one represents a situation where the ranks are the complete opposite of each other, and 1 occurs when the ranks follow the exact same order. Spearman's ρ was obtained using eq. A.1, where n is the size of the sample and d_i represents the difference between the two ranks of each observation.

$$\rho = 1 - \frac{6\Sigma d_i^2}{n(n^2-1)} \tag{A.1}$$

The p-value represents the level of significance of a statistical hypothesis test, representing the probability of the occurrence of a given event. It is used as a mean to provide the smallest level of significance that can reject the null hypothesis. Smaller p-values indicate that the evidence is in favor of the alternative hypothesis. The p-value was obtained using the t distribution [24].

Table A.1 Example ranking table.

Color pair	Color Space 1		Color Space 2		Subject	
	Score	Rank	Score	Rank	Score	Rank
A	100	3	7	1	10	3
B	98	2	8	2	9	2
C	90	1	9	3	8	1

Table A.1 depicts an example that demonstrates the comparison between ranks and scores. One can easily notice that the absolute score error between color space 1 and subject is

significantly higher than the one calculated between color space 2 and subject, however, the rank correlation (ρ) has an higher value on the second case due to the order on which the ranks appear: on "Rank - color space 1" one can see that the values follow a descending order, something that can also be perceived on the column "Rank - subject". On the other hand, the values on "Rank - color space 2" follow an ascending order, reflecting on a negative ρ when calculated comparing to "Rank - subject". In fact, the relation in terms of sorting, which demonstrates priority, is more important than the score, although a minimum score error can also be a desirable result. Considering what was previously stated, despite "Color Space 2" having a smaller error compared to "Subject" (the secondary objective), "Color Space 1" and "Subject" constitute a better result, due to having a similar ranking, which is our main goal.

Appendix B

Normalized Color Spaces – Additional Information

Normalized RGB color space

Considering a color whose RGB coordinates are (R,G,B) we can obtain the Normalized RGB channels (nR,nG,nB) by:

$$S = R + G + B$$

$$nR = \frac{R}{S}$$
$$nG = \frac{G}{S}$$
$$nB = \frac{B}{S}$$

HCL color space

Considering a color on the RGB space represented as (R,G,B) we calculate:

$$M = Max(R,G,B)$$

$$m = Min(R,G,B)$$

Where $Max(R,G,B)$ is the component with the highest value among (R,G,B), and $Min(R,G,B)$ is the component with the lowest value.

$$L = \frac{Q.M + (1-Q).m}{2}$$

Where $Q = e^{\alpha\gamma}$ is a "parameter that allows a tuning of the variation of luminosity between a saturated hue (color) and a hue containing a great amount of white, with $\alpha = (\frac{m}{M.Y_0})$ and $Y_0 = 100$" [4]. The correction factor $\gamma = 3$ coincides with the one used for the Lab space.

C can be defined through a combination of the 3 channels of an RGB coordinate (red-green, green-blue and blue-red), multiplied by the Q parameter [4] previously defined:

$$C = \frac{Q.(|R-G| + |G-B| + |B-R|)}{3}$$

And

$$H = arctan(\frac{G-B}{R-G})$$

The value of H should be expressed on degrees and not radians, and varies between -90 and 90. However, this value has to vary from -180 to 180, so, a few if clauses have to be implemented to guarantee that H is expressed on this scale [4]:

- if $((R-G<0)$ and $(G-B) \geq 0)$, then $H = 180 + H$
- if $((R-G<0)$ and $(G-B)<0)$, then $H = 180 - H$

$L_1L_2L_3$ color space

Considering a color whose RGB coordinates are (R,G,B) we can obtain the $L_1L_2L_3$ channels by [15]:

$$L_1 = \frac{(R-G)^2}{(R-G)^2 + (R-B)^2 + (G-B)^2}$$

$$L_2 = \frac{(R-B)^2}{(R-G)^2 + (R-B)^2 + (G-B)^2}$$

$$L_3 = \frac{(G-B)^2}{(R-G)^2 + (R-B)^2 + (G-B)^2}$$

If $R = G$ we have:

$$L_1 = \frac{(R-R)^2}{(R-R)^2 + (R-B)^2 + (R-B)^2} = 0$$

$$L_2 = \frac{(R-B)^2}{(R-R)^2 + (R-B)^2 + (R-B)^2} = \frac{(R-B)^2}{2 \times (R-B)^2} = \frac{1}{2}$$

$$L_3 = \frac{(R-B)^2}{(R-R)^2 + (R-B)^2 + (R-B)^2} = \frac{(R-B)^2}{2 \times (R-B)^2} = \frac{1}{2}$$

In the same degree, if we have $R = B$:

$$L_1 = \frac{(R-G)^2}{(R-G)^2 + (R-R)^2 + (G-R)^2} = \frac{(R-G)^2}{2 \times (R-G)^2} = \frac{1}{2}$$

due to the fact that $(R-G)^2 = (G-R)^2$.

$$L_2 = \frac{(R-R)^2}{(R-G)^2 + (R-R)^2 + (G-R)^2} = 0$$

$$L_3 = \frac{(G-R)^2}{(R-G)^2 + (R-R)^2 + (G-R)^2} = \frac{(G-R)^2}{2 \times (G-R)^2} = \frac{1}{2}$$

Analogously, if $G = B$ we have $L_1 = \frac{1}{2}$, $L_2 = \frac{1}{2}$ and $L_3 = 0$.

In case the three components are equal ($R = G = B$) the result would be a division by 0, but no matter what denominator we choose, the coordinates for $L_1 L_2 L3$ would be the same.

Appendix C

Experimental Data

C.1 Behavioral Studies

Tables C.1, C.2, C.3 and C.4 contain experimental data for Experiments 4, 1, 2 and 3.

In Tables C.3 and C.4 the column "Method" specifies the method used to obtain the values compared to the median participant. The norm used to discern the methods follows the template: [Pre-processing]_[statistical method]_[color_space]. On this case "Pre-processing" either is "WR" (with Retinex pre-processing) and "NR" (without pre-processing). "Statistical_method" refers to the operand used on the method (mean, median or mode) and "color_space" specifies the color space used to calculate the color differences using the metric specified in "metric" (E - Euclidian; WE - Weighted Euclidian). The Retinex processing used the demo from [19]. The line filled with green represents the method and metric that scored the highest ρ, and the line highlighted in cyan refers to the method and metric that scored the minimum mean error. The highlight in red represents the version of the method and metric highlighted cyan but with no Retinex pre-processing.

C.2 Implementation Testing

This section refers to tables C.5, C.6, C.7, C.8, C.9, C.10 and C.11.

Table C.1 Abstract classification for the reference color provided by the subjects on experiment 4. Some of the colors suffered a literal translation from Portuguese into English, whilst others already had a proper translation.

Dark Spring Green	Tyrian Purple	Dark Cerulean	Mango Tango	Pastel Yellow	Patriarch	Electric Blue	Maroon	Very Dark Gray
green	dark pink	dark blue	salmon	light yellow	pink	light blue	red	gray
green	magenta	dark blue	orange	yellow	purple	light blue	dark red	dark gray
green	dark pink	blue	orange	light yellow	light purple	baby blue	brick red	dark gray
dark green	dark violet	neutral blue	light orange	light yellow	violet	cyan	dark red	gray
dark green	dark magenta	dark blue	orange	light yellow	lilac	cyan	dark red	dark gray
dark green	dark pink	ocean blue	orange	yellow	dark pink	baby blue	dark red	dark ash gray
green	burgundy	dark blue	orange	worn-out yellow	violet	cyan	dark red	dark gray
grass green	egg plant	navy blue	mate orange	mustard yellow	violet	neon blue	wine red	tar gray
lizard green	pink	dark blue	orange	light yellow	lilac	turquoise	brownish red	gray
dark green	violet	dark blue	orange	light yellow	purple	light blue	dark red	gray
green	bordeaux	blue	orange	light yellow	purple	cyan	reddish brown	gray
green	dark pink	dark blue	orange	yellow	dark pink	light blue	dark red	gray
dark green	magenta	dark blue	orange	light yellow	violet	light blue	dark red	dark gray
green	pink	blue	orange	yellow	purple	blue	red	gray
flag green	fuscia	blue	orange	yellow	purple	light blue	dark red	dark ash gray
green	dark lilac	navy blue	orange	light yellow	purple	cyan	dark red	gray
sporting green	grenat	blue	orange	yellow	purple	light blue	dark red	gray
green	darkest pink	blue	orange	yellow	purple	blue	terracota	dark gray
green	violet	dark blue	orange	yellow	purple	baby blue	dark red	gray
green	dark pink	dark blue	orange	yellow	dark pink	baby blue	dark red	dark green
green	pink	blue	orange	yellow	purple	light blue	red	gray
green	wine red	dark blue	orange	yellow	crimson	turqouise	bordeaux	gray
green	dark pink	turquoise	orange	lime yellow	purple	light blue	dark red	gray
green	purple	blue	orange	yellow	purple	light blue	dark red	gray
dark green	pink	blue	orange	yellow	purple	light blue	dark red	gray
green	violet rose	blue	orange	yellow	lilac	baby blue	blood red	dark gray
dark green	lilac	blue	orange	light yellow	lilac	light blue	reddish brown	dark ash gray

Table C.2 Data regarding the results obtained in experiment 1. The column "Method" refers to the color space where the color differences specified by "metric" were calculated. The E on "metric" stands for euclidian and the WE stands for weighted euclidian. ρ represents Spearman's rank correlation. "Min Error" and "Max Error" represent the minimum and maximum error obtained when comparing the method with the median participant. The standard deviation is represented on the column identified as "stdev". The line filled with green represents the method and metric that scored the highest ρ, and the line highlighted in cyan refers to the method and metric that scored the minimum mean error.

Method	metric	ρ	Min Error	Max Error	Mean Error	stdev	Median Error	p-value
HSV	E	0.5390	0.11	81.11	41.46	22.71	44.44	4.08E-14
	WE	0.4916	0.33	83.17	42.91	24.49	48.42	1.00E-11
LAB	E	0.6222	0.33	79.94	52.17	18.18	57.17	2.43E-19
RGB	E	0.6818	0.33	66.69	27.12	14.75	27.07	3.34E-24
	WE	0.6855	0.33	63.5	27.47	14.93	26.80	1.56E-24
YUV	E	0.6748	0.33	75.35	44.89	16.51	47.15	1.43E-23
HCL	E	0.5495	0.33	95.59	45.40	26.18	47.63	1.07E-14
n*RGB*	E	0.5379	0.33	69.13	32.37	18.81	34.66	4.63E-14
	WE	0.5367	0.33	72.27	35.82	19.05	39.08	5.38E-14
L1L2L3	E	0.3945	0.06	95.67	18.99	24.07	10.06	7.88E-08
	WE	0.3354	0.10	95.67	22.27	23.77	15.16	5.31E-06

Table C.3 Data regarding the results obtained in experiment 2. Detailed explanation of this table can be found in the main text (Section C.1).

Method	metric	ρ	Min Error	Max Error	Mean Error	stdev	Median Error	p-value
NR_Mean_LAB	E	0.6276	0.60	79.61	52.10	20.65	57.43	8.38E-10
NR_Mean_YUV	E	0.3997	1.68	91.65	50.52	24.02	53.82	1.90E-04
NR_Mean_HCL	E	0.6526	0.01	88.37	42.52	22.21	42.52	1.12E-10
NR_Mean_HSV	E	0.6302	1.13	72.83	41.16	20.60	43.51	6.89E-10
	WE	0.6304	0.34	70.57	40.60	20.64	45.28	6.77E-10
NR_Mean_L1L2L3	E	0.4879	0.00	84.1	17.26	19.27	9.06	4.49E-06
	WE	0.5250	0.30	85.47	19.00	19.14	12.97	6.65E-07
NR_Mean_N*RGB*	E	0.5852	0.81	77.12	38.85	21.08	41.43	1.76E-08
	WE	0.5741	0.31	77.68	39.69	20.84	39.24	3.62E-08
NR_Mean_*RGB*	E	0.6853	2.59	63.67	27.02	15.91	27.13	5.92E-12
	WE	0.6731	0.31	62.31	26.20	16.19	25.26	1.85E-11
NR_Median_LAB	E	0.6267	1.60	78.78	51.95	20.53	57.19	9.01E-10
NR_Median_YUV	E	0.4049	0.66	93.91	50.41	24.18	53.60	1.57E-04
NR_Median_HCL	E	0.6475	0.66	88.45	42.55	21.98	42.55	1.71E-10
NR_Median_HSV	E	0.6294	0.83	72.09	41.12	20.41	44.47	7.30E-10
	WE	0.6208	0.62	70.14	40.55	20.46	44.43	1.41E-09
NR_Median_L1L2L3	E	0.4864	0.25	83.48	17.29	19.03	9.06	4.83E-06
	WE	0.5202	0.60	84.91	18.92	18.73	13.00	8.61E-07
NR_Median_N*RGB*	E	0.6146	2.60	78.94	38.64	20.93	40.92	2.24E-09
	WE	0.5872	2.58	79.22	39.50	20.66	37.23	1.53E-08
NR_Median_*RGB*	E	0.6789	2.04	66.47	27.82	15.95	27.55	1.09E-11
	WE	0.6625	0.20	65.07	26.49	16.22	24.63	4.79E-11
NR_Mode_LAB	E	0.6210	1.88	79.24	51.72	20.40	55.93	1.39E-09
NR_Mode_YUV	E	0.4152	0.64	95.28	50.61	24.08	52.51	1.06E-04
NR_Mode_HCL	E	0.6656	0.53	87.82	42.59	21.93	41.60	3.63E-11
NR_Mode_HSV	E	0.6262	0.68	71.37	40.57	20.05	43.66	9.40E-10
	WE	0.6180	2.28	70.48	39.99	20.03	44.52	1.74E-09
NR_Mode_L1L2L3	E	0.4576	0.20	86.58	17.48	20.54	7.99	1.83E-05
	WE	0.4832	0.50	87.7	19.11	20.38	12.72	5.63E-06
NR_Mode_N*RGB*	E	0.6307	3.93	79.62	39.27	20.51	41.31	6.61E-10
	WE	0.6110	3.94	79.87	40.16	20.24	39.35	2.91E-09
NR_Mode_*RGB*	E	0.6774	1.30	67.42	28.07	15.99	26.94	1.25E-11
	WE	0.6586	0.15	66.02	26.71	16.30	24.91	6.71E-11
WR_Mean_LAB	E	0.6212	0.04	79.98	52.15	20.80	57.44	1.37E-09
WR_Mean_YUV	E	0.4050	1.82	92.87	50.40	20.85	53.70	1.57E-04
WR_Mean_HCL	E	0.6490	0.02	88.53	42.55	22.32	42.16	1.52E-10
WR_Mean_HSV	E	0.6294	1.04	72.77	41.03	20.65	43.47	7.30E-10
	WE	0.6295	0.45	70.70	40.47	20.70	44.39	7.27E-10
WR_Mean_L1L2L3	E	0.4954	0.18	84.19	17.22	19.25	9.06	3.11E-06
	WE	0.5236	0.31	85.55	18.97	19.12	13.00	7.18E-07
WR_Mean_N*RGB*	E	0.5783	0.95	77.51	38.91	21.18	41.58	2.76E-08
	WE	0.5684	0.40	78.04	39.75	20.93	39.39	5.19E-08
WR_Mean_*RGB*	E	0.6821	0.17	64.08	26.77	16.03	27.85	8.03E-12
	WE	0.6668	0.42	62.09	26.22	16.26	25.84	3.26E-11
WR_Median_LAB	E	0.6239	1.58	79.54	51.93	20.54	56.75	1.12E-09
WR_Median_YUV	E	0.4018	0.92	94.09	50.39	24.18	53.16	1.76E-04
WR_Median_HCL	E	0.6530	0.38	88.14	42.45	22.02	41.66	1.08E-10
WR_Median_HSV	E	0.6296	0.62	71.94	41.13	20.43	44.55	7.22E-10
	WE	0.6244	0.87	69.79	40.56	20.49	44.36	1.07E-09
WR_Median_L1L2L3	E	0.4889	0.09	83.91	17.12	18.98	9.06	4.28E-06
	WE	0.5224	0.64	85.29	18.83	18.66	13.00	7.64E-07
WR_Median_N*RGB*	E	0.6111	3.00	78.76	38.66	20.99	40.50	2.89E-09
	WE	0.5835	2.97	79.16	39.51	20.73	37.26	1.97E-08
WR_Median_*RGB*	E	0.6780	1.57	66.2	27.62	15.99	26.86	1.18E-11
	WE	0.6626	0.91	64.87	26.31	16.31	25.46	4.76E-11
WR_Mode_LAB	E	0.6155	2.00	79.92	51.77	20.61	56.37	2.10E-09
WR_Mode_YUV	E	0.3918	1.00	91.98	50.23	20.59	53.46	2.54E-04
WR_Mode_HCL	E	0.6705	0.62	88.67	42.49	21.92	41.66	2.35E-11
WR_Mode_HSV	E	0.6178	0.22	72.04	40.41	29.12	41.91	1.77E-09
	WE	0.6048	1.79	70.78	39.83	20.47	41.56	4.54E-09
WR_Mode_L1L2L3	E	0.4573	0.49	83.55	16.81	19.40	9.06	1.86E-05
	WE	0.5029	0.11	84.97	18.35	19.19	12.61	2.13E-06
WR_Mode_N*RGB*	E	0.6123	3.23	80.15	38.29	21.17	38.64	1.72E-08
	WE	0.5790	3.83	80.31	39.09	20.93	37.87	2.64E-08
WR_Mode_*RGB*	E	0.6462	0.89	67.86	27.72	16.40	26.06	1.91E-10
	WE	0.6371	0.03	66.66	26.45	16.47	24.39	4.00E-10

Table C.4 Data regarding the results obtained in experiment 3. Detailed explanation of this table can be found in the main text (Section C.1).

Method	metric	ρ	Min Error	Max Error	Mean Error	stdev	Median Error	p-value
NR_Mean_LAB	E	0.3947	0.09	78.07	49.97	23.02	58.06	1.08E-06
NR_Mean_YUV	E	0.2876	0.40	92.51	52.21	39.12	57.01	3.60E-04
NR_Mean_HCL	E	0.4643	2.31	94.03	44.68	24.63	46.05	7.06E-09
NR_Mean_HSV	E	0.4063	0.10	78.97	37.49	21.33	40.55	5.05E-07
	WE	0.4090	0.23	78.09	37.52	20.95	39.76	4.21E-07
NR_Mean_L1L2L3	E	0.3172	0.08	78.27	20.62	21.46	12.00	8.89E-05
	WE	0.3521	0.02	79.97	21.99	21.74	14.85	1.41E-05
NR_Mean_N*RGB*	E	0.3717	0.02	93.23	31.26	23.21	28.04	4.53E-06
	WE	0.3819	0.78	93.63	32.35	22.80	30.64	2.43E-06
NR_Mean_*RGB*	E	0.5532	0.41	68.39	31.60	17.33	32.37	1.72E-12
	WE	0.5398	0.41	70.15	32.14	17.37	34.37	7.08E-12
NR_Median_LAB	E	0.3998	0.02	77.82	49.86	22.95	57.89	7.77E-07
NR_Median_YUV	E	0.2868	1.16	92.47	52.07	25.19	57.33	3.72E-04
NR_Median_HCL	E	0.4687	1.71	94.06	44.71	24.66	47.01	4.92E-09
NR_Median_HSV	E	0.4022	0.08	78.47	38.24	21.42	40.69	6.64E-07
	WE	0.4052	0.39	77.37	38.31	21.02	40.12	5.44E-07
NR_Median_L1L2L3	E	0.3252	0.24	83.66	20.75	21.80	12.06	5.95E-05
	WE	0.3508	0.30	83.91	22.08	21.94	15.03	1.51E-05
NR_Median_N*RGB*	E	0.3838	0.40	93.03	30.76	23.12	27.90	2.16E-06
	WE	0.3883	0.80	93.43	32.06	22.81	30.82	1.63E-06
NR_Median_*RGB*	E	0.5636	0.02	68.16	31.46	17.35	32.96	5.56E-13
	WE	0.5496	0.07	69.98	31.99	17.39	34.03	2.54E-12
NR_Mode_LAB	E	0.3996	0.66	76.65	49.53	22.38	57.91	7.88E-07
NR_Mode_YUV	E	0.3179	0.94	91.16	50.56	25.50	57.15	8.60E-05
NR_Mode_HCL	E	0.4691	1.58	94.34	46.26	24.66	48.61	4.77E-09
NR_Mode_HSV	E	0.3765	0.63	84.47	37.22	21.35	40.47	3.39E-06
	WE	0.3787	0.22	82.98	37.22	21.05	40.14	2.97E-06
NR_Mode_L1L2L3	E	0.3452	0.69	81.37	21.19	22.60	12.00	2.06E-05
	WE	0.3709	0.16	81.84	22.81	21.92	15.02	4.75E-06
NR_Mode_N*RGB*	E	0.3955	0.24	93.02	30.11	22.75	25.65	1.03E-06
	WE	0.4013	0.09	93.46	31.43	22.79	30.87	7.05E-07
NR_Mode_*RGB*	E	0.5798	0.03	79.92	29.85	17.57	31.03	8.67E-14
	WE	0.5657	0.07	79.90	30.56	17.50	31.94	4.37E-13
WR_Mean_LAB	E	0.3976	0.11	78.07	49.99	23.01	58.03	8.97E-07
WR_Mean_YUV	E	0.2881	0.18	92.38	52.23	25.09	57.09	3.52E-04
WR_Mean_HCL	E	0.4651	2.24	94.03	44.71	24.64	46.42	6.58E-09
WR_Mean_HSV	E	0.4062	0.02	79.01	37.46	21.31	40.54	5.08E-07
	WE	0.4082	0.21	78.15	37.48	20.93	39.87	4.44E-07
WR_Mean_L1L2L3	E	0.3369	0.03	78.21	20.60	21.59	11.97	3.23E-05
	WE	0.3519	0.02	79.48	22.02	21.82	14.86	1.43E-05
WR_Mean_N*RGB*	E	0.3730	0.27	93.71	31.21	23.20	28.26	4.18E-06
	WE	0.3794	0.89	93.55	32.44	22.76	30.65	2.85E-06
WR_Mean_*RGB*	E	0.5579	0.83	68.09	31.50	17.26	32.46	1.03E-12
	WE	0.5398	0.78	69.90	32.05	17.30	34.22	7.07E-12
WR_Median_LAB	E	0.4078	0.08	77.86	49.88	22.93	58.03	4.59E-07
WR_Median_YUV	E	0.2939	0.30	92.41	52.08	25.21	56.91	2.71E-04
WR_Median_HCL	E	0.4636	2.14	94.06	44.63	24.75	46.44	7.45E-09
WR_Median_HSV	E	0.4054	0.38	78.74	38.25	21.40	40.49	5.37E-07
	WE	0.4057	0.61	77.68	38.33	20.98	40.00	5.28E-07
WR_Median_L1L2L3	E	0.3121	0.28	83.66	20.76	22.10	12.06	1.14E-04
	WE	0.3445	0.15	83.91	22.06	22.22	14.47	2.15E-05
WR_Median_N*RGB*	E	0.3993	0.26	92.99	30.51	22.98	27.82	8.02E-07
	WE	0.4008	0.43	93.43	32.06	22.62	30.41	7.29E-07
WR_Median_*RGB*	E	0.5694	0.24	67.94	31.34	17.29	32.80	2.87E-13
	WE	0.5562	0.14	69.80	31.89	17.35	33.90	1.25E-12
WR_Mode_LAB	E	0.3929	0.36	77.77	49.72	22.55	57.85	1.22E-06
WR_Mode_YUV	E	0.3051	1.50	90.85	50.81	25.49	54.95	1.61E-04
WR_Mode_HCL	E	0.4545	1.39	94.33	46.03	24.99	48.55	1.53E-08
WR_Mode_HSV	E	0.3164	0.30	86.52	38.41	21.98	40.56	9.24E-05
	WE	0.3182	0.20	85.55	38.34	21.69	40.28	8.45E-05
WR_Mode_L1L2L3	E	0.3454	0.09	90.15	21.32	22.65	12.06	2.04E-05
	WE	0.3472	0.18	90.27	22.80	22.25	15.22	1.84E-05
WR_Mode_N*RGB*	E	0.3981	0.19	93.09	30.70	22.96	27.55	8.67E-07
	WE	0.4093	0.11	93.56	32.71	22.44	31.99	4.12E-07
WR_Mode_*RGB*	E	0.5662	0.10	66.77	30.81	17.59	33.16	4.15E-13
	WE	0.5465	0.13	68.37	31.48	17.51	33.61	3.53E-12

Table C.5 The segmentation was done using a maximum of 40 proto-objects and did not use any type of pre-processing nor proto-object compression. "JM" and "JFF" refer to the initials of the human experts whose analysis was compared to the method.

Image	JM % agreement	JFF %agreement	mean %agreement
1	4.90%	20.00%	12.45%
2	1.27%	16.67%	8.97%
3	8.51%	18.18%	13.35%
4	5.71%	10.00%	7.86%
5	15.38%	16.67%	16.03%
6	4.55%	18.18%	11.36%
7	3.77%	7.69%	5.73%
8	15.79%	11.11%	13.45%
9	2.78%	10.00%	6.39%
Mean	6.96%	14.28%	10.62%

Table C.6 The segmentation was done using a maximum of 400 proto-objects and did not use any type of pre-processing nor proto-object compression. "JM" and "JFF" refer to the initials of the human experts whose analysis was compared to the method.

Image	JM % agreement	JFF %agreement	mean %agreement
1	5.88%	30.00%	17.94%
2	6.33%	50.00%	28.16%
3	8.51%	18.18%	13.35%
4	5.71%	30.00%	17.86%
5	5.13%	50.00%	27.56%
6	15.15%	36.36%	25.76%
7	5.66%	46.15%	25.91%
8	10.53%	66.67%	38.60%
9	0.00%	0.00%	0.00%
Mean	6.99%	36.37%	21.68%

Table C.7 The segmentation was done using a maximum of 400 proto-objects with proto-object compression. No pre-processing was executed. "JM" and "JFF" refer to the initials of the human experts whose analysis was compared to the method.

Image	JM % agreement	JFF %agreement	mean %agreement
1	9.80%	60.00%	34.90%
2	6.33%	33.33%	19.83%
3	8.51%	9.09%	8.80%
4	20.00%	30.00%	25.00%
5	0.00%	0.00%	0.00%
6	7.58%	9.09%	8.33%
7	13.21%	61.54%	37.37%
8	21.05%	44.44%	32.75%
9	0.00%	0.00%	0.00%
Mean	9.61%	27.50%	18.55%

Table C.8 The segmentation was done to an image pre-processed using the PDE-Retinex method (section 4.2.1) using using a maximum of 40 proto-objects. No proto-object compression was executed. "JM" and "JFF" refer to the initials of the human experts whose analysis was compared to the method.

Image	JM % agreement	JFF %agreement	mean %agreement
1	6.86%	20.00%	13.43%
2	1.27%	16.67%	8.97%
3	4.26%	0.00%	2.13%
4	2.86%	0.00%	1.43%
5	5.13%	0.00%	2.56%
6	0.00%	0.00%	0.00%
7	1.89%	7.69%	4.79%
8	31.58%	0.00%	15.79%
9	0.00%	0.00%	0.00%
Mean	5.98%	4.93%	5.46%

Table C.9 The segmentation was done to an image pre-processed using the PDE-Retinex method (section 4.2.1) using using a maximum of 400 proto-objects. No proto-object compression was executed. "JM" and "JFF" refer to the initials of the human experts whose analysis was compared to the method.

Image	JM % agreement	JFF %agreement	mean %agreement
1	8.82%	40.00%	24.41%
2	2.53%	50.00%	26.27%
3	8.51%	18.18%	13.35%
4	11.43%	20.00%	15.71%
5	0.00%	16.67%	8.33%
6	18.18%	45.45%	31.82%
7	9.43%	53.85%	31.64%
8	10.53%	55.56%	33.04%
9	0.00%	0.00%	0.00%
Mean	7.72%	33.30%	20.51%

Table C.10 The segmentation was done to an image pre-processed using the PDE-Retinex method (section 4.2.1) using using a maximum of 400 proto-objects, followed by proto-object compression. "JM" and "JFF" refer to the initials of the human experts whose analysis was compared to the method.

Image	JM % agreement	JFF %agreement	mean %agreement
1	6.86%	40.00%	23.43%
2	2.53%	33.33%	17.93%
3	6.38%	18.18%	12.28%
4	8.57%	10.00%	9.29%
5	0.00%	16.67%	8.33%
6	6.06%	9.09%	7.58%
7	9.43%	30.77%	20.10%
8	15.79%	33.33%	24.56%
9	0.00%	0.00%	0.00%
Mean	6.18%	21.26%	13.72%

Table C.11 The segmentation was done to an image pre-processed using the Multiscale Retinex method (section 4.2.1) using using a maximum of 400 proto-objects. No proto-object compression was executed. "JM" and "JFF" refer to the initials of the human experts whose analysis was compared to the method.

Image	JM % agreement	JFF %agreement	mean %agreement
1	6.86%	30.00%	18.43%
2	0.00%	16.67%	8.33%
3	8.51%	27.27%	17.89%
4	8.57%	20.00%	14.29%
5	5.13%	50.00%	27.56%
6	16.67%	54.55%	35.61%
7	5.66%	38.46%	22.06%
8	10.53%	44.44%	27.49%
9	0.00%	0.00%	0.00%
Mean	6.88%	31.27%	19.07%