TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Murat Birinci

Perceptual Approaches in Image and Video Analysis

Julkaisu 1482 • Publication 1482

Tampere 2017

Tampereen teknillinen yliopisto. Julkaisu 1482
Tampere University of Technology. Publication 1482

Murat Birinci

Perceptual Approaches in Image and Video Analysis

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Festia Building, Auditorium Pieni sali 1, at Tampere University of Technology, on the 7th of August 2017, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2017

Doctoral candidate:	Murat Birinci Computing and Electrical Engineering Tampere University of Technology Finland
Supervisor:	Prof. Moncef Gabbouj Computing and Electrical Engineering Tampere University of Technology Finland
Instructor:	Prof. Serkan Kiranyaz Electrical Engineering Qatar University Qatar
Pre-examiners:	Prof. Zygmunt Pizlo Psychological Sciences Purdue University USA
	Assoc. Prof. Chaker Larabi XLIM Laboratory University of Poitiers France
Opponent:	Prof. Vladimir Lukin Signal Reception, Transmission and Processing Kharkov Aviation Institute Ukraine

ISBN 978-952-15-3975-6 (printed)
ISBN 978-952-15-3982-4 (PDF)
ISSN 1459-2045

Abstract

Recent advances in digital technology enabled the use of multimedia in various fields of our lives. Education, health, security, entertainment, business and many other sectors started using all kinds of multimedia material for their benefits to provide better services. In order to utilize the full potential of such material and enable their effective consumption in those areas, accurate analysis and understanding of the multimedia content is essential. Content based multimedia analysis aims to provide this insight through various computer algorithms and extract relevant information to support different fields. When designing such algorithms, in order to lead to practical solutions, it is essential to keep in mind that both performance and efficiency are of significant importance. Considering the fact that humans have remarkable ability in analyzing visual content, this thesis presents algorithms for image and video analysis by taking the perspective of human visual perception.

The algorithms presented in this thesis follow the perceptual rules proposed by Gestalt Psychology, which suggests that our perceptions are based on the emergent properties that result from the organization of individual percepts. Such a stance is often overlooked – if not ignored – in content analysis algorithms, and the offered solutions are generally based on analyzing individual components only. This typically results in either inadequate or overcomplicated solutions. By following the perceptual organization rules defined by Gestalt Psychology, it has been shown in this thesis that content analysis can be performed in a significantly more efficient and effective manner. These improvements are revealed in miscellaneous topics, such as color content description, image segmentation, object recognition and video shot change detection.

The main contribution of this thesis is to demonstrate the significance of taking a perceptual standpoint in image and video content analysis. This significance can be examined through the benefits it brings in, namely the improvements in performance and efficiency. Performance improvements in this thesis are realized in the aforementioned fields, specifically by attaining more accurate characterization of the color composition of an image, more precise segmentation of the objects, higher accuracy in recognizing objects and higher accuracy in detecting shot boundaries in a video. Achieving such improvements via simple and lightweight algorithms without over complicating or over engineering the underlying problem proves the efficiency of the proposed algorithms. Algorithms presented in this thesis are evaluated according to both criteria, i.e. performance and efficiency, and it will be shown in the thesis that they achieve exceptional results when compared to the state of the art. In other words, describing the color content of an image, segmenting an image into meaningful objects, recognizing objects and detecting shot changes in a video are all successfully accomplished with minimal effort – just as we humans perform such tasks.

Preface

The work presented in this thesis has been carried out at Tampere University of Technology (Finland), Nokia Research Center (Finland) and during visit to Purdue University (USA).

First and foremost, I would like to express my gratitude to my supervisor Prof. Moncef Gabbouj for his support and guidance throughout my research whenever needed. Second, I would like to extend my gratitude to Prof. Serkan Kıranyaz for his technical, academic and personal supervision and confidence in me. I also would like to thank Prof. Edward Delp from Purdue University for his supervision while I was a visiting researcher in his group. This thesis wouldn't have been possible without any of you.

I would also like to thank Dr. Kemal Uğur for his guidance during my internship in Nokia Research Center which gave rise to one of the works presented in this thesis, and also for his personal support as a friend since the day I moved to Finland.

I am grateful to Prof. Fernando Diaz-de-Maria and Dr. Golnaz Abdollahian for their exceptional teamwork in our collaboration during my visit in Purdue University.

I am also thankful to all my colleagues that I have worked with throughout this thesis for all their help, support and friendship. It has been a privilege to know and work with all of you. I especially cannot thank enough to Dr. Esin Güldoğan, Dr. Stefan Uhlmann, Dr. Jenni Pulkkinen, Guanqun Cao and Dr. Uygar Tuna for always being there for me and turning my time during my studies into an enjoyable journey.

Finally, I would like to thank my parents for their endless support and belief in me, to whom I owe everything I am.

Tampere, May 2017

Murat Birinci

List of Figures

List of Tables

Abbreviations

AI	Artificial Intelligence
ANMRR	Average Normalized Modified Retrieval Rank
BG	Background
BRISK	Binary Robust Invariant Scalable Keypoints
DC	Dominant Color
DoG	Difference-Of-Gaussian
FG	Foreground
HCI	Human Computer Interaction
HSV	Hue Saturation Value
MP3	MPEG-1 Audio Layer 3
ORB	Oriented Brief
PROSAC	Progressive Sample Consensus
RANSAC	Random Sample Consensus
RBC	Recognition By Components
RGB	Red Green Blue
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization And Mapping
SLIC	Simple Linear Iterative Clustering
SURF	Speeded Up Robust Features
TRECVid	Text Retrieval Conference Video Retrieval Evaluation

Contents

List of Publications

This thesis is a compound thesis and is based on the following publications:

[P1] S. Kiranyaz, M. Birinci, and M. Gabbouj, "Perceptual Color Descriptor Based on Spatial Distribution: A Top-Down Approach," Image and Vision Computing, vol. 28, pp. 1309-1326, 2010.

[P2] M. Birinci and K. Ugur, “Interactive Image Segmentation Based on Superpixel Grouping for Mobile Devices with Touchscreen,” In Proceedings of IEEE International Conference on Multimedia & Expo (ICME), Chengdu, 2014, pp. 1-6.

[P3] M. Birinci, F. D. Maria, G. Abdollahian, E. J. Delp, M. Gabbouj, “Neighborhood Matching for Object Recognition Algorithms Based on Local Image Features,” In Proceedings of Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE), Sedona, AZ, 2011, pp. 157-162.

[P4] G. Abdollahian, M. Birinci, F. D. Maria, M. Gabbouj and E. J. Delp, “A Region-Dependent Image Matching Method for Image and Video Annotation,” 9th International Workshop on Content-Based Multimedia Indexing (CBMI), Madrid, 2011, pp. 121-126.

[P5] M. Birinci and S. Kiranyaz, “A Perceptual Scheme for Fully Automatic Video Shot Boundary Detection,” Signal Processing: Image Communication, vol. 29, pp. 410-423, 2014.

Chapter 1

Introduction

Humans have extraordinary ability to observe and interpret their environment. However, we do it so naturally and effortlessly that most of us take it for granted and do not even realize how complicated and challenging it actually is. Seeing a leaf slowly falling from a tree, hearing the sound of a car horn, recognizing the smell of our favorite food, or any trivial phenomenon occurring during our daily life in fact stems from a flurry of activity of our senses and brain. We use our senses to *see*, *hear*, *taste*, *smell* and *feel* things around us, yet our perception of our environment is more than simple transmission of these senses. What goes on behind the curtain, how our senses, our past experience, attention, interests etc. are analyzed by our brain to allow us *understand* our environment has always been part of the focus of human psychology – more specifically psychophysics, which is formally defined as the branch that quantitatively studies human perception and examines the reasons and relations behind it. Moreover, as we start to understand more about how our brain works, other fields, such as cognitive neuroscience, has also joined the quest of solving the mystery of perception. Together with the advances in computer science, computational models have been used in modeling perceptual theories [1]-[4]. Such models not only allow better quantitative analysis and evaluation of these theories, but also enable utilizing them in various areas such as artificial intelligence (AI) [5], computational photography [6], audio processing [7], robotics [8], human computer interaction (HCI) [9] etc. Ultimately, the main aim in all these areas is to simulate human perception so that they can lead to better engineering solutions. For instance, designing a better microphone, a better speaker or a better audio codec is a result of proper understanding of human the hearing and auditory perception. The well-known audio standard MP3 uses perceptual audio coding [10] in order to offer a better compression by reducing the file size without sacrificing the audio quality. Similarly, camera design is vastly influenced by the human visual perception where the lens, aperture and sensor strive to replicate the corresponding elements in the human eye (specifically, the cornea, the pupil and the retina, respectively). Digital video coding also resorts to visual perception in order to remove details which cannot be seen by the human eye [11]. Robotics, AI and HCI focus additionally on recognizing and comprehending the sensory input that allows them to *act* based on their inference. For example, when you search online for a video based on its content, this means the system needs to *know* what is inside all the videos that you are searching so that it can retrieve *related* ones. This requires answering questions such as “what is the video about” and “what is interesting in the video” and finding the relations between videos based on such questions, which undoubtedly requires a good understanding of both visual and auditory perception. Similarly, a self-driving car should also be able to localize the source of any incoming sound and recognize what sound it is – be it ambulance, a shouting pedestrian or another car’s horn, or realize any moving object in its field of view – be it other vehicles, pedestrians or bicycles. Considering how easily we humans perform such tasks, understanding human perception has always been at the core of such engineering designs.

1.1. Visual Perception

Visual perception refers to information processing that allow us to process and *understand* our surroundings from the information that we gain through our vision. However, as mentioned above, such understanding is not always the same as what is out there. The light rays reflecting from surfaces, going through our eyes and reaching our brain may be reconstructed as a completely different object than what is actually there. In other words, our perception can often be different than the physical world. Since we end up seeing things that actually are not there, many perceptual phenomena are often called optical illusions. However, even though the word *illusion* make it sound like a malfunction of the visual system, it is in fact an effective demonstration of how our visual perception works.

Figure 1-1 shows an example of how our perception differs from the measured physical reality. The phenomenon is referred as *simultaneous contrast*, where the perceived lightness of the patches are affected by their background. This is a simple demonstration of the fact that our perception is not a direct aggregation (or concatenation) of our senses. If it were, we would receive all the stimulus from those patches independently and see the patches on the same row with exactly the same lightness. However, their surroundings lead us to perceive them as different.

There have been many attempts to explain and model human visual perception – i.e. vision theories [12]. Each approaches the problem from a different perspective, but not all of them are able to explain the relativity and dependency we observe in Figure 1-1. Still, one particular theory stands out since its main motivation is built on such inter-relations among visual percepts. Gestalt Psychology (see Chapter 2) claims that our perception is more than the collection of individual percepts, and focuses on the emergent features that stem from their organization [13]. However, just like all the other vision theories, Gestalt Psychology is also a *theory*. It tries to explain why we see the world the way we do by various observations. Even though it provides us several descriptive principles, today we still lack a complete model and understanding of our visual perception. Yet, these theories and observations continue to inspire current technologies and innovations.

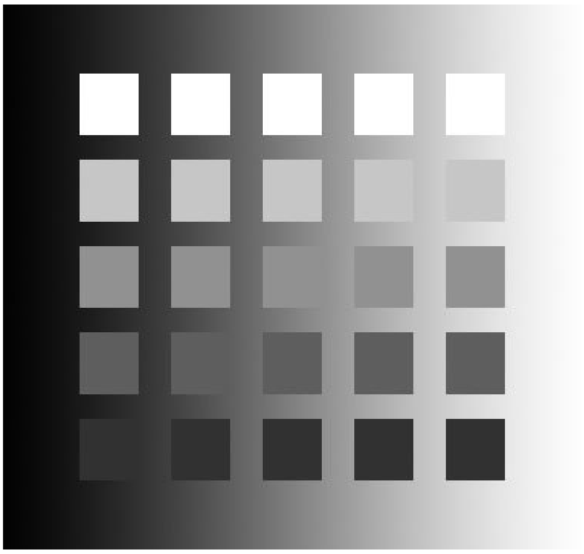


Figure 1-1. Illustration of simultaneous contrast. Patches on the same row have exactly the same intensity; however, they appear to get darker from left to right.

In recent years, algorithms utilizing Machine Learning techniques have dominated the field of image processing. These methods try to learn generic representations of input images for a given particular task. As we will see in Chapter 6, they may bring in certain advantages in certain cases. However, machine learning techniques are outside the scope of this thesis. The interested reader is referred to [14], [15] for further reading.

1.2. Objectives of the Thesis and Author’s Contributions

This thesis aims to demonstrate that image processing algorithms can significantly benefit from taking the perspective of human visual perception. Considering how efficient and successful we humans are in the tasks that we try to solve in image processing, substantial improvements can be achieved if we successfully reflect what we perceive to such algorithms. In this thesis, various image processing problems are tackled by designing solutions based on the principles of the aforementioned Gestalt Psychology. Following subsections summarize each area that is tackled together with the author’s contributions in solving the specific problem.

1.2.1. Color Perception

The aforementioned relativity we observed in simultaneous contrast is not specific to lightness, we can also observe similar behavior in color perception. That is to say, similar to Figure 1-1, our perception of a certain color is significantly affected from its neighborhood. Figure 1-2 illustrates a situation where the color of a tile appears (or perceived) entirely different when it is surrounded by different colors. This is a clear demonstration of the fact that we cannot consider color elements individually and independently from their surroundings. In [P1] a perceptual color descriptor is proposed that aims to characterize what we see in an image in terms of colors. It is clear that a competent color descriptor should take such spatial relationship into account in order to reflect our

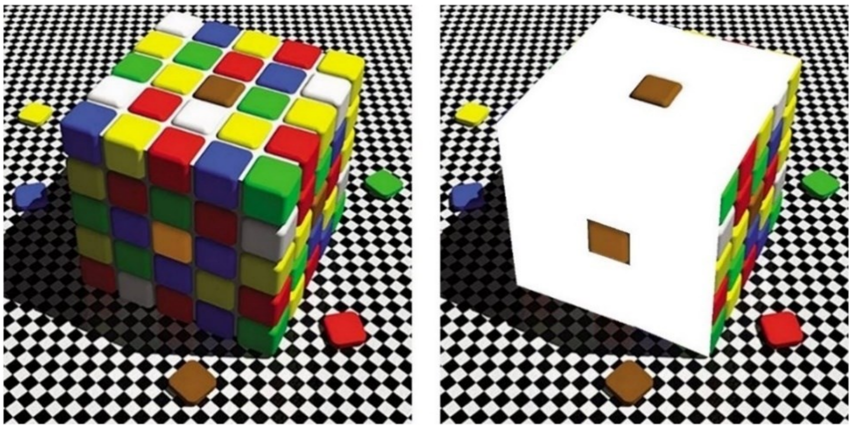


Figure 1-2. The two patches, despite having the exact same physical color values, are perceived to be different in the context. The lower patch is perceived more "yellow" on the left, whereas it has exactly the same color with the above "brownish" patch – which is revealed on the right when taken out of context.

visual perception. Otherwise, if colors are taken out of context and handled independently of their spatial distribution, what we actually perceive can never be accurately described by a feature. The author contributed to the design and implementation of the color descriptors and writing of the paper. The author planned and executed the comparative experiments.

1.2.2. Image Segmentation

Perception of basic visual features such as color, lightness and shape may be lying in the roots of our vision; however, when we look outside we do not see light rays entering our eyes, lines or colors. Similarly, when we look at an image, we do not see individual pixels. What we see are objects, surfaces, structures, and so on. So how do we go from basic features to objects? How do we form objects from those features? In fact, the situation we face in image processing is a lot like what our visual system faces, i.e. how we go from the simple visual input in our retinal receptors to a coherent visual world, and how we go from individual pixels to a structured image. Thus, a digital image represented merely with numerical values of pixels need to be organized into meaningful objects. There are potentially unlimited number of possible organizations, however what we perceive is typically only one of them. [P2] tackles the problem of image segmentation, i.e. partitioning the image into meaningful objects, from a similar perspective. Stemming from how humans group individual percepts into meaningful objects, it proposes a method for segmenting digital images into objects by grouping image pixels. The author contributed to the design of the proposed algorithm and writing of the paper. The author implemented the proposed algorithm and performed the experiments.

1.2.3. Object Recognition

Our daily lives involve continuous decisions we make and actions we take based on our surroundings. However, simply "seeing" what is around us is not sufficient for this purpose. We need to "understand" what is around us, give meaning to them, and finally decide our actions accordingly. In that sense, seeing objects is simply the initial step for this process and recognizing those object is the next step. Humans are exceptionally talented when it comes to recognizing objects. We can recognize multitude of objects around us with little effort, even if we see them from different angles, in different size or scale, or even when they are partially occluded. But how do we do this? Do we simply keep every possible view of an object in our memory? Or do we follow a more elaborate path? Recognizing an object from its parts has been one of the popular theories on object recognition. For example, a chair is perceived to be composed of four legs, a seat and a back. However, as we discussed above, our perception is never a simple registration of our senses. In addition to these parts, object perceptions also include spatial relations among those parts. A disassembled pile of legs, seat and a back does not give rise to the perception of a chair. Therefore, [P3] and [P4] propose methods on how image features can be organized in a similar way followed by humans to organize them in order to recognize objects. In [P3], the author contributed to the design of the algorithm. The author implemented the algorithm, performed the experiments and wrote the paper. In [P4], The author contributed to the design and implementation of the algorithm, experiments and writing of the paper.

1.2.4. Video Shot Change Detection

Our visual system takes in continuous flow of information as long as our eyes are open. We use this information to see our environment, recognize it and act upon it. However, our perception is not always about seeing things, sometimes it is rather about not seeing things. Every so often, a certain object is right in front of us, within our field of view, yet we still do not see it – or rather do not perceive it. The visual stimulus is presented to us, the light rays from the object reaches to our eyes. So why can we not perceive it? One should keep in mind that, in designing perceptual approaches, in addition to studying how we perceive things, it is also important to understand how we cannot perceive things. For instance, in [P5], by understanding the limitations of the human visual system on detecting changes, a method is proposed for detecting shot changes in a video. The author contributed to the design of the algorithm. The author implemented the algorithm, performed the experiments and wrote the paper.

1.3. Outline of the Thesis

Based on the above discussion, the thesis is organized as follows: In Chapter 2, Gestalt Psychology, which inspired most of the works in this thesis, is presented. In Chapter 3 color vision and perception is described by discussing the color descriptor proposed in [P1]. Chapter 4 addresses the problem of interactive image segmentation and how it is tackled in [P2]. Object recognition based on local image features is discussed in Chapter 5 together with the contributions of [P3] and [P4]. Chapter 6 examines our ability of change detection and how it is utilized in [P5]. Chapter 7 concludes the thesis.

Chapter 2

Gestalt Psychology

The Gestalt school of thought evolved from the research by Max Wertheimer, Kurt Koffka and Wolfgang Köhler in the beginning of the 20th century. Wertheimer published his famous monograph on *φ-motion* in 1912 [16] where he noted that we perceive motion when there is nothing more than a rapid sequence of individual sensory events – in contrast to the conventional view of apparent motion (commonly known as *β-motion*), where we see an object at several successive positions and motion is then "added" subjectively. In fact, the difference between *φ* and *β* motions is often unclear to most. In order to demonstrate the difference, consider the formation in Figure 2-1 where a number of black discs are distributed to form a large circle. Now, consider one of them is missing. When we change the location of the missing disc sequentially around the circle, we start noticing a motion. At lower speeds, what we notice is the disc adjacent to the missing location is moving to the empty space and leaving its own space empty. This is *β-motion* where we see the object at the beginning and end points which in return gives us the sensation of that object is moving. As the speed goes higher, we no longer see the discs as moving, they all appear stationary. Instead, we only see as if the white space moving around the circle. This is *φ-motion*. The interesting phenomenon here is that we cannot even describe the object that is moving around, we only have the sensation of motion. That's why Wertheimer also called this "pure motion" since it is not bound to any object. For a more detailed discussion on *φ* and *β* motions, the reader is referred to [17].

The *φ-motion* was the perception of a pure process, which could not be composed from more primitive percepts of a single object at multiple locations. In other words, the perceived motion was not added subjectively after the sensory registration of several spatiotemporal events but had its own characteristics and status. From this phenomenon, Wertheimer concluded that "structured wholes" are the primary units of our perception. This was the key idea of the new and revolutionary Gestalt Psychology.

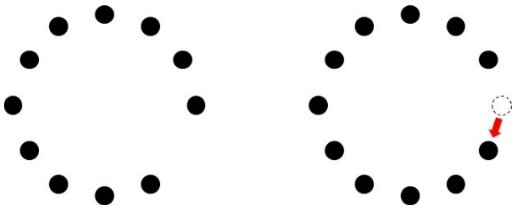


Figure 2-1. When the location of the missing disc is sequentially changed, we start observing φ and β motions depending on the speed of change.

“Gestalt” is a German word, which can be translated as “whole” – more in the sense of shape or form. What Gestalt Psychology suggests is that our perception of the whole is not simply the aggregation of individual stimuli, but is a distinct percept on its own which cannot be reduced to parts or even piecewise relations among them. *“The whole is different than the sum of its parts”* is the famous motto of Gestalt Psychology that conveys this very idea. To illustrate, consider the formation in Figure 2-2. where we see a disc in the middle of the black lines. Some may also perceive it as a white disc on top of four crossing lines. However, the disc does not exist. There is no closed curve forming a circle nor any other elements forming a disc. Yet, we still perceive it as such. This shows us that simple sensory inputs are not sufficient to explain perception.

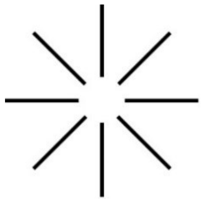


Figure 2-2. The figure has eight black lines on a white background. There is no disc in the figure, nor any circle to begin with. However, we perceive it as a white disc placed over straight lines.

Even though Figure 2-2 demonstrates the fact that our perception requires more understanding than the simple analysis of the parts, it does not tell us “why” or “how” it is so. How do we see something that is not there? What makes a disc appear in front of our eyes even though there is no such disc or circle there? Gestaltists used such examples as evidence in order to validate the *emergent properties* of the whole which are not possessed by any of its parts. In Figure 2-2 properties of a disc such as its diameter, area and circumference are not properties of the lines we see. We are able to talk about those properties due to the configuration of the lines when they are arranged in this specific manner. In other words, the organization in which the lines are arranged gives rise to an entirely different perception than the lines themselves. Such organization plays a key role in our perception. In fact, Gestalt psychologists were the first to realize the importance of perceptual organization and first to systematically study the properties that govern it. Typically, such organization is directly associated with perceptual grouping – often used synonymously. However perceptual grouping is simply one particular type of organizational property. Another crucial one is figure/ground organization, which basically is one of the most fundamental ways our perception simplifies a scene. In the next sub-sections will detail these organizational phenomena and discuss other vison theories in order to give a complete picture.

2.1. Perceptual Grouping – Prägnanz

After postulating the concept of Gestalt [1] – the whole – Wertheimer published another groundbreaking paper in 1923 [16] in pursuance of illuminating the fundamental principles behind what construes the “whole”. He studied perceptual grouping and investigated what factors are affecting perceived grouping of individual components. He started with constructing a scene from very basic visual elements and by varying the relations among them he proposed a set of principles that govern how various stimuli are perceived as belonging together.

Prägnanz – which in German means succinctness or pithiness – embraces all the other laws Wertheimer proposed and construes a basis for the whole Gestalt theory. It suggests that humans tend to form the *simplest* possible organization from their visual field. Consider Figure 2-3.A, where most people see five overlapping circles. Even though it is possible to break it apart as in Figure 2-3.B, our perception is attracted by the simpler solution in A and see overlapping circles instead. In order to clarify what is “simple” and what is not, a set of laws were described that allows us to predict our comprehension of perceptions.

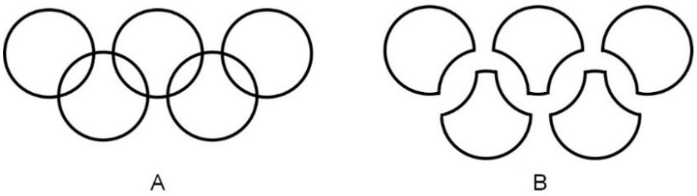


Figure 2-3. Law of Prägnanz. We tend to perceive visual scenery in the simplest form.

The very first factor Wertheimer studied was *proximity*, where he started with a group of equally spaced dots and observed that the dots do not group together into a larger entity – except that they form a “line” all together (Figure 2-4.A). He then increased the spacing between some adjacent dots so that some pairs were closer to each other than others, and noted that closer dots are grouped together into pairs (Figure 2-4.B). The effect was so strong that even if one tries to perceive the dots in a different grouping (such as ● / ●● / ●● / ●● / ● instead of ●● / ●● / ●● / ●●), it is extremely difficult if not impossible. Next, Wertheimer studied the concept of *similarity*, where he altered various properties of the elements such as color, size, orientation, shape etc. (Figure 2-4.C-F). All else being the same, elements with the same property tend to be grouped together. Again, it is rather difficult to perceive the elements in Figure 2-4.C in different groupings than based on their color similarity – similarly in Figure 2-4.D-F for size, orientation and shape. Another factor he studied in grouping was *common fate*, where elements that move in the same way tend to be grouped together (Figure 2-4.G). Proximity and common fate are sometimes considered to be special cases of similarity where the common properties of the grouped elements are respectively their relative positions and

[1] The term “Gestalt” has actually been introduced to psychology in late 1800s before Wertheimer, yet the notion was somewhat different. Unlike Wertheimer, the whole was described to be constructed from its parts, where Gestalt Psychology sees the whole as a separate entity of its own.

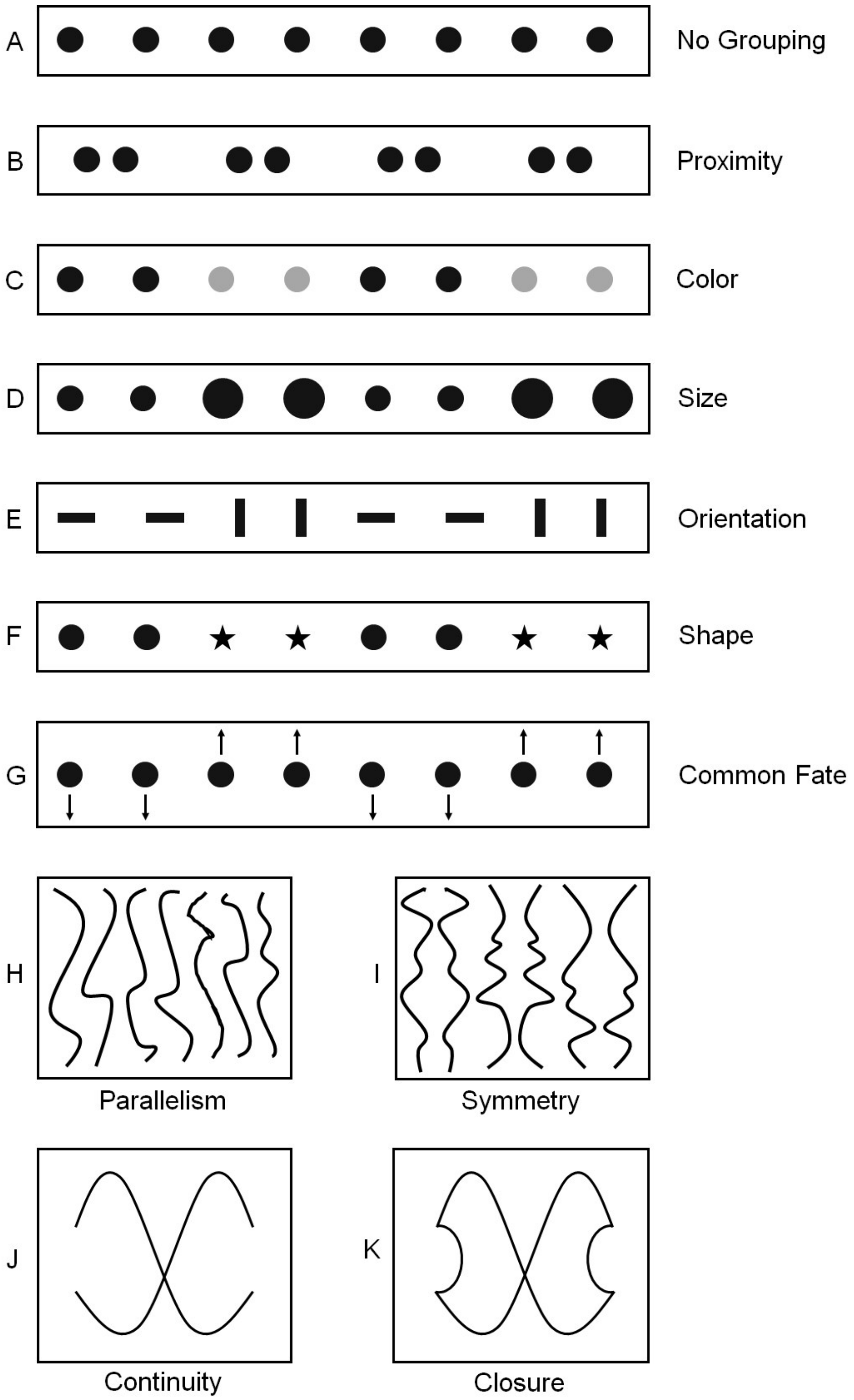


Figure 2-4. Classical principles of grouping (adapted from [19]).

velocities. *Symmetry, parallelism*, *continuity* and *closure* are further factors affecting perceptual grouping (Figure 2-4.H-K). Continuity and closure are particularly interesting in Figure 2-4 since Figure 2-4.J is perceived as two intersecting curves (rather than four line segments whose ends are touching at one single point), whereas the same curves are perceived as two closed shapes when their ends are connected to form a closed shape. We never group the line segments in Figure 2-4.H such that it is perceived as two parts divided to left and right from the intersection point. However, those segments are immediately grouped together in Figure 2-4.K to form two separate closed regions touching at one point. Here, we observe how one grouping overcomes another. Such predominance may occur in any law of grouping discussed above. One good example is camouflage, where an object is not perceivable since it is grouped together with its background due to similarity of color, texture etc. However, when the object starts moving, common fate overcomes other groupings and the object starts to be perceivable. This is why Gestaltists call these laws "*ceteris paribus rules*" (translates from Latin as "with all other things being equal"), which means that the final perception can only be predicted when no other grouping factor is affecting it.

2.2. Figure/Ground Organization

The idea of figure/ground organization relates to one of the most fundamental ways we simplify a visual scene by means of partitioning it into an object (figure) and background (ground). The Rubin's vase in Figure 2-5 is one of the most frequent demonstrations of the phenomenon. Whether you see a vase or two faces looking at each other depends on whether you see the black or white as the background. Most people can also switch back and forth between a vase and faces.

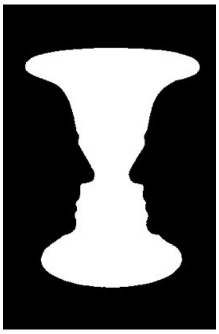


Figure 2-5. Rubin's Vase

How does our brain decide what is the object and what is the background? How does figure/ground perception occur? Traditionally, the factors that determine figure/ground organization have been tested via subjective experiments using examples composed of various black and white regions where either the black or the white region could be either figure or ground. Subjects were asked to choose which region they perceived as the figure. These experiments support less formal early demonstrations of the importance of certain configural cues for initial figure/ground segregation, without relying on past experience (familiarity). These features – as in laws of grouping – are also considered to be ceteris paribus rules and their affects are analyzed independently. However, as we will see, certain features override others easily in certain cases.

The classic configural cues proposed by Edgar Rubin [18] and the Gestalt psychologists are associated with the figures rather than grounds. One basic feature that affects figure/ground organization is the *size* of the region. Typically, the smaller the region is, the more likely it will be perceived as figure. Consider Figure 2-6.A and B where we instantly perceive the vertical lines as the foreground regardless of the polarity. *Symmetry* is another property that influences our perception of figure/ground where symmetrical regions tend to be perceived as figure (see Figure 2-6.C). However, *convexity* dominates symmetry in Figure 2-6.D and we are more likely to perceive white regions as the figure even though black regions are symmetrical and white regions are not. Finally, if a region is completely surrounded by another region, the surrounded region is perceived to be the figure and the surrounding one is the ground. These aforementioned properties are known to be the *classical configural cues*.

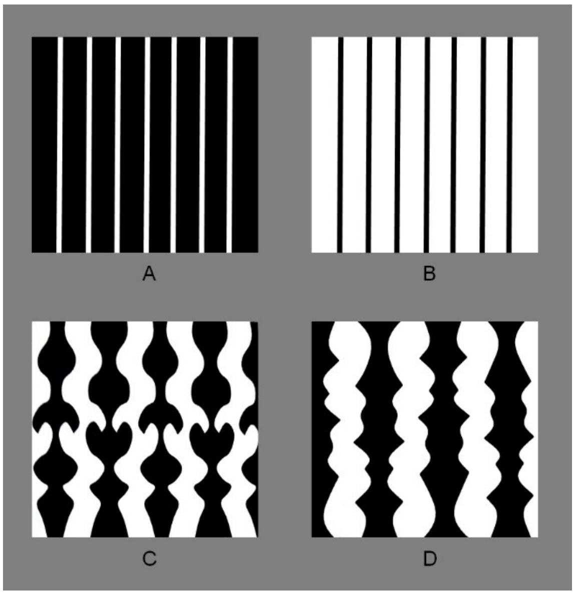


Figure 2-6. Thin lines in A and B are almost always perceived as figure, since we tend to perceive smaller regions as figure and larger regions as ground [21]. Whereas black regions in C are being perceived as figure due to symmetry, convexity takes over in D and white regions are perceived as figure even though black regions are symmetrical [19].

Following the introduction of classical configural cues in early 20th century, even today scientists continue to propose new image based features that effect our perception of figure/ground. Saliency [22], extremal edges [23], lower region [24], top-bottom polarity [25], edge-region grouping [26] are examples of such features (see [27] for further details).

More recent studies focus on the effect of past experience on figure/ground (often referred to as foreground/background especially in computer vision literature) organization and study how it is affected if a previously known figure is presented. Does it contradict the classical (or in general "image based") configural cues? Historically, past experience was held solely responsible for figure/ground perception by Structuralists (see Section 2.3) and Gestaltists opposed that idea by proposing the aforementioned classical configural cues and claimed that the visual input is organized into figures and grounds based on factors readily apparent in the image before memories of past experiences are accessed. However, note that evidence indicating that the Gestalt configural properties are relevant to figure assignment does not entail that past experience is not relevant. Today, evidence show that past experience plays a role in in such organization [28], [29]. In order to demonstrate the effect of prior experience, Peterson et al. [28] presented the experiment in Figure 2-7 where the white region in A resembles the figure of a standing woman. Subjects were more likely to choose the white region in Figure 2-7.A as the figure, but not in Figure 2-7.B or Figure 2-7.C. They deduced that the familiarity of the object in A (i.e. past experience) was the reason for such result, since image based properties should not be affected by reversal or scrambling.

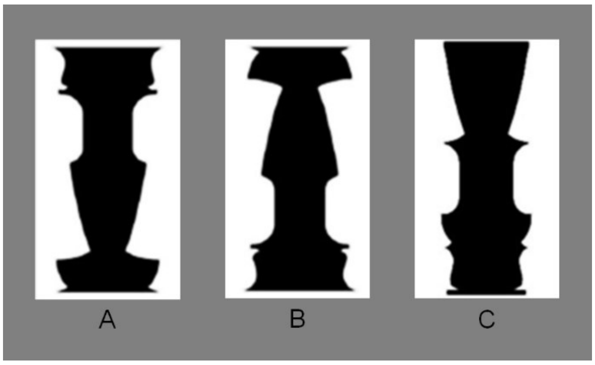


Figure 2-7. Figure of a standing woman is given in two portions (left and right) in A. The same figure in A is given upside-down in B. The same figure in A is given with scrambled parts in C.

After a century of the initial ideas of Gestalt Psychology, we now know that other image based features than the initially proposed classical cues do effect our final figure/ground perception. Moreover, it is also proven that initially discarded cues such as past experience, attention and intention also play a significant role. As a final point, what recent research concludes is that figure/ground perception results from a “winner takes all” competition, where all of the aforementioned cues compete to dominate the other cues [30]-[31].

2.3. Other Vision Theories

Wilhelm Wundt (1832–1920), who is often referred as the father of experimental psychology, was the first scientist to investigate human mind scientifically in a controlled environment. His aim was to analyze thoughts and sensations into their fundamental elements. His ideas and way of thinking were expanded by one of his students, Edward Titchener (1867–1927), who later formally established the first psychological approach to perceptual theory – known as *structuralism* [12]. Structuralists believed that our perceptions could be broken down into individual sensations, in a theoretical analogy to chemistry where primitive atoms come together to form more complex molecules. In this perspective, Gestalt Psychology arose as a reaction to structuralism, rejecting almost everything they put forward.

William James (1842–1910) opposed structuralism stating that the mind is fluid, not stable. Therefore, instead of trying to understand its structure, understanding its function would be more beneficial. He later named this viewpoint *functionalism*. Another psychological school opposing structuralism was *behaviorism*, which was also at the target of Gestaltists. Behaviorists claim that emphasis should be on observable behavior and not on mental events or subjective.

Another classical theory of visual perception is called *Ecological Optics*, which essentially came from the works of James Gibson (1904–1979), states that environmental perception is entirely a function of the stimulation received from the environment – i.e. it is direct. In Gibson's terms, humans receive information directly from the environment and view it as a whole meaningful entity rather than in a disaggregated way, and their perception is based on the use of these entities rather than their form, color or other attributes.

One of the latest theories on visual perception is proposed by David Marr (1945–1980), which is called *computational theory*. He claims that visual perception is nothing but an information processing task, and characterizes it in three levels in a similar way that a machine would carry out the task: computational theory (what is the goal and logic?), representation and algorithm (how can it be implemented?), and hardware implementation (how can it be realized physically?). Interested reader is referred to [32] for a more comprehensive discussion of theories on vision.

One important point to notice is that, throughout the history, theories of visual perception did not follow a straight line of development that builds on top of each other. Instead, the theories evolved from various approaches influencing one another. For instance, despite losing attention for several decades after their founding fathers died, the influence of Gestalt Psychology on both ecological and computational approaches has been acknowledged explicitly and many of its principals – mainly the laws of grouping and figure/ground organization – have again gained attention in the forefront of the field.

Chapter 3

Color Perception and Description

When we look around our environment, we observe all kinds of objects with all kinds of colors: A red rose, a green car, a brown cup… This leads us to believe that color is the physical property of these objects. However, in contrast to the mainstream belief, it is in fact a psychological property of our visual experience. It is of course true that our perceptions stem from the physical properties of those objects, but our final perceptions are more often than not different than their physical properties. What we actually perceive is the result of complex interactions between those physical properties, our eyes, nervous system and brain. Despite being one of the most disclosed topics in vision science, there is still considerable amount of undiscovered territory in color perception. For instance, even though the sensory processing of color is well understood, less is known about how it is processed in the brain.

The notion of perception in color vision was first proposed by the famous writer Johann Wolfgang von Goethe (1749–1832), who recognized that colors which are physically the same may appear different to humans or conversely, different colors may be perceived physically the same by the human brain under certain circumstances. By stating that our perceptions can be different than the sensory inputs we obtain from our environment, Goethe actually paved the way for Gestalt Psychology that we discussed in Chapter 2. In fact, he was the first to introduce the word and concept of Gestalt to science [33]. It is rather easy to demonstrate what he intended to say. For example, when we look at a patch of green grass under a blue sky and then later at sunset, the color of the grass seems unchanged. However, the reflected light reaching the eye has a very different spectrum in the two situations – a phenomenon today called *color constancy* (see Section 3.1.2). Similarly, if we reconsider Figure 1-2, two color patches which are physically the same are perceived different in different settings. Goethe was undoubtedly right, but what is it really that makes us perceive different colors to be similar and similar colors to be different? Is it simply the context or background?

Does memory (i.e. familiarity) have any role in it? Do other visual properties such as shape, texture or geometry has any effect? Even though there are various descriptions, demonstrations or theories undertaking such questions, there is still no scientifically proven and accepted model on color perception. For instance, it has been typically believed that perception of color is independent of the perception of other visual features [34]-[36], however further research showed that there are in fact neural connections between color and 2D shape processing [37], [38]. It has also been shown that color perception is strongly influenced by 3D shape perception [39]. However, even though such findings shed light on the issue, the puzzle is yet to be solved. Furthermore, what we know for certain is that our perception of color is not merely bound to physical measurements.

3.1. Color Vision

Isaac Newton (1642–1726) was the first one to propose that light is responsible for color, by obtaining different colors of light via refracting it through a prism and refracting them back together to form a white light. The color of these refracted lights are determined by their wavelength. The human eye is able to distinguish different colors between 400-700nm of wavelengths – which is a minuscule fraction of the entire electromagnetic spectrum. We can detect the light within these wavelengths via the photoreceptor cells located in the retina of our eyes that are called *rods* and *cones*. While there is only one type of rod cell in our eyes, there are three types of cone cells and our color perception is based on the relative responses of these three types of cones. Cones are sensitive at high luminance levels whereas rods serve our vision at low luminance levels (ergo, we do not perceive colors at very low level of illumination). At high luminance levels the rods are effectively saturated and only the cones function. The three types of cones are most properly named as S, M and L cones, which refer to short-wavelength, middle-wavelength and long-wavelength sensitive cones respectively. They are often inaccurately named as R, G, and B cones referring to red, green and blue. However, these cells are not specifically sensitive to one single color (i.e. wavelength), but rather a range of wavelengths broadly overlapping with each other. Figure 3-1 shows the spectral response of the rods and the three types of cones found in the human eye. Note here that even though cones are responsive to some ultraviolet light (wavelengths shorter than 400nm), we cannot see them since they are blocked by the lens of the eye before reaching the cones [40].

It would be rather straight-forward and simple to assume that each color we perceive is formed by a mere combination of photoreceptor responses. Unfortunately, the process is much more complicated. These signals that are formed in the retina are then processed through the network of retinal neurons and sent to the brain via the optical nerve. The neural processing of visual information is already quite complex within the retina, but it becomes significantly more complex at later stages. Interested reader is referred to [12] for an overview of the paths that some of this visual information follows. However, there are certain theories and phenomena that help us comprehend the processing of color signals in the human visual system.

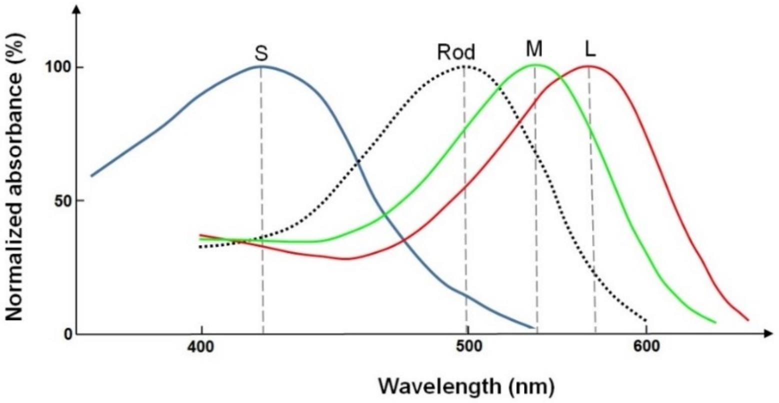


Figure 3-1. Normalized absorbance of short (S), medium (M) and long (L) cone cells and Rod cells (R) in the eye with respect to different wavelengths of light (adapted from [42]).

3.1.1. Theories on Color Vision

There are two major theories that explain and guide research on color vision: the *trichromatic theory*, and the *opponent process theory*. Today, both theories are accepted as complementary to each other, and explain processes that operate at different levels of the visual system.

Trichromatic theory was proposed by Thomas Young in 1802, which was later extended by James Maxwell and Herman von Helmholtz. The theory proposed that the human visual system performs additive color mixing via three primaries. In other words, there are three photoreceptors in the human eye that have different peak sensitivities (namely at red, green and blue) and our perception of different colors occur through activation of these photoreceptors in different relative ratios. Even though the physical validation of these photoreceptors was made more than 100 years after the initial proposal of the theory [43], trichromatic theory was able to dominate the field of color vision for over a century because it was able to unravel many facts via relatively simple expositions. However, there are certain aspects that the theory still cannot account for. For example, there is no explanation for red/green color blindness, which would require the absence of red and green primary photoreceptors based on the theory. Yet it fails to explain the ability of the same person to perceive “yellow” which would be the combination of red and green based on the theory. Besides, the theory also fails to explain certain phenomena such as “after images” (see Section 3.1.2).

Edward Hering (1834–1918) noticed the aforementioned shortcomings of the trichromatic theory, and proposed the opponent process theory. He realized that certain color pairs never perceived together – namely red/green or yellow/blue. Therefore, he proposed four primary colors red, green, yellow and blue, working in pairs red-green and yellow-blue and a third, black-white (or light-dark) to account for our perception of brightness. All three of these mechanisms work in opposing pairs, i.e. the perception of red opposed to perception of green, perception of yellow opposed to perception of blue and perception white opposed to perception black. In fact, the main stance of the theory lies in such observations that we never observe a reddish green or yellowish blue color.

The two theories, trichromatic theory and opponent process theory, competed for decades and there were endless debates between two factions. In the middle of the 20th century, together with support from overwhelming quantitative and physiological data and research [44]-[46], *modern opponent theory* (also called as *stage theory*) was shaped. The theory, unlike earlier disputes, proposes that Young, Maxwell, Helmholtz and Hering indeed were all correct, but their theories refer to different stages in visual perception. Whereas trichromatic theory models how the initial signals are received by cone receptors in the eye, those signals are not directly sent to brain as the theory claimed. Instead the colors are encoded into opponent channels as the opponent process theory suggests. Figure 3-2 shows the first stage of color vision suggested by the modern opponent theory.

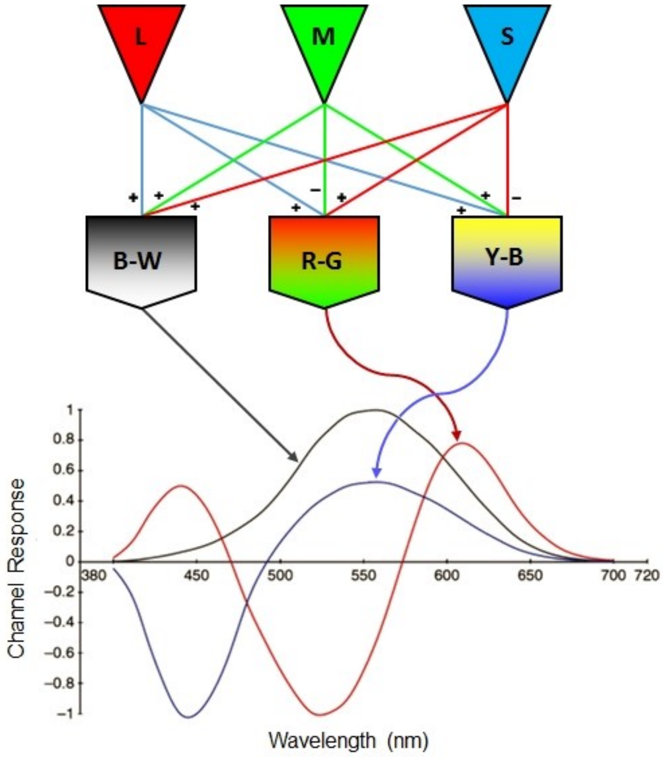


Figure 3-2. Encoding of signals received in cones into opponent channels and the response of opponent channels with respect to wavelength (adapted from [12]).

3.1.2. Basic Phenomena Affecting Color Perception

The aforementioned theories explain early stages of our perception of color and shed light to various phenomena, yet they still fail to explain the subjectivity that Goethe mentioned: "*physically same colors may appear different to humans or conversely, different colors may be perceived physically same under different circumstances*". Here, the key part of Goethe's statement is "*under different circumstances*", because those circumstances in fact have significant impact on our perception.

The adaptation mechanisms of our visual system, namely *brightness and color adaptation*, determine how our perception behaves under changing illumination conditions. For example, when we walk into a dark room from a bright room our eyes adapt to the new conditions by changing the sensitivity of the photoreceptors. At first, the cones gradually become more sensitive. Then, until about 10 minutes have passed, visual sensitivity is roughly constant. At that point, the rod system, with a longer recovery time, has recovered enough sensitivity to outperform the cones and thus the rods begin controlling the overall sensitivity. The rod sensitivity continues to improve until it becomes asymptotic after about 30 minutes. The same physiological mechanisms serve when we move from a dark room into a bright room, but there is an asymmetry in the forward and reverse kinetics resulting in the time course of light adaptation being on the order of 5 minutes rather than 30 minutes as in dark adaptation [12]. One interesting phenomenon during brightness adaptation is the so called *Purkinje effect*, which causes a difference in color contrast as the illumination level changes from bright to dark and shades of blue look relatively lighter than shades of red. This is due to the fact that as cone vision gradually switches over to rod vision, the peak of visual sensitivity shifts towards shorter wavelengths [47]. Another event that causes an apparent change in color as the illumination level changes is called *Bezold–Brücke effect*. Specifically with increasing intensity, longer wavelengths appear more yellow and shorter wavelengths appear bluer. When intensity is decreased, shorter and longer wavelengths become redder in appearance while middle-wavelengths appear greener [48].

Whereas the sensitivity of all photoreceptors adapt in case of brightness adaptation, our eyes can also adjust the sensitivity of each type of cones (S, M or L) independently after being exposed to a certain color of light – this is called chromatic adaptation. In other words, if we are exposed to a specific color for a prolonged time, our visual system's sensitivity to that color decreases. In order to demonstrate, if you cut a yellow Ping-Pong ball into two and place each half over your eyes, after a few minutes you will start seeing a colorless gray fog rather than a yellow surface. This is because the sensitivity of your eyes' to yellow has decreased. Then if you remove the balls after complete adaptation, the world will look slightly bluish – complementary of yellow. This is called an aftereffect (or afterimage). After images can be defined as the after effects of viewing highly saturated colors for a prolonged period of time. These afterimages were among the important bases for the opponent process theory Edward Hering proposed. Similar afterimages can also be observed for brightness adaptation.

3.1.3. Spatial Color Vision

The different types of adaptation discussed so far are observed after a prolonged exposure to a certain stimulus over time. However, similar effects can also be observed when such stimulus is placed in a certain organization over space. For instance, consider Figure 1-1 where the patches are perceived with different lightness depending on their background. In fact, all patches on the same row have exactly the same lightness; however, the ones placed on darker background appear to be brighter than the ones placed on light background. This phenomenon is called *simultaneous contrast*. Simultaneous contrast can also be observed in colors (in which case it is referred as *simultaneous color contrast*). In Figure 1-2 the small patches appear in different colors because of their surroundings. The upper patch appears brownish while the lower patch looks yellow. However, when isolated from their background, it is easy to observe that they are exactly the same color. Notice that in both cases of simultaneous contrast, the region of interest is shifted towards the complementary color of the surrounding background. In Figure 1-1 the dark background forces us to perceive the patch lighter, and light background forces us to see it darker. Similarly, in Figure 1-2, the lower patch is surrounded with blue patches which shift its color towards yellow – complementary of blue.

An interesting Gestalt observation was made by Kurt Koffka on simultaneous contrast. He presented the Koffka ring in Figure 3-3 and pointed out that the effect of simultaneous contrast disappears (or suppressed) when the objects of interest in different backgrounds are brought together. What happens is that in Figure 3-3.A the ring is observed as a whole object, whereas in Figure 3-3.B each half has its own identity.

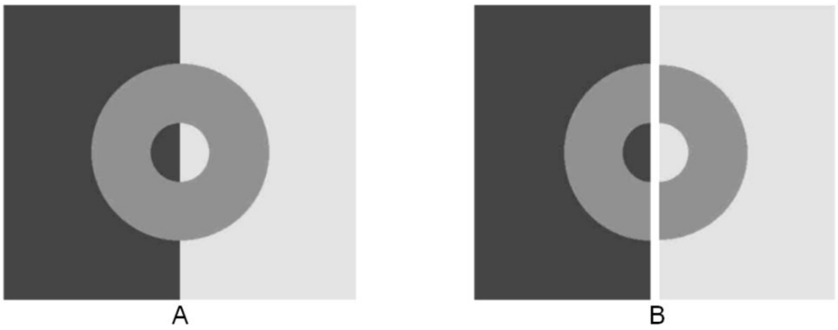


Figure 3-3. The Koffka ring. (A) The ring is perceived to be uniform, (B) Each half of the ring is perceived to have a different lightness due to simultaneous contrast.

3.1.4. Color Similarity

During his proposal of the trichromatic theory, Helmholtz performed color matching experiments to investigate the additive property of the human color matching. The subject is given a region illuminated by a certain color and asked to match another region's color via adjusting three primary light sources. On the other hand, Hering proposed six colors working in opponent pairs for human color perception; which are red, green, blue, yellow, white and black. These colors are also referred as Hering primaries. Moreover, psychological and linguistic studies have also revealed that humans have a small number of basic color terms for specifying colors, but whether these colors are universal or not is still an ongoing debate. Whereas one line of though follows the findings of Berlin and Kay that perception of primaries is universal [49], others claim that color also has cultural and linguistic aspects, hence each culture (language) has its own primaries [50]. For example, they categorized colors for English in eleven primaries which in addition to Hering primaries, include orange, purple, pink, brown and gray. Naming of colors is critical also when we judge the similarity between them. If we formulate the problem as a clustering problem (i.e. dividing the entire color space into a certain number of regions where the aforementioned primaries are the centers of those regions), then where we draw the border between different colors becomes of significant importance in terms of color similarity. For instance, Broek et al. used the above eleven primaries for introducing a new color matching method and referred to these colors as *focal colors* [51]. At this point it is also important to notice the fact that the human eye cannot perceive a large number of colors at the same time, nor able to distinguish their similarity (or dissimilarity) [52]. This fact also agrees with the concept of focal colors, leading to the conclusion that a small number of colors are sufficient to represent a multicolored pattern; which are commonly called *dominant colors*. For instance, Figure 3-4.A shows an image where in total there are 71893 different colors. Figure 3-4.B on the other hand represents the same image using only 6 colors. The two images are immediately recognized as similar – if not the same – when they are compared in terms of color similarity. Dominant color extraction is also a fitting example of Gestalt grouping since what happens in Figure 3-4 is that our perception groups pixels together based on their similarity and proximity. The image in Figure 3-4.B may look different than the original image in Figure 3-4.A, however it is a better representation of what we perceive in terms of colors. If we were to describe the color composition of the original image, it would be something like: “a white and brown horse on a green field and yellow flowers”, which is exactly what Figure 3-4.B shows. In other words, what we perceive is different than what individual pixels have – the whole is different than the sum of its parts.

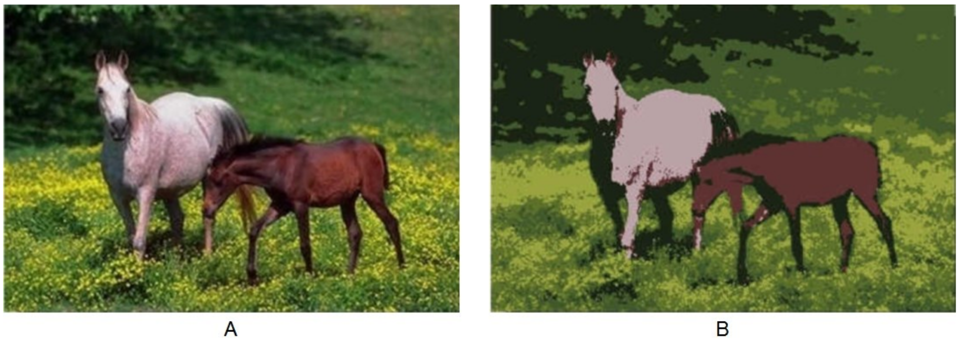


Figure 3-4. Illustration of dominant colors. (A) Original image with 71893 unique colors. (B) Same image represented with 6 colors.

3.2. Perceptual Color Description

Color is one of the most frequently used features in image processing due to its robustness to noise, image degradations, changes in size, resolution and orientation. In fact, color composition of an image turns out to be a powerful feature when judging image similarity, particularly if such composition is represented in a perceptually oriented way. It should however be noted that color properties correlate with the true image content only to a certain extent, and cannot be used as a single cue to characterize the entire content [53]. In order to use color as an analytic measure to judge similarity, mainly two steps are essential: *representation of individual colors* and *description of the color composition*.

3.2.1. Color Space

Colors are typically represented by means of different color spaces. The vision theories discussed in 3.1.1 both suggest separate ways of representing a color. For instance, trichromatic theory proposes that a color can be represented as a mixture of three primary colors. Typical example of this idea is the RGB color model[1] where each color is represented as a weighted combination of the primary colors red, green and blue. On the other hand, the opponent theory suggests that colors should be represented with three components black/white, red/green and yellow/blue. The Lab color space, for instance, follows the opponent theory and represents each color with L: black/white, a: red/green and b: yellow/blue. HSV color space, on the other hand, is designed to be more intuitive and easier to interpret. H (Hue) refers to the pure color without any shade or tint (or "dominant wavelength" in more physical terms). S (Saturation) refers to the purity of the color and V (Value) stands for brightness. Figure 3-5 depicts these color spaces in 3D space.

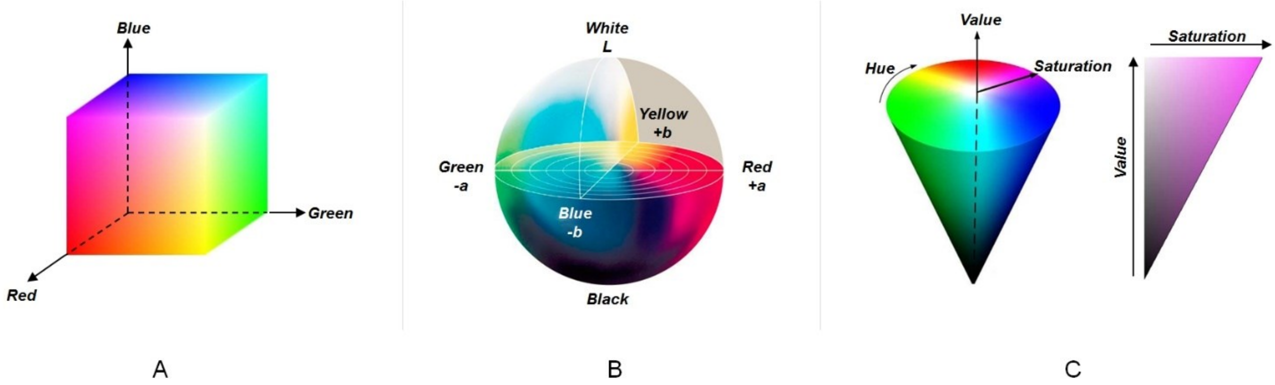


Figure 3-5. Different color spaces. (A) RGB, (B) Lab (adapted from [54]), (C) HSV.

Each color space is typically designed to serve a specific purpose. For example, the sRGB color space (i.e. standard RGB) is widely used and standardized in color reproduction such that almost all LCDs, digital cameras, printers, and scanners follow the sRGB standard. HSV color space is more preferred by designers where color modifications and understanding is necessary such as computer graphics or image editing software. This is due to its intuitiveness that we would know how changing each channel would affect the resulting color. Due to its ease of interpretation for humans, HSV is also widely used in computer vision and image processing applications, such as robotics, face detection/recognition, object detection/recognition, content based image retrieval etc. However, our ability in comprehending the dynamics of the HSV color space does not necessarily transfer to those algorithms. For a computer algorithm, a color in a color space is simply a set of numbers. While concepts such as hue, saturation and value make perfect sense to us and help us understand how colors are affected by changing these channels, for a computer these concepts do not mean much unless the algorithm is specifically designed to deal with them. Lab color space on the other hand is designed to be perceptually uniform, which means that two colors that are equally distant in the color space are also equally distant perceptually. In this sense it is much more suitable for computer vision algorithms than any other color space – particularly if the algorithm involves color similarity judgment – since our notion of color similarity is inherently transferred to the algorithms.

[1] The color model only defines the mixing of the colors relative to the primary colors red, green and blue. It becomes a color space only when the exact meaning of those primaries are specified colorimetrically. There are various color spaces that follow the RGB color model such as sRGB, Adobe RGB etc.

3.2.2. Color Descriptor

Color content and composition are among the key elements that have been utilized in image processing applications. They have either been used as direct attributes to represent an image or image region, or combined with other features with the purpose of increasing the description power. In order to capture such content and composition accurately, a descriptor needs to be designed carefully bearing in mind the target application. For instance, there are many color descriptors that only contain the information of the colors that are present in the image. The composition or structure may or may not be significant for the intended use case. Such descriptors are typically referred as *global color descriptors*, meaning that no spatial information is relayed by the descriptor. The well-known color histograms [55] are classic examples of this kind, where the entire color space is quantized into a predefined number of bins and each pixel in the image is assigned to the appropriate bin. Due to their simplicity and reasonable performance, there are wealth of methods incorporating color histograms – or histogram based approaches [56]-[58]. The dominant colors discussed in Section 3.1.4 also fall in the category of global descriptors, where they cluster the image pixels (i.e. colors) into typically a few clusters – which are called the dominant colors. If they are extracted properly according to the aforementioned color perception rules, they can indeed represent the prominent colors in any image while discarding the unperceivable elements in the image. Moreover, since they only include the few (dominant) colors that are perceivable in the image, they are significantly more compact than histogram based descriptors where the entire range of colors is quantized and included in the descriptor.

Global color descriptors provide information of which colors are present in the image and how much. However, they provide no information on the structure or distribution of those colors within the image – which can be of significant importance. In other words, in addition to describing ‘‘what” and ‘‘how much” color is present in an image, specifying ‘‘where” and ‘‘how” they are distributed may prove to be of significant importance – particularly for the purpose of image similarity. Ideally, from a perceptual standpoint, it makes more sense to speak about objects’ colors and their spatial distribution. However automatic image segmentation is an ill-posed problem and is not reliable and robust enough to serve as a basis for object based image analysis (see Chapter 4).

3.2.3. Proximity Histograms and Grids

In [P1] we have discussed the drawbacks of commonly used global and spatial color descriptors and proposed two new descriptors based on human color perception discussed in this chapter and Gestalt Psychology discussed in Chapter 2 – namely proximity histograms and proximity grids. These descriptors are designed to address the drawbacks and problems of the conventional color descriptors. In order to achieve this, they are mainly motivated by the human color perception rules and therefore, global and spatial color properties are extracted and described in a way our visual system perceives them. The descriptors carry both global and spatial information regarding colors in the image. Whereas the global component is common for both, spatial component can be extracted as either proximity histograms or proximity grids. Based on the earlier discussion, global information of the colors in the image is acquired entirely based on dominant colors in order to get the most perceptual, reliable and compact information. Dominant colors are extracted in a similar fashion in [59], where the colors in the image are clustered until a maximum number of clusters with a maximum allowed distortion is reached. The clusters are also allowed to be similar up to a certain level. These limits are tunable in order to allow the algorithm to serve different purposes. The extracted statistics, i.e. the color value, normalized area and standard deviation of each color, are part of the final color descriptor. These colors are then back-projected onto the image in order to further analyze their spatial distribution.

As we discussed in Section 3.1.4, the image with dominant colors back-projected is a proper illustration of how we perceive the original image based on the gestalt rules of grouping. However, the clustering and back-projection operations may naturally lead to some isolated clusters (see Figure 3-6). These clusters are outliers for our perception – again due to the rules of gestalt grouping. Even though these small clusters are visually similar to other larger clusters in terms of color, their size and proximity disqualify them from being perceptually grouped with larger similar clusters. Therefore, these spatial outliers need to be removed in order to have a more (perceptually) accurate spatial representation of the image colors. In order to achieve this, a quad-tree decomposition is used which starts from the whole image and incrementally goes to its parts. Quad-tree keeps partitioning the image until either a certain uniformity is reached in a block, or the maximum level of depth is reached. These two limits control the level of "resolution" in the final image. Finally, the colors of the host quad-tree blocks are back-projected onto the image in order to remove the aforementioned spatial outliers (see Figure 3-6).

The final image, where outliers in both color and spatial domains are removed and the dominant colors are assigned to their blocks, can be conveniently used for further spatial analysis. Note that quad-tree blocks can vary in size depending on the depth, yet even the smallest block is large enough to be perceivable and carry a homogenous dominant color. So instead of performing a pixel level analysis, the uniform grid of blocks in the highest depth of the quad-tree can be used for analyzing the spatial characteristics accurately. Two alternative descriptors are proposed in [P1] where both of them reveals the distance of each color in the image relative to another color. Whereas *proximity histograms* describe such inter-relation in a scalar measure, *proximity grids* also include the direction information. In other words, while proximity histograms can state "17% of red is 8 units (blocks) away from blue", proximity grids can say "17% of red is 8 units (blocks) right of blue". Note that such directional information may or may not be important based on the application in which the descriptor is exploited. For instance, in an image/video recognition system where we want to detect the sky, we may need to know where in the image the "blue" region is located. Or in a system where we want to detect national flags, relative location of color regions is crucial.

A proximity histogram for a color pair c_i and c_j stores in its k^{th} bin the number of blocks hosting c_j at a distance k from all blocks hosting c_i. Such a histogram clearly indicates how close or far two colors are and their spatial distribution with respect to each other. Note that the size of the histogram indicates the maximum distance k that is checked between two colors. Whereas it is possible to perform a full range search within the entire image, that is in general redundant since their spatial proximity will seize to produce a gestalt after a certain distance and whether they are N blocks away or $N+5$ blocks away will not affect our perception. Similarly, a proximity grid for a color pair c_i and c_j stores in its $(k,l)^{th}$ bin the number of blocks hosting c_j at the (k,l) coordinate relative to all blocks hosting c_i. As a result, such a grid characterizes both inter-color proximities and the relative spatial position between the two colors. The gist of the description for both proximity histogram and grid can be seen in Figure 3-7, where proximity grid distinguishes the relative direction of a color pair, but proximity histogram cannot due to its scalar metric. Together with the colors' global properties (i.e. the color value, normalized area and standard deviation) the final descriptor holds all necessary information in order to portray the color composition of an image.

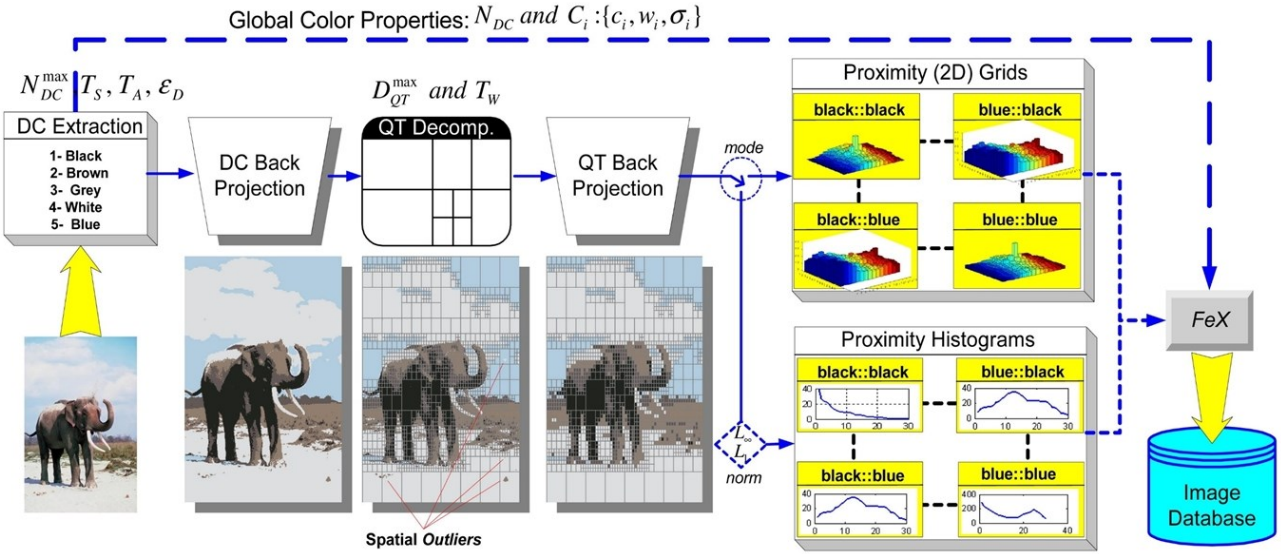


Figure 3-6. Overview of Proximity Histograms and Proximity Grids.

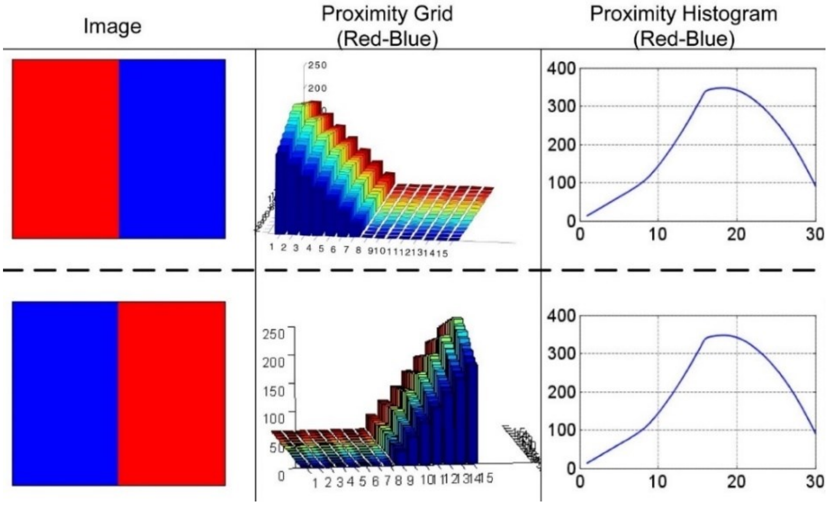


Figure 3-7. Proximity histogram vs. grid for a simple image (above) and its horizontally flipped version (below).

In order to compare two images based on their color similarity, a penalty-trio model is proposed in [P1] where both the difference of the similar colors and the amount of different colors in each image are taken into account. Since the colors have both global and spatial properties, the model penalizes both global and spatial differences. Equation 3-1 shows the calculation of final distance from three penalties where P_φ is the penalty for different colors in the images, P_G is the penalty for the global properties in similar colors, P_S is the penalty for the spatial properties in similar colors, and α is the weighting between global and spatial properties. While P_φ results from the difference in the area (i.e. coverage in the image) of the non-matching colors in two images, P_G considers both area and color distances (i.e. distance in the color space) of the matching colors. P_S is simply the difference of the spatial descriptors (i.e. proximity grid or histogram).

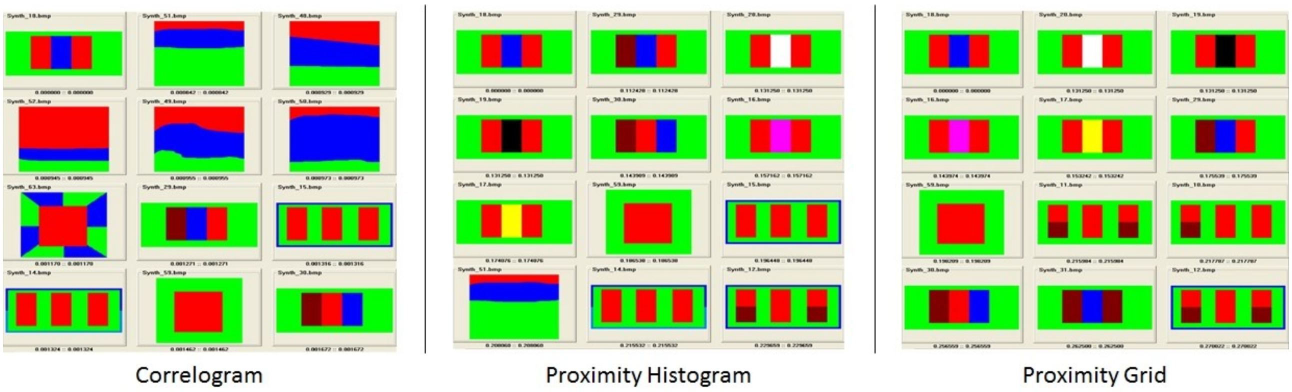
$$P_\Sigma(Q,I) = P_\varphi(Q,I) + (\alpha P_G(Q,I) + (1-\alpha)P_S(Q,I)) \tag{3-1}$$

Note that the above penalty model performs a color matching operation on color sets from two images. However, matching two sets of colors is not that straightforward. One color in one set can easily be similar to many colors in the other set, making descriptor comparison rather complicated. Whereas enforcing a one-to-one matching would be an easy solution, it could introduce significant errors particularly due to the dynamic clustering in dominant color extraction. Therefore, initially a one-to-many matching is allowed, then a color fusion is performed on those “many” colors that the “one” color matched and their global and spatial descriptors are combined accordingly. Since any color in any image can match to many colors in the other image, color matching is performed twice – first $\mathrm{I} \rightarrow \mathrm{Q}$, then $\mathrm{Q} \rightarrow \mathrm{I}$. Once the matching color pairs are established, calculation of each penalty term is straightforward.

[P1] evaluates the performance of the proposed descriptors on a content based image retrieval system [60], and calculates the Average Normalized Modified Retrieval Rank (ANMRR) [61] to assess retrieval performance. Basically three databases of different sizes are used – namely with 1000, 10000 and 20000 images. However, since the ground-truth in these databases are extracted based on the actual semantic content, they do not necessarily reflect the color content similarity. That’s why a third database consisting of 1089 synthetic images with various color compositions is used in order to demonstrate the true description power of the proposed descriptors since in that database, color alone characterizes the entire content. Figure 3-8 shows a sample query and its retrieval results based on three different algorithms on the synthetic database. The power of the penalty-trio model can easily be seen where Color Correlogram [62] only reveals spatial characteristics in pixel level and the proposed descriptors reflect both the global spatial dissimilarities of the query image. Table 3-1 shows the ANMRR scores of the proposed descriptors, Color Correlogram and the MPEG-7 Dominant Color descriptor. Note that neither the global nor spatial features are distinctive enough on their own in order to represent color content, and both of the proposed descriptors outperform their competitors.

Table 3-1. ANMRR scores of the proposed and the competing descriptors for three Corel databases.

Descriptors	**Corel 1K**	**Corel 10K**	**Corel 20K**
MPEG-7 Dominant Color	0.180	0.458	0.461
Auto-Correlogram	0.222	0.381	0.444
Correlogram	0.195	0.357	*NA*
Proximity Histogram	**0.154**	**0.263**	**0.357**
Proximity Grid	0.162	0.291	0.390



Correlogram

Proximity Histogram

Proximity Grid

Figure 3-8. Result of a sample query for the proposed descriptors and Color Correlogram in the synthetic database.

Since the publication of [P1], there has been various attempts for proposing a successful color descriptor particularly for the purpose of content based image retrieval [63]-[67]. Whereas most of these works acknowledge the perceptual approach taken by [P1], few have included the algorithm in their comparative experiments. Color correlogram and color histograms are still employed as the most popular competitors, arguably due to their ease of implementation. Therefore, an admissible direct comparison with [P1] still ceases to exist.

Chapter 4

Interactive Image Segmentation

Image segmentation aims to partition an image into smaller regions in order to form a simpler and more meaningful representation. It is a fact that humans cannot see or distinguish between different pixels, hence a pixel-wise representation neither reflects how we see an image nor forms a proper basis for further analysis. Therefore, image segmentation assigns every pixel in the image to a segment such that the resulting segments cover the entire image. Whereas these segments would ideally correspond to objects in the real scene, they may also represent only part of the objects. In either case, a more convenient and semantic representation of the image data is reached compared to the pixel representation. Such representation not only enables further applications, but also improves most of the image analysis algorithms providing a more solid and meaningful basis. For example, content based image retrieval problem can be performed on object level instead of the entire image content. One can search for a "football", and the system would bring images with football in it. This would be impossible without segmentation since the football object may cover only a small portion of the image, therefore a global descriptor of the image would barely include a hint of the football. However, if the descriptors are extracted for every object in the image, then an object based search and retrieval will be possible. Another example can be the object detection and recognition problem discussed in Chapter 5. The local patches that are assumed to be part of the object are extracted from the entire image and often include parts of both object and background depending on their size and location. However, if those patches are extracted from the segmented image, object and background will be in different segments and such errors can easily be sidestepped.

There are typically two main approaches to image segmentation. One is to group adjacent portions of the image based on their similarity, and the other is to detect local differences in adjacent regions and separate them. The methods using the former approach are commonly referred as *region based* methods, while those in the latter category are called *edge based* methods. In fact, defining regions and edges cannot be separated from each other. In region based approach, edges emerge naturally as a byproduct between the formed regions. On the other hand, in edge based methods regions are the byproduct of edge detection, provided that the detected edges form closed contours. However, that is not usually the case and additional complex algorithms are often required to group piecewise edges and define regions. Still, these defined regions, either by region or edge based methods, rarely capture the entire object and typically over or under segment[1] the image. Hence, in order to extract the entire object, those regions need to be grouped to form an object by further processing. This is rather expected, since most real life objects are composed of smaller regions. Consider, for example, Figure 4-1 where each person in the image contains multiple segments. If the objective is to segment people, then the smaller segments need to be grouped together. However, if the objective is to segment their clothing, then the segmentation is rather accurate.

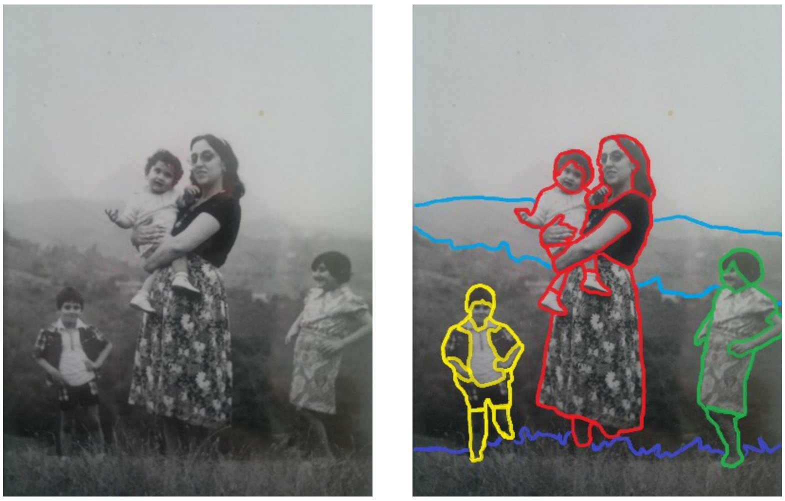


Figure 4-1. Left: Original image, Right: Segmented image.

4.1. Superpixels

When Gestalt psychologists were proposing their famous laws of perceptual grouping (see Section 2.1), they assumed that the parts that are to be grouped in order to reveal the whole are already present. However, those elements are not directly given by the stimulus, but they also require analysis just like the whole needs to be analyzed from its parts. The obvious basis in an image for such elements are the abovementioned regions obtained via over segmentation. The over segmented regions provide proper elements that are to be grouped based on perceptual criteria – such as the perceptual laws of grouping from Gestalt Psychology. Palmer and Rock [68] proposed the concept of *uniform connectedness* in order to explain how these elements might be formed. They claim that humans tend to perceive connected regions of uniform image properties – such as

[1] Over segmentation occurs when multiple segments cover one object in the image. Conversely, under segmentation happens when a segment covers more than one object.

luminance, color, texture, motion and disparity – as the initial units of perceptual organization. They also argue that uniform connectedness cannot be reduced to any of the principles of grouping, because grouping principles assume the existence of independent elements that are to be grouped, whereas uniform connectedness is defined on an unsegregated image. For this reason, uniform connectedness must logically operate before any principles of grouping can take effect [12]. Therefore, if we intend to follow the flow of human perceptual organization, these regions need to be extracted first before applying any organizational constraints. Superpixels are a fitting example that targets to extract such regions from the image that are uniform in certain image properties such as luminance, color, texture, shape[1] (see Figure 4-2).They essentially generate over segmented images, where the purpose is to capture pixel level image redundancy, provide a convenient primitive from which to compute image features, and reduce the complexity of subsequent image processing tasks [68].

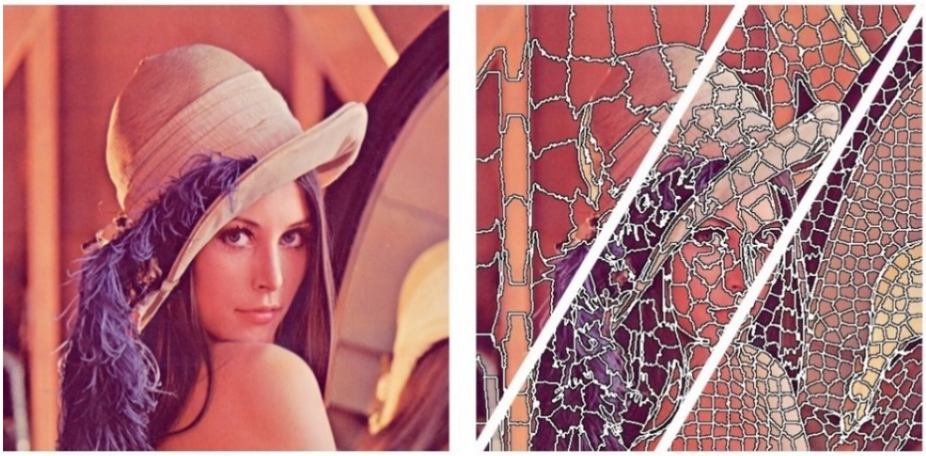


Figure 4-2. Left: Original Image, Right: Superpixels extracted in different granularities (namely 100, 500 and 1000 superpixels extracted from the original image). White lines denote superpixel boundaries.

The name “superpixel” in fact explains the underlying purpose properly: They are meant to replace pixels as the building blocks of the image, and they are capable of much more than regular pixels since they contain much more information than a single pixel. Whereas it is hard to define an ideal way of extracting superpixels that would perfectly serve any application, in [69] Achanta et al. defined three properties that any superpixel algorithm should have:

1. Superpixels should adhere well to image boundaries.
2. When used to reduce computational complexity as a preprocessing step, superpixels should be fast to compute, memory efficient, and simple to use.
3. When used for segmentation purposes, superpixels should both increase the speed and improve the quality of the results.

[1] Features such as motion or disparity are rather used in video segmentation, and the segmented regions are then referred as supervoxels.

Based on the above criteria, the authors in [69] compared state-of-the-art methods for superpixel generation, yet they were not satisfied with the outcome. For instance, some algorithms had high boundary recall rates, yet their under segmentation error was also high. Some were computationally too expensive, some had little or no control over the amount or compactness of the superpixels. Therefore, they proposed a new algorithm called SLIC (Simple Linear Iterative Clustering), which is basically an adaptation of the well-known k-means clustering algorithm. SLIC superpixels are currently one of the most popularly used superpixel generation algorithms, and there are implementations that enables generating SLIC superpixels much faster than real-time [70].

4.2. Region Merging

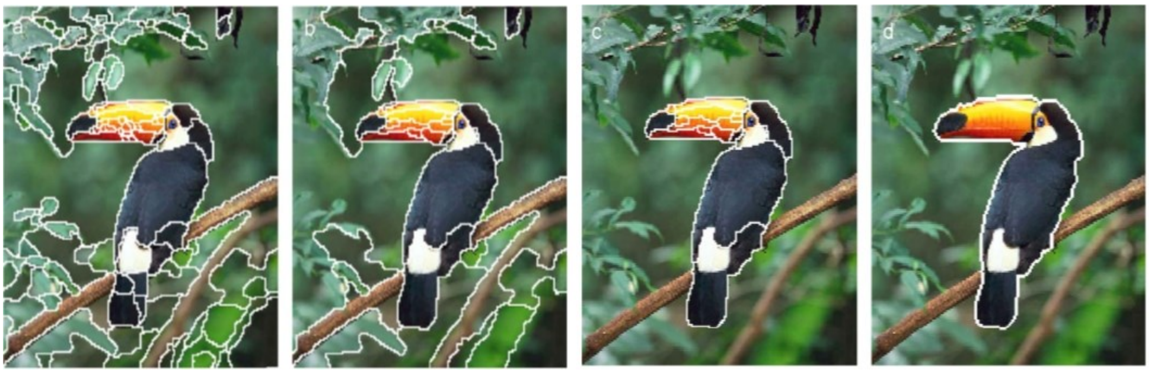


Figure 4-3. Left to Right: Region Merging algorithm iteratively merges neighboring regions [80].

Superpixels have been used at the core of numerous algorithms serving as a preprocessing block before the actual segmentation algorithm. Starting from these primitive regions, the segmentation is conducted by progressively merging *similar* neighboring regions according to a certain predicate, such that a certain *homogeneity* criterion is satisfied. Here, the two critical issues are the similarity and the homogeneity criteria. In other words, how to merge the underlying regions and when to stop merging. Whereas many methods utilize basic image properties such as color, texture, luminance in order to judge similarity between regions [71], there are also methods that use statistical properties [72] or graph properties [73], [74]. Stopping criterion can be defined rather straightforward via some homogeneity threshold based on the underlying similarity measure. Another criterion may also be imposed to limit the size of the merged region – not to allow it to grow above a certain limit in order not to lose local information.

Regardless of the similarity and stopping criteria used, most methods follow the same philosophy and progressively merge neighboring regions to obtain larger regions that cover the entire image as in Figure 4-3. In other words, they start from smaller superpixels and end up with larger superpixels. [P2] distinguishes itself from other methods mostly in this sense, such that the merged regions in [P2] are overlapping. Figure 4-4 illustrates the grouping algorithm that forms the overlapping regions, where each superpixel (call it “center superpixel”) is iteratively grouped with its similar neighbors, similar neighbors’ neighbors and so on until no similar superpixel is found within a limited radius of the center superpixel. Note that every superpixel in the image may belong to multiple regions at the end. This feature not only allows the algorithm to be more error tolerant, but also allows it to serve multiple purposes allowing various groupings. For instance, the branch in Figure 4-3 may or may not be intended to be grouped with the background. Allowing it to be grouped with both the object and background enables the algorithm to select the appropriate group in the next stage (see Section 4.3).

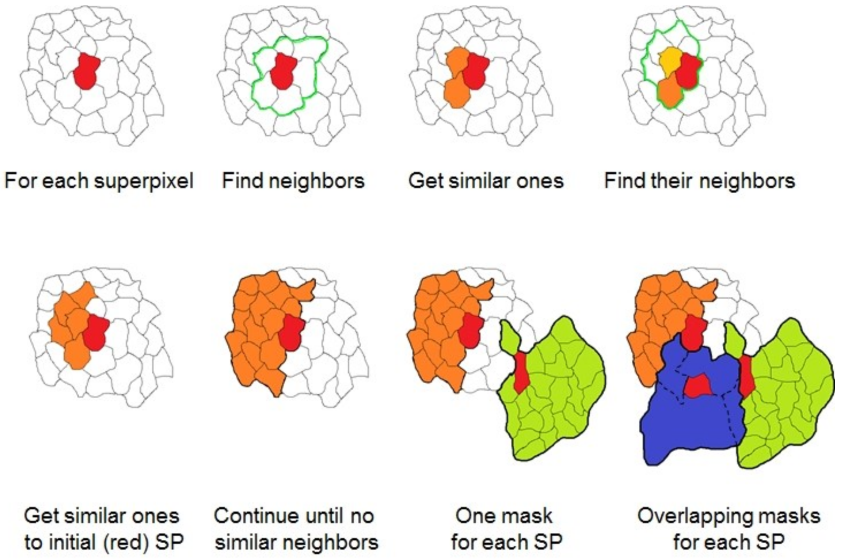

For each superpixel
Find neighbors
Get similar ones
Find their neighbors
Get similar ones to initial (red) SP
Continue until no similar neighbors
One mask for each SP
Overlapping masks for each SP

Figure 4-4. For each superpixel, a region is formed by merging it with its similar neighbors, neighbors' neighbors and so on until no similar neighbor is found or a maximum size is reached. Orange, green and purple regions above are formed by grouping superpixels around red superpixels.

Figure 4-5 illustrates the overlapping regions on a real image. Note that due to local similarities between foreground (FG) and background (BG), some regions may contain superpixels from both – such as the pink region in Figure 4-5. However, note also that there are other regions that cover the FG nicely without including any superpixels from the BG. The next step of the algorithm in [P2] selects the correct regions to be included in the final segmentation mask, which is discussed in the next section.

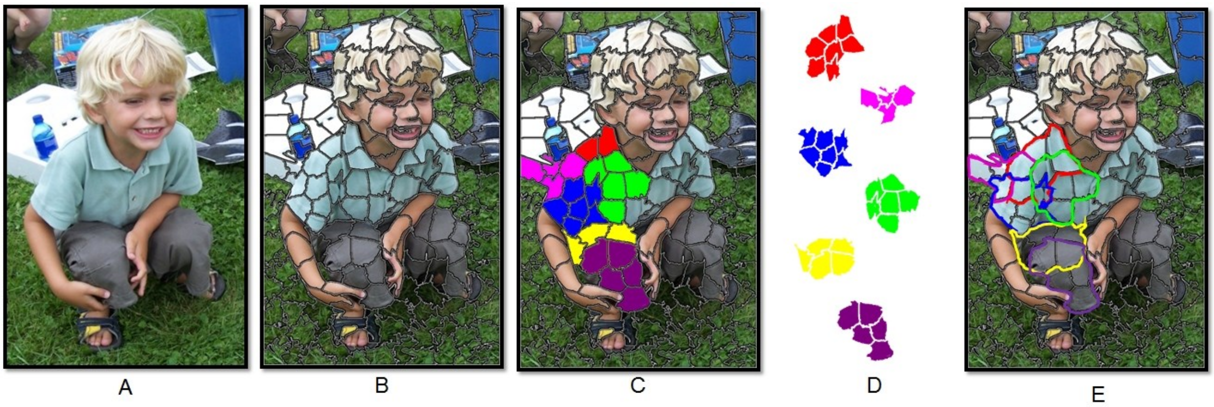


Figure 4-5. Overlapping regions formed by superpixel grouping as in [P2]. (A) Original image, (B) Initial superpixels, (C) Some overlapping regions formed by grouping superpixels, (D) Some individual regions, (E) Borders of the regions in C and D.

4.3. Interactive Object Segmentation

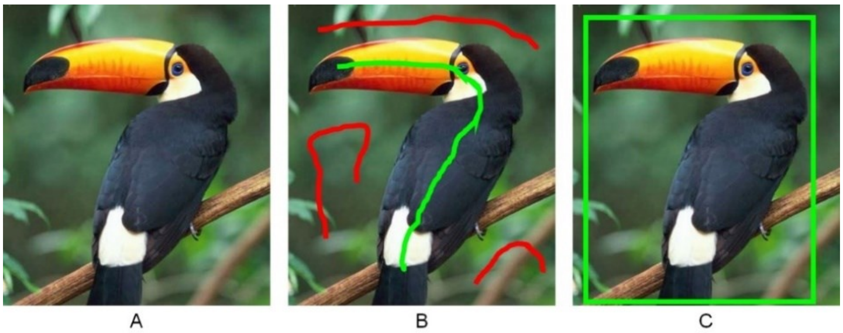


Figure 4-6. Popular methods of user interaction for image segmentation. (A) Original image, (B) FG and BG scribbles, (C) FG box.

Until this point, the definition of image segmentation we have discussed is to partition the entire image into meaningful regions as in Figure 4-1. However, there are various algorithms that define the problem as segmenting the image into only two regions: FG and BG instead of segmenting every object in the image all at once. This approach can also be referred as object extraction, since the ultimate goal of the process is to extract only the FG object by segmenting the image into two regions. In Section 2.2 we have discussed how figure/ground organization occurs in human perception, i.e. how our perception decides what is FG and what is BG. Those cues without a doubt shed light to a proper FG/BG segmentation; however, considering the variety of the context and applications where segmentation is required, defining such FG and BG becomes rather ambiguous – turning fully automatic image segmentation into an ill-posed problem without any semantic knowledge of the target object [75]. For instance, what is the FG object in Figure 4-1? Is it the mother and her child in the middle? Or just the child? Why do they belong together? Or are all people in the picture FG? What makes the people on the back FG but not BG? What if there were more people, would they all still be FG? The problem is that there is no definite correct answer to these questions. Specifically, based on the target application, we may even be interested in the BG, not FG. One way to answer these questions is semi-supervised image segmentation, also called interactive image segmentation, which aims to overcome such difficulties by taking advantage of user input for assistance. Typically, the user specifies FG and BG objects (more correctly, object of interest), or at least gives hints about them. The algorithm then uses these inputs to separate the image into FG and BG regions. Note that in most algorithms such input not only helps to decide what is FG and BG, but also assists the segmentation process via telling the algorithm what properties the object and background have so that similar pixels to the user input can be grouped together. In other words, instead of grouping pixels similar to each other, pixels similar to user input are grouped together.

Figure 4-6 shows some examples of user interactions used in most popular algorithms [76], [77]. In Figure 4-6.B the user provides separate scribbles to mark both FG and BG, and in Figure 4-6.C a box is drawn around the object of interest. Typically, since most methods use these inputs in their calculations, the user is waiting idly during segmentation. Next, since the initial result is rarely

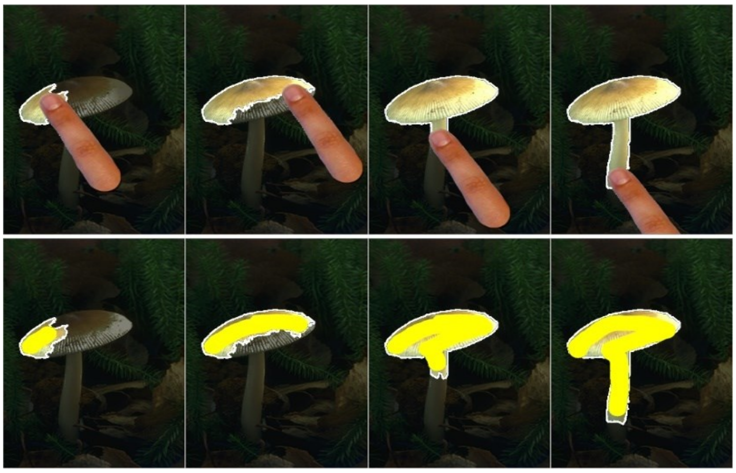


Figure 4-7. Top row: User interaction of the proposed algorithm from user's point of view (progresses left to right). Bottom row: User scribble used by the algorithm for hypotheses selection (invisible to the user).

satisfactory, the user provides more scribbles in order to fine tune the result. Whereas there are methods that try to overcome this iterative approach, they usually come with a loss in accuracy [78], [79].

In [P2], various algorithms and interaction techniques are discussed and a novel interactive segmentation algorithm is proposed. The main contribution of the algorithm is to provide effective results with minimal idle time for the user. This is achieved by moving the user interaction to the end of the entire process, so that no heavy computation is done after the user interaction. The interaction is built on the overlapping groups that are formed as described in Section 4.2. In [P2], these groups are referred as *hypotheses* since each region represents an alternative grouping. Remember from Section 4.2 that some of these hypotheses may nicely adhere to object boundaries and some may not due to local similarities between the object and the background. Such erroneous groupings do occur inevitably in any region merging algorithm due to the infinite possibilities of FG and BG compositions. Most algorithms try correcting such errors with further iterations, which in return increases the required user interaction, overall complexity and degrades the user experience. By treating the overlapping groups as hypotheses, [P2] skillfully minimizes such tedious operations. The user simply moves his/her finger (or the cursor in case of non-touchscreen devices) over the FG as if painting over the object and the segmentation mask automatically snaps to object boundaries. In fact, such interaction has already been proposed in [80]. However, it is built over the well-known graph cut algorithm [76] and modifies it in order to decrease complexity and achieve instant feedback so that the users feel like they are painting the FG object, which comes with a loss in accuracy. [P2] uses the same user interaction; however, what happens behind the scene is that the user's scribble is simply used for selecting which hypotheses belong to FG and which do not. Figure 4-7 shows the user interaction both from the user's and the algorithms perspective. While the user moves his/her over the object, the hypotheses that are painted by the scribble more than a predefined threshold are immediately included in the segmentation mask. Such a threshold naturally determines the error tolerance of the algorithm, or in other words, the sensitivity of the user interaction. If the threshold is too low, any hypothesis barely painted over will be included in the mask. On the other hand, if it is too high, the user will have to paint over almost the entire object to include it in the mask. By setting

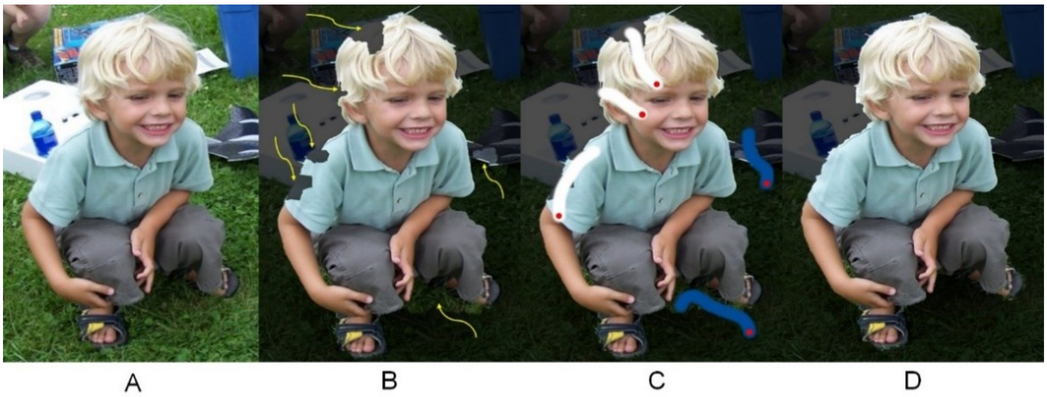


Figure 4-8. Fine tuning scribbles. (A) Original image, (B) Segmentation errors, (C) Fine tuning interaction with FG and BG scribbles (red dots indicate the starting end of the scribbles), (D) Final segmentation mask

it properly, a seamless interaction can be achieved where the user roughly scribbles over the object and extracts the FG object.

[P2] proposes a novel and proficient method for image segmentation. However, no matter how capable an algorithm is, it is possible to encounter segmentation errors due to various reasons. The aforementioned local similarities between FG and BG are one possible cause of errors. Having a complicated FG or BG is another reason such that small objects or object parts end up as hypotheses themselves since there are no similar superpixels around them to be grouped with. In such cases, as mentioned above, most algorithms require additional user input and re-iterate the algorithm. The user typically provides separate scribbles for FG and BG corrections. Such toggling is rather demanding particularly for touchscreen devices, since the user needs to select a different brush (i.e. FG or BG) in order to correct errors. Especially when heavy calculations are done after the user input, fine tuning becomes a tedious process as the user ends up switching back and forth and waiting idly in between until all errors are corrected. In [P2] a novel interaction method is proposed that eliminates the necessity of selecting a different brush for FG and BG corrections. Figure 4-8 shows the fine tuning scribbles provided by the user for an erroneous segmentation. Note how the scribble is regarded as FG scribble when it starts from FG, and BG scribble when it starts from BG. In other words, the user is not required to alter the brush between FG and BG, such selection is done automatically based on the starting location of the scribble. Whereas such an interaction is suitable for any algorithm that provides separate scribbles for FG and BG, no heavy computation should be performed between scribbles to achieve a smooth user interaction. During the initial interaction, [P2] uses user scribbles simply as a mask over the grouped regions. Similarly, in fine tuning stage they are used to paint over the initial superpixels. By doing so, a higher precision is achieved in error correction and also correcting possible errors that stem from grouping is enabled.

The proposed method in [P2] provides an efficient and effective image segmentation algorithm by proficiently incorporating human perceptual rules and considering “the user” as the upmost concern of the entire design. However, the main contribution of the whole scheme is not limited to its high performance, but can be listed as follows:

1) Inconvenient menu operations are avoided by utilizing only a single brush.
2) User’s idle waiting time is eliminated by performing all time consuming operations prior user interaction.
3) The method can handle significant amount of user error.
4) A novel fine-tuning method is proposed where both FG and BG corrections are enabled with automatically altering brushes in an intuitive manner.

These contributions are shown in [P2] with Figure 4-9 and Table 4-1 by comparing it to two of the state-of-the-art interactive segmentation methods that also perform superpixel grouping. Note here that the algorithms in [71] and [81] require different user inputs for both FG and BG. Moreover, the computation times for both [71] and [81] in Table 4-1 are spend after the user interaction, i.e. the user needs to wait idly during these time intervals. However, all the calculations for [P2] are performed before the user interaction, thus these intervals are not reflected to the user.

Table 4-1. Computation times (in seconds) for the tested images in Figure 4-9

Images	**[81]**	**[71]**	**[P2]**
Bird (163 x 192)	2.68	0.53	2.63
Flower (229 x 216)	4.13	1.15	1.84
Tiger (264 x 192)	8.49	2.14	1.94
Dogs (335 x 295)	5.01	1.14	1.85
Horses (481 x 321)	22.43	2.28	2.37
Sculptures (321 x 481)	33.01	5.44	1.95
MonaLisa (376 x 425)	13.21	3.49	1.64

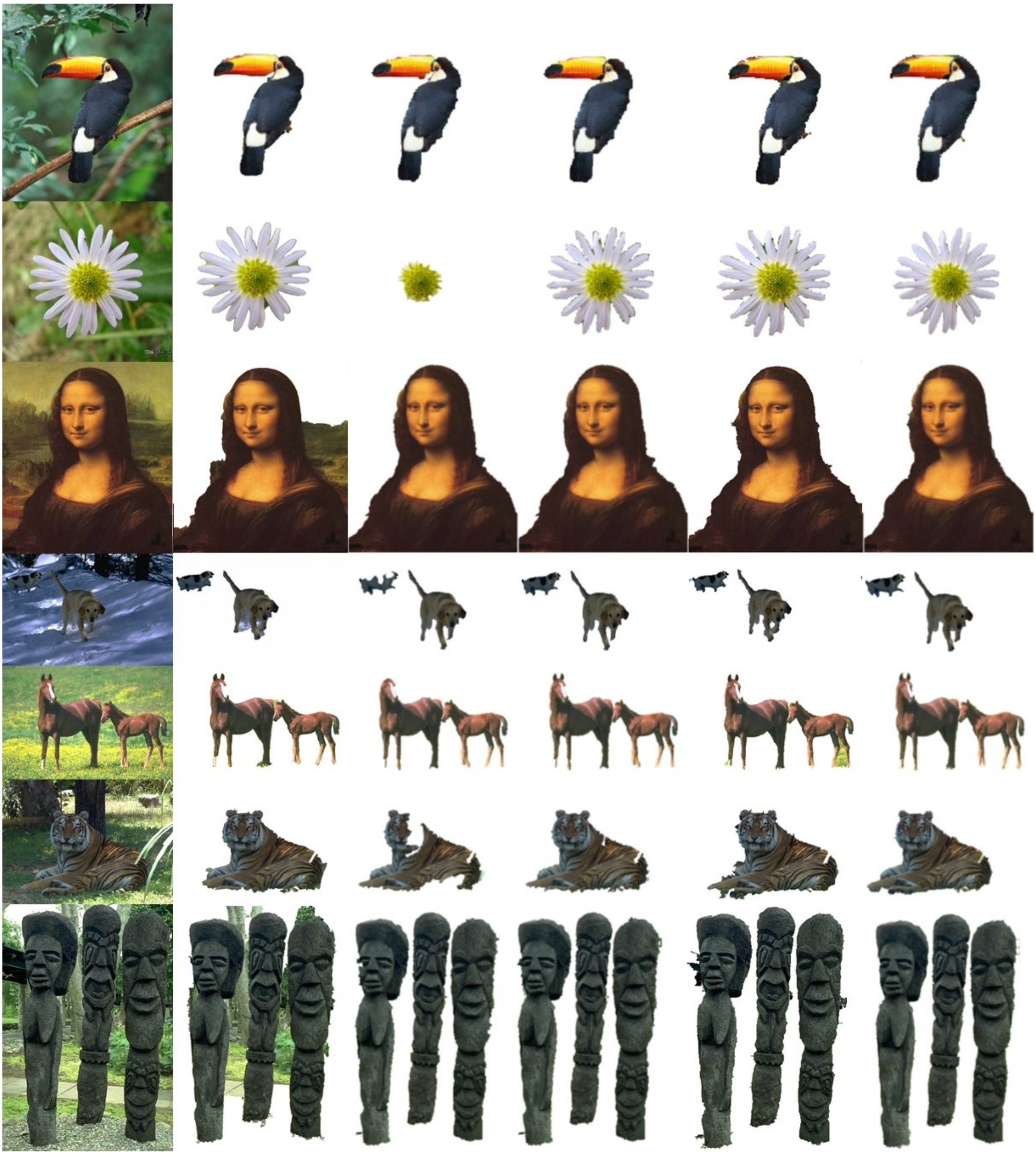


Figure 4-9. Experimental results for (top to bottom) bird, flower, monalisa, dogs, horses, tiger and sculptures. Left to right: Original image, [79], [81], [71], [P2] and ground truth. Presented images are the result of the initial interaction, i.e. no fine-tuning is performed.

Chapter 5

Feature Based Object Recognition

Objects are the most fundamental units of our visual perception. When we look around us, what we see are solid meaningful objects rather than lines, edges, uniform patches etc. Chapter 4 discusses how these objects are formed in our perception from their parts under the light of Gestalt Psychology which is also discussed in Chapter 2. However, what is important to us in our daily experiences is the assessment of what that object is – i.e. recognizing the object. This is indeed how we evaluate the usefulness of an object and act accordingly. Our actions in our daily lives, no matter how simple or complicated they are – be it drinking water, taking a bus, or playing football – basically all stem from recognizing an object and deciding what to do accordingly. Even discarding an object in our view requires us first to recognize it.

Recognizing an object is basically matching it to an already known object and categorizing it based on our past experience. Palmer [12] defines the four basic components of object recognition as follows:

- *Object representation*: The relevant characteristics of the object must be represented.
- *Category representation*: Each of the set of possible categories must be represented.
- *Comparison process*: There must be a way which the object representation is matched or compared to the category representation.
- *Decision process*: There must be a way to decide to which category the object belongs on the basis of comparison.

Based on the above scheme, objects are represented based on their characteristics and compared to each other. Additionally, certain decision criteria have to be set in order to assign the object to the appropriate category.

Note that the humans' ability to recognize objects is astonishing. We can recognize an object despite varying viewing conditions, orientations etc. This aspect of object recognition is referred as *object constancy*. Even if the object is viewed in a completely different lighting condition, or from a totally different angle, we can successfully tell what it is. Therefore, representation of the object and the category in question need to capture the commonalities across varying conditions and viewpoints.

One noteworthy theory of object recognition is the "Recognition by Components (RBC) Theory" which was proposed by Irving Biederman [82]. The theory proposes that objects can be specified as spatial arrangements of primitive volumetric components, which are called *geons*. Arcs, cylinders, spheres, blocks are all examples of geons and Biederman suggested thirty-six different geons. Figure 5-1 shows sample geons and sample objects that are formed from their combinations. Then, recognizing an object is simply matching a geon description of the target object with geon descriptions of object categories.

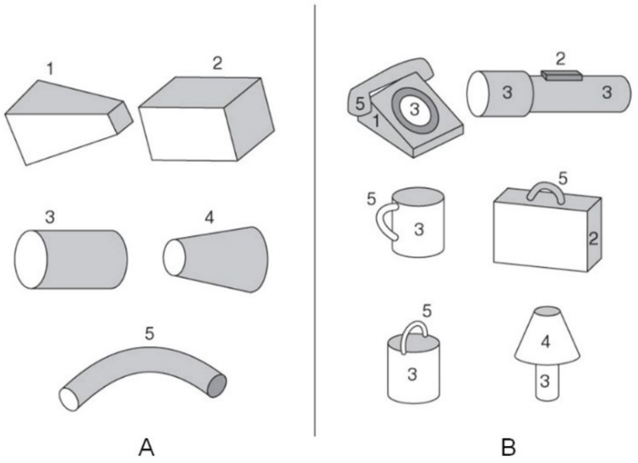


Figure 5-1. (A) Sample geons and (B) objects formed by combination of geons [82].

Note that the Gestalt view of perception is dependent upon the whole object and less so upon its individual features. However, Biederman claims that the Gestalt principles serve to determine the individual geons, rather than the complete object. A complete object, such as a chair, can be highly complex and asymmetric, but the components will be simple volumes. A consequence of this interpretation is that it is the components that will be stable under noise or perturbation. If the components can be recovered and object perception is based on the components, then the object will be recognizable [82]. On the other hand, one should notice that the perception of a chair is different than the perceptions of its parts brought together. In other words, the bucket in Figure 5-1.B may be composed of geons 3 and 5 in Figure 5-1.A, but its perception is different than a mere combination of the perceptions of its geons. Even though the bucket is composed of those geons, a new identity, new properties and a new perception arises when those geons come together to form the bucket – this is what Gestalt Psychology essentially underlines.

Given an input image and 3D model of an object (i.e. the "category" in Palmer's scheme), The classical approach to object recognition in computer vision is to interpret the image as a part of the model by checking whether its parts and spatial arrangements match the model. The image can be the view of the object from a certain viewing angle, certain distance etc. However, this method requires a 3D representation of the object. Another approach is to represent the object category by a small set of 2D views and match the input image to these 2D views. Hence, the problem scales down to matching 2D images and trying to find the same object in both of them. In fact, typically in computer vision, the problem is to construct the 3D model of an object from its various 2D images – which understandably requires recognizing the object in those images first. The most popular way of achieving this, is to take a similar approach to Biederman's Recognition by Components theory and try to match components of the objects in an image (i.e. local parts) to different images. Such an approach allows recognizing the object even under strong occlusions. Figure 5-2 shows a typical example of such a matching, where parts of objects are recognized in a complex scene. In order to achieve this, local patches of both images need to be represented, compared and matched following Palmer's scheme. A common name for these local patches and their descriptions is *local image features*.

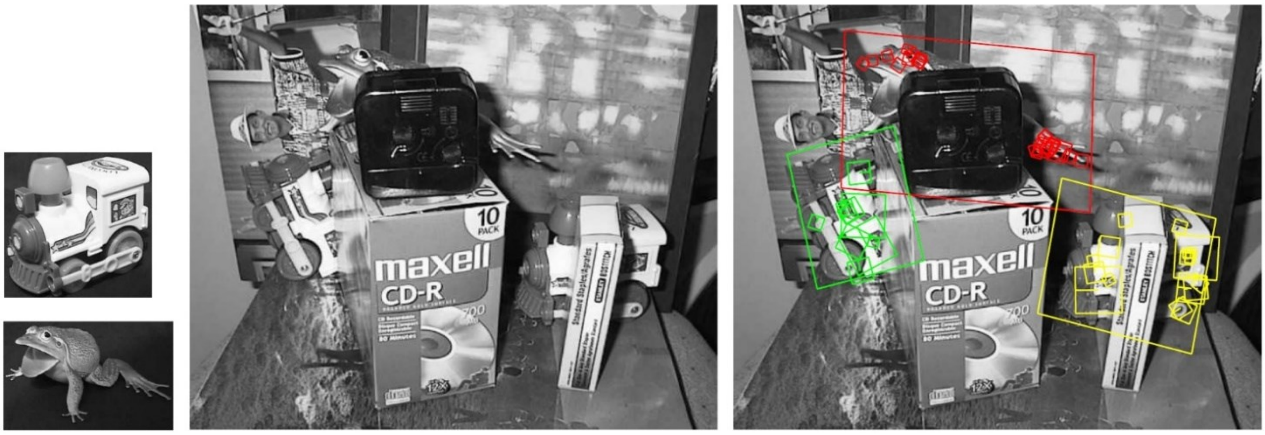


Figure 5-2. Object recognition via local image patches. Objects can still be detected despite occlusions, rotations and scale differences [83].

Local image features are being used to enable various applications. Automatic photo categorization in large photo albums, automatic tag suggestions in various photo sharing platforms such as Instagram, Facebook, Twitter are just the tip of the iceberg that are allowed by recognizing the object(s) in an image. They are also used in camera pose estimation, which is a prerequisite for constructing the 3D model of the object from its 2D images from multiple views. They can also be used in image alignment, e.g., for panorama imaging or video stabilization, simultaneous localization and mapping (SLAM) in robotics, video tracking etc.

5.1. Local Image Features

The idea behind using local image features in object recognition is to detect certain points in the image that are *stable* and *repeatable*, so that they will be detected on the object no matter what kind of transformation it is subject to. In other words, those points will be detected on the object on every image despite being viewed from another angle, another distance etc. Next, those points (more accurately, local regions around these points) are described uniquely via some descriptor in order to make it possible to identify and recognize the same point in different images. In order to do so, those descriptors are compared with each other and matched based on certain criteria. Finally, a decision is made based on those matches whether the objects in the images are the same or not.

5.1.1. Local Feature Detectors

Considering that the target of local image features is to recognize an object from its components, the locations where these local features are extracted are of crucial importance. Figure 5-3 shows some example patches from an image. Note here that if we search for the patch B in another image, it would be rather difficult to find the exact match since there is nothing that discriminates it from any other homogeneous patch except its color. Similarly, patch A is another difficult case since we only see a line segment, which can be placed anywhere along the line. Patches with gradients in at least two distinct directions (for example corners) are typically the easiest to localize. Note for instance how easy it is to locate patch C in Figure 5-3, on the other hand there are multiple alternative locations for patches A and B.

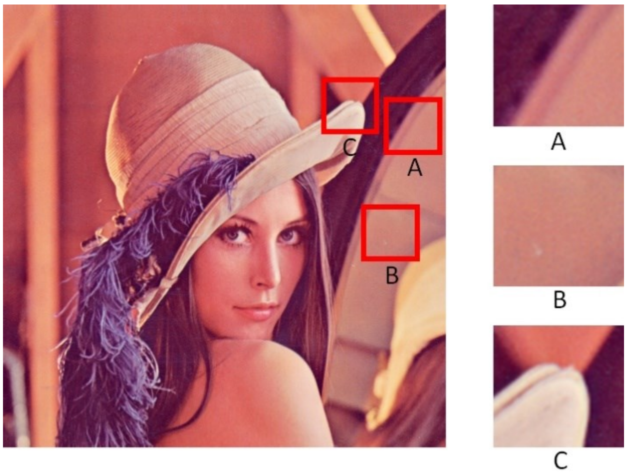


Figure 5-3. Local patches from various locations in the image. Corners are typically more distinctive than lines and homogeneous regions.

The history of local interest points (i.e. keypoints) can be dated back to 1980s and Moravec's corner detector [84], however they became truly popular after the milestone papers of Lowe in 1999 [85] and 2004 [83]. Lowe's proposal, Scale Invariant Feature Transform (SIFT), has been widely used ever since on most object recognition tasks. It uses the local minima and maxima of difference-of-Gaussian (DoG) function in order to detect stable keypoints that are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. In order to realize this, the image is convolved with Gaussian filters at different scales, and then the differences of successive Gaussian-blurred images are taken. Keypoints are then taken as minima/maxima of the difference images at multiple scales. Accordingly, each keypoint is assigned a scale which brings in scale invariance. Additionally, in order to achieve rotation invariance, each keypoint is also assigned an orientation based on local image gradient directions.

Numerous variations and improvements of SIFT have been proposed, such as SURF [86], KAZE [87], ORB [88], BRISK [89], where some of them bring in speed improvements and some provide increased stability and repeatability. For instance, SURF uses integral images in order to efficiently calculate convolutions and reduce computation time. In [90], authors proposed an accelerated version of KAZE (A-KAZE) and compared it with SIFT and the above popular feature detectors in terms of repeatability under different distortions such as blur, zoom and rotation, compression, viewpoint, noise and synthetic rotation. Their results show that A-KAZE outperforms all its competitors also in terms of computational complexity. The reader is referred to [91] for a complete survey of recent feature detection methods.

5.1.2. Local Feature Descriptors

Once keypoints are detected in different images, they need to be compared with each other in order to decide if they belong to the same object. Note that certain detectors assign a scale and an orientation to each keypoint. Hence, once these differences are compensated, one may expect to use image patches around these keypoints directly (i.e. via calculating the correlation of pixel intensities) to compare and judge similarity. However, even after such compensations, local appearances of image patches typically show significant differences making it hard – if not impossible – to match with each other via such a correlation measure. Therefore, certain features are extracted from those image patches in order to uniquely describe the region around each keypoint which allows recognizing similarities between similar patches and yet are distinct enough to realize the differences among them.

In theory, any image feature can be used as a local feature descriptor, such as color/intensity histograms and/or any texture or shape descriptor. However, the discriminative power of such global image features is typically not sufficient when it comes to matching local image patches. As mentioned above, multiple gradients such as corners are typical candidates for a keypoint. Therefore, the descriptor should be able to characterize these gradients appropriately. Gradient histograms are good examples for such descriptors which are proposed by Lowe as a part SIFT (hence commonly referred as SIFT descriptor). Lowe designed his descriptor similar to the response properties of complex neurons in the visual cortex, in which a feature position is allowed to vary over a small region while orientation and spatial frequency specificity are maintained. This is basically based on the experiments of Edelman et al. [92] that simulated the responses of complex neurons to different 3D views of computer graphic models, and found that the complex cell outputs provided much better discrimination than simple correlation-based matching. Their experiments showed that matching gradients while allowing for shifts in their position results in much better classification under 3D rotation. This can be seen, for example, if an affine projection stretches an image in one direction relative to another, which changes the relative locations of gradient features while having a smaller effect on their orientations and spatial frequencies [93]. Accordingly, Lowe created a local descriptor by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint. These samples are then accumulated into orientation histograms (see Figure 5-4).

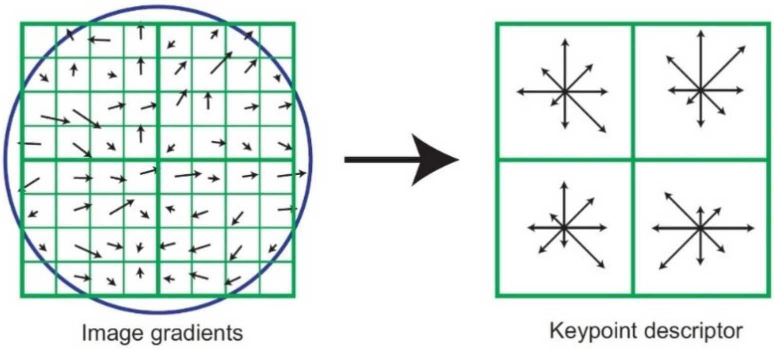


Figure 5-4. SIFT Descriptor extracted from 8x8 sample region into a 4x4 descriptor. Gradients are weighted by the Gaussian window indicated by the blue [83].

One group of local descriptors worth mentioning are the so called binary descriptors. Considering that storing and comparing high dimensional floating-point data becomes a problem particular in case of large image datasets, binary descriptors are designed to be efficient both in computation and size. They are typically built from a set of pairwise intensity comparisons and each bit in the descriptor is the result of one comparison. In [95] the authors provided a comparative study of popular binary descriptors BRIEF [94], ORB [88] and BRISK [89]. They also compared them with respect to other floating-point descriptors. They have found SIFT to have the best overall performance; however, significant performance gain can be achieved via binary descriptors which can be useful in many applications. Another noteworthy conclusion the authors made in [95] is that after experimenting different detector/descriptor pairings, they showed that the best performance did not always correspond to the original authors' recommendations.

5.1.3. Feature Matching

Once the local features are detected and their descriptors are extracted, these descriptors are compared with each other in order to judge their similarity – ultimately to decide whether they belong to the same location on the same object or not. Comparison is based on the distance between descriptor vectors, such as Euclidean, Mahalanobis, EMD [96] etc. Note that the feature matching process can easily be computationally expensive considering that matching two images involves comparison of all possible keypoint pairs and most applications require comparing multiple images. Therefore, some methods utilize smart data structures and indexing methods, such as multidimensional search trees or hash tables, in order to reduce the computation load caused by feature matching [97]. Muja et al. demonstrated in [97] that these can speed the matching of high-dimensional vectors by up to several orders of magnitude compared to linear search. However, no matter how successful and discriminative the feature descriptors are, incorrect matches are unavoidable particularly due to difficult image content such as cluttered backgrounds, repetitive patterns etc. When matching SIFT descriptors, Lowe proposed to discard a match if there is any uncertainty of its correctness. He achieved this by comparing the distance of the best matching feature to the second best match and rejecting any match if their distance ratio is greater than 0.8, which he claims to eliminate 90% of the false matches while discarding less than 5% of the correct matches.

One popular method for decreasing the number of false matches is to use geometric verification methods and consider matches that do not fit the geometric transform as outliers. However, this method assumes that we already know the nature of the geometric relation between the images. Random Sample Consensus (RANSAC) [98] is a parameter estimation algorithm commonly used in object recognition, where parameters of the transformation between images are estimated based on the matched features. RANSAC is an iterative approach, starting from a random set of samples. For each iteration, a transformation is obtained from the set and all other points are tested for consistency. An error is calculated between the obtained and assumed model if sufficiently large number of inliers are found to be consistent. It can also operate in the presence of outliers, this is why many algorithms utilize it for the sole purpose of discarding incorrect matches. However, the estimated transformation is also useful for various applications such as camera pose estimation, 3D registration etc. However, apart from its randomness and heavy computational cost, inliers need to be dominating outliers for it to give accurate results. Chum and Matas proposed a method in [99] called Progressive Sample Consensus (PROSAC). They claimed that they achieved significant speed improvements over RANSAC by assuming that the ordering by similarity computed on local descriptors works better than random selection. This also means that the similarity measure predicts correctness of a match better than random selection over the whole set of matches.

5.2. Neighborhood Matching

The study in [P3] proposes a feature matching algorithm that innately avoids incorrect matches. By relying on Gestalt laws of perception, the method induces some structural restrictions on the matches so that not only incorrect matches are avoided but also additional correct matches can be found. The idea stems from the famous Gestalt motto “the whole is different than the sum of its parts”. In fact, what is done in regular straight-forward feature matching is simply summing up the individual matches in order to draw conclusions about the whole object. Instead, [P3] suggests that we need to consider those individual matches together in order to be able to draw accurate conclusions about the object, i.e. the whole. An interesting observation is also made in [P3], where matching local features independently is related to the perception of a man who has a disorganized vision due to brain damage. The man explains his confusion in recognizing objects and how he perceives parts of different objects belonging together as follows:

> *“If I saw a complex object, such as a person, and there were several people in my field of view, I sometimes saw the different parts of the people as not, in a sense, belonging together, although... if a given person moved so that all the parts of him went in one direction, that would... tend to make him into a single object. Otherwise there was this confusion of lots of things, all of which were there, but did not seem to belong together... Several of these cases of things not belonging together gave quite absurd results. For*

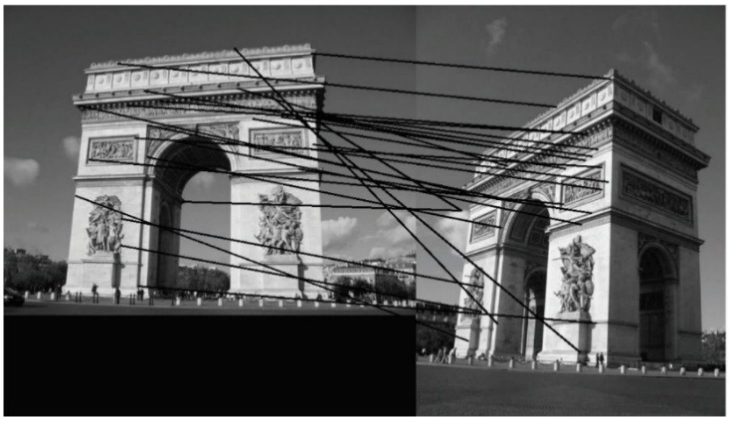


Figure 5-5. Regular matching with distance ratio 0,5. 28 matches.

> *instance, I do remember one case where there was what seemed to me to be one object which was partly motor car, partly tree and partly a man in cricket shirt. They seemed somehow to belong together. More frequently, however, a lot of things which to any ordinary viewer would be parts of the same thing were parts of different things." [12]*

Figure 5-5 shows two pictures of "Arc de Triomphe" taken from different angles. They are matched using their local features. SIFT features extracted from both images are matched using regular feature matching. Note that even though the distance ratio method proposed by Lowe is applied, numerous incorrect matches are visible. Some of these are caused by the repetitive patterns on the object.

Figure 5-6 illustrates the algorithm proposed in [P3] by comparing it to regular matching. The idea is to match the neighboring points together, instead of matching them individually. In other words, if point $\boldsymbol{X}$ is matched to point $\boldsymbol{Y}$, $\boldsymbol{X}$'s neighbor $\boldsymbol{X_N}$ is likely to have a match in the neighborhood of $\boldsymbol{Y}$. If $\boldsymbol{X_N}$ matches to a point far away from $\boldsymbol{Y}$, one of the matches is likely an incorrect match. So [P3] proposes that in order to match $\boldsymbol{X}$ to $\boldsymbol{Y}$, certain number of matches should also be found in their neighborhood.

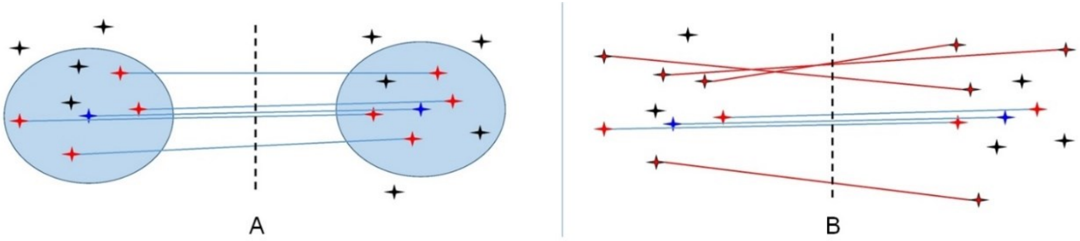


Figure 5-6. Neighborhood matching (A) compared to regular matching (B). Points are matched only if their neighboring matches also agree on the match. While incorrect matches are filtered out (matches exist in B but not A), it also introduces correct matches (matches exist in A but not B) by relaxing the similarity constraints.

One critical point in the above description of the algorithm is the definition of "neighborhood". [P3] mentions two options: a fixed size neighborhood – which should be selected very carefully not to invalidate scale invariance, or using K-nearest neighbors which selects the K closest keypoints as the “neighbors”. However, a keypoint may not have any other keypoints in its close vicinity. This can easily invalidate the idea of proximity. Another method, which is not mentioned in [P3] is to utilize the scale information of the keypoint, if available. For instance, SIFT keypoints are assigned a scale and orientation when they are detected. Hence, for a matching pair $\boldsymbol{X} \leftrightarrow \boldsymbol{Y}$, if a certain size of neighborhood is selected for point $\boldsymbol{X}$, $\boldsymbol{Y}$’s neighborhood should be in proportion to their scales.

The above definition successfully filters out the incorrect matches if their neighborhoods do not agree on the match, provided that the same matching criteria is applied on all keypoints. However, the method in [P3] is also capable of increasing the number of initial matches. This can be achieved by relaxing the matching criteria for the neighborhood matches. Such tolerance is admissible since the candidate matches are highly restricted by the proximity limitation. Figure 5-7 shows how neighborhood matching improves the recognition process by filtering out incorrect matches. However, it is clear that regular matching cannot be performed with the criteria in Figure 5-7 since it creates abundant incorrect matches. It should be noted that both regular and neighborhood matching methods have their own optimal criteria. Therefore, it will be a fairer comparison to compare Figure 5-7.B to Figure 5-5.

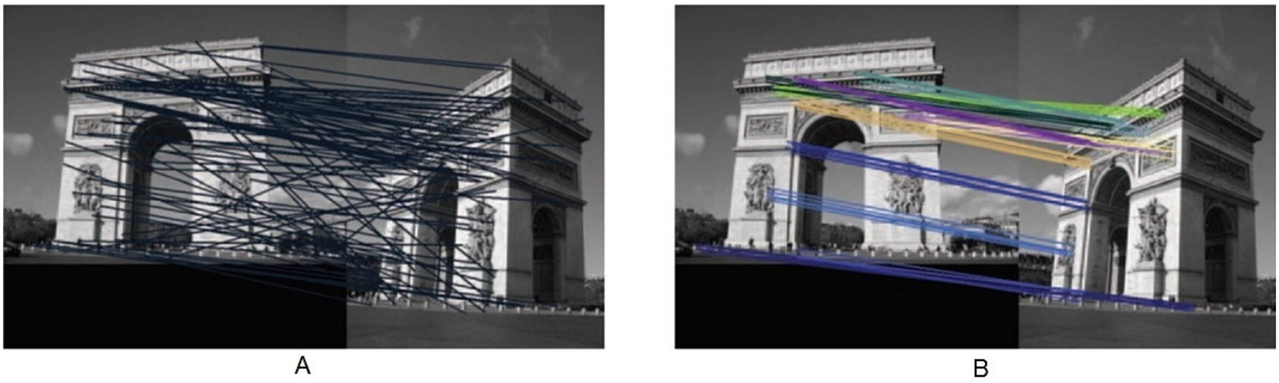


Figure 5-7. Regular matching (A) has 101 matches, compared to Neighborhood matching (B) has 70 matches. Both methods use distance ratio 0,64. Each color in B represents a different matched neighborhood.

Note that neighborhood matching is not an alternative to geometric verification methods such as RANSAC or PROSAC. In fact, it improves their performance by decreasing the number of incorrect matches. Considering both RANSAC and PROSAC are iterative algorithms, they will converge much faster given that neighborhood matching increases the ratio of inliers.

5.3. Feature Integration

Recognizing an object based on its components is indeed how we perceive objects, and local image features together with neighborhood matching realize this approach fittingly. However, trying to recognize a random object that is present in two images solely based on local features does not always give the intended result – and in fact it does not reflect our way of perception entirely. Note that what we try to achieve with local image features is to capture the object’s local information.

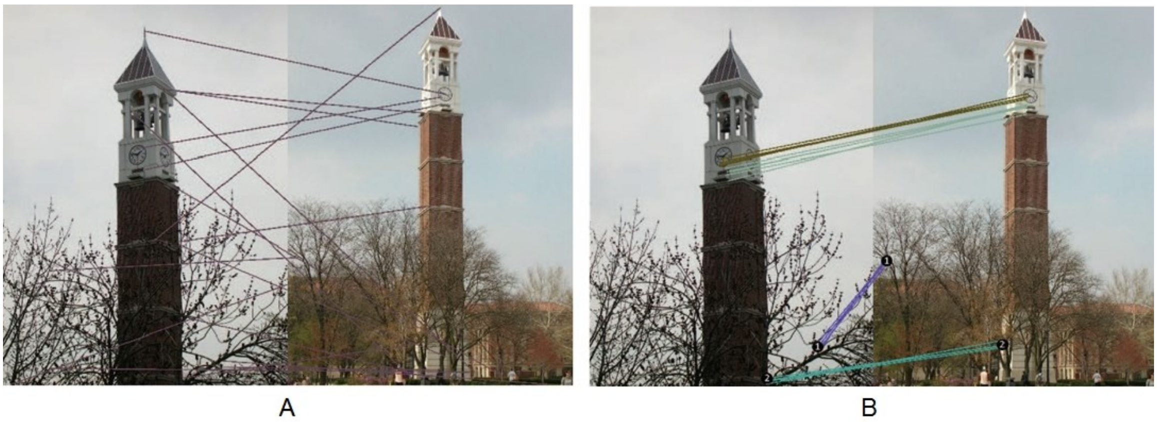


Figure 5-8. Local feature matching performance is usually affected by strong textured regions (A). Whereas neighborhood matching removes most of the incorrect matches, it is still possible to end up with erroneous results (B).

However, if the image has more content than the object itself, that extra content may easily degrade the performance of the algorithm. Figure 5-8 shows an example of such a case, where background objects interfere with local matches introducing incorrect matches. Particularly strong textured regions tend to yield abundant number of keypoints, which in return can easily create false positives. Typically, neighborhood matching eliminates most of those incorrect matches too, but it may not remove all of them (see Figure 5-8.B).

Feature Integration Theory, proposed by Treisman et al., suggests that attention must be directed to each stimulus in a display whenever combinations of features are needed to characterize or distinguish the possible objects presented [100]. They also state that before focused attention, the visual system contains separate representations of features as in Figure 5-9. Each of those "feature maps" is organized based on the locations in space and is constructed independently from the others, and when we focus on an object appropriate feature maps from its location are activated and combined. Accordingly, feature integration theory is analyzed in two stages: pre-attentive and focused attention. Even though the theory is targeted mainly on the role of the focused attention, it also has significant proposals on the pre-attentive stage. For instance, Treisman also states that their findings suggest a convergence between two perceptual phenomena – parallel detection of visual targets and perceptual grouping or segregation. Both appear to depend on a distinction at the level of separable features. Neither requires focal attention, so both may precede its operation [100].

When we look at feature integration theory from object recognition perspective, in order to recognize an object our focus of attention needs to be on that object. Then the features of that object are extracted and bind together for us to perceive that object. In other words, features from different spatial locations are not used when the focus is on the object. Figure 5-8 can be thought as doing otherwise, where features from other heavily textured areas are used while we were trying to match object features.

In [P4], a basic implementation of the pre-attentive stage of feature integration theory is realized in order to overcome the aforementioned problems. Large heavily textured areas are detected and

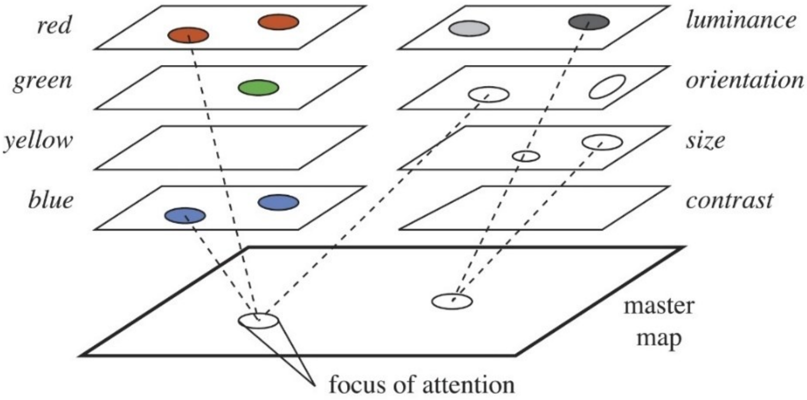


Figure 5-9. Treisman's feature integration model of early vision. Individual maps can be accessed in parallel to detect feature activity, but focused attention is required to combine features at a common spatial location [101].

excluded from the object recognition process. Consequently, color and texture features are used in order to describe those textured regions, and local image features are used to describe the remaining regions. By doing so, not only the descriptive powers of multiple distinct features are harvested, but also the processing time significantly improved since textured areas typically introduce abundant number of keypoints which takes substantial time to detect, extract descriptors and match. Figure 5-10 shows how the algorithm reduces the number of keypoints significantly.

The method in [P4] is proposed as a means for automatic image and video annotation. Naturally, images with similar content (i.e. that has the same objects and/or similar regions) are expected to have similar annotations. Thus, a query image is compared to a database of annotated images and similar images are used to annotate the query image automatically. Annotation performance is measured with 85 query images on a database of 2360 manually annotated images from [102]. Whereas a slight improvement in both precision and recall are reported in [P4], the main improvement is observed in the computational complexity. The total time for the extraction of SIFT descriptors and matching them across the set of query images improved almost 40% compared to the time for extracting and matching the SIFT descriptors from the non-texture regions, and extracting and matching the texture regions. Moreover, an image does not have to include a prominent object, particularly when it comes to annotating generic image databases. For instance, landscape content such as ocean, forest, sunset, sky, grass are typical texture areas that can establish the entire content themselves without any object. Local image features obviously would not be of any use in those cases. Using texture or color descriptors on the other hand would provide better image matching, hence better annotations.

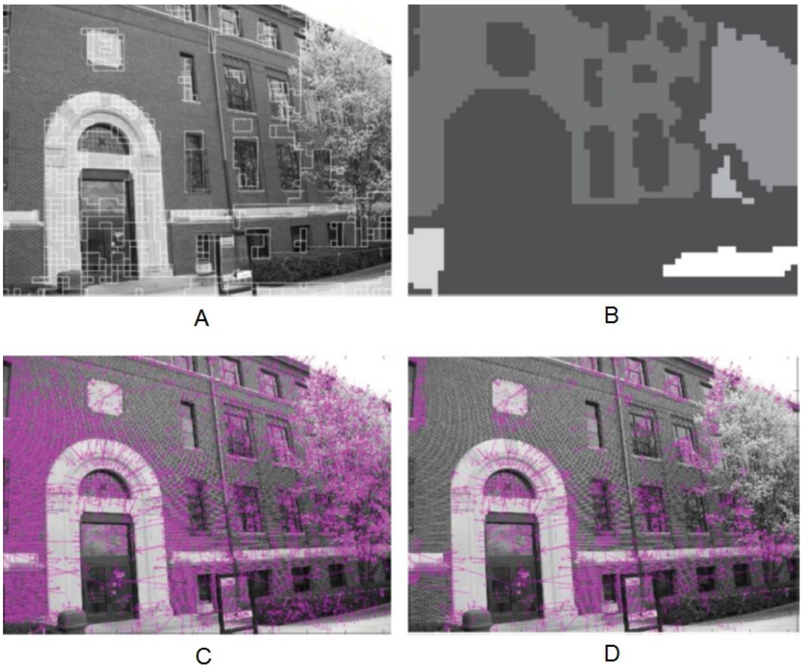


Figure 5-10. Original image with overlaid region boundaries (A). Detected regions. Each region is shown as a different shade of gray (B). Detected local keypoints on the original image (C). Detected local keypoints after applying the texture mask (D).

Chapter 6

Video Shot Change Detection

Digital videos take advantage of the apparent motion mentioned in Chapter 2, where motion arises from the rapid presentation of completely static images – so called *video frames*. In this sense a video frame is the most basic building block of a video. A video frame is technically no different than a digital image, which in return enables any image processing technique also available to digital video processing. However, apparent motion received considerable attention from Gestalt psychologists in order to demonstrate the emergent properties in perception. As they propose, our perception of motion is also different than the sum of its parts – i.e. it cannot be reduced to simple relations between frames. Hence, in order to analyze motion, a larger portion of the video is required than a single frame. Thompson et al. defines a video shot as the smallest unit of visual information captured at one time (i.e. uninterruptedly) by a camera that shows a certain action or event [103]. Based on this definition, in order to understand the video content properly, detecting video shots from the entire video is of crucial importance. Such understanding enables various applications, such as video summarization, indexing & retrieval, numerous post-processing and video editing applications. For example, video summarization typically selects a certain number of frames to represent a shot (or an excerpt of the shot, depending on the application), and then uses these frames from each shot to represent the video. For instance, if we want to see whether "Barack Obama" is in a certain video or not, we can simply skim through those frames instead of watching the entire video. These scenarios are particularly gaining importance in the current digital age, where the amount of video content has been growing with an astonishing speed. YouTube, globally the 3rd most popular website, announced in 2016 that 300 hours of video are being uploaded to the website every minute – up from 72 hours in 2013, and the number of hours people spend watching videos on YouTube is up by 60% year-over-year. With this much content being created and consumed, manual processing of these videos is out of question. Automatic, efficient and proficient tools are necessary and essential.

Our visual perception is particularly sensitive to sudden changes in the visual field such as objects appearing/disappearing, or changes in color, structure etc. Bearing in mind the concepts of attention, semantics, familiarity etc.; we are capable of detecting even small changes particularly when they occur abruptly. In fact, psychological tests measuring *change blindness*[1] typically places a blank frame between two different frames when they are presented to the subject. The subject sees the first frame, then a blank frame is shown, and then another frame appears asking the subject to spot any difference if present. It is observed that subjects are often blind to large changes, suggesting that any visual disruption that masks the location of the change can induce change blindness [104]. Correspondingly, when a shot abruptly changes in a video, we can immediately detect it without any difficulty. However, not all shot transitions in a video are abrupt. Video creators often use artistic effects and transitions for the benefit of special effects or better story telling. As a result, many shot transitions occur in a gradual fashion instead of changing abruptly. Figure 6-1 shows both an abrupt change and a dissolve type gradual change from one shot to another. Whereas it is still easy for a human observer to detect that a shot change has occurred even when the transition is gradual, our perception reacts in a different way to gradual changes compared to abrupt changes. This has been investigated in [104] from a perspective of change blindness. When subjects were provided different images with changes that are detectable when the original and modified image were swapped instantaneously without a visual disruption, they were able to detect 97% of addition/deletion changes and 92% of color changes. On the other hand, when subjects were presented with either a disruption (a blank frame in between two images) or when the change is presented as a gradual dissolve from one image to another, their detection rates are significantly reduced (~60% for addition/deletion, ~35% for color changes). This is because gradual changes are noticed consciously, i.e. we need to evaluate the new situation and judge it to be different from the previously analyzed situation. Because of this, gradual change blindness can be decreased by increasing the viewer's ability to consciously perceive differences in their surroundings [104].

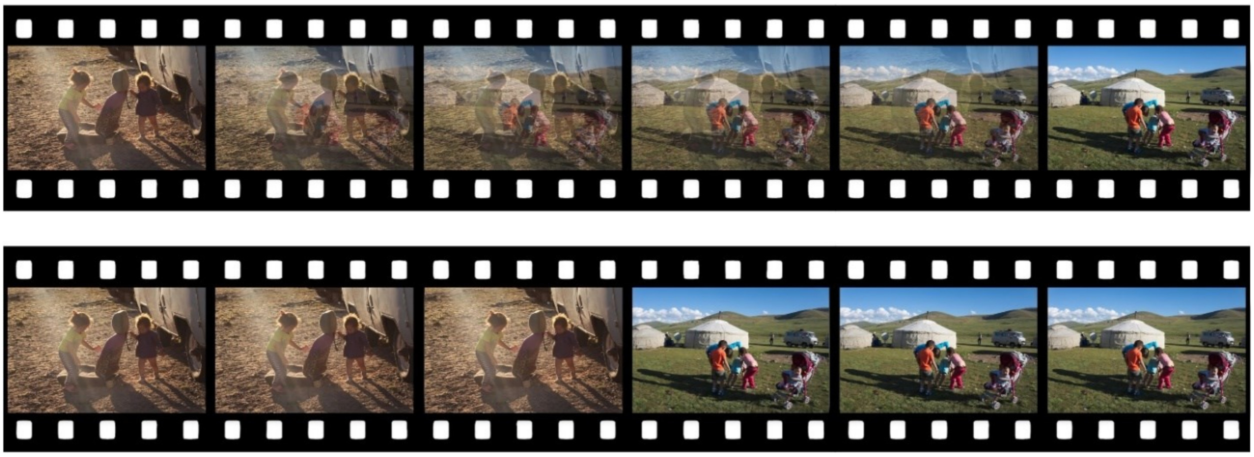


Figure 6-1. Gradual (above) and abrupt (below) shot changes in a video. Note that whereas it is easy to tell when the shot changes in an abrupt change, it is relatively hard to pinpoint when exactly the change occurs when shots are dissolved into each other.

[1] Change blindness is defined as the induced failure to detect major changes in an image.

Figure 6-2 is taken from a perceptual test performed in Computational Visual Cognition Laboratory in MIT, where the beginning and the end frames of a gradual change are shown in the figure. There are multiple differences between two images, such as the sign of the market on the left, interior of the market, color of the building door, middle window on the balcony and the entire building on the right. It is rather easy to notice some of these changes when two images are next to each other and we keep looking back and forth between the two of them. However, when the content changes gradually from Figure 6-2.A to Figure 6-2.B, these changes – even the entire building – can easily pass unnoticed.

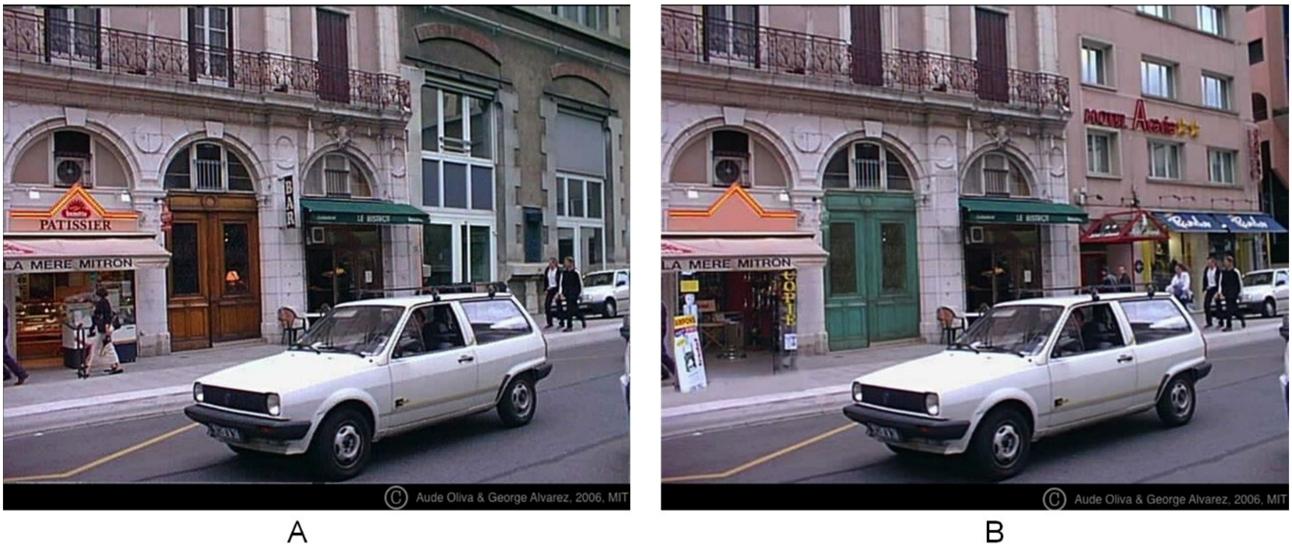

PATISSIER
LA MERE MITRON
BAR
© Aude Oliva & George Alvarez, 2006, MIT
A
LA MERE MITRON
© Aude Oliva & George Alvarez, 2006, MIT
B

Figure 6-2. Multiple changes occur from A to B. However, when the changes are gradual, they are easily missed even if an entire building is replaced [105].

The difficulty of detecting gradual changes relative to abrupt changes is also reflected in video processing algorithms, most of which typically suffer from low detection rates for gradual transitions. The problem arises from the fact that during a gradual change, the similarity between two consecutive frames is too high. As mentioned above, gradual changes are noticed consciously; so the algorithm needs to be smarter than simply comparing consecutive frames since a gradual change cannot be detected that way. In [106], it has been shown that a holistic approach boosts the ability to detect changes. The study shows that when our perception forms holistic forms, i.e. Gestalts, our ability to detect any changes that happens to that Gestalt is superior to the case where no Gestalt is formed. Even though this improvement comes at the cost of the ability to identify the change, it is evident that a holistic top-down approach is beneficial for change detection.

6.1. Information Seeking Mantra

In addition to perceptual psychology, the field of human computer interaction is also highly interested in how humans perform visual search. In the context of shot change detection, visual information search may point us to where shot changes occur in the video. From this perspective the well-known information visualization technique, namely the Information Seeking Mantra [107], follows a fitting holistic approach for the shot change detection problem. The mantra aims to visualize the data in such a way that it guides users to what they are looking for in a fast and efficient way. Therefore, it adopts a top-down approach by following its famous motto:

Overview first, zoom and filter, then details-on-demand.

Overview provides a general context for understanding the data by taking a viewpoint that comprises the whole data. From this perspective, major components and their relationships to one another are made clear. Simply the overall shape of the data itself can provide assistance in understanding the information at hand. Significant features can be discerned and selected for further examination. Such features might not be readily viewable from another part of the data representation or might be obscured from certain viewpoints. Revealing these features at the outset can aid the user in filtering the extraneous information so that they can complete their task more efficiently by excluding unimportant aspects of the representation [108]. Zooming and filtering both involve reducing the complexity of the data representation by removing unnecessary information. By taking a closer viewpoint to the data, unnecessary data is naturally filtered out and items of interest are brought into view. Once the entire information is trimmed to the "region of interest", details of one or a group of items can be provided.

From the perspective of shot change detection, if the problem is defined as visually searching the frame where the shot change occurs, the path that "Information Seeking Mantra" suggests is a fitting approach where the frame of interest (i.e. shot boundaries) is sought among all video frames. Following the Mantra, we need to first overview the whole video, then zoom in to shot boundaries and filter out the uninteresting frames (non-boundary frames). The overview phase will then reveal the relations between frames and direct us to the part where the shots change. Then, when we zoom in to those parts we can precisely tell where the shot changed.

6.2. Top-Down Shot Change Detection

[P5] proposes a technique, a *modus operandi* so to speak, for detecting shot boundaries based on the aforementioned Gestalt principles and Information Seeking Mantra. As discussed at the beginning of this chapter, a holistic top-down approach significantly improves perception of changes, hence is suitable for shot change detection. What the Mantra suggests is also a top-down method where we start from the overview of the data and go down (i.e. zoom in) until individual elements. Accordingly, [P5] suggests the video to be first overviewed, and then zoomed in to the region of interests. Here, region of interest is clearly limited by the shot boundaries. But how do we overview a video and reveal where shot boundaries are? How do we zoom in afterwards? Considering that a video is a continuous flow of information, [P5] proposes to uniformly sample this information to bring out its overview. Therefore, every N^{th} frame is sampled and compared with the consecutive sample, i.e. $(N+1)^{th}$. If a large enough content change is detected between two sampled frames, then it is concluded that a shot change has occurred somewhere between those samples. Then, the algorithm

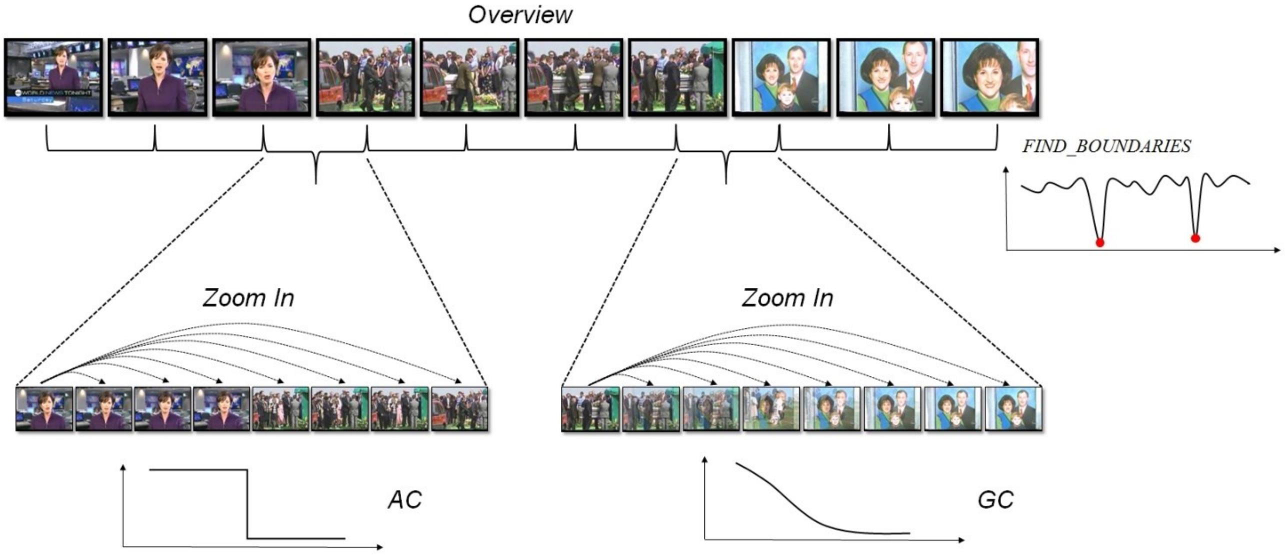


Figure 6-3. Summary of the algorithm proposed in [P5] showing the overview and zoom in phases. The algorithm uniformly samples the video and zooms in wherever there is a content change. This way, both gradual and abrupt changes can be detected.

zooms in to that interval by analyzing only the frames between *Nth* and *(N+1)th*. In the zoom in phase, a new sample is taken by gradually decreasing the distance from the *(N+1)th* frame. The overview and zoom in phases are illustrated in Figure 6-3. Note how the intervals where no shot change occurs are omitted and the algorithm only zooms in wherever there's a significant content change.

The top-down processing strategy saves significant computation time and resources by avoiding redundant processing of irrelevant frames, yet this is not its only benefit – it also enables effective detection of gradual changes. Considering that the content similarity between two consecutive frames is significantly high, by taking a broader view, the content change from one shot to another is naturally realized by the overview method. Remember the discussion on how the changes in Figure 6-2 are easily noticeable when the beginning and end frames of the gradual change are presented. However, when the changes occur gradually, subjects often fail to notice most of the changes. Therefore, by sampling the frames before and after the change, the overview of the video (conceptually) transforms the gradual changes into abrupt changes making them easier to detect. It is of course possible that the overview frame may end up being sampled among the frames of the gradual change. In that case its content may be similar to both shots. That case is easily handled in [P5] by using two shifted overviews so that if one overview encounters the case and fails to realize the content change, the second will catch it. However, even this method "may" fail to catch gradual changes longer than the sampling period. Therefore, an acceptable duration for a shot change should be considered while deciding the sampling period. In fact, certain datasets mentioned in [P5] contain transitions as long as seven seconds, which may be considered as shots themselves.

One important concept in the above method is how we judge the content change/similarity. How do we say if the content has changed from one frame to the other? There are various methods and features to judge image similarity (for instance one example is the color descriptor described in Chapter 3). In [P5], an object based similarity measure is utilized. In other words, if two frames have

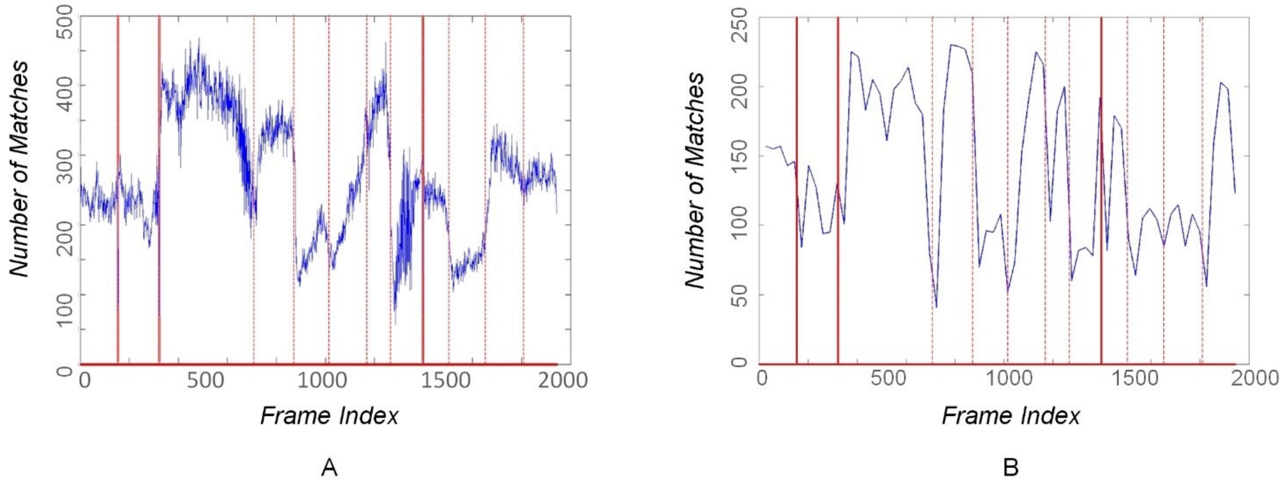


Figure 6-4. Number of matched keypoints when consecutive frames are compared in a video (A) and when consecutive frames of the video overview are compared (B). Horizontal lines denote where shot changes occur – dashed lines show gradual changes and solid lines show abrupt changes.

the same objects, they are considered to be from the same shot. Of course, it is possible that the same object is present in two separate consecutive shots. However, changing background or several foreground objects impact frame similarity which can be detected easily if proper measures are used. In Chapter 5, local image features and their utilization in object recognition have been discussed in detail. Figure 6-4 shows how the number of matched keypoints changes when we compare the consecutive overview frames instead of comparing every consecutive frame in the video. Whereas significant local variations in Figure 6-4.A makes it difficult to realize most gradual changes, Figure 6-4.B shows clear variations at each shot change. The clear difference in the ability to discern shot changes demonstrates the power of using video overview. Besides, [P5] takes a step further and relieves the similarity measure from the effect of the number of keypoints present in each frame by normalizing the number of matches keypoints with the number of keypoints on each frame. In other words, if a frame with K keypoints is compared to a frame with L keypoints and M matches are found, the rate of similarity between those frames is calculated as:

$$R = \frac{2M}{K + L} \tag{5-1}$$

The above measure of similarity is more effective than simply using the number of matched keypoints and signifies shot changes even more clearly. Note how the peaks in Figure 6-5.A at the locations of shot changes become deeper in Figure 6-5.B making them easier to discern from others.

Remember from the earlier discussion in Chapter 5 that using Gestalt principles improves the performance of keypoint matching and hence object recognition. The algorithm in [P5] harvests those benefits by utilizing the same approach proposed in [P3] and discussed in Section 5.2. By doing so, better and more reliable keypoint matches are obtained; however, that is not its only benefit. Remember also the discussion in the beginning of this chapter about the study in [106], and how

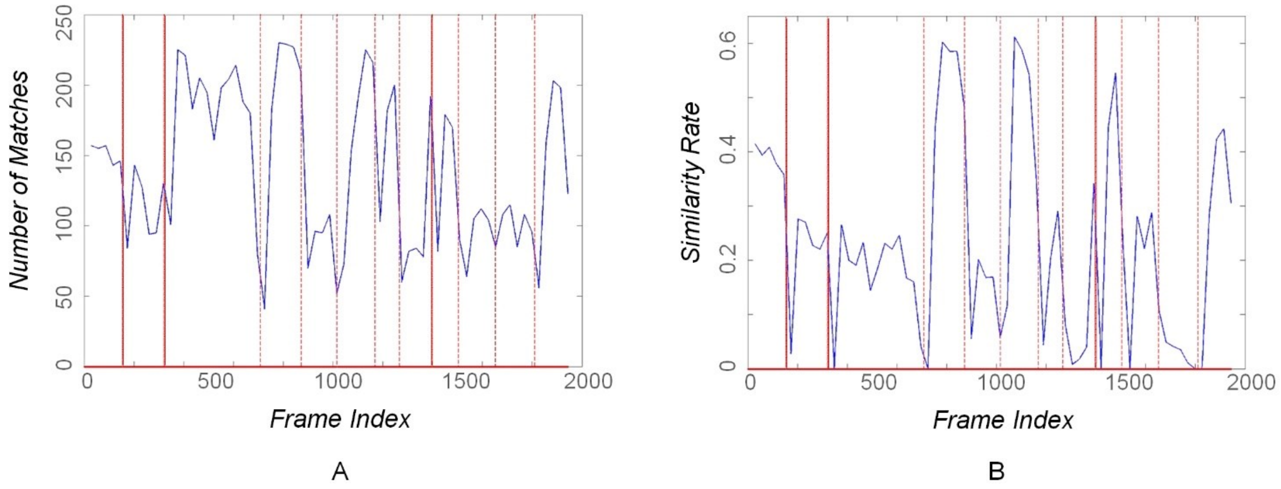


Figure 6-5. By normalizing the number of matches with the number of keypoints, a more affective similarity measure is obtained (B) compared to the case where number of matches are used as is (A).

forming holistic forms, i.e. Gestalts, help our ability to detect changes compared to the case where no Gestalt is formed. This effect is in fact demonstrated in Figure 6-5, where Gestalt grouping signifies shot changes more clearly compared to the case where no grouping is made.

The performance of [P5] has been evaluated on a dataset of 8 videos, total length of 91460 frames and containing 689 transitions (585 abrupt and 104 gradual). This dataset is mainly used to demonstrate the advantages of the proposed *modus operandi*; hence, it is compared to [109] where the same idea of content change is employed (i.e. object recognition and local keypoint matching) but processes the video in an opposite manner to [P5]. In other words, [109] starts from the zoomed in phase and compares every single consecutive frame, and zooms out whenever it suspects a shot change. So, in a way, it realizes the same idea that a broader view is necessary to detect shot changes. This is in fact reflected to their remarkable detection performance of 93% precision and 96% recall. However, the path they follow results in exhaustive feature detection, description and matching making the method impractical. In fact, [P5] achieved an on par performance (84% precision, 96% recall) in less than 13% of [109]'s processing time. This clearly demonstrates the benefit of the proposed overview and zoom in way of analysis.

TRECVid is a project which had an activity track for video shot boundary detection from 2001 to 2007 joining 57 different research groups in order to determine the best approaches [110]. Despite providing valuable insight, working on predefined development and test databases induces several (dis)advantages. For instance, development and test datasets have significantly similar contents. The fact that 6 out of top 10 performing algorithms using flash detectors, which is a very specific case commonly appearing in news videos (which also constitutes most of the TRECVid dataset), is also a clear indication that the competing methods were tuned to perform only for the specific TRECVid dataset. Therefore, as expected, machine learning methods dominate the top performing algorithms (9 out of 10). The single non-machine learning approach is also vastly tuned to work on the TRECVid dataset. To be exact, their system is composed of a cut detector, a flash detector and a dissolve detector (78% of the gradual transitions in TRECVid dataset are of dissolve type).

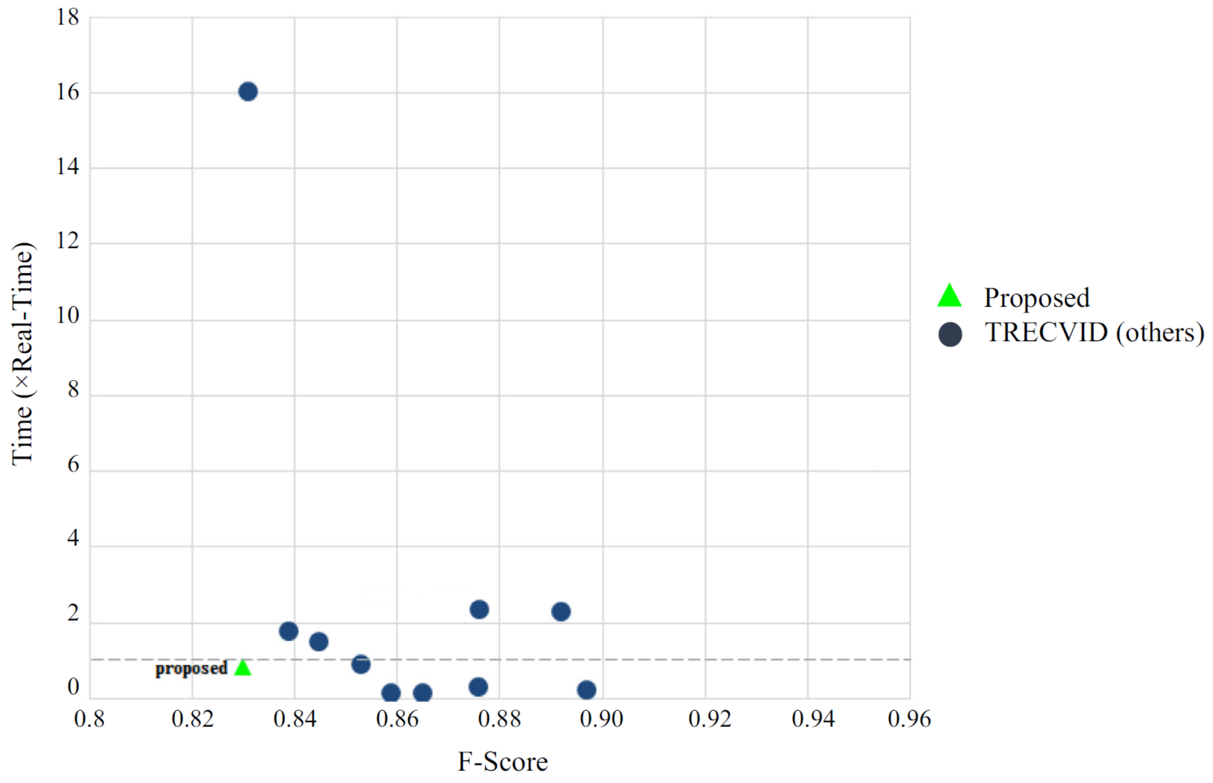


Figure 6-6. Performance vs. Computation Time in TRECVid dataset. The horizontal line denotes the speed of real-time operation.

The TRECVid dataset contains 12 videos (7h, 744,604 frames) and has 4535 total transitions (60.8% abrupt, 39.2% gradual changes). Figure 6-6 shows the retrieval performance and computation time of the algorithm proposed in [P5] compared to the top performing algorithms in TRECVid. Despite the aforementioned controversial objectivity of the dataset and the clear advantage of the machine learning algorithms, the proposed algorithm in [P5] performs on par with the leading algorithms.

Chapter 7

Conclusions

Understanding the content of an image or video is at the core of image and video analysis. Whether it is as simple as colors and edges in an image, or as complicated as recognizing people and their actions, understanding and consuming such content is already a big part of our lives. In fact, we humans are not the only ones who consume that content any more. Computers already started taking the lead in content analysis tasks. Our cameras do not simply capture an image any more, they tell us who or what is in it and suggest annotations. Security cameras do not simply stream the video content to a hard drive for someone to inspect any more, they detect intruders themselves without even confusing them with family members or dog, they detect fire or house leakage, they analyze traffic and arrange it accordingly. Satellites not only detect natural disasters, but also started predicting them. In other words, computers already started understanding the content of an image or video as well as we do. All these tasks – almost always – require a certain amount of human supervision; because at the end, the visual capabilities of humans are still far superior to computers. However, this is not simply for the reason that our brains function faster than their processors. No matter how much computation power you have, if you do not know how to use it, it will never be enough. From this perspective, teaching computers how to analyze an image – how *we* analyze an image – and helping them “understand” what they “see” is the key to achieving more capable and self-sufficient computer systems. Unfortunately, even we humans do not precisely know every detail about how our perception works. What we do know is that our perception is not as straightforward as simple rendering of the visual input we receive. Therefore, our knowledge on human visual perception is based on theories that not only organize and explain known facts, but also make predictions about new ones.

In this thesis, we have shown how understanding human visual perception through these theories can help computer algorithms to be more efficient and more effective. It is in fact fascinating that these improvements have been achieved by incorporating relatively simple rules. However, it is rather expected as well since this is also how our perception works. We often do not even think any explanation is necessary for our perception. In fact, when you try explaining perceptual theories to people who are outside this field, you usually end up facing blank expressions. Because to them, what these theories tell are so obvious and clear. To them, it is no different than saying “sky is blue” or “snow is white”. However, the purpose and importance of these theories become clearer when you try teaching human perception to a computer since you need to give very specific instructions on where to look and what to look for. You realize that there are rules to follow – no matter how simple – in order to match our perception.

In Chapter 3 we discussed color perception and a perceptual color descriptor. Instead of handling each color independently in a pixel-wise manner, we extracted the dominant colors and their spatial inter-relations by processing the image in a top-down manner. Such a method has proven to achieve a significant improvement compared to pixel-wise processing. There are of course various color descriptors that fairly describe the color composition of an image. However, we have shown that when we follow simple perceptual rules and describe that content similar to how humans perceive it, superior outcomes can be achieved with great efficiency. Similarly, in Chapter 4, the process of perceptual image segmentation is discussed and an interactive image segmentation algorithm is presented. The algorithm follows the steps of our perception, starting from image pixels to form uniform regions, and then group those regions to form objects. User interaction is simply used to select the foreground object. By moving all significant computations before the user interaction, we minimized the user’s idle time to achieve a better user experience. In the end, compared to the state-of-the-art algorithms, the resulting algorithm is remarkably simple and successful – just as our perception is. Similar grouping principles are also utilized in object recognition, which we explored in Chapter 5. We have discussed how humans recognize objects from their smaller regions and how various algorithms successfully utilized that fact. However, as we mentioned above, our perceptions are never mere rendering of the visual input we receive. Therefore, by applying perceptual grouping principles, local image features are grouped and matched together with their neighboring features. As a result, the efficiency of recognizing objects from their local regions is significantly improved without introducing any complexities. Chapter 6 discusses the problem of detecting shots changes in a video. It has been shown that if we approach the problem from the perspective of visual perception and handle it the same way as humans detect changes, shot changes reveal themselves naturally. Therefore, the video is processed in a top-down manner in order to first reveal the rough whereabouts of the shot boundaries, and then zooming in to those locations in order to determine the exact boundary location. It has been shown that tremendous reduction in computational complexity can be achieved without sacrificing performance by following such a scheme.

It is rather intriguing to see that in all the topics that are covered in this thesis, all state-of-the-art methods have a glimmer of perceptual methodology. However, they mostly fail to stay on the right path and end up with either excessive computational loads or insufficient performance. The proposed solutions in this thesis to the abovementioned problems take such methodology as a basis, and follow the steps of our visual perception. By doing so, it has been shown that complicated and cumbersome algorithms can be avoided, and impressive performance can be achieved if such an approach is taken.

This thesis aims to put forward a mentality, a mode of work or a way of thinking. Taking a perceptual perspective in image processing is of course not limited to the topics tackled in this thesis. Computational photography, robotics, medicine, surveillance, virtual reality, artificial intelligence, human–computer interaction are only few examples that can benefit from such a standpoint. Besides, in addition to psychology, other disciplines such as cognitive neuroscience, biology or even sociology significantly contribute to the quest of understanding our perception. Therefore, fully understanding how our perception works and applying it on any image processing task requires a multidisciplinary research undertaking the topic from different perspectives. In this regard, extending ones knowledge into a wider spectrum of disciplines shall be a fitting immediate step towards producing innovative and effective solutions to the aforementioned problems.

> *Creativity, more often than not, comes about through the interaction of different disciplinary ways of seeing things.*
>
> *−Ken Robinson*

References

[1] Desolneux A., Moisan L. and Morel J.-M., “*From Gestalt Theory to Image Analysis*”, Springer-Verlag New York, 2008.

[2] Pineo D. and Ware C., "*Data Visualization Optimization via Computational Modeling of Perception*," in IEEE Trans. on Vis. and Comp. Graphics, vol. 18 (2), pp. 309-320, Feb. 2012.

[3] Li Y., Sawada T., Shi Y., Kwon T. and Pizlo Z., “*A Bayesian model of binocular perception of 3D mirror symmetric polyhedral*”. Journal of Vision, vol. 11(4):11, pp. 1-20, 2011.

[4] Li Y., Sawada T., Latecki L.J., Steinman R.M. and Pizlo Z., “*A tutorial explaining a machine vision model that emulates human performance when it recovers natural 3D scenes from 2D images*,” Journal of Mathematical Psychology, vol. 56, pp. 217-231, 2012.

[5] Riordan D., Doody P., Walsh, J. “*The Use of Artificial Neural Networks in The Estimation of the Perception of Sound by the Human Auditory System*,” Int. Journal on Smart Sensing & Intelligent Systems, vol. 8 (3), pp. 1806-1836, Sep 2015.

[6] Masia B., Wetzstein G., Didyk P., Gutierrez D., “*A Survey on Computational Displays: Pushing the Boundaries of Optics, Computation, and Perception,*” Computers & Graphics, vol. 37(8), pp. 1012–1038, 2013.

[7] Rämö, J., Marsh, S., Bech, S., Mason, R. and Jensen, S. H., “*Validation of a Perceptual Distraction Model in a Complex Personal Sound Zone System*,” in Audio Engineering Society Convention, vol. 141, 2016.

[8] Pizlo, Z., Li, Y., Sawada, T. and Steinman, R.M., “*Making a Machine That Sees Like Us*,” New York, NY: Oxford University Press, 2014.

[9] Murphy T., Morison A.M., “*Can I Reach that? An Affordance Based Metric of Human-Sensor-Robot System Effectiveness,*” HCI, Theory, Design, Development and Practice, vol. 9731, pp. 360-371, 2016.

[10] Noll P., "*MPEG digital audio coding*", IEEE Signal Processing Magazine, pp. 59—81, Sept. 2001.

[11] Sullivan G. J., Ohm J. R., Han W. J. and Wiegand T., "*Overview of the High Efficiency Video Coding (HEVC) Standard*," in IEEE Trans. on Circuits and Syst. for Video Tech., vol. 22 (12), pp. 1649-1668, Dec. 2012.

[12] Palmer S.E., “Vision Science: Photons to Phenomenology”, The MIT Press, 1999.

[13] Wagemans J., Elder J. and Kubovy M. “*A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure–ground organization*,” Psychological Bulletin, vol. 138, pp. 1172–1217, 2012.

[14] Watt J., Borhani R. and Katsaggelos A.K., “*Machine Learning Refined*”, Cambridge University Press, 2016.

[15] Goodfellow I., Bengio Y. and Courville A., “*Deep Learning*”, MIT Press, 2016.

[16] Wertheimer M., Spillmann L. (ed.) “*On Perceived Motion and Figural Organization*,” MIT Press, 2012.

[17] Steinman R.M., Pizlo Z. and Pizlo F.J., "*Phi is not beta, and why Wertheimer's discovery launched the Gestalt revolution*", Vision Research, 40 (17), 2257–2264, 2000.

[18] Rubin E., "*Figure and ground*" In D.C. Beardslee & M. Wertheimer (Eds.), Readings in perception, pp. 194–20, New York: Van Nostrand, 1958.

[19] Palmer, S.E. "*Perceptual organization in vision.*" In: Pashler, H., editor. Stevens handbook of experimental psychology: Vol. 1 Sensation and perception. 3rd ed. New York, NY: Wiley; pp. 177-234, 2002.

[20] Pao H.K., Geiger D., Rubin, N., "*Measuring Convexity for Figure/Ground Separation*", in Proc. of the 7th IEEE International Conference on Computer Vision (ICCV),vol 2, pp.948 - 955, 1999.

[21] Koffka K., "*Principles of Gestalt Psychology*", New York: Harcourst. 1935.

[22] Hoffman, D.D., Singh, M., "*Salience of visual parts*", In: Cognition, 63, 29–78, 1997.

[23] Palmer S.E., Ghose T., "*Extremal edges: a powerful cue to depth perception and figure-ground organization*", Psychological Science, 19(1), pp. 77-84, 2008.

[24] Vecera S.P., Vogel E.K., & Woodman, G.F. "*Lower-region: A new cue for figure-ground assignment*", Journal of Experimental Psychology: General, 131, 194-205, 2002.

[25] Hulleman J., Humphreys G.W. "*A new cue to figure–ground coding: Top–bottom polarity*", Vision Research, 44(24), 2779-2791, 2004.

[26] Palmer S.E., & Brooks J.L. "*Edge-region grouping in figure-ground organization and depth perception*", Journal of Experimental Psychology: Human Perception and Performance, 34(6), 1353-1371, 2008.

[27] Peterson M.A., "*Low-level and high-level contributions to figure-ground organization: evidence and theoretical implications*", In: Wagemans J, editor. The Oxford Handbook of Perceptual Organization. NY: Oxford University Press, 2014.

[28] Peterson M.A., Harvey E.H., Weidenbacher H. L., "*Shape recognition inputs to figure-ground organization: Which route counts?*" Journal of Experimental Psychology: Human Perception and Performance, 17, pp. 1075-1089, 1991.

[29] Peterson M.A., Gibson B.S., "*Object recognition contributions to figure-ground organization: Operations on outlines and subjective contours.*" Perception & Psychophysics, 56, pp. 551-564, 1994.

[30] Peterson M.A., Lampignano D.L., "*Implicit memory for novel figure-ground displays includes a history of border competition*", Journal of Experimental Psychology: Human Perception and Performance, 29, pp. 808-822, 2003.

[31] Peterson M.A., Enns J.T. "*The edge complex: Implicit perceptual memory for cross-edge competition leading to figure assignment*", Perception & Psychophysics, 14, pp. 727-740, 2005.

[32] Gordon, I.E., "*Theories of Visual Perception*", John Wiley, Chichester, 1997.

[33] Goethe, J.W, Miller D. (ed.), "*Goethe: Scientific Studies*", Princeton University Press, 1988.

[34] Livingstone M.S., Hubel D.H., "*Psychophysical evidence for separate channels for the perception of form, color, movement, and depth*", Journal of Neuroscience, vol. 7, pp. 3416–3468, 1987.

[35] Arend L.E., Reeves A., "*Simultaneous color constancy*", J. Opt. Soc. Am. A ,3 (10), 1743-1751, 1986.

[36] Troost J.M. "*Perceptual Constancy: Why Things Look As They Do?*", Cambridge University Press, 1998.

[37] Gegenfurtner K.R., Kiper D.C., Fenstemaker S.B., "*Processing of color, form and motion in macaque area V2*", Vis. Neurosci. 13, 161-172, 1996.

[38] Zeki S., Shipp S., "*Modular Connections between areas V2 and V4 of macaque monkey visual cortex*", Eur. J. Neurosci., 1 (5), 494-506, 1989.

[39] Bloj M.G., Kersten D., Hurlbert A. C., "*Perception of three-dimensional shape influences colour perception through mutual illumination*", Nature 402, 877-879, 1999.

[40] Anderson R.M., "*Visual Perceptions and Observations of an Aphakic Surgeon*", Perceptual and Motor Skills: vol. 57, pp. 1211-1218, 1983.

[41] Bowmaker J.K., Dartnall H.J., "*Visual pigments of rods and cones in a human retina*", The Journal of Physiology, 298:501-511, 1980.

[42] Svaetichin G., "*Spectral response curves from single cones*", Acta Physiol. Scand. 39 (134), pp. 17-46, 1956.

[43] Fairchild M.D., "*Color Appearance Models*", John Wiley & Sons, 2013.

[44] Dartnall H.J.A., Bowmaker J.K., Mollon J.D., "*Human visual pigments: microspectrophotometric results from the eyes of seven persons*", Proceedings of the Royal Society of London, B 220, 115-130, 1983.

[45] Hurvich L.M., Jameson D., "*An opponent-process theory of color vision*". Psychological Review 64 (6, Part I), pp. 384–404, 1957.

[46] De Valoris R.L., Smith C.J., Kitai S.T., Karoly A.J., "*Response of single cells in monkey lateral geniculate nucleus to monochromatic light*", Science, 127(3292):238-9, 1958.

[47] Anstis S., "*The Purkinje rod-cone shift as a function of luminance and retinal eccentricity*", Vision Res., 42(22):2485-91, 2002.

[48] Sheila M.I., Vicki J.V., Janice L.N., "*A new look at the Bezold–Brücke hue shift in the peripheral retina*", Vision Research, vol. 44 (16), pp. 1891-1906, 2004.

[49] Berlin B., Kay P., "*Basic color Terms: Their Universality and Evolution*", Berkeley, CA: Univ. Of California Press, 1969.

[50] Roberson D., Davies I., Davidoff J., "*Color categories are not universal: Replications and new evidence from a stone age culture*", Journal of Experimental Psychology: General, vol. 129, pp. 369-398, 2000.

[51] Broek E.L. van den, Kisters P.M.F., Vuurpijl L.G., "*The utilization of human color categorization for content-based image retrieval*", Proc. of Human Vision and Electronic Imaging IX, pp. 351-362, San José, CA (SPIE, 5292), 2004.

[52] Mojsilovic A., Kovacevic J., Hu J., Safranek R. J., Ganapathy K., "*Matching and Retrieval based on the Vocabulary and Grammar of Color Patterns*", IEEE Trans. on Image Proc., vol. 9(1), pp. 38–54, 2000.

[53] Rogowitz B., Frese T., Smith J., Bouman C. A., Kalin E., "*Perceptual Image Similarity Experiments*", Proc. of SPIE, Human Vision and Electronic Imaging III, vol. 3299, pp. 576-590, 1997.

[54] Agudo J.E., Pardo P.J., Sánchez H., Pérez A.L., Suero M.I., "*A Low-Cost Real Color Picker Based on Arduino*", Sensors, 14(7), 11943-11956, 2014.

[55] Swain M.J., Ballard D.H., "*Color indexing*", International Journal of Computer Vision, vol. 7 (1), pp. 11–32, 1991.

[56] Tran L.V., Lenz R., "Compact colour descriptors for colour-based image retrieval", Signal Processing, vol. 85, pp. 233–246, 2005.

[57] Zoidi O., Tefas A., Pitas I., "*Visual Object Tracking Based on Local Steering Kernels and Color Histograms*", in IEEE Trans. on Circ. and Syst. for Video Tech., vol. 23, no. 5, pp. 870-882, 2013.

[58] Kim W., Kim C., "*Background Subtraction for Dynamic Texture Scenes Using Fuzzy Color Histograms*", in IEEE Signal Processing Letters, vol. 19, no. 3, pp. 127-130, 2012.

[59] Deng Y., Kenney C., Moore M.S., Manjunath B.S., "*Peer group filtering and perceptual color image quantization*", in Proc. of IEEE International Symposium on Circuits and Systems, ISCAS, vol. 4, pp. 21–24, 1999.

[60] Gabbouj M. and Kiranyaz S., "*Audio-visual content-based multimedia indexing and retrieval - the MUVIS framework*", Proc. of the 6th Int. Conf. on Digital Signal Processing and its Applications (DSPA), pp. 300-306, Moscow, Russia, March 31 - April 2, 2004.

[61] Ndjiki-Nya P., Restat J., Meiers T., Ohm J.-R., Seyferth A., Sniehotta R., "*Subjective evaluation of the MPEG-7 retrieval accuracy measure (ANMRR)*", Doc. M6029, ISO/IEC JTC1/SC29/WG11, 2000.

[62] Huang J., Kumar S.R., Mitra M., Zhu W.-J., Zabih R., "*Image indexing using color correlograms*", Proceedings of Computer Vision and Pattern Recognition, pp. 762–768, 17–19 June 1997.

[63] Talib A., Mahmuddin M., Husni H. and George L.E., "*A weighted dominant color descriptor for content-based image retrieval*," Journal of Visual Communication and Image Representation, vol. 24 (3), pp. 345-360, 2013.

[64] Liu G-H., Yang J-Y., "*Content-based image retrieval using color difference histogram*," Pattern Recognition, vol. 46 (1), pp. 188-198, 2013.

[65] An J., Lee S.H. and Cho N.I., "*Content based image retrieval using color features of silent regions*", Proc. IEEE Int. Conf. Image Process (ICIP), pp. 3042-3046, 2014.

[66] Zou Q., Qi X., Li Q. and Wang S., "*Discriminative regional color co-occurrence descriptor*" Proc. IEEE Int. Conf. Image Process (ICIP), pp. 696-700, 2015.

[67] Fierro-Radilla A., Perez-Daniel K., Nakano-Miyatake M. and Benois J., "*Dominant Color Correlogram Descriptor for Content-Based Image Retrieval*," International Conference on Graphic and Image Processing (ICGIP), vol. 9443, 2014.

[68] Palmer S.E., Rock I., "Rethinking perceptual organization: The role of uniform connectedness", Psychonomic Bullettin & Review, 1(1), pp. 29–55, 1994.

[69] Achanta R., Shaji A., Smith K., Lucchi A., Fua P., Süsstrunk S., "*SLIC Superpixels Compared to State-of-the-art Superpixel Methods*", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 34, num. 11, p. 2274 - 2282, 2012.

[70] Ren, C. Y., Prisacariu, V. A., Reid, I. D., "*gSLICr: SLIC superpixels at over 250Hz*", arXiv preprint arXiv:1509.04232, 2015.

[71] Li, T., Xie, Z., Wu, J., Yan, J., Shen, L. "*Interactive object extraction by merging regions with k-global maximal similarity*," Neurocomputing, vol. 120, pp. 610–623, 2013.

[72] Calderero F., Marques F., "*Region merging techniques using information theory statistical measures*," IEEE Trans. Image Process., vol. 19(6), pp. 1567–1586, 2010.

[73] Liu H., Guo Q., Xu M., Shen I., "*Fast image segmentation using region merging with a k-nearest neighbor graph*," in Proc. IEEE Conf. Cybern. Intell. Syst., pp. 179–184, 2008.

[74] Peng B., hang L., Yang J., "*Iterated graph cuts for image segmentation*," in Proc. Asian Conf. Comput. Vis., pp. 677–686, 2009.

[75] Béréziat D., Herlin I., "*Solving ill-posed Image Processing problems using Data Assimilation*", Numerical Algorithms, vol. 2, no. 2, pp. 219–252, 2011.

[76] Boykov Y., Jolly M.-P., "*Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images*", IEEE Int. Conf. on Computer Vision (ICCV), vol. 1, pp. 105–112, 2001.

[77] Rother C., V. Kolmogorov V., Blake A., "*Grabcut: Interactive foreground extraction using iterated graph cuts*", in ACM SIGGRAPH, 2004.

[78] Liu J., Sun J., Shum H., "*Paint Selection*," In ACM SIGGRAPH, 2009.

[79] Sener O., Ugur K., Alatan A., "*Robust Interactive Segmentation via Coloring*," In ACM VIGTA, 2012.

[80] Liu J., Sun J., Shum H., "*Paint Selection*," In ACM SIGGRAPH, 2009.

[81] Ning, J., Zhang, L., Zhang, D., & Wu, C., "*Interactive image segmentation by maximal similarity based region merging*", Pattern Recognition, vol. 43(2), pp. 445–456, 2010.

[82] Biederman, I., "*Recognition-by-components: a theory of human image understanding*", Psychological Review, vol. 94(2), pp. 115–147, 1987.

[83] Lowe, D. G., "*Distinctive image features from scale-invariant keypoints*", International Journal of Computer Vision, vol. 60(2), pp. 91–110, 2004.

[84] Moravec. H.P., "*Rover visual obstacle avoidance*," In Proc. of the 7th Int. joint Conf. on Artificial intelligence (IJCAI), vol. 2, pp. 785–790, 1981.

[85] Lowe, D. G., "*Object recognition from local scale-invariant features*," In Int. Conf. on Computer Vision (ICCV), pp. 1150–1157, 1999.

[86] Bay H., Ess A., Tuytelaars T., Gool L.V., "*Speeded-up robust features (SURF)*," Computer Vision and Image Understanding, vol. 110 (3), pp. 346–359, 2008.

[87] Alcantarilla P. F., Bartoli A., Davison A. J., "*KAZE features*," In Eur. Conf. on Computer Vision (ECCV), pages 214–227, 2012.

[88] Rublee E., Rabaud V., Konolige K., Bradski G., "*ORB: An efficient alternative to SIFT or SURF*," Int. Conf. on Computer Vision (ICCV), pp. 2564–2571, 2011.

[89] Leutenegger S., Chli M., Siegwart R.Y. "*BRISK: Binary robust invariant scalable keypoints*," In IEEE Intl. Conf. on Computer Vision (ICCV), pages 2548–2555, 2011.

[90] Alcantarilla P. F., Nuevo J., & Bartoli A., "*Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces*," British Machine Vision Conference, pp.13.1–13.11, 2013.

[91] Li Y., Wang S., Tian Q., Ding X., "*A survey of recent advances in visual feature detection*," Neurocomputing, vol.149, pp.736–751, 2015.

[92] Edelman S., Intrator N. and Poggio T., "*Complex cells and object recognition*," Unpublished manuscript: http://kybele.psych.cornell.edu/~edelman/archive.html

[93] Lowe D.G., "*Towards a Computational Model for Object Recognition in IT Cortex*," In Proc. of the First IEEE Int. Workshop on Biologically Motivated Computer Vision (BMVC), vol. 1811, pp. 20–31, 2000.

[94] Calonder M., Lepetit V., Strecha C., "*Brief: Binary robust independent elementary features*," In Proc. Of European Conference on Computer Vision (ECCV), pp. 778–792, 2010.

[95] Heinly J., Dunn E., Frahm J.M., "*Comparative evaluation of binary features,*" In Proc. of European Conference on Pattern Recognition (ECCV), pp. 759–773, 2012.

[96] Rubner Y., Tomasi C., Guibas L. J., "*The Earth Mover's distance as a metric for image retrieval*," Technical Report STAN-CS-TN-98-86, Department of Computer Science, Stanford University, Sept. 1998.

[97] Muja M., Lowe D.G., "*Fast approximate nearest neighbors with automatic algorithm configuration*," VISAPP, vol. 2, pp. 331–340, 2009.

[98] Fischler M.A., Bolles R.C., "*Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*," Commun. ACM, vol. 24, no. 6, pp. 381–395, 1981.

[99] Chum O., Matas J., "*Matching with PROSAC – Progressive Sample Consensus*," In Proc. of Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 220–226 vol. 1, 2005.

[100] Treisman A., Gelade G., "*A feature-integration theory of attention*," Cognitive Psychology, vol(1), pp. 97–136, 1980.

[101] Healey C., Enns J., "*Attention and Visual Memory in Visualization and Computer Graphics*," In IEEE Trans. on Visualization and Computer Graphics, vol. 18(7), pp. 1170–1188, 2012.

[102] Li, J., "Photography Image Database," UCI KDD Archive, online: http://www.stat.psu.edu/~jiali/index.download.html

[103] Thompson R., "*Grammar of the Shot*," Focal Press,1998.

[104] Rensink R.A., O'Regan J.K., Clark J.J., "*To see or not to see: The need for attention to perceive changes in scenes,*" Psychological Science, vol 8, pp. 368–373, 1997.

[105] cvcl.mit.edu, "*Visual Intelligence: Predicting where people look*" [Online]. Available: http://cvcl.mit.edu/modeling_attentionCBdemo.html [accessed 31 October 2016]

[106] Mathis K.M., Kahan T., "*Holistic processing improves change detection but impairs change identification*," Psychonomic Bulletin & Review, vol. 21, pp.1250–12544, 2014.

[107] Shneiderman B., "*The eyes have it: A task by data type taxonomy for information visualizations*," In Proc. of IEEE Symposium on Visual Languages, pp.336–343, 1996.

[108] Craft B., Cairns P., "*Beyond Guidelines: What Can We Learn from the Visual Information Seeking Mantra?*," In Proc. of the 9th Int. Conf. on Information Visualization (IV '05), pp.110-118, 2005.

[109] Huang C.R., Lee H.P., Chen C.S., "*Shot change detection via local keypoint matching*," IEEE Trans. Multimedia, vol. 10(6), pp. 1097–1108, 2008.

[110] Smeaton A. F., Over P., Doherty A. R., "Video Shot Boundary Detection: Seven Years of TRECVid Activity," Comput. Vis. Image Underst., vol. 114(4), pp. 411–418, 2010.

Publication 1

S.Kiranyaz, M. Birinci, and M. Gabbouj, "Perceptual Color Descriptor Based on Spatial Distribution: A Top-Down Approach," Image and Vision Computing, vol. 28, pp. 1309-1326, 2010.

Image and Vision Computing 28 (2010) 1309–1326

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Perceptual color descriptor based on spatial distribution: A top-down approach

Serkan Kiranyaz *, Murat Birinci, Moncef Gabbouj

Tampere University of Technology, Department of Signal Processing, P.O. Box 553, 33101 Tampere, Finland

ARTICLE INFO

Article history:
Received 20 July 2008
Received in revised form 26 January 2010
Accepted 27 January 2010

Keywords:
Perceptual color descriptor
Human visual system
Content-based image indexing and retrieval
Spatial color distribution

ABSTRACT

Color features are the key-elements widely used in content-analysis and retrieval. However, most of them show severe limitations and drawbacks due to their inefficiency of modeling the human visual system with respect to color perception. Moreover, they cannot characterize all the properties of the color composition in a visual scenery. In this paper we present a perceptual color feature, which describes all major properties of prominent colors both in spatial and color domains. In accordance with the well-known *Gestalt* law, we adopt a global, top-down approach in order to model (see) the whole color composition before its parts and in this way we can avoid the problems of pixel-based approaches. In color domain the dominant colors are extracted along with their global properties and quad-tree decomposition partitions the image so as to characterize the spatial color distribution (SCD). We propose two efficient SCD descriptors; the proximity histograms, which distill the histogram of inter-color distances and the proximity grids, which cumulate the spatial co-occurrence of colors in a 2D grid. Both approaches are configurable and provide means of modeling SCD in a scalar and directional way. Combination of the extracted global and spatial properties forms the final descriptor, which is unbiased and robust to non-perceivable color elements in both spatial and color domains. Finally a penalty-trio model fuses all color properties in a similarity distance computation during retrieval. Experimental results approve the superiority of the proposed technique against powerful global and spatial color descriptors.

1. Introduction

The color composition of an image can turn out to be a powerful feature for the purpose of content-based image retrieval (CBIR), if extracted in a perceptually oriented way and kept semantically intact. Furthermore, color structure in a visual scenery is robust to noise, image degradations, changes in size, resolution and orientation. Eventually most of the existing CBIR systems use various color descriptors in order to retrieve relevant images (or visual multimedia material); however, their retrieval performance is usually limited especially on large databases due to lack of discrimination power of such color descriptors. One of the main reasons for this is because most of them are designed based on some heuristics or naïve rules that are not formed with respect to what humans or more specifically the human visual system (HVS) finds "relevant" in color similarity. The word "relevance" is described as "the ability (as of an information retrieval system) to retrieve material that satisfies the needs of the user". Therefore, it is of decisive importance that human color perception is respected whilst modeling and describing any color composition of an image. In other words, if and only when a particular color descriptor is designed based entirely on HVS and human color perception rules, further discrimination power and hence certain improvements in the retrieval performance can be achieved.

Accordingly, the study of human color perception and similarity measurement in the color domain become crucial and there is a wealth of research performed in this field. For example in [3], van den Broek et al. focused on the utilization of color categorization (called as *focal* colors) for CBIR purposes and introduced a new color matching method, which takes human cognitive capabilities into account. They have exploited the fact that humans tend to think and perceive colors only in 11 basic categories. In [23], Mojsilovic et al. performed a series of psychophysical experiments analyzing how humans perceive and measure similarity in the domain of color patterns. Their experiments concluded five perceptual criteria (called "basic color vocabulary") which are important for comparing the color patterns as well as a set of rules (called "basic color grammar") which are governing the use of these criteria in similarity judgment. One observation worth mentioning here is that the human eye cannot perceive a large number of colors at the same time, nor it is able to distinguish similar (close) colors well. Based on this, they showed that at the coarsest level of judgment, HVS primarily uses *dominant colors* (i.e. the few prominent colors in the scenery) to judge similarity. Henceforth, the two rules are particularly related for modeling the similarity metrics of the human color perception. The first one indicates that the two color patterns that have similar dominant colors (DCs) are perceived as

* Corresponding author. Tel.: +35 850432 4123; fax: +35 8033115 4989.
E-mail address: serkan.kiranyaz@tut.fi (S. Kiranyaz).

0262-8856/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2010.01.012

similar. The second rule states that two multicolored patterns are perceived as similar if they possess the same (dominant) color distributions regardless of their content, directionality, placement or repetitions of a structural element.

In short, humans focus on a few DCs and their (spatial) distributions while judging the color similarity between images and our ability to extract such a global color view out of a visual scenery, irrespective of its form, be it a digital image or a natural 3D view is indeed amazing. However, it is not that straightforward to accomplish this while dealing with digital images for CBIR purpose. Note that on a standard 24 bit representation, there is a span of 16 million colors, which can be assigned on thousands of individual pixels. Such a "high resolution" representation might be required for current digital image technologies; however, it is not too convenient for the purpose of describing color composition or performing a similarity measurement based on the aforementioned human color perceptual rules. Nevertheless, it is obvious that humans can neither see individual pixels, nor perceive even a tiny fracture of such a massive amount of color levels and thus it is crucial to perform certain steps in order to extract the true "perceivable" elements (the true DCs and their global distributions). In other words the un-perceivable elements (we call them *outliers*), which do not have significant contribution or weight over the present color structure, in both color and spatial (pixel) domain, should be suppressed or removed. Recall that according to two color perception rules presented in [24], two images that are perceived as similar in terms of color composition have similar DC properties; however, the color properties of their outliers might be entirely different and hence this can affect (degrade, bias or shift) any similarity measurement if not handled accordingly. For example in the well-known perceptual audio coding schemes such as MP3 and AAC [2], in order to maximize the *coding* efficiency such outliers (the sound elements that humans cannot hear) in both spatial (time) and spectral (frequency) domains are removed and thus more bits can be spent for the "dominant" sound elements. In a similar fashion, the outliers both in color and spatial domain should be removed for *description* efficiency. Henceforth in this paper, we present a systematic approach to extract such a perceptual (color) descriptor and then propose an efficient similarity metric to achieve the highest discrimination power possible for color-based retrieval in general-purpose image databases.

In order to remove outliers and to secure the global (perceptual) color properties, one alternative is to apply non-linear filters (e.g. median or Bilateral [42]). However, there would be no guaranty that such a filter will remove all or the majority of the outliers and yet several filter parameters are needed to be set appropriately for an acceptable performance, which is not straightforward to do so especially for large databases. Instead, we adopt a top-down approach both in DC extraction and modeling their global spatial distribution. This approach is in fact phased from the well-known *Gestalt* rule of perception [48]: "Humans see the whole before its parts", therefore, the method strives to extract what is the (next) global element both in color and spatial domain, which are nothing but the DCs and their spatial distribution within the image. In order to achieve such a (global) spatial representation within an image, starting from the entire image, quad-tree decomposition is applied to the current (parent) block only if it cannot host the majority of a particular DC; otherwise, it is kept intact (non-decomposed) representing a single, homogeneous DC presence in it. So this approach tries to capture the "whole" before going through "its parts" and whenever the whole body can be perceived with a single DC, it is kept "as is". Hence outliers can be suppressed from the spatial distribution and furthermore, the resultant (block-wise) partitioned scheme can be efficiently used for a global modeling and due description of the spatial distribution. Finally a penalty-trio model uses both global and spatial color properties and performs an efficient similarity metric. After the image is (quad-tree) decomposed, we then represent this global spatial distribution via inter-proximity statistics of the DCs, both in *scalar* and *directional* modes. These modes of spatial color distribution (SCD) can both describe the distribution of a particular DC with itself (auto SCD) and with other DCs (inter SCDs).

The proposed method is fully automatic. Forming the whole process as a Feature eXtraction (*FeX*) module into MUVIS framework [19], allows us to test the mutual performance in the context of multimedia indexing and retrieval. The rest of the paper is organized as follows. Before going into the details of the proposed approach, Section 2 presents the related studies in the area of color based CBIR, stressing particularly their limitations and drawbacks under the light of the earlier discussion on human color perception. In Section 3 we introduce a generic overview of the proposed color descriptor together with the extraction, formation of the feature vector and calculation of the similarity distances. Section 4 presents the retrieval results of the proposed color descriptor. Section 5 concludes the paper and suggests topics for future research.

2. Related work

There is a wealth of research done and still going on in developing content-based multimedia indexing and retrieval systems such as MUVIS [19], QBIC [10], PicHunter [6], Photobook [32], VisualSEEk [38], Virage [46], Image-Rover [36], VideoQ [4], etc. In such frameworks, database primitives are mapped into some high dimensional feature domain, which may consist of several types of descriptors such as visual, aural, etc. From the latitude of low-level descriptors, careful selection of some sets to be used for a particular application may capture the semantics of the database items in a content-based multimedia retrieval (CBMR) system. Although color is used in many areas such as object and scene recognition [35], in this article we shall restrict the focus on CBIR domain, which employ only *color* as the descriptor for image retrieval.

2.1. Global color descriptors

In one of the earlier works on color descriptors, Kato et al. [12] used the color of every corresponding pixel in two images for comparison and the number of corresponding pixels having the same color determines the similarity between them. Recall the HVS fact mentioned earlier about humans inability to see individual pixels or to perceive large amount of color levels and hence this approach did not provide robust solutions, i.e. slight changes in camera position, orientation, noise or lightning conditions may cause significant degradations in the similarity computation. Swain and Ballard [41] proposed the first color histogram, which solves this sensitivity problem. In their work color histograms are extracted and histogram intersection method is utilized for comparing two images. Since this method is quite simple to implement and gives reasonable results especially in small to medium size databases, several other histogram-based approaches emerged, such as [6,9,10,19,26,32,36,37,44,47] and [49]. MPEG-7 Color Structure Descriptor (CSD) [22], is also based on color histogram, but provides a more accurate color description by identifying localized color distributions of each color. Unlike the conventional color histograms, CSD is extracted by accumulating from a 8 × 8 structuring window. The image is scanned and CSD counts the number of times a particular color is contained within the structuring window. A good review and an efficient representation of color histograms based on Karhunen–Loeve transform (KLT) can be found in [43]. The primary feature of such histogram-based color descriptors (be it in RGB, CIE-Lab, CIE-Luv, or HSV) is that they cluster the pixels into fixed color bins, which are quantizing the entire color space using a pre-defined color palette. This two-fold approach, clustering all the pixels having similar color and reducing the color levels from millions to (usually) thousands or even hundreds via quantization, is the main reason behind the limited success that the color histograms achieved since both operations are indeed the small steps through obtaining the perceivable elements (the true DCs and their global distributions); yet their performance is still quite limited and usually degrades drastically in large databases due to several reasons. First and the foremost, they apply static-quantization, where the color palette boundaries are determined empirically or via some heuristics–yet nothing based on human color perception rules. If, for example, the number of bins are set too high (fine quantization) then similar color pairs will end up in different bins. This will eventually cause erroneous similarity computation whenever using any of the naïve metrics such as L_1, L_2 or using the histogram intersection method as in [41]. On the other hand if the number of bins is set too low (coarse quantization) then there is an imminent danger of completely different colors falling into the same bin and this will obviously degrade the similarity computation and reduce the discrimination power. No matter how the quantization level (number of bins) is set, pixels with such similar colors but happens to be opposite sides of the quantization boundary, separating two consecutive bins will be clustered into different bins and this is an inevitable source of error in all histogram-based methods. The color quadratic distance [10] proposed in the context of QBIC system provides a solution to this problem by fusing the color bin distances into the total similarity metric. Let X and Y be two color histograms with total number of N bins and if we write them as pairs of color bins and weight: $X = \{(c_1, w_1^X), (c_2, w_2^X), \ldots, (c_N, w_N^X)\}$ and $Y = \{(c_1, w_1^Y), (c_2, w_2^Y), \ldots, (c_N, w_N^Y)\}$ then the quadratic distance between X and Y is as follows:

$$D_Q(X,Y)^2 = (X-Y)^T A (X-Y) = \sum_i^N \sum_j^N (w_i^X - w_i^Y)(w_j^X - w_j^Y) a_{ij} \tag{1}$$

where $A = [a_{ij}]$ is the matrix of color similarities between the bins c_i and c_j. This formulation allows the comparison of different histogram bins with some inter-similarity between them; however, it underestimates distances because it tends to accentuate the color similarity [39]. Furthermore, Po and Wong in a recent study [33] showed that the quadratic distance formulation has serious limitations: it does not match the human color perception well enough and may result in incorrect ranks between regions with similar salient color distributions. Hence, it gives even worse results than the naïve L_p metrics in some particular cases.

Besides the aforementioned clustering drawbacks and the resultant erroneous similarity computation, color histograms have computational deficiencies due to the hundreds (or even thousands) of redundant bins created for each image in a database, although ordinary images usually contain few DCs (i.e. <8), and more than that cannot anyway be perceived by HVS [24] according to the second color perception rule mentioned earlier. Therefore, color histograms do not only create a major computational deficiency in terms of storage, memory limit and computation (CPU) time due to spending hundreds or thousands of bins for the few DCs present, moreover their similarity computations will be biased by the *outliers* hosted within those redundant bins. Recall that two images with similar color composition will have similar DC properties; however, there is no such requirement for the outliers as they can be entirely different. Hence including color outliers into similarity computation may cause misinterpreting two similar images as dissimilar or vice versa and usually reduce the discrimination power of histogram-based descriptors, which eventually makes them unreliable especially in larger databases.

In order to solve the problems of static-quantization in color histograms, various DC descriptors, e.g. [1,7,8,22,24,49] have been developed using dynamic-quantization with respect to image color content. DCs, if extracted properly according to the aforementioned color perception rules, can indeed represent the prominent colors in any image. They have a global representation, which is compact and accurate and they are also computationally efficient. We implement a top-down DC extraction scheme, similar to the one in [7], where the method is entirely designed with respect to HVS color perceptual rules. For instance, HVS is more sensitive to the changes in smooth regions than in detailed regions. Thus in this work colors are quantized more coarsely in the detailed regions while smooth regions have more importance. To exploit this fact, a smoothness weight ($w(p)$) is assigned to each pixel (p) based on the variance in a local window. Afterwards, the *General Lloyd Algorithm* (*GLA*, also referred to as *Linde–Buzo–Gray* and it is equivalent to the well-known *K-means* clustering method [21]) is used for color quantization.

2.2. Spatial color descriptors

Although the true DCs, which are extracted via such perceptually oriented scheme with the proper metric can address the aforementioned problems of color histograms, global color properties (DCs and their coverage areas) alone are not enough for characterizing and describing the real color composition of an image since they all lack the crucial information of spatial relationship among the colors. In other words, describing "what" and "how much" color is used will not be sufficient without specifying "where" and "how" the (perceivable) color components (DCs) are distributed within the visual scenery. For example all the patterns shown in Fig. 1 have the same color proportions (be it described via DCs or color histograms), but different spatial distributions and thus cannot be perceived as the same. Especially in large image databases, this is the main source of erroneous retrievals, which makes "accidental" matches between images with "similar" global color properties but different in the color distribution.

There are several approaches to address such drawbacks. Segmentation-based methods may be an alternative; however, they

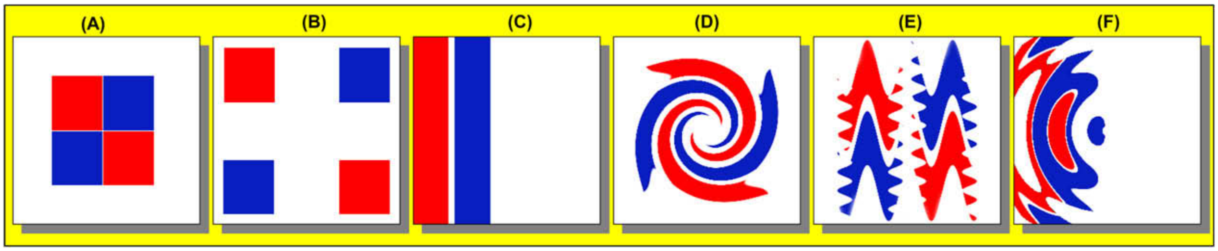


Fig. 1. Different color compositions of red, blue and white with the same proportions (weights).

are not feasible since in most cases automatic segmentation is an ill-posed problem, therefore, it is not reliable and robust for applications on large databases. For example in a recent work [40] DCs are associated with the segmented regions but the method can only be applied to a small size (i.e. 200) *National Flags* database, where segmentation is trivial. Some studies used the local positions of color blocks for characterizing the spatial distributions. For instance in an earlier study, Gong et al. [9] divided the image into nine equal sub-images and represented each of them by a color histogram. In a similar work, Stricker and Dimai [39] split the image into five regions: an oval central region and four corners. They tried to combine color similarity from each region whilst attributing more weight to the central region. A similar approach is proposed by Valova and Rachev in [45] where they split image into 16×16 blocks and each block is represented by a unique dominant color. Due to the fixed partitioning, such methods become strictly domain dependant solutions. Ooi et al. [28] enhances the idea of using a statistically derived quad-tree decomposition to obtain homogeneous blocks but again comparing the matching blocks (in the same position) to obtain SCD similarity. Basically in such approaches the local position of a certain color in an image cannot really describe the true SCD due to several reasons. First the image dimensions, resolution and their aspect ratio can vary significantly. So an object with a certain size can fall (perhaps partially) into different blocks in different locations. Furthermore, such a scheme is not rotation and translation invariant. Pass et al. [31] presented Color Coherence Vector (CCV), which partitions the histogram bins based on the spatial coherence of the pixels. A given pixel is "coherent" if its color is similar color to a colored-region and "incoherent" otherwise. For each color c_i, let $\alpha(c_i)$ and $\beta(c_i)$ be the number of coherent and incoherent pixels, thus the pair $(\alpha(c_i), \beta(c_i))$ is called a coherence pair for the ith color, and the coherent vector can be defined for an image I as:

$$CCV(I) = \{(\alpha(c_1), \beta(c_1)), (\alpha(c_2), \beta(c_2)), \ldots, (\alpha(c_N), \beta(c_N))\} \tag{2}$$

L_1 metric is used to compare two images. A nice property of this method is the classification of the outlier (color) pixels in spatial domain (i.e. incoherent) from the prominent (i.e. coherent) ones. They report a better retrieval performance than traditional histogram-based methods. Yet, apart from the aforementioned drawbacks of histogram-based methods with respect to individual pixels, classifying color pixels alone, without any metric or characterization for the SCD will not describe the real color composition of an image. Another variation of this approach is characterizing adjacent color pairs, i.e. color boundaries. Nagasaka and Tanaka [25] developed a color matching technique to model color boundaries. Thus, two images are expected to be similar if they have similar sets of color pairs. In a similar approach, Stricker [39] used the boundary histograms to describe the length of the color boundaries. Another color adjacency based descriptor can be found in [14]. Such a heuristic approach of using color adjacency information might be more intuitive than the ones using fixed blocks, since they at least used "relative" features instead of "static" ones. Yet the approach is likely to suffer from changes in background color or relative translations of the objects in an image. The former case implies to a strong dissimilarity although only the background color is changed whilst the rest of the object(s) or color elements stay intact. In the latter case there is no change in the adjacent colors, however, the inter-proximities of the color elements (hence the entire color composition) are changing and hence a certain dissimilarity should occur. Therefore, the true characterization of SCD lies in the inter-proximities (the relative distances) of color elements with respect to each other. In other words, characterizing inter- or self-color proximities (e.g. the relative distances of the DCs) shall be a reliable and discriminative cue about the color composition. This property is invariant to translations, rotations and variations in image properties (dimensions, aspect ratio and resolution) and hence will be the basis of the proposed descriptor for spatial color description.

2.3. The color correlogram

One of the most promising approaches among all SCD descriptors is the color Correlogram [11,13], which is a table, where the kth entry for the color histogram bin pair (i,j) specifies the probability of finding a pixel of color bin j at a distance k from a pixel of color bin i in the image. Recently a similar technique, the color edge co-occurrence histogram, has also been used for color object detection in [17]. Let I be an $W\,x\,H$ image quantized with m colors $(c_1, \ldots, c_i, \ldots, c_m)$ via RGB color histogram. For a pixel $p = (x, y) \in I$, let $I(p)$ denotes its color value and let $I\langle c_i\rangle \equiv \{p | I(p) = c_i\}$. So the color histogram value of a quantized color c_i, $h(c_i, I)$, can be defined as:

$$h(c_i, I) = WH\Pr(p \in I\langle c_i\rangle) \tag{3}$$

Accordingly, the color Correlogram $\gamma^{(k)}_{c_i,c_j}$, for the quantized color pair (c_i, c_j) and a pixel distance $k \leqslant d$ can be expressed as:

$$\gamma^{(k)}_{c_i,c_j} = \Pr_{p_1 \in I\langle c_i\rangle, p_2 \in I}(p_2 \in I\langle c_j\rangle \| p_1 - p_2| = k) \tag{4}$$

where $c_i, c_j \in \{c_1, \ldots, c_m\}, k \in \{1, \ldots, d\}$ and $|p_1 - p_2|$ is the distance between pixels p_1 and p_2 in L_∞ norm. Since the feature vector size of Correlogram is $O(m^2 d)$, a simplified version, the so-called Auto-Correlogram, which only captures the spatial correlation between the same colors and thus reduces the feature vector size to $O(md)$ bytes, was proposed in [11]. A variant of the Correlogram based on HSV color domain is proposed in [27].

In the spatial domain and pixel level, Correlogram can characterize and thus describe the relative distances of distinct colors between each other and thus such a description can indeed reveal a high resolution model of SCD. Accordingly, Ma and Zhang in [20], and recently Chun et al. in [5] conducted comprehensive performance evaluations among several global/spatial color descriptors for CBIR and reported that (Auto-) Correlogram achieves the best retrieval performance among the others, such as color histograms, CCV, color moments, etc. In another recent work, Li et al. [16] proposed Markov Stationary Features (MSFs), which is an extension of the color auto-correlogram and compared it with the auto-correlogram and other MSF based CBIR features such as color histograms, CCVs, texture and edge. Among all color descriptors, Auto-Correlogram (extended by MSF) performs the best but only slightly better than the Auto-Correlogram. Another extension is the Wavelet Correlograms proposed by Lee et al. in [15] and it performs slightly better than the Correlogram and surpasses other color descriptors such as color histograms and scalable color descriptor. Moghaddam and Saadatmand-Tarzjan in [18] proposed another approach, called Gabor wavelet Correlogram for image indexing and retrieval and further improved the retrieval performance. Therefore, in this work we shall make comparative evaluations of the proposed technique against the color Correlegram whenever applicable, because, it suffers from a serious computational complexity and a massive memory requirement problems. Nowadays digital image technology offers several mega-pixel (Mpel) image resolutions. For a conservative assumption, consider a small size database with only 1000 images each of which in only 1 Mpel resolution. Without any loss of generality, assume that $W = H = 1000$. In such image dimensions, a reasonable setting for d would be $100 < d < 500$, corresponding to ~10%–50% image dimension range. Any d setting <100 pixels would be too small for characterizing the true SCD of the image –probably describing only a thin layer of adjacent colors (i.e. colors that can be found within a small range). Assume the

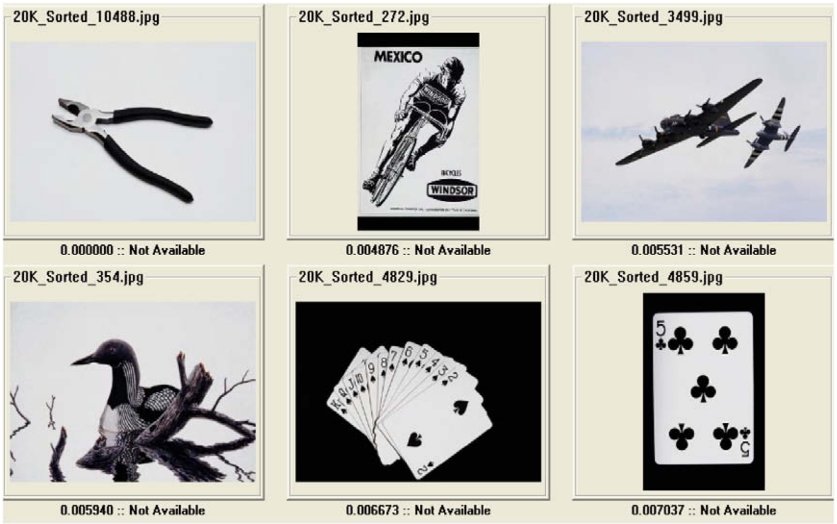

20K_Sorted_10488.jpg
0.000000 :: Not Available
20K_Sorted_272.jpg
0.004876 :: Not Available
20K_Sorted_3499.jpg
0.005531 :: Not Available
20K_Sorted_354.jpg
0.005940 :: Not Available
20K_Sorted_4829.jpg
0.006673 :: Not Available
20K_Sorted_4859.jpg
0.007037 :: Not Available

Fig. 2. Top six ranks of Correlogram retrieval in a 20,000-image database. Top-left is the query image.

lowest range setting: $d = 100$ (yet a Correlogram working over only a 10% range of the image dimension is hardly a "spatial" color descriptor). Even with such "minimal" settings, the naïve algorithm will require $\sim O(10^{10})$ computations (including divisions, multiplications and additions). Even with fast computers, this will require several hours of computation per image and infeasible time is required to index even the smallest databases. In order to achieve a feasible computational complexity for the naïve algorithm, the range has to be reduced drastically (i.e. $d \sim 10$) and the images should be decimated by 3–5 times in each dimension. Such a solution unfortunately changes (decimates) the color composition of the scheme and with such limited range, the true SCD cannot anymore be characterized. The other alternative is to use the fast algorithm. A typical quantization for RGB color histogram can be eight partitions in each color dimension (i.e. $8 \times 8 \times 8 = 512$ bins RGB histogram), the fast algorithm will speed up the process around 25 times; however, it will also require a massive *memory* space, (>400 Gb per image) and this time neither decimation, nor drastic reduction on the range will make it feasible and practically speaking, one can hardly make it work only for thumbnail size images and only when $d < 10$ and much coarser quantization (e.g. using $4 \times 4 \times 4$ RGB histogram) is used. Furthermore, its massive storage requirement is another serious bottleneck of the Correlogram. Note that for the minimal range ($d = 100$) and typical quantization settings (i.e. $8 \times 8 \times 8$ RGB partitions), the amount of space required for the feature vector storage of a single image is above 400 Mb. This allows the Correlogram barely applicable only for small size databases, i.e. for 1000 image database the storage space required is above 400 Gb. To make it work, the range value has to be reduced drastically along with using a much coarser quantization ($4 \times 4 \times 4$ bins or less). Unfortunately with such settings, recall the problems of coarse quantization of color histograms and such a diminished range setting. The only alternative is to use Auto-Correlogram instead of Correlogram, which is eventually recommended and used in [11]; however, without characterizing spatial distribution of distinct colors with respect to each other, the performance of the color descriptor may be degraded.

Apart from all such feasibility problems, Correlogram may exhibit several limitations and drawbacks. The first and the foremost is its pixel-based structure, which characterizes the color proximities at a pixel level. Such a high resolution description not only makes it too complicated and infeasible to perform, it also becomes meaningless with respect to HVS color perception rules simply because individual pixels do not mean much for the human eye. As an example, consider a Correlogram description such as "the probability of finding a *red* pixel within a 43 pixel proximity of a *blue* pixel is 0.25" and so what difference does it make to have this probability in 44 or 42 pixels proximity for the human perception? Another similar image might have the same probability but in 42 pixels proximity, which makes it indifferent or even identical for the human eye; however, a significant dissimilarity will occur via Correlogram's naïve (dis-)similarity computation. Furthermore, since Correlogram is a pixel level descriptor working over RGB color histogram, the *outliers*, both in color and spatial domains have an imminent affect both over computational complexity and the retrieval performance of the descriptor. Hundreds of color outliers hosted in the histogram, even though not visible to the human eye, will cause computational (memory, storage and speed) problems, making the Correlogram inapplicable in many cases. Yet the real problem lies in degradation caused by the *outliers* directly over the description accuracy such as their bias (shift) over the true (perceivable) probabilities (inter-color proximities). Finally, using the probability alone makes the descriptor insensitive to the dominance of a color or its area (weight) in the image. This is basically due to the normalization by the amount (weight or area) of color, $h(c_i, I)$, and such an important perceptual cue is lacking in the Correlogram's description. This might be a desirable property to find the similar images simply "zoomed" as in [11], and hence the color areas significantly vary but the distance probabilities do not. However, it may also cause severe mismatches especially in large databases since the probability of the pair-wise color distances might be the same or close independent of their *area* and hence regardless of their dominance (whether they are DCs or outliers). An example of such a descriptor deficiency can be seen in a query of the sample image shown in Fig. 2. In short, these properties make Correlogram more of a colored *texture* descriptor rather than a *color* descriptor since its pixel level, area insensitive, co-occurrence description is quite similar to texture descriptors based on co-occurrence statistics (e.g. Gray-Level Co-occurrence Matrix (GLCM) [29]) only with a major difference of describing *color* co-occurrences instead of gray level (intensity) values.

3. The proposed color descriptor

Under the light of the earlier discussion, the proposed color descriptor is designed to address the drawbacks and problems of the color descriptors, particularly the color Correlogram. In order to achieve this, it is mainly motivated by the human color perception rules and therefore, global and spatial color properties are extracted and described in a way HVS perceives them. Therefore, *outliers*, in color and spatial domains, are suppressed or eliminated by adopting a top-down approach during feature extraction. The proposed color descriptor is formed by a proper combination of global and spatial color features. During the retrieval phase, the (dis-)similarity between two images is computed using a penalty-trio model, which penalizes the individual differences in global and spatial color properties. In the following sub-sections, we will detail both indexing (feature extraction) and retrieval schemes.

3.1. Formation of the color descriptor

As explained in Section 2.1, the DCs represent the prominent colors in an image whilst the unperceivable color components (outliers) are discarded. As a result, they have a global representation, which is compact and accurate, and they represent the few (dominant) colors that are present and perceivable in an image. For a color cluster C_i, its centroid c_i is calculated by

$$c_i = \frac{\sum w(p)x(p)}{\sum w(p)}, \quad x(p) \in C_i \tag{5}$$

and the initial clusters are determined by using a weighted distortion measure, defined as,

$$D_i = \sum w(p)\|x(p) - c_i\|^2, \quad x(p) \in C_i \tag{6}$$

This is used to determine which clusters to split until either a maximum number of clusters (DCs), $N_{DC}^{\max}$, is achieved or a maximum allowed distortion criteria, ε_D, is met. Hence, pixels with smaller weights (detailed sections) are assigned fewer clusters so that the number of color clusters in the detailed regions, where the likelihood of outliers' presence is high, is therefore suppressed. As the final step, an agglomerative clustering (AC) is performed on the cluster centroids to further merge similar color clusters so that there is only one cluster (DC) hosting all similar color components in the image. A similarity threshold T_S is assigned to the maximum color distance possible between two similar colors in a certain color domain (CIE-Luv, CIE_Lab, etc.). Another merging criterion is the color area, that is, any cluster should have a minimum amount of coverage area, T_A, so as to be assigned as a DC; otherwise, it will be merged with the closest color cluster since it is just an outlier. Another important issue is the choice of the color space since a proper color clustering scheme for DC extraction tightly relies on the metric. Therefore, a perceptually uniform color space should be used and the most common ones are CIE-Luv and CIE-Lab, which are designed such that color distances perceived by HVS are also equal in L_2 (Euclidean) distance in these spaces. HSV space, although an intuitive color domain, suffers from discontinuities and RGB color space is not perceptually uniform. Therefore, among CIE-Luv and CIE-Lab, we select the former since it yields a lower transformation cost from native RGB space. For CIE-Luv, a typical value for T_S is between 10 and 20, T_A is between 2% and 5% [22] and $\varepsilon_D < 0.05$. Based on the earlier remarks, $N_{DC}^{\max}$ can be conveniently set to 8. As shown in Fig. 3, the DC extraction method used is similar to the one in [7], where it is entirely designed with respect to HVS color perceptual rules and configurable with few thresholds, T_S (color similarity), T_A (minimum area), ε_D (minimum distortion) and $N_{DC}^{\max}$ (maximum number of DCs). As the first step, the true number of DCs present in the image (i.e. $1 \leqslant N_{DC} \leqslant N_{DC}^{\max}$) is extracted in CIE-Luv color domain and back-projected to the image for further analysis involving extraction of the spatial properties (SCD) of DCs. Let C_i represents the ith DC class (cluster) with the following members: c_i is the color value (centroid), w_i is the weight (unit normalized area) and σ_i is the standard deviation obtained from the distribution of (real) colors clustered by C_i. Due to the DC thresholds set beforehand, $w_i > T_A,\ |c_i - c_j| > T_S$ *for* $1 \leqslant i, j \leqslant N_{DC}$.

During the back-projection phase, the DC, which has the closest centroid value to a particular pixel color, will be assigned to that pixel. As a natural consequence of this process, spatial outliers, i.e. isolated pixel(s), which are not populated enough to be perceivable, can emerge (e.g. see the example in Fig. 3) and should thus be eliminated. Due to the perceptual approach based on the Gestalt rule, "Humans see the whole before its parts", a top-down approach such as quad-tree decomposition can process the "whole" first, meaning the largest blocks possible, which can be described (and perceived) by a single DC, before going into its "parts". Due to its top-down structure, the proposed scheme does not suffer from the aforementioned problems of some pixel-based approaches.

Two parameters are used to configure the quad-tree: T_W, which is the minimum weight (dominance) within the current block

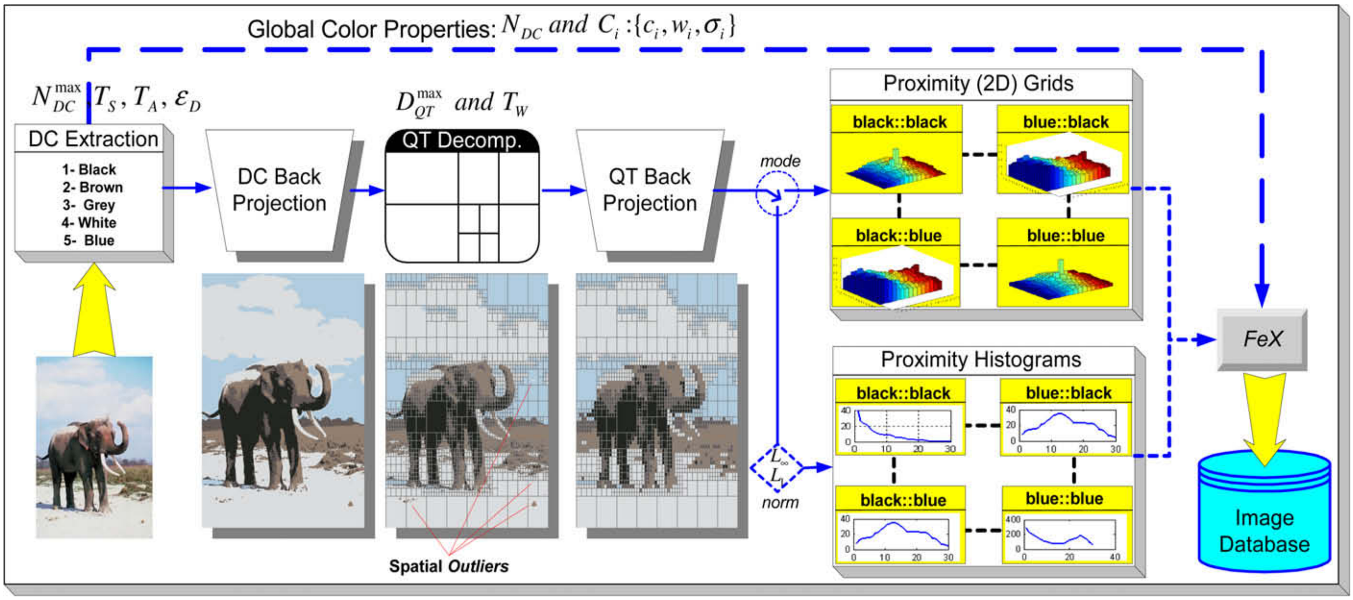


Fig. 3. Overview of the proposed color descriptor formation.

required from a DC not to go down for further partition and $D_{QT}^{\max}$, which is the depth limit indicating the maximum amount of partition (decomposition) allowed. Note that with the proper setting of T_W and $D_{QT}^{\max}$, QT decomposition can be carried out to reach the pixel level; however, such an extreme partitioning should not be permitted to avoid the aforementioned problems of pixel level analysis. Using a similar analogy T_W can be set in accordance with T_A, i.e. $T_W \cong 1 - T_A$. Therefore, for the typical T_A setting (between 2% and 5%), T_W can be conveniently set as $T_W \geqslant 95\%$. Since $D_{QT}^{\max}$ determines when to stop the partitioning abruptly, it should not be set too low so that it does not cause inhomogeneous (mixed) blocks and on the other hand, extensive experimental results suggest that $D_{QT}^{\max} > 6$ is not required even for the most complex scenes since the results are almost identical to the one with $D_{QT}^{\max} = 6$. Therefore, the typical range is $4 \leqslant D_{QT}^{\max} \leqslant 6$. Let B^p corresponds to the pth partition of the block B, where $p = 0$ is the entire block and $1 \leqslant p \leqslant 4$ represents the pth quadrant of the block. The four quadrants can be obtained simply by applying equal partitioning to the parent block or via any other partitioning scheme, which can be optimized to yield most homogenous blocks possible. For simplicity we use the former case and accordingly a generic QT algorithm, QuadTree, can be expressed as follows:

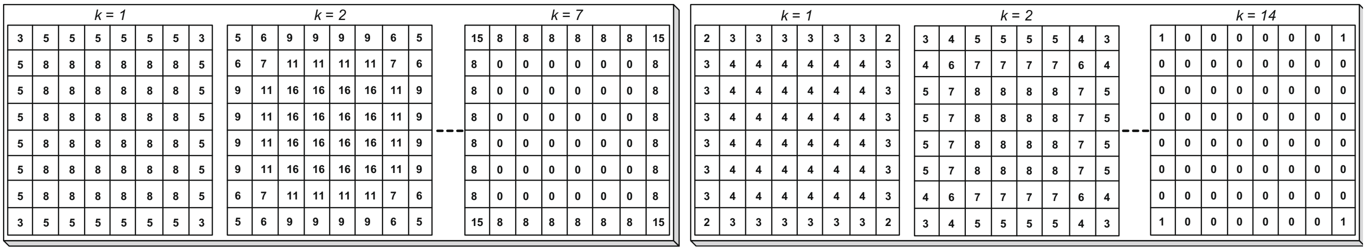
```
QuadTree (parent, depth)

• If depth = $D_{QT}^{\max}$ then Return.
• Let $W_{\max}$ be the weight of the DC, which has the maximum
  coverage in parent block.
• If $(W_{\max} > T_W)$ then Return.
• Let $\boldsymbol{B}^0 = Parent$.
• For $\forall p \in [1, \ldots, 4]$ do:
    ○ QuadTree ($\boldsymbol{B}^p$, depth+1)
• Return.
```

The QT decomposition of a (back-projected) image I can then be initiated by calling ***QuadTree*** (I, 0) and once the process is over, each QT block carries the following data: its depth $D \leqslant D_{QT}^{\max}$, where the partitioning is stopped, its location in the image and the major DC, which has the highest weight in the block (i.e. $w_{\max} > T_W$) and perhaps some other DCs, which are eventually some spatial outliers. In order to remove those spatial outliers, a QT back-projection of the major DC into its host block is sufficient. Fig. 3 illustrates the removal of some spatial outliers via QT back-projection on a sample image. The final scheme, where outliers in both color and spatial domains are removed and the (major) DCs are assigned (back-projected) to their blocks, can be conveniently used for further (SCD) analysis to extract spatial color features. Note that QT blocks can vary in size depending on the depth, yet even the smallest (highest depth) block is large enough to be perceivable and carry a homogenous DC. So instead of performing pixel level analysis such as in Correlogram, the uniform grid of blocks in the highest depth ($D = D_{QT}^{\max}$) can be used for characterizing the global SCD and extracting the spatial features in an efficient way. As shown in Fig. 3, one of the two modes, which perform two different approaches to extract spatial color features can be used. The first is the *scalar* mode, over which inter-DC proximity histograms are computed within the full image range. These histograms indicate the amount of a particular DC that can be found from a certain distance of another DC; however, this is a scalar measure, where the direction information is lacking. For example, such a measure can state "17% of red is 8 units (blocks) away from blue" but without any directional information. Therefore, the second mode is designed to represent inter-occurrence of one DC with respect to another over a 2D (proximity) grid from which both distance and direction information can be obtained. Note that inter-color distances are crucial for characterizing the SCD of an image; however, the direction information may or may not be useful depending on the content. For example, the direction information in "17% of red is 8 units (blocks) *right* of blue" is important for describing a national flag (and hence the content) but "One black and one white horse are running together on a green field" is sufficient to describe the content without any need to know the exact directional order of black, white and green. In the following sub-sections we will first detail both modes and then evaluate their computational and retrieval performances individually.

3.1.1. SCD description via proximity histograms

Once the QT back-projection of major DCs into their host blocks are completed, all QT blocks hosting a single (major) DC with a certain depth ($D \leqslant D_{QT}^{\max}$) are further partitioned into the blocks in highest depth (i.e. $D = D_{QT}^{\max}$) so as to achieve a proximity histogram in the highest block-wise resolution. Therefore, in such a uniform block-grid, the image I will have $N \times N$ blocks, where $N = 2^{D_{QT}^{\max}}$, each of which hosts a single DC. Accordingly the problem of computing inter-DC proximities turns out to be block distances and hence the block indices in each dimension (i.e. $\forall x, y \in [1, N]$) can directly be used for distance (proximity) calculation. Since the number of blocks does not change with respect to image dimension(s), resolution invariance is, therefore, achieved (e.g. the same image in different resolutions will have identical proximity histograms/grids as opposed to significantly varying Correlograms due to its pixel-based computations). As shown in Fig. 3, we can use either L_1 or L_∞ norms for block-distance calculations. Let $b_1 = (x_1, y_1)$ and $b_2 = (x_2, y_2)$ be two blocks, the distance in L_1 norm can be defined as, $L_1 : \|b_1 - b_2\| = |x_1 - x_2| + |y_1 - y_2|$, and for the L_∞ norm, $L_\infty : \|b_1 - b_2\| = \max(|x_1 - x_2|, |y_1 - y_2|)$, respectively. Using the block indices in both norms, the block distances become integer numbers and note that for a full range histogram, the maximum (distance) range will be [1,L], where L is $N - 1$ in L_∞ and $2N - 2$ in L_1 norms, respectively. A block-wise proximity histogram for a DC pair c_i and c_j stores in its k^{th} bin the number of blocks hosting c_j (i.e. $\forall b_j | I(b_j) = c_j$, equivalent to the amount of color c_j in I) from all blocks hosting c_i (i.e. $\forall b_i | I(b_i) = c_i$, equivalent to amount of color c_i in I) at a distance k. So such a histogram clearly indicates how close or far two DCs and their spatial distribution with respect to



k = 1

3	5	5	5	5	5	5	3
5	8	8	8	8	8	8	5
5	8	8	8	8	8	8	5
5	8	8	8	8	8	8	5
5	8	8	8	8	8	8	5
5	8	8	8	8	8	8	5
5	8	8	8	8	8	8	5
3	5	5	5	5	5	5	3

k = 2

5	6	9	9	9	9	6	5
6	7	11	11	11	11	7	6
9	11	16	16	16	16	11	9
9	11	16	16	16	16	11	9
9	11	16	16	16	16	11	9
9	11	16	16	16	16	11	9
6	7	11	11	11	11	7	6
5	6	9	9	9	9	6	5

k = 7

15	8	8	8	8	8	8	15
8	0	0	0	0	0	0	8
8	0	0	0	0	0	0	8
8	0	0	0	0	0	0	8
8	0	0	0	0	0	0	8
8	0	0	0	0	0	0	8
8	0	0	0	0	0	0	8
15	8	8	8	8	8	8	15

k = 1

2	3	3	3	3	3	3	2
3	4	4	4	4	4	4	3
3	4	4	4	4	4	4	3
3	4	4	4	4	4	4	3
3	4	4	4	4	4	4	3
3	4	4	4	4	4	4	3
3	4	4	4	4	4	4	3
2	3	3	3	3	3	3	2

k = 2

3	4	5	5	5	5	4	3
4	6	7	7	7	7	6	4
5	7	8	8	8	8	7	5
5	7	8	8	8	8	7	5
5	7	8	8	8	8	7	5
5	7	8	8	8	8	7	5
4	6	7	7	7	7	6	4
3	4	5	5	5	5	4	3

k = 14

1	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1

Fig. 4. $N(b_i, k)$ templates in 8 × 8 block-grid ($D_{QT}^{\max} = 3$) for three range values in L_∞ (left) and L_1 (right) norms.

each other. Yet the histogram bins should be normalized by the total number of blocks, which can be found k blocks away from the source block b_i hosting the DC c_i because this number will significantly vary with respect to the distance (k), the position of source block (b_i) and the norm (L_1 or L_∞) used. Therefore, the kth bin of the normalized proximity histogram, $\Phi^{c_j}_{c_i}(k)$, between the DC pair c_i and c_j can be expressed as,

$$\Phi^{c_j}_{c_i}(k) = \sum_{b_i}\sum_{b_j}\Delta(b_i,b_j,k) \quad \text{where}$$

$$\Delta(b_i,b_j,k) = \left\{ \begin{array}{ll} N(b_i,k)^{-1} & \text{if } b_i \in I(c_i), b_j \in I(c_j), \|b_i - b_j\| = k \\ 0 & \text{else} \end{array} \right\} \tag{7}$$

Note that the normalization factor, $N(b_i,k)$, by the total number of neighbor blocks in distance k, is independent from the DC distribution and hence it is only computed once and used for all images in the database. Fig. 4 presents $N(b_i,k)$ templates computed for all blocks ($\forall b_i \in I$), both norms and some range values. In the figure for illustration purposes N is kept as 8 ($D^{\max}_{QT} = 3$) and note that normalization cannot be applied for those blocks, where $N(b_i,k) = 0$ since the range (k) is out of image boundaries and hence $\Phi^{c_j}_{c_i}(k) = 0$ *for* $\forall c_i$.

Once the $N(b_i,k)$ templates are formed, normalized proximity histogram computation takes $O(N^4)$. Note that this is basically independent from the image dimensions, W and H, and it is also a full-range computation (i.e. $k \in [1,\ldots,L]$), which may not be necessary in general (say, half image range may be quite sufficient since above this range most of the (central) blocks will have either out-of-boundary case, where $\Phi^{c_j}_{c_i}(k) = 0$ *for* $\forall c_i$ or only few blocks in the range, which is too low for obtaining "useful" statistics). For $D^{\max}_{QT} = 5 \Rightarrow N = 32$ and $N^{\max}_{DC} = 8$, as a typical setting, it requires 10,000 times less compared to Correlogram with a minimal range setting (i.e. 10% of image dimension range). In fact the real speed enhancement is much more since the computations in Correlogram involve several additions, multiplications and worst of all, divisions for probability computations; whereas, only additions are sufficient for computing $\Phi^{c_j}_{c_i}(k)$ as long as $N(b_i,k)^{-1}$ is initially computed and stored as the template. The memory requirements for the full-range computation are $O(N^2L)$ for storing $N(b_i,k)^{-1}$ and plus $O(N^2_{DC}L)$ for computing $\Phi^{c_j}_{c_i}(k)$, respectively. The memory space required for the typical settings given earlier will thus be ~500 Kb, which is a significant reduction compared to Correlogram. The typical storage space required per database image is <17 Kb with L_∞, and <33 Kb with L_1 norm), which is eventually 50 times smaller than the Auto-Correlogram's requirement ($O(md)$) with minimal m and d settings.

3.1.2. SCD Description via 2D proximity grids

This is an alternative approach for characterizing the inter-DC distribution by not only the respective proximities, but also their inter-occurrences accumulated over a 2D proximity grid. The process starts from the same configuration outlined earlier. Let the image I have $N \times N$ blocks, each of which hosts a single DC. 2D proximity grid, $\Psi^{c_j}_{c_i}(x,y)$, is formed by cumulating the co-occurrence of blocks hosting c_j (i.e. $\forall b_j | I(b_j) = c_j$) in a certain vicinity of the blocks hosting c_i (i.e. $\forall b_i | I(b_i) = c_i$) over a 2D (proximity) grid. In other words, via fixing the block b_i (hosting c_i) in the center bin of the grid (i.e. $x = y = 0$), the particular bin, which corresponds to the relative position of block b_j (hosting c_j) is incremented by one and this process is repeated for all blocks hosting c_j in a certain vicinity of b_i. Then the process is repeated for the next block (hosting c_i) until the entire image blocks are scanned for the color pair (c_i, c_j). As a result the final grid bins represent the inter-occurrences of the c_j blocks with respect to the ones hosting color c_i, within a certain range L (i.e. $\forall x,y \in [-L, L], L \leqslant N-1$). Although L

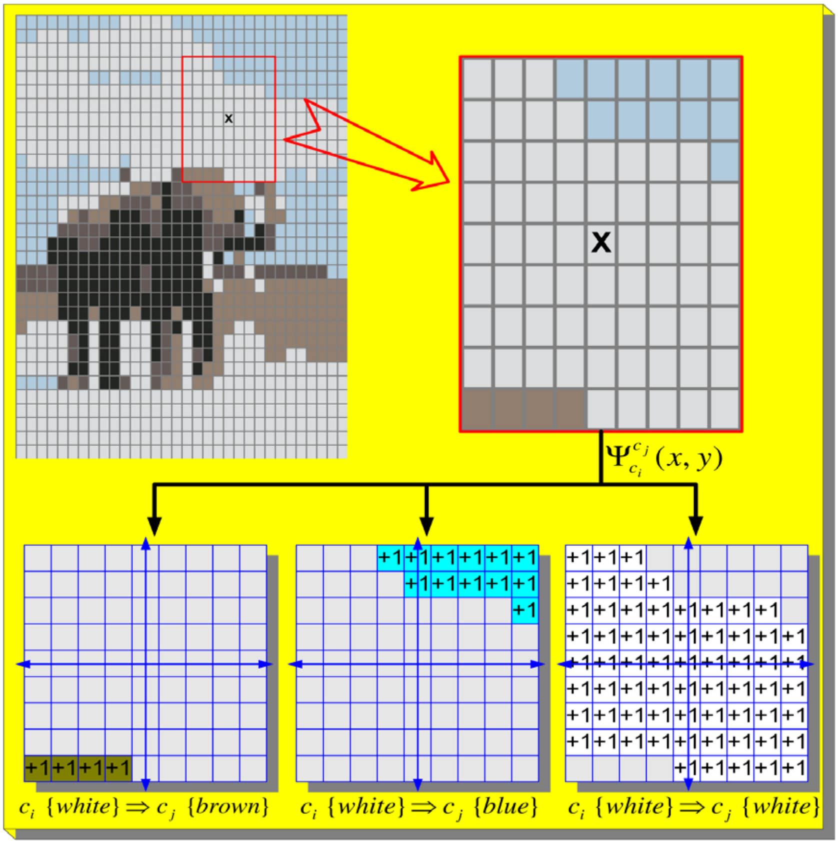

x
X
$\Psi^{c_j}_{c_i}(x,y)$
+1
c_i {white} ⇒ c_j {brown}
c_i {white} ⇒ c_j {blue}
c_i {white} ⇒ c_j {white}

Fig. 5. The process of proximity grid formation for the block (X) for $L = 4$.

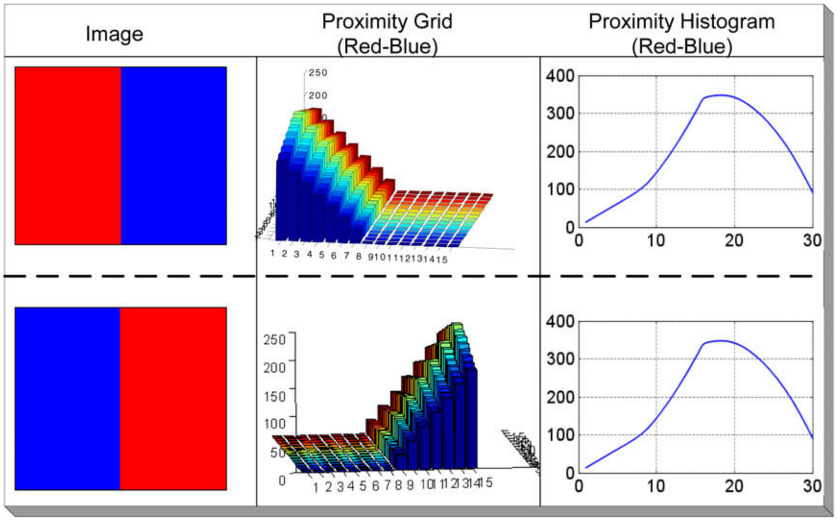


Fig. 6. Proximity grid vs. histogram for a sample color pair: red–blue.

can be set as $N-1$ for a full-range representation, it is, however, a highly redundant setting since $L \geqslant N/2$ cannot be fit exactly for any block without exceeding the image (block) boundaries. Therefore, $L < N/2$ would be a reasonable choice for L.

The computation of $\Psi_{c_i}^{c_j}(x,y)$ can be performed in a single pass through all image blocks. Let $b_i = (x_i, y_i)$ be the next block hosting the DC c_i. Fixing the b_i in the center (i.e. $\Psi_{c_i}^{c_j}(0,0)$), all image blocks within the range L from b_i (i.e. $\forall b_j = (x_i + x, y_i + y) \in I | \forall x, y \in [-L, L]$) are scanned and the corresponding (proximity) grid bin, $\Psi_{c_i}^{c_j}(x,y)$, for a color c_j in a block $b_j = (x_i + x, y_i + y) \in I$ is incremented by one. This process is illustrated on a sample image shown in Fig. 5. During the raster-scan of uniform blocks, the block with *white* DC updates only three proximity grids (*white* to *white, brown* and *blue*) since those DCs can only be found within the range of ±*L*. For illustration purposes we kept $D_{QT}^{\max} = 5 \Rightarrow N = 32$ and L as 4.

As a result such a proximity grid characterizes both inter-DC proximities and the relative spatial position (inter-DC direction) between two DCs. This is straightforward to see in the sample images in Fig. 6, where proximity grid distinguishes the relative direction of a DC pair, (red–blue) whilst proximity histogram cannot due to its scalar metric. Note that $\Psi_{c_i}^{c_j}(0,0) = 0$ for $i \neq j$ and $\Psi_{c_i}^{c_i}(0,0)$ indicates the total number of blocks hosting c_i. Since this is not a SCD property – rather a local DC property showing a noisy approximation of w_i (weight of c_i), it can be conveniently excluded from the feature vector and the remaining $(2L+1)^2 - 1$ grid bins are (unit) normalized by the total number of blocks, N^2, to form the final descriptor, $\overline{\Psi}_{c_i}^{c_j}(x,y)$, where $\overline{\Psi}_{c_i}^{c_j}(x,y) \leqslant 1, \forall x, y \in [-L, L]$.

Proximity grid computation takes $O(N^2 L^2)$. Similar to proximity histogram this is also independent from original image dimensions, W and H, and for a full range process, $(L = N/2)$, the same number of computations, $O(N^4)$, is obtained. However, instead of regular addition operations required for proximity histogram or multiplications and divisions for Correlogram, proximity grid computation requires only incrimination,. So for a typical grid dimension range, e.g. $N/8 < L < N/4$, the computation of proximity grid takes the shortest time. The memory space requirement is in $O(N_{DC}^2 \cdot L^2)$ and for a full range process $(L = N/2)$ with the typical settings $D_{QT}^{\max} = 5 \Rightarrow N = 32$ and $N_{DC}^{\max} = 8$, the memory required per database image will be 256 Kb, which is still smaller than the Auto-Correlogram $(O(md))$ even with the minimal m and d settings and it is equivalent to half of the memory required for the proximity histogram. Since $\Psi_{c_i}^{c_j}(x,y) = \Psi_{c_j}^{c_i}(-x,-y)$ (symmetry with respect to origin), the storage (disc) space requirement is even less, $O(N_{DC}^2 L^2)$; however, it requires 8 times more space than the proximity histogram. This is the cost of computing full-range proximity grid and therefore, it is recommended to employ the typical grid dimension range (e.g. $N/8 < L < N/4$) to reduce this cost to an acceptable level.

3.2. The proposed similarity metric: penalty-trio model

In a retrieval operation in an image database, a particular feature of the query image, *Q*, is used for (dis-) similarity measurement with the same feature of a database image, *I*. Repeating this process for all images in the database, *D*, and ranking the images according to their similarity distances yield the retrieval result. As shown in Fig. 3, the proposed color descriptors of *Q* and *I* contain both global and spatial color properties. Let C_i^Q and C_j^I represent the *i*th and *j*th $(i \leqslant N_{DC}^Q, j \leqslant N_{DC}^I)$ DC classes where N_{DC}^Q and N_{DC}^I are the number of DCs in *Q* and *I*, respectively. Along with these global properties, the proposed SCD descriptors of *Q* and *I* contain either proximity histogram $(\Phi_{c_i}^{c_j}(k))$ or grid $(\Psi_{c_i}^{c_j}(x,y))$ depending on the SCD *mode*. Henceforth for the similarity distance computation over the proposed color descriptor, both global and spatial color properties are used within a penalty-trio model, which basically penalizes the following mismatches between *Q* and *I*:

- P_ϕ : the amount of different (mismatching) DCs,
- the differences of the matching DCs in:
  - ○ P_G : global color properties,
  - ○ P_{SCD} : SCD properties.

So the penalty-trio over all color properties can be expresses as,

$$P_\Sigma(Q,I) = P_\phi(Q,I) + (\alpha P_G(Q,I) + (1-\alpha) P_{SCD}(Q,I)) \quad (8)$$

where $P_\Sigma \leqslant 1$ is the (unit) normalized total penalty, which corresponds to (total) color similarity distance and $0 < \alpha < 1$ is the weighting factor between global and spatial color properties. Note that all global color descriptors mentioned in Section 2.1 use only the first two (penalty) terms whilst discarding P_{SCD} entirely. Correlogram, on the other hand, works only over P_{SCD} without considering

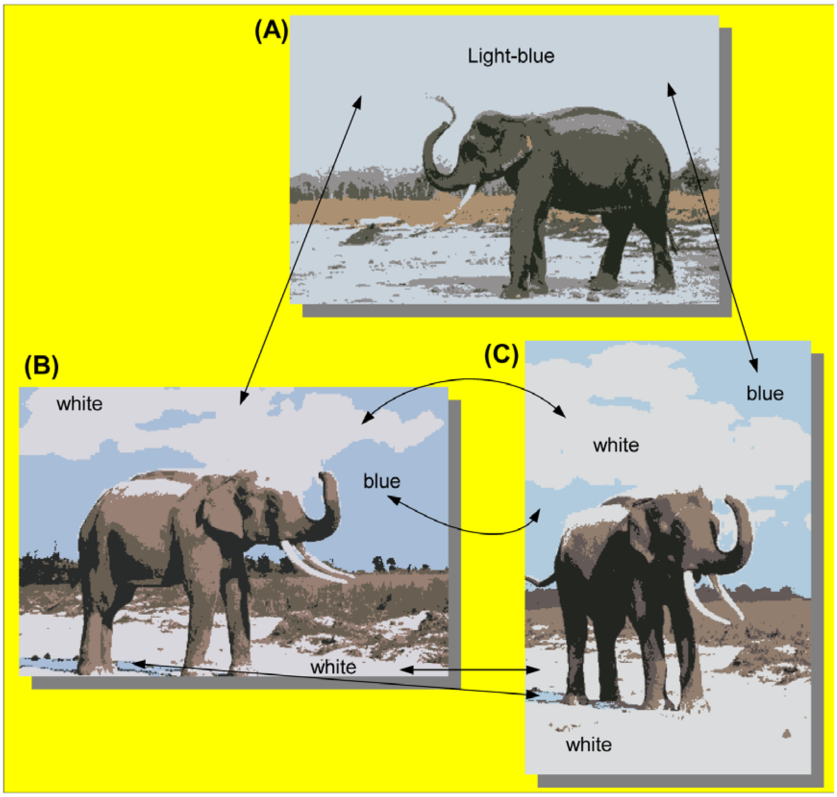


Fig. 7. One-to-one matching of DC pairs among three images (A–C).

any global properties. Therefore, the proposed penalty-trio model fuses both approaches to compute a complete distance measure from *all* color properties.

Color (DC) matching is a key factor in the underlying application. We therefore propose a two-level color partitioning: the first level partitions the group of color elements, which are too close for the human eye to distinguish, using a minimum (color) threshold, $T_C^{\min}$. Recall from the earlier discussion that such close color elements are clustered into DC classes, i.e. $|c_i - c_j| \leqslant T_S \textit{ for } \forall c_j \in C_i$ and using the same analogy $T_C^{\min}$ can conveniently be set as T_S. Another threshold, $T_C^{\max}$, is empirically set for the second level partitioning above which no color similarity can be perceived. Finally, for a given two DCs, where the inter-color distance falls between the two levels, i.e. $T_C^{\min} < |c_i - c_j| < T_C^{\max}$, there exists a certain level of (color) similarity but not too close so as to be perceived as identical.

Define such colors, which show some similarity, as "matching" and let $T_C^{\max}$ be used to partition the mismatching colors from the matching ones. One can form two sets: matching (S^M) and mismatching (S^ϕ) DC classes from C_Q and C^I by assigning each DC, $c_i \in C_i$, in one set, which cannot match any DC, $c_j \in C_i$, in the other (i.e. $|c_i - c_j| > T_C^{\max} \textit{ for } \forall i,j$) into S^ϕ and the rest (with at least one match) into S^M. Note that $S^M + S^\phi = C^Q + C^I$ and using the DCs in S^ϕ, P_ϕ can directly be expressed as,

$$P_\phi(Q, I) = \frac{\sum(w_i | C_i \in S^\phi)}{2} \leqslant 1 \tag{9}$$

The dissimilarity (penalty, P_ϕ) increases proportionally with the total amount (weight) of mismatching DCs. In one extreme case, where there are no colors matching, $S^M = \{\phi\} \Rightarrow \quad P_\Sigma = P_\phi = 1$ means that the two images have no similar (matching) colors. In another extreme case, where all DCs are matching, so $S^\phi = \{\phi\} \Rightarrow P_\phi = 0$, color (dis-)similarity will only emerge from global (P_G) and spatial (P_{SCD}) color properties of the (matching) DCs. Typically, P_ϕ contributes a certain color distance as a natural consequence of mismatching colors between Q and I, yet the rest of the distance will result from the cumulated difference of color matching. This is, however, not straightforward to compute since one DC in Q can match one or more DCs in I (or *vice versa*). One solution is to apply color quadratic distance [10] to fuse DC distances into the total similarity metric. However, besides its serious drawbacks mentioned earlier, this formulation can be applied only to distance calculation from *global* DC properties and hence cannot address how to fuse SCD distances (from proximity grid or histogram of each individual DC pair). Another alternative is to enforce a one-to-one DC matching, i.e. one DC alone in Q can match a single DC in I by choosing the best match and discarding the other matches. This, as well, induces serious errors due to the following fact: DC extraction is a dynamic clustering algorithm in color domain and due to variations in color composition of the scenery or its pre-fixed parameters (thresholds), it can result in over- or under-clustering. Therefore, similar color compositions can be clustered into different number of DCs and enforcing one-to-one matching may miss part of matching DCs from both global and spatial similarity computation. A typical example of such a consequence can be seen in Fig. 7, where there are three images with highly similar content, i.e. "an elephant under cloudy sky". In two images (B and C), the *cloud* and *sky* are distinguished during DC extraction with separate *blue* and *white* DCs; however, in image A, only one DC (light-blue) is extracted with the same parameters. Consequently there is no (one-to-one) matching problem between B and C and such a matching will naturally reflect similar global and spatial DC properties, but between A and B or C, if the single DC (*light-blue*) is matched only with one DC (*white* or *blue*) this will obviously yield an erroneous result on both global and spatial similarity computations since neither DC (*white* or *blue*) properties (weight, distribution, proximities to other DCs, etc.) are similar to the one in A (light-blue).

As a result, before computing P_G and P_{SCD}, the DC sets in Q (or I), which are in a close vicinity of a single DC in I (or Q) should be first fused into a single DC. For instance, in Fig. 7, the DC *light-blue* in image A is close to both *white* and *blue* in image B (and C); therefore, both colors in B should be fused into a new DC (perhaps a similar light-blue color) and then P_G and P_{SCD} can be computed accurately between A and B. In order to accomplish this, $T_C^{\min} = T_S$ is used for matching the close DCs and a two-fold matching process is performed via function **TargetMatch**, which first verifies and then fuses some DCs in the target set, T, if required by any DC in the source set, S. Let $S_Q^M \subset S^M$ and $S_I^M \subset S^M$ be the sets of matching DCs for Q and I, respectively. Since any DC in any set can request fusing two or more DCs in the other set, the function is called twice, i.e. first **TargetMatch** (S_Q^M, S_I^M), then **TargetMatch** (S_I^M, S_Q^M). Accordingly, **TargetMatch** can be expressed as follows:

```
TargetMatch $(S, T)$
• For $\forall c_i \in S$ do:
    ○ Let $L_i^M$ be the matching DC list for $c_i$
    ○ For $\forall c_j \in T$ do:
        – If $|c_i - c_j| \leqslant T_S$ then $c_j \rightarrow L_i^M$
    ○ If $\left|L_i^M\right| \geqslant 2$ then
        – Let $L_i^N = T - L_i^M$ be the non-matching list
        – $C_X = \textbf{FuseDCs}\left(L_i^M, L_i^N\right)$
        – UPdate $T = L_i^N + C_X$
• Return.
```

```
FuseDCs $(L_i^M, L_i^N)$
• Create $C_X : \{c_x, w_x, \sigma_x\}$ by using $\forall C_j \in L_i^M$
    ○ $w_x = \sum_{C_j \in L_i^M} w_j$
    ○ $c_x = \frac{\sum_{C_j \in L_i^M} w_j c_j}{\sum_{C_j \in L_i^M} w_j}$ and $\sigma_x = \frac{\sum_{C_j \in L_i^M} w_j \sigma_j}{\sum_{C_j \in L_i^M} w_j}$
• From both $\mathrm{X}_{c_x}^{c_x}$ and $(\mathrm{X}_{c_x}^{c_j} - \mathrm{X}_{c_j}^{c_x}) | \forall c_j \in L_i^N$
    ○ $\mathrm{X}_{c_x}^{c_x} = \sum_{C_j \in L_i^M} \sum_{C_k \in L_i^M} \mathrm{X}_{c_j}^{c_k}$
    ○ $\mathrm{X}_{c_j}^{c_x} = \sum_{C_k \in L_i^M} \mathrm{X}_{c_j}^{c_k} | \forall c_j \in L_i^N$
    ○ Compute $\mathrm{X}_{c_x}^{c_j}$ from $\mathrm{X}_{c_j}^{c_x}, \forall c_j \in L_i^N$
• Return $C_X$
```

The function, **FuseDCs**, fuses all DCs in the list, L_i^M, reforms the SCD descriptors of all (updated) DC pairs ($\Phi_{c_i}^{c_j}(k)$ or $\Psi_{c_i}^{c_j}(x, y)$) and finally returns a new (fused) DC, C_X. Then the target set, T, is updated accordingly. Let $\mathrm{X}_{c_i}^{c_j}$ be the SCD operator (i.e. $\Phi_{c_i}^{c_j}$ or $\Psi_{c_i}^{c_j}$ depending on the SCD *mode* as shown in Fig. 3) and $\mathrm{X}_{c_i}^{c_1} + \mathrm{X}_{c_i}^{c_2}$ can be defined as:

$$\mathrm{X}_{c_i}^{c_1} + \mathrm{X}_{c_i}^{c_2} = \left\{ \begin{array}{l} \Phi_{c_i}^{c_1}(k) + \Phi_{c_i}^{c_2}(k) \forall k \in [1, L] \\ \Psi_{c_i}^{c_1}(x, y) + \Psi_{c_i}^{c_2}(x, y) \forall x, y \in [-L, L] \end{array} \right\} \tag{10}$$

Let $\oplus$ be the fusing operator over DC classes. It is simple to show that $\mathrm{X}_{c_i}^{c_1 \oplus c_2} = \mathrm{X}_{c_i}^{c_1} + \mathrm{X}_{c_i}^{c_2}, c_{1,2} \neq c_i$. Once the DCs in L_i^M are fused, then they are removed along with their SCD descriptors whilst keeping the DCs (and their internal SCD descriptors) in L_i^N intact. The new (fused) DC, C_X (along with its SCD descriptors) is inserted into the target set, T. Recall from the earlier remarks on SCD descriptor properties, i.e. $\Phi_{c_i}^{c_j}(k) = \Phi_{c_j}^{c_i}(k)$ and $\Psi_{c_i}^{c_j}(x, y) = \Psi_{c_j}^{c_i}(-x, -y)$, therefore, once $\mathrm{X}_{c_j}^{c_x}, \forall c_j \in L_i^N$ are formed, it is straightforward to compute $\mathrm{X}_{c_x}^{c_j}, \ \forall c_j \in L_i^N$. After the consecutive calls of **TargetMatch** function, all DC sets in each set, which are close (matching) to a particular DC in the other set are fused and thus one-to-one matching can be conveniently performed by selecting the best matching pair in both sets. As a result the number of DCs in both (updated) sets, $S_Q^M, \ S_I^M$ become equal (i.e. $\left|S_Q^M\right| = \left|S_I^M\right| = N_M$). Assume without loss of generality that ith DC class in set $C_i^Q : \{c_i^Q, w_i^Q, \sigma_i^Q\} \in S_Q^M$ matches the ith DC in set $C_i^I : \{c_i^I, w_i^I, \sigma_i^I\} \in S_I^M$ (i.e. via sorting one set with respect to the other). So the penalties for global and SCD properties can be expressed as,

$$P_G(Q, I) = \beta \sum_{i=1}^{N_M} \left|w_i^Q - w_i^I\right| + (1 - \beta) \frac{\sqrt{\sum_{i=1}^{N_M} \left(c_i^Q - c_i^I\right)^2}}{T_C^{\max} N_M} \leqslant 1$$

$$P_{SCD}(Q, I) = \left\{ \begin{array}{l} \dfrac{\sum_{i=1}^{N_M} \sum_{j=1}^{N_M} \sum_{x,y=-L}^{L} \Delta\left(\overline{\Psi}_{c_i^Q}^{c_j^Q}(x,y) - \overline{\Psi}_{c_i^I}^{c_j^I}(x,y) \right)}{N_M^2 (2L+1)^2} \leqslant 1 \\ \\ \dfrac{\sum_{i=1}^{N_M} \sum_{j=1}^{N_M} \sum_{k=1}^{L} \Delta\left(\frac{\Phi_{c_i^Q}^{c_j^Q}(k)}{\max(w_i^Q, w_j^Q)} - \frac{\Phi_{c_i^I}^{c_j^I}(k)}{\max\left(w_i^I, w_j^I\right)} \right)}{N_M^2 L} \leqslant 1 \end{array} \right\}$$

$$\text{where} \quad \Delta(x - y) = \left\{ \begin{array}{ll} 0 & \text{if } x = y = 0 \\ \frac{|x-y|}{(x+y)} & \text{else} \end{array} \right\} \tag{11}$$

where $0 < \beta < 1$, similar to α, is the weighting factor between the two global color properties: DC weights and centroids. Δ is the normalized difference operator, which emphasizes the difference from zero–nonzero pairs (e.g. =1). This is a common consequence when the DC pairs' area is relatively small but their SCDs are quite different. It also suppresses the bias from similar SCDs of two DCs with large weights. Note that P_{SCD} computation should be independent from the effect of DC weights since this is already taken into consideration within P_G computation. As a result the combination of P_G and P_{SCD} represents the amount of dissimilarity present in all color properties and the unit normalization allows the combination in a configurable way with weights α, β, which can favor one color property to another. With the combination of P_ϕ, which represents the natural color dissimilarity due to mismatching, the penalty-trio models a complete similarity distance between two color compositions.

4. Experimental results

Simulations are performed to evaluate the proposed color descriptor efficiency with respect to HVS perceptive criteria (subjective test) and to compare retrieval (via QBE) performances within image databases indexed by the proposed and competing (Correlogram and MPEG-7 DCD [22,33]) *FeX* modules. We do not see any need to compare with other color descriptors such as color histograms, CCV, color moments, etc. since the study in [20] clearly demonstrates the Correlegram's superiority over them. In the experiments performed in this section, we used four sample databases:

(1) **Corel_1K** Image Database: There are total of 1000 medium resolution (384 × 256 pixels) images from 10 classes with diverse contents such as wild life, city, buses, horses, mountains, beach, food, African natives, etc.
(2) **Corel_10K** Image Database: There are 10,000 images from Corel database bearing 100 distinct classes, each of which contains 100 images with a similar content.
(3) **Corel_20K** Image Database: There are 20,000 images from Corel database bearing 200 distinct classes, each of which contains 100 images with a similar content.
(4) **Synthetic** Image Database: There are 1089 synthetic images covering various color compositions that are artificially created.

The classes in Corel databases are extracted by the *ground-truth*, considering the content similarity—not the color distribution similarity. For instance a red car and blue car are still in the same "Cars" class, although their colors do not match at all. Accordingly, color-based retrievals are also evaluated using the same *ground-truth* methodology, i.e. considering a retrieval as relevant only if its content matches with the query. Note that we had to select all sample databases containing images with mediocre resolutions; otherwise it is not feasible to apply Correlogram method and especially for **Corel_10K** and **Corel_20K**, as we have witnessed severe feasibility problems due to its computational complexity. Finally the performance evaluation is presented over **Synthetic** database is to demonstrate the true description power of the proposed technique whenever color alone entirely characterizes the content of the image. Moreover, the robustness of the proposed descriptor is also evaluated against the changes of resolution, aspect ratio, color variations, translation, etc.

All experiments are carried out on a Pentium-5 1.8 GHz computer with 1024 MB memory. If not stated otherwise, the following parameters are used for all the experiments performed throughout this section: $N_{DC}^{\max} = 6,\ T_A = 2\%,\ T_S = 15$ for DC extraction, $T_W = 96\%,\ D_{QT}^{\max} = 6$ for QT decomposition and $T_C^{\min} = 45$, $T_C^{\min} = T_S, \alpha = \beta = 0.5$ for penalty-trio model. For Auto-Correlogram, we set RGB color histogram quantization as $8 \times 8 \times 8$ (m = 512 colors) with d = 20 for **Corel_1K** but $4 \times 4 \times 4$ (m = 64 colors) with d = 10 for **Corel_10K** and **Corel_20K**. For Correlogram, we use $4 \times 4 \times 4$ bins for **Corel_1K** and $3 \times 3 \times 3$ bins **Corel_10K** with d = 10. We had to use only Auto-Correlogram for **Corel_20K** due to Correlogram's infeasible memory requirement for this database size. We use the same DC extraction parameters for MPEG-7 DCD and the proposed descriptor. A MUVIS application, *DbsEditor*, dynamically uses the respective *FeX* modules for feature extraction to index sample databases with the aforementioned parameters. Afterwards, *MBrowser* application is used to perform similarity-based retrievals via QBE (Query-by-Example) operations. A query image is chosen among the database items to be the "Example" and a particular *FeX* module (e.g. MPEG-7 DCD) is selected to retrieve and rank the similar (based on color) images using only the respective (MPEG-7 DCD) features and an appropriate distance metric implemented within the *FeX* module. The recommended distance metrics are implemented for each *FeX* module, i.e. quadratic distance for MPEG-7 DCD and L_1 norm for Correlogram.

In order to measure the retrieval performance, we used an unbiased and a limited formulation of the *Normalized Modified Retrieval Rank* (*NMRR*(q)), which is defined in MPEG-7 as the retrieval performance criteria per query (q). It combines both of the traditional hit-miss counters; *Precision–Recall*, and further takes the ranking information into account as given in the following expression:

$$\begin{aligned} AVR(q) &= \frac{\sum_{k=1}^{N(q)} R(k)}{N(q)} \text{ and } W = 2N(q) \\ NMRR(q) &= \frac{2AVR(q) - N(q) - 1}{2W - N(q) + 1} \leqslant 1 \\ ANMRR &= \frac{\sum_{q=1}^{Q} NMRR(q)}{Q} \leqslant 1 \end{aligned} \tag{12}$$

where $N(q)$ is the minimum number of relevant (via *ground-truth*) images in a set of Q retrieval experiments, $R(k)$ is the rank of the kth relevant retrieval within a window of W retrievals, which are taken into consideration during per query, q. If there are less than $N(q)$ relevant retrievals among W then a rank of W + 1 is assigned for the remaining (missing) ones. $AVR(q)$ is the average rank obtained from the query, q. Since each query item is selected within the database, the first retrieval will always be the item queried and this obviously yields a biased $NMRR(q)$ calculation and it is, therefore, excluded from ranking. Hence the first relevant retrieval ($R(1)$) is ranked by counting the number of irrelevant images *a priori* and note that if all $N(q)$ retrievals are relevant, then $NMRR(q)$ = 0, the best retrieval performance is thus achieved. On the other hand, if none of relevant items can be retrieved among W then $NMRR(q)$ = 1, as the worst case. Therefore, the lower $NMRR(q)$ is the better (more relevant)

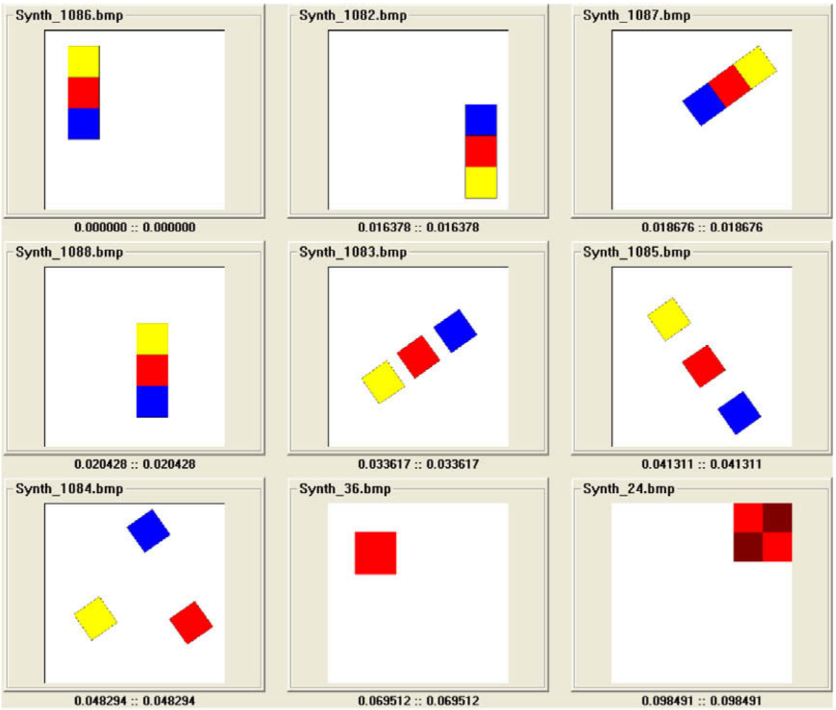


Fig. 8. Query of a three-color object (top-left) in Synthetic database.

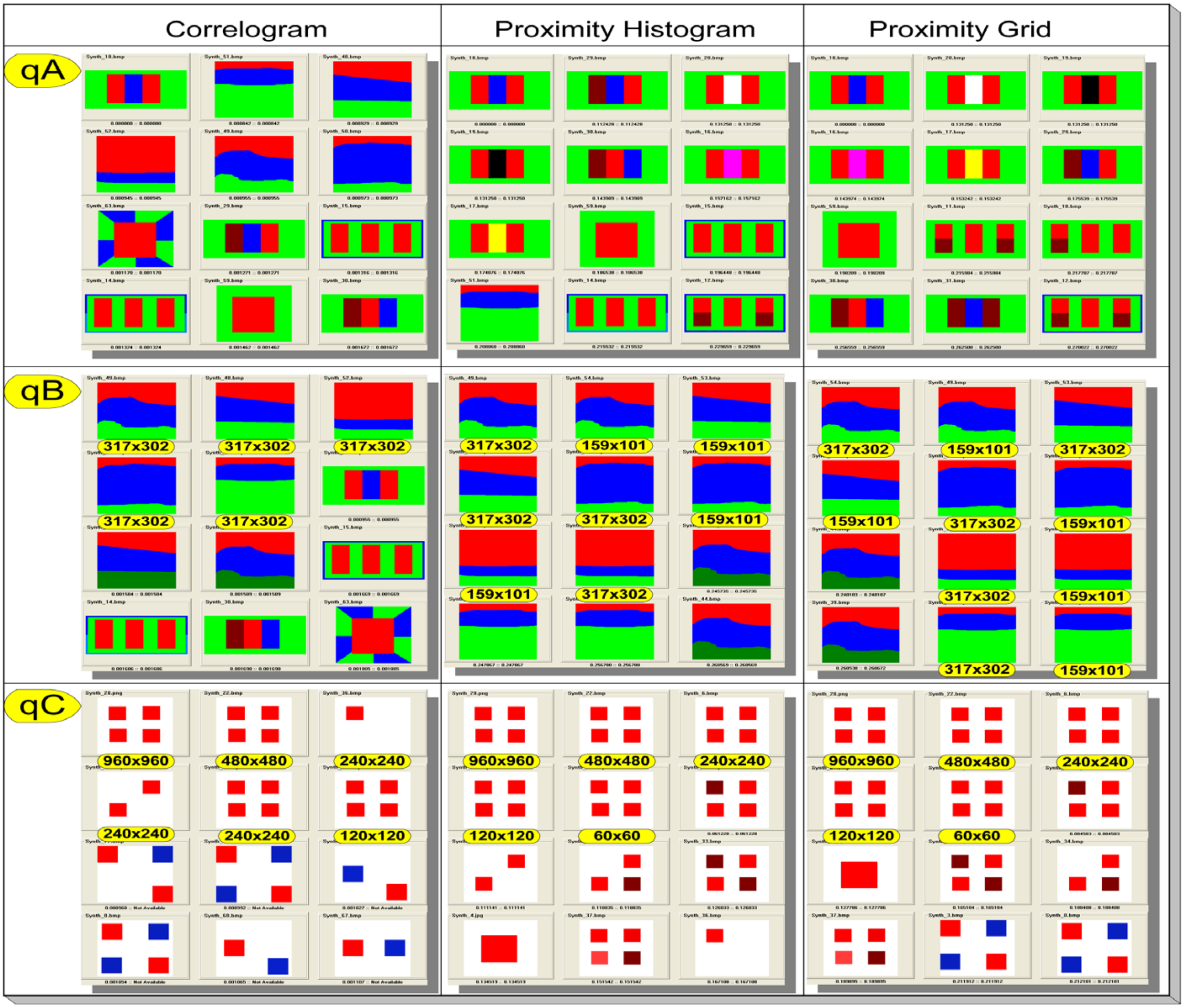


Fig. 9. Three queries, qA–qC, in Synthetic database via Correlogram (left) and the proposed descriptor with proximity histogram (middle) and proximity grid (right). Some dimensions are tagged in yellow boxes. Top-left image is the query.

the retrieval is, for the query, *q*. Keeping the number of QBE experiments sufficiently high, the average *NMRR*, *ANMRR*, as expressed in Eq. (12) can thus be used as the retrieval performance criteria.

4.1. Retrieval performance on synthetic images

The images in **Synthetic** database contain colored-regions in geometric and arbitrary shapes within which uniform samples from the entire color space are represented. In this way the color matching accuracy can be visually evaluated and the first two penalty terms, P_ϕ and P_G can be individually tested. Furthermore, the same (or matching) colors form different color compositions by varying their region's shape, size and/or inter-region proximities. Hence, this allows us to test both individual and mutual penalty terms P_G and P_{SCD}. Finally the penalty-trio's cumulative accuracy and robustness against variations of resolution, translation and rotation can also be tested and compared against the Correlogram.

Fig. 8 presents a snapshot of the query of an image with 3-color squares on a white background. The proposed color descriptor is used with proximity histogram as the SCD descriptor and the re-

Table 1
Similarity distances and ranks of *A* and *B* in Fig. 7 when C is queried in Corel_1K.

Query: C	P_Σ		Rank	
	A	*B*	*A*	*B*
Fusing	0.176	0.156	3	1
Without Fusing	0.585	0.205	258	1

Table 2
ANMRR scores of the proposed and the competing descriptors for three Corel databases.

Descriptors	Corel_1K (34 queries)	Corel_10K (176 queries)	Corel_20K (222 queries)
MPEG-7 DCD	0.18	0.458	0.461
Auto-Correlogram	0.222	0.381	0.444
Correlogram	0.195	0.357	*NA*
Proposed (**Prox. Histogram**)	**0.154**	**0.263**	**0.357**
Proposed (**Prox. Grid**)	**0.162**	**0.291**	**0.39**

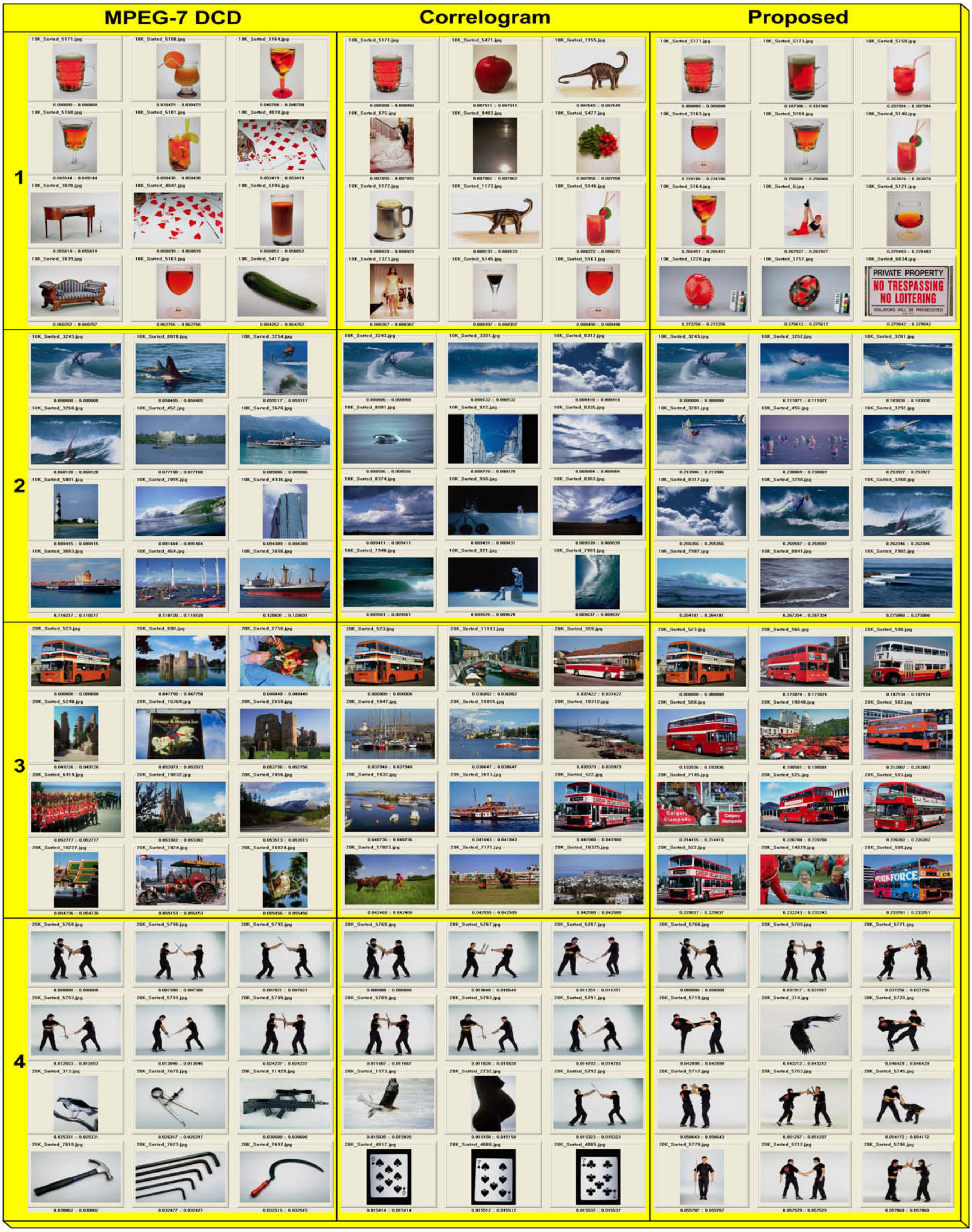


Fig. 10. Four typical queries using three descriptors in Corel_10K database. Top-left is the query image.

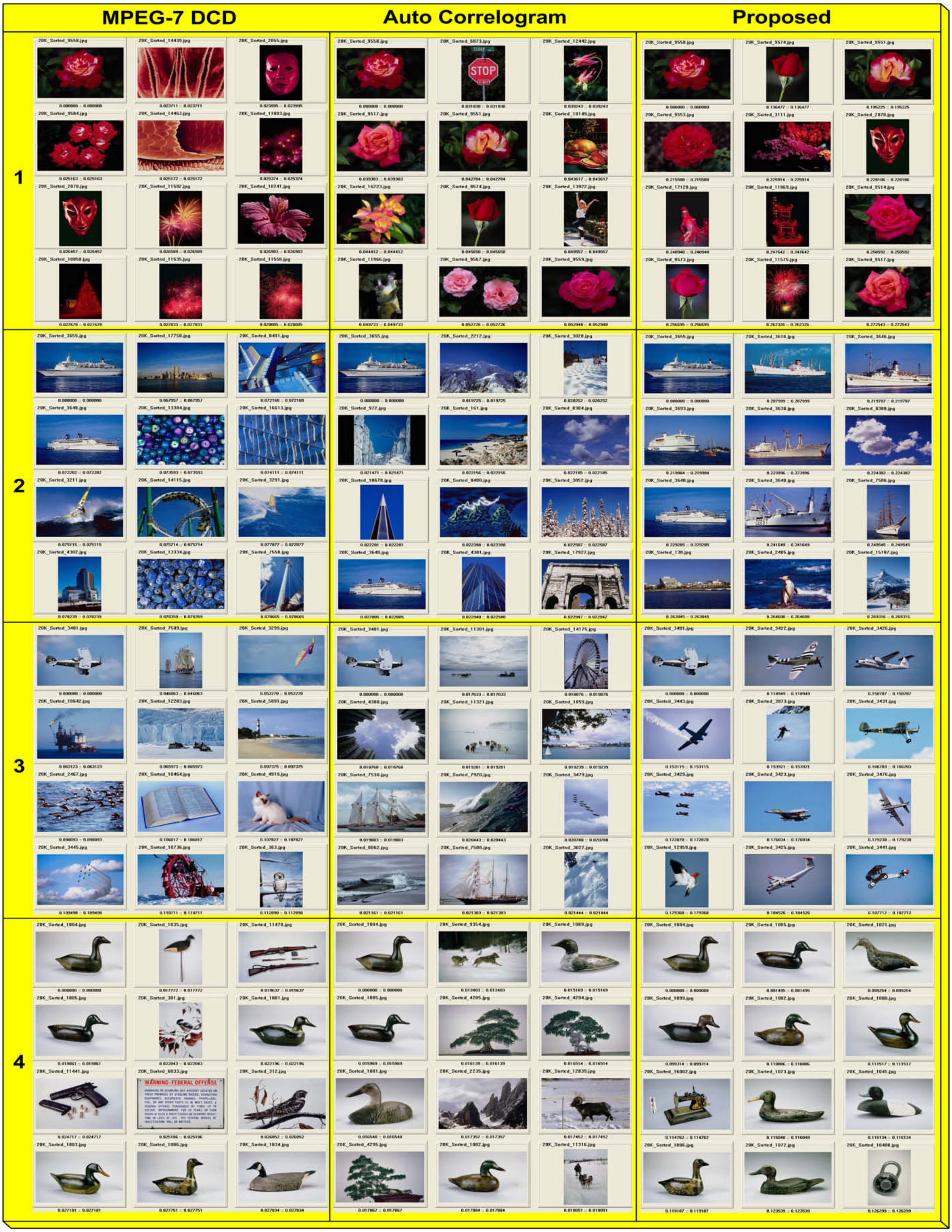


Fig. 11. Four typical queries using three descriptors in Corel_20K database. Top-left is the query image.

trieval results are ranked from left to right and top to bottom and the similarity distances are given on the bottom of the images. Among the first six retrievals, the same amount of identical colors are used and hence $P_\phi = P_G = 0$, which allows us to test the accuracy of P_{SCD} alone. The first three retrievals have insignificant (dis-)similarity distances and this demonstrates the robustness of P_{SCD} against the variations of rotation and translation. The 4th, 5th and 6th ranks present cases, where spatial proximity between the three colors starts to differentiate and hence SCD descriptor reflects the proximity differences successfully. For the 7th ($P_\phi \neq 0$) and 8th ranks ($P_G \neq 0$)P_Σ starts to build up significantly since the color composition changes drastically due to emerging and missing color components.

Fig. 9 shows three queries in **Synthetic** database with different color compositions and resolutions. In **qA**, both proximity histogram and proximity grid successfully retrieve images with similar color compositions; whereas, the Correlogram cannot due to its invariance to weight (area) and limited range. The area invariance effect can be easily seen in 2nd and particularly 3rd ranks, where entirely different red and green weights occur. The same comments can be made for **qB** for 5th and all ranks above 7th. Moreover in **qB**, it is obvious that Correlogram cannot retrieve the image with identical color composition among the first 11 ranks due to its resolution (pixel-based) sensitivity. Note further that the proposed descriptor with both proximity histogram and grid first retrieves the color compositions, where all colors are perfectly matching ($P_\phi = 0$) with the weights in a close vicinity ($P_G \neq 0$) and then balances between mismatching colors and weight differences of the matching ones. **qC** is particularly shown here to emphasize the effect of image resolution over Correlogram and the proposed descriptor. The query of the largest image among the others with dimensions in five different resolutions logarithmically scaled from 60 to 960 but the same color composition (four red squares over white background), result in accurate ranking for the proposed descriptor; however, Correlogram retrieves accurately only one whilst the other two are shifted to lower ranks and the one (with 60 × 60 dimension) is missed within the first 12 ranks.

4.2. Retrieval performance on natural image databases

In this section, three sample databases (**Corel_1K, Corel_10K** and **Corel_20K**) are indexed using each *FeX* module and each individual (sub-) feature is used for retrieval. As presented in Table 1, the first retrieval experiment is performed to demonstrate the effect of DC fusing over the retrieval accuracy. Similar results of several retrieval experiments approve that DC fusing becomes the key factor for the success of the proposed descriptor. Therefore, DC fusing is applied for the rest of the experiments presented in this section.

Table 2 presents ANMRR results and the query dataset size of each of the three Corel databases, respectively. The query dataset is prepared a priori by regarding a certain degree of color content coherency, that is, the content similarity can mostly be perceived by color similarity; however, a unique, one-to-one correspondence between content and color similarities, as in the synthetic images given in the previous section, can never be guaranteed in such natural images due to the presence of other visual cues, such as texture, shape, etc. Nevertheless, according to ANMRR scores presented in the table, in all **Corel** databases the proposed descriptor with either SCD modes achieves superior retrieval performance than the competing methods, i.e. Correlogram, Auto-Correlogram and MPEG-7 DCD combined with the quadratic distance computation. Moreover, we observed that in the majority of the queries (be-

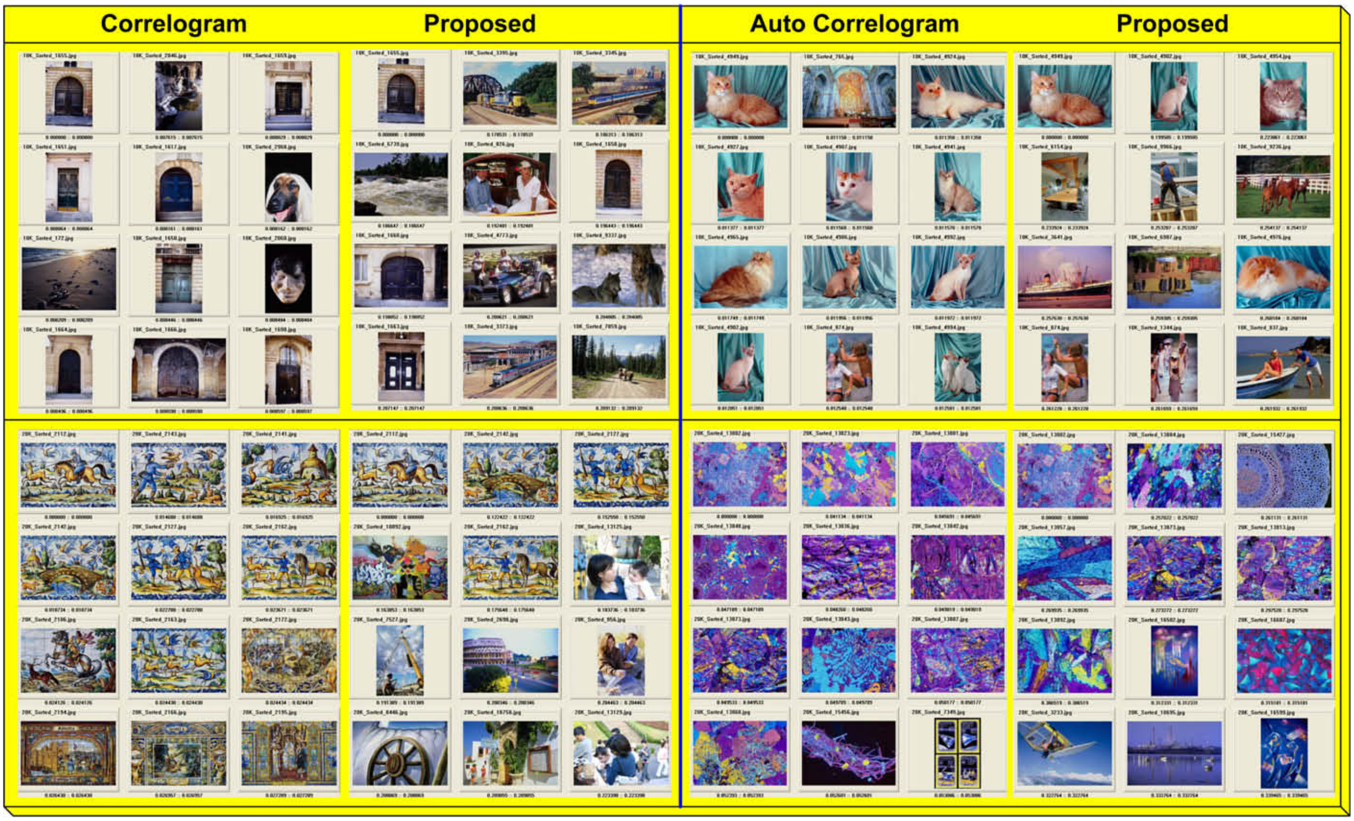


Fig. 12. Two queries in Corel_10K (left) and Corel_20K (right) databases, where (Auto-) Correlogram performs better than the proposed descriptor. Top-left is the query image.

tween 58% and 78%), the proposed method outperforms (auto-) Correlogram, whereas the figure is even higher (76–92%) with MPEG-7 DCD. Finally, for shorter descriptor size with proximity histograms, we use L_∞ norm since comparative retrieval results promise no significant gain of using L_1 (e.g. for **Corel_10K**, ANMRR score of the proposed method with proximity histogram using L_1 is 0.254).

For visual evaluation, we present four retrieval results in both **Corel_10K** and **Corel_20K** databases using all three descriptors. For the queries as shown in Fig. 10, we used proximity grid in the proposed descriptor against Correlogram and MPEG-7 DCD. In the 1st, 2nd and 4th queries, one can easily notice the erroneous retrievals of Correlogram due to its color area insensitivity (e.g. compare the amount of *red, white* and *black* colors between the query and 5th ranked image in the 1st query). As mentioned earlier, in such large databases the co-occurrence probabilities can (accidentally) match images with significantly different color proportions. Particularly in the 1st and 4th queries, erroneous retrievals of MPEG-7 DCD occur due to the lack of SCD description, which also makes accidental matches between (dissimilar) images with close color proportions (e.g. in the 1st query, the amount of *white, red* and *black* colors is quite close between the query and 6th, 7th and 8th ranks; however, their SCDs are not).

For the queries shown in Fig. 11, we used proximity histogram in the proposed descriptor against Auto-Correlogram and MPEG-7 DCD. Similar conclusions can be drawn for the retrieval results. Furthermore, note that the amount of erroneous retrievals is increased particularly in 2nd and 3rd queries since the database size is doubled and hence accidental matches occur more often than before. However, in both databases (Auto-) Correlogram may occasionally perform better than the proposed descriptor, such as the queries shown in Fig. 12, where significant (color) textures are present in all query images. This is indeed in accordance with the earlier remark stating that Correlogram is indeed a colored *texture* descriptor and hence it can outperform any *color* descriptor whenever a textural structure is dominant.

5. Conclusions

The color descriptor presented in this paper characterizes the perceptual properties of the color composition in a visual scenery in order to maximize the description power. In other words, the so-called outliers, which are the un-perceivable color elements, are discarded for description efficiency using a top-down approach while extracting global and spatial color properties. In this way, severe problems and limitations of traditional pixel-based methods are effectively avoided and in spatial domain only the perceived (visible) color components can be truly extracted using QT decomposition. In order to reveal the true SCD properties, proximity histogram and proximity grid, representing the inter-proximity statistics in scalar and directional modes, are proposed.

During the retrieval phase, one-to-many DC matching is performed in order to apply the penalty-trio model over matching (and possibly fused) DC sets. This greatly reduces the faulty mismatches and erroneous similarity distance computations. The proposed penalty-trio model computes the normalized differences in both spatial and global color properties and combines all so as to yield a complete comparison between two color compositions. Experimental results approve the superiority of the proposed descriptor over the competing methods in terms of discrimination power and retrieval performance especially over large databases. The proposed color descriptor has a major advantage of being applicable to any database size and image resolution. Thus it does not suffer from the infeasibility problems and severe limitations of Correlogram. Finally, it achieves a significant performance gain in ANMRR scores. However, this remained below our higher performance expectations particularly when compared with Correlogram due to two reasons: first and foremost, Correlogram has the advantage of describing texture in color images thanks to its pixel level analysis via co-occurrence probabilities. Yet the major reason is that the color similarity alone does not really imply the content-similarity. This degrades the retrieval performance of the proposed technique in great amount on several experiments. For instance when an image with *gray horse* on a *green* field and a *blue* sky is queried, all retrievals with a *gray elephant* and similar background are counted as irrelevant (since they do not belong to *horse* class) although the color distribution is quite similar. Many other such "irrelevant" retrievals with similar color properties can be seen in the figures in Section 4. In short color properties correlate with the true content only in a certain extend, but cannot be used as the single cue to characterize the entire content [34].

Current and planned research work include: configuring our penalty-trio model dynamically and adaptively according to color compositions of the images compared and integrating second order statistics from both global and spatial properties into the descriptor. Adopting a multi-scale approach into both SCD modes will also be considered. Finally, combining the proposed approach with successful texture descriptors, such as [30] may prove useful.

Acknowledgement

This work was supported by the Academy of Finland, Project No. 213462 (Finnish Centre of Excellence Program (2006–2011).

References

[1] G.P. Babu, B.M. Mehtre, M.S. Kankanhalli, Color indexing for efficient image retrieval, Multimedia Tools and Applications 1 (November) (1995) 327–348.

[2] K.-H. Brandenburg, MP3 and AAC explained, in: AES 17th International Conference, Florence, Italy, September 1999, pp. 17–009.

[3] E.L. van den Broek, P.M.F. Kisters, L.G. Vuurpijl, The utilization of human color categorization for content-based image retrieval, in: Proceedings of Human Vision and Electronic Imaging IX, San José, CA (SPIE, 5292), 2004, pp. 351–362.

[4] S.F. Chang, W. Chen, J. Meng, H. Sundaram, D. Zhong, VideoQ: an automated content based video search system using visual cues, In: Proceeding of ACM Multimedia, Seattle, 1997.

[5] Y.D. Chun, N.C. Kim, I.H. Jang, Content-based image retrieval using multiresolution color and texture features, IEEE Transactions on Multimedia 10 (6) (2008) 1073–1084.

[6] I.J. Cox, M.L. Miller, S.O. Omohundro, O.N. Yianilos, PicHunter: bayesian relevance feedback for image retrieval, in: Proceedings of Int'l Conference on Pattern Recognition, 1996, pp. 361–369.

[7] Y. Deng, C. Kenney, M.S. Moore, B.S. Manjunath, Peer group filtering and perceptual color image quantization, in: Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS, vol. 4, 1999, pp. 21–24.

[8] J. Fauqueur, N. Boujemaa, Region-Based Image Retrieval: Fast Coarse Segmentation and Fine Color Description, in: Proceedings of IEEE International Conference on Image Processing (ICIP'2002), Rochester, USA, September 2002.

[9] Y. Gong, C.H. Chuan, G. Xiaoyi, Image indexing and retrieval using color histograms, Multimedia Tools and Applications 2 (1996) 133–156.

[10] J. Hafner, H.S. Sawhney, W. Esquitz, M. Flickner, W. Niblack, Efficient color histogram indexing for quadratic form distance functions, IEEE Transaction on Pattern Analysis and Machine Intelligence 17 (1995) 729–736.

[11] J. Huang; S.R. Kumar, M. Mitra, W.-J. Zhu, R. Zabih, Image indexing using color correlograms, in: Proceedings of Computer Vision and Pattern Recognition, 17–19 June 1997, pp. 762–768.

[12] T. Kato, T. Kurita, H. Shimogaki, Intelligent visual interaction with image database systems—toward the multimedia personal interface, Journal of Information Processing 14 (2) (1991) 134–143.

[13] I. Kunttu, L. Lepistö, J. Rauhamaa, A. Visa, Image correlogram in image database indexing and retrieval, in: Proceedings of 4th European Workshop on Image Analysis for Multimedia Interactive Services, London, UK, April 9–11, 2003, pp. 88–91.

[14] H.Y. Lee, H.K. Lee, Y.H. Ha, Spatial color descriptor for image retrieval and video segmentation, IEEE Transaction on Multimedia 5 (3) (2003) 358–367.

[15] S.-J. Lee, Y.-H. Lee, H. Ahn, S.-B. Rhee, Color image descriptor using wavelet correlogram, in: The 23rd International Technology Conference on Circuits/Systems, Computers and Communications (ITC–CSCC), 2008, pp. 1613–1616.

[16] J. Li, W. Wu, T. Wang, Y. Zhang, One step beyond histograms: Image representation using Markov stationary features, in: Computer Vision and Pattern Recognition IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, 23–28 June 2008, pp. 1–8.

[17] J. Luo, D. Crandall, Color object detection using spatial-color joint probability functions, IEEE Transaction on Image Processing 15 (6) (2006) 1443–1453.

[18] H.A. Moghaddam and M. Saadatmand-Tarzjan, Gabor wavelet correlogram algorithm for image indexing and retrieval, in: Proceedings of the 18th International Conference on Pattern Recognition, (ICPR), vol. 02, August 20–24, 2006, pp. 925–928.

[19] MUVIS. <http://muvis.cs.tut.fi/>.

[20] W.Y. Ma, H.J. Zhang, Benchmarking of image features for content-based retrieval, in: Proceedings Conferences Signals, Systems and Computers, 1998, pp. 253–257.

[21] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceeding of Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–296.

[22] B.S. Manjunath, J.-R. Ohm, V.V. Vasudevan, A. Yamada, Color and texture descriptors, IEEE Transaction on Circuits and Systems for Video Technology 11 (June) (2001) 703–715.

[23] A. Mojsilovic, J. Kovacevic, J. Hu, R.J. Safranek, K. Ganapathy, Matching and retrieval based on the vocabulary and grammar of color patterns, IEEE Transaction on Image Processing 9 (1) (2000) 38–54.

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

[25] A. Nagasaka, Y. Tanaka, Automatic video indexing and full video search for objects, in: Visual Database Systems II, IFIP, 1992, pp. 113–127.

[26] V. Ogle, M. Stonebraker, Chabot, retrieval from a relational database of images, IEEE Computer 28 (9) (1995) 40–48.

[27] T. Ojala, M. Rautiainen, E. Matinmikko, M. Aittola, Semantic image retrieval with HSV correlograms, in: Proceedings 12th Scandinavian Conference on Image Analysis, Bergen, Norway, 2001, pp. 621–627.

[28] B.C. Ooi, K.L. Tan, T.S. Chua, W. Hsu, Fast image retrieval using color spatial information, The VLDB Journal 7 (2) (1998) 115–128.

[29] M. Partio, B. Cramariuc, M. Gabbouj, A. Visa, Rock texture retrieval using gray level co-occurrence matrix, in: Proceedings of 5th Nordic Signal Processing Symposium, October 2002.

[30] Mari Partio, Bogdan Cramariuc, Moncef Gabbouj, An ordinal co-occurrence matrix framework for texture retrieval, in: Proceedings of EURASIP Journal on Image and Video Processing, vol. 2007, Article ID 17358, 2007, p. 15.

[31] G. Pass, R. Zabih, J. Miler, Comparing images using color coherence vectors, in: Proceedings of the ACM multimedia'96, Boston, November 1996, pp. 65–72.

[32] A. Pentland, R.W. Picard, S. Sclaroff, Photobook: tools for content based manipulation of image databases, in: Proceedings of SPIE (Storage and Retrieval for Image and Video Databases II), vol. 2185, 1994, pp. 34–37.

[33] L.-M. Po, K.-M. Wong, A new palette histogram similarity measure for MPEG-7 dominant color descriptor, in: Proceedings International Conference on Image Processing, ICIP 2004, 2004, pp. 1533–1536.

[34] B. Rogowitz, T. Frese, J. Smith, C.A. Bouman, E. Kalin, Perceptual image similarity experiments, Proceedings of SPIE 3299 (1997) 576–590.

[35] K. van de Sande, T. Gevers, C. Snoek, Evaluation of color descriptors for object and scene recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, 23–28 June 2008, pp.1–8.

[36] S. Sclaroff, L. Taycher, M. La Cascia, Image-Rover: a content-based image browser for the world wide web, in: Proceedings of IEEE Workshop on Content-based Access Image and Video Libraries, Puerto Rico, June 1997, pp. 2–9.

[37] J.R. Smith, S.F. Chang, Single color extraction and image query, in: Proceedings of ICIP, vol. 3, October 1995, pp. 528–531.

[38] J.R. Smith, S.F. Chang, VisualSEEk: a fully automated content-based image query system, in: Proceedings of ACM Multimedia, Boston, November 1996, pp. 87–98.

[39] M. Stricker, M. Orengo, Similarity of color images, in: Proceedings SPIE, 1995, pp. 381–392.

[40] M.V. Sudhamani, C.R. Venugopal, Grouping and indexing color features for efficient image retrieval, International. Journal of Applied Mathematics and Computer Sciences 4 (3) (2007) 150–155.

[41] M.J. Swain, D.H. Ballard, Color indexing, International Journal of Computer Vision 7 (1) (1991) 11–32.

[42] C. Tomasi, R. Manduchi, Bilateral filtering for gray and color images, in: Proceedings of the Sixth International Conference on Computer Vision, Bombay, India, January 1998.

[43] L.V. Tran, R. Lenz, Compact colour descriptors for colour-based image retrieval, Signal Processing 85 (February) (2005) 233–246.

[44] A. Utenpattanant, O. Chitsobhuk, Image retrieval using hair color descriptor incorporating with pruning techniques, in: Proceedings of the 9th International Conference on Advanced Communication Technology, February 2007, pp. 1123–1126.

[45] I. Valova, B. Rachev, Retrieval by color features in image databases, in: European Conference on Advances in Databases and Information Systems (ADBIS 2004), Budapest, Hungary, September 2004.

[46] Virage. <www.virage.com>.

[47] S. Wang, L.T. Chia, D. Rajan, Image retrieval using dominant color descriptor, in: Conference on Imaging Science, Systems and Technology (CISST 2003), Las Vegas, USA, June 2003, pp. 107–110.

[48] M. Wertheimer, Laws of organization in perceptual forms, partial translation, in: W.B. Ellis (Ed.), A Sourcebook of Gestalt Psychology, Brace and Company, Harcourt, NY, 1938, pp. 71–88.

[49] K.M. Wong, L.M. Po, K.W. Cheung, A compact and efficient color descriptor for image retrieval, in: Proceedings of IEEE International Conference on Multimedia and Expo (ICME 2007), Beijing, China, July 2007, pp. 611–614.

Publication 2

M. Birinci and K. Ugur, “Interactive Image Segmentation Based on Superpixel Grouping for Mobile Devices with Touchscreen,” In Proceedings of IEEE International Conference on Multimedia & Expo (ICME), Chengdu, 2014, pp. 1-6.

INTERACTIVE IMAGE SEGMENTATION BASED ON SUPERPIXEL GROUPING FOR MOBILE DEVICES WITH TOUCHSCREEN

Murat Birinci
Tampere University of Technology
murat.birinci@tut.fi

Kemal Ugur
Nokia Research Center
kemal.ugur@nokia.com

ABSTRACT

This paper proposes a novel method for semi-supervised image segmentation that is particularly targeted for interaction on mobile devices with touchscreen. In order to extract an object from a complicated scene with minimal user input, superpixels are first grouped into overlapping regions based on their visual similarity and the groups belonging to the object are then selected via user scribbles. By moving the user interaction to the end of the whole process, users' idle time is minimized hence an engaging user experience is achieved. The proposed method can effortlessly handle imprecise and inaccurate scribbles and is inherently suitable for touchscreen devices since it eliminates the necessity of a separate brush for marking the background as majority of state-of-the-art methods do. A novel fine-tuning method is also proposed where both foreground and background corrections are possible without entailing the user to change the brush. Experimental results prove that successful results are achieved with a slick and pleasant user experience.

Index Terms — Interactive image segmentation, user interaction, user experience, superpixel, region merging, Gestalt, perceptual grouping

1. INTRODUCTION

Breaking up images into meaningful objects, often referred as image segmentation, has been at the core of many computer vision and computational photography tasks since it enables understanding the semantics of the image. Understanding such semantics facilitates further applications such as image editing, image enhancement, image manipulation, object detection, object tracking, content based image retrieval, medical image processing etc. Even the basic discrimination of foreground (FG) and background (BG) in a visual scene significantly assists such applications. However, considering the variety of the context and applications where segmentation is required, definition of such FG and BG becomes rather ambiguous – turning fully automatic image segmentation into an ill-posed problem without any semantic knowledge of the target object [1]. Semi-supervised image segmentation, on the other hand, aims to overcome such difficulties by taking advantage of user input for assistance. Typically user specifies FG and BG objects, or at least gives hints about them. However, currently no successful algorithm exists that can produce satisfactory results for any image with few imprecise user inputs. Thus, the user is usually expected to further fine-tune the result for several iterations via precise scribbles around and within the object to be segmented. Such interactions can be exhausting for the user, especially if they need to be performed several times. Moreover, additional complications arise in case of touch devices where the brush size is determined by users' finger which usually lacks the precision required for such interactions particularly on smaller screens.

Graphcut [2] is one of the most popular and distinctive semi-supervised segmentation method, where the user is asked to mark FG and BG regions via scribbles and an energy minimization algorithm is then applied in order to separate the graph (image) into two regions, i.e. FG and BG. Whereas numerous variants of Graphcut have been proposed, Grabcut [3] stands out not only due to its improvement in performance, but also because of the simplification it delivers in user interaction. Instead of providing separate scribbles for FG and BG, the user is only asked to mark the object of interest by drawing a bounding box. Even though such interaction is undeniably more desirable than providing separate scribbles, performance of the algorithm is far from satisfactory after the initial interaction. Hence the user is again expected to fine-tune the segmentation result via FG/BG scribbles. In [4], authors proposed a modified Graphcut with a much more intuitive, progressive painting interaction tool. The user simply scribbles the FG and selection automatically snaps to object boundaries. However, the proposed "multi-core Graphcut" comes with a loss in accuracy. A similar interaction is proposed in [5], where the selection (i.e. segmentation mask) is updated dynamically as the user scribbles the FG. They used dynamic and iterative Graphcut that works on superpixels and updates the segmentation mask locally, i.e. in close vicinity of the user input. Similar to [3], the user interactions in both [4] and [5] are clearly less burdensome and more appealing compared to regular Graphcut style interaction that involves FG and BG scribbles. Additionally, in [5] authors performed a user study in order to evaluate the user interaction in terms of easiness and entertainment and obtained better results than Grabcut and Intelligent Scissors [6]. A noteworthy feature of [5] is its error tolerant nature that allows it to work even with inaccurate user scribbles. If a recently scribbled superpixel is dissimilar to the object color model, the algorithm expects the next scribbled superpixel to be similar to that superpixel. If not, it is assumed to be accidentally scribbled and hence ignored. However such a control mechanism requires large enough superpixels relative to scribble size, which in return might affect the boundary adherence of the segmentation. Otherwise it can handle only very minor scribble errors. Moreover, the utilized Graphcut based approach brings in a laggy user interaction due to the relatively heavy computational costs for smooth real-time operations.

Gestalt psychology suggests that perception is not simply the concatenation of one's senses, but is a result of perceptual organization [7]. Such organization is explained via *laws of grouping* among percepts based on their mutual properties; such as proximity, similarity etc. The idea of superpixels is a fitting example to demonstrate such grouping where individual percepts, i.e. pixels, are grouped into more perceptual clusters. They also serve as better primitives than pixels by getting rid of the pixel-level image redundancies [8]. However, superpixels simply represent an intermediate state. Even though there are numerous approaches that utilize superpixels for a mere speedup, they have also been used as the building blocks of many algorithms. For instance region merging algorithms follow the same Gestalt philosophy and try to obtain objects by grouping superpixels. In [9], Peng et al. proposed a fully automatic segmentation algorithm via region merging and proposed solutions to two major problems of such algorithms, namely merging order and stopping criteria, via sequential probability ratio test and the minimal cost criteria. It is also claimed that the evolution of the regions follow Gestalt laws of perception. Another solution to these problems is proposed in [10] by incorporating user interaction. Users are asked to mark FG and BG regions (as in Graphcut) and the algorithm iteratively merges superpixels based on their color histograms. The merging process starts from the user marked superpixels and continues until none of the neighboring regions meet the merging criteria. A noteworthy property of the algorithm is that it doesn’t require any preset thresholds. A superpixel is merged to its neighbor only if it is the most similar neighbor to its neighbor. They compared their results with both pixel-based and superpixel-based Graphcut and achieved significantly better performance in terms of accuracy. However, the fact that a superpixel is merged with only one neighbor at each iteration introduces a significant slowdown, hence the algorithm is even slower than pixel based Graphcut. This shortcoming is pointed out in [11] and authors proposed another merging algorithm where any two adjacent superpixels with *k-global maximal similarity* (i.e., one of the *k* most similar pairs among all possible pairs of adjacent superpixels) could be merged with each other if they are the same types of superpixels (marked as FG, marked as BG or unmarked) or either of them is an unmarked superpixel. Also one superpixel can merge with several superpixels in an iteration as long as their similarity scores are among the top *k* ones of all the scores. They achieved on-par performance with [10] with a significant boost in computation speed. However the user still has several seconds of idle time after providing the scribbles. It should be noted that, in practice, several iterations of user interaction is typically required in order to obtain the desired output. Therefore such repeated idle intermissions can be frustrating for the users.

Almost all interactive segmentation methods take user input as the starting point and build the whole algorithm on top of it. The algorithm simply re-runs if further interaction is required (which, in most practical cases, is inevitable). Moreover, most interactive segmentation methods take the user input as the absolute truth (i.e. consider the labels given via user input are absolutely correct) and leave the user with little error space. However, in practice, particularly on touchscreen devices such as mobile phones where the screen is relatively small and the input is given with finger strokes, expecting such accuracy from the user can easily create erroneous results. Another restriction touchscreen devices impose is that providing separate FG and BG scribbles require bothersome menu operations weakening the user experience. In this paper we address all aforementioned issues and present a pleasant user experience with a proficient performance. User interaction is moved to the very end of the process so that the user does not wait idly after the interaction until the segmentation mask is updated. The user simply scribbles over the object of interest as in [5] and the segmentation mask is updated on-the-go as the user continues to scribble. All required information is gathered via FG scribbles negating any BG scribble. The algorithm is also capable of handling noticeable user errors. In order to achieve this, the method takes an initially over-segmented image (i.e. superpixels) as input and creates overlapping regions abiding by Gestalt laws of perception. Then the user merely selects the regions to be included in the object mask via simple scribbles. Experimental results compare the proposed method to state-of-the-art methods and prove that high performance is achieved with an intuitive and simple user interaction providing an outstanding user experience.

The rest of the paper is organized as follows. Section 2 details the proposed method. Section 3 presents experimental results proving the proficiency of the algorithm. Section 4 concludes the paper.

2. PROPOSED METHOD

Stemming from Gestalt laws of perception, we propose a region merging algorithm aspiring to provide “better”, i.e. more perceptual, cues for interactive segmentation. The proposed method uses an over-segmented image (i.e. superpixels) as input and groups those superpixels based on their proximity and visual similarity. However, FG and BG may have similar visual properties locally or globally. Hence a straightforward merging may result easily in erroneous segmentation masks. In order to tackle such incidences, the proposed method merges superpixels into overlapping regions where each region is treated as an alternative way of grouping superpixels, i.e. *hypothesis*. The selection of object regions among these multiple hypotheses is accomplished via user input (see Section 2.2).

2.1. Hypotheses Creation

The algorithm takes an over-segmented image as input (i.e. superpixels) and creates a hypothesis $\{H_i\}_{i=1,2,...,N}$ for every superpixel $\{S_i\}_{i=1,2,...,N}$ in the image. In order to achieve this, each S_i is compared to its neighbors $\{S_i^j\}_{j=1,2,...,R}$. If S_i and S_i^j are visually similar, S_i^j is included in the hypothesis H_i. Then for each $S_i^j \in H_i$, a set $\{S_i^{j,k}\}_{k=1,2,...,K}$ is formed from those S_i^j’s neighbors. Each $S_i^{j,k}$ is compared to S_i and if S_i and $S_i^{j,k}$ are visually similar, $S_i^{j,k}$ is included in the hypothesis H_i. Then, a new set is formed from the neighbors of $S_i^{j,k} \in H_i$ and the algorithm continues until the stopping criteria is met (Note here that each superpixel may have different number of neighbors, i.e. *R* and *K* might be different for each superpixel). Even though the information within H_i is kept local by comparing the neighboring superpixels always to the initial superpixel (i.e. $S_i^{j,k,...}$’s to S_i), the stopping criteria also acts as a buffer to prevent any possible error to propagate. Let $S(x_c, y_c)$ denote the center of mass of a superpixel S. Then, each $S_i^{j,k,...} \in H_i$ must satisfy:

$$d_1\left(S_i(x_c, y_c)\,,\, S_i^{j,k,...}(x_c, y_c)\right) < T_R \tag{1}$$

where $d_1(p\ ,q)$ denotes the L_1 distance between the points *p* and *q*. The decision on T_R should be defined in accordance with the user interaction and will be further discussed in Section 2.2.

Another key decision is the visual similarity judgment between two superpixels. In order to evaluate such similarity, those regions first need to be represented via some descriptors. Whereas one can use any visual feature such as color, texture, edge etc., we preferred color histograms for their proven efficacy and simplicity. Moreover, the potency of texture and edge features diminish as the size of the region they are extracted gets smaller. Hence, they are not suitable for small superpixels. We used perceptually uniform *L*a*b* color space and uniformly quantized each histogram into 8×8×8=512 bins. In the end, each superpixel is represented by a 512 bin normalized *L*a*b* histogram. There are many possible ways to compare two histograms such as Euclidean distance (i.e. L_2), Bhattacharyya coefficient [12], Histogram Intersection [13] and Earth Mover's Distance (EMD) [14]. We preferred EMD in our experiments for its significantly better performance; however, Bhattacharyya coefficient also provides satisfactory results and is relatively faster compared to EMD. Let the distance between two histograms be denoted as $dist(Hist_1, Hist_2)$. Then, two superpixels are considered as visually similar if their histograms satisfy:

$$dist(Hist_1, Hist_2) < T_{hist} \tag{2}$$

Then, the whole process for hypotheses creation can be considered as a function that takes all image superpixels as input and outputs the hypotheses:

```
CreateHypotheses( $\{S_i\}_{i=1,2,\dots,N}$ ):
for every superpixel $S_i$
  create a hypothesis $H_i$
  add $S_i$ to $H_i$ ($S_i \rightarrow H_i$)
  CheckNeighbors( $S_i$, $S_i$, $H_i$ )
end
return $\{H_i\}_{i=1,2,\dots,N}$
```

where the function *CheckNeighbors* is defined as:

```
CheckNeighbors( S, C, H ):
for every neighbor $N$ of $S$
  if $N$ is similar to $C$ ($N \approx C$) and $d_1(N,C) < T_R$
    add $N$ to $H$ ($N \rightarrow H$)
    CheckNeighbors( $N$, $C$, $H$ )
  end
end
```

2.2. Hypotheses Selection

In order to select the set of hypotheses that form the FG object(s), the algorithm resorts to user input. In terms of user interaction, on the surface, the proposed method is highly similar to [4] and [5]; such that the user draws simple FG scribbles and the segmentation mask is updated accordingly. However, unlike typical abstract scribbles, user provides continuous scribbles as if s/he is painting over the FG object and the segmentation mask is updated instantly. Fig. 1 shows how the algorithm proceeds as the user keeps scribbling where the top row illustrates what is visible to the user and the bottom row shows the input taken by the algorithm for hypotheses selection. The user is first presented with a dimmed version of the original image. As the user scribbles over the image, the selected hypotheses are included in the segmentation mask where the mask is visualized as the original, i.e. brighter, image is being revealed together with a border around it for better perception. A hypothesis is considered as "selected" if it is covered, i.e. painted over, by the user scribble more than a certain ratio. This is controlled by the parameter σ, which consequently determines the smoothness of the user interaction. Let α be defined as:

$$\alpha = \frac{A_S}{A_H} \tag{3}$$

where A_H is the area of the entire hypothesis and A_S is the area of the hypothesis covered by the scribble. Then a hypothesis is regarded as "selected" if $\alpha > \sigma$. If σ is too low (~0.1), so little user input can be enough to reveal the object. However, the algorithm will be more prone to errors. If, on the other hand, σ is too high (~0.9), the algorithm will be more robust but it will require more scribbles to reveal the object. Hence the interaction will be more demanding, burdensome and unpleasant. Note here that even though the amount of work appears to be more than providing typical abstract scribbles, by avoiding a turn based interaction where the user waits idly while the algorithm executes, a more engaging and pleasant interaction is achieved. Such user interaction has already been proved to be easier and more entertaining than other single brush interactive segmentation tools such as Grabcut and intelligent scissors [5].

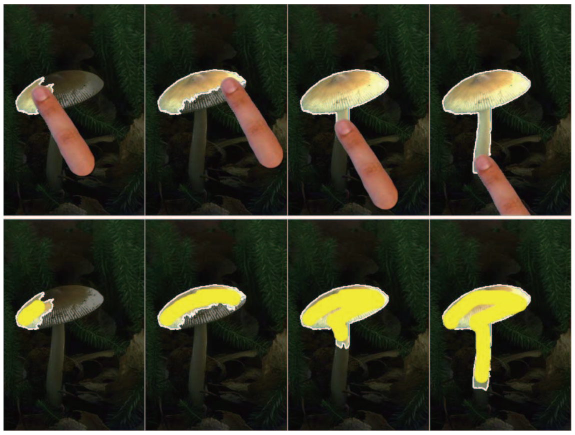


Fig. 1. *Top row:* User interaction of the proposed algorithm from user's point of view (progresses left to right). *Bottom row:* User scribble used by the algorithm for hypotheses selection (invisible to the user).

In order to provide a slick and appealing user experience, the selected region should be large enough to be able to snap to object boundaries that are further away from the user scribble. As discussed in Section 2.1, this is defined by the relation between the maximum size of a hypothesis, i.e. T_R, and the scribble size (the scribble can be considered as a circle dragged around the image, whereupon the size of the scribble refers to the radius of that circle r_B). Since the brush represents the finger of the user, the size of the brush r_B should depend on the screen size of the device. Whereas on mobile phones the area covered by a finger would be an accurate choice, on larger displays such as tablets, that area can be too small compared to the screen size; hence a larger r_B can be used.

Fig. 2 shows a more severe case where the user provided even more erratic scribbles. Boundary of the final segmentation mask is highlighted in Fig. 2b. Cases where the scribble is larger or smaller than the target object (or part of the object) may easily be encountered. Overlapping hypotheses for local sections of the image are shown in Fig. 2c and Fig. 2d. Note how several hypotheses cover the same part of the object and how the algorithm successfully selects the appropriate ones.

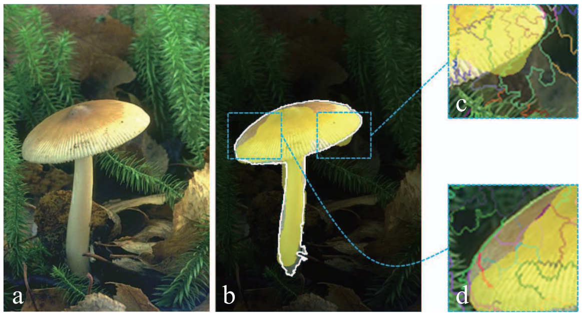


Fig. 2. Error tolerance of the proposed algorithm. Hypotheses for which $\alpha \le \sigma$ are excluded from the segmentation mask.

2.3. Fine Tuning

Despite the efficacy of the proposed algorithm, it is still possible to encounter erroneous results due to several reasons, such as complicated background where small background objects or object parts (segments) appear next to the object border. Since such small segments would form hypotheses themselves, they can easily be painted by the user scribble and included in the segmentation mask. Fig. 3 demonstrates such a case where small pebbles appear next to the snake (i.e. FG object) that are covered by the user scribble, hence included in the segmentation mask. Even though it is unlikely, it is also possible that a FG and BG superpixel have very similar color histograms and grouped together into a hypothesis. Whereas such occurrences are seldom encountered, they should also be handled for a well-founded segmentation algorithm.

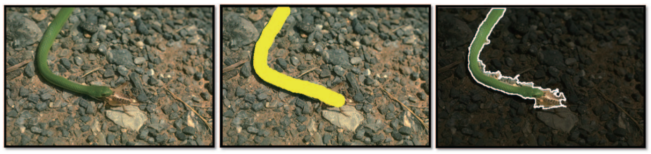
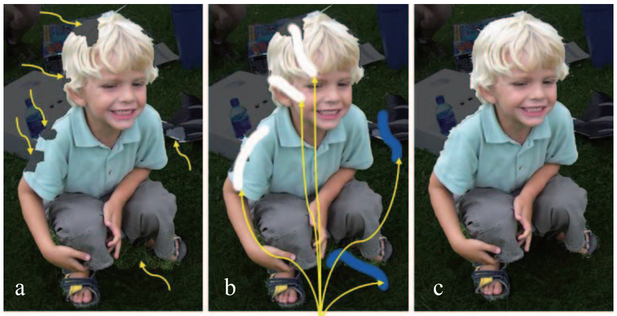


Fig. 3. Erroneous segmentation due to complicated background.

In order to enable correcting such errors, we propose fine-tuning as a final stage of the algorithm. In principle, this step is algorithmically no different than hypotheses selection except that here, each superpixel $\{S_i\}_{i=1,2,...,N}$ is treated as a hypothesis $\{H_i = S_i, \forall i = 1,2,..., N\}$. In other words, the user simply paints over the initial superpixels instead of (relatively large) hypotheses which in return allows to obtain a more precise segmentation mask. Exactly the same algorithm is followed as the hypotheses selection. However, unlike the hypotheses selection stage where the user was selecting only FG regions, in fine-tuning stage, it is possible to select a superpixel as both FG and BG so that any preceding error can be corrected. As discussed in Section 1 switching repeatedly between FG and BG brushes can be inconvenient for the user, particularly on touchscreen devices, which in return has a significant negative impact on user experience. Therefore we propose a novel technique for conveniently switching between FG and BG brushes where the type of the brush is decided based on the location of the initial interaction. In other words, the scribble "paints" the superpixels as FG if the user starts "painting" from a FG region, and vice versa. Fig. 4 shows an example of fine-tuning process where several regions were incorrectly classified as FG or BG (Fig. 4a). Note how FG scribbles initiate from a FG region and BG scribbles initiate from a BG region (Fig. 4b).



Scribbles start from this end

Fig. 4. (a) Segmentation errors, (b) FG (white) and BG (blue) fine-tuning scribbles, (c) final segmentation mask.

The proposed fine-tuning method handles any user or algorithm related error and it seamlessly blends in to the proposed segmentation method. The essence of the user interaction is still the same, i.e. the user still "paints" over the object, without any necessity to select different brushes. One difference is that in fine-tuning, since there are two separate automatically altering brushes, user scribbles are also visible to the user to improve awareness, i.e. so that the user knows which brush (FG or BG) s\he is using. However, this can easily be left as a design decision as it does not in any way affect the algorithm's operation. Since no additional computation is done between hypotheses selection and fine-tuning, there is no idle waiting time at this stage either. Therefore the user experiences the whole process (hypotheses selection + fine-tuning) within the flow of the algorithm providing a pleasing user experience.

3. EXPERIMENTAL RESULTS

The proposed algorithm is tested on several images from [10] and [15] and compared to the algorithms in [5], [10] and [11]. We believe some images are too small to facilitate interactive segmentation, yet still included in the dataset for complete comparison with competing algorithms. SLIC superpixels [7] are used for initial over-segmentation due to their high performance in boundary adherence and computational speed. 1000 SLIC superpixels are extracted from each image with a compactness factor of 20 for both [5] and the proposed algorithm. Parameters for SLIC are selected empirically. Whereas the proposed method performs better with higher number of superpixels (i.e. ~2000-

3000), a trade-off with the computation time has to be made. r_B is set to 5% of *min*(*image height*, *image width*) to mimic user finger area on a mobile phone and $T_R = 2 \times r_B$. EMD is used for histogram comparison with $T_{Hist} = 0.15$ and finally the smoothness parameter is set to $\sigma = 0.6$. Fig. 5 shows several segmentation results for the proposed and competing methods.

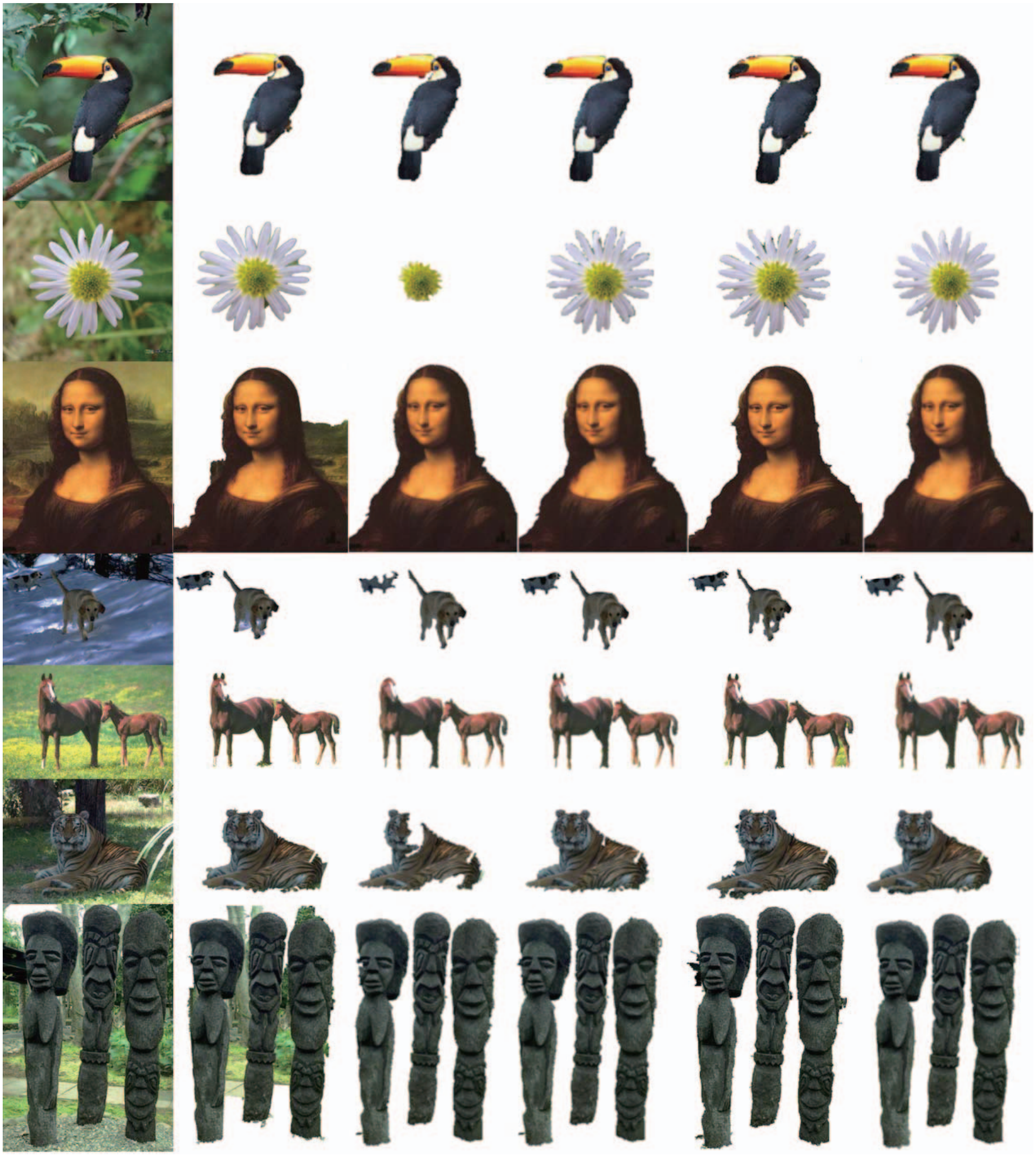


Fig. 5. Experimental results for (top to bottom) bird, flower, monalisa, dogs, horses, tiger and sculptures. *Left to right*: Original image, Coloring [5], MSRM [10], KSRM [11], proposed, ground truth.

Some examples in Fig. 5 also contain several minor errors as discussed in Section 2.3. For instance legs of the horse are too thin compared to user scribble, hence part of the BG is also included in the segmentation mask. A similar error can also be seen in sculptures due to local similarities between FG and BG. Note that such minor errors can easily be handled by the proposed fine-tuning method (see Fig. 6). Even though fine-tuning is possible for any interactive segmentation method, mostly the algorithm is re-run and the result is updated. However the proposed method enables fine-tuning within the flow of the application without re-running the algorithm or introducing any further idle waiting time.

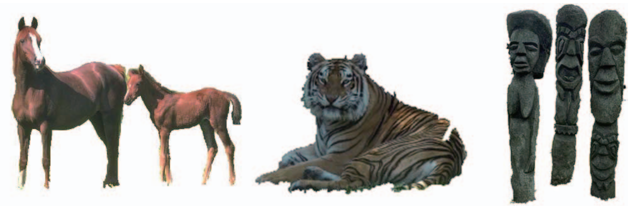


Fig. 6. Fine-Tuned results for horses, tiger and sculptures.

Table 1 shows the computation times for the methods in [10], [11] and the proposed method on images in Fig. 5. Computation time for [5] could not be included in the table since the algorithm performs the computation throughout the user interaction. Yet, it should be noted that the utilized Graphcut based approach induces noticeable lags during the interaction. Computation times for [10] and [11] are taken from [11] without any loss of accuracy since our systems are near identical. Note that the reported times for the proposed method are spent on hypotheses creation – which is an offline operation – and not reflected to the user, i.e. the user does not have to wait idly during the reported times unlike [10] and [11] or experience a laggy interaction as in [5].

Table 1. Computation Times for the tested images in Fig. 5.

Time(sec.)	MSRM [10]	KSRM [11]	Proposed
Bird	2.68	0.53	2.63
Flower	4.13	1.15	1.84
MonaLisa	13.21	3.49	1.64
Dogs	5.01	1.14	1.85
Horses	22.43	2.28	2.37
Tiger	8.49	2.14	1.94
Sculptures	33.01	5.44	1.95

The reader is referred to [5] for the evaluation of the user interaction compared to other interactive segmentation methods based on single brush.

4. CONCLUSION

An interactive segmentation method is proposed essentially targeted to improve user experience on touchscreen devices. The method achieves on-par performance with state-of-the-art methods, yet is distinguished in several ways: First, a single FG brush is used to avoid inconvenient menu operations to alternate between separate brushes. Second, by performing all time consuming operations before the user interaction, user's idle waiting time is eliminated. Such operations can easily be handled while loading the image or even before, depending on the application. Third, the method can handle significant amount of user error. Such capability is particularly eminent for touchscreen devices with small displays and inaccurate input brushes (i.e. user finger). Finally, a novel fine-tuning method is proposed where both FG and BG corrections are enabled with automatically altering brushes in an intuitive manner. Such features are achieved by proficiently incorporating human perceptual rules and considering "the user" as the upmost concern of the entire design.

5. REFERENCES

[1] D. Béréziat and I. Herlin, "Solving Ill-Posed Image Processing Problems Using Data Assimilation," *Numerical Algorithms*, vol. 2, no. 2, pp. 219–252, 2011.

[2] Y. Boykov, M.-P. Jolly, "Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images," *IEEE Int. Conf. on Computer Vision (ICCV)*, vol. 1, pp. 105–112, 2001.

[3] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive Foreground Extraction Using Iterated Graph Cuts," *in ACM SIGGRAPH*, 2004.

[4] J. Liu, J. Sun, and H. Shum, "Paint Selection," *In ACM SIGGRAPH*, 2009.

[5] O. Sener, K. Ugur, and A. Alatan, "Robust Interactive Segmentation via Coloring," *In ACM VIGTA*, 2012.

[6] E. N. Mortensen and W. a. Barrett, "Intelligent Scissors for Image Composition," *In ACM SIGGRAPH*, 1995.

[7] S. E. Palmer, "Vision Science: Photons to Phenomenology", *MIT Press*, 1999.

[8] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC Superpixels Compared to State-of-the-art Superpixel Methods.," *IEEE Trans. on Pattern Analysis and Machine Int.*, vol. 6, no. 1, pp. 1–8, 2012.

[9] B. Peng, L. Zhang, and D. Zhang, "Automatic Image Segmentation by Dynamic Region Merging," *Image Proc. IEEE Trans.*, vol. 20, no. 12, pp. 3592–3605, 2011.

[10] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Interactive Image Segmentation by Maximal Similarity Based Region Merging," *Pattern Recognition*, vol. 43, no. 2, pp. 445–456, 2010.

[11] T. Li, Z. Xie, J. Wu, J. Yan, and L. Shen, "Interactive Object Extraction by Merging Regions with K-Global Maximal Similarity," *Neurocomputing*, vol. 120, pp. 610–623, November 2013.

[12] T. Kailath, "The Divergence and Bhattacharyya Distance Measures in Signal Selection," *IEEE Trans. Comm. Technology*, vol. 15, no. 1, pp. 52–60, Feb. 1967.

[13] M. J. Swain and D. H. Ballard, "Color Indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[14] C.T.Y. Rubner, L.J. Guibas, "A Metric for Distributions with Applications to Image Database," *IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 59–66, 1998.

[15] D. Martin, C. Fowlkes, D. Tal and J. Malik, "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics," *IEEE Int. Conf. on Computer Vision (ICCV)*, vol.2, pp.416–423, 2001.

Publication 3

M. Birinci, F.D. Maria, G. Abdollahian, E.J.Delp, M. Gabbouj “Neighborhood Matching for Object Recognition Algorithms Based on Local Image Features,” In Proceedings of Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE), Sedona, AZ, 2011, pp. 157-162.

Publication 4

G. Abdollahian, M. Birinci, F.D. Maria, M. Gabbouj and E.J .Delp “A Region-Dependent Image Matching Method for Image and Video Annotation,” 9th International Workshop on Content-Based Multimedia Indexing (CBMI), Madrid, 2011, pp. 121-126.

Publication 5

M. Birinci and S. Kiranyaz, “A Perceptual Scheme for Fully Automatic Video Shot Boundary Detection,” Signal Processing: Image Communication, vol. 29, pp. 410-423, 2014.

Signal Processing: *Image Communication* 29 (2014) 410–423

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

A perceptual scheme for fully automatic video shot boundary detection☆

Murat Birinci*, Serkan Kiranyaz

Department of Signal Processing, Tampere University of Technology, Finland

ARTICLE INFO

Article history:
Received 22 April 2013
Received in revised form
18 December 2013
Accepted 18 December 2013
Available online 27 December 2013

Keywords:
Video shot boundary detection
Human perception
Local image features
Video content analysis

ABSTRACT

In this paper, we propose a novel and robust modus operandi for fast and accurate shot boundary detection where the whole design philosophy is based on human perceptual rules and the well-known "Information Seeking Mantra". By adopting a top–down approach, redundant video processing is avoided and furthermore elegant shot boundary detection accuracy is obtained under significantly low computational costs. Objects within shots are detected via local image features and used for revealing visual discontinuities among shots. The proposed method can be used for detecting all types of gradual transitions as well as abrupt changes. Another important feature is that the proposed method is fully generic, which can be applied to any video content without requiring any training or tuning in advance. Furthermore, it allows a user interaction to direct the SBD process to the user's "Region of Interest" or to stop it once satisfactory results are obtained. Experimental results demonstrate that the proposed algorithm achieves superior computational times compared to the state-of-art methods without sacrificing performance.

1. Introduction

The amount of available video content is growing exponentially with the development in content creation technology. Moreover, content sharing has become immensely popular, enabling every individual to access a vast amount of video content. YouTube, globally the 3rd most popular website [1], announced that more than 72 h of video are uploaded to the website every minute, and more than 4 billion h of video are watched every month [2]. It is therefore inevitable that such amount of visual information and growth demands efficient content management tools.

In [3], Thompson et al. defined a video shot as the smallest unit of visual information captured at one time by a camera that shows a certain action or event. Therefore, in order to capture the entire visual content properly and attain a complete grasp of the video, shot detection is a fundamental step of content based video analysis. Whereas there is a wealth of research on shot boundary detection (SBD), the main bottleneck of the problem is the relative difficulty in detecting gradual transitions (GT) between shots compared to the detection of the abrupt changes, i.e. abrupt cuts (AC). Even though gradual transitions used to appear more frequently in professionally edited videos, nowadays even personal cameras and camera-equipped cell phones are capable of editing videos to comprise such transitions. Therefore, a proficient SBD algorithm should be able to handle gradual shot transitions regardless of their nature (dissolve, fade, wipe etc.), as well as abrupt changes. Whereas, any SBD algorithm stems from the same assumption that there is a visual discontinuity between consecutive shots, most of them

☆ This work was supported in part by the Doctoral Program of Tampere University of Technology's President.

* Corresponding author.
E-mail addresses: murat.birinci@tut.fi (M. Birinci), serkan@cs.tut.fi (S. Kiranyaz).

0923-5965/$ - see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.image.2013.12.003

suffer from performance, computational cost and sometimes even both.

Gargi et al. [4] presented a performance analysis for several color histogram based SBD algorithms, where shot boundaries are detected via computing the histogram differences of consecutive frames in various color spaces using various difference measures. They concluded that the histogram intersection method [5] performed the best; however, their evaluation did not cover GT detection. They further analyzed different compressed domain algorithms that utilize compressed domain features such as DCT coefficients and motion vectors [6–12]. However, they concluded that, despite being computationally efficient, their performance levels were even below histogram based approaches. In a more recent study, Teng [13] proposed a method based on texture features extracted from non-overlapping blocks, and classified video frames via Support Vector Machines (SVM) to detect shot boundaries based on cosine distances. Another classifier based method is presented in [14] utilizing the U component of the YUV histogram and classifying the difference curves using Particle Swarm Optimization (PSO). In [15], Hanjalic provided a thorough analysis of the previous methods and proposed a probabilistic method based on YUV color components from non-overlapping blocks. The method provides satisfactory results for AC and dissolves; however, it requires a specific implementation for each individual type of GTs. Another extensive evaluation came as a result of TRECVid, which had an activity track for SBD from 2001 to 2007 joining 57 different research groups in order to determine the best approaches [16]. Whereas the idea of various algorithms working on a common dataset with common scoring metrics provides the means for objective evaluation, it also brings in a clear advantage for machine learning algorithms since the whole TRECVid dataset (development+testing) is composed of vastly similar content which in return inevitably bias the overall results. The fact that 9 out of top 10 performing groups utilize machine learning algorithms is a clear indication of such bias where the algorithms are specifically tuned for the development data which is highly similar to the test data. Moreover, the fact that 6 out of top 10 performing algorithms using flash detectors, which is a very specific case commonly appearing in news videos (which also constitutes most of the TRECVid dataset), is also a clear indication that the competing methods were tuned to perform only for the specific TRECVid dataset and seriously questions the applicability of such methods to generic video content.

In [17], Boccignone et al. proposed a perceptual stand point to the SBD problem and suggested that “visual attention” is the key to detect scene changes. They extracted the focus of attention (FOA) points from each frame, where the variations in the consistency of FOA revealed shot boundaries. The motivation of the paper, as the authors stated, was not only to achieve high SBD performance, but also to introduce a different angle for the problem. However, their results were still comparable to the state-of-art algorithms. Another high level analysis was proposed by Park et al. [18] where they made use of object recognition techniques, namely Scale Invariant Feature Transform (SIFT) [19]. They proposed that the objects or background do not differ significantly within the same shot, whereas a notable difference occurs across shot boundaries. In order to measure such dissimilarity, they extracted and matched interest points (SIFT) between consecutive frames and monitored the variation in the number of matches in order to detect ACs. However, their method failed to detect GTs since the visual similarity between two consecutive frames is significantly high during a GT, which in return yields a high number of matches. In order to cure this deficiency, the authors additionally compared every *Nth* frame in order to attain sufficiently high content change for GT detection. This method can tell that a shot transition occurred somewhere between those *N* frames, but it still fails to determine its exact location. However, even though the method suffered from the heavy computation of the SIFT that has to be computed for each frame, it can still be regarded as innovative due to its incorporation of objects and object recognition algorithms in order to bring in a higher level standpoint to SBD problem. Moreover, similar to the work in [16] that used FOA in order to extract the essential information throughout the frame, utilization of local image features aims to achieve the same goal by detecting objects through such invariant (to the scale, rotation and translation) points and the features computed over the local regions around them.

There have been several methods concerning local image features prior to SIFT; however, it has been regarded as a milestone due to its remarkably high performance and stability under relatively reasonable computational costs. One of the oldest methods, yet still popular, is the Harris corner detector [20] that is based on the autocorrelation matrix. Whereas being translation and rotation invariant, Harris (corner) points are not scale invariant. The scale invariant version of the Harris detector was proposed by Lindeberg [21], which is also referred as Harris–Laplace detector. Mikolajczyk and Schmid further improved this method to provide an affine invariant detector called Harris–Affine [22]. Lowe in Ref. [19] proposed SIFT, which uses Difference of Gaussians (*DoG*) as an approximation to Laplacian of Gaussians (*LoG*) and their local maxima to detect scale and rotation invariant keypoints. Bay et al. used integral images to detect keypoints in close to real time [23]. Integral images were already known to be used for fast computation of Haar wavelets. However, Bay et al. used those to approximate the Hessian matrix, which they claimed to be more stable and repeatable than Harris-based detectors. There are numerous adaptations and successors of the aforementioned detectors; however, while choosing the appropriate local feature, typically a trade-off has to be made between efficiency on one hand, and accuracy or repeatability on the other. Harding and Robertson [24] compared six keypoint detection methods (namely SIFT, Harris–Laplace, SURF, MSER [25], FAST [26,27] and Kadir–Brady Saliency [28]) with two visual saliency methods and concluded that SURF has the highest correspondence surpassing other methods by 15% higher overlap. For a more comprehensive study on keypoint detectors, the reader is referred to the survey by Tuytelaars et al. [29].

Despite the wealth of research in local image features and their relevance to SBD problem, utilization of these features in SBD has been surprisingly limited. Huang et al. [30] reported that the work proposed in [18] is the first method that employed keypoint-based analysis. They further proposed a parallel approach to [18] where they used a relatively light descriptor of their own design extracted around Harris keypoints, i.e. Contrast Context Histogram (CCH) [31], in order to avoid the computational load of SIFT feature extraction. Additionally, they performed a more in depth analysis of frame similarities in order to detect both ACs and GTs with high accuracy. They initially compare every adjacent frame and observe the variation in the number of matched keypoints, where they take every local minimum as a candidate shot boundary. They assume that the local maxima before and after the candidate transitions are the possible start and end frames of the GT. They further require those local maxima to be followed (and preceded) by a stable number of matches in order to claim it as a shot boundary. This also allows them to determine exact transition intervals. However, this method still considers the frame similarity only between adjacent frames. As we mentioned earlier adjacent frames have significantly high visual similarity, hence such an analysis can easily lead to false negatives or even false positives. In order to avoid such deficiency, they followed a similar strategy to [18] and compared frames that are certain distance apart, namely the frames at the beginning and end of the candidate transition interval. If those frames are found to be similar they regard it as a false alarm, otherwise the final decision is given as a shot boundary. Whereas the authors reported significantly higher accuracy compared to [18], computational cost is an obvious drawback of the algorithm – not because of the underlying feature detector and descriptor, but due to the manner the algorithms searches for the boundaries. The provided per-frame time analyses are encouraging, thanks to the low computational cost of CCH. However, the total processing time of the video will significantly be affected by the employed searching algorithm, where *every single frame* in the video is processed and matched to its neighboring frames, and on top of that, the interval around *every single local minimum* is inspected for a possible shot boundary. In other words, the reported per-frame execution times will undergo considerable amount of repetitions resulting in excessive computation times for videos. In order to exemplify this, consider Fig. 1, which shows the variation of the number of matched keypoints between adjacent frames in a video. It is obvious that an innumerable number of local minima exists due to the oscillations in the number of matches within a shot. Moreover, no significant changes occur for certain shot boundaries (particularly GT), verifying the fact that adjacent frames have significantly high visual similarity and seeking shot boundaries through such an inspection will inevitably lead to erroneous results.

Following the advancements in content based image and video analysis, state-of-the-art SBD algorithms are inclined to use high-level descriptors such as object/scene detection, visual attention analysis, rather than relying on low-level descriptors. However, whereas almost all previous attempts focused on the discriminative power of underlying features, few seem to have realized the importance of the employed search scheme. We believe in the search for shot boundaries, *how* you search is as important as *what* you search for. Therefore, it is of decisive importance that a proficient search scheme is followed in order to find the shot boundaries accurately and efficiently. Humans enjoy an extraordinary ability to recognize and interpret visual similarities, differences and alterations. Hence, understanding human visual perception and how humans perform visual search will lead to an effective and competent search scheme. There have been numerous studies on human visual perception; however, our understanding of visual perception comes substantially from the *Gestalt Psychology* [32]. By taking a holistic standpoint, Gestaltism focuses on the emergent properties of visual stimuli rather than considering them individually. Following its well-known rallying cry, “The whole is greater than the sum of its parts,” Gestaltism provides a set of perceptual rules (Prägnanz) in order to explain that perception cannot be reduced to parts or even to piecewise relations among parts. Such a top-down manner has been neglected in SBD methods so far, since almost all of the previous approaches are designed in a bottom-up fashion to build on the information that is based on the relation between consecutive frames – which are basically the parts of the video – instead of considering the fact that transitions naturally emerge when the video is considered as a whole.

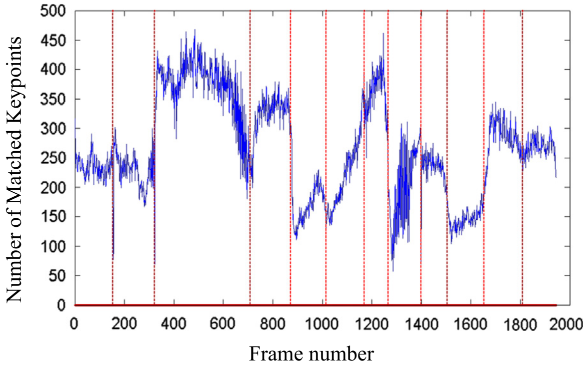


Fig. 1. Variation of adjacent frame similarity based on the number of matched keypoints. Dashed lines denote the true shot boundaries.

In addition to perceptual psychology, the field of Human-Computer Interaction is also particularly interested in the same question in order to understand “How humans perform visual search?” and reflect the answer to user interaction designs in order to provide effective means of search tools to users. The well-known “*Information Seeking Mantra*” was proposed accordingly by Ben Shneiderman in order to provide better means of information visualization [33]. In other words, it guides users to the data they are searching for in a fast and efficient way. In an abstract level, the Mantra abides by the following principle: *Overview first, than zoom and filter*. The overview phase lets the user gain an overall understanding of the data such as distribution, internal relations, etc. Then, the user zooms in to the particular item of interest and filters out uninteresting items. Even though no perceptual roots were mentioned in the proposal, the Mantra also agrees with Gestaltism by its nature, where the whole visual perception is assumed to be a top-down process. Such a search scheme is particularly suitable for SBD since shot boundaries emerge as we take a broader view to the video instead of taking a close up view, i.e. consider only adjacent frames. In order to illustrate this further, consider that all frames of a video are arranged as a sequence of images in temporal order. When we try to find the shot boundaries (visually) we do not start from the first frame and proceed frame-by-frame until we reach visually different frames to judge as the boundary. Instead, we have a broader look at the images and recognize the difference between two shots and gradually narrow our focus down to the particular location where the transition occurs. In other words, we first *overview* the video, and then *zoom in* to the boundary *filtering out* the redundant frames. This is the complete opposite of the manner that [30] searches for the boundaries, where every single pair of adjacent frames is compared and then *zoomed out* at every suspicion of a boundary.

In [34], Feng et al. followed a similar "overview first, than zoom in" mind-set in utilizing the encoded bitstream in order to detect ACs. They compare the consecutive I-Frames and continue analyzing that particular GoP (Group of Pictures) if they notice a difference by sampling the GoP further via selecting P and B-Frames. However, despite getting the inkling of the idea, they failed to grasp the importance of it since their intention was purely to exploit the encoding algorithm. In [40] we have proposed a similar approach in order to obtain the overview of the video. But, instead of selecting the I-Frames from the encoded bitstream, we performed uniform temporal sampling since the number and frequency of I-Frames is an encoding decision that mainly depend on the application (streaming, storage, mobile etc.), not the content. If the quality is of biggest concern, I-Frames might be too close to each other, or if the video is intended for streaming they might be too infrequent to save bandwidth. However, such sampling is prone to errors in case of a GT since a sample can easily be selected among the frames of a GT. Authors realized this weakness in [34] and claim to detect ACs only, however we have addressed this issue in Section 2.

Multi-Resolution Analysis (MRA) has also been employed by several methods mostly in order to be able to detect GTs [35–39]. MRA analyzes the video under several "resolutions", i.e. several levels of focus, such that high resolution provides high precision whereas low resolution enables to catch the GTs of various durations. In analogy with the aforementioned methods in [34,40], MRA samples the video with several frequencies varying from frame-by-frame to "overview". However, that results in processing every frame in the video and analyzing every possible pair-comparison resulting in a lot of redundant computation. The method in [18] may also be considered as MRA where the authors examined only two resolutions, i.e. frame-by-frame and *N* frames apart.

As mentioned above, we have proposed an earlier version of this work in [40], where we have also employed the "Overview first, than zoom and filter" principle. However, the aforementioned overview phase, i.e. uniform sampling, in [40] suffers in GT detection performance since a sample can easily be taken within a GT which in return may result in missing that boundary. This deficiency is discussed in detail in Section 2 together with the provided solution. Moreover, we have proposed a better similarity judgment through a more robust "similarity rate" definition and significantly decreased the time spent on feature matching via "Fast Approximate Nearest Neighbors" proposed by Muja and Lowe in [41].

In order to address the aforementioned drawbacks of the state-of-the-art algorithms and provide an efficient and accurate solution to the SBD problem, in this paper we propose a method that is modeled based on Shneiderman's Information Seeking Mantra that employs local image features in order to reveal inter frame dissimilarities. The proposed algorithm incorporates the proven potency of local image features, and the effectively utilized top-down search scheme provides a fast and systematic way to locate shot boundaries avoiding any unnecessary feature extraction and feature matching. We further analyzed spatial distribution of keypoints in order to increase the similarity judgment performance, which enables us to adapt to the content and content changes more accurately. The primary objective above all is to design a generic and robust SBD technique, which neither requires nor relies on any training or tuning while showing a superior performance on any video content in a computationally efficient manner.

The rest of the paper is organized as follows. In Section 2 the proposed method is explained in detail together with the underlying feature extraction, spatial analysis and the top-down search scheme. Section 3 provides the performance evaluation of the proposed method and Section 4 concludes the paper.

2. The proposed SBD algorithm

Under the light of earlier discussion, the proposed algorithm is designed to overcome the limitations and deficiencies of the preceding SBD algorithms. Such improvements are achieved by taking a perceptual point of view under the supervision of information visualization tools. The proposed algorithm starts with overviewing the video and gradually zooms in wherever a shot boundary exists as illustrated in Fig. 2. In order to judge frame (dis) similarities, local image features and their spatial distribution are analyzed. An earlier version of the proposed work was briefly described in [40]. The following subsections provide details of the proposed algorithm with justification. First, we explain the employed search scheme, i.e. *how* we perform the search, without going into details of the underlying feature, i.e. *what* we search for. The latter is

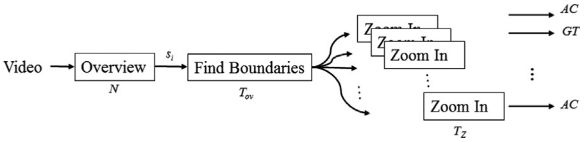


Fig. 2. Outline of the proposed SBD method.

clarified in the next subsection by giving details of how visual similarity judgment is performed.

2.1. The top-down search scheme

Stemming from perceptual rules of Gestalt psychology, we designed a top-down SBD scheme that follows the aforementioned “Information Seeking Mantra”. Recall that the Mantra suggests a perceptual path for efficiently accessing the desired information: Overview first, than zoom and filter. Accordingly, by completely rejecting a frame-by-frame processing manner, we implemented a method that provides the overview of a video, i.e. roughly gives the locations of shot boundaries. Hence, at the end of the overview phase the algorithm provides the information of how many shots exist in the video and imprecise locations of their boundaries. Then, in the next step, the algorithm gradually zooms in to those locations in order to localize the boundaries precisely. Since the algorithm only zooms in wherever there is a boundary, unnecessary processing of video frames within any shot can naturally be avoided. In other words such massive number of “uninteresting” frames are filtered out from the search.

In order to obtain the overview of the video the video is uniformly sampled in temporal domain and successively sampled frames are compared for visual similarity. The algorithm concludes that a shot transition has occurred between those frames whenever a visual discontinuity is detected. The judgment of such visual discontinuity will be detailed in Section 2.2. Let us denote the nth frame of the video as $F(n)$. Then, for every $n=N$, $2N$, $3N\ldots$ $F(n)$ is compared to $F(n-N)$ where N is the temporal sampling period. With the proper choice of N, such sampling permits sufficient content change to occur and hence enables the system to detect both AC and GT. Fig. 3 shows the overview of the same video that is used to generate Fig. 1 with $N=30$ (1*sec.*). Whereas sharper variations “near” shot boundaries are clearly seen, only ~3% of the total frames are processed in order to acquire that information.

In order to classify those variations as boundaries, we have detected the peaks and required the peaks to be “deep enough” to be regarded as boundaries. Let s be the set of similarities obtained in *Overview* such that s_i denotes the similarity between $(i\times N)th$ and $((i\text{-}1)\times N)th$ frames. Then the detection of the boundaries is achieved via the function FIND_BOUNDARIES(s) as follows:

```
FIND_BOUNDARIES (s)

for every $s_i$ i=1,2,...
  if $s_i < s_{i-1}$ and $s_i < s_{i+1}$                 : if $s_i$ is a peak
    then L←i-1, R←i+1                                 : find left and right end
        while $s_L < s_{L-1}$ {L←L-1}                    of the peak ($s_L$, $s_R$)
        while $s_R < s_{R+1}$ {R←R+1}
        if $s_i < s_L \times T_{ov}$ or $s_i < s_R \times T_{ov}$
          then zoom in to [F((i-1)×N), F(i×N)]
```

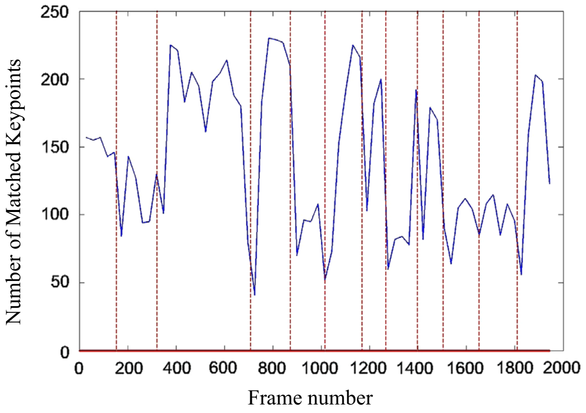


Fig. 3. Overview of the video that is used to generate Fig.1. Dashed lines denote shot boundaries.

where T_{ov} is the threshold to judge how “deep” the peaks are.

Note that the choice of the sampling period N is of decisive importance. Whereas a sparse under-sampling results in lower computational complexity, a reduced accuracy in return is inevitable especially for videos having many shots with short duration. This is due to the fact that if N is too large, an entire shot may end up in between the sampled frames and, therefore, missed. Also high object and/or camera motion may easily yield false positives. On the other hand an oversampling with a very low N value increases the computational cost and more importantly gets us closer to frame-by-frame analysis that we strive to avoid in the first place. Therefore, a reasonable and practical assumption for the minimum shot duration should be considered while deciding on N. Considering the definition of a shot (see Section 1) it should be long enough to comprise of a certain event or action. The selection of N can also be left up to the encoding scheme as in [34] where the authors selected the I-Frames to sample the video and focus on that particular GoP if there is a noticeable change. However, as we have discussed in Section 1, the distance between two I-Frames is decided during encoding depending on the target application and does not reflect the content by any means.

It should be noted that this information, i.e. the number of shot boundaries and their approximate locations, might be sufficient for various applications; however, further analysis is needed for accurate localization of the shot boundaries. The next step of the proposed search scheme realizes that by zooming into the locations where significant discontinuities in visual similarity are observed during the overview phase. That is achieved by gradually decreasing the distance between the frames that are compared for similarity. As the distance decreases, the

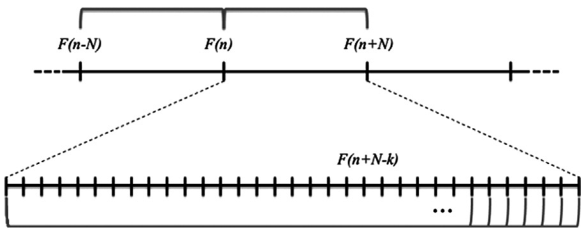


Fig. 4. *Overview* and *zoom in* phases of the proposed search scheme.

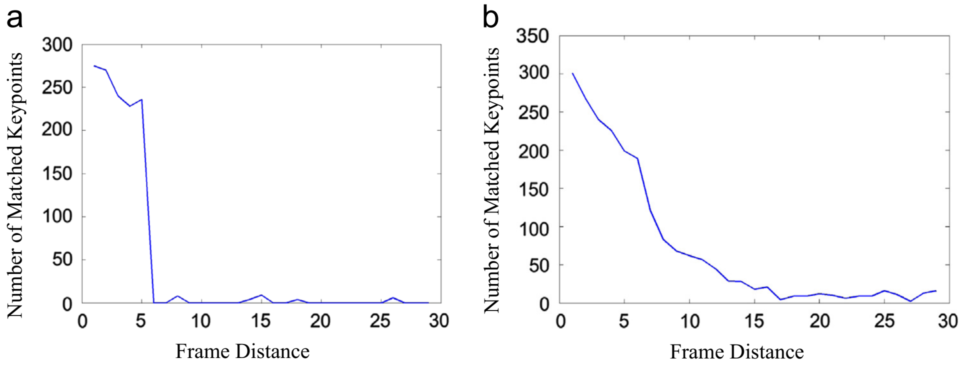


Fig. 5. Variation in the number of matches as the algorithm zooms in, reveals both the exact location and type of the transition, i.e. AC (a) or GT (b).

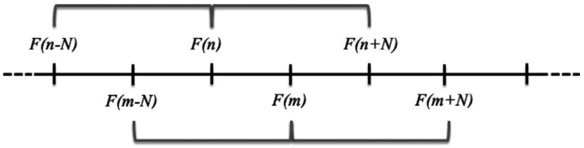


Fig. 6. Shifted tracing in the overview phase. ($m-n=N/2$).

change in frame similarity reveals not only the location of the shot boundary, but also the nature of it.

Consider the case in Fig. 4, where a shot boundary is detected between $F(n)$ and $F(n+N)$. Then, the algorithm gradually decreases the distance between frames and starts comparing $F(n)$ to $F(n+N-k)$ where $k=1, 2, 3\ldots N-1$. The variation in the similarity of frames as k approaches $N-1$ unveils the exact location and nature of the transition.

Fig. 5 illustrates how visual similarity between frames $F(n)$ and $F(n+N-k)$ changes as the algorithm zooms in (i.e. k: $1\rightarrow N\text{-}1$). Two different cases are exemplified, namely for AC (Fig. 5a) and GT (Fig. 5b). In case of an AC, the visual similarity abruptly drops, also revealing the exact location of the transition. On the other hand, when there is a GT, frame similarity gradually diminishes. It is rather easy to detect and distinguish between AC and GT by simply monitoring the change in visual similarity (the number of matches in this case) at each distance. If the change is larger than a predefined threshold T_Z, an AC is revealed with its exact location, otherwise it is a GT. Note that the algorithm zooms in to the interval between $F(n)$ and $F(n+N)$ if and only if a boundary is detected during the overview phase; otherwise, the frames in the interval are "filtered out" avoiding unnecessary feature extraction and matching.

2.1.1. Shifted tracing

Although the search scheme that is discussed so far is capable of achieving high accuracy with considerably low computational demand, it comprises an apparent imperfection. The fact that uniform temporal sampling is utilized "may" produce false negatives if $F(n)$ is sampled among the frames within a GT. In that case, since the visual content changes gradually during that interval, the sampled frame will somewhat be similar to both the preceding and the succeeding shots; hence the aforementioned algorithm will fail to realize the shot boundary by filtering it out. In order to avoid such misjudgment, we propose an improved version of the overview phase. Fig. 6 depicts the proposed shifted tracing of the video where the same sampling and comparison procedure is applied with an $[N/2]$ frames shift. In other words, overview of the video is obtained twice with temporal shift of $[N/2]$ frames. This way, if a shot boundary is missed due to a GT during the first trace, it will be detected during the second one. Considering the insignificant computational weight of the overview phase, a significant improvement is achieved with such a minimal effort.

It should be noted that shifted tracing is not a blindfolded reiteration of the overview phase. As its purpose is to enhance the accuracy of single trace overview, it therefore, avoids any redundant recalculation that has already been carried out by the first trace. In other words unnecessary "zoom in" are avoided by simply ignoring any boundaries if they have already been detected by the other trace. Consequently, the shifted tracing allows a complete overview of the video, detecting every single shot boundary and eliminating the shortcomings arising from any GT and uniform sampling applied.

In addition to avoiding unnecessary processing of video frames, the proposed algorithm is also suitable for parallel processing by its nature which further enables significant performance improvement. Both the shifted traces in the overview phase and every single "zoom in" are independent processes, hence can be handled in parallel.

2.2. Frame similarity via local features

In order to judge whether two video frames belong to the same shot, the following test is performed: if the same objects are detected in two different frames, they are considered to belong to the same shot. Such a manner of similarity judgment also tackles the prominent problem of object and camera motions innately, since the objects will still be detected (either on the foreground or background) despite any object or camera motion assuming that the two frames are not excessively apart from each other in temporal domain. Still, the choice of image feature should be able to handle possible object deviations such as variations in scale, rotation, and translation.

Local image features that are invariant to those changes are thus utilized in order to match objects between frames. First, interest points are detected throughout the frames, and then descriptors around each point are extracted. The proposed SBD method is independent of the underlying point detector and descriptor, and in this work, SURF is chosen for both detection and description due to its high correspondence with human visual saliency, improved repeatability over other detectors and lower computational complexity (see Section 1). Finally, descriptors from two frames are matched against each other to find visual correspondence. However, in addition to feature extraction, another source of the computational load is feature matching particularly if a blunt linear search is utilized. In order to further reduce the overall computational cost, we employed “Fast Approximate Nearest Neighbors” [41] which has proven to speed up the matching process up to several orders of magnitude compared to linear search by using multiple randomized k-d trees. The algorithm was tested for SIFT descriptors and achieved significant performance improvement with minimal loss in accuracy. Similarly, during our experiments using SURF descriptor, no significant performance loss is observed despite the considerable decrease in computational cost.

As discussed in the previous section and depicted in Figs. 3 and 5, variations in the number of matches between frames reveal the location of the shot boundaries. However, it should be noted that the number of total interest points detected in a frame depends entirely on the content of that frame. The variations in substantial amount of matches are relatively informative; however, with the limited number of matches due to the limited number of keypoints, the reliability of such variations degrades significantly. Considering that the keypoints reflect the visual content of a video frame, the change in the content should be revealed regardless of the number of keypoints it is represented with. In order to achieve this, we normalize the number of matches with the total number of keypoints in both frames which gives us the degree of similarity between two frames. Consider the case where K and L are the number of keypoints extracted from $F(n)$ and $F(n+N)$, respectively, and M is the number of matched keypoints between those frames. Then, the rate of similarity, R, between $F(n)$ and $F(n+N)$ can be formulated as:

$$R = \frac{2M}{K+L} \tag{1}$$

This phenomenon can easily be observed by comparing Figs. 3 and 8. Note that in Fig. 3 the variation around frame 500 can easily be mistaken as a boundary since it is comparable to real shot boundaries, whereas in Fig. 8 the variation in similarity rate is minor compared to the boundaries. Similarly the variation in the number of matches around frame 1650 may not be enough to detect it as a boundary, yet the change in similarity rate in Fig. 8 clearly signifies it as a shot boundary.

2.2.1. Spatial analysis of keypoints

Matching objects in order to reveal visual similarity is a well-reasoned perceptual approach; however, a comprehensive discussion has been made in [42] that merely matching individual local features is far from reflecting human perception. Again, following Gestalt′s rule of perception “the whole is greater than sum of its parts”, it is shown in [42] that matching complete objects is more (informative) than the sum of individually matched keypoints. Hence, following the aforementioned “Prägnanz”, certain perceptual constraints are imposed by considering keypoints′ spatial distribution. In other words their spatial proximity is taken into account and it is proposed that if two keypoints are spatially close to each other, it is highly unlikely that their corresponding matches are significantly isolated. This is due to the natural fact that the objects are solid and follow a slightly rigid motion within a shot. This is no longer valid for (accidental) matches between the frames from two distinct shots (e.g. see Fig. 7). Thus, whenever a match is found, their neighborhoods are matched against each other in order to validate the match and hence to avoid any potential false positives. Following this principle the number of both false positives and false negatives can be decreased considerably; and due to the nature of the imposed criteria, groups of matches emerge naturally instead of single individual matches as shown in Fig. 7b.

In addition to its undeniable improvement in matching performance, elimination of false negatives as in Fig. 7c particularly assists the detection of shot boundaries by providing sharper variations (i.e. deeper peaks) in the number of matches (thus, the rate of similarity) during shot transitions as shown in Fig. 8.

Huang et al. [30] also made use of the spatial information of keypoints in a similar manner. However, their analysis of spatial distribution is limited to matching adjacent frames only such that they simply limit the spatial displacement of possible matching keypoints to a certain number of pixels. In other words, a keypoint is not allowed to match another if their spatial locations are separated by more than a predefined distance threshold. However, such a limitation is reasonable only for neighboring frames due to the considerably limited content change among adjacent frames. For cases where frames at a certain temporal distance apart are compared for similarity (as in the overview phase of the proposed algorithm or the false alarm detection in [30]), such a restriction should definitely be avoided since any object or camera motion can easily violate this constraint.

Fig. 9 summarizes the whole algorithm visually. To sum up, the *Overview* phase uniformly samples the video by taking every *Nth* frame from the video and compares consecutive samples to obtain the similarities s_i. FIND_BOUNDARIES function detects the intervals where a shot change has occurred by analyzing the change in s_i. Then the algorithm *zooms in* to each of these intervals and monitors how the similarity changes as the interval gradually narrows down. The nature of the change in similarity also reveals the nature of the transition, i.e. AC or GT.

3. Experimental results

In order to demonstrate the performance of the proposed algorithm and prove its improvements over the

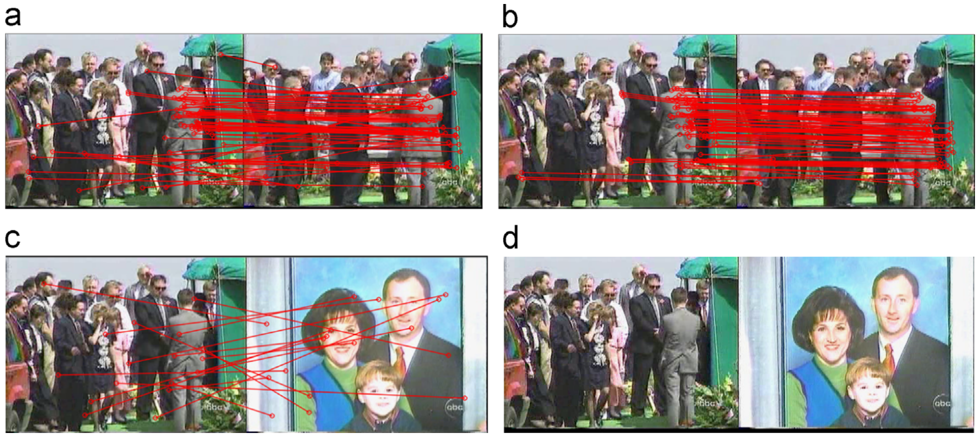


Fig. 7. Local feature matching of two frames from the same shot (*a* and *b*) and two consecutive shots (c and d) with (b and d) and without spatial analysis (a and c).

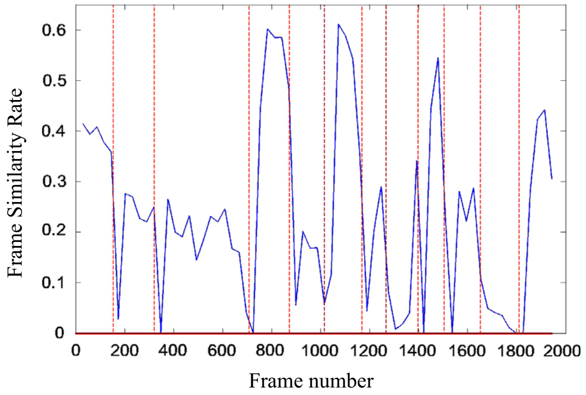


Fig. 8. Overview of the video that is used to generate Figs.1 and 3 in single trace with spatial analysis. Dashed lines denote shot boundaries.

state-of-the-art methods, we performed SBD experiments on two separate video databases: First set is the TRECVid 2005 SBD test set [16]. The dataset contains 12 videos (7 h, 744,604 frames) and has 4535 total transitions (60.8% AC, 39.2% GT). Even though we discussed in Section 1 that the strong similarity between the development and test sets of this dataset induces a strong bias to the results especially when machine learning algorithms are considered, we provide our results for the sake of completeness since it is still considered as one of the benchmark datasets for SBD. Moreover, there is still one fully automatic method that managed to make its way to the top 10 performing algorithms. The second dataset over which we performed our experiments is an extension of the dataset we have used in [40] and consists of five publicly available video sequences from *Open Video Project* [43]. The selected sequences were chosen to comprise various transition types such as wipe, dissolve, fade in/out etc., object/camera motions and to be in different video qualities. Additionally, we included some video sequences that are used in [30]. Strictly speaking, we believe that none of the videos used in [30] are suitable for testing the SBD performance due to their ambiguous content such as unclear and highly subjective shot boundaries and transitions, embedded subtitles, etc. For example, sequences shorter than 10 frames hardly qualify as shots since they are barely perceivable, yet they occur abundantly in the dataset. Transitions as long as 200 frames may even be considered as a separate shot (an overlaid shot for dissolve type transition for instance). An object passing in front of the camera is structurally identical to a wipe transition and it requires semantic comprehension to distinguish them. Embedded subtitles can be considered as a part of the visual information, but then the definition of a shot should also be well-defined in advance, i.e., what happens if subtitles stay intact but the background content changes – a new shot? Such occurrences and more arise abundantly in the dataset used in [30] which will inevitably bias the results both positively and/or negatively due to such ambiguous occurrences. Fig. 10 shows examples of such occurrences where on the top row the object in focus moves outside the camera scope within ~10 frames and the blurred background gradually comes into focus (structurally this is not different than a dissolve transition). Similarly the second row shows a sequence where an object occludes the entire view as the camera moves to the right and the scene continues as the camera keeps moving and leaving the object outside the view (again, structurally the same as a wipe transition). A similar instance also occurs in the third example. Despite such deficiencies and inaptness, we decided to include three video sequences from the dataset used in [30] for the sake of completeness, namely *News*1, *Documentary*1 and *TV Serial* (*Lost*).

Table 1 summarizes the eight video sequences in the second set used in the experiments. *Video#*1 consists of 10 TV commercials each separated by ~50 blank frames. *Video#*2 and #3 are educational videos that contain various synthetic content (such as animations, frame borders, etc.). *Video#*4 and 5 are excerpts from industrial documentaries and together with *Video* #6, #7 and #8, they have relatively small frame size. *Video#*6 is a NASA documentary containing mostly dissolve type GT and also various shots with significantly short duration (only 25–30 frames). *Video#*1, #4, #5 and #6 are all from 1950s, thus have low video quality. Moreover, particularly *Video#*4 comprises challenging boundaries where shots with high motion are connected with slow wipe or dissolve type GT around 2 s. (~50–60

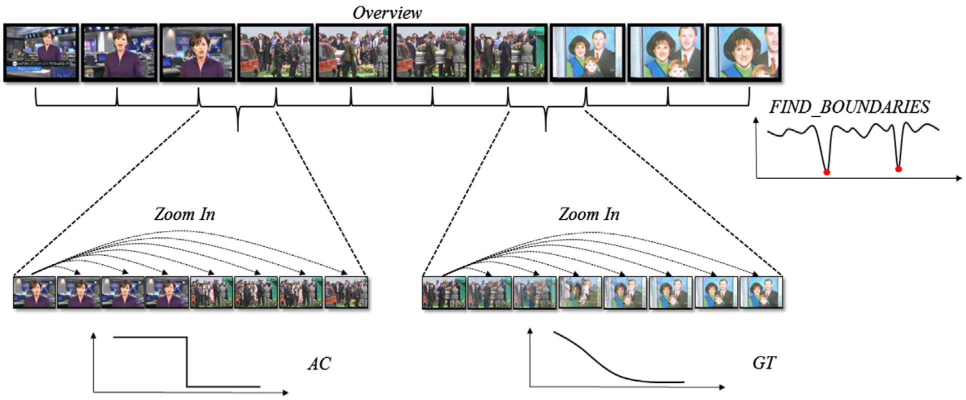


Fig. 9. Summary of the proposed algorithm.

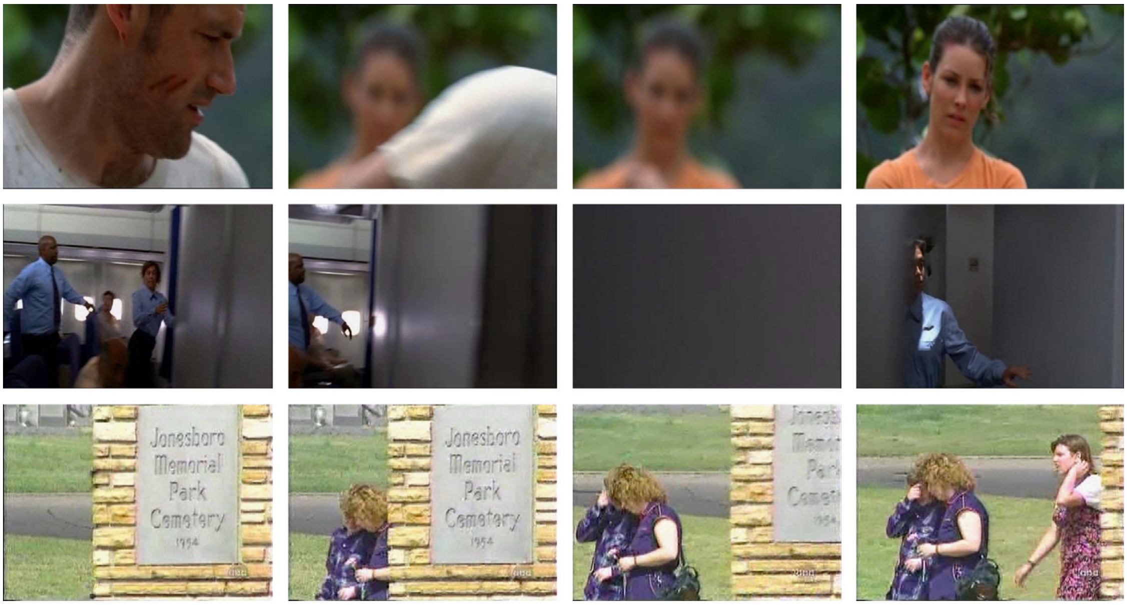


Fig. 10. Examples of misleading shots from the videos used in [30].

Table 1
Experiment Dataset 2.

#	*Name*	*Size*	*Total # of Frames*	*AC*	*GT*	*Total*
1	1955 Chevrolet Screen Ads	480 × 368	15,802	69	9	78
2	Volcano Eruptions	720 × 480	3332	25	2	27
3	History Of Flight	720 × 480	2801	19	4	23
4	American Look	320 × 240	1945	3	8	11
5	Human Dividends from American Industry	320 × 240	1870	19	1	20
6	Documentary1 (ANNI005)	320 × 240	11,363	37	29	66
7	News1 (19980328_ABC)	352 × 264	23,642	116	50	166
8	TV Serial (Lost)	352 × 240	30,705	297	1	298
	Total		**91,460**	**585**	**104**	**689**

frames). *Video* #8 is particularly challenging due to style that the series is shot where very close facial shots dominate the video. Such a technique results in significant content change even under the slightest object movements. Moreover, significantly short shots (as short as 10 frames) and high motion content makes this video further challenging. Fig. 12 can be referred in order to grasp the gist of the contents of the videos.

As mentioned in Section 2.2, we used SURF for both feature detection and description due to its consistency

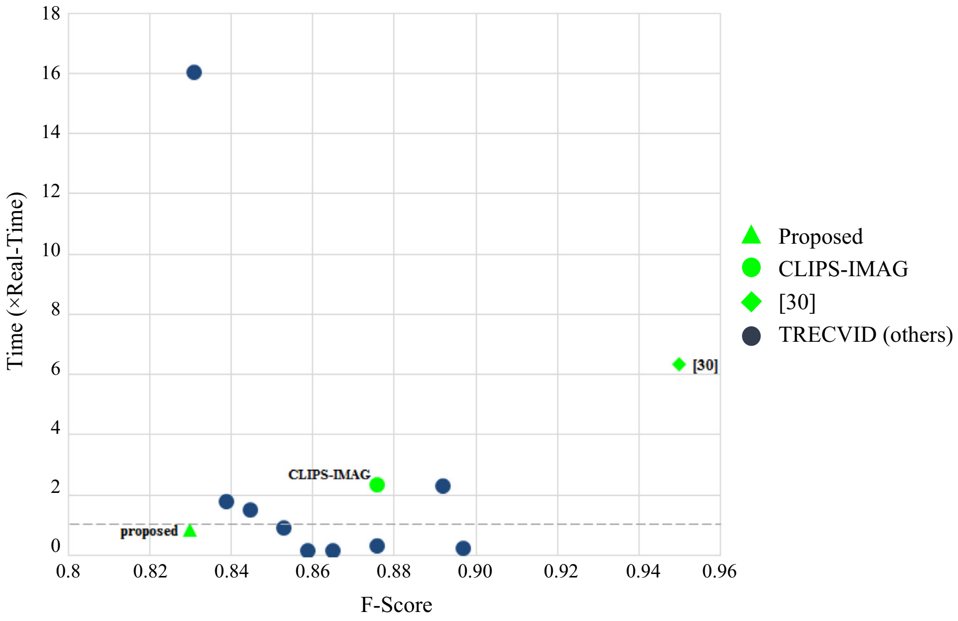


Fig. 11. Performance vs. Computation Time analysis for the competing methods. The method proposed in this paper, in [30] and the only non-machine learning algorithm in TRECVID 2005 (CLIPS-IMAG) are labeled whereas the unlabeled data points are the remaining 9 of the top 10 performing algorithms in TRECVID 2005. The dashed horizontal line represents the speed equivalent to real-time operation.

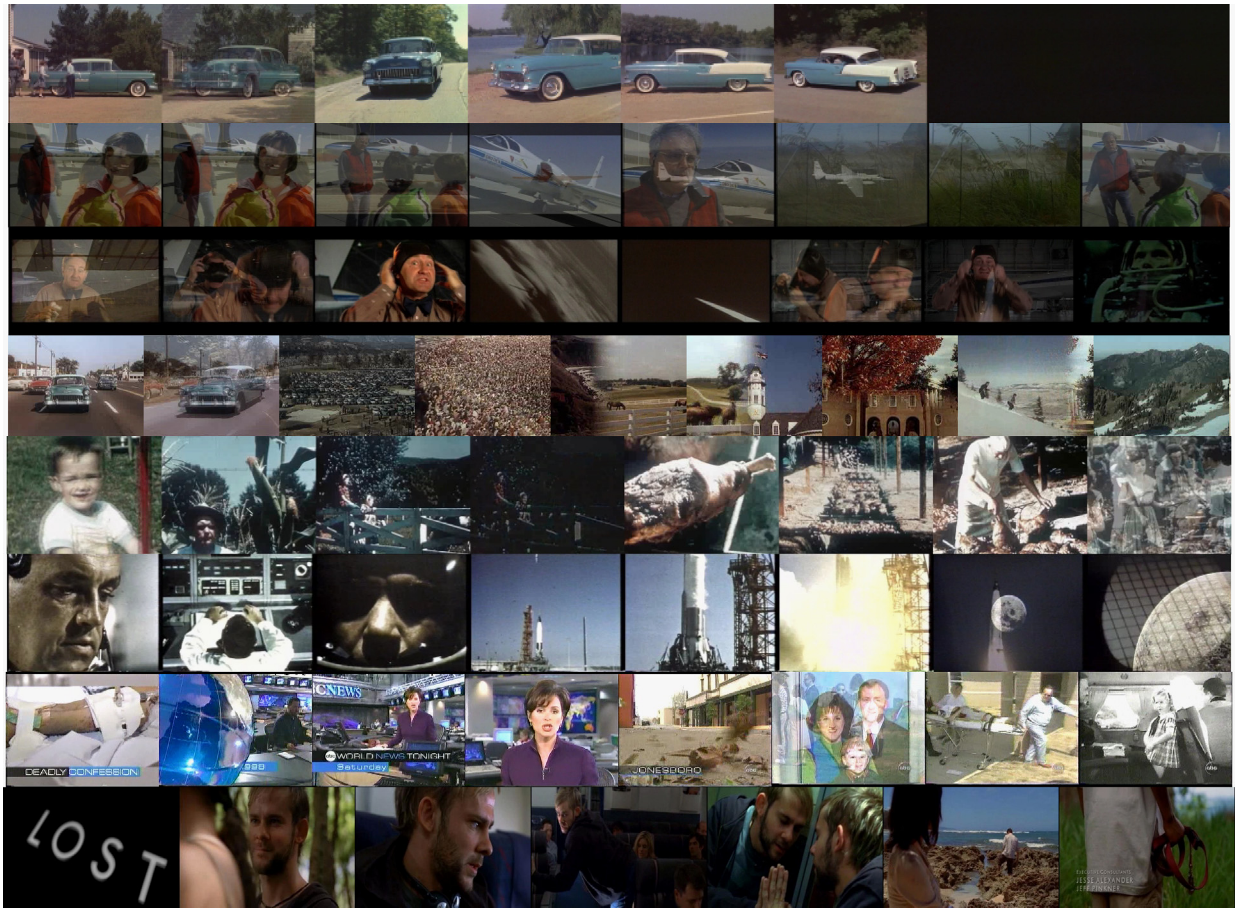


Fig. 12. Excerpts from the SBD results of the proposed method from the second dataset. *Video*#1 (*top row*)-*Video*#8 (*bottom row*).

with human visual saliency and ease of computation [24]. Videos are sampled with 0.5 s. period (i.e. half of the frame rate of the video) in the overview phase, inferring from the definition of a video shot in Section 1 that any shorter duration will be impractical if not imperceptible. Moreover, as opposed to the initial conception, a more sparse sampling (larger N) does not yield smaller computation times since a larger N means more frames to process during zoom-in. Even though such a condition is heavily dependent on the video content (i.e. number of shots in the video), we observed insignificant variations in computation times for those N settings for 1sec. and 0.5 s. sampling. Also $T_{ov}=0.5$ and $T_Z=0.5$ are used for all the experiments. The ground truths are extracted manually for all videos in the second dataset and precision-recall (P–R) values are calculated as performance measures. In order to provide a complete comparison in each dataset, performance measures used by competing methods have also been calculated: *F*1-*score* for the first dataset [16] and *Q-value* for the second dataset [45]. The experiments are carried out on a hardware with 4.00GB RAM and 2.20GHz Core2Duo CPU. The software relies on OpenCV libraries [44] for loading videos, querying frames and extracting/matching keypoints.

In addition to performance evaluation, computational time analysis is also provided in order to demonstrate that such performance is achieved with tremendous computational efficiency. In order to exhibit the improvement achieved by the employed search scheme over frame-by-frame methods in the second dataset, we followed the same search scheme utilized in [18,30] and compared every adjacent frame by extracting features from every frame in the video by the same descriptor, SURF. By doing so, we intend to demonstrate how much time it would take if a frame-by-frame search scheme is instead utilized as in [18,30]. In order to provide an accurate comparison against the competing methods, any approximate measure (such as [41]) is avoided. To our best knowledge, [30] achieved the best SBD performance using local image features. Despite the inappropriateness of the videos (Video #6, #7 and #8), the proposed approach achieved results on a par with [30]. Visual excerpts from the detection results are also provided in Fig. 12 where different types of GTs such as wipe, fade and dissolve are easily observed together with ACs. An immediate remark from Fig. 12 is that some of the ACs, particularly in Video#2 and #3, appear like dissolve type GTs. This is due to the fact that a single transition frame exists that is imperceptible by the human eye, yet detected by the proposed method.

Despite the clear advantage that the machine learning algorithms have which we discussed in Section 1, our method still managed to achieve a performance on par with all the top 10 performing algorithms on the first dataset. Excluding the machine learning approaches for the obvious reasons, our algorithm ranked the second among all algorithms involved in TRECVid 2005 based on F1-measure. The best performance came from the CLIPS-IMAG laboratory which does not use any machine learning algorithm, yet still uses specifically selected algorithms for TRECVid dataset. To be exact their system is composed of a cut detector, a flash detector (vastly present in the dataset) and a dissolve detector (78% of the GT′s in TRECVid dataset are of dissolve type). However, despite its significant performance, the algorithm runs considerably slow. Overall performance and execution time comparisons in the whole TRECVid 2005 dataset are given in Table 2 where time measures are given relative to real-time. The results in this dataset are a clear demonstration of the huge efficiency gain that the proposed SBD scheme provides without sacrificing high performance. In fact, the proposed method runs even faster than several machine learning algorithms (ranks 5th among all) despite the fact that the time for training the whole system is excluded from those algorithms′ execution time.

Table 2
Performance of the proposed method in TRECVid 2005 dataset.

Algorithm	*F*	*Time*	*Time* (*Overview only*)
CLIPS-IMAG	0.88	× 2.3	*N.A.*
Proposed	0.83	× 0.82	× 0.52

Table 3
Performance analysis on the second dataset.

Name	*Precision*	*Recall*	*Q*
1955 Chevrolet Screen Ads	0.83	1.00	0.83
Volcano Eruptions	0.87	1.00	0.87
History Of Flight	0.95	0.91	0.87
American Look	0.85	1.00	0.85
Human Dividends from American Industry	0.83	1.00	0.83
Documentary1 (ANNI005)	0.83	0.94	0.78
News1 (19980328_ABC)	0.85	0.93	0.79
TV Serial (Lost)	0.73	0.92	0.67
Average	**0.84**	**0.96**	**0.81**

Table 3 summarizes the results of our experiments on SBD performance on the second dataset. The results indicate that on the average 96% of the shot boundaries can be detected by the proposed SBD technique in a generic way. This is most likely in the close vicinity of the upper recall limit that can be achieved without any training, learning or manual tuning involved. Considering that the experimental set contains a wide selection of GT types, video qualities, frame sizes and shot durations, it can easily be inferred from the results that the proposed algorithm is capable of detecting any type of GT and AC with such an elegant recall rate.

It should be noted that the main goal of the proposed algorithm is not only to achieve such an elegant performance, but also to achieve it under low computational costs. The computational times for the second dataset presented in Table 4 demonstrate that the proposed method is significantly superior in terms of computational efficiency. On the average around 87% improvement is achieved in terms of computation complexity compared to the methods [18,30] that employ frame-by-frame analysis. In other words, the proposed scheme enables around 7 times faster processing compared to any frame-by-frame processing scheme. Even though [30] has a comparable

Table 4
Computation time analysis on the second dataset.

#	*Name*	*Total* (*sec*)	*Overview* (*sec*)	*Zoom-In* (*sec*)	[18,30] (*sec*)	*Computational Gain*
1	1955 Chevrolet Screen Ads	692.17	508.97	183.20	6104.74	× 0.11
2	Volcano Eruptions	239.17	153.19	85.98	2463.71	× 0.10
3	History Of Flight	236.06	152.24	83.82	2874.00	× 0.08
4	American Look	47.94	34.45	13.49	349.92	× 0.14
5	Human Dividends from American Industry	61.22	36.54	24.68	397.81	× 0.15
6	Documentary1 (ANNI005)	218.88	147.88	71.00	1606.38	× 0.14
7	News1 (19980328_ABC)	914.90	678.37	236.53	5784.65	× 0.16
8	TV Serial (Lost)	689.87	412.46	277.40	3575.83	× 0.19
	Average	**387.53**	**265.51**	**122.01**	**2894.63**	× **0.13**

Fig. 13. Two adjacent frames belonging two adjacent shots (*Video*#5).

detection performance, it is obvious that their frame-by-frame analysis leads to an impractical computational complexity for a real-time SBD operation. Moreover, note that the computation time for the proposed approach is obtained by fully employing the "Overview, zoom-in and filter" procedure (including the spatial analysis of the keypoints), whereas times for the frame-by-frame analysis approach includes *only* the feature detection, extraction and matching. It should be noted that, particularly [30] performs significant number of additional keypoint matchings and an intensive and computationally complex analysis on the number of matched keypoints which are excluded from the computational times reported in Table 4. It is possible to decrease the computation times given in Table 4 by using simpler and faster feature detectors/extractors (as in [30] via CCH), yet that possibility exists for *any* approach utilizing local image features bearing in mind that the proposed SBD scheme is independent of the utilized image feature. In other words, thanks to the efficiently utilized top-down search scheme, the computational supremacy of the proposed approach over any frame-by-frame processing algorithm will still prevail.

In short, the main advantage of the proposed algorithm is the ability to find out the exact locations of shot boundaries with a significantly low computational complexity (see Table 4). As discussed in Section 2.1 the outcome of the overview phase is the total number of shot boundaries and their imprecise locations (with a maximum deviation of $N-1$ frames). Note that this information alone can be useful and even sufficient for various applications, e.g., consider the case where a storyboard is to be extracted from a video where each shot is represented by a single video frame. Whereas the selection of representative frames (i.e. keyframes) among all shot frames is another research topic, the proposed overview scheme provides an immediate and fast solution to the problem without requiring any further implementation and computational cost. Another crucial advantage of the proposed method is that it allows the SBD results to be presented to the user in a progressive manner and furthermore allows user interactions with the ongoing process; i.e. the initial results (outcome of the overview phase) can immediately be presented to the user while the system can then continue to the *zoom in* phase if the need arises or alternatively, it can be stopped by the user if the results found so far are satisfactory. By doing so, not only excessive idle intervals are avoided, but also the possibility to interact with the system is granted to the user. Consider another use case where the user aims to extract only certain shots from the video. The overview phase initially provides representative frames from each shot in the video (those are the sampled frames mentioned in Section 2.1). This way, the user can directly access the shots of interest and the proposed method will then only zoom in to those shots′ boundaries, thus avoiding redundant processing.

Performance vs. Computation Time analysis is also provided in Fig. 11 comparing the proposed approach with the top 10 performing algorithms in TRECVid 2005 dataset. For illustrative purposes, we have also added the algorithm in [30] despite the fact that there is no evaluation data on TRECVid dataset for that algorithm. Thus, we have

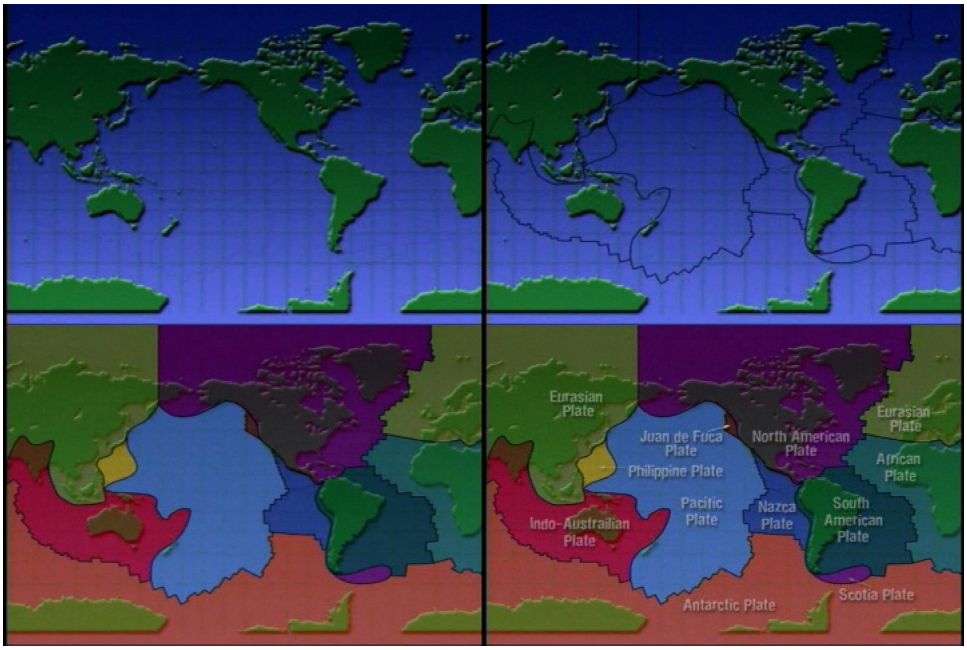


Fig. 14. Four frames from the same shot in *Video*#2. The map is left almost imperceptible by the appearing colored regions.

used the performance score they have reported in [30] and the computation time we have simulated and reported in Table 4. It is clear that despite the high performance score reported in [30], the computational efficiency is a huge handicap. The figure also demonstrates the on-par performance and computation time achieved by the proposed algorithm despite the aforementioned controversial objectivity of the machine learning algorithms used in TRECVid 2005.

Despite the fact that remarkable results are achieved in terms of accuracy, localization and computational complexity, there are rare cases where the proposed method failed to detect shot boundaries. One example of such occurrences is from *Video*#5 and shown in Fig. 13, where both shots are from the same scene and have the same camera angle. Moreover, considerably dark content of the shots weakens the discriminative power of the features, yielding a misjudgment that both frames belong to the same shot. Note that although these frames are from the same scene, a shot-cut occurred in between.

Another case, which is shown in Fig. 14, arises mainly from the uncertainty about the definition of a shot. The frames are from *Video*#2 and all from the same shot, where the color regions and text appears on top of the map gradually and leave the map vaguely visible. Such a change is regarded as a GT by the proposed algorithm. Yet, since those changes cannot be regarded as object or camera motion, it is hard to classify such artificial content changes as shot boundaries or not, even by a human observer.

4. Conclusions

A novel modus operandi for shot boundary detection is proposed where Gestalt laws of visual perception are taken as a model for both recognizing shot changes and seeking the location of the boundaries. In order to locate shot boundaries accurately and quickly, an efficient search scheme is proposed based on the “Information Seeking Mantra”. The proposed method provides an outstanding improvement in terms of computational complexity while achieving an elegant performance. Yet, the key contribution of the paper is in demonstrating how a proper understanding of human perception can lead a simple and effective solution for content analysis, and avoiding any over-engineering of the problem under the guidance of human psychology and human-computer interaction. Furthermore, the proposed method allows a user interaction to direct the SBD process to user's “Region of Interest” or to stop it once satisfactory results are obtained. Considering that SBD is a prominent enabler in video content analysis, such interaction might be of valuable importance to certain applications minimizing user's idle time and further lowering the computational cost significantly.

References

[1] alexa.com, Available: 〈http://www.alexa.com/topsites〉, 2013.

[2] youtube.com, Available: 〈http://www.youtube.com/yt/press/statistics.html〉, 2013.

[3] R. Thompson, Grammar of the Shot, Focal Press, 1998.

[4] U. Gargi, R. Kasturi, S.H. Strayer, Performance characterization of video-shot-change detection methods, Circuits Syst. Video Technol. 10 (1) (2000) 1–13.

[5] M. Swain, D. Ballard, Color indexing, Int. J. Comput. Vis. 7 (1) (1991) 11–32.

[6] F. Arman, A. Hsu, M.-Y. Chiu, Feature management for large video databases, in: Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases I, vol. SPIE 1908, 1993, pp. 2–12.

[7] H.J. Zhang, Video parsing using compressed data, in: Proceedings of SPIE Symp. Electronic Imaging Science and Technology: Image and Video Processing II, 1994, pp. 142–149.

[8] J. Meng, Y. Juan, S.F. Chang, Scene change detection in a MPEG compressed video sequence, in: Proceedings of SPIE/IS&T Symposium on Electronic Imaging Science and Technology: Digital Video Compression: Algorithms and Technologies, vol. 2419, 1995.

[9] H. C. Liu, G.L. Zick, Automatic determination of scene changes in MPEG compressed video, in: Proceedings of ISCAS-IEEE International Symposium on Circuits and Systems, 1995, pp. 764–767.

[10] B.-L. Yeo, B. Liu, A unified approach to temporal segmentation of motion JPEG and MPEG compressed video, in: Proceedings of 2nd International Conference on Multimedia Computing and Systems, 1995, pp. 81–83.

[11] K. Shen, E.J. Delp, A fast algorithm for video parsing using MPEG compressed sequences, IEEE International Conference Image Processing, October 1995, pp. 252–255.

[12] I.K. Sethi and N. Patel, A statistical approach to scene change detection, in: Proceedings of IS&T/SPIE Conference Storage and Retrieval for Image and Video Databases III, vol. SPIE 2420, 1995, pp. 329–338.

[13] S. Teng, Video temporal segmentation using support vector machine,, Proceedings of the 4th Asia Information Retrieval (2008) 442–447.

[14] Y.U. MengL.-Gong Wang, L.-Zeng Mao, A shot boundary detection algorithm based on Particle Swarm Optimization Classifier, in: International Conference on Machine Learning and Cybernetics, July 2009, pp. 1671–1676.

[15] A. Hanjalic, Shot-boundary detection: unraveled and resolved? IEEE Trans. Circuits Syst. Video Technol. 12 (2) (2002) 90–105.

[16] A.F. Smeaton, P. Over, A.R. Doherty, Video shot boundary detection: seven years of TRECVid activity, Comput. Vis. Image Underst. 114 (4) (2010) 411–418.

[17] G. Boccignone, A. Chianese, V. Moscato, A. Picariello, Foveated shot detection for video segmentation,, IEEE Trans. Circuits Syst. Video Technol. 15 (3) (2005) 365–377.

[18] M.-H. Park, R.-H. Park, S.W. Lee, Shot boundary detection using scale invariant feature matching, in: Proceedings of SPIE Visual Communications and Image Processing, 2006, 6077, pp. 569–577.

[19] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vision 60 (2) (2004) 91–110.

[20] C. Harris, M. Stephens, A combined corner and edge detector, in: Proceedings of the 4th Alvey Vision Conference, 1988, pp. 147–151.

[21] T. Lindeberg, Feature detection with automatic scale selection, Int. J. Comput. Vis. 30 (1998) 79–116.

[22] K. Mikolajczyk and C. Schmid, An affine invariant interest point detector, in: Proceedings of the 7th ECCV-Part I, London, UK, Springer-Verlag, 2002, pp. 128–142.

[23] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (Surf),, Comput. Vis. Image Underst. 110 (3) (2008) 346–359.

[24] P. Harding, N.M. Robertson, A comparison of feature detectors with passive and task-based visual saliency, in: SCIA ′09: Proceedings of the 16th Scandinavian Conference on Image Analysis, Berlin, Heidelberg, Springer-Verlag, 2009, pp. 716–725.

[25] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions,, Image Vis. Comput. 22 (10) (2004) 761–767.

[26] E. Rosten, T. Drummond, Fusing Points and Lines for High Performance Tracking, in 10th IEEE International Conference on Computer Vision (ICCV′05), 2005, vol. 2, pp. 1508-1515.

[27] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, Comput. Vis.–ECCV 2006 (2006) 430–443.

[28] T. Kadir, M. Brady, Saliency, scale and image description,, Int. J. Comput. Vis. 45 (2) (2001) 83–105.

[29] T. Tuytelaars, A survey on local invariant features, Found. Trends Comput. Graph. Vis. 1 (1) (2008) 177–280.

[30] C.-R. Huang, H.-P. Lee, C.-S. Chen, Shot change detection via local keypoint matching, IEEE Trans. Multimedia 10 (6) (2008) 1097–1108.

[31] C.-R. Huang, C.-S. Chen, P.-C. Chung, Contrast context histogram—an efficient discriminating local descriptor for object recognition and image matching, Pattern Recog. 41 (10) (2008) 3071–3077.

[32] S.E Palmer, Vision Science: Photons to Phenomenology, MIT Press, 1999.

[33] B. Shneiderman, The eyes have it: a task by data type taxonomy for information visualizations, in: Proceedings 1996 IEEE Symposium on Visual Languages, 1996, pp. 336–343.

[34] J. Feng; K.-T. Lo, H. Mehrpour, Scene change detection algorithm for MPEG video sequence, 1996. in: Proceedings International Conference on Image Processing, vol.1, pp.821,824 vol.2, September 1996, pp. 16–19.

[35] Y. Lin, M.S. Kankanhalli, T.-S. Chua, Temporal multiresolution analysis for video segmentation, in: Proceedings SPIE Conference Storage Retrieval Media Database VIII, vol. 3972, 2000, pp. 494–505.

[36] R. Lienhart, Reliable dissolve detection, in: Proceedings SPIE Storage Retrieval Media Database, Jan. 2001, vol. 4315, pp. 219–230.

[37] C.-W. Ngo, A robust dissolve detector by support vector machine, in: Proceedings of ACM Multimedia, 2003, pp. 283–286.

[38] T.-S. Chua, H. Feng, C.A., An unified framework for shot boundary detection via active learning, in: Proceedings of ICASSP, Hong Kong, 2, April 2003, pp. 845–848.

[39] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, B. Zhang, A formal study of shot boundary detection, IEEE Transactions on Circuits and Systems for Video Technology, 17, 2, Feb. 2007, pp. 168–186.

[40] M. Birinci, S. Kiranyaz, M. Gabbouj Video Shot Boundary Detection by Structural Analysis of Local Image Features, 12th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), April 2011.

[41] M. Muja, D.G. Lowe, Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration, in: International Conference on Computer Vision Theory and Applications, VISAPP, vol. 340, 2009, pp. 331–340.

[42] M. Birinci, F. Diaz-de-Maria, G. Abdollahian, E.J. Delp, M. Gabbouj, Neighborhood matching for object recognition algorithms based on local image features, in: Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE), 2011, pp. 157–162.

[43] open-video.org, The Open Video Project, Available: ⟨http://www.open-video.org⟩, 2011.

[44] G. Bradski, A. Kaehler, Learning OpenCV, O′Reilly Media, 2008.

[45] R.A. Joyce, B. Liu, Temporal segmentation of video using frame and histogram space, IEEE Transactions on Multimedia 8 (1) (2006) 130–140.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-3975-6
ISSN 1459-2045