NORTHWESTERN UNIVERSITY

Perceptual Texture Similarity Metrics

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Electrical Engineering

By

Jana Žujović

EVANSTON, ILLINOIS

August 2011

ABSTRACT

Perceptual Texture Similarity Metrics

Jana Žujović

With the ubiquity of computers and "smart" devices, it is important to obtain intuitive visual interactions between humans and computers, which include the ability to make human-like judgments of image similarity. However, pixel-based image comparisons are not suited for this task, especially when comparing texture images, which can have significant point-by-point differences, while to humans they appear to be identical. Instead, "Structural Similarity Metrics" attempt to incorporate "structural" information in image comparisons by relying on local image statistics. We develop new "Structural Texture Similarity Metrics" that are based on an understanding of human visual perception and incorporate a broad range of texture region statistics. We develop separate metrics for the grayscale component of texture and its color composition, which are attributes associated with different perceptual dimensions.

A major contribution of this thesis is a new methodology for systematic performance evaluation of texture similarity metrics, which should be targeted to each specific application. The proposed methodology considerably simplifies the testing procedures, and

dramatically increases the chances of obtaining consistent subjective results. It is based on the realization that quantifying similarity when textures are dissimilar is difficult to achieve by humans or machines, and should be limited to the high end of the similarity scale. Thus, in content-based image retrieval (CBIR), the focus should be on distinguishing between similar and dissimilar textures, while in image compression, quantifying similarity should be limited to the high end of the similarity scale. For CBIR, we develop "Visual Similarity by Progressive Grouping (ViSiProG)," a new experimental procedure for subjective grouping of similar textures to serve as a benchmark for the development and testing texture similarity metrics. For image compression, we develop algorithms for generating texture deformations that facilitate subjective and objective tests at the high end of texture similarity. We also examine "known-item search" (retrieval of "identical" textures), where the ground truth can be obtained without extensive subjective tests.

Experimental results demonstrate that texture retrieval and compression performance evaluation based on the proposed metrics substantially outperforms those based on existing metrics.

Acknowledgements

I would like to thank my advisor and committee chair Dr. Thrasyvoulos Pappas for his guidance and the long brainstorming sessions we shared. Also, Dr. David Neuhoff helped me many times when I needed an outside point of view and his comments and ideas inspired and steered this work in the right direction. I would like to thank Dr. Jack Tumblin, Dr. Ying Wu and Dr. Alok Choudhary for serving on my committee and spending their valuable time.

I also want to thank Dr. Huib de Ridder and Dr. Rene van Egmond from Delft University, Netherlands. Their help was invaluable in creating the subjective experiments and analyzing the results in a proper way.

Finally, I wish to thank Andreas Ehmann for all the advice and suggestions he offered during the course of my studies, as well as for his help with the writing of this thesis.

Mojim roditeljima

To my parents

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

The ubiquity of personal computers and "smart" electronic devices in contemporary life necessitates the development of new techniques that will make human-computer interaction easier and more intuitive. In addition to communicating with other humans, one of the primary functions of such devices is to search for content on the Internet. Since the Internet is rich in graphical and pictorial content, understanding and imitating the way humans react to and make comparisons of visual stimuli is a key step on the way to achieving better human-computer interaction.

One of the shortcomings of existing search engines is the inability to perform Content-Based Image Retrieval (CBIR) without human intervention. Automated content analysis is necessary for assigning similarity scores between millions of images for CBIR, as it would be extremely time consuming, expensive, and inefficient to have it done by humans. Therefore, the goal is to develop efficient and accurate techniques that would allow computers to organize and retrieve visual content like humans do but without human intervention.

However, this is not a trivial problem: visual stimuli activate a large part of the brain cortex [**1**], which implies that our vision system is a complicated mechanism. As of now, it has not been possible to build a system that can successfully reproduce the functioning of the human brain and perform the complex data processing innate to living beings. Thus, the ability to simulate human visual perception using the existing capabilities of

(a) Query image (b) Same texture (c) Similar texture

Figure 1.1. Example texture retrieval results

computers remains an extremely ambitious goal. Instead, we focus our attention on the more attainable but challenging the problem of imitating subjective judgments of image similarity. For example, if the image given in Figure 1.1(a) is the query, we wish to have an algorithm that would be able to say that the image in Figure 1.1(b) is essentially the same texture (even though the two images differ pixel-wise), and that the image in Figure 1.1(c) is very similar to the previous two, although not necessarily representing the same, identical texture.

The concept of image similarity is closely related to the concept of image quality, and to be more precise, image fidelity, even though image quality is the most widely used term. Image quality has maintained its popularity among the engineers and researchers from the early days of digital imaging until the present. This has been conditioned on the constant development of new imaging devices, the ever-increasing resolution of advanced displays, the explosion of "smart" gadgets, as well as the pervasive use of the Internet. As the new technologies emerge, they usually require modifying or even completely redefining the existing image similarity metrics, since each one of the novel applications has its own constraints on the performance of the similarity metrics.

The conventional problem of image quality, and by extension similarity, evaluation consists of measuring the distortions between the original and the compressed images or video. This problem is usually solved by computing the mean-squared error (MSE) or its derivative, the peak peak-signal-to-noise ratio (PSNR), between the original and distorted images, and this is true even for the latest video standards such as H.264 [**2**]. There has also been a considerable body of work on "perceptual metrics," which incorporate low-level properties of the human visual system in order to achieve "perceptually lossless" compression, that is, images cannot be distinguished in a side-by-side comparison at a given display resolution and viewing distance [**3**]. However, this type of metrics also fall in the point-by-point comparison category.

This point-by-point measurement of image similarity has been shown to not be in accordance with human perception, especially in textured areas [**4**, **5**]. Thus, one of the main avenues for large improvements in image and video compression algorithms is a better understanding of texture and the development of perceptually-based texture predictors and coders, that take into account higher-level properties of the human visual system. The goal is to achieve "structurally lossless" image compression [**4**, **5**], whereby the original and compressed images, while visually different in a side-by-side comparison, both have high quality and look "structurally" similar, so that human subjects would not be able to tell which one of the two is the original. This can be seen in Figure 1.2, where the parts of the baboon fur in the image on the right are coded by reutilizing some portions of the fur from the other locations in the image. Another example for this type of compression would be replacing parts of the grass of football fields, or textures like sand, water, forest, starry night, etc. in still images and videos.

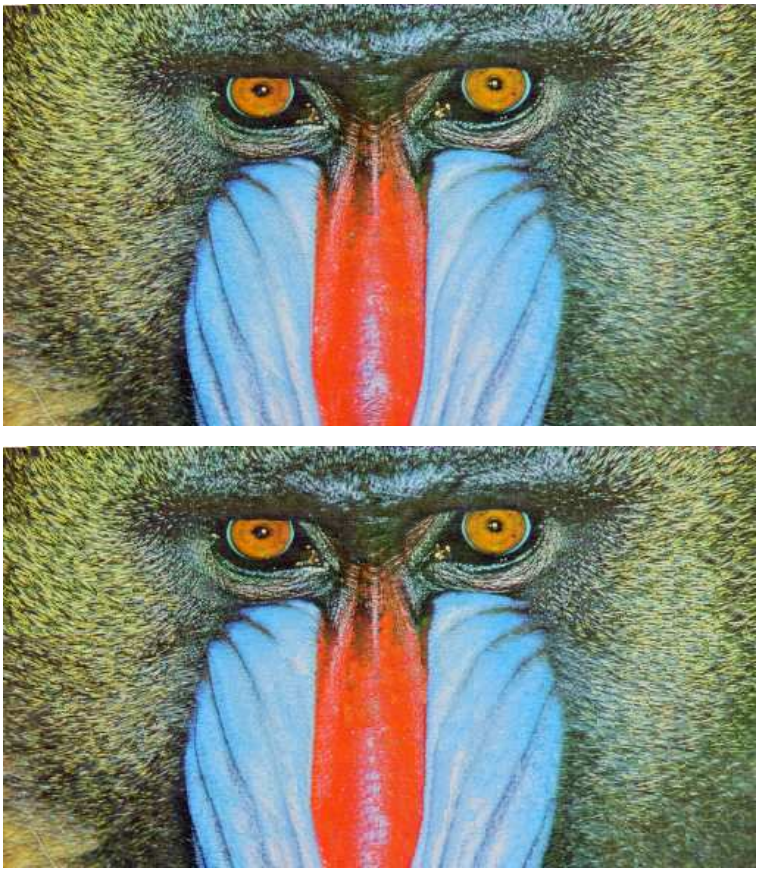


Figure 1.2. A 288 × 512 section of original "Baboon" image (top) and texture-based coded version (bottom) with 24% of the pixels between rows 65 and 288 replaced by pixels from previous blocks.

Yet another branch that requires novel image similarity algorithms is the blooming of immersive technologies and augmented reality applications. Their advancement requires an in-depth research on human multimodal perception, and how the combination of different senses – visual, auditory, tactile, even taste – affect their interaction and the impression they leave on humans. One of the most interesting applications is sense substitution. For example, converting visual stimuli into tactile form can be of great help for the handicapped population [**6**]. Another interesting application is the alteration of

perception using multimodal stimuli. An elaborate example is the Meta Cookie [**7**], where the sense of smell in combination with adequate visual system stimulation alters the taste of a plain cookie.

All of these applications require a thorough knowledge of how the human visual system works, and how its properties are affecting our perception of the outer world, and in particular, image similarity. In order to develop systems and algorithms that are capable of approximating human reactions to visual stimuli, it is essential to understand human visual perception and to exploit its characteristics in their development. Of equal importance is the understanding of signal characteristics, and in particular, of visual textures, the stochastic nature of which dictates that we exploit their statistical characteristics. Finally, it is important to identify the target applications, such as image compression, image retrieval, segmentation, etc., so that we can tailor algorithm development and testing to the specific tasks and to exploit the relevant human visual system (HVS) characteristics.

Since our focus is on human perception of textures, it is important that metric performance agrees with human judgments. Our goal is not necessarily to imitate the exact mechanism with which HVS recognizes textures, but to use it as a guide for what can be achieved, and when possible, to incorporate an understanding of the basic principles on which human visual perception is based. Thus, based on the fact that the HVS utilizes some type of multiscale frequency decomposition [**8**, **9**], we base the metrics that analyze grayscale structure on a steerable filter decomposition [**10**]. We also base the color composition metric on the idea of dominant colors, on which human perception is based [**11**–**14**].

The abilities of human perception provide a guide not only to what can be achieved, but also of what we cannot reasonably expect to achieve. For example, while humans can easily distinguish similar from dissimilar textures, they are inconsistent when it comes to assigning similarity values to dissimilar textures. Such observations have major implications for the design of subjective tests and the procedures for algorithm development and testing. In particular, they considerably simplify the testing procedures and make it possible to obtain consistent subjective results.

The focus of this thesis is on perceptual texture similarity and the development of objective texture similarity metrics that correlate well with subjective similarity judgments. When comparing different visual stimuli, existing objective similarity metrics – for the most part metrics for the evaluation of image quality, but also used for content-based image retrieval – rely on pixel-by-pixel comparisons, that is they are directly based on the intrinsic (raw) image representation in computers as a collection of pixels. It has been shown, however, that such metrics do not correlate well with human judgments of similarity. In contrast, the human brain "codes" visual stimuli by extracting features that are better representations of spatial, temporal, and color characteristics [**15**]. The limitations of point-by-point representations are particularly important for images of textures, the stochastic nature of which requires more elaborate statistical models. For example, two texture images may have significant point-by-point differences, even though a human subject considers them virtually identical. Here, it is important to point out that we are not talking about "visually indistinguishable" images, a notion that has been exploited by the perceptual quality metrics and "perceptually lossless" image compression techniques [**3**] we mentioned above. Two textures that human subjects consider identical may still have

visible differences. We should also point out that our goal is to evaluate texture similarity based on appearance only, not on semantics.

As we discussed above, a number of applications can make use of such metrics and each application imposes different requirements on metric performance. Thus, in image compression it is important to provide a monotonic relationship between measured and perceived distortion. In contrast, in CBIR applications it may be sufficient to distinguish between similar and dissimilar images, while the precise ordering may not be important. In some cases, as in the compression application, it is important to have an absolute similarity scale, while in others a relative scale may be adequate.

In the remainder of this thesis, we first review existing image quality and similarity metrics (Chapter 2). We examine different grayscale similarity metrics, ranging from the basic mean squared error to the more sophisticated Structural Similarity Metrics [**16**], as well as some of the texture-specific similarity measures, such as statistical methods of Ojala *et al.* [**17**], and the spectral methods of Do *et al.* [**18**]. We also present an overview of color similarity metrics, as well as methods to combine grayscale and color similarity into a unified similarity score.

Chapter 3 presents an in-depth look at the proposed grayscale texture similarity metrics. The proposed metrics rely on the complex steerable pyramid wavelet decomposition of images, and compare different local statistics in subbands, as opposed to finding point-by-point differences in two texture images. A detailed overview of the original Structural Texture Similarity Metric (STSIM) is given, followed by the improvements incorporated into the proposed STSIM2. A discussion of the necessity of utilizing a complex wavelet

transform as opposed to the real wavelet transform is also presented, as well as the analysis of the effects of different window sizes. Finally, two alternatives for computing a final grayscale similarity score are proposed.

Chapter 4 describes the proposed color texture similarity metric. The algorithm relies on knowledge about human perception and utilizes results from image segmentation and existing color image similarity metrics into a novel color texture similarity algorithm. In addition, we introduce a method for removing the *structure* from a color texture image, i.e., generating a generic texture image that maintains the original color composition.

Before proceeding with the procedures for testing the performance of the grayscale and color composition texture similarity metrics and the experimental results, in Chapter 5, we take a careful look at the problem of texture similarity, the fundamental assumptions about the underlying signals, the capabilities of human perception, and the requirements of the intended applications. Based on the analysis of the results of an initial set of experiments, we conclude that the color composition and grayscale texture similarity problems should be considered separately, and separate subjective experiments should be designed to test the performance of the metrics described in Chapters 3 and 4. More importantly, using human perception as a guide, we examine what is achievable and what is not, in general and in the context of different applications. One of the most significant conclusions is that it does not make sense to quantify similarity when textures are dissimilar. Thus, in CBIR, it suffices to distinguish between similar and dissimilar textures, while in image compression, quantifying similarity should be limited to the high end of the similarity scale.

Chapter 6 presents the details of the texture database generation, the three main different experimental setups, and the metric performance evaluation criteria. This chapter also introduces a new procedure for conducting subjective tests, the Visual Similarity by Progressive Grouping (ViSiProG), which facilitates evaluation and development of the newly proposed metrics. The goal is to label each pair of textures in a database as similar of dissimilar for CBIR applications. ViSiProG accomplishes this goal by organizing textures into similar groups, thus avoiding a huge number of texture-to-texture comparisons.

Finally, the experimental results are presented in Chapter 7. We show that the proposed grayscale and color similarity metrics perform well in every experiment, outperforming the competitors. The final conclusions are drawn in Chapter 8, which also contains a brief section on the possible avenues for the extension of the presented work.

1.1. Contributions

The major contributions of this thesis are summarized as follows.

Grayscale and Color Texture Similarity Metrics

We developed two novel texture similarity metrics, one for comparing the grayscale structure of texture images, and one for comparing their color composition, independent of structure. We have argued that color and grayscale structure are attributes that correspond to different perceptual dimensions of texture, and as such, should be considered separately, at least for metric development and testing. The appropriate combination of the two metrics is left to the end-user and the details of the target application. The advantages of the proposed metrics are demonstrated by the fact that texture CBIR (known

item search and retrieval of similar textures) based on these metrics is significantly better than when it is based on other metrics in the literature. The same is true for the evaluation of the severity of compression artifacts.

Methodology for Systematic Testing of Texture Similarity Metrics

We examined the fundamental assumptions about the underlying signals (texture images), human perception of textures, and the intended applications (CBIR, image compression), which resulted in a better understanding of the texture similarity problem, what can or cannot be achieved, and a new methodology for conducting subjective and objective tests for the evaluation of the performance of the proposed texture similarity metrics in the context of each application. This set the stage, not only for this thesis research, but also for future research in the area of texture similarity metric development and evaluation.

Visual Similarity by Progressive Grouping (ViSiProG)

Visual Similarity by Progressive Grouping (ViSiProG) is a new subjective testing procedure that we designed to facilitate evaluation and development of the newly proposed metrics. Its focus is on content-based image retrieval applications, where the goal is to label each pair of textures in a database as "similar" or "dissimilar," as judged by human subjects. Since human subjects do not always agree, some pairs may be labeled as "uncertain." The input to ViSiProG is a large collection of texture images and its output is a label for each pair of textures. ViSiProG accomplishes this goal by organizing textures into similar groups, thus avoiding a huge number of texture-to-texture comparisons. Experimental results with grayscale textures (grouped according to overall visual similarity)

and color textures (grouped according to similarity of color composition) demonstrate that ViSiProG succeeds in forming perceptually similar groups. ViSiProG can be used in any task that requires grouping according to visual similarity.

Systematic Subjective Tests for Image Compression Applications

We have developed algorithms for deforming texture images with different types and levels of distortions to be used in metric development and testing. Experimental results with a variety of textures demonstrate that the algorithm can successfully generate distortions with perceptually increasing severity across each type of distortion.

Comprehensive Texture Database

For the purposes of performing extensive performance tests in the context of the above-cited applications, we collected a large number (approximately 1500) of color texture images. The images were carefully selected to meet the fundamental assumptions about the texture signals and the intended applications. The key criteria include (a) each image represents a uniform texture and (b) the database contains examples of "identical," similar, and dissimilar textures. The variety of informative results obtained with subjective and objective tests is a strong indication of the relevance and completeness of the database.

CHAPTER 2

Review of Image Similarity Metrics

The focus of this thesis is on texture similarity metrics for color texture images. This chapter presents an overview of existing image similarity metrics, which also include a few algorithms that were specifically design for the problem of texture similarity.

The metrics that evaluate only grayscale similarity and the metrics that evaluate only color similarity between two images will be presented separately. Depending on the target applications for the metrics, if the goal is to determine overall similarity of color images, one approach is to design two separate grayscale and color metrics and to combine their results. Another common alternative is to apply one of the grayscale metrics to different color channels and pool the per-channel scores together to form a single similarity value. These two alternatives will be discussed further in the last section of this chapter, as well as in Chapter 4.

2.1. Grayscale Image Similarity Metrics

Grayscale similarity metrics differ in their design based on the applications they are developed for. For the purposes of image compression, similarity metrics – more commonly referred to as quality or fidelity metrics – try to evaluate the fidelity of the coded image with respect to the original data. The main goal is to determine how visible the compression artifacts are and what the overall perceived quality of the compressed image is. On the other hand, image similarity metrics aimed at content-based image retrieval

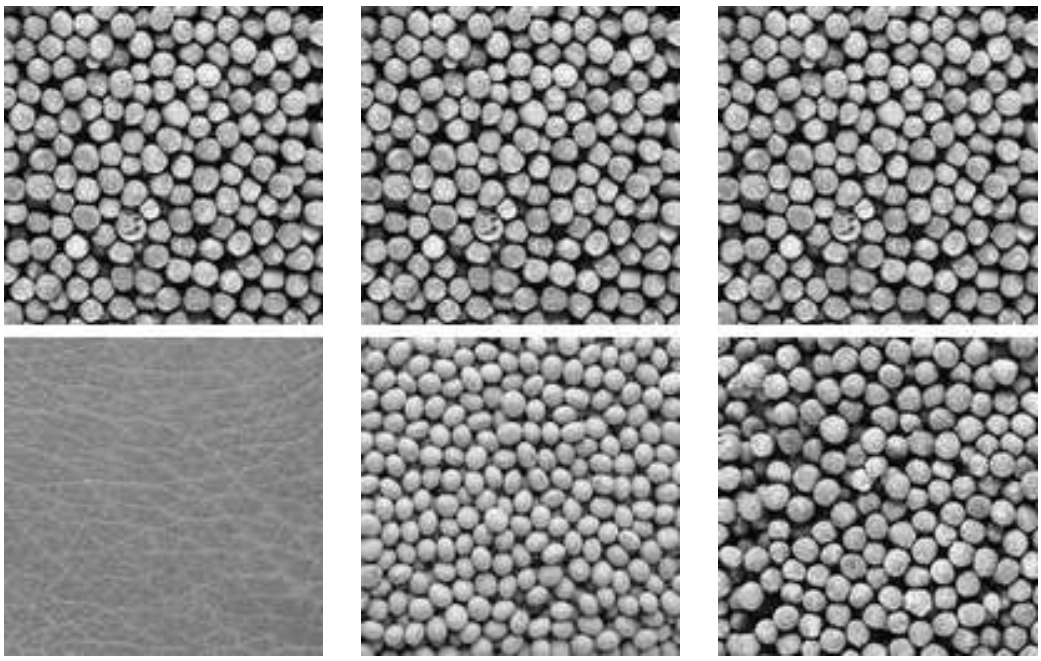
applications analyze the content of the images and compare it to the query, without necessarily making any quality judgments. The texture similarity metrics we consider in this thesis fall somewhere between these two categories, as they are intended for either of the two applications. In this section, we examine different grayscale similarity metrics, and discuss their applicability to texture images.

2.1.1. Mean Squared Error (MSE)

Point-by-point comparisons are the simplest image metrics that reflect the differences in how images are represented in a computer, rather than how those differences are perceived by humans. Metrics such as mean squared error (MSE) and peak signal-to-noise ratio (PSNR) have been shown to poorly correlate with human perception of images [**19**, **20**]. Such measurements of similarity do not take into consideration models of the human visual system, and can only be used for limited applications. Yet, MSE-based metrics are still the most commonly used metrics for evaluating different compression algorithms, even though they do not always give appropriate image quality scores. This is particularly true for textured regions of images, as illustrated in Figure 2.1, where PSNR orders the pairs from best-to-worst from left to right, while the opposite ordering would be correct for the perceived texture similarity.

2.1.2. Low-Level versus High-Level Perceptual Image Quality Metrics

The main idea behind *perceptual image quality metrics* is to penalize image differences according to how visible they are [**3**, **21**]. Such metrics are typically based on a multiscale frequency decomposition and incorporate low-level human vision characteristics, such as



(a) PSNR = 11.3dB (b) PSNR = 9.0dB (c) PSNR = 8.2dB

Figure 2.1. Illustration of inadequacy of PSNR for texture similarity. Subjective similarity increases left to right, while PSNR indicates the opposite.

a contrast sensitivity function (CSF), luminance masking, and contrast masking. Overall, they have a better correlation with human judgments of similarity than MSE-based metrics.

However, they can still be considered as point-by-point metrics, as they aim at "perceptual transparency" and "perceptually lossless" compression applications, and are very sensitive to any image deviations that can be detected by the eye. Thus, image differences due to zooming, small rotations or translations, and contrast changes, that are detectable but do not affect the overall quality of an image are heavily penalized by such metrics. To accommodate such deviations and to provide better predictions of perceived image

similarity, metrics need to incorporate high-level HVS characteristics; this can be done either explicitly or implicitly.

2.1.3. Texture Similarity Metrics

Metrics that evaluate texture similarity can be broadly grouped into two categories: statistical and spectral methods [**22**, **23**]. The statistical methods are based on calculating certain statistical properties of the gray-levels of the pixels in an image, while the spectral methods utilize Fourier spectrum or filtering results to characterize and compare textures.

The statistical methods include co-occurrence matrices [**24**], first and second order statistics [**25**], random fields models [**26**], local binary patterns [**27**] etc.

Perhaps the best-known is the method of co-occurrence matrices [**24**]. Co-occurrence matrices employ relationships between adjacent pixels, like Haralick's features, calculating the differences in luminance values within a small neighborhood, usually 2×2. Co-occurrence matrices have found application in CBIR systems [**28**], medical image analysis [**29**], as well as object detection applications [**30**]. However, given its usually limited neighborhood, this method would not be appropriate for computing similarity of textures other than the so-called microtextures [**31**].

Chen *et al.* [**25**] used the local correlation coefficients for texture segmentation applications. However, Julesz *et al.* [**32**, **33**] have shown that humans can easily discriminate some textures, that may have the same global second-order statistics, thus utilizing only these statistics is not enough for evaluation of perceptual texture similarity.

Kashyap *et al.* [**26**] utilize random field as the underlying model for the realization of pixel values in a texture image. Each pixel is characterized by the probability distribution,

given the other pixels in its vicinity. This method is used for texture classification, where the image is assigned to the class for which it yields the highest probability of realization, given the underlying model associated with that class. It does not, however, compute similarity scores between two different texture images.

Ojala *et al.* [**17**] propose utilizing Local Binary Patterns (LBP) to characterize textures, mainly for the retrieval applications. This method applies circular kernels of different sizes and rotations, analyzing the local neighborhoods of each pixel by performing binary operations on the central pixel and the pixels in the kernel. The results of these operations over the whole image are then pooled together into a histogram, with kernels of different sizes producing each a different histogram. This method was applied to texture classification problem, where two images are compared using their respective histograms, computing a log-likelihood statistic that the two images come from the same class. This method is very simple yet effective for the task of classifying textures, however, it cannot be effectively used for determining similarity between two textures, as will be shown in Chapter 7.

The statistical methods are appealing mostly because of their simplicity and the speed with which the features can be computed and comparisons can be carried out. However, most of these methods have been applied to a very limited set of applications – segmentation and texture classification – and without incorporating any HVS properties, they would likely fail in other setups, assuming that they in fact can be computed outside of their own original problem settings.

The spectral methods provide a better link between the pixel representation of images, and the way that humans perceive images and similarities between them. Initially, the

spectral methods were based on the Fourier transform of the images, but given that the basis functions for Fourier analysis do not provide efficient localization of texture features [**34**], they were quickly replaced by wavelet analysis methods. This way, HVS characteristics may be directly used in the texture similarity metrics by decomposing the images using wavelet filter banks that model explicitly the human visual system.

The idea that has been utilized the most in the spectral methods [**31**, **35**–**38**] is to extract the energies of different subbands (outputs of filter banks) and use them as features for texture segmentation, classification or for content-based image retrieval.

For example, Unser [**31**] proposes using the wavelet frames for decomposing the images, and to perform texture classification using the minimum error Bayes classifier of the feature vectors (that are composed of energies of subbands), or, for segmentation purposes, to cluster all the vectors in an image to obtain a segmentation map.

Do *et al.* [**35**] also adopt the features extracted from the wavelet coefficients, but show that the distribution of the wavelet coefficients is better modeled as generalized Gaussian density, which requires the estimation of two parameters (as opposed to only one – the variance – which assumes the Gaussian density). The extracted features are compared using the Kullback-Leibler distance between the two feature vectors, and this method has been shown to perform better than the method of Unser [**31**].

However, some methods choose to explicitly model the HVS characteristics. An example of this explicit utilization is the use of filter banks that are orientation-sensitive, mimicking the orientation selectivity of simple receptive fields in the visual cortex of the higher vertebrates [**39**]. Gabor filters are one example of such filter banks and they are thought to describe well the first stages of image processing in humans filters [**40**, **41**].

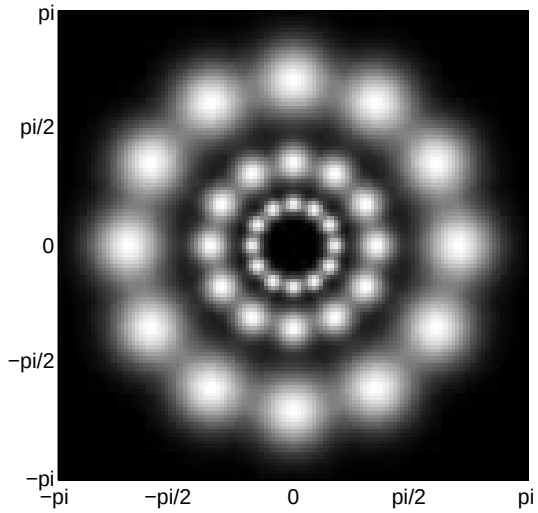


Figure 2.2. Gabor filter bank

One of the pioneering works was done by Clark *et al.* [**38**], where Gabor filters were used for texture segmentation. The frequency domain covered by Gabor filters in 3 scales and 6 orientations is given in Figure 2.2.

The main idea is to filter the image with Gabor wavelets and take the mean and standard deviation of each subband as a feature. Features extracted from Gabor subbands have found various applications, as in CBIR [**36**], steganography [**42**,**43**], object detection [**44**], medical image segmentation [**45**], texture similarity and classification [**46**, **47**] and so on.

Some methods for evaluating texture similarity combine the statistical and the spectral approaches, in order to increase the performance of either. For example, Yang *et al.* [**48**] combine Gabor features and co-occurrence matrices for the CBIR applications. Also, popular standards like MPEG7 contain texture descriptors, in order to make video

coding and retrieval more efficient. The three variations of texture descriptors used in MPEG-7 – the Homogeneous Texture Descriptor, Edge Histogram Descriptor and Texture Browsing Descriptor – are described in detail in [**49**]. An overview of these descriptors can also be found in [**50**]. The Homogeneous Texture Descriptor is a vector of means and variances of a Gabor filtered image, and in this sense incorporates HVS properties. It is useful in characterizing images that contain homogeneous texture patterns. The Edge Histogram Descriptor partitions the image into 16 blocks, applies edge detection algorithms and computes local edge histograms for different edge directions. This descriptor is said to work well in image similarity assessment, in cases when images contain non-homogeneous textures, since the edges are descriptive clues for image perception. The Texture Browsing Descriptor characterizes regularity, directionality and coarseness of the texture image, which is related to high-level human perception of images. It is useful for coarse classification of textures. Different algorithms have been proposed for calculating these descriptors and their efficiency has been examined for the purposes of texture image retrieval [**17**, **51**]. Ojala *et al.* [**17**] have shown that these descriptors are rather limited and can be used only for a very crude texture retrieval task. The variations of techniques used in MPEG-7 exist in other applications like CBIR [**52**], where different edge detectors are used.

Even though some of these methods have been shown to have very good texture clustering or segmentation abilities, little has been done in evaluating the subjective, perceptual texture similarity. There is still an unfilled void that requires the design of the algorithms that would be able to compare two texture images by assigning them a similarity score, which would be in accordance with perceptual texture similarity, as well

as usable for more than one application. The method proposed in Chapter 3 attempts to achieve this.

2.1.4. Structural Similarity Metrics

A class of metrics that attempt to – implicitly – incorporate high-level properties of the HVS, are the Structural Similarity Metrics (SSIMs) [**16**]. The main idea is to compare local image statistics in corresponding sliding windows in the two images and to pool the results spatially. SSIMs can be applied in either the spatial or transform domain. They adapt to lighting changes and have implicit masking capabilities, but no explicit HVS models, such as a contrast sensitivity function and contrast or luminance masking. When implemented in the complex wavelet domain, they are tolerant of (i.e., they do not penalize) small spatial shifts (and as a result also small rotations or zoom), but only up to a few pixels. Wang *et al.* [**53**] used the complex steerable pyramid, which is an overcomplete wavelet transform [**10**]. Complex steerable pyramids, like Gabor filters, are inspired by biological visual processing and have nice properties, such as translation-invariance and rotation-invariance, as claimed by Portilla and Simoncelli [**54**]. The complex-wavelet implementation of SSIM is denoted as CWSSIM. A schematic representation of the subbands is given in Figure 2.3. The locations of subbands in the frequency plane are given for three scales (S1, S2, S3) and four orientations. Also, this decomposition produces the residual low-pass (LP) and high-pass (HP) bands.

Whether implemented in the original image domain (SSIM) or in the complex wavelet domain (CWSSIM), the algorithms consist of three terms that compute and compare image statistics in corresponding sliding windows in the two images: luminance, contrast,

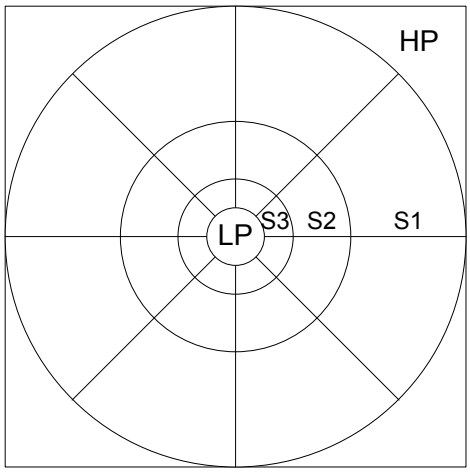


Figure 2.3. Steerable filter bank; the axes ranges are $[-\pi, \pi]$ in both vertical and horizontal direction

and "structure." The luminance term compares the mean of intensities within each window, the contrast term compares the standard deviations, and the structure term depends on the cross-correlation between the two windows.

We now define the metrics more formally. First, we establish the following notation, which will be used throughout this thesis:

- $\mathbf{x}$ and $\mathbf{y}$ are two images to be compared
- spatial indices of pixel values (or coefficients, in transform domain) are denoted by (u, v) or (i, j); (u, v) usually denotes the center of a sliding window, while (i, j) are the coordinates of the coefficients within a sliding window
- W is the local neighborhood of size $w \times w$, centered at (u, v)
- in case of subband analysis, the bands are denoted by m and n
- for image subbands: the subband index is in the superscript

- for image statistics: the superscript denotes the subband, the subscript denotes the image
- all variables of type C_n are small constants.

The various terms for calculating the SSIM and CWSSIM metrics are defined as follows:

$$\mu_{\mathbf{x}}^m(u,v) = \frac{1}{w^2} \sum_{(i,j)\in W} \mathbf{x}^m(i,j) \tag{2.1}$$

$$\sigma_{\mathbf{x}}^m(u,v) = \sqrt{\frac{1}{w^2-1} \sum_{(i,j)\in W} (\mathbf{x}^m(i,j) - \mu_{\mathbf{x}}^m(u,v))^2} \tag{2.2}$$

$$\sigma_{\mathbf{xy}}^m(u,v) = \frac{1}{w^2-1} \sum_{(i,j)\in W} (\mathbf{x}^m(i,j) - \mu_{\mathbf{x}}^m(u,v))(\mathbf{y}^m(i,j) - \mu_{\mathbf{y}}^m(u,v)) \tag{2.3}$$

To simplify the notation, we will drop the spatial coordinates (u, v). It will be assumed that all the terms are computed in the corresponding local sliding windows, centered at (u, v). We will also use "SSIM" to denote both the image domain and complex wavelet domain implementations; it will be understood that in the image domain implementation there are no subbands, and thus, the subband index m should be dropped and any summations over subbands should be eliminated. The luminance term is defined as:

$$l_{\mathbf{x},\mathbf{y}}^m = \frac{2\mu_{\mathbf{x}}^m \mu_{\mathbf{y}}^m + C_0}{(\mu_{\mathbf{x}}^m)^2 + (\mu_{\mathbf{y}}^m)^2 + C_0}, \tag{2.4}$$

the contrast term is defined as:

$$c_{\mathbf{x},\mathbf{y}}^m = \frac{2\sigma_{\mathbf{x}}^m \sigma_{\mathbf{y}}^m + C_1}{(\sigma_{\mathbf{x}}^m)^2 + (\sigma_{\mathbf{y}}^m)^2 + C_1}, \tag{2.5}$$

and the structure term is defined as:

$$s^m_{\mathbf{x},\mathbf{y}} = \frac{\sigma^m_{\mathbf{xy}} + C_2}{\sigma^m_{\mathbf{x}}\sigma^m_{\mathbf{y}} + C_2}. \tag{2.6}$$

For each sliding window in each subband, a similarity value is computed as:

$$Q^m_{\mathrm{SSIM},\mathbf{x},\mathbf{y}} = (l^m_{\mathbf{x},\mathbf{y}})^\alpha (c^m_{\mathbf{x},\mathbf{y}})^\beta (s^m_{\mathbf{x},\mathbf{y}})^\gamma. \tag{2.7}$$

Usually, the parameters are set to be $\alpha = \beta = \gamma = 1$ and $C_2 = C_1/2$ to get:

$$Q^m_{\mathrm{SSIM},\mathbf{x},\mathbf{y}} = \frac{(2\mu^m_{\mathbf{x}}\mu^m_{\mathbf{y}} + C_0)(2\sigma^m_{\mathbf{xy}} + C_1)}{((\mu^m_{\mathbf{x}})^2 + (\mu^m_{\mathbf{y}})^2 + C_0)((\sigma^m_{\mathbf{x}})^2 + (\sigma^m_{\mathbf{y}})^2 + C_1)}. \tag{2.8}$$

Typically, the SSIM is evaluated in small sliding windows (e.g., 7×7) and the final metric is computed as the average of $Q^m_{\mathrm{SSIM},\mathbf{x},\mathbf{y}}$ over all spatial locations and all subbands. The size of the window affects the metric in the sense that as it becomes smaller, it becomes closer to point-by-point comparisons and as it grows, it becomes more of a global structure metric.

If implemented in the image domain, this metric is highly sensitive to image translation, scaling, and rotation, as shown in [**53**]. This is to some extent remedied by implementing the SSIM metric in the complex wavelet domain. By utilizing an overcomplete transform such as the steerable pyramid, small spatial translations affect to a lesser degree the metric value. Thus, CWSSIM is invariant to luminance and contrast changes as well as spatial translations, as proven by Wang and Simoncelli [**53**]. The key is in the fact that these distortions lead to consistent magnitude and phase changes of local wavelet coefficients. The structural information of local image features is mainly contained in

the relative phase patterns of the wavelet coefficients and a consistent phase shift of all coefficients does not change the structure of the local image feature [**53**].

Finally, we should mention that Brooks and Pappas [**55**] proposed a perceptually weighted metric (W-CWSSIM), which incorporates the noise sensitivities of the different subbands, for a given display resolution and viewing distance. The W-CWSSIM thus provides a link between the perceptual metrics we discussed in Section 2.1.2 and the SSIM-type approaches. The perceptual weighting is useful for measuring distortions that dependent on viewing distance, such as white noise and DCT compression [**55**].

The idea of using structural similarity metrics for evaluating texture similarity comes as a direct consequence of their definition: computing *structural*, as opposed to point-by-point, similarity of images. To what extent this is true with the original SSIM and CWSSIM metrics, and how the SSIM ideas can be extended to address the peculiarities of the texture similarity problem, will be discussed in Chapter 3, where the grayscale texture similarity metric proposed in this thesis is presented.

2.2. Color Image Similarity Metrics

Color is perhaps the most expressive of all the visual features and has been extensively studied in image retrieval research during the last decade [**11**]. The simplest methods for describing and comparing the color content of different images is to produce color histograms with a fixed color codebook [**56**], which are then compared by simple L_p distance, histogram intersection metrics [**56**], or more sophisticated color quadratic distance [**57**]. These methods are fast, easy to implement and, as was the case for determining grayscale similarity, reflect the image representation in computers, rather than how the colors and

their differences are perceived by humans. In fact, as shown in [**11**] and [**12**], the human vision system is not designed to distinguish well between similar colors. Studies have shown that humans cannot simultaneously perceive a large number of colors present in one image. As pointed out in [**15**], the human visual system extracts chromatic features from the image, as opposed to recording the colors point-by-point, and in effect only sees a few "dominant colors." Thus, color descriptors and comparison methods need to move from direct histogram comparisons to more sophisticated techniques that account better for the HVS properties.

The color descriptors developed by Manjunath *et al.* [**11**] for the MPEG-7 standard contain 3 sets: Histogram Descriptors (Scalable Color and Color Structure), Dominant Color Descriptors and Color Layout Descriptors. The histograms descriptors are based on quantized images and an L-norm is used to compute the distance. This is argued to be adequate for natural images, since they tend not to have discrete color histograms and there is high redundancy between adjacent histogram bins. Dominant Color Descriptor extraction is explained in detail in [**58**]. First, an image is segmented using the edgeflow algorithm [**59**], then colors are clustered within each segment by using a modified generalized Lloyd algorithm proposed in [**60**]. The clustering algorithm consists of pre-processing of the images to remove noise and smooth images and iteratively breaking up clusters and re-assigning their elements. Clustering is performed with respect to the smoothness of the regions – colors are coarsely quantized in the detailed regions, since the human eye is more sensitive to lighting changes in smooth regions. After clustering, the Dominant Color Descriptor is constructed from the cluster centroids and the corresponding percentages of pixels belonging to the clusters. The distance between two descriptors is similar to the

quadratic color distance [**57**]. Finally, the Color Layout Descriptor is designed to capture the spatial distribution of colors, which is adequate for scribble-based image retrieval. The feature extraction process is done in two steps. First, the image is partitioned into 64 blocks (8×8) and the average color for each block is computed. This results in an 8×8 matrix of local means. Then, an 8×8 DCT is applied and a few low frequency coefficients are chosen by the zigzag method. This descriptor is said to be compact yet efficient.

Although these descriptors might be appropriate for compact and fast image and video retrieval algorithms, as we argued above, histogram representations lack discriminatory power in retrieval of large image databases and do not match human perception [**12**]. Mojsilovic *et al.* have also shown that if two patterned images have similar Dominant Color Composition, they shall be perceived as similar by humans even if their content, directionality, placements or repetitions of structural elements are not the same [**61**]. This is the basis for an extraction algorithm of perceptually important colors, as developed by Mojsilovic *et al.* [**12**].

In [**61**], the chosen working colorspace is L*a*b*, since the CIELAB (or L*a*b*) and CIELUV (L*u*v*) colorspaces exhibit perceptual uniformity, meaning that the Euclidean distance separating two similar colors is proportional to their visual difference [**62**, **63**]. However, we should note that the Euclidean distances in these colorspaces are not linearly proportional to the visual judgment when the colors are dissimilar; that is, the perceptual uniformity is limited to small neighborhoods in color space. Thus, and this is one of the major contributions of this thesis, such distance metrics are only suitable for image

retrieval applications when a task is finding images with similar color compositions. Otherwise, all the metrics can do is differentiate between "similar" and "dissimilar" colors. Quantifying the degree of color differences when the colors are very distinct is a difficult, if not impossible, task even for humans and it can be shown that human subjects are inconsistent in their judgments of differences between dissimilar colors. In Chapter 5, we will draw similar conclusions for grayscale and color textures, and such conclusions will be of critical importance for the experimental design and methodology on which this thesis is based.

In the method of Mojsilovic *et al.* [**61**], the colors in the image are first quantized into m colors according to a developed codebook. Given the non-linearity of the CIELAB space, this codebook is not a simple uniform quantization of the colorspace, but rather uniform sampling of chromaticity planes in the L*a*b* space. Then, the image is divided into $N \times N$ subimages (N typically being 20), and for each subimage, a Neighborhood Color Histogram (NCH) matrix is computed. NCH matrix contains information about the relative occurrence of pixels of color c_j within a small $D \times D$ neighborhood of all the pixels of color c_i. Depending on the ratio of occurrence of the same color c_i and the occurrence of the color that occurs most around c_i (and being different than c_i), c_r, all pixels of color c_i are either kept as perceptually important, or they are marked as speckle noise and remapped to c_r. Finally, the remaining colors from all subimages are pooled together and each color that occupies more than a predefined area percentage is determined to be a dominant color. Typically 3–10 dominant colors are detected in each image.

This approach of extracting the perceptually important colors is somewhat similar to the well-known Color Correlogram method [**64**]. It differs in the sense that the correlogram captures the information about frequency of color c_j occurring at exactly distance k from color c_i, while NCH computes the probability of the color occurring *within* a neighborhood. NCH is thus more suitable for removing speckle noise.

A method proposed by Birinci *et al.* [**65**] combines the dominant color approach and correlogram calculations. They named their approach Perceptual Color Correlogram, since it extracts dominant colors, in accordance with human perception, and they utilize a weighted metric of L-type to compare dominant colors and correlograms of two images.

When the color information is extracted from the image, the next question is how to compute the distances between two color signatures. Birinci *et al.* utilize a combination of L_1 and L_2 metrics for determining similarity between dominant colors and a modified L_1 norm for correlogram distances. Huang *et al.* [**64**] utilize the simple L_1 distance measure. Manjunath *et al.* [**11**] use the quadratic color histogram to compute distances between dominant colors. However, as shown in [**66**], the metric that has superior classification and retrieval results with compact representation is the Earth Mover's Distance.

2.2.1. Earth Mover's Distance (EMD)

The Earth Mover's Distance [**67**] is based on the minimal cost that must be paid to transform one distribution into another. Informally speaking, the Earth Mover's Distance (EMD) measures how much work needs to be applied to move earth distributed in piles p_x so that it turns into the piles p_y.

This can be formalized as a linear optimization problem: Let's denote two images as $\mathbf{x}$ and $\mathbf{y}$ and their representative color compositions $C_{\mathbf{x}}$, $C_{\mathbf{y}}$, as $C_{\mathbf{x}} = \{(c_{x1}, p_{x1}), ...(c_{xm}, p_{xm})\}$ and $C_{\mathbf{y}} = \{(c_{y1}, p_{y1}), ...(c_{yn}, p_{yn})\}$, where the c-elements denote the colors and the p-elements their respective percentages within the image. The colors and their percentages can be represented either as simple histograms, or as dominant colors. If we denote by $\mathbf{D} = [d_{i,j}]$ the set of distances between colors (c_{xi}, c_{yj}) (which is the L_2 distance in this case) and by $\mathbf{F} = [f_{i,j}]$ the set of all possible *flow* mappings between colors (c_{xi}, c_{yj}) (how much of color c_{xi} gets "transported" to color c_{yj}), then the problem can be stated as:

$$\min_{\mathbf{F}} \frac{\sum_{i,j} d_{i,j} f_{i,j}}{\sum_{i,j} f_{i,j}} \tag{2.9}$$

subject to:

$$f_{i,j} \geq 0 \qquad 1 \leq i \leq m,\ 1 \leq j \leq n \tag{2.10}$$

$$\sum_{j=1}^{n} f_{i,j} \leq p_{xi} \qquad 1 \leq i \leq m \tag{2.11}$$

$$\sum_{i=1}^{m} f_{i,j} \leq p_{yj} \qquad 1 \leq j \leq n \tag{2.12}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} f_{i,j} = \min \left(\sum_{i=1}^{m} p_{xi}, \sum_{j=1}^{n} p_{yj} \right). \tag{2.13}$$

These conditions can be explained by looking at the informal problem of earth transportation between centers $C_{\mathbf{x}}$ and $C_{\mathbf{y}}$. Assume that we want to move *from* each center c_{xi} at most p_{xi} amount of earth and we want to put *in* each center c_{yj} at most p_{yj} of earth. The condition given in (2.10) means we cannot have "negative transportation,"

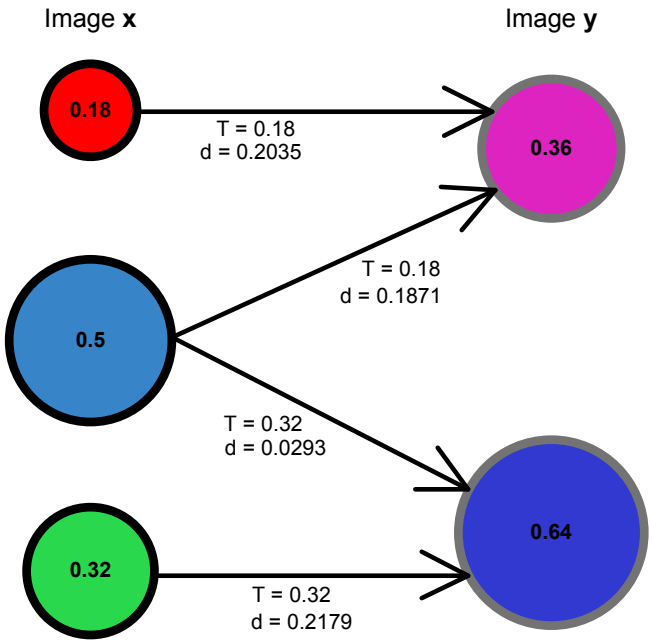


Figure 2.4. EMD example of color matching

i.e., transportation between c_{xi} and c_{yj} can only go $c_{xi} \rightarrow c_{yj}$. The condition in (2.11) means that we cannot take out of c_{xi} more than there is inside; the condition in (2.12) means that we cannot put in c_{yj} more than it can receive; the last condition (2.13) means that the maximum transportation cannot exceed the sending or receiving capacities.

Example 1. This is an example to illustrate how EMD works. The reference image is given in Figure 2.4. The color composition of image **x** (black bordered circles) is given in Table 2.1, while the color composition of image **y** (gray bordered circles) is given in Table 2.2.

c_X	R	G	B	p_X
c_{x1}	56	132	201	0.5
c_{x2}	41	216	77	0.32
c_{x3}	255	0	0	0.18

Table 2.1. Color composition, image $\mathbf{x}$

c_Y	R	G	B	p_Y
c_{y1}	49	57	208	0.64
c_{y2}	221	36	193	0.36

Table 2.2. Color composition, image $\mathbf{y}$

The L_2 distances between colors (normalized to have a maximum distance of 1) are given in Table 2.3.

d_{XY}	c_{y1}	c_{y2}
c_{x1}	0.0293	0.1871
c_{x2}	0.2179	0.4012
c_{x3}	0.4560	0.2035

Table 2.3. Differences of colors from image $\mathbf{x}$ and image $\mathbf{y}$

The computed color matching via EMD is given in Figure 2.4; the amounts transported and distances between colors are given above the arrows connecting the circles. We can see that EMD followed the intuition of connecting blue with blue and also that pink gets associated with blue and red, instead of green. The total cost for the matching operations is $EMD(C_{\mathbf{x}}, C_{\mathbf{x}}) = 0.1494$.

2.2.2. Optimal Color Composition Distance (OCCD)

An approach that follows the same philosophy as EMD is Optimal Color Composition Distance (OCCD) developed by Mojsilovic *et al.* [**12**]. In this case, the color composition

descriptors are the extracted dominant colors and their respective percentages. The dominant color components of each image are quantized into a set of n color units and each color unit represents the same percentage p, i.e., $np = 100$. Each image is represented with n units, and each unit is labeled with a color value; percentages are not needed anymore since the number of units with the same color label is proportional to the percentage of that color over the whole image. The problem is now transformed into a minimum cost graph matching problem – the bijective matching between two sets of n units. Let the units from one image be denoted as $C_{\mathbf{x}} = \{c_{x1}, ..., c_{xn}\}$ and $C_{\mathbf{y}} = \{c_{y1}, ..., c_{yn}\}$ and let $m_{\mathbf{xy}}$ be a bijective function that maps the set $C_{\mathbf{x}}$ onto the set $C_{\mathbf{y}}$, $\{m_{\mathbf{xy}} : C_X \rightarrow C_Y\}$; also denote the distance between two colors (c_x, c_y) as $d(c_x, c_y)$. The problem can be formalized as minimizing the sum of distances with respect to the mapping function $m_{\mathbf{xy}}$:

$$\min_{m_{\mathbf{xy}}} \sum_{i=1}^{n} d(c_{xi}, m_{\mathbf{xy}}(c_{xi})). \tag{2.14}$$

Example 2. Using the same color compositions as for EMD example (Ex.1), we can quantize the colors with, e.g., 5% steps, yielding the following $n = 20$ units for each image:

- For $C_{\mathbf{x}}$: 10 units of c_{x1}, 6 units of c_{x2}, 4 units of c_{x3}
- For $C_{\mathbf{y}}$: 13 units of c_{y1}, 7 units of c_{y2}

OCCD tries to find the best match between the units so that the sum of distances is minimized. The problem is depicted in Figure 2.5.

After applying the minimum cost matching algorithm, the solution is similar to the EMD example in the sense that the association of colors between images stayed the same. However, due to the quantization of percentages, the cost is not equal to the one in EMD example: $OCCD(C_{\mathbf{x}}, C_{\mathbf{y}}) = 0.1444$ (compared to $EMD(C_{\mathbf{x}}, C_{\mathbf{y}}) = 0.1494$). On the other

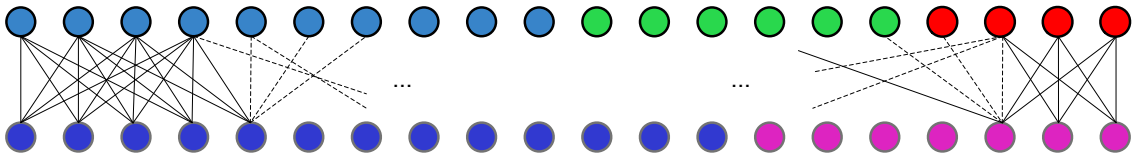


Figure 2.5. OCCD problem statement

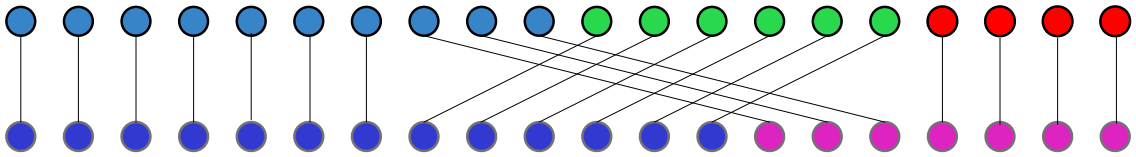


Figure 2.6. OCCD problem solution

hand, if we use 1% units, we will obtain the same matching cost, which means that these metrics are essentially the same. The difference is that for OCCD, by quantizing the percentages, we turn the problem into a weighted graph matching that can be solved by deterministic algorithms, unlike the linear-programming based EMD calculation. The final result for 5% unit matching is given in Figure 2.6.

The conclusions of the above analysis are that the best color matching results are obtained by utilizing the dominant color descriptors, and a sophisticated metric that incorporates properties of the HSV, like EMD and OCCD do.

2.3. Combining Grayscale and Color Similarity Scores

In the previous sections, we have discussed different methods for determining the grayscale and color similarity of images. However, in cases where the interest is overall image similarity or image quality, the two aspects – grayscale structure and color composition – should be combined to obtain a total similarity score for the two images.

This can be accomplished in different ways. One common approach to represent the texture characteristics and color content of images is to use feature vectors, where each point in the vector represents one trait. Then, the combined texture-color similarity can be computed by taking the distance of the two feature vectors, using some vector distance measure. For example, the distance could be of $\mathcal{L}_p$ type, some kind of histogram intersection measure, e.g., as was done by Park *et al.* [**68**], or a more sophisticated histogram distance measure such as the Earth Mover's Distance [**66**]. This approach doesn't provide any flexibility for balancing grayscale and color similarity, but this can be easily alleviated by applying different weights to different points in feature vectors.

An alternative, straightforward approach that allows the user to put different emphasis on the different similarity components is to linearly combine the results from the color and texture similarity algorithms. This approach has been adopted by a few authors [**52**, **69**–**71**]. However, a potential drawback of this approach lies in the fact that the choice of linear weights for combining grayscale and color similarity measures should be based on the particular application and image database [**69**, **70**]. This suggests that there is no universally optimal way of linearly combining grayscale and color similarity scores that would perform well in any given setup.

Another possibility is to use data fusion algorithms. When the target application is image retrieval, the results of different searches (grayscale-based and color-based search for similar images) can be combined for a final result (e.g., CombMNZ, [**72**]). However, this is only applicable in CBIR applications where the result is in the form of a ranked database and it cannot be used for pairwise image comparisons.

A more elaborate technique for combining grayscale and color similarity is presented in the work by Mojsilovic et al [**61**]. In their algorithm, image similarity is determined according to grammar rules, which are developed based on the analysis of human judgments. Different aspects of grayscale and color similarity are measured independently, such as overall color, color purity, (grayscale) regularity and placement, and (grayscale) directionality, and then combined according to the grammar rules to obtain a single similarity measure. It is shown that people are strongly influenced by the pattern (grayscale texture) similarity; if the pattern similarity is not very emphasized, the next step is determining how similar the dominant colors and directionalities of patterns are; the third level consists of similarity of directionalities of textures, regardless of the color, and the last step is to calculate color composition similarity.

CHAPTER 3

Grayscale Structural Texture Similarity Metrics

The main advantage of the structural similarity metrics we reviewed in Chapter 2, SSIM and CWSSIM [**16**, **53**], is that they attempt to move away from point-by-point comparisons, and instead, to compare the *structure* of the images. On the other hand, the structure term given in (2.6), from which SSIM got its name, is actually a point-by-point comparison [**73**]. This follows from the fact that the cross-correlation between the patches of two images in (2.3), which is the main element of the "structure" term, is computed on a point-by-point basis. As a result, Reibman and Poole [**74**] have shown that the function for computing the image domain SSIM, which uses local means, variances and cross-correlations as arguments, can be worked out so that it uses local means, variances and MSE between two image patches. The CWSSIM, on the other hand, is more tolerant of small shifts (by a couple of pixels) since such perturbations produce consistent phase shifts of the transform coefficients, and they do not change the relative phase patterns that characterize local features in images [**53**]. However, pairs of texture images can have large point-by-point differences and pixel shifts, while still preserving a high degree of similarity.

Thus, the first step towards fully embracing the structural similarity idea of relying on local image statistics, and developing a metric that can address the peculiarities of the texture similarity problem is to completely eliminate point-by-point comparisons by

dropping the "structure" term. As we will see shortly, we will replace it by additional local statistics.

3.1. Structural Texture Similarity Metric - STSIM

The initial effort in establishing a Structural Texture Similarity Metric (STSIM) was done by Zhao *et al.* [**73**]. While following the Wang *et al.* [**53**] approach of comparing local image statistics, Zhao *et al.* proposed removing the structure term (2.6) from the CWSSIM and to use additional subband statistics, that can account for texture characteristics.

The subband statistics that all of the structural similarity metrics so far (SSIM, CWSSIM, and STSIM) compare are the means and variances. On top of those, Zhao *et al.* added the correlations of neighboring subband coefficients, since they can account for certain patterns that characterize texture images.

To remind the reader, the notation is as follows:

- $\mathbf{x}$ and $\mathbf{y}$ are two images to be compared
- spatial indices of transform domain coefficients are denoted by (u, v) or (i, j); (u, v) usually denotes the center of a sliding window, while (i, j) are the coordinates of the coefficients within a sliding window
- W is the local neighborhood of size $w \times w$, centered at (u, v)
- the subbands are denoted by m and n
- for image subbands: the subband index is in the superscript
- for image statistics: the superscript denotes the subband, the subscript denotes the image.

To simplify the notation, the spatial coordinates (u, v) can be dropped, and it is assumed that all the terms are computed in the corresponding local sliding windows, centered at (u, v).

The first order autocorrelation coefficients can be computed as empirical averages, in the horizontal direction as

$$\rho_{\mathbf{x}}^{m}(0,1) = \frac{\frac{1}{w^2}\sum_{(i,j)\in W}(\mathbf{x}^m(i,j) - \mu_x^m)(\mathbf{x}^m(i,j+1) - \mu_x^m)}{(\sigma_x^m)^2} \tag{3.1}$$

in the vertical direction as

$$\rho_{\mathbf{x}}^{m}(1,0) = \frac{\frac{1}{w^2}\sum_{(i,j)\in W}(\mathbf{x}^m(i,j) - \mu_x^m)(\mathbf{x}^m(i+1,j) - \mu_x^m)}{(\sigma_x^m)^2} \tag{3.2}$$

in the diagonal direction as

$$\rho_{\mathbf{x}}^{m}(1,1) = \frac{\frac{1}{w^2}\sum_{(i,j)\in W}(\mathbf{x}^m(i,j) - \mu_x^m)(\mathbf{x}^m(i+1,j+1) - \mu_x^m)}{(\sigma_x^m)^2} \tag{3.3}$$

and in the anti-diagonal direction as

$$\rho_{\mathbf{x}}^{m}(-1,1) = \frac{\frac{1}{w^2}\sum_{(i,j)\in W}(\mathbf{x}^m(i,j) - \mu_x^m)(\mathbf{x}^m(i-1,j+1) - \mu_x^m)}{(\sigma_x^m)^2} \tag{3.4}$$

Note that the full notation would also include the location of the center of the sliding window (u, v), $\rho_{\mathbf{x}}^{m}(-1, 1; u, v)$, which for simplicity we drop.

STSIM used only the horizontal and vertical autocorrelation coefficients, $\rho_{\mathbf{x}}^{m}(0,1)$ and $\rho_{\mathbf{x}}^{m}(1,0)$, since adding the diagonal and anti-diagonal coefficients did not contribute to any significant improvements in the texture similarity metric.

In addition, by utilizing a multi-scale decomposition of the images, higher order correlation coefficients are essentially computed with every decimation of the highest level pyramid. Therefore, for a pyramid with N_S levels, the computed autocorrelations are effectively of order $1, 2, ..., 2^{N_S-1}$.

Another important thing to note is that special attention has to be given to the numerical computation (as empirical average) of the autocorrelation terms $\rho^m_{\mathbf{x}}(0,1)$ and $\rho^m_{\mathbf{x}}(0,1)$. Since real images (or patches) are finite-length, discrete data, in order to correctly compute the correlation coefficients the equations should be rewritten as follows, in the horizontal direction:

$$\rho^m_{\mathbf{x}}(0,1) = \frac{\frac{1}{w^2}\sum_{(i,j)\in W}(\mathbf{x}^m(i,j) - \mu^m_{\mathbf{x}}(u,v))(\mathbf{x}^m(i,j+1) - \mu^m_{\mathbf{x}}(u,v+1))}{\sigma^m_{\mathbf{x}}(u,v)\sigma^m_{\mathbf{x}}(u,v+1)} \tag{3.5}$$

and similarly in the vertical direction:

$$\rho^m_{\mathbf{x}}(1,0) = \frac{\frac{1}{w^2}\sum_{(i,j)\in W}(\mathbf{x}^m(i,j) - \mu^m_{\mathbf{x}}(u,v))(\mathbf{x}^m(i+1,j) - \mu^m_{\mathbf{x}}(u+1,v))}{\sigma^m_{\mathbf{x}}(u,v)\sigma^m_{\mathbf{x}}(u+1,v)}. \tag{3.6}$$

In conclusion, STSIM compares two images by comparing the means (2.1), variances (2.2), and first-order autocorrelations (3.1) and (3.2) in corresponding sliding windows of corresponding subbands.

Note that the means, variances, and autocorrelations are calculated on the *raw*, complex subband coefficients. Since the subband decomposition (apart from the low-pass filtering) does not include the origin of the frequency plane, the subbands will be *zero-mean* over the *whole* image; however, within small windows of size $w \times w$, e.g. 7×7, this does not have to be true; thus, the means $\mu^m_{\mathbf{x}}$ have to be computed in each sliding window, despite the band-pass filtering. Standard deviations $\sigma^m_{\mathbf{x}}$ describe the spectral

power within the sliding window for the given subband and are descriptive features for natural images. The autocovariances $\rho^m_{\mathbf{x}}(0,1)$ and $\rho^m_{\mathbf{x}}(1,0)$ give additional directionality information, for an overall better texture comparison algorithm.

In computing the STSIM metric, the means and variances are compared as in the previously defined luminance (l) and contrast (c) terms in (2.4) and (2.5), respectively. However, similar comparisons cannot be used for the autocorrelation terms, which unlike the variances, are bounded and their values lie in the unit circle of the complex plane. That is because they are computed on complex coefficients of the steerable pyramid subbands. Therefore, a different comparison term was suggested in [**73**]:

$$c^m_{\mathbf{x},\mathbf{y}}(0,1) = 1 - 0.5|\rho^m_{\mathbf{x}}(0,1) - \rho^m_{\mathbf{y}}(0,1)| \tag{3.7}$$

$$c^m_{\mathbf{x},\mathbf{y}}(1,0) = 1 - 0.5|\rho^m_{\mathbf{x}}(1,0) - \rho^m_{\mathbf{y}}(1,0)|. \tag{3.8}$$

These four terms, l, c, $c(0,1)$ and $c(1,0)$ are combined into the Structural Texture Similarity Metric (STSIM) as follows:

$$Q^m_{\mathrm{STSIM},\mathbf{x},\mathbf{y}} = (l^m_{\mathbf{x},\mathbf{y}})^{\frac{1}{4}}(c^m_{\mathbf{x},\mathbf{y}})^{\frac{1}{4}}(c^m_{\mathbf{x},\mathbf{y}}(0,1))^{\frac{1}{4}}(c^m_{\mathbf{x},\mathbf{y}}(1,0))^{\frac{1}{4}} \tag{3.9}$$

Since a different value of $Q^m_{\mathrm{STSIM},\mathbf{x},\mathbf{y}}$ is computed for every window in every subband, the question is how to combine the values for all the subbands. Zhao *et al.* considered two approaches. One approach is the so-called "additive" approach where the total resulting STSIM value is calculated in the same manner as SSIM, by taking the mean across all spatial locations and across all subbands. The other approach is "multiplicative," where corresponding STSIM values for each window get multiplied across the subbands and

then the final metric is calculated as the spatial mean of these multiplied coefficients. The multiplicative approach is depicted in Figure 3.1: if there are N different subbands, at each location, the corresponding STSIM values are multiplied, then the N-th root is taken of each value (so that the numbers do not become too small) and in the end, those values get spatially averaged for the final metric result.

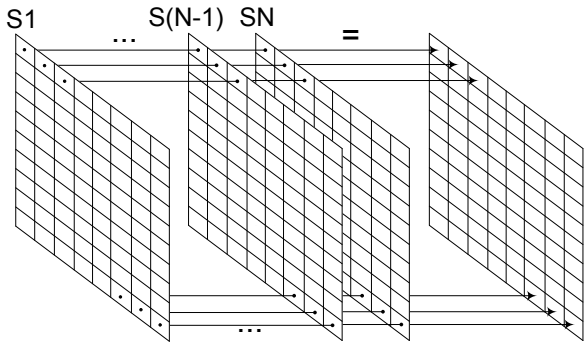


Figure 3.1. Multiplicative computation of STSIM

The STSIM has been shown in [**73**] to perform better, i.e., closer to human judgements of texture similarity than SSIM and CWSSIM.

3.2. Improved Structural Texture Similarity Metric - STSIM2

We now propose a metric that extends the ideas of [**73**] by including a broader set of local image statistics. The motivation comes from the work of Portilla and Simoncelli [**54**]. They have selected a set of subband statistics that can be used for analysis/synthesis of a broad class of textures. Indeed, after extensive experimentation, they claim that the set of statistics they came up with are necessary and sufficient. Now, if a set statistics is good for texture generation, then these statistics should also be suitable as features for texture

comparisons. However, while texture synthesis requires several hundred parameters, we believe that a lot fewer will be adequate for texture similarity.

Among the various statistics that Portilla and Simoncelly proposed, the new STSIM2 metric adopts the cross-correlations between subbands. The argument for adding cross-correlations between coefficients in different subbands lies in the fact that the image representation by steerable filter decomposition is overcomplete, and thus, the coefficients are correlated. The overlap of subbands can easily be seen in Figure 3.2, where the 3-scales, 4-orientations decomposition's Fourier spectra are given. Note that, apart from the 12 steerable subband filters, the decomposition also contains the low-pass and the high-pass subband.

More importantly, the cross-correlation statistics are computed on *magnitudes* of subband coefficients. The raw, complex coefficients may in fact be uncorrelated, since phase information can lead to cancelations. As shown by Simoncelli [**75**], the wavelet coefficients' magnitudes are *not* statistically independent and large magnitudes in subbands of natural images tend to occur at the same spatial locations in subbands at adjacent scales and orientations. The intuitive explanation may be that the "visual" features of natural images do give rise to large local neighborhood spatial correlations, as well as large scale and orientation correlations [**54**].

Dropping the spatial coordinates for the simplicity, the cross-correlation coefficient between subbands m and n is computed as:

$$\rho_{\mathbf{x}}^{m,n}(0,0) = \frac{\sum_{(i,j)\in W}\{(|\mathbf{x}^m(i,j)| - \mu_{|\mathbf{x}|}^m)(|\mathbf{x}^n(i,j)| - \mu_{|\mathbf{x}|}^n)\}}{\sigma_{|\mathbf{x}|}^m \sigma_{|\mathbf{x}|}^n} \tag{3.10}$$

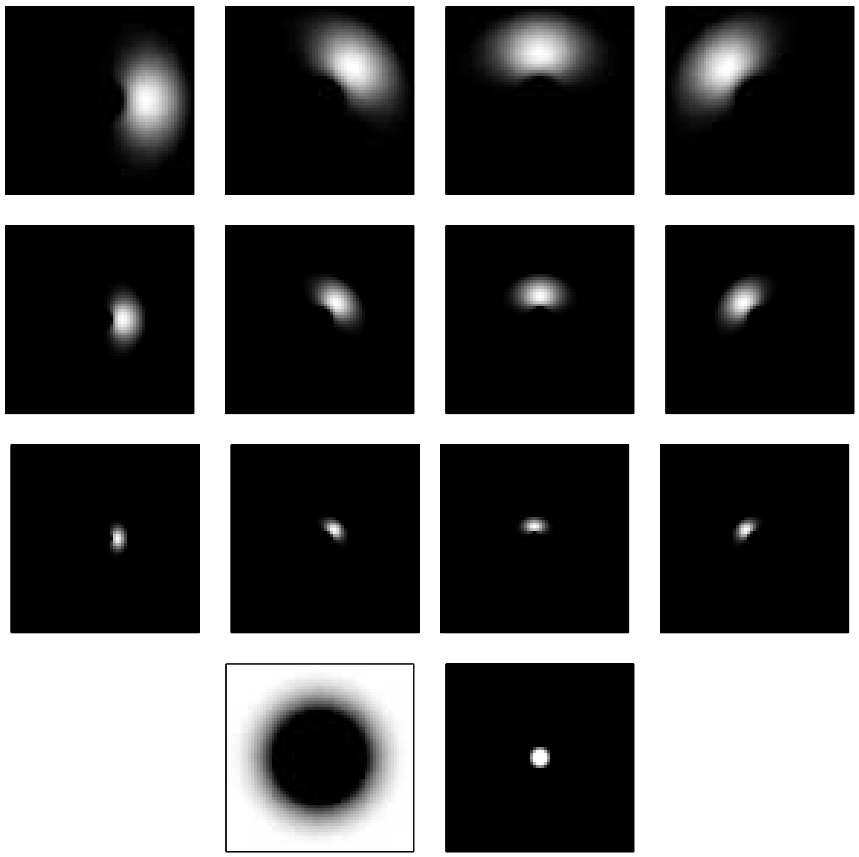


Figure 3.2. Frequency responses of Steerable Filters in 3 scales and 4 orientations; the axes ranges are $[-\pi, \pi]$ in both vertical and horizontal direction

We propose including the correlations between subbands at adjacent scales for a given orientation and between all orientations for a given scale. This is agreement with the findings of Hubel and Wiesel [**76**] that the spatially close simple cells in the primary visual cortex exhibit amplification of the responses of cells whose preferred orientations are similar. For the example in Figure 2.3 with $N_S = 3$ scales and $N_O = 4$ orientations, there would be, for each scale, $\binom{4}{2} = 6$ coefficients and for each orientation 2 correlation coefficients which gives total of $3 \cdot 6 + 4 \cdot 2 = 26$ new comparison terms.

After defining the subband statistics we wish to compare between two images, we should consider different ways of comparing them and combining the results into a single metric value. Also, we need to investigate (and justify) the use of the complex steerable pyramid transform, as opposed to using some other transform. The important questions that should be answered are:

- Should the statistics be computed on sliding window basis or over the global window that spans the whole image?
- Does the steerable pyramid have to be complex or could we use the real steerable pyramid transform?
- How do we compare the subband statistics?

3.3. Local Versus Global Processing

In SSIM, CWSSIM, and STSIM the processing is done on a small sliding window basis. This is a reasonable way to compare two images when the focus is on compression and image quality comparisons, where we want to ignore point-by-point differences, but want to make sure that local variations of the scale of the window size are accounted for by the metric. On the other hand, when the goal is overall similarity of two texture patches, either the textures are uniform (homogeneous), in which case a global window produces more robust statistics, or local variations are not as important, and a small window may overemphasize differences in the two textures we are comparing. An additional consideration is the scale of the texture. The window should be large enough to include several repetitions of the basic pattern of the texture (e.g., several peas) in order to be treated as a texture by the metric.

Since we are interested in both compression and texture retrieval applications, in this thesis, both local and global windows have been tested.

3.4. Complex Steerable Pyramid versus Real Steerable Pyramid

The complex steerable pyramid decomposes the real image $\mathbf{x}$ into complex subbands $\mathbf{x}^m$. The real and the complex part of $\mathbf{x}^m$ are not independent of each other, in fact, the imaginary part $\Im\{\mathbf{x}^m\}$ is the Hilbert transform of the real part $\Re\{\mathbf{x}^m\}$. This means that the real and imaginary parts of $\mathbf{x}^m$ are in quadrature. Quadrature filters are used for envelope detection and for local feature extraction in images. By applying filters in quadrature, we are able to capture the local phase information, which is consistent with receptive field properties of neurons in mammalian primary visual cortex [**77**].

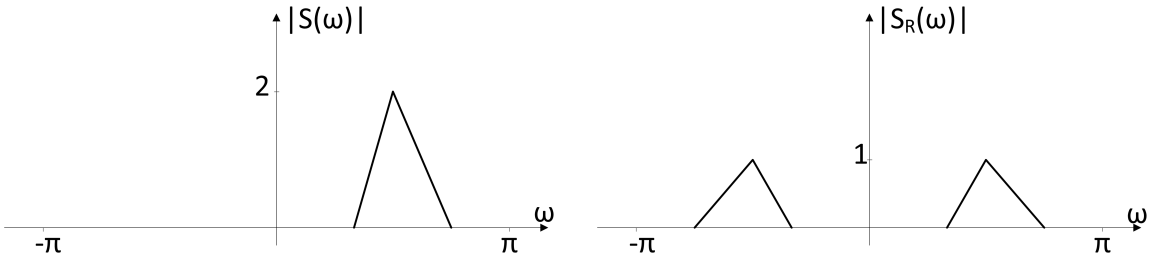
However, Aach *et al.* [**78**] have shown that the spectral energy signatures from the subbands obtained with quadrature filters are linearly related to the energies obtained by the "texture energy transform," which performs local variance estimation on the image filtered with the in-phase filter. This is true when we perform the calculations over the windows that are the same size as the filter support. In the case of the complex steerable pyramid which is applied in the frequency domain as opposed to the image domain, the filter size is the same as the support of the image. Thus, for the proposed metric, the same performance is expected when using either complex or real steerable pyramids when a global window is applied. For a local window, which is inherently smaller than the filter support, the conclusions from [**78**] no longer hold and the complex transform is favorable, given its invariance to small rotations, translations and scaling changes, as shown by Wang *et al.* [**53**].

Let us observe the effect of choosing the complex versus the real transform on the subband statistics we compare for texture similarity. The derivations will be given for 1-D signals, but the same analogy can be extended to 2-D.

If we denote by $s[n]$ the 1-D analytic signal (i.e., signal resulting from filtering with quadrature filters), then we can write:

$$s[n] = s_R[n] + j \cdot s_I[n].$$

$s_R[n]$ and $s_I[n]$ are the outputs of the quadrature filters which are related by the Hilbert transform. Using the Hilbert transform properties, we can draw the spectra of $s[n]$ and $s_R[n]$, $|S(\omega)|$ and $|S_R(\omega)|$, as given in Figure 3.3. The shape of the lobes is the same for both signals while the positive frequency lobe of the analytic signal has twice the magnitude of the lobe of the real signal.



(a) Spectrum of the analytic signal $s[n]$ (b) Spectrum of the real signal $s_R[n]$

Figure 3.3. Signal spectra for analytic signal $s[n]$ and real signal $s_R[n]$

3.4.1. Mean of subbands

Since $s[n]$ and $s_R[n]$ are band-pass signals, their mean values (computed on the global window) are zero. Thus, $\mu_s = \mu_{s_R} = 0$.

3.4.2. Variance of subbands

The variance of $s[n]$, computed over the global window, will be equal to the variance of $s_R[n]$, multiplied by a constant. We can prove this using Parseval's theorem, as follows. For the real signal $s_R[n]$, the variance is:

$$\sigma^2_{s_R} = \sum_{n=-\infty}^{\infty} s_R^2[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} |S_R(\omega)|^2 d\omega = \frac{1}{\pi}\int_{0}^{\pi} |S_R(\omega)|^2 d\omega \tag{3.11}$$

and for the complex signal $s[n]$:

$$\begin{aligned}\sigma_s^2 &= \sum_{n=-\infty}^{\infty} s[n]\cdot s^*[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} |S(\omega)|^2 d\omega \\ &= \frac{1}{2\pi}\int_{0}^{\pi} |2S_R(\omega)|^2 d\omega = 4\cdot\frac{1}{2\pi}\int_{0}^{\pi} |S_R(\omega)|^2 d\omega = 2\cdot\sigma^2_{s_R}.\end{aligned} \tag{3.12}$$

Thus, in terms of the variance subband statistics, the performance of the two possible transforms will be the same.

3.4.3. First-order autocorrelation of subbands

One of the main advocated reasons for using the complex transform versus the real one lies in the fact that, in case there is a small perturbation of the original signal such as small rotations, translations or scalings, the coefficients of the complex pyramid transform will have the same magnitudes while there will only be a constant shift in phase [**53**, **79**]. This will render the structure term 2.6 in CWSSIM equal to one, or very close to one. Since the structure term does not figure in STSIM or STSIM2, the effect of building the complex subbands as opposed to real has to be evaluated in terms of the first order autocorrelations.

Given that the steerable pyramid is shift-invariant [**10**] and that it is implemented in the Fourier domain, shifts and small rotations of the original image will be translated into phase shift in the Fourier domain and subsequently into shifts of subband coefficients. Thus, the autocorrelation terms will also be shifted in space and if we take their mean value over the whole subband, it will be unchanged with respect to the corresponding autocorrelation of the original image's subband.

Moreover, large shifts of image will also be tolerated – for uniform texture images, the shifts should not affect the perception of the structure present and the first-order correlation terms are forgiving to large translations. They are not forgiving to large rotations, which agrees with the goal for the metric to not be rotation invariant (horizontal stripes are considered to be different than diagonal or vertical ones).

3.4.4. Cross-correlation between subbands

In the case of the complex steerable pyramid, the cross-correlation terms between subbands are computed on the *magnitudes* of coefficients, since the "features" of real images give rise to large magnitudes in coefficients in neighboring orientations, scales and spatial locations, even if the raw coefficients are uncorrelated [**54**].

On the other hand, the cancelations that would occur when computing the cross-correlations of raw coefficients come from the changes in phase [**54**]. If we have a real transform, taking the magnitude of the coefficients will result in full-wave rectification of the signal and if the features across scales and orientations occur at the same spatial locations, their cross-correlations would not be zero. However, given the shape of the full-rectified signal, the correlations might be lower than in the case of the magnitude of

the complex coefficients. Thus, using the magnitudes of real coefficients as opposed to magnitudes of complex coefficients do change the cross-correlation values. However, if the correlation coefficients are consistently lower across all pairs of images, then we simply have a lower effect of the differences between $\rho_{\mathbf{x}}^{m,n}(0,0)$ and $\rho_{\mathbf{y}}^{m,n}(0,0)$, both in global and in local windows.

In conclusion, for the *global* window, there is no significant difference between using the complex or the real transform. However, when the *local* window is applied, this equivalence no longer holds. The spectral characteristics of signals are computed on the *whole* subband, over all of the coefficients, while locally, the spectral characteristics are different. In other words, if we took the subband coefficients within a small window, their real and imaginary parts are no longer related by the Hilbert transform and we will benefit from the properties of the complex steerable pyramid such as scale, rotation and shift invariance for small distortions.

3.5. Comparing the Subband Statistics

The final form of the proposed structural texture similarity metric, STSIM2, compares four subband statistics in each subband m ($\mu_{\mathbf{x}}^m$, $\sigma_{\mathbf{x}}^m$, $\rho_{\mathbf{x}}^m(0,1)$, $\rho_{\mathbf{x}}^m(1,0)$) and also the cross-correlations of subbands m and n, $\rho_{\mathbf{x}}^{k,m}(0,0)$, when m and n are adjacent bands of the same orientation or different orientations of the same scale. There are several questions here. First, how do we compare a given statistic across two images. For the luminance and contrast terms, all metrics so far (SSIM, CWSSIM, STSIM, STSIM2) use the ratio of the geometric and squared quadratic mean, while STSIM and STSIM2 use a different combination for the correlation coefficient comparisons. Then the various

terms are combined multiplicatively within a subband, and additively across subbands and spatial locations (of the sliding window). Zhao *et al.* [**73**] also considered a "multiplicative" approach, where the terms were combined multiplicatively within and across subbands and additively across spatial locations. The is another alternative, the Mahalanobis distance, which is a purely additive approach that we will examine in Section 3.5.2 below.

3.5.1. Comparing Statistics in STSIM2

The approach of combining the different subband statistics multiplicatively, originally suggested by Wang *et al.* [**16**], offers a few advantages.

First, this approach indirectly accounts for contrast masking, since the difference of terms compared (means, variances) are scaled by their values and thus they are implicitly weighted by how visible they are. Second, the consequence of the multiplicative combination of different terms in a subband, is that the weakest link dominates the similarity value. That is, if one term is very small, the product is very small, no matter how large the other terms are. Finally, the metric does not need any statistical information outside of the two images compared; i.e., the performance of the metric does not take into account the statistics of the other images in the database as the Mahalanobis approach we will examine below does.

Since STSIM2 extends the ideas of SSIM, CWSSIM, and STSIM, the comparison of the cross-correlation coefficients between different subbands, the values of which are in the interval $[-1, 1]$, should be done in the same way as the comparison of autocorrelations in STSIM:

$$c^{m,n}_{\mathbf{x},\mathbf{y}}(0,0) = 1 - 0.5|\rho^{m,n}_{\mathbf{x}}(0,0) - \rho^{m,n}_{\mathbf{y}}(0,0)|^p \tag{3.13}$$

Typically, $p = 1$.

For a steerable pyramid decomposition of image $\mathbf{x}$ into N_S scales and N_O orientations, we have a total of $N_B = 2 + N_O \cdot N_S$ subbands (lowpass, highpass and bandpass oriented and scaled subbands). The total number of cross-correlations between subbands is equal to:

$$N_C = N_S \cdot \binom{N_O}{2} + N_O \cdot (N_S - 1),$$

where the first term comes from the correlations across all possible orientation combinations for a given scale and the second term comes from the correlations of adjacent scales for a given orientation.

If for each subband we get one value of STSIM as a spatial mean, and for each pair of subbands we get one spatial mean of $c^{m,n}_{\mathbf{x},\mathbf{y}}(0,0)$, we can derive the STSIM2 metric as:

$$Q_{\text{STSIM2},\mathbf{x},\mathbf{y}} = \frac{1}{N_B + N_C} \left(\sum_{m=1}^{N_B} Q^m_{\text{STSIM},\mathbf{x},\mathbf{y}} + \sum_{i=1}^{N_C} c^{m_i,n_i}_{\mathbf{x},\mathbf{y}}(0,0) \right). \tag{3.14}$$

3.5.2. Comparing the Statistics with Mahalanobis Distance

Another approach to compare subband statistics is to form a large feature vector that contains all the statistics for each image and then to find the Mahalanobis distance [**80**] between the two feature vectors. Under the assumption that the different features are mutually uncorrelated, the Mahalanobis distance is in fact reduced to Weighted Euclidean distance, i.e., the MSE weighted by the variance of each of the terms. The variance is computed over all images in the database, so that differences between statistics that are not commonly occurring in the database are more heavily penalized than those whose

variance is large across all the images in the database. We will refer to this type of comparison as the STSIM2-M metric, where "M" stands for "Mahalanobis."

Note that, in principle, this metric can be computed on a sliding or global window basis, but we found that it is more effective for retrieval applications and global windows. As we pointed out, this metric assesses the similarity between two images not only in relation to each other, but also in relation to the other images present in the database.

The feature vectors can be constructed as follows. For each of the N_B subbands we compute four statistics on the global window:

- mean value $\mu_{\mathbf{x}}^m$,
- variance $(\sigma_{\mathbf{x}}^m)^2$,
- autocorrelation coefficient in horizontal direction $\rho_{\mathbf{x}}^m(0,1)$,
- autocorrelation coefficient in vertical direction $\rho_{\mathbf{x}}^m(1,0)$,

and for each of the N_C pairs of subbands we compute the cross-correlation term $\rho_{\mathbf{x}}^{m,n}(0,0)$.

The feature vector for image $\mathbf{x}$ has a total of $N_P = 4 \cdot (2 + N_O \cdot N_S) + N_S \cdot \binom{N_O}{2} + N_O \cdot (N_S - 1)$ points and can be written as:

$$\begin{aligned} F_{\mathbf{x}} &= [f_{1\mathbf{x}}, f_{2\mathbf{x}}, \ldots, f_{N_P\mathbf{x}}] \\ &= [\mu_{\mathbf{x}}^1, (\sigma_{\mathbf{x}}^1)^2, \rho_{\mathbf{x}}^1(0,1), \rho_{\mathbf{x}}^1(1,0), \ldots, \mu_{\mathbf{x}}^{N_B}, (\sigma_{\mathbf{x}}^{N_B})^2, \rho_{\mathbf{x}}^{N_B}(0,1), \rho_{\mathbf{x}}^{N_B}(1,0), \\ &\quad \rho_{\mathbf{x}}^{k_1,m_1}(0,0), \ldots, \rho_{\mathbf{x}}^{k_{N_C},m_{N_C}}(0,0)] \end{aligned}$$

To compute the distance between two images $\mathbf{x}$ and $\mathbf{y}$, we take the Mahalanobis distance between their two feature vectors $F_{\mathbf{x}}$ and $F_{\mathbf{y}}$, which is in our case the weighted Euclidean

distance between the two N_P-dimensional points representing $\mathbf{x}$ and $\mathbf{y}$, where the weight for a given term is the inverse of the variance of that term across the database.

If we denote by σ_{f_i} the standard deviation of the i^{th} feature across all the feature vectors in the database, the Mahalanobis distance is computed as:

$$Q_{\mathrm{STSIM2-M},\mathbf{x},\mathbf{y}} = \sqrt{\sum_{i=1}^{N_P} \frac{(f_{i\mathbf{x}} - f_{i\mathbf{y}})^2}{\sigma_{f_i}^2}}. \tag{3.15}$$

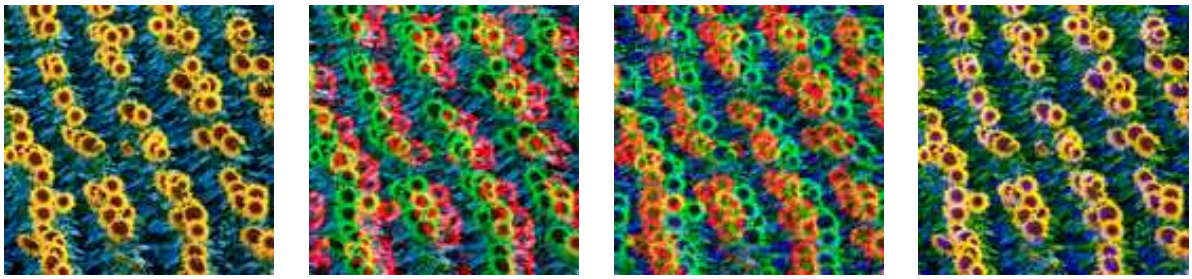
CHAPTER 4

Color Composition Similarity Metrics

A straightforward approach for extending an image quality metric to color is to apply the grayscale metric to each of the three color components in a trichromatic space; this is what is typically used in image compression applications. This approach is suitable for perceptually lossless and visually lossy algorithms; however, for structurally lossless compression and applications that are looking for *perceptual* similarity between images that preserves their structure, this approach is not suitable since humans process color differently than the grayscale structure of the images.

In particular, when we consider the similarity of texture images, as we saw in the previous chapter, the texture similarity metrics should not penalize small spatial shifts and rotations that do not affect the structure of the image. However, spatially shifting one channel of an RGB image relative to the other channels can alter or completely destroy the color composition of the image, while the texture of the shifted channel remains, in effect, the same. An example of this effect is illustrated in Figure 4.1, which shows the original image in (a) and images with spatial shifts in the R, G, and B channel in (b), (c), and (d), respectively. Each of the spatial shifts is by four pixels vertically and ten pixels horizontally. The effects of these shifts are clear in the distorted images, particularly in Figures 4.1(b) and 4.1(c).

In this thesis, we propose an alternative approach for color texture similarity that relies on separate estimates of the grayscale structure and the color composition of the image.



(a) Original (b) Shifted R channel (c) Shifted G channel (d) Shifted B channel

Figure 4.1. Inadequacy of channel-by-channel comparison of color texture images

This is more effective for image retrieval applications and structurally lossless coding. Here, we need to clarify what constitutes color composition and grayscale structure. *Color composition* is a listing of the colors of an image – as perceived by a human, not a histogram of the colors of the individual pixels – without regard to their spatial arrangement in the texture. *Structure* on the other hand, is the form of this spatial arrangement of colors, which includes both their grayscale intensity and their chrominance. *Grayscale structure* is thus only one part of structure; for example, we may have images with constant grayscale intensity but highly textured color. However, such cases are very unlikely in natural textures. Figure 4.2 shows samples of textures that have similar color composition but different grayscale structure, and textures that have similar grayscale structure but different color composition. By narrowing down the problem to separate estimates of similarity in terms of the color composition and the grayscale structure of two textures, we are making an important hypothesis, namely, that the structure does not affect the perception of the color composition, and conversely, that color composition does not affect the perception of structure. We are also making the assumption that structure is well-represented by grayscale structure. One of the goals of this thesis is to show that,

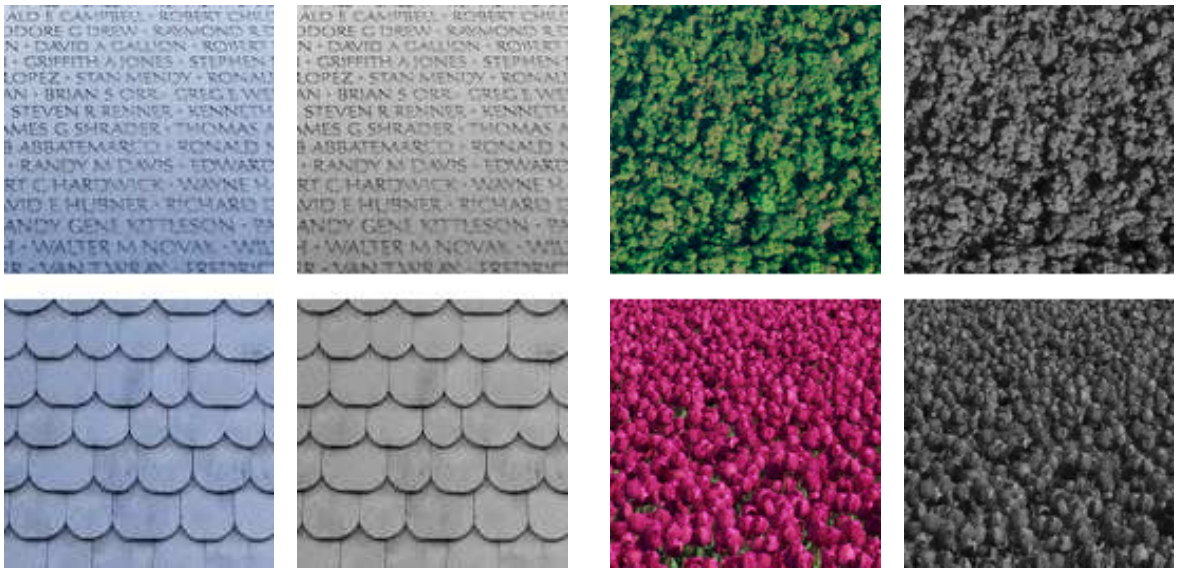


Figure 4.2. Textures with similar color composition and different structure (left) and with similar structure and different color composition (right).

even if this hypothesis is not true, this is a reasonable approximation in the context of texture retrieval and compression applications. In particular, in Chapter 7 we will show that texture retrieval based on the color composition similarly metric we develop in the remainder of this chapter is basically consistent with texture retrieval based on perceived color composition (determined on the basis of subjective experiments). Similar results hold for the evaluation of compression artifacts. While our main focus is on color composition of texture images, the metric applies to any color image.

In regard to a color composition metric, the conclusions that can be drawn from the background in Section 2.2 can be summarized as the following requirements:

- The image should be pre-processed to remove the noise.

- The color composition of an image should be represented by a compact feature vector that at the same time provides an accurate representation of the color composition.
- The color composition of two images should be compared in a way that agrees with human perception.

The proposed color composition metric will address each of these requirements.

4.1. Color Composition Feature Extraction

To address the first requirement, the first step of a color composition similarity metric would be image processing to eliminate the noise. However, simple smoothing of each of the RGB channels, as done in [**67**], may lead to "leaking" of colors across segment boundaries. To avoid such unwanted local color distortions, we propose a segmentation based approach that, in addition to noise removal, also addresses the second requirement, namely, the compactness and accuracy of the representation of the color composition. In the proposed approach, the images are first segmented into regions of similar color using the adaptive clustering algorithm (ACA) [**81**]. ACA is a generalization of K-means clustering that adds spatial constraints and adapts to local image characteristics. The K in K-means clustering stands for the number of segmentation classes. We use K_{ACA} to denote the number of classes obtained by ACA. The adaptation to spatial variations is particularly important for images of natural textures. Once we have the ACA segmentation, we can obtain a smoothed version of the image by replacing each pixel with the average of the pixel intensities in its neighborhood, but only those that correspond to pixels with the same segmentation label as the current pixel; that is, smoothing is

performed within each segmentation class (even across regions that are not connected, as long as they are in the neighborhood and belong to the same class) but not across boundaries between different region classes [**81**]. The resulting smoothed image consists of what Chen *et al.* [**14**] called ***spatially adaptive dominant colors.*** The color composition of the image is taken to be the color composition of this smoothed image of spatially adaptive dominant colors. Note that, since the (dominant) colors that correspond to each segmentation class are slowly varying, in the vicinity of each pixel, the histogram of this image is peaky and the number of dominant colors is essentially equal to K_{ACA}. However, over larger distances, since ACA is adaptive (smoothing is done over a relatively small area of the image), the same class can represent substantially different colors.

The use of spatially adaptive dominant colors to represent the color composition of an image is motivated by (a) human perception (humans perceive only a few dominant colors) and (b) the need for adaptation to local variations in the texture characteristics. The choice of K_{ACA}, along with the spatial constraint parameters of ACA, determines the amount of color detail that is preserved. A low value of K_{ACA} may miss some important colors, but a high value of K_{ACA} may preserve unimportant detail – that the human visual system typically ignores. We found that a relatively small value of K_{ACA}, e.g., $K_{\mathrm{ACA}} = 5$, can accomplish the goals we set at the beginning of this chapter: noise removal, compactness of the representation, and agreement with human perception. The requirement that the compact representation provide an accurate representation of the color composition is addressed by the adaptive nature of the ACA algorithm.

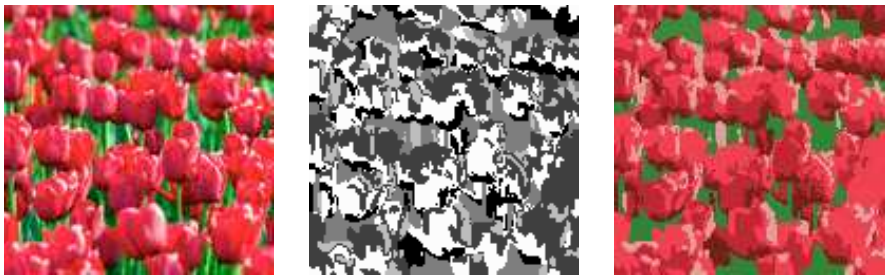
In their subjective experiments, Mojsilovic *et al.* [**82**] found that human subjects were not able to perceive nor distinguish more than six or seven colors in an image, even when

the patterns were very busy or multicolored. Thus, seven is an upper bound to K_{ACA}; in practice, we found that $K_{\mathrm{ACA}} = 4$ or $K_{\mathrm{ACA}} = 5$ are almost always adequate in terms of the compactness and fidelity of the representation of the dominant colors of the original image.

Figure 4.3 demonstrates the extraction of the dominant colors. An original image, shown in Fig. 4.3(a), is segmented by ACA to obtain a $K_{\mathrm{ACA}} = 5$ level segmentation, shown in Fig. 4.3(b); this is a segmentation map for segmentation into five different classes. Two kinds of dominant colors are shown in Figs. 4.3(c) and 4.3(d), global and local (spatially adaptive), respectively. The amount of smoothing can be varied by changing the size of the smoothing window, from global averaging in Fig. 4.3(c) to local averaging in Fig. 4.3(d). Note, however, that in all cases, averaging is done only within each segmentation class. This ensures that there is no "leaking" of colors or blurring across segments.

We are now ready to define the feature vector that characterizes the color composition of an image. In the case of global averages, it consists of the K_{ACA} dominant colors – one corresponding to each class, obtained by averaging the intensities of all the pixels that belong to this class – and the associated percentages in the image. In the case of spatially adaptive (varying) dominant colors, the feature vector consists of the histogram of color values in the smoothed image. The goal of the color composition similarity metric is then to compare these two feature vectors. As we will see, this can be done using the Earth Mover's Distance or the OCCD, both of which were discussed in Section 2.2.

However, as in the discussion of the grayscale STSIM2 in Section 3.3, an important question is whether the metric should be applied once to the entire image or it should



(a) Original (b) Segment labels (c) Global averages

(d) Local averages (e) Quantized local averages (f) Colors in Mojsilovic codebook

Figure 4.3. Color composition feature extraction: (a) Original images, (b) ACA segment labels; dominant colors obtained as (c) global averages, (d) local averages, (e) quantized local averages, and (f) colors in Mojsilovic codebook.

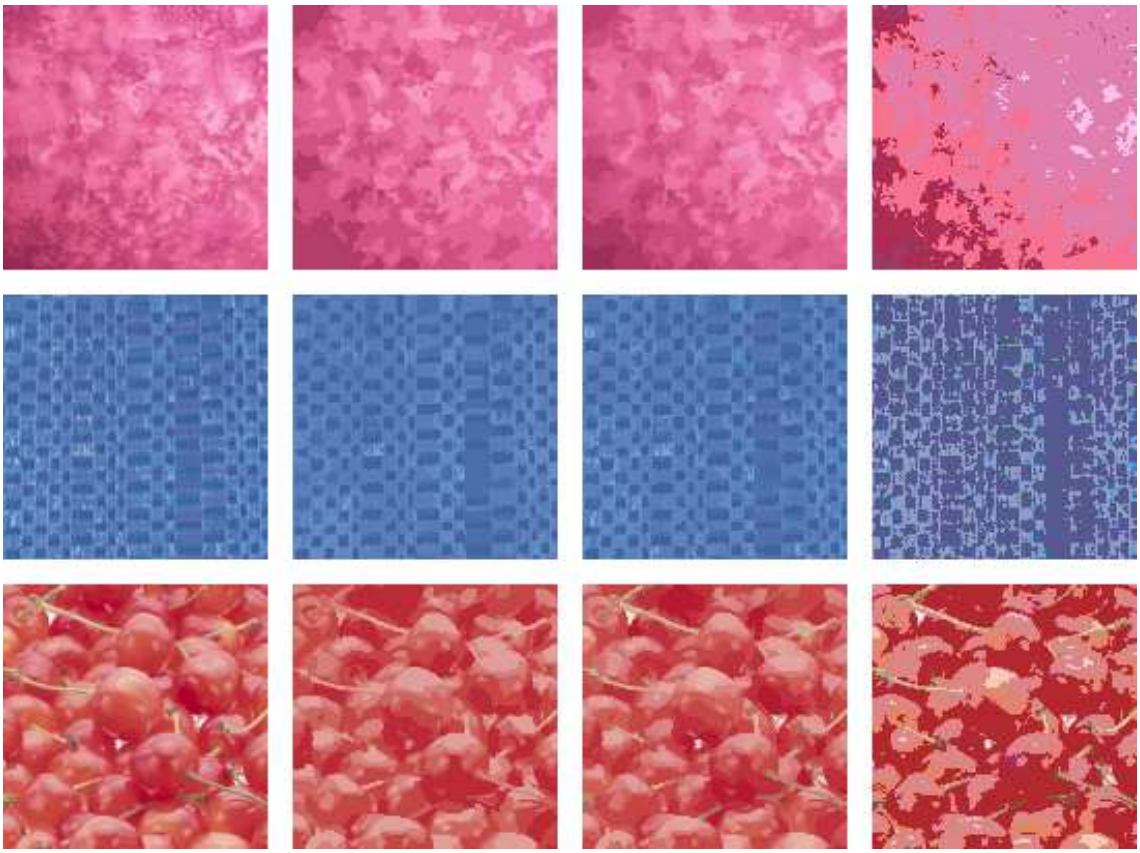
be applied to a sliding window, and then averaged over the image. Note, that we should distinguish between two different windows. One is the sliding *metric window* and the other is the *ACA smoothing window* that is used to obtain the dominant colors. As we discussed, in some applications, such as compression or image quality evaluation, where we want to account for local variations, we should use the sliding window approach to compute the metric, while in retrieval applications where we might be interested in the overall similarity of two textures patches, we should apply the metric to the entire image.

When the sliding window approach is used and the metric window is small, while there may be significant dominant color variations across the image, the dominant color

variations within the window are negligible (i.e., the histogram of the smoothed image is peaky) and, essentially, the number of dominant colors is equal to K_{ACA}. Thus, there is a good balance between compactness and accuracy of the feature vector.

When the metric is applied to the entire image, i.e., the metric window is the same size as the image, then due to the local variations of natural images, and textures in particular, the dominant color variations that humans perceive may be significant, and thus global averaging may result in dominant colors that are not representative of the image. In such cases, the spatially adaptive dominant colors should be used. If, on the other hand, the texture is homogeneous, the global averages can be used as the dominant colors.

When the spatially adaptive dominant colors are used, as we discussed above, the feature vector is the histogram of the smoothed image (the image shown in Fig. 4.3(d)). However, this feature vector can become very long, which can considerably increase the amount of computation for the comparison of two feature vectors. To shorten the feature vector without sacrificing the fidelity of the representation, we can perform classical K-means clustering on the local average colors, e.g., on the image shown in Fig. 4.3(d). The number of clusters K_{D} (where "D" stands for dominant colors) can be chosen so that the feature vector is not too long, yet it faithfully represents the (dominant) colors of the original image. A reasonable choice is $K_{\mathrm{D}} = 32$; the result is shown in Figure 4.3(e). Note that $K_{\mathrm{D}} = 32$ does not mean that there are 32 dominant colors. Most of the characteristic values will be very similar; the reason we pick a bigger K_{D} is to make sure we do not miss any important dominant colors. Having more colors that are close to each other does



(a) (b) (c) (d)

Figure 4.4. Color composition comparisons (a) Original texture (b) ACA-based global averages (c) ACA quantized local averages (d) Colors in Mojsilovic codebook.

not significantly affect the similarity computation. On the other hand, $K_{\mathrm{D}} = 32$ is much smaller than the size of the unquantized feature vector (128^2 for a 128×128 image).

An interesting point of comparison with the proposed color composition representation is the color quantization scheme (and corresponding dominant color representation) proposed by Mojsilovic *et al.* [**12**], shown in Figure 4.3(f). Three more examples are shown in Figure 4.4, which compares the original textures with the ACA global averages, the

ACA quantized local averages, and the colors based on the Mojsilovic codebook. Note that when there is a certain amount of local variation of texture characteristics, then global averaging result in colors that are not characteristic of the image, as is clear in the example in the bottom. Since the Mojsilovic codebook is not image-adaptive, one should expect, and this is clear for all three examples, potentially serious color mismatches, which again result in colors that are not characteristic of the image.

In summary, when global averages are used as dominant colors, the feature vector is the set of K_{ACA} colors and their respective percentages; when the locally adaptive dominant colors are used, the feature vector consists of the K_{D} quantized local average colors and their respective percentages.

Since the key idea is to extract perceptually dominant colors, it makes sense to set a lower threshold q for percentages below which a color is not considered to be "dominant," in which case it should be ignored. The assumption here is that such colors have a minimal effect on the perception of the color composition of an image. The percentages of such colors are assigned to the closest "dominant" color, without changing its value. In our experiments, this cutoff threshold was set to $q = 2\%$. Thus, the color composition feature vectors end up with *at most* K_{ACA} or K_{D} colors.

4.2. Color Composition Feature Comparison

To compare the feature vectors, we can use the Earth Mover's Distance (EMD) [**67**] or the Optimal Color Composition Distance (OCCD) [**12**], both of which were discussed in Chapter 2 and are well-established procedures for comparing color compositions in

accordance with human perception. We chose to use OCCD because of its straightforward implementation.

In the OCCD implementation, the colors in the feature vector are converted to CIELAB color space using D65 as the reference white [**83**], so that their Euclidean distances correspond to perceived dissimilarities [**62**, **63**]. The color percentages are quantized into n units of percentage p, so that $np = 100$. To keep things simple, the value of q (discussed on the previous section) should also be selected to be a multiple of p.

Finally, we should note that OCCD computes the *distance* between two color feature vectors, while the proposed grayscale texture similarity metric computes *similarities*. This has to be taken into consideration if the results of the grayscale and color composition similarity metrics are to be combined. However, we will see in the next chapter, the choice whether to combine and how should be left to the user and the target application.

4.3. Summary of Color Composition Metric

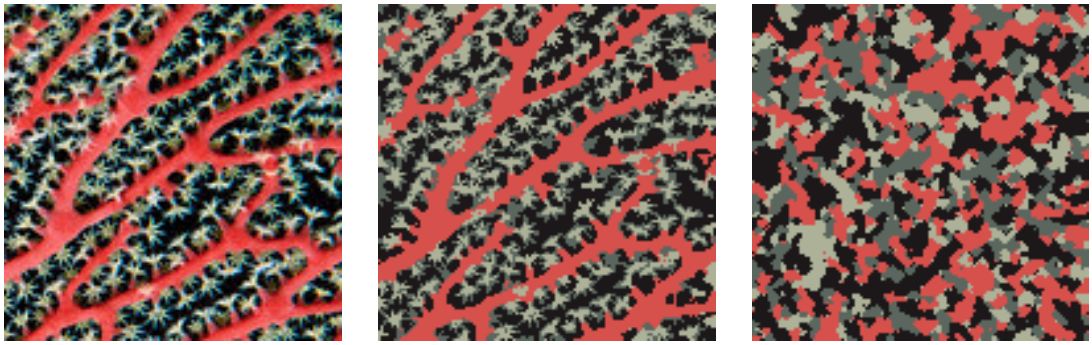
We now summarize the proposed color composition metric. Each image is represented by a feature vector that consists of the dominant colors and the associated percentages. When global averages are used as dominant colors, the feature vector consists of the set of K_{ACA} colors and their respective percentages; when locally adaptive dominant colors are used, the feature vector consists of the K_{D} quantized local average colors and their respective percentages. Colors with percent representation below a given threshold are not considered dominant and are thus ignored. The comparison between two feature vectors is then done using OCCD. To simplify the computation, the percentages are quantized.

4.4. Separating Color Composition and Grayscale Structure

In the development of the grayscale similarity metrics in Chapter 3 and the color composition similarity metrics of this chapter, we have decoupled the color composition and grayscale structure. As we discussed, whether this decoupling agrees with human perception of texture similarity can be checked indirectly in the context of texture retrieval and compression applications. If subjective texture similarity agrees with our similarity metrics, then the decoupling is justified. No matter what the outcome of such experiments, the question still remains: Does structure affect the perception of color composition, and conversely, does the color composition affect the perception of structure?

One approach to a better understanding of this question would be to conduct subjective experiments with original color textures, and textures with only structure but no color. This is easy to do, provided the structure is associated with the grayscale component, not the chrominance component of the texture. Conversely, we could conduct subjective experiments with the original color textures, and textures with the same color composition but the structure removed, or with some generic structure. This, however, is not a straightforward task.

One possible solution for removing the structure is shown in Figure 4.5. The original image (Fig. 4.5(a)) is first segmented using ACA and the dominant colors calculated using *global* averages (Fig. 4.5(b)), as described in Section 4.1. We then generate a sample Markov Random Field image with K_{ACA} levels based on the model described in [**81**], which assumes that the only nonzero potentials are those that correspond to the one- and two-point cliques, and where two neighboring pixels are more likely to belong to the same class than to different classes. The one point-clique potentials are used to control



(a) Original Image (b) Global averages (c) Markov Random Field image

Figure 4.5. Process of removing the structure in color textures

the percentage of labels in each class. An iterative procedure is necessary to obtain an MRF image with the same percentages of labels as the ACA segmentation. Finally, we "color" the MRF-image with the global averages of the ACA reconstruction (Fig. 4.5(b)), to obtain an image with the same color composition feature vector as the original image. The final result is shown in Figure 4.5(c).

Summarizing this chapter, we have developed color composition representations and associated similarity metrics that ignore the structure of the texture image. We also argued that the validity of the decoupling of color composition and texture structure can be checked, either indirectly in the context of texture retrieval applications, or with direct subjective tests where subjects are asked to evaluate texture similarity with and without one of the (color composition and structure) texture attributes removed. The second approach is beyond the scope of this thesis. The first will be considered in Chapters 6 and 7.

CHAPTER 5

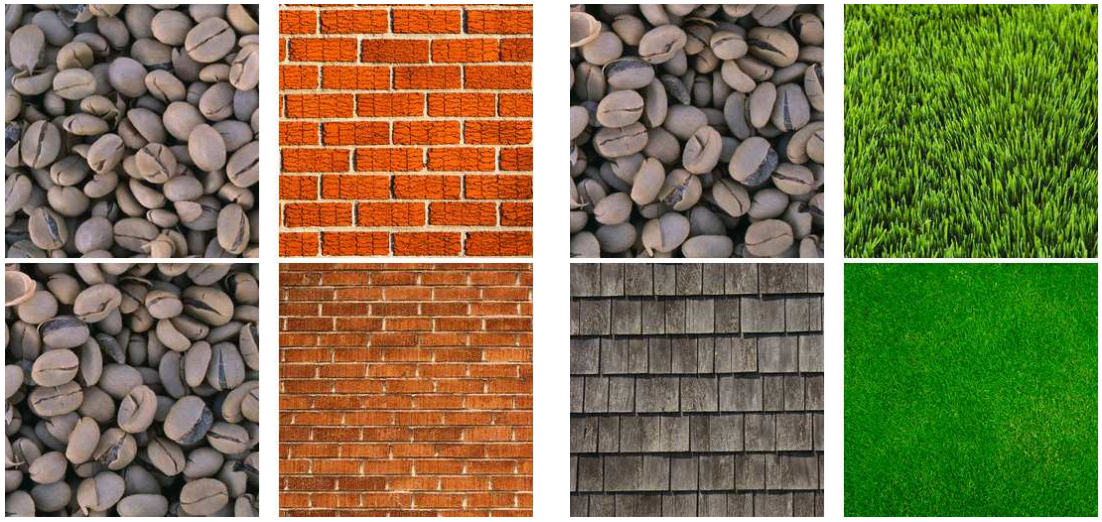
Initial Results and Basic Assumptions

In spite of a large body of research, the topic of texture analysis, and in particular, texture similarity, remains quite open. Further progress depends on a careful examination of the problem of texture similarity, the fundamental assumptions about the underlying signals, the capabilities of human perception, and the requirements of the intended applications.

Since most of the applications relate to human perception of textures, it is important that metric performance agrees with human judgments. Also, since humans are really good at recognizing textures, human perception provides a guide as to what can be achieved, an existence proof and performance bound. Existing algorithms are far from approaching human performance. On the other hand, for the most part, we cannot expect algorithms to do better than human perception.

In this chapter, we describe our initial experiments. We also conduct a careful analysis of their results, which leads to a better understanding of the texture similarity problem and sets the stage for the experimental design and methodology that will form the foundation for the remainder of this thesis, as well as future research in the area of texture similarity metric development and evaluation.

In our initial experiments, we carried out a subjective texture similarity test using 30 color texture images, organized in 50 pairs. Some examples of pairs can be seen in Figure 5.1. The subjects were asked to rate the similarity of each pair on a scale from 1 to 10,



(a) Similar colors, similar grayscale structure

(b) Similar colors, different grayscale structure

(c) Different colors, similar grayscale structure

(d) Different colors, different grayscale structure

Figure 5.1. Examples of texture pairs used in early experiments

with 10 being the highest score. The results were pooled together, and each pair was assigned the mean value of the subjective scores as its final similarity score.

For this initial experiment, we used the proposed structural texture similarity metric STSIM2 described in Chapter 3, computed with a sliding window of size 7×7. We denote the value of this metric, averaged across the sliding windows as Q_{STSIM2}. We also used the color similarity metric described in Chapter 4, also computed with a sliding window of size 7×7. Since the window is small, we used the histogram of the 49 local average colors as the color composition feature vector. The distance between the feature vectors was computed using the OCCD. To obtain a metric value that is comparable to the values of the grayscale metric, we scaled the OCCD results so that they lie in the interval $[0, 1]$, and then calculated the similarity of each window as $1-$*scaled distance*. The final value of the color composition metric, Q_{C}, was computed as a spatial average of similarities across the sliding windows.

In order to calculate a single similarity score for a pair of color textures, we used a linear combination of the two metrics, in accordance with some of the existing literature [**52**, **69**–**71**]. The final, composite similarity score is thus computed as:

$$Q_{\text{STSIM2+C}} = w_t \cdot Q_{\text{STSIM2}} + w_c \cdot Q_{\text{C}}. \tag{5.1}$$

For the given database of 30 textures, the best results were achieved with $w_t = 0.6$ and $w_c = 0.4$.

To test the performance of different objective metrics, the final mean subjective scores were correlated with the values that metrics assigned to each pair using the Spearman rank correlation coefficient. However, the correlation coefficients were too low for any metric to be considered useful; the proposed metric yielded the highest Spearman rank

Algorithm	PSNR	SSIM	CWSSIM	STSIM	STSIM2+C
Spearman ρ	0.28	0.51	0.56	0.60	0.66

Table 5.1. Spearman correlation coefficients, initial experiment

correlation coefficient, which was equal to 0.66. A comparison of the different metrics is given in Table 5.1.

Another performance parameter is the Pearson's correlation coefficient. These results are depicted in Figure 5.2. All the variables (subjective scores, metric values) were first "standardized" (i.e., converted into zero-mean, unit-variance variables), and then the different metric values were plotted versus the subjective scores. We then performed a minimum-mean-square error (MMSE) linear fitting of the data, and computed the direction coefficients to obtain Pearson's ρ. It is clear that the performance of all the metrics is again far away from the ideal ($\rho = 1$).

To find out the reasons for such a low correlation, in the remainder of this chapter, we reexamine the results of the subjective tests paying special attention to individual preferences. As we will see, the observations are quite revealing about the way humans judge the similarity of color textures, and represent a turning point in this thesis research, setting the stage for the experimental design and methodology on which the remainder of this thesis is based.

5.1. Decoupling Perceptual Dimensions of Texture Similarity: Grayscale and Color

Our analysis of the initial experiment, described in the previous section, reveals that the subjects were consistent at the high end of the similarity scale where images exhibit

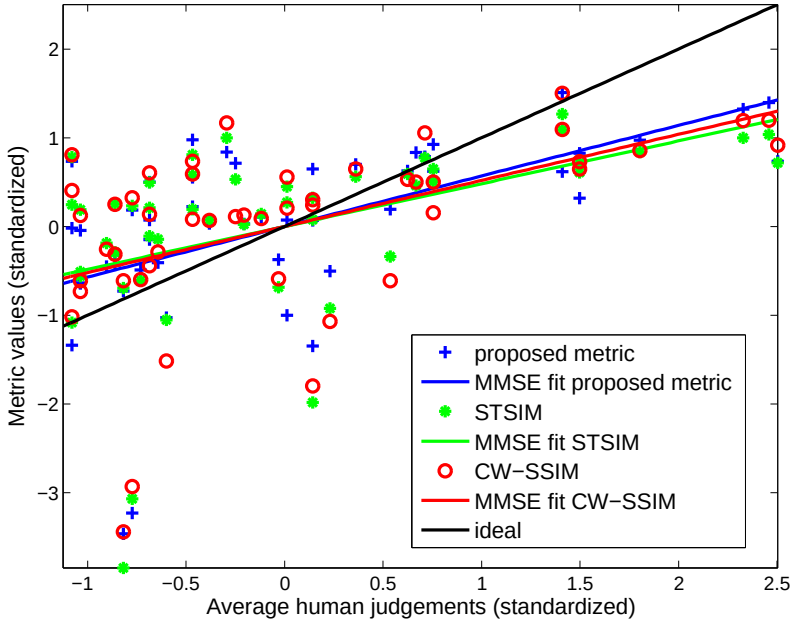


Figure 5.2. Scatterplot of metric values vs. average subjective scores, initial experiment

similarity both in grayscale structure and in color composition, as for example do the pairs shown in Figure 5.1(a). The agreement between the subjects was also high at the other end of the similarity scale, where images in a pair were *very dissimilar*, i.e., they both had very different color composition and grayscale structure, examples of which are shown in Figure 5.1(d). However, when the images were similar in some respect but different in another – for example, images had similar color composition but different grayscale structure like the pairs shown in Figure 5.1(b) or the other way around as shown in Figure 5.1(c) – the subject-to-subject agreement was poor because different subjects put different weights on these two texture attributes in determining overall texture similarity. Thus, some subjects gave a higher score to the left pair in Figure 5.1(b) than to the left pair in Figure 5.1(c), while other subjects rated them the opposite way. As we will see

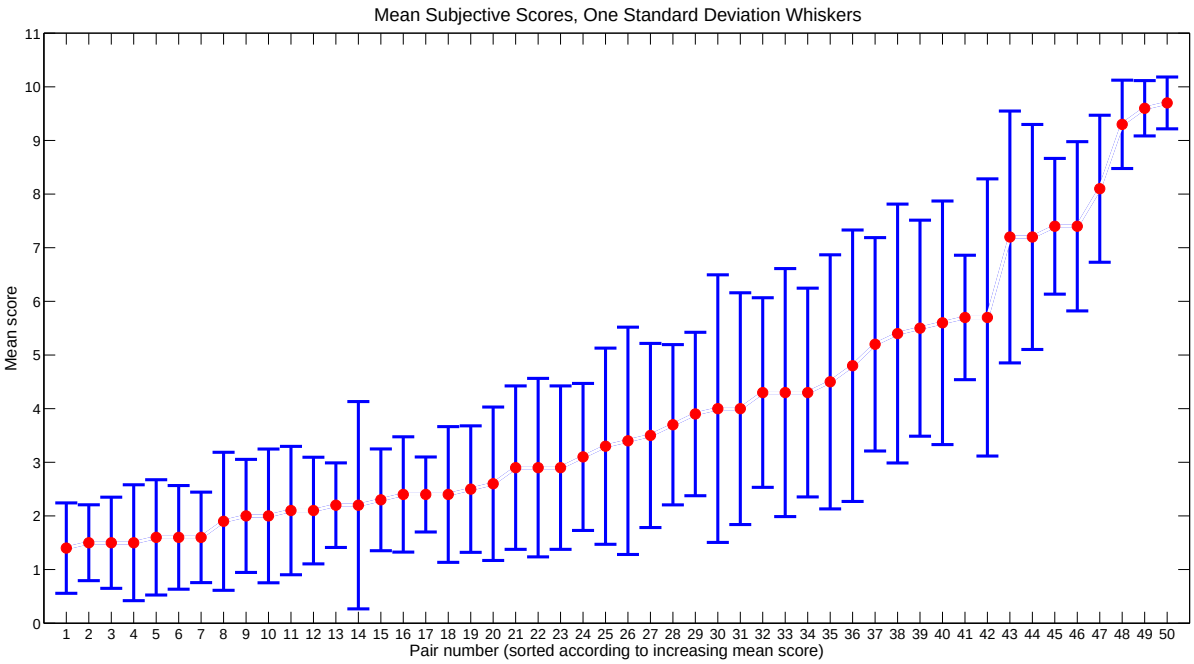


Figure 5.3. Mean subjective similarity scores with one standard deviation whiskers

below, similar observations hold for other texture attributes, such as regularity, scale, orientation, etc.

The disagreement in human judgments can be seen in Figure 5.3. Texture pairs have been ordered according to increasing average human score, which is represented by red dots. The whiskers around the red dots in the plot represent one standard deviation of the subjective scores for each pair. Where there is more agreement, the whiskers are shorter, which happens at the beginning and at the end of the plot, i.e., at the extremes of the similarity scale.

Similar behavior can be seen when the median values are plotted instead of the mean values (Figure 5.4, red lines), with boxes around them spanning the values of the subjective scores between the 25^{th} and the 75^{th} percentile for each pair. The black dashed whiskers

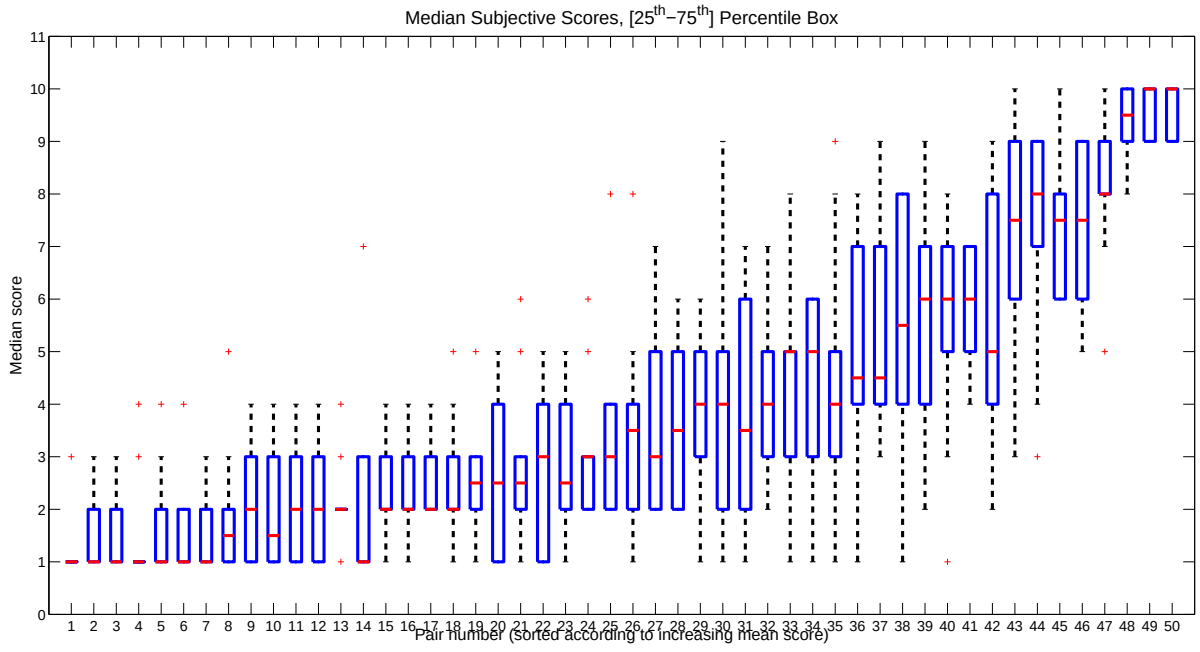


Figure 5.4. Median subjective similarity scores with the $[25^{th} - 75^{th}]$ percentile box

span all the scores given to a pair, except outliers, which are shown as red crosses in the plot. We can see that the boxes and whiskers are wider in the middle of the plot, where the outliers are also more common. For example, Pair 30 was given scores ranging from 1 to 9, which is a clear case of high disagreement between human subjects.

The *first conclusion* that can be drawn from this analysis is that different subjects weigh differently color and grayscale texture similarity, and that these two dimensions should be considered separately. This is in accordance with findings in the literature that different weights should be applied in different experimental settings [**69**, **70**]. Therefore, in the remainder of the thesis, we will decouple the study of grayscale and color texture similarity. The grayscale and color composition texture similarity metrics we developed in Chapters 3 and 4, respectively, will be used as the objective criteria to be compared

with subjective evaluations. Whether it makes sense to combine the results and how, will be determined by the end-user, given the target application, database content, and performance requirements.

Another example that illustrates the difficulty of quantifying texture dissimilarities came up during our initial tests in the domain of grayscale textures (i.e., the grayscale component of color texture images). When we asked human subjects whether an image of vertical stripes is more similar to an image of peas than an image of a brick wall to an image of a tulip flowerbed, the answer again varied from subject to subject. Thus, in principle, image attributes associated with different perceptual dimensions (scale, directionality, regularity, granularity, etc.) should be considered separately, and different metrics should be used to quantify similarity along each of these dimensions. However, this is beyond the scope of this thesis. As we will see in the next section, the most important conclusion is that quantifying texture similarity makes sense only when textures are similar along all the perceptual dimensions. Separating the grayscale and color attributes is convenient for testing the corresponding similarity metrics, and also, eliminating one dimension – color – facilitates the construction of our database, in the sense that it is easier to find textures that are similar along the remaining dimensions. Here, we should point out that eliminating color is easy; we simply obtain the luminance component, in the appropriate color space. In Chapter 4, we outlined a technique for eliminating structure while retaining color composition. Eliminating other dimensions (regularity, orientation, scale, or overall structure) is much more difficult, if at all possible.

5.2. Quantifying Similarity in Different Regions of Similarity Scale

The *second,* and perhaps most important, *conclusion* of our preliminary experiments is that human subjects make consistent judgments about texture similarity only when the textures are similar in every perceptual dimension, i.e., when the textures are *very* similar. This should not be surprising, as even in the case of color, which is considerably simpler than texture, quantifying similarity only works for colors that are close to each other. It does not make much sense to ask whether blue and orange are more different than green and red, and if we did, we would not get consistent answers. Thus, attempts to derive perceptually uniform color spaces, such as CIELAB and CIELUV, have resulted in consistent results only over small distances.

We can further extend the color analogy to the texture similarity problem. In Chapter 6, we will propose a procedure for forming clusters of similar images, so that we can test whether the predictions of our texture similarity metric are consistent with these clusters. The similarity clusters are analogous to MacAdam's ellipses in the space of colors, depicted in Figure 5.5 (solid lines), where each ellipse encompases the colors that are perceptually indistinguishable from the color at the center of that ellipse. In the case of textures, we wish to find the N-dimensional ellipses that contain all the textures that are considered to be very similar by the human observers, a relaxed condition compared to MacAdam's indistinguishable colors. Thus, our similarity clusters correspond to enlarged ellipses in the MacAdam analogy, as indicated by the dashed line in Figure 5.5

This observation has major implications for the design of the subjective tests that will facilitate the development and testing of the texture similarity metrics. If the testing procedure asks subjects to give numerical values for the similarity of images that are not

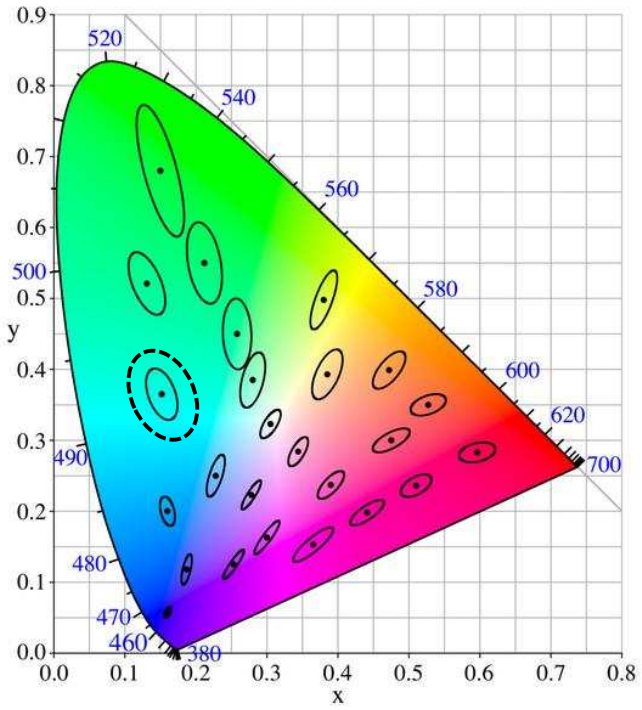


Figure 5.5. MacAdam's ellipses in CIE 1931 xy chromaticity diagram

similar in all perceptual dimensions, as was the case in our preliminary experiments, then the answers are inconsistent and of little value as a benchmark for the design of texture similarity metrics. Thus, quantifying texture similarity should be limited to the high end of the similarity scale. Moreover, as we will see in the next section, quantifying similarity is necessary only in certain applications, such as image compression. On the other hand, for most other applications, such as content-based retrieval, the most important question may be to determine whether two textures are similar or dissimilar. This considerably simplifies the testing procedure, and dramatically increases the chances of obtaining consistent subjective results. As we will see, even the subjective labeling of each pair of textures in a large database as similar or dissimilar would require an enormous number of tests. In the

subjective tests described in Chapter 6, we develop an efficient procedure for clustering textures in a large database into similarity groups, so that textures within a cluster should be considered similar, and textures across clusters dissimilar.

5.3. Summary of Conclusions from Analysis of Initial Results

In this section, we summarize the conclusions and associated corollaries from the discussions of the previous two sections.

Conclusion 1. Different subjects weigh differently color and grayscale texture similarity.

Corollary 1.1. Color and grayscale texture similarity problems should be considered separately. Different objective similarity metrics should be used for each of these dimensions, and separate subjective experiments should be designed to test their effectiveness in predicting human preferences.

Conclusion 2. Human subjects make consistent judgments about texture similarity only when the textures are similar in every perceptual dimension.

Corollary 2.1. Quantifying texture similarity should be limited only to the high end of the similarity scale, i.e., where the textures are similar along all perceptual dimensions.

Corollary 2.2. Determining whether two textures are similar or dissimilar is a more important question than quantifying their similarity for a majority of applications, such as CBIR.

Conclusion 3. Quantifying texture similarity is necessary only for certain applications, such as image compression.

5.4. Metric Performance in Different Applications

As we already hinted in the previous section, the texture similarity problem should be considered in the context of different applications, each of which imposes different requirements on metric performance.

For applications such as image compression it is necessary to have the correct ordering of images according to perceived similarity. This holds true for both applying the metric to image quality assessment, and for using it as a tool within a compression algorithm. Thus, a metric must provide a monotonic relationship between measured and perceived distortion, up to a point where the distorted images are no longer of acceptable quality. In addition, it is important to have an absolute measure of image similarity, so that consistent image quality can be achieved across different types of image content, both within an image and across different images.

On the other hand, in content-based retrieval applications, most of the time, it is enough to distinguish between similar and dissimilar images, while the precise ordering of the results may be useful but of lesser importance. An absolute similarity scale is not necessary as the task is to retrieve a certain number of images from a database that are most similar – preferably in order of similarity – to the query image.

Figure 5.6, which plots subjective rankings versus objective metric values, illustrates the requirements on an (ideal) metric performance: a monotonic relationship in the region of very similar textures, i.e., the compression area; a clear gap between the metric values

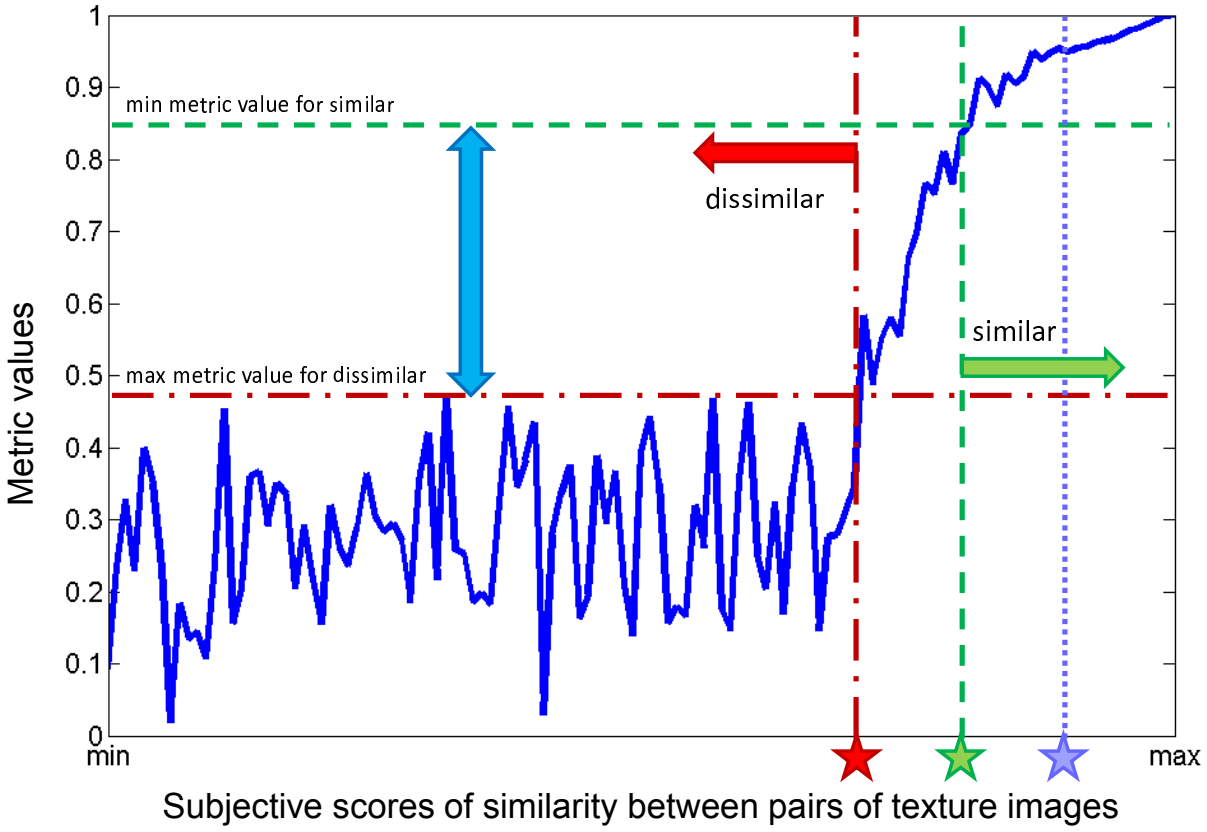


Figure 5.6. Ideally looking plot of metric values vs. subjective similarity scores

assigned to *similar* and *dissimilar* pairs; and relaxed constraints on the metric behavior in the region of *dissimilar* textures. The subjective similarity scores on the x-axis are for all possible pairs of textures in a database. Thus, for a database of N images, the x-axis would contain a total of $\binom{N}{2}$ points. The subjective scores are ordered from lowest to highest. For the sake of argument, we assume here that it is possible to derive consistent subjective similarity scores for all texture pairs. As we discussed, in reality this would be very difficult, if not impossible.

Several points of interest should be noted in this plot, which separate the subjective similarity range into important regions:

- The region of *dissimilar* pairs is to the left of the red star. In this region, the subjects agree that the textures are "significantly different," even though they may not assign consistent similarity values.
- The region of *similar* pairs is to the right of the green star. Again, the subjects agree that the textures are similar but may not assign consistent similarity values.
- The region between the red and green stars is the "non-consensus area," where the subjects cannot agree whether the texture pairs are similar or dissimilar.
- Finally, the region to the right of the violet star is where monotonic behavior of the metric is desirable, primarily for compression applications. In this region the textures are essentially "the same," even though significant differences (distortions) are still visible.

The (ideal) range of metric values is also established by the following horizontal partitions

- The horizontal red dash-dotted line represents the highest metric value for dissimilar pairs.
- The horizontal green dashed line represents the lowest metric value for similar pairs.

The blue arrow represents the gap between the scores of the metric for "similar" pairs and "dissimilar" pairs.

The plot emphasizes the fact that quantifying similarity in the range of dissimilar textures is not important, and perhaps unachievable. This removes unnecessary constraints on metric performance, and makes the problem more tractable. Assuming that there

is a transition region where subjects disagree on the question of similar versus dissimilar textures, the gap in metric values between similar and dissimilar textures becomes an important new measure of metric performance. However, in reality, this gap is only conceptual (statistical), as there will always be some noise in the data. The metric performance measures we discuss in Chapter 6 are indirectly but definitely related to the "size" of this gap.

Note that even in the range of similar textures, for many applications, it may not be important to provide a monotonic relationship between subjective and objective values. Moreover, while subjects agree that the textures are similar, they may not be able to assign consistent similarity values, in which case, there is no point in trying to derive metrics that agree with subjective evaluations (ordering). Thus, the relationship between subjective and objective similarity needs to be monotonic only at the high end of the similarity range. This is only a small, very restricted subset of texture pairs. Again, by focusing on the appropriate subset, this eliminates unnecessary experiments and makes the problem more tractable. In Chapter 6, we will consider systematic ways for generating texture pairs to populate this similarity range, in order to facilitate image compression applications.

By understanding the requirements of different applications and identifying constraints on metric performance in the appropriate regions of the similarity scale, this thesis research has enabled real progress toward texture similarity metric development and validation. In the remainder of the thesis, we will design and carry out experiments by focusing in different application areas:

- Content-based retrieval of similar textures: The key here is separating similar from dissimilar textures.
- Image compression: The key here is quantifying similarity (in the presence of visible image degradations), in relative or absolute scale.
- Known-item search: Here the focus is on content-based retrieval of "identical" textures.

The last problem is of particular importance, because unlike the other two, it does not require extensive subjective tests. By "identical" textures we mean they are perceptually the same, even though they may be significantly different in a point-by-point sense. The ground truth for such an experiment comes from the database construction algorithm. We extract smaller patches from a big, uniform texture image, and label the textures that come from the same original as "identical" textures. Given the relative ease of database construction and labeling, it is not surprising that this problem has received the most attention in the literature.

The detailed description of experiments is given in Chapter 6 while the results and performance analysis are given in detail in Chapter 7.

CHAPTER 6

Experimental Setup

In this chapter we discuss the construction of the texture database and the basic experimental setup, organized according to application. We also introduce Visual Similarity by Progressive Grouping (ViSiProG), a method for subjectively grouping texture images into visually similar groups to provide ground truth for testing the proposed texture similarty metrics.

As we discussed in Chapter 5, the first step in the experiment design is the identification of the target applications, each of which imposes its own requirements for metric performance and testing procedures. We have identified three different applications:

- Known-item-search (CBIR)
- Retrieval of similar textures (CBIR)
- Quantification of texture distortions (Compression and Quality)

In accordance with the conclusions of Chapter 5, our subjective and objective tests will be conducted separately for grayscale and color textures. However, the same experiments can be carried out for both grayscale and color images; they will simply be performed independently in each database.

We first look at the construction of the texture image database. We then discuss each of the applications separately.

6.1. Database Construction

In order to perform the extensive tests in the context of the above-cited applications, we collected a large number of color texture images. The images were carefully selected to meet some fundamental assumptions about texture signals as well as the needs of the target applications.

The precise definition of texture is not widely agreed on in the literature. However, several authors (e.g., Portilla and Simoncelli [**84**]) define texture as *an image that is spatially homogeneous and that typically contains repeated structures, often with some random variation (e.g., random positions, size, orientations or colors).* The textures we collected had to meet the requirement of spatial homogeneity and repetitiveness; the latter we defined as at least five repetitions of a basic structuring element. We also made sure that there is a wide variety of textures and a good balance of similar and dissimilar textures. Texture similarity, of course, is subjective and the goal of our experiments, so our selection process was not very precise. Another requirement was that the database contains examples of "identical" textures (for known-item search applications), which we discuss below, as well as image compression distortions, which we discuss later in this chapter.

To construct the database, we downloaded around 1000 color images from the *Corbis* website [**85**]. All of the textures were photographic, mostly of natural or man-made objects and scenes. No synthetic textures were included. Roughly 300 of those were discarded, as they did not represent perceptually uniform textures. The resolution varied from 170×128 to 640×640 pixels.

To obtain groups of "identical" textures, each of the remaining approximately 700 images were cut into two or three smaller images of size 128×128 pixels. Depending on the size of the original image, the extracted images had different degrees of spatial overlap, but we made sure that there were substantial point-by-point differences, such as those shown in Figures 6.1 and 6.2. The idea was to minimize overlap while maintaining texture homogeneity. In some cases, when the original image was large, we downsampled the image – typically by a factor of two – in order to meet the repetitiveness requirement. The group of smaller images originating from the same larger texture are considered to be "identical" textures. Several texture images from our database, one from each identical group, are shown in Figure 6.3.

Finally, for the compression application, a selected subset of the images from the database were distorted according to the procedures described later in this chapter in Section 6.5.

6.2. Known-Item Search

The first application focuses on the right-most part of the curve in Figure 5.6, where the goal is to retrieve textures that are "identical" to the query texture, in the sense that they are pieces of the same texture, which is in accordance with the database construction we discussed above. This type of problem is well-known and studied in the text retrieval community, who refer to is as the *known-item search* [**86**]. It has also been addressed by the image processing community for texture retrieval applications [**17**,**18**,**87**]. This is not surprising, as the problem is well-defined and relatively easy to generate data for.

(a) Original peppers image

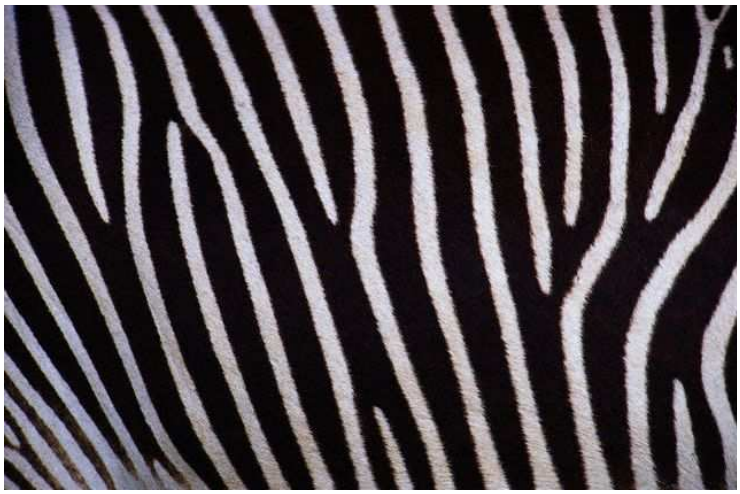
(b) Cut-out 1

(c) Cut-out 2

(d) Cut-out 3

Figure 6.1. Example 1 of populating the image database

The known-item search is an important problem in CBIR, where one may want to find images that contain a particular texture. This type of match should result in the highest metric values, short of the case of "perceptually identical" textures we discussed in Chapter 2, and of course, pixel-by-pixel identical textures. In this sense, it is also relevant to image compression applications. Ignoring boundary continuity issues, this is the highest level of similarity (quality) one can expect in "structurally lossless" compression [**5**]. (See also Chapter 1.)



(a) Original zebra image

(b) Cut-out 1

(c) Cut-out 2

(d) Cut-out 3

Figure 6.2. Example 2 of populating the image database

One advantage of this experiment is that it can be carried out on a large database of images effectively *without* any subjective experiments, which become prohibitively lengthy as the number of images in the database grows. The ground truth follows from the database construction. All that is needed is that the larger textures from which the test samples are obtained are homogeneous (perceptually uniform). Thus, the subjective evaluations are implicit in the database construction.

A number of evaluation measures have been developed to assess the performance of know-item search systems. Common measures for this type of retrieval systems include

Figure 6.3. Examples of textures in the database

precision at one (measures in how many cases the first retrieved document is relevant), mean reciprocal rank (measures how far away from the first retrieved document is the first relevant one), and mean average precision. More details will be provided in Chapter 7.

6.3. Retrieval of Similar Textures

We now turn our attention to the problem of distinguishing between similar and dissimilar textures. As we saw in Chapter 5, this is important in CBIR applications where one may be interested in retrieving textures similar to a given query without the need for an absolute measure of similarity, or even for one that is monotonically related to subjective evaluations.

To obtain the ground truth for this experiment, we need to carry out subjective tests that classify pairs of textures as similar or dissimilar, with possibly some pairs of textures being not clearly similar or dissimilar, as illustrated in Figure 5.6. Since the retrieval of identical textures is addressed separately, for the ground truth experiments, we only need to use a subset of the database that includes one sample from each group of "identical" textures.

The process of separating similar texture pairs from dissimilar ones can be done in a number of different ways. The most straightforward approach is to perform a single-stimulus binary forced choice test, where the subjects are asked to rate each pair in the database as "similar" or "dissimilar." However, the number of comparisons grows quadratically with the number of textures in the database.

A more laborious approach is the one we followed in our initial experiment, described in Chapter 5, where we ask subjects to assign a numerical value (e.g., in the range 1 to 10) to the similarity of each texture pair. Alternatively, we could ask subjects to compare two texture pairs at a time and choose the most similar (two-alternative forced choice – 2AFC). In principle, these two experiments can lead to equivalent results [**88**]. In both cases, the subjective scores can then be pooled together, forming a preference matrix, which can be analyzed using one of the popular methods, such as multidimensional scaling [**89**, **90**] or the law of comparative judgment [**91**]. However, when the dataset is large, conducting such subjective tests becomes infeasible, since the number of pairs that need to be judged grows quadratically with the number of images. The number of trials grows as the power of four for the pairs of pairs in the 2AFC. Moreover, these tests are conducted in order to extract more information than just to classify into similar and dissimilar pairs.

Another approach would be to conduct the test as multiple rank order judgments [**92**], where the subjects are asked to rank a small subset of images based on the similarity to the query. The database is partitioned in such a way that each subject makes a judgment on the similarity between all possible pairs of images. Using this method the subject gets to compare all the pairs in the database with a reduced number of trials, since they are simultaneously comparing a few images, as opposed to comparing them one pair at a time. This method has been applied in image similarity experiments and can reduce the needed number of comparisons by four [**93**]. However, it still requires lengthy tests and many subjects when the database under consideration is large. And again, this test can extract more information than just classification into similar and dissimilar pairs.

For our purposes, since we are primarily interested in discriminating between similar and dissimilar texture pairs and do not aim at extracting any other information about the image pairs, a more efficient test procedure that can provide the same data, is a *texture clustering experiment,* whereby the subjects are asked to form *similarity clusters*, i.e., groups of images that are similar to each other and dissimilar from images in other groups. However, while such a test is well-defined and easy to carry out for a relatively small set of images, when the number of images is in the order of several hundred or more, practical problems arise, as it is difficult for subjects to see images in multiple clusters simultaneously, in order to decide how to allocate them to clusters. This is particularly difficult when the experiment is conducted electronically, and the database is too large to be presented in its entirety on a single computer screen.

To alleviate this problem, in the next section, we propose a progressive testing scheme. The key idea is that, when the database is relatively large, i.e., too large to be presented

on a single computer screen, it will be easier for the subjects to form the similarity groups one at a time, in a step-by-step fashion, picking similar images out of a small set of images, and repeating the process with a new set that contains the group and a new set of images, progressively refining the similarity group.

6.4. Visual Similarity by Progressive Grouping (ViSiProG)

The goal of the Visual Similarity by Progressive Grouping (ViSiProG) procedure is to cluster textures into groups of similar textures. ViSiProG does this *one* group at a time in a progressive fashion. By pooling together the results of multiple groups of multiple subjects, we can then form a similarity matrix for the entire database, which can be analyzed by methods such as multidimensional scaling [**89**, **90**] or spectral clustering [**94**] to form the final similarity clusters. The similarity matrix is a square matrix in which each row and column represents one texture image in the database, with each entry representing the number of times the images were placed by a subject in the same group.

Note that the term "group" refers to the outcome of one trial, for one subject, while the term "cluster" refers to the similarity group that results from analyzing all the groups from all the subjects.

In this experiment, the subjects are sequentially presented with subsets of the database, and build the similarity group in a step-by-step fashion. Initially, they are presented with a set of images, and after choosing the first group of "most similar" images, they get a new batch of images they can use to refine their group. The subjects can keep refining the group for as long as they like, and only after they have seen all the images in the database, they are allowed to save their results. Note that the subjects do not *have* to

end the test immediately after all the images have been presented, but they cannot end the test *before* they have seen each and every image from the database. Thus, this feature ensures that a subject has considered all the images before making the final decision on a similarity group.

Given that in the database there is usually more than one similarity cluster, the choice of the initial subset of images shown to the subject will greatly influence the subsequently chosen group. Therefore, for each trial, the initial subset presented to the subject is randomly chosen from the database, so that the chance of the subject forming a different group in each trial is increased.

6.4.1. Detailed ViSiProG Testing Procedure

Let us denote the total number of images in the database as N, the number of images shown to subjects in a batch as N_b, and the number of images that are chosen to be in the group as N_g. This number N_g stays fixed throughout the procedure, i.e., in each iteration subjects have to pick N_g images out of N_b presented. Typical values are $N = 500$, $N = 36$, and $N_g = 9$.

The test begins by randomly selecting a batch of N_b images from the N images in the database, as illustrated in Figure 6.4. Initially, all the images have the same probability of being selected. The chosen images are presented on the screen in a box, organized in a regular grid. As illustrated in Figure 6.5, the subject then selects N_g images to form a group in a separate box, shown in green and labelled as "GROUP" in the figure, using the mouse to drag the images into the box. This enables the subject to visualize the similarity of the images in the group, as they are visually separated from the remainder

Figure 6.4. Snapshot 1 ViSiProG interface (grayscale similarity test): This is the first batch before the subject has selected any images. Each time a new batch is requests, and the images are shuffled, and before the group stabilizes, the screen looks like this.

of the batch. The subject now has the option to replace individual images in the chosen group of N_g images with other images in the batch, until she/he is satisfied that the group represents the N_g most similar images in the batch. Figure 6.6 shows the snapshot that immediately follows the one in Figure 6.5. Note that the texture in the lower right corner of the green GROUP box has been replaced with one that visually blends better with the other images in the box. The subject can then press a button to request a new set of N_b images, out of which N_g are the ones previously selected to form a group, and $N_b - N_g$ images are randomly selected from the remaining $N - N_g$ images in the database. The new set of images is shuffled before it is presented to the subject. The process is repeated

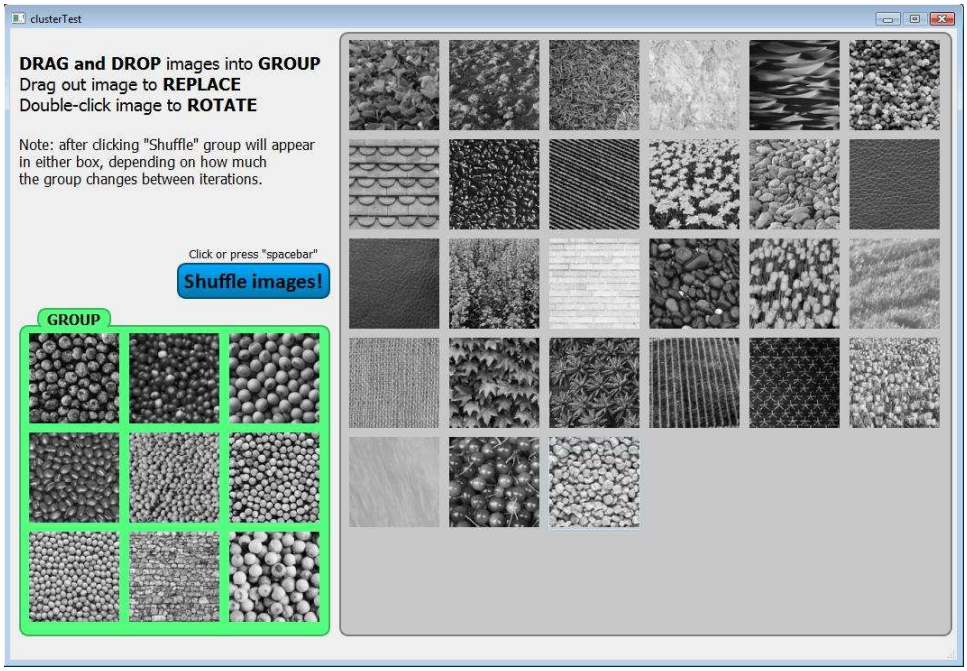


Figure 6.5. Snapshot 2 of ViSiProG interface (grayscale similarity test)

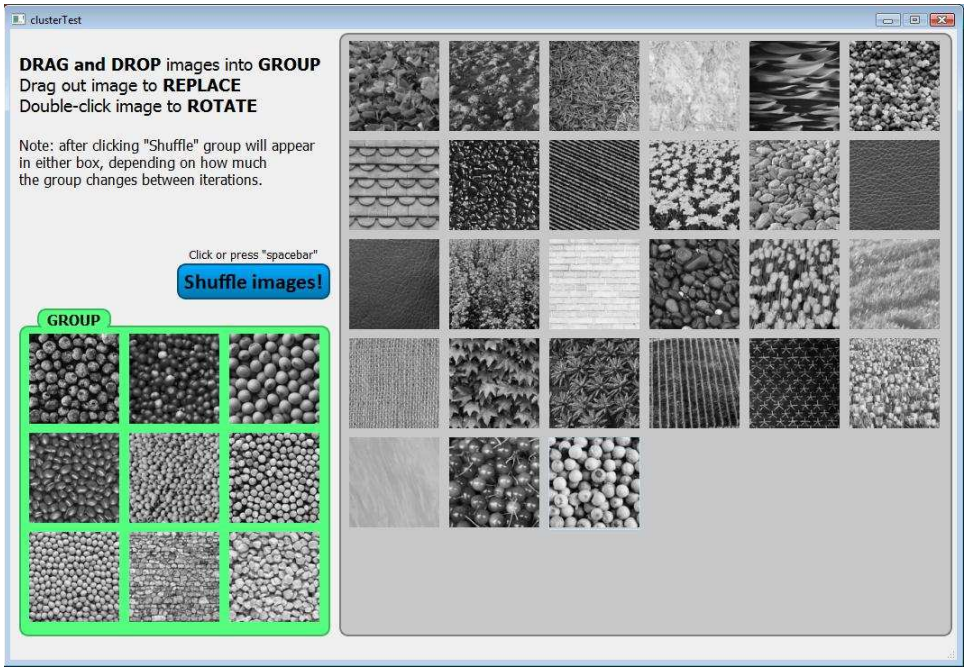


Figure 6.6. Snapshot 3 of ViSiProG interface, immediately follows Snapshot 2 above, showing a new texture in the lower right corner of the green box that blends better with the rest of the textures in the box.

several times. However, after the first update, the selection process is not uniform; the probability that an image will show up in the new batch is inversely proportional to the number of times it was selected to show up in a batch, i.e., the number of times the subject has seen and rejected it. This holds for all the subsequent iterations, ensuring that the subject will be able to see all the N images without going through too many steps in the procedure.

From the second iteration on, every step is the essentially the same. The subject is presented with a batch of N_b images, N_g from the previous group and $N_b - N_g$ newly selected ones, and she/he can use any of the N_b to form the current group. There is no restriction on the number of images from the previous group the subject has to keep, i.e., she/he can completely change the groups between two iterations, if she/he can find a more similar set of images. However, if the groups in two consecutive iterations overlap by more than 50%, in the following iteration the chosen group will stay together in the box. On the other hand, if the groups overlap by less than 50%, all the N_b images in the following iteration will be shuffled before they are presented to the subject, so that the grouping will start from scratch. This feature exists to ensure that the subject does not feel forced to refine the group she/he selected in the first iteration, but is allowed to drift until converging to a stable group.

The iterations continue until the subject has been exposed to all the images in the database. When every image in the database has been displayed at least once, the subject is given the option to end the procedure. This is indicated by the appearance of a special (red) button, which the subject may click to finish the procedure, as shown in Figure 6.7. Note that the group is much more cohesive. However, the subject is allowed to continue

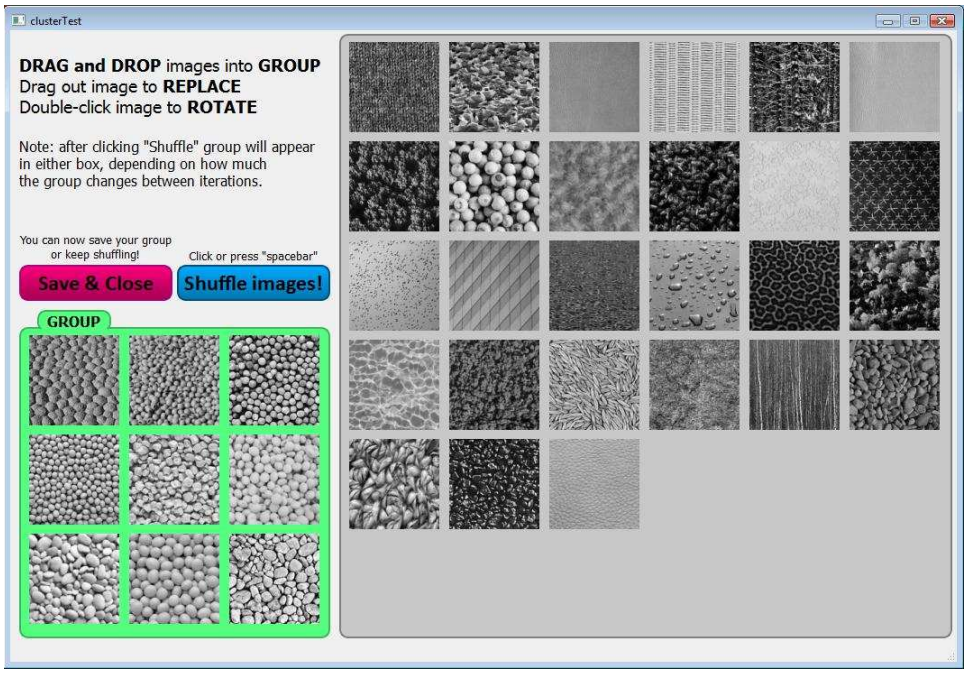


Figure 6.7. Snapshot 4 of ViSiProG interface (grayscale similarity test)

the iterations indefinitely, until she/he is satisfied with the similarity group or loses hope of improving it. Thus, in effect, the test has only a lower bound on duration – it may not end until the subject has seen all the images. Note that toward the later stages of the test, before or after the red button appears, when the similarity group has stabilized, the subject is more or less "hunting" for textures to refine the group.

6.5. Quantification of Texture Distortions

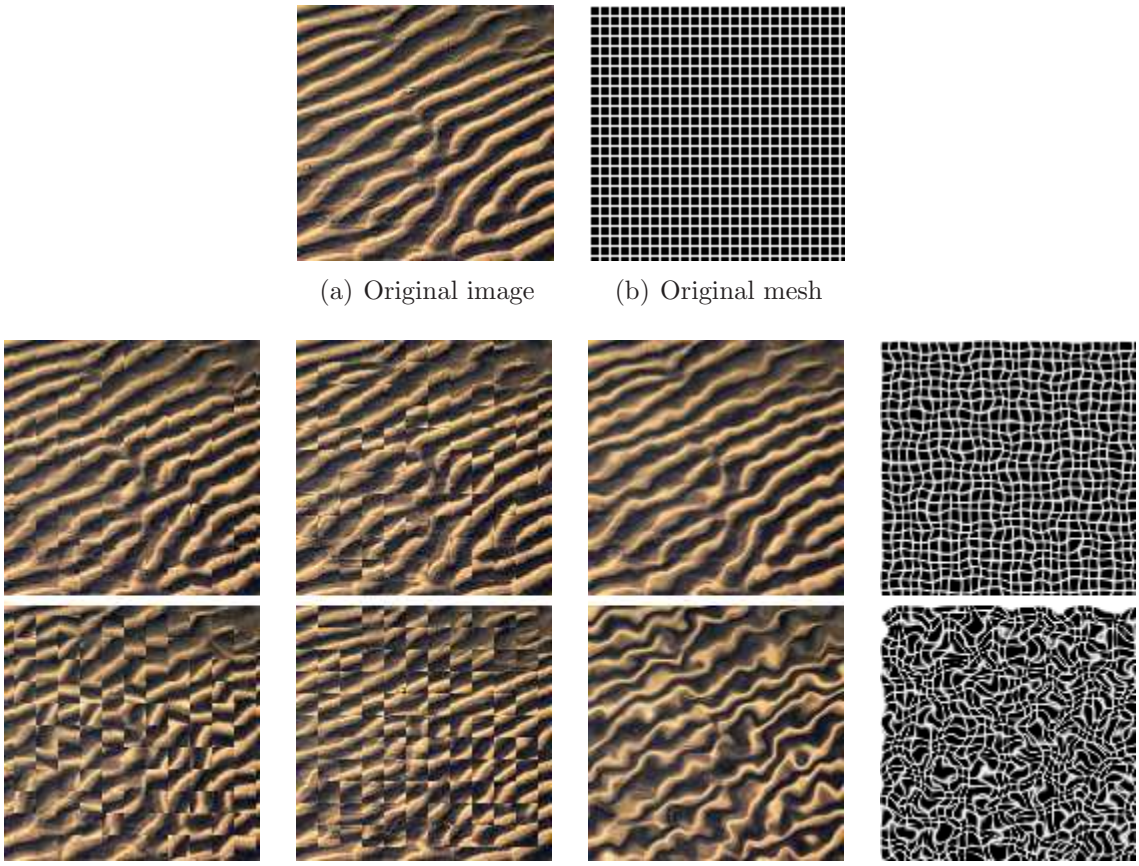
We now turn to the third application, image compression, and in particular, "structurally lossless," or nearly structurally lossless compression. Here the goal is for the texture similarity metric to give the correct ordering of images according to perceived similarity. As we discussed, such a metric can be used for quality assessment and as a

tool within a compression algorithm. The metric is expected to provide a monotonic relationship between measured and perceived distortion at the high end of the scale, where the structural distortions have either a small effect on perceived quality, or do not affect perceived quality at all. Of course, after a certain point, when the quality is unacceptable, there is no need for monotonicity; the metric should simply give low values. It is also important to have an absolute measure of image similarity, so that consistent image quality can be achieved across different types of image content, both within an image and across different images, as well as across different compression techniques.

To examine the performance of different metrics for the purposes of compression, we need subjective experiments to rate various distorted versions of an original image according to their perceived quality.

More importantly, we have to select a set of original images and a set of distortions for each image. As we discussed above, the range of distortions should be at the high end of the similarity scale, and since we are interested in testing the monotonic behavior of a metric and whether it provides an absolute distortion scale, it is important to generate a set of images that covers fine differences in distortion level. Generating such a set would be a difficult if not impossible task in the context of real applications. We thus chose to generate synthetic distortions that model distortions that occur in real compression applications.

Since our goal is to achieve structurally lossless compression, and since textured images exhibit variations in position, orientation, and color [**84**], we implemented the following distortions. The first was *random rotation* of small local patches, the second *random shifts* of small local patches, and the third *image warping*, whereby the images are distorted



(a) Original image (b) Original mesh

(c) Local rotations (d) Local shifts (e) Mesh warping (f) Underlying mesh

Figure 6.8. Examples of texture distortions (low and high degree)

according to the random deviations of the control points of the underlying mesh. Examples are shown in Figure 6.8. The severity of each type of distortion can be easily manipulated by varying the distortion parameters (probabilistic distribution of rotations, shifts, and mesh points).

6.6. Summary of Experiments

We now summarize the experiments to be carried out in this thesis in Table 6.1.

	Known-item search	Retrieval of Similar Textures	Compression
Grayscale	•	•	•
Color	•	•	

Table 6.1. Experiments presented in the thesis

For the experiments that are done both on grayscale and color images, identical testing procedures are applied. For the grayscale experiments, the images in the testing set have been converted from color to 256-gray-level images.

CHAPTER 7

Experimental Results

In Chapter 6, we presented a general description of the experiments for testing the performance of the texture similarity metrics proposed in Chapters 3 and 4. This chapter is organized as follows. First, we will introduce different performance evaluation measures that can be applied to both of the retrieval tasks that we have considered, known-item search and retrieval of similar textures, followed by the experimental results for those two tasks. Finally, we present the performance criteria and results for the texture distortion experiment.

7.1. Evaluation of Performance for Retrieval Applications

The results of known-item search and the retrieval of similar textures can be analyzed using the same performance criteria, since they are both retrieval algorithms for which standard evaluation procedures have been established.

For retrieval applications, the general performance evaluation setup is as follows: a query image is being compared to each and every image in the database, using one of the similarity metrics. The goal of the analysis of the results is to determine how well a metric retrieves textures that are "relevant" to the query. In case of known-item search, all the "identical" textures, i.e., all the images originating from the same, larger texture, are considered to be relevant to each other, while for the retrieval of similar textures, all

the images that belong to the same similarity cluster are considered to be relevant to each other.

7.1.1. Performance Based on Information Retrieval Statistics

For each query image i, the remaining images in the database can be ordered according to decreasing similarity scores with the query image. Thus, the first retrieved document is the image with highest similarity to the query; the second retrieved document is the ones with the second-highest similarity, etc.

One informative measure of performance is the number of times the first retrieved image is a *relevant* one; in the case of the known-item search, this means that it comes from the same original image, and in the case of retrieval of similar textures, this means that it belongs to the same similarity cluster. This is commonly referred to as *precision at one*.

Another way of assessing the performance of the various metrics is to compute the *mean reciprocal rank (MRR)*, which is the average value of the inverse rank of the first correctly retrieved image [**95**]. A "correct" retrieval is the one that returns a relevant image. This parameter tells us, on average, how far away from the first *retrieved* image the first *relevant* image is.

Since for each texture in our database there are typically several "identical" textures, in the known-item search experiments, there may be several relevant images for each query image. Also, in the retrieval of similar textures experiments, there are usually more than two images in a similarity cluster, and thus more than one relevant images for each query. In such cases, the usual value to report is *mean average precision (MAP)* [**96**]. The MAP

is calculated as follows: for each query, we calculate the precision (the ratio of number of relevant documents retrieved and the total number of retrieved documents) for all possible numbers of retrieved documents: the first one, the first two, etc., all the way to $N-1$ images, N being the total number of images in a database. Then, the precisions of the sets of retrieved images for which the last retrieved image was identical to the query are averaged to obtain an average precision value for every image in the database. Finally, we average these values across all images.

For image i, let us denote the number of relevant documents in the database as n_i, that is, there are a total of n_i images in the dataset (excluding image i) that are relevant to image i. Also, let us define the indicator function $I_i(j)$, such that $I_i(j) = 0$ if the j^{th} retrieved image is not relevant to image i, and $I_i(j) = 1$ if it is. If we define $p_i(r)$ as the precision of retrieval when we retrieve r images, then the average precision is defined as:

$$P_{avg,i} = \frac{1}{n_i} \sum_{r=1}^{N-1} p_i(r) \cdot I_i(r) \tag{7.1}$$

with precision when we retrieve r documents being

$$p_i(r) = \sum_{j=1}^{r} \frac{I_i(j)}{r}. \tag{7.2}$$

Therefore, the mean average precision is:

$$\text{MAP} = \underset{i}{\text{mean}}\, P_{avg,i}. \tag{7.3}$$

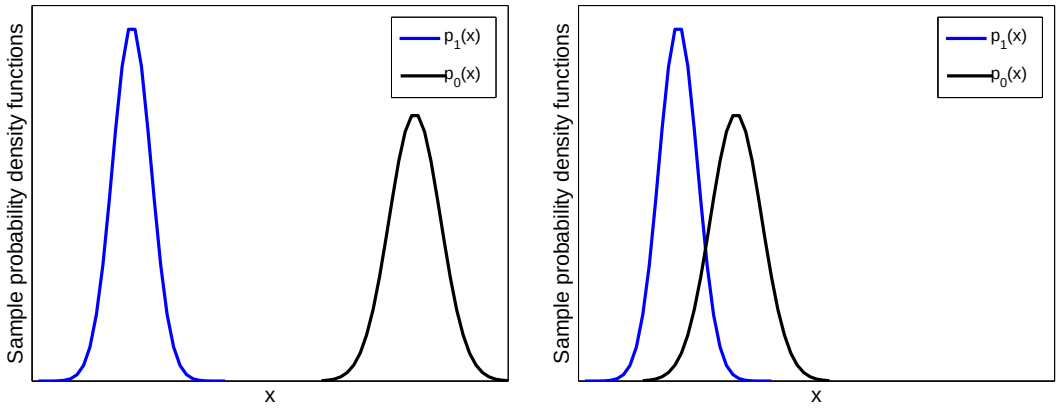
7.1.2. Performance Based on the Receiver Operating Characteristic

Another way to compare metric performance is to plot receiver operating characteristic (ROC) curves and measure the area underneath them. We can treat known-item search

as a binary classification problem, where the task is to determine whether two images are "identical" textures or not. A similar reasoning follows for the retrieval of similar textures experiments: here, the task is to determine whether two images belong to the same similarity cluster or not. In summary, for this evaluation method, the main task is to determine the relationship between two images, i.e., whether they are relevant to each other or not.

The test variable is the similarity value a metric produces for two images. Our system should decide whether the two images are relevant to each other by comparing the probability of the given similarity value under two hypothesis, $\mathcal{H}_0$ being that images are relevant and $\mathcal{H}_1$ being that they are not. Each hypothesis is associated with a different distribution of metric values, which are approximated as normalized histograms. The probability density function for $\mathcal{H}_0$ is obtained by pooling together all the values of a metric that correspond to pairs of textures that are relevant to each other, and the density function for $\mathcal{H}_1$ is obtained by pooling together all the values of a metric that correspond to pairs of non-relevant textures. If we plot the two sample probability density functions, as in Figure 7.1, ideally they should be completely separate as in Figure 7.1(a), and thus, a misclassification would never occur. However, it is realistic to expect some overlap as in 7.1(b). In such cases, we can characterize the performance of the system by plotting the ROC curve for given probability density functions $p_0(x)$ and $p_1(x)$.

The ROC curve represents the true positives rate (TPR) as a function of the false positives rate (FPR). Given a threshold t, the true positives rate is the cumulative distribution function $P_0(t)$, while the false positives rate is the cumulative distribution function $P_1(t)$.



(a) Ideal pdfs (b) Realistic pdfs

Figure 7.1. Ideal and realistic probability density functions for detection of "identical" textures

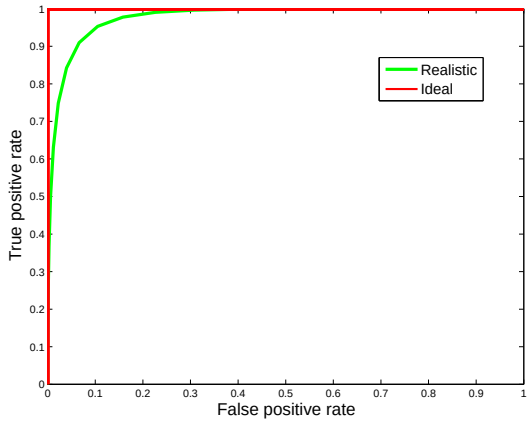


Figure 7.2. Example ROC curves

Given the probability density functions in Figure 7.1, we can draw the ROC curves as in Figure 7.2. The area under the curves can be used as a measure of performance. Ideally, the area is equal to 1.

The results of performance analysis, based on the presented two methods, are given in the subsequent sections.

7.2. Known-Item Search Experiment Results

In known-item search experiment, "identical" textures originating from 425 different original images were used. This resulted in a dataset consisting of 1180 images and each image was assigned one of the 425 distinct labels. The images with the same labels were the "identical" textures and thus considered to be "relevant" to each other. A minimum of two and a maximum of twelve images had the same label.

The setup of the experiment is straightforward. To assess one metric's performance, a 1180×1180 matrix is populated with the similarity scores that the metric produces. If we denote this matrix by M, then $M(i, j)$ gives the similarity score between images i and j, respectively. Some of the metrics need not be symmetrical, thus $M(i, j)$ is not necessarily always the same as $M(j, i)$.

The known-item search was performed on both grayscale and color texture images, and the numerical results are given in the following subsections.

7.2.1. Grayscale Texture Similarity Results for Known-Item Search

The grayscale similarity metrics we compared are:

(1) PSNR

(2) Structural Similarity Metric, SSIM, local window 7×7

(3) Complex-Wavelet Structural Similarity Metric, CWSSIM, local window 7×7

(4) Complex-Wavelet Structural Similarity Metric, CWSSIM, global window

(5) Structure Texture Similarity, STSIM, local window 7×7

(6) Structure Texture Similarity, STSIM, global window

(7) Proposed method, Structure Texture Similarity 2, STSIM2, local window 7×7

Algorithm	Precision at one	Mean reciprocal rank	Mean average precision
PSNR	0.04	0.07	0.06
SSIM	0.09	0.11	0.06
CWSSIM	0.39	0.46	0.40
CWSSIM global	0.27	0.36	0.28
STSIM	**0.74**	**0.80**	0.72
STSIM global	0.86	0.90	0.81
STSIM2	**0.74**	**0.80**	**0.74**
STSIM2 global	0.93	0.95	0.89
STSIM2-M	**0.96**	**0.97**	**0.92**
Do *et al.*	0.84	0.89	0.80
Ojala *et al.*	0.87	0.89	0.82

Table 7.1. Information retrieval statistics, known-item search experiment, grayscale texture similarity

(8) Proposed method, Structure Texture Similarity 2, STSIM2, global window

(9) Proposed method, Structure Texture Similarity 2 features with Mahalanobis distance, STSIM2-M

(10) Method of Do *et al.* [**35**], Kullback-Leibler distance on wavelet subbands

(11) Method of Ojala *et al.* [**27**], Local Binary Patterns

The implementation of the texture similarity algorithm of Do *et al.* [**35**] was downloaded from the author's website. The method of Ojala *et al.* [**27**] was downloaded from the authors' website and uses the $LBP^{riu2}_{8,1} + LBP^{riu2}_{24,3}$ combination of features. It should be noted that both of these methods are global, i.e., the images are compared based on the globally-computed features.

7.2.1.1. Information retrieval statistics. The results are summarized in Table 7.1. The highest value in each category is highlighted in black for global methods and in gray for local methods. According to these results, the proposed grayscale texture similarity

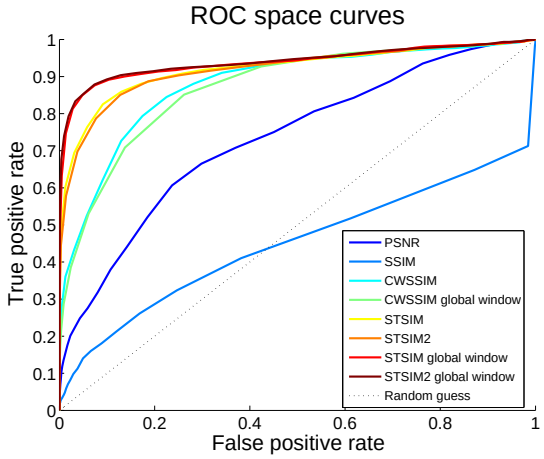


Figure 7.3. ROC curves, known-item search, grayscale texture similarity

methods outperform the remaining algorithms.

Another thing to note is that the global methods have significantly better performance than the local, sliding window-based ones. This can be explained by the fact that we are comparing more or less homogeneous texture images, and thus, want to capture global, overall texture statistics, rather than to compare images in (corresponding) small regions. The small sliding windows, on the other hand, may not capture the statistics of the homogeneous texture. For example, for images of higher scales, the small window may only capture a few texture elements, thus making the patch under observation a non-texture. This may be in the future solved by coupling the structural texture similarity metric with a texture and scale detector, where the metric would be able to adaptively choose the window size so that the basic assumption of comparing uniform textures is not violated within the sliding window.

Algorithm	AUC
PSNR	0.73
SSIM	0.45
CWSSIM	0.88
CWSSIM global	0.87
STSIM	0.92
STSIM global	**0.94**
STSIM2	0.92
STSIM2 global	**0.94**
STSIM2-M	**0.94**
Do *et al.*	–
Ojala *et al.*	0.61

Table 7.2. Area under the ROC curve, known-item search experiment, grayscale texture similarity

7.2.1.2. ROC curves. ROC curves extracted from the results of grayscale texture similarity experiments are given in Figure 7.3. For simplicity, only the curves for first 8 metric are plotted.

The areas under the curves (AUC) are given in Table 7.2. Notice that this type of analysis was not possible on the results of algorithm by Do *et al.* [**35**], since it returns singular values for some pairs of images.

Again, the global algorithms outperform the local ones and the performance results are consistent with the information retrieval statistics.

7.2.2. Color Texture Similarity Results for Known-Item Search

The color similarity metrics we compared are:

(1) Quantizing the colors according to the codebook of Mojsilovic *et al.* [**12**]

(2) Proposed method, using global averages ($k_1 = 5$) as dominant colors

Algorithm	Precision at one	Mean reciprocal rank	Mean average precision
Quantization by codebook	0.76	0.83	0.77
Global Averages	**0.93**	**0.95**	0.91
Quantization of local averages	**0.93**	**0.95**	**0.92**

Table 7.3. Information retrieval statistics, known-item search experiment, color texture similarity

(3) Proposed method, using quantized local averages ($k_2 = 32$) as dominant colors

7.2.2.1. Information retrieval statistics. The results are given in Table 7.3 and the best performing method is highlighted.

The results suggest that applying a rigid quantization method to images yields worse results than the adaptive color quantization. The benefit of having a universal codebook is that it can significantly reduce the amount of memory needed to store the color feature vectors of images and it can also reduce processing time, since the colors in use are known ahead of time and their differences can be pre-computed and stored. Thus, for applications where the processor time and memory storage are critical, it may be better to use a fixed codebook, at the expense of lower performance. For a finer assessment of similarity between two images, adaptive codebooks, as the one proposed in this thesis, are preferable.

7.2.2.2. ROC curves. The ROC curves are given in Figure 7.4.

For all three methods, the area under the curve was almost equal to 1, which suggests that these methods have a clear gap between the values assigned to "identical" textures and to non-identical pairs. However, there are a few cases where textures that are not identical are quite similar, and thus the metrics fail to retrieve the identical texture. Such

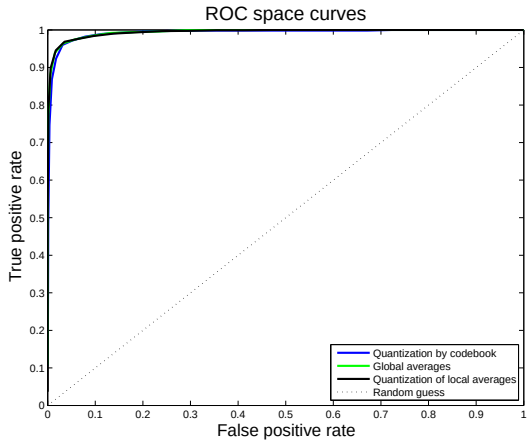


Figure 7.4. ROC curves, known-item search, color texture similarity

Algorithm	AUC
Quantization by codebook	**0.99**
Global Averages	**0.99**
Quantization of local averages	**0.99**

Table 7.4. Area under the ROC curve, known-item search experiment, color texture similarity

failures have minimal effect on the ROC curves, but are better captured by the information retrieval statistics. This explains why there is a significant difference in performance between the Mojsilovic *et al.* metric and our metrics according to the information retrieval statistics, and no difference according to the ROC curves.

7.3. Retrieval of Similar Textures Experiment Results

The goal of this experiment is to partition the dataset into subjectively similar clusters. All the images belonging to the same cluster are considered to be similar among each other, while an image is considered to be dissimilar to the images that are outside of its own cluster.

This experiment was carried out using the ViSiProG procedure, explained in detail in Chapter 6. The dataset contains a total of $N = 505$ color texture images, and they all came from distinct 505 original, larger images (i.e., there were no "identical" textures present). For the grayscale experiments, the $N = 505$ color textures were converted to 256-level grayscale images. The users are presented with $N_b = 36$ different texture images in a batch, and their task was to form a group of $N_g = 9$ similar textures in each run of the test. A graphical user interface (GUI) was developed using Qt APIs.

The same procedure (ViSiProG) and graphical user interface were used for experiments with both grayscale and color images; however, the users were instructed to judge different aspects of image similarity for those two different cases. In the grayscale experiments, the subjects were told to form the similarity groups based on the overall similarity of the images. The emphasis was on "visual blending" of the images in the similarity group, which facilitates similarity in all perceptual dimensions, in agreement with our conclusions of Chapter 5. For the color experiment, the subjects were told to judge the similarity between images based on their color composition (which includes both colors and their respective percentages). In both cases, they were instructed to focus on visual similarity, not on semantics. The subjects were asked to *ignore* any semantic information they can extract from the images, e.g., images of flowers that look different should be classified as dissimilar, and images of different things that look similar should be classified as similar.

Each subject was asked to perform four trials of the ViSiProG test, each resulting in one subjective similarity group. Note that, even though the random selection of the initial batch of textures presented to the user usually leads to different groups in different trials, there is no requirement or guarantee that a user cannot form groups that are similar to

each other. After collecting all the groups from all the subjects, we formed a graph whose vertices were texture images, and the weights of the edges are proportional to the number of times the adjacent images were placed by a user in the same group. These weights are stored in the so-called graph adjacency matrix.

The graph adjacency matrix can be analyzed using the spectral clustering algorithm [**94**]. Spectral clustering extracts the eigenvectors and eigenvalues from the modified graph adjacency matrix and the points of the eigenvectors are used to cluster the graph nodes.

Once we form the clusters, we can perform the same performance analysis as in Section 7.2. If we assign to each image a label corresponding to its cluster membership, then all the images from the same clusters are "relevant" to each other and that is considered to be the ground truth against which we compare the performance of different metrics.

7.3.1. Grayscale Texture Similarity Results for Retrieval of Similar Textures

The total number of users was 33. Not all the subjects performed 4 runs of the test and the total number of groups we collected was 126.

One particular feature of texture clustering experiments for grayscale images is that the users were allowed to rotate the images by increments of 90° to align textures' rotations, if needed. For constructing the graph adjacency matrices, we discarded the rotational information and formed clusters regardless of the chosen rotations of the images; after the clusters were formed, the images were realigned to match the user data.

Since we had 126 groups, we decided to threshold the weights of the edges so that all the edges weaker than 3 were discarded. After this initial thresholding, only 120 images

remained "active," i.e., only 120 images had at least one adjacent edge whose weight was not zero.

The spectral clustering algorithm helped us identify the 11 non-overlapping clusters. The images in clusters have no edges connecting them to images that are outside of their own cluster. The extracted 11 clusters are given in Figures 7.5 through 7.15.

In this experiment, we compared the following nine similarity metrics:

(1) PSNR
(2) Structural Similarity Metric, SSIM, local window 7×7
(3) Complex-Wavelet Structural Similarity Metric, CWSSIM, local window 7×7
(4) Complex-Wavelet Structural Similarity Metric, CWSSIM, global window
(5) Structure Texture Similarity, STSIM, local window 7×7
(6) Structure Texture Similarity, STSIM, global window
(7) Proposed method, Structure Texture Similarity 2, STSIM2, local window 7×7
(8) Proposed method, Structure Texture Similarity 2, STSIM2, global window
(9) Proposed method, Structure Texture Similarity 2 features with Mahalanobis distance, STSIM2-M

We can compute the same statistics (information retrieval statistics and the area underneath the ROC curves) as we did in the known-item search experiment.

7.3.1.1. Information retrieval statistics. For the information retrieval statistics, we treat as queries only the images that belong to a cluster, since for the ones lying outside of any cluster we do not have any relevant documents to retrieve. We do, however, keep them in the database, when querying the clustered images. Our M matrix is in effect of

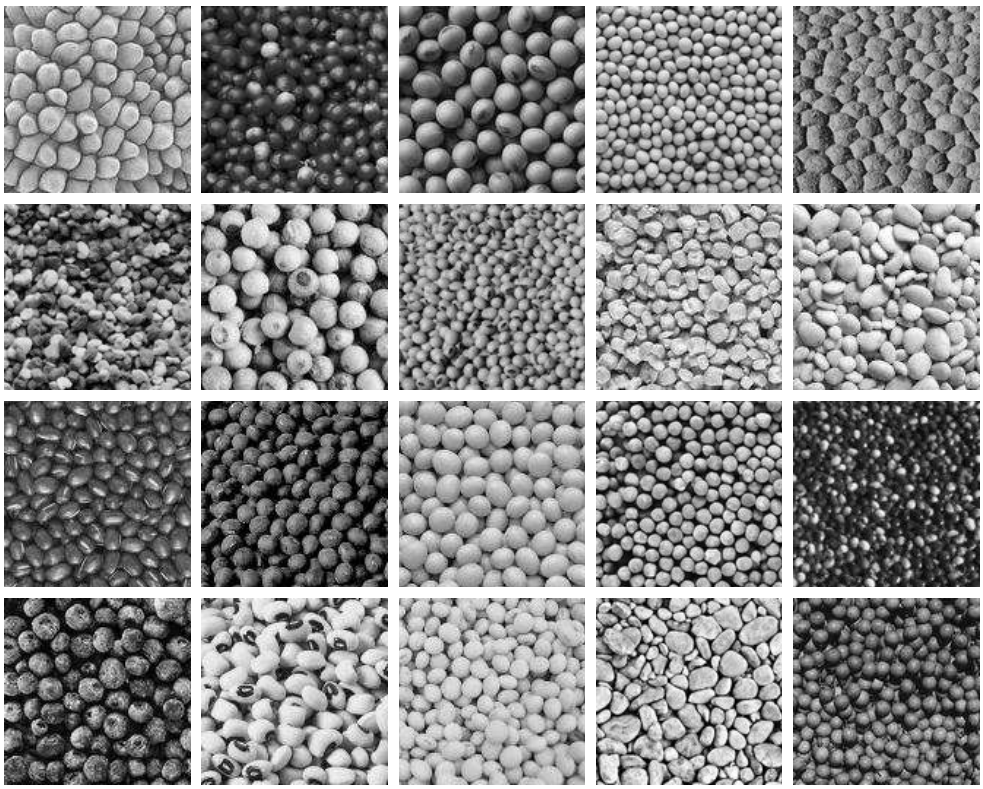


Figure 7.5. Grayscale cluster 1

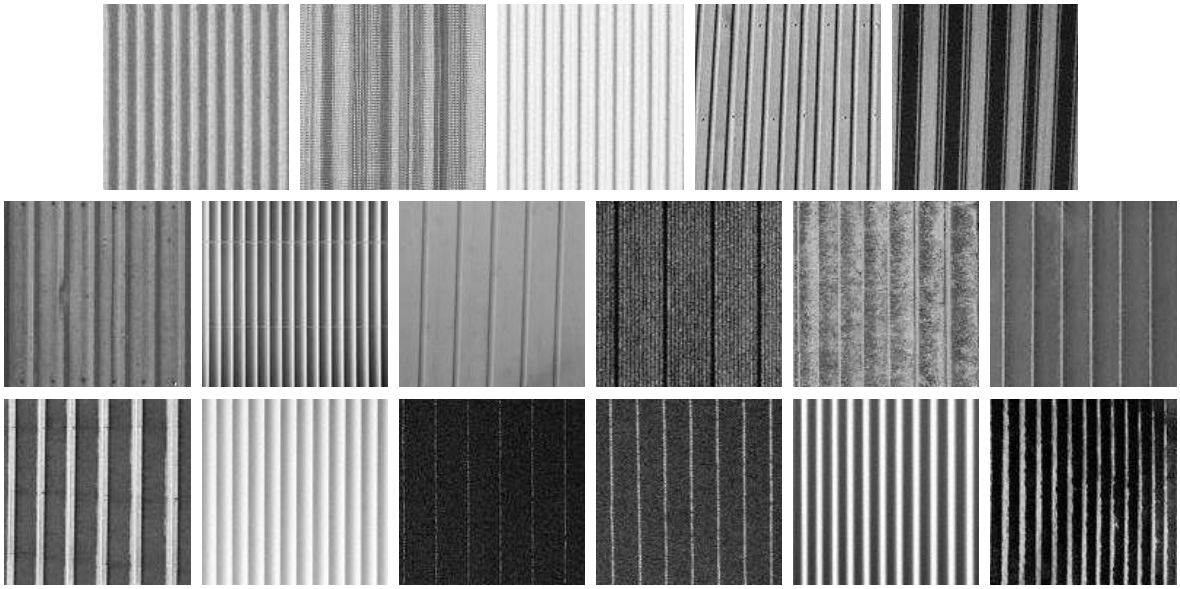


Figure 7.6. Grayscale cluster 2

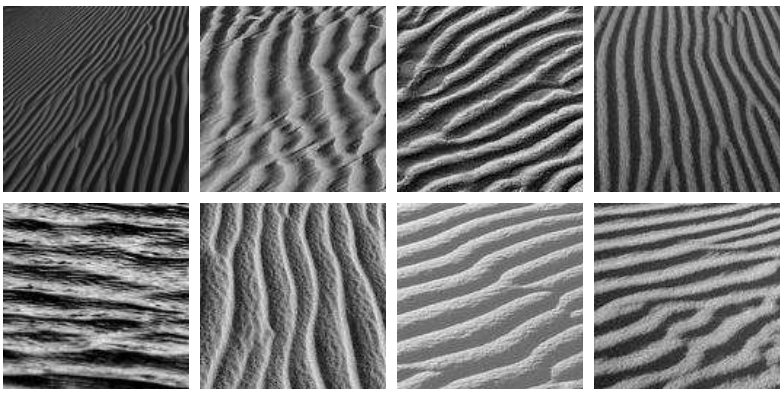


Figure 7.7. Grayscale cluster 3

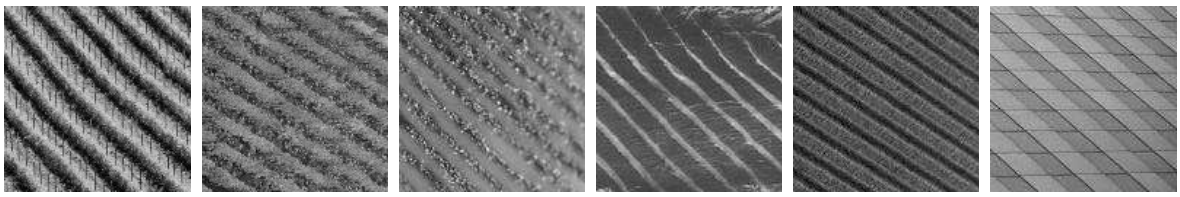


Figure 7.8. Grayscale cluster 4

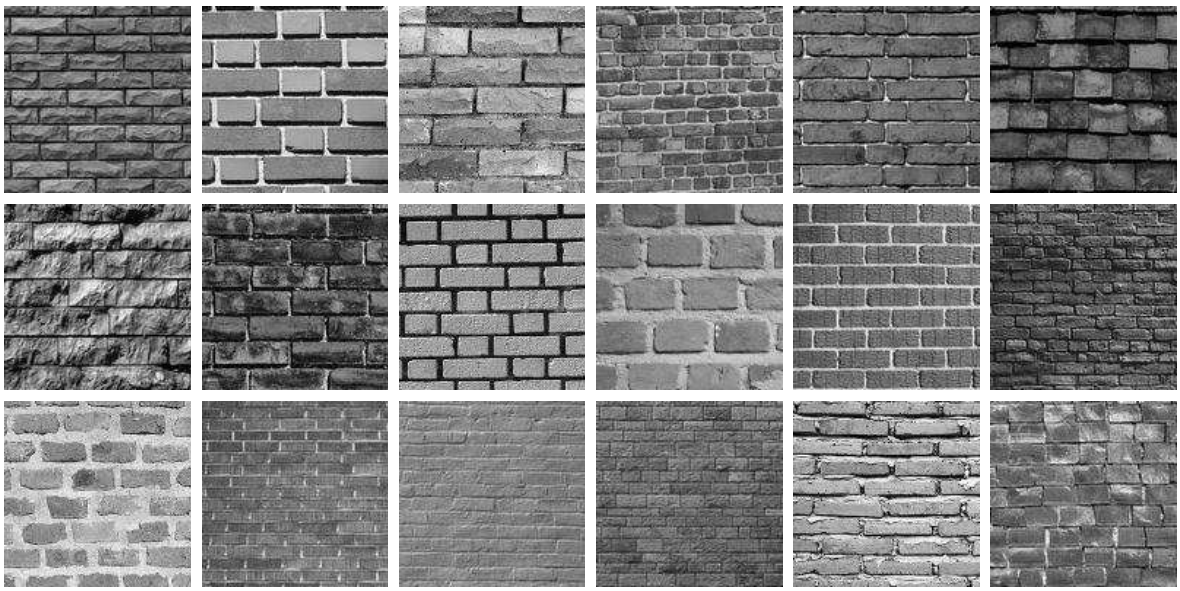


Figure 7.9. Grayscale cluster 5

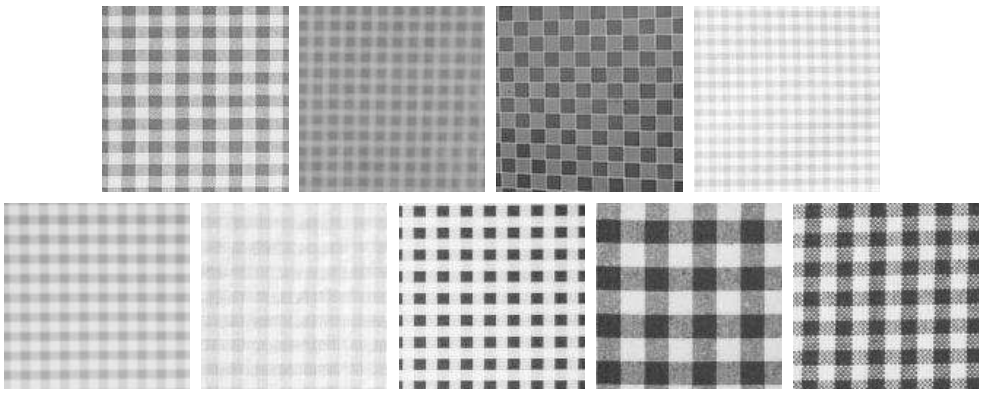


Figure 7.10. Grayscale cluster 6

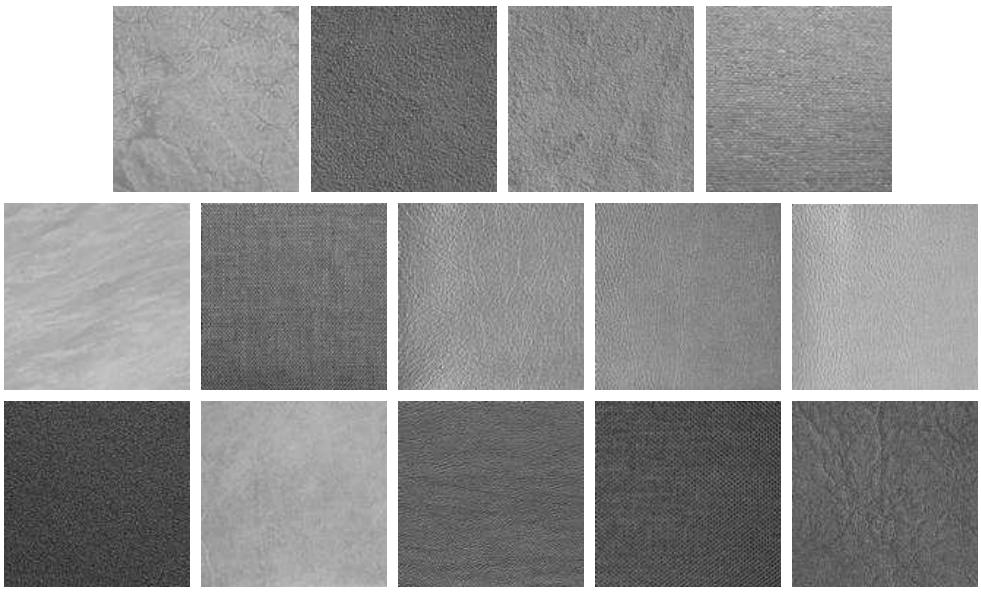


Figure 7.11. Grayscale cluster 7

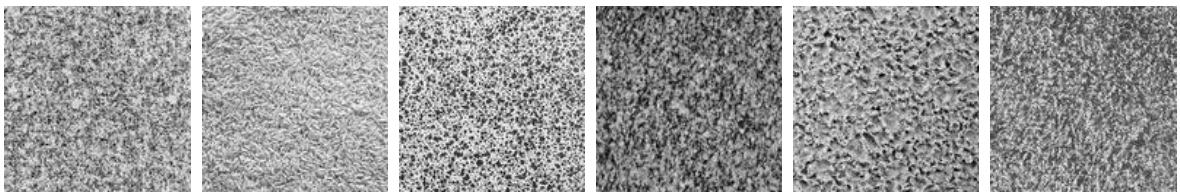


Figure 7.12. Grayscale cluster 8

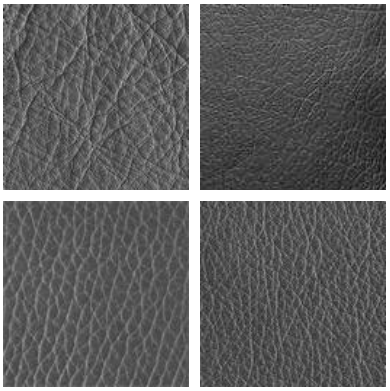


Figure 7.13. Grayscale cluster 9

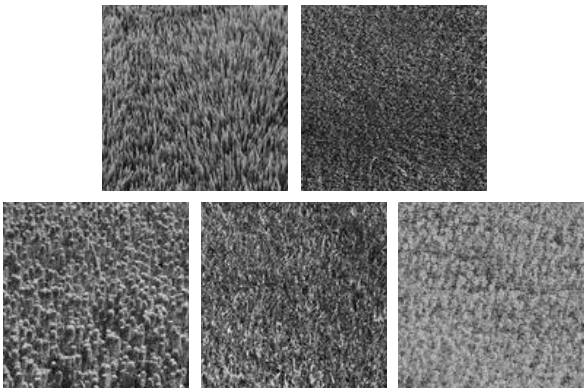


Figure 7.14. Grayscale cluster 10

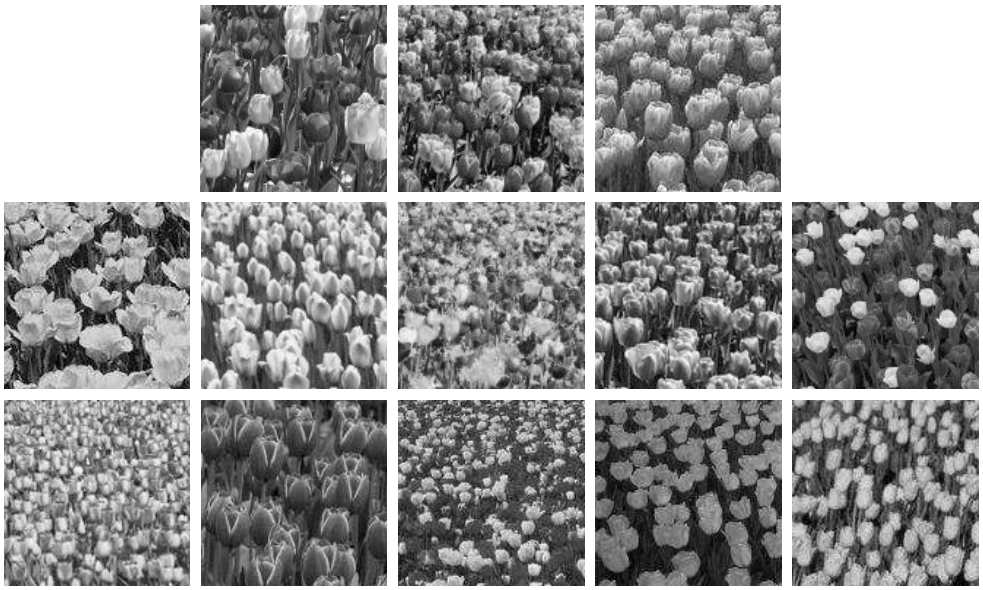


Figure 7.15. Grayscale cluster 11

Algorithm	Precision at one	Mean reciprocal rank	Mean average precision
PSNR	0.09	0.14	0.06
SSIM	0.23	0.33	0.10
CWSSIM	0.52	0.67	0.31
CWSSIM global	0.38	0.54	0.25
STSIM	0.56	0.67	0.32
STSIM global	0.48	0.62	0.30
STSIM2	**0.62**	**0.71**	**0.37**
STSIM2 global	0.48	0.62	0.30
STSIM2-M	0.45	0.58	0.24

Table 7.5. Information retrieval statistics, clustering experiment, grayscale texture similarity

size 120×505, since we only have 120 queries for which we know the ground truth, i.e. the images relevant to them. The results are presented in Table 7.5.

Again, one of the proposed metrics outperforms the other algorithms. Interestingly, this time the local metrics consistently perform better than the global ones. This suggests that applying local windows better captures the texture similarities, even though it is an inferior algorithm for identical texture retrieval.

7.3.1.2. ROC curves. When we perform the signal detection analysis (ROC curves), we get results as in Table 7.6.

The performance evaluation based on the areas underneath the ROC curves is consistent with the performance evaluation we got from the information retrieval statistics. Local metrics outperform the global ones consistently, which strengthens the argument that we need different metrics that would adapt to different applications.

Algorithm	AUC
PSNR	0.51
SSIM	0.53
CWSSIM	**0.82**
CWSSIM global	**0.82**
STSIM	0.80
STSIM global	0.79
STSIM2	**0.82**
STSIM2 global	0.77
STSIM2-M	0.71

Table 7.6. Area under the ROC curve, clustering experiment, grayscale texture similarity

7.3.2. Color Texture Similarity Results for Retrieval of Similar Textures

For this experiment we only had 9 participants, yielding a total of 37 groups. Again, we thresholded the graph adjacency matrix, but this time we only removed the links of strength 1. This left us with a total of 75 images that have at least one adjacent edge whose weight is not zero. Spectral clustering identified 10 non-overlapping clusters that are illustrated in Figures 7.16 through 7.25.

For the clustering experiment, we used the same color similarity metrics:

(1) Quantizing the colors according to the codebook of Mojsilovic *et al.* [**12**]

(2) Proposed method, using global averages ($k_1 = 5$) as dominant colors

(3) Proposed method, using quantized local averages ($k_2 = 32$) as dominant colors

7.3.2.1. Information retrieval statistics. Information retrieval statistics are reported in Table 7.7.

In this experiment, the proposed method is better than non adaptive quantizing of the colors. What is also interesting to note is that the information retrieval statistics

Figure 7.16. Color cluster 1

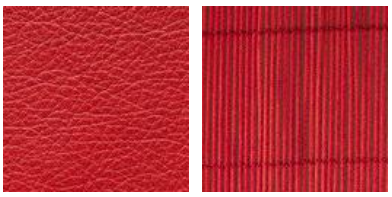


Figure 7.17. Color cluster 2

Figure 7.18. Color cluster 3

Figure 7.19. Color cluster 4

Figure 7.20. Color cluster 5

Figure 7.21. Color cluster 6

Figure 7.22. Color cluster 7

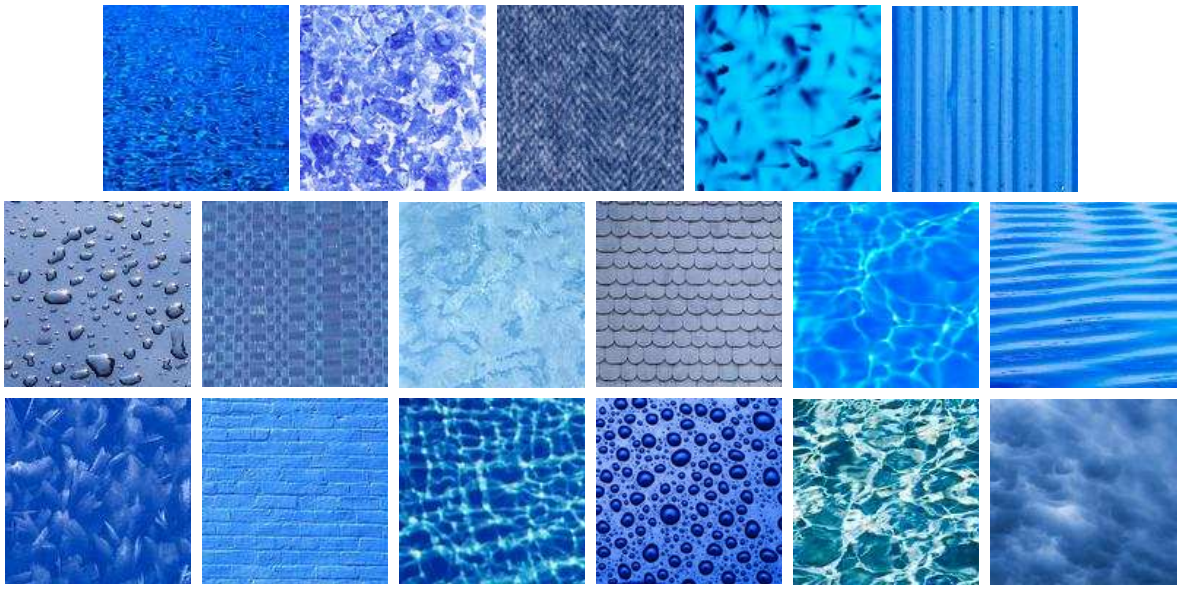


Figure 7.23. Color cluster 8

Figure 7.24. Color cluster 9

Figure 7.25. Color cluster 10

Algorithm	Precision at one	Mean reciprocal rank	Mean average precision
Quantization by codebook	0.37	0.56	0.40
Global Averages	**0.49**	**0.65**	0.46
Quantization of local averages	**0.49**	**0.65**	**0.47**

Table 7.7. Information retrieval statistics, clustering experiment, color texture similarity

are rather low, for both the grayscale and color experiments. It is usually understood that we understand colors better than the grayscale texture structure of images. We have a better understanding of the underlying physiological and psychological processes that are involved in color vision, yet, even the most sophisticated techniques for color retrieval cannot mimic human behavior. This leaves an open road for improvement in both grayscale and color image similarity metrics.

Algorithm	AUC
Quantization by codebook	**0.92**
Global Averages	**0.92**
Quantization of local averages	0.91

Table 7.8. Area under the ROC curve, clustering experiment, color texture similarity

7.3.2.2. ROC curves. The performances according to the receiver operating characteristic are given in Table 7.8.

The values for all three metrics are very similar, which supports the claim that non-adaptive color quantizing is suitable for coarse retrieval purposes.

7.4. Texture Distortion Experiment Results

Ten different grayscale texture images were chosen for this experiment. They are shown in Figure 7.26.

Each of the ten textures was distorted with three distortion algorithms described in more detail in Chapter 6 and with three degrees of severity for each distortion. The first distortion algorithm performs *random rotation* of small local patches, the second performs *random shifts* of small local patches and the third performs *image warping* where the images get distorted according to the deviations of the control points of the underlying mesh. In our experiments, the patches were of size 11×11, while the warping meshes were 5×5. This is because we found that the smaller meshes result in similar scale artifacts as those of the patches.

The original textures and their distortions are given in Figure 7.27. The original images are the first ones on the left, followed by three rotation-distorted images, three

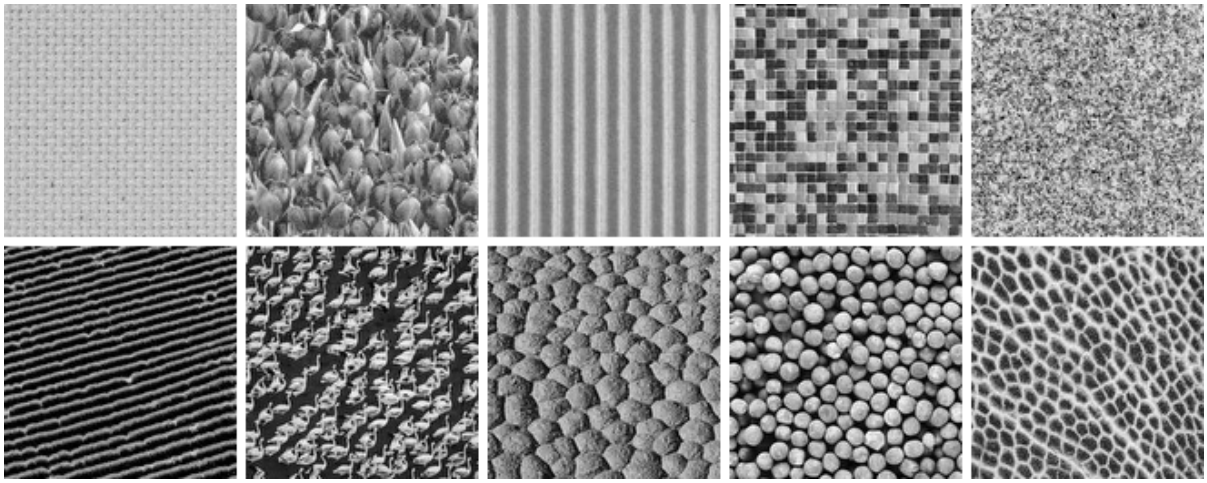


Figure 7.26. Original textures for texture distortion experiment

shift-distorted images and three warped images, with the severity of the distortions is increasing from left to right.

The subjects were asked to rank, for each of the ten originals, the distorted images from "best" to "worst," as compared to the original signal. They were not allowed to give the same ranking to two distorted images. As a result, every user gives ranks between 1 and 9 to the nine distorted images, for each of the ten originals.

This, however, produces data that can be only processed for a given original and not across originals. In order to be able to determine how well a metric performs not only for a given original, but also across different originals, the subjects were asked at the end of the test to rank the ten images they have previously chosen as the "worst" ones, as compared to the ten original images. This gives us an opportunity to see if a metric can predict the perceived distortions of a certain original, but also if it can predict the relationship between the distortions of different originals. A total of eleven subjects participated in this experiment.

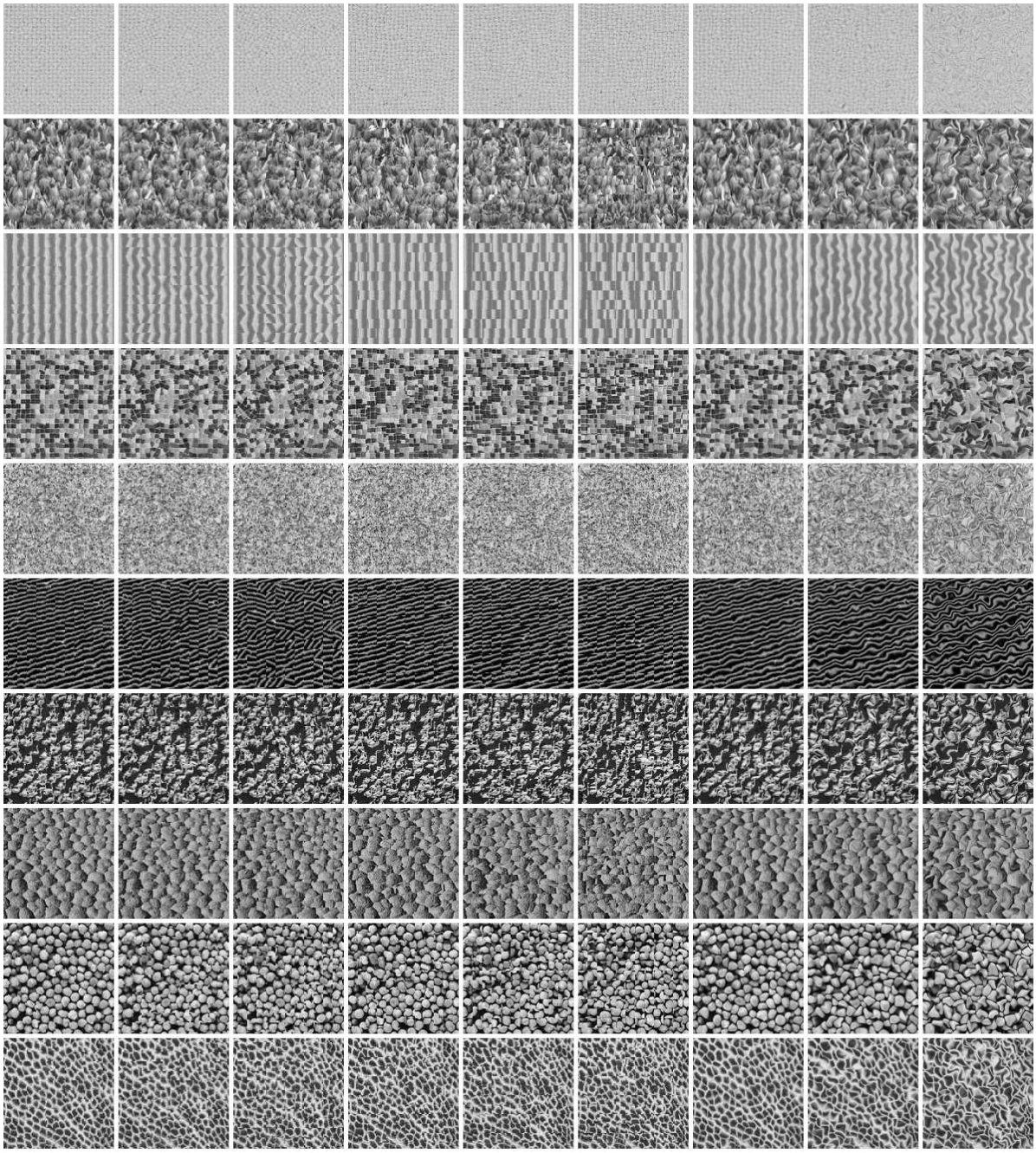


Figure 7.27. Original textures and their distortions

The metrics analyzed are the same nine metrics whose performances were reported in the texture clustering experiment:

(1) PSNR

(2) Structural Similarity Metric, SSIM, local window 7×7

(3) Complex-Wavelet Structural Similarity Metric, CWSSIM, local window 7×7

(4) Complex-Wavelet Structural Similarity Metric, CWSSIM, global window

(5) Structure Texture Similarity, STSIM, local window 7×7

(6) Structure Texture Similarity, STSIM, global window

(7) Proposed method, Structure Texture Similarity 2, STSIM2, local window 7×7

(8) Proposed method, Structure Texture Similarity 2, STSIM2, global window

(9) Proposed method, Structure Texture Similarity 2 features with Mahalanobis distance, STSIM2-M.

7.4.1. Analyzing the results of rankings per original

Analyzing the ranking data can be done in various ways. For any of the methods we choose, for each original image we extract a 1-D vector that describes the perceived similarity between the original and the nine distorted images. This vector is compared to the values a similarity metric produces. To measure the goodness of fit, we use both Pearson's correlation coefficient (because we are interested in the absolute performance of the metrics), as well as the Spearman rank correlation coefficient, which describes how well a metric ranked the distorted images, as compared to the ranking of the extracted 1-D subjective vector.

Algorithm	Pearson correlation	Spearman rank correlation
PSNR	0.72	0.67
SSIM	0.74	0.72
CWSSIM	0.84	0.81
CWSSIM global	0.73	0.73
STSIM	**0.88**	0.85
STSIM global	**0.88**	0.84
STSIM2	**0.88**	0.86
STSIM2 global	**0.88**	**0.87**
STSIM2-M	0.81	0.75

Table 7.9. Correlation coefficients between metrics and mean ranks

7.4.1.1. Borda's rule. The simplest approach is to find the mean ranking for each distorted image and use that as its "subjective" position with respect to the original. This is perhaps one of the oldest techniques, proposed in 1770 by Jean-Charles de Borda, and today usually known as Borda's rule. He called this method "election by order of merit," i.e., the cumulative preference given to a candidate is its final score. The mean values for correlation coefficients, taken across 10 originals, are given in Table 7.9.

7.4.1.2. Thurstonian scaling. One popular way to analyze this type of data is to use Thurstonian scaling [**91**]. It is applied on the *preference* matrix P, where $P(i, j)$ denotes how many times image i was preferred to image j, i.e., how many times image i was ranked as closer to the original than image j. After pooling all the results, the preference matrix is scaled to represent percentages ("image i was preferred to image j in p percent of cases") and percentages are converted into z-scores. This can produce singular values when we have perfect agreement among raters, so an alternative has been proposed by Krus *et al.* [**97**], which avoids such undesirable behavior. The correlation coefficients are reported in Table 7.10.

Algorithm	Pearson correlation	Spearman rank correlation
PSNR	0.72	0.66
SSIM	0.74	0.71
CWSSIM	0.84	0.81
CWSSIM global	0.73	0.73
STSIM	0.88	0.85
STSIM global	**0.89**	0.84
STSIM2	0.88	0.86
STSIM2 global	**0.89**	**0.87**
STSIM2-M	0.81	0.76

Table 7.10. Correlation coefficients between metrics and Thurstonian scaling results

Algorithm	Pearson correlation	Spearman rank correlation
PSNR	0.72	0.67
SSIM	0.74	0.72
CWSSIM	0.83	0.81
CWSSIM global	0.73	0.74
STSIM	**0.87**	0.85
STSIM global	**0.87**	0.82
STSIM2	**0.87**	**0.86**
STSIM2 global	**0.87**	**0.86**
STSIM2-M	0.78	0.71

Table 7.11. Correlation coefficients between metrics and multidimensional scaling results

7.4.1.3. Multidimensional scaling. Yet another way to analyze the data is to treat the ranks as distances between images. For example, the image that was ranked as number 1 and the image that was ranked as number 5 would be assigned distance of 4. After aggregating all the data from all the users, we can then perform multidimensional scaling to extract the perceptual dimensions embedded in the data. The results for this type of analysis are reported in Table 7.11.

7.4.1.4. Summary of performance analysis. From the results given in Tables 7.9, 7.10 and 7.11, the absolute performances of the proposed metrics do not change much across the different methods; also, analyzing the data in different ways consistently puts the proposed structural texture similarity metric ahead of or tied with the other algorithms in terms of performance. High Pearson correlation coefficients indicate that the metric can in fact predict the perceived distortion in texture images.

7.4.2. Analyzing the ranking results across originals

In the final step of the test, each user had to rate 10 images they previously labeled as the "worst" ones, from "best" to "worst." Thus, every user gives a ranking of a subset of all possible pairs of images across different originals. Given that there are 9^{10} possible different subsets, it is clear that the data gathered in this manner produces very sparse matrices.

Methods for comparing the similarity or preference matrices with incomplete data do exist. However, in this case, when analyzed with non-metric multidimensional scaling techniques [**98**], the results are very unreliable, due to the high sparsity of the formed preference matrix.

An alternative is to use a metric that estimates the agreement between the subjective scores and a metric's scores. One possible test to perform is Kendall coefficient of agreement [**99**], which is designed to measure inter-rater agreement. To analyze the performance of a metric, we can treat it as yet another rater and then compute the joint agreement between the metric values and the subjective scores. The higher the overall coefficient of agreement, the better the metric represents the subjective data. Values for

Algorithm	Kendall agreement coefficient
PSNR	0.53
SSIM	0.60
CWSSIM	0.66
CWSSIM global	0.63
STSIM	0.67
STSIM global	0.71
STSIM2	0.68
STSIM2 global	**0.73**
STSIM2-M	0.72

Table 7.12. Kendall agreement coefficients between metrics and subjective results

this test are reported in Table 7.12. In this experiment, the best performing similarity metric was the proposed STSIM2 computed over a global window. A close second is the proposed STSIM2-M, again a global metric.

7.5. Summary of Experiment Results and Performance Evaluations

The results of subjective and objective experiments presented in the previous sections show that the proposed methods outperform the existing similarity metrics. This is summarized in Table 7.13. The reported values for CBIR-type experiments (known-item search and retrieval of similar textures) are precision at one for the best-performing existing similarity metric, and the best-performing metric overall, which is always one of the metrics proposed in this thesis. For the compression experiment, the values in the table are the Pearson correlation coefficients between the subjective scores derived using multidimensional scaling and the metric values – the best performing existing one and the best overall, which is again the proposed metric, STSIM2.

	Known-item search	Retrieval of Similar Textures	Compression
Grayscale	0.87 ⇒ **0.96** (10%)	0.52 ⇒ **0.62** (19%)	0.83 ⇒ **0.87** (5%)
Color	0.76 ⇒ **0.93** (22%)	0.37 ⇒ **0.49** (32%)	

Table 7.13. Improvements in performance evaluation statistics for different experiments

CHAPTER 8

Conclusions and Future Work

We developed new "Structural Texture Similarity Metrics" that are based on an understanding of human visual perception and incorporate a broad range of texture region statistics. We proposed separate metrics for the grayscale component of texture and its color composition. This is consistent with the fact that texture image structure and color composition are different perceptual dimensions. While we cannot claim that image structure and color composition are perceptually independent, nor that image structure is equivalent to grayscale structure, our results indicate that our approach provides a reasonable approximation in the context of content-based retrieval and compression applications.

The proposed grayscale texture similarity metrics are based on a steerable filter decomposition and incorporate texture region statistics, including the mean, variance, first order correlation coefficients, as well as cross-subband correlation coefficients. A number of variations of the basic metric are targeted at different applications and performance requirements. We also proposed a texture color composition texture similarity metric, whereby each image is represented by a feature vector that consists of the dominant colors and the associated percentages. The comparison between two feature vectors is based on the Optimal Color Composition Distance. The appropriate combination of the two metrics is left to the user and the demands of the target application.

A major contribution of this thesis is a new methodology for systematic performance evaluation of texture similarity metrics, which considerably simplifies the testing procedures, and dramatically increases the chances of obtaining consistent subjective results. It is based on the realization that it is difficult for both humans and machines to quantify similarity when textures are dissimilar, and that this should be limited to the high end of the similarity scale. In content-based image retrieval (CBIR), the goal is to distinguishing between similar and dissimilar textures, while in image compression, the goal is to quantify similarity (only) at the high end of the similarity scale. For CBIR, we developed "Visual Similarity by Progressive Grouping (ViSiProG)," a new experimental procedure for subjective grouping of similar textures that provides a benchmark for testing texture similarity metrics. For image compression, we developed algorithms for generating texture deformations that we used to carry out subjective and objective tests. We also considered "known-item search," an important special case of CBIR where the goal is to identify "identical" textures, and where the ground truth is obtained by careful construction of the database without the need for extensive subjective tests.

Another key contribution of this thesis is the collection of a large database of approximately 1500 color texture images. The database was carefully constructed to meet the demands of the CBIR and compression applications. We conducted extensive subjective and the objective tests to evaluate the performance of the proposed texture similarity metrics. Our results demonstrate that the use of the proposed metrics in texture retrieval and compression applications provides superior performance to what can be obtained on the basis of existing metrics.

This thesis research represents a significant step toward a better understanding of the texture similarity problem. However, substantial issues remain to be investigated. For example, there is a need to develop metrics that evaluate image similarity along specific perceptual dimensions, such as scale, orientation, regularity, and roughness. As we saw, a better understanding of the relationship between color composition and texture structure would also be a great topic for future research; such research could build on the results of Section 4.4. Another important direction for future research is the adaptation of the proposed metrics to specific applications in compression and CBIR, as well as the consideration of entirely different problem areas, such as multimodal signal analysis.

References

[1] M. F. Bear, B. Connors, and M. Paradiso, *Neuroscience: Exploring the Brain (Third Edition)*. Lippincott Williams & Wilkins, February 2006.

[2] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[3] T. N. Pappas, R. J. Safranek, and J. Chen, "Perceptual criteria for image quality evaluation," in *Handbook of Image and Video Processing*, 2nd ed., A. C. Bovik, Ed. Academic Press, 2005, pp. 939–959.

[4] P. Ndjiki-Nya, T. Hinz, and T. Wiegand, "Generic and robust video coding with texture analysis and synthesis," in *IEEE International Conference on Multimedia and Expo, 2007*. IEEE, 2007, pp. 1447–1450.

[5] T. N. Pappas, J. Zujovic, and D. L. Neuhoff, "Image analysis and compression: Renewed focus on texture," in *Visual Information Processing and Communication*, vol. Proc. SPIE, Vol. 7543, San Jose, CA, Jan. 2010.

[6] T. N. Pappas, V. Tartter, A. G. Seward, B. Genzer, K. Gourgey, and I. Kretzschmar, "Perceptual dimensions for a dynamic tactile display," in *Human Vision and Electronic Imaging XIV*, ser. Proceedings of SPIE, B. E. Rogowitz and T. N. Pappas, Eds., vol. 7240. San Jose, CA: SPIE, January 2009, pp. 72 400K–1–12.

[7] T. Narumi, S. Nishizaka, T. Kajinami, T. Tanikawa, and M. Hirose, "Meta cookie+: An illusion-based gustatory display," *Virtual and Mixed Reality-New Trends*, pp. 260–269, 2011.

[8] D. Van Essen and J. Maunsell, "Hierarchical organization and functional streams in the visual cortex," *Trends in Neurosciences*, vol. 6, pp. 370–375, 1983.

[9] K. Tanaka, "Mechanisms of visual object recognition: monkey and human studies," *Current opinion in neurobiology*, vol. 7, no. 4, pp. 523–529, 1997.

[10] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multiscale transforms," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 587–607, 1992.

[11] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 703–715, 2001.

[12] A. Mojsilović, J. Hu, and E. Soljanin, "Extraction of perceptually important colors and similarity measurement for image matching, retrieval, and analysis," *IEEE Trans. Image Process.*, vol. 11, no. 11, pp. 1238–1248, Nov. 2002.

[13] J. Chen, "Perceptually-based color and texture features for image segmentation and retrieval," Ph.D. dissertation, Northwestern Univ., Evanston, IL, Dec. 2003.

[14] J. Chen, T. N. Pappas, A. Mojsilovic, and B. E. Rogowitz, "Adaptive perceptual color-texture image segmentation," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1524–1536, Oct. 2005.

[15] D. Granrath, "The role of human visual models in image processing," *Proceedings of the IEEE*, vol. 69, no. 5, pp. 552–561, May 1981.

[16] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error measurement to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[17] T. Ojala, T. Menp, J. Viertola, J. Kyllnen, and M. Pietikinen, "Empirical evaluation of MPEG-7 texture descriptors with a large-scale experiment," in *Proceedings of* 2^{nd} *International Workshop on Texture Analysis and Synthesis*, 2002, pp. 99–102.

[18] M. N. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance," *IEEE Trans. Image Process.*, vol. 11, no. 2, pp. 146–158, Feb. 2002.

[19] B. Girod, "What's wrong with mean-squared error?" in *Digital images and human vision*. MIT Press, 1993, pp. 207–220.

[20] Z. Wang, A. C. Bovik, and L. Lu, "Why is image quality assessment so difficult?" in *IEEE Int. Conf. on Acoustics, Speech, and Signal Proc.*, vol. IV, 2002, pp. 3313–3316.

[21] M. P. Eckert and A. P. Bradley, "Perceptual quality metrics applied to still image compression," *Signal Processing*, vol. 70, no. 3, pp. 177 – 200, 1998.

[22] M. Tuceryan and A. K. Jain, "Texture analysis," *Handbook of pattern recognition and computer vision*, vol. 276, pp. 235–276, 1993.

[23] H. Z. Long, W. K. Leow, and F. K. Chua, "Perceptual texture space for content-based image retrieval," in *Proceedings of the International Conference on Multimedia Modeling (MMM 2000)*, Nagano, Japan, November 2000, pp. 167–180.

[24] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.

[25] P. Chen and T. Pavlidis, "Segmentation by texture using correlation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 1, pp. 64–69, 1983.

[26] R. Kashyap, R. Chellappa, and A. Khotanzad, "Texture classification using features derived from random field models," *Pattern Recognition Letters*, vol. 1, no. 1, pp. 43–50, 1982.

[27] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, pp. 971–987, 2002.

[28] S. K. Saha, A. K. Das, and B. Chanda, "CBIR using perception based texture and colour measures," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2, 2004, pp. 985–988.

[29] B. J. Woods, B. D. Clymer, T. Kurc, J. T. Heverhagen, R. Stevens, A. Orsdemir, O. Bulan, and M. V. Knopp, "Malignant-lesion segmentation using 4d co-occurrence texture analysis applied to dynamic contrast-enhanced magnetic resonance breast image data," *Journal of Magnetic Resonance Imaging*, vol. 25, no. 3, pp. 495–501, 2007.

[30] T. Mita, T. Kaneko, B. Stenger, and O. Hori, "Discriminative feature co-occurrence selection for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1257–1269, 2008.

[31] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Image Process.*, vol. 4, no. 11, pp. 1549–1560, Nov. 1995.

[32] B. Julész, E. Gilbert, and J. Victor, "Visual discrimination of textures with identical third-order statistics," *Biological Cybernetics*, vol. 31, no. 3, pp. 137–140, 1978.

[33] B. Julesz, "Textons, the elements of texture perception and their interactions," *Nature*, vol. 290, pp. 91–97, 1981.

[34] P. Ndjiki-Nya, D. Bull, and T. Wiegand, "Perception-oriented video coding based on texture analysis and synthesis," in *Proceedings of 16th IEEE International Conference on Image Processing (ICIP), 2009.* IEEE, 2009, pp. 2273–2276.

[35] M. N. Do and M. Vetterli, "Texture similarity measurement using kullback-leibler distance on wavelet subbands," in *Proceedings of International Conference on Image Processing*, vol. 3, Vancouver, BC, Canada, Sep 2000, pp. 730–733.

[36] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 837–842, Aug. 1996.

[37] V. Wouwer, G. Scheunders, P. Livens, and S. van Dyck, "Wavelet correlation signatures for color texture characterization," *Pattern Recognition*, vol. 32, pp. 443–451, 1999.

[38] M. Clark, A. Bovik, and W. Geisler, "Texture segmentation using a class of narrowband filters," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP '87.*, vol. 12, 1987, pp. 571–574.

[39] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, pp. 106–154, Jan. 1962.

[40] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America A*, vol. 2, pp. 1160–1169, 1985.

[41] J. P. Jones and L. A. Palmer, "An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex." *Journal of Neurophysiology*, vol. 58, no. 6, pp. 1233–1258, December 1987.

[42] I. Cox, M. Miller, Jeffrey, J. Fridrich, and T. Kalker, Eds., *Digital Watermarking and Steganography*, 2nd ed. Morgan Kaufmann, 2008.

[43] Z. Z. Kermani and M. Jamzad, "A robust steganography algorithm based on texture similarity using gabor filter," in *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology, 2005.*, 2005, pp. 578–582.

[44] J. Ilonen, J. K. Kamarainen, P. Paalanen, M. Hamouz, J. Kittler, and H. Kalviainen, "Image feature localization by multiple hypothesis testing of gabor features," *IEEE Transactions on Image Processing*, vol. 17, no. 3, pp. 311–325, 2008.

[45] J. Xie, Y. Jiang, and H.-T. Tsui, "Segmentation of kidney from ultrasound images based on texture and shape priors," *IEEE Transactions on Medical Imaging*, vol. 24, no. 1, pp. 45–57, 2005.

[46] S. Arivazhagan, L. Ganesan, and S. Priyal, "Texture classification using gabor wavelets based rotation invariant features," *Pattern Recognition Letters*, vol. 27, no. 16, pp. 1976–1982, December 2006.

[47] J. Mathiassen, A. Skavhaug, and K. Bø, "Texture similarity measure using Kullback-Leibler divergence between gamma distributions," in *Proceedings of the European Conference on Computer Vision,ECCV 2002*, vol. 2352, 2002, pp. 19–49.

[48] G. Yang and Y. Xiao, "A robust similarity measure method in cbir system," in *Proceedings of Congress on Image and Signal Processing, 2008. CISP '08.*, vol. 2, 2008, pp. 662–666.

[49] B. S. Manjunath, P. Salembier, and T. Sikora, *Texture Descriptors.* John Wiley and Sons, 2002, ch. 14, pp. 213–228.

[50] J. M. Martinez, "Mpeg-7 overview (version 10)," ISO/IEC JTC1/SC29/WG11, Palma de Mallorca, Spain, Doc. N6828, Oct 2004.

[51] K. L. Lee and L. H. Chen, "An efficient computation method for the texture browsing descriptor of MPEG-7," *Image and Vision Computing*, vol. 23, no. 5, pp. 479–489, 2005.

[52] Y. Lu, Q. Zhao, J. Kong, C. Tang, and Y. Li, "A two-stage region-based image retrieval approach using combined color and texture features," in *AI 2006: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science. Springer Berlin/Heidelberg, 2006, vol. 4304/2006, pp. 1010–1014.

[53] Z. Wang and E. P. Simoncelli, "Translation insensitive image similarity in complex wavelet domain," in *IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, vol. II, Philadelphia, PA, 2005, pp. 573–576.

[54] J. Portilla and E. P. Simoncelli, "Texture modeling and synthesis using joint statistics of complex wavelet coefficients," in *IEEE Workshop on Statistical and Computational Theories of Vision, Fort Collins*, 1999.

[55] A. C. Brooks, X. Zhao, and T. N. Pappas, "Structural similarity quality metrics in a coding context: Exploring the space of realistic distortions," *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1261–1273, Aug. 2008.

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

[57] H. S. Sawhney and J. L. Hafner, "Efficient color histogram indexing," in *Proceedings of IEEE International Conference Image Processing, 1994. ICIP-94.*, vol. 2, 1994, pp. 66–70 vol.2.

[58] Y. Deng, B. Manjunath, C. Kenney, M. Moore, and H. Shin, "An efficient color representation for image retrieval," *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 140–147, Jan 2001.

[59] W.-Y. Ma and B. Manjunath, "Edgeflow: a technique for boundary detection and image segmentation," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1375–1388, Aug 2000.

[60] Y. Deng, C. Kenney, M. Moore, and B. Manjunath, "Peer group filtering and perceptual color image quantization," *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, 1999. ISCAS '99.*, vol. 4, pp. 21–24, Jul 1999.

[61] A. Mojsilovic, J. Kovacevic, J. Hu, R. J. Safranek, and S. K. Ganapathy, "Matching and retrieval based on the vocabulary and grammar of color patterns," *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 38–54, 2000.

[62] G. Wyszecki and W. S. Stiles, *Color Science: concepts and methods, quantitative data and formulae*, 2nd ed. New York, NY: Addison-Wesley Publishing Co., 1982.

[63] J. M. Kasson and W. Plouffe, "An analysis of selected computer interchange color spaces," *ACM Transactions on Graphics*, vol. 11, no. 4, pp. 373–405, 1992.

[64] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 0, pp. 762–768, 1997.

[65] M. Birinci, S. Kiranyaz, and M. Gabbouj, "Image color content description utilizing perceptual color correlogram," *International Workshop on Content-Based Multimedia Indexing, 2008. CBMI 2008.*, pp. 200–207, June 2008.

[66] Y. Rubner, J. Puzicha, C. Tomasi, and J. Buhmann, "Empirical evaluation of dissimilarity measures for color and texture," *Computer Vision and Image Understanding*, vol. 84, pp. 25–43, Oct 2001.

[67] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, pp. 99–121, Nov 2000.

[68] D. K. Park, Y. S. Jeon, C. S. Won, S.-J. Park, and S.-J. Yoo, "A composite histogram for image retrieval," *IEEE International Conference on Multimedia and Expo, 2000. ICME 2000.*, vol. 1, pp. 355–358, 2000.

[69] A. Guerin-Dugue, S. Ayache, and C. Berrut, "Image retrieval: a first step for a human centered approach," *Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia.*, vol. 1, pp. 21–25, Dec 2003.

[70] I. Markov, N. Vassilieva, and A. Yaremchuk, "Image retrieval: Optimal weights for color and texture features combining based on query object," *Proceedings of RCDL*, pp. 195–200, 2007.

[71] I. Markov and N. Vassilieva, "Image retrieval: Color and texture combining based on query-image," *Image and Signal Processing*, pp. 430–438, 2008.

[72] J. A. Shaw and E. A. Fox, "Combination of multiple searches," *Proceedings of Third Text REtrieval Conference (TREC-3)*, pp. 105–109, 1995.

[73] X. Zhao, M. G. Reyes, T. N. Pappas, and D. L. Neuhoff, "Structural texture similarity metrics for retrieval applications," in *Proc. Int. Conf. Image Processing (ICIP-08)*, San Diego, CA, Oct. 2008, pp. 1196–1199.

[74] A. Reibman and D. Poole, "Characterizing packet-loss impairments in compressed video," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 5. IEEE, 2007, pp. 77–80.

[75] E. Simoncelli, "Statistical models for images: compression, restoration and synthesis," *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers, 1997.*, vol. 1, pp. 673–678, Nov 1997.

[76] D. Hubel and T. Wiesel, "Ferrier lecture: Functional architecture of macaque monkey visual cortex," *Proceedings of the Royal Society of London. Series B, Biological Sciences*, vol. 198, no. 1130, pp. 1–59, 1977.

[77] K. H. Foster, J. P. Gaska, S. Marcelja, and D. A. Pollen, "Phase relationships between adjacent simple cells in the feline visual cortex," *Journal of Physiology (London)*, vol. 345, p. 22P, 1983.

[78] T. Aach, A. Kaup, and R. Mester, "On texture analysis: Local energy transforms versus quadrature filters," *Signal processing*, vol. 45, no. 2, pp. 173–181, 1995.

[79] M. Sampat, Z. Wang, S. Gupta, A. Bovik, and M. Markey, "Complex wavelet structural similarity: A new image similarity index," *Image Processing, IEEE Transactions on*, vol. 18, no. 11, pp. 2385–2401, 2009.

[80] P. C. Mahalanobis, "On the generalized distance in statistics," in *Proceedings of the National Institute of Science,India*, vol. 2, 1936, pp. 49–55.

[81] T. N. Pappas, "An adaptive clustering algorithm for image segmentation," *IEEE Trans. Signal Process.*, vol. SP-40, no. 4, pp. 901–914, Apr. 1992.

[82] A. Mojsilovic, J. Kovacevic, D. Kall, R. Safranek, and S. Kicha Ganapathy, "The vocabulary and grammar of color patterns," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 417–431, 2000.

[83] C. Poynton, *A technical introduction to digital video*. Wiley, 1996.

[84] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statictics of complex wavelet coefficients," *Int. J. Computer Vision*, vol. 40, no. 1, pp. 49–71, Oct. 2000.

[85] "Corbis stock photography," http://www.fotosearch.com/corbis/. [Online]. Available: www.corbis.com

[86] C. T. Meadow, B. R. Boyce, D. H. Kraft, and C. Barry, *Text information retrieval systems*. Emerald Group Publishing, 2007.

[87] Z. He, X. You, and Y. Yuan, "Texture image retrieval based on non-tensor product wavelet filter banks," *Signal Processing*, vol. 89, no. 8, pp. 1501–1510, 2009.

[88] N. A. Macmillan and C. D. Creelman, *Detection Theory: A User's Guide*, 2nd ed. Lawrence Elbaum Associates, 2005.

[89] W. S. Torgerson, *Theory and methods of scaling*. New York, NY: Wiley, 1958.

[90] J. B. Kruskal and M. Wish, *Multidimensional scaling*. Beverly Hills, CA: Sage Publications, 1977.

[91] L. Thurstone, "A law of comparative judgment." *Psychological review*, vol. 34, no. 4, p. 273, 1927.

[92] H. Gulliksen and L. Tucker, "A general procedure for obtaining paired comparisons from multiple rank orders," *Psychometrika*, vol. 26, no. 2, pp. 173–183, 1961.

[93] B. Rogowitz, T. Frese, J. Smith, C. Bouman, and E. Kalin, "Perceptual image similarity experiments," *Human Vision and Electronic Imaging III*, vol. 3299, no. 1, pp. 576–590, 1998.

[94] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395–416, 2007.

[95] E. M. Voorhees, "The trec-8 question answering track report," in *In Proceedings of TREC-8*, 1999, pp. 77–82.

[96] ——, "Variations in relevance judgments and the measurement of retrieval effectiveness," *Information Processing & Management*, vol. 36, no. 5, pp. 697–716, September 2000.

[97] D. Krus and P. Krus, "Normal scaling of the unidimensional dominance matrices: the domain referenced model," *Educational and Psychological Measurement*, vol. 37, no. 1, p. 189, 1977.

[98] J. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

[99] M. Kendall and B. Smith, "On the method of paired comparisons," *Biometrika*, vol. 31, no. 3/4, pp. 324–345, 1940.