NORTHWESTERN UNIVERSITY

Structure Similarity Metrics for Texture Analysis and Retrieval

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

MASTER OF SCIENCE

Field of Electrical Engineering

By

Jana Žujović

EVANSTON, ILLINOIS

December 2008

To my parents

ABSTRACT

The presence of computers in almost every sphere of today's human life creates the need for bridging the gap between human and machine perception of the world. It is a difficult task, given that the complexity and structure of human mind and the architecture of computers are not in accordance. Since humans are mainly visual creatures, developing ways to compare the similarity of images which would mimic human perception is an important issue. It has been shown that the traditional metrics that are intrinsic to the computer representation of images, such as point-by-point comparisons, do not perform well when compared to human judgments of similarity. Therefore, there have been several attempts to incorporate structural information in image comparisons, such as Structural Similarity Metrics (SSIM). More recently, Complex-Wavelet SSIM (CWSSIM) and Structural Texture SIM (ST-SIM) have been explored in this context. The drawbacks of these approaches are that they only use the gray-scale aspect of images, and do not utilize the (usually strong) correlations between information contained in different subband decompositions of an image. In this work, an attempt is made to overcome these drawbacks and juxtapose the results to human similarity judgments for the purpose of evaluation and comparison. The information about the color composition of images is incorporated into the similarity metrics. Also, measures that represent inter-subband correlations are added to provide results closer to human evaluations.

Acknowledgments

I would like to thank my advisor and chair of my M.S. committee Dr. Thrasyvoulos Pappas for the guidance and inspiration during the work on this thesis. I would also like to thank Dr. Aggelos Katsaggelos, Dr. David Neuhoff, Dr. Jack Tumblin and Dr. Ying Wu for serving on my committee and spending their valuable time. I would also like to thank my family and friends for their help and support, in particular J. Stephen Downie whose help on information retrieval problems was invaluable and Andreas Ehmann whose knowledge on signal processing provided helpful tips and good sanity-check points.

Table of Contents

List of Figures

CHAPTER 1

Introduction

Computers and other electronic gadgets have become ubiquitous in the lives of people. From already old-fashioned desktop computers, new speedy laptops, the mini-yet-powerful-computers present in our iPhones and MP3 players, to ”smart” ovens, the need for meaningful communication between people and devices is on the rise. Naturally, the desire for computers to perceive better the needs of humans and to have higher mutual understanding arises from this, sometimes overwhelming, presence of machines in almost every aspect of everyday life.

Humans are mainly visual creatures. Vision is the sense humans mostly rely upon for leading their lives because it provides immediate information about threats and opportunities in their surroundings. Therefore, it is important to find a way for machines to perceive the world as similarly to humans as possible for making human-machine interaction easier and more intuitive. The processing of information obtained from the sense of vision activates a large part of the brain cortex [**1**], leading us to the conclusion that it is a rather complicated mechanism. As of now, there hasn't been a system built by humans that can successfully reproduce the functioning of the human brain and perform the complex data processing innate to living beings. Thus, an effort is being made attempting to use the existing capabilities of computers and the operations they can perform well in order to simulate human perception of the world.

Today's computers perform various tasks and one of their main uses is visual communication among humans and between humans and machines. In particular, when humans are the end-users for applications, it is very important for computers to understand how people see and to act accordingly. For most of these applications providing human input on performance is virtually impossible. For example, having people manually segmenting medical images, or assigning similarity scores between millions of images for Content-Based Image Retrieval (CBIR) would be extremely time consuming, expensive and inefficient. Whether it is image sharing, video streaming or a simple web-site containing images compressed with lossy techniques, it is crucial for the machine to have a built-in ability to assess the content and/or quality of images and to understand what the user, i.e. human, wants. The goal is to develop efficient and accurate techniques that would allow computers to, independently of human input, perform this task at a satisfying level.

One of the obvious ways to analyze an image is to decompose it and understand its parts. This task of parsing am image is performed by segmentation algorithms. There are numerous segmentation techniques in use, for example the popular Mean-Shift Algorithm [**2**] that relies on color only, or the Adaptive Clustering Algorithm (ACA) [**3**] that utilizes both color and texture information. Different applications require different information, thus the segmentation and information extraction techniques greatly depend on the field of application and the type of images to be processed.

Various applications include CBIR systems, systems for processing medical images, feature extraction/detection algorithms and so on. In all cases, there are both color and texture aspects that need to be addressed for the systems to have satisfactory performance. However, given the different targets, the systems may have significant implementation differences.

There are various techniques used for determining image similarities. They exist both in the spatial domain, and in the transform domain (wavelet transforms, DCT, Gabor filters, Fourier transform etc.). However, since the goal is to acquire a system that would perform similarly to how people see, the best techniques are the ones that utilize the properties of human vision. The methods that have proven to relate fairly well to characteristics of human vision are Structure Similarity Metrics (SSIM) [**4**] and its variations for evaluating texture similarity, and utilizing dominant colors for color representation and color comparisons [**5**].

In this work, an attempt is made to fuse the two techniques of extracting texture and color information, consistent with human perception. Assessing the similarity of images is based again on techniques that have also been shown to be in accordance with human visual system (HVS). The contribution of this work is in improving the texture similarity metrics by adding novel measurements, and combining the texture and color information in a way that agrees most with human judgments and evaluations of visual similarity.

The thesis is organized as follows: the necessary background is presented in Chapter 2, the proposed improvements are explained in Chapter 3, description of the experiments and results are given in Chapter 4, and conclusions are drawn in Chapter 5.

CHAPTER 2

Background

With the expansion of technology, we are facing two trends: growing amount of data available to people, and growing need for diverse applications. Image analysis, segmentation, quality and similarity assessment for the applications where humans are the end-users would ideally be performed by - humans. However, this is unfeasible, thus we need techniques that would be able to automatically and with little or no human input perform these operations.

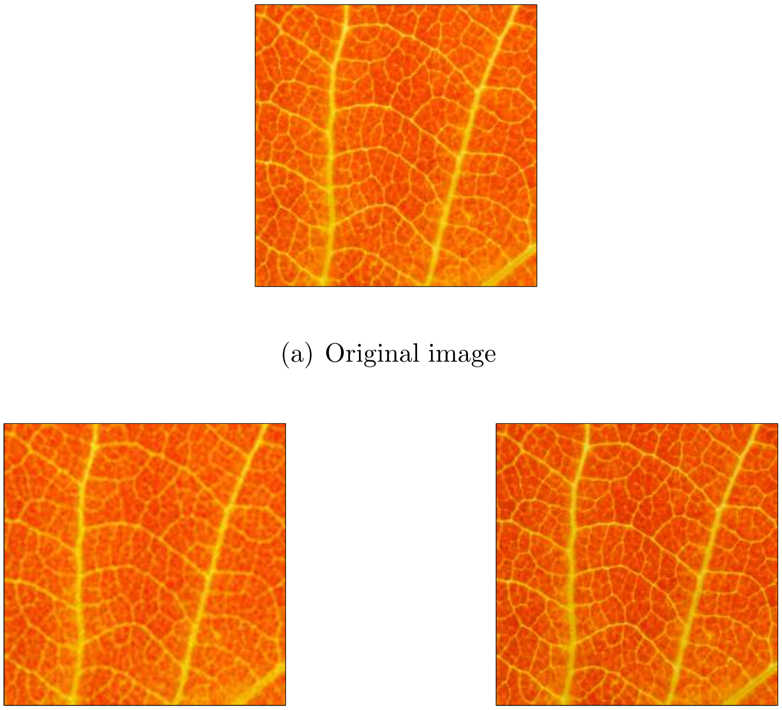
For this task, we need to define what type of similarity metrics to use, and also which type of information we will evaluate our metrics on. In the following sections, a brief description of different types of metrics and features will be given.

2.1. Texture Description

We will note the most commonly used texture descriptors, and also give a more detailed description of the method chosen for the purposes of this work.

2.1.1. Non-Perceptual Metrics

The simplest and most obvious techniques for image comparisons are the ones that comply with computer architecture characteristics, that is, point-by-point comparisons. Those metrics (like MSE and PSNR) are not, however, in accordance with human perception of images [**6**],[**7**]. This can be seen in Figure 2.1.1. Image 2.1(a) is the original image, image 2.1(b) is the original image compressed with DCT, and image 2.1(c) is the original image with subtle lighting changes. In both cases, PSNR is the same, 28.7422dB, but it is obvious that image 2.1(c) looks much closer to the original than image 2.1(b).



(a) Original image

(b) DCT compressed image

(c) Image with lighting changes

Figure 2.1. Illustration of inadequacy of PSNR metrics

2.1.2. Low-Level Human Perception Metrics

The non-perceptual metrics obviously fail at determining image similarity and quality. Somewhat better are low-level human perception models (e.g. Perceptual PSNR), that incorporate low-level human vision characteristics. These models try to penalize errors according to how visible they are. In Eckert and Bradley's paper [**8**], the effect of incorporating contrast sensitivity function (CSF), luminance masking and contrast masking is juxtaposed to the traditional MSE metrics. The images are analyzed by multiscale frequency decomposition, and it is shown that additional functionalities that account for HVS characteristics improve the results of point-by-point metrics. However, this type of analysis cannot accommodate subtle structural changes, such as zooming, contrast and intensity changes. Thus, we need metrics that would incorporate high-level HVS characteristics.

2.1.3. Perceptual Metrics

Today, there are numerous methods for describing textures and determining their similarity that utilize the properties of HVS. The most commonly used feature extraction algorithms are relying on processing images with Gabor filters. One of the pioneering works was done by Clark et al. [**9**], and soon the Gabor filters became the most popular tool. 2D Gabor filters are shown to be analogous to simple receptive fields (receptive fields of simple cells) in the visual cortex of higher vertebrates [**10**],[**11**], which means they describe well the first stages of image processing in humans. Gabor filters in effect use the actual characteristics of HVS, thus they are adequate for use in perceptual meaningfulness sense. The frequency domain covered by Gabor filters in 3 scales and 6 orientations is given in Figure 2.2.

Gabor features have found various applications, as in CBIR [**12**], steganography [**13**], object detection [**14**], medical image segmentation [**15**], texture similarity and classification [**16**], [**17**] and so on. The main idea is to filter the image with Gabor wavelets, and take the mean and standard deviation of each subband as a feature. However, images can have very similar energies in subbands, and yet not be very similar.

Also very popular are co-occurrence matrices, that have found application in CBIR systems [**18**], medical image analysis [**19**], as well as object detection applications [**20**]. Some recent works even combine the two popular features, like [**21**] for CBIR applications. In essence, co-occurrence matrices deploy relationships between adjacent pixels, like Haralicks'

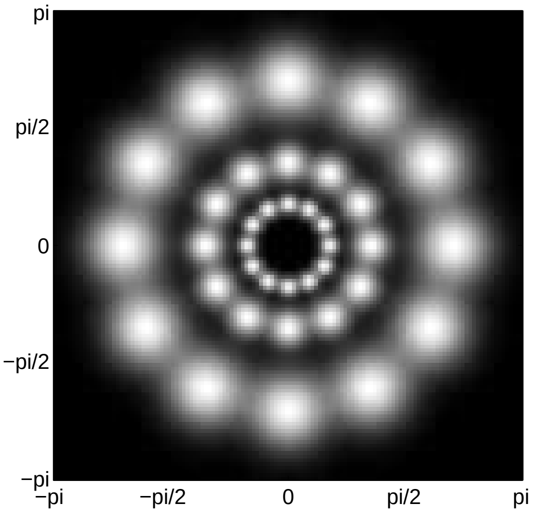


Figure 2.2. Gabor filter bank

features [**22**], calculating the differences in luminance values within a small neighborhood, usually 2x2. This approach may easily fail when we have image distortions like salt-and-pepper noise, zoom etc.

Texture description and analysis has also been employed in popular standards like MPEG7. These techniques are developed in order to make video coding and retrieval more efficient. The three variations of texture descriptors used in this standard - the Homogenous Texture Descriptor, Texture Browsing Descriptor and Edge Histogram Descriptor - are described in detail in the book [**23**]. Homogenous Texture Descriptor is again a vector of means and variances of Gabor filtered image, which addresses the aspect how HVS works. Texture Browsing Descriptor characterizes the directionality and coarseness of the texture image, which is related to high-level human perception of images. Edge Histogram Descriptor partitions the image into 16 blocks, applies edge detection algorithms and computes local edge histograms for different edge directions, and this descriptor is said to work well in image similarity assessment, since the edges are descriptive clues for texture perception. The variations of techniques used in MPEG7 exist in other applications like CBIR [**24**], where different edge detectors are used.

Another approach that focuses on high-level properties of HVS without actually modeling it (like e.g. Gabor filters) is the Structure Similarity Metric [**4**]. Although it is not relying on explicit functions that describe HVS characteristics, like Perceptual PSNR does, it still adapts to lighting changes and has masking capabilities. The idea is to analyze images in sliding windows and compare their luminance, contrast and ”structure” in corresponding windows. Luminance is characterized by the mean of intensities within the window (eq. 2.1), contrast is characterized by the standard deviation (eq. 2.2) and structure is characterized by the cross-correlation between two patches (eq. 2.3):

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{2.1}$$

$$\sigma_x = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2} \tag{2.2}$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) \tag{2.3}$$

The luminance term is:

$$L(x,y) = \frac{2\mu_x\mu_y + c_0}{\mu_x^2 + \mu_y^2 + c_0}, \tag{2.4}$$

the contrast term is:

$$C(x,y) = \frac{2\sigma_x\sigma_y + c_1}{\sigma_x^2 + \sigma_y^2 + c_1}, \tag{2.5}$$

and the structure term is:

$$S(x,y) = \frac{\sigma_{xy} + c_2}{\sigma_x\sigma_y + c_2}, \tag{2.6}$$

where c_0, c_1 and c_2 are small constants (to prevent divisions of type 0/0).

For each sliding window, we get a similarity value computed as:

$$SSIM(x,y) = L(x,y)^\alpha C(x,y)^\beta S(x,y)^\gamma. \tag{2.7}$$

Usually, the parameters are set to be $\alpha = \beta = \gamma = 1$ and $c_2 = c_1/2$ to get:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_0)(2\sigma_{xy} + c_1)}{(\mu_x^2 + \mu_y^2 + c_0)(\sigma_x^2 + \sigma_y^2 + c_1)}. \tag{2.8}$$

The final SSIM metric is computed as the spatial average of the $SSIM(x,y)$ computed for each sliding window. The size of the window affects the metric in the sense that as it becomes smaller, it becomes closer to the point-by-point comparisons, and as it grows, it becomes more of a structural metric. However, it is highly sensitive to translation, scaling and rotation of images, as shown in [**25**]. This can be remedied to some extent by implementing the SSIM metric in complex wavelet domain.

An improvement to SSIM is Complex Wavelet Structure Similarity Metric (CW-SSIM). To account for small spatial translation, we can utilize an overcomplete wavelet transform such as the steerable pyramid [**26**]. The locations of subbands in the frequency plane are given in Figure 2.3, for three scales and four orientations. CW-SSIM is invariant to luminance change, contrast change and spatial translation, as proven by Wang and Simoncelli [**25**]. 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 consistent phase shift of all coefficients does not change the structure of the local image feature [**25**].

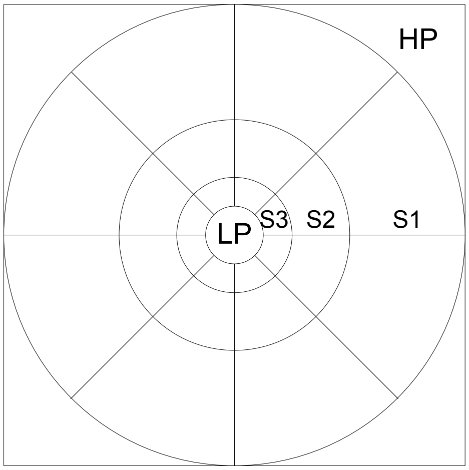


Figure 2.3. Steerable filter bank; the axes ranges are $[-\pi, \pi]$ in both vertical and horizontal direction

The images to be compared are decomposed into wavelet subbands, yielding having two sets of complex coefficients for each subband. For each subband, the CW-SSIM term is computed as (K is a small constant):

$$\tilde{S}(c_x, c_y) = \frac{2|\sum_{i=1}^{N} c_{x,i}c_{y,i}^*| + K}{\sum_{i=1}^{N} |c_{x,i}|^2 + \sum_{i=1}^{N} |c_{y,i}|^2 + K} \tag{2.9}$$

CW-SSIM is calculated in each subband by averaging over all spatial locations, and the final CW-SSIM is calculated as the mean of the subband CW-SSIM coefficients (or a subset of subband CW-SSIM coefficients). In effect, CW-SSIM has all the terms like the regular SSIM metric, only without the luminance term - since the means of subband decompositions are zero (due to the band-pass filtering), the L-term would be 1 in all cases (apart from the baseband images).

However, as shown by Brooks and Pappas [**27**], CW-SSIM is sensitive to low-frequency spatial image distortions, in particular to luminance changes. Thus, they proposed a weighted metric (WCWSSIM) that would compensate for this, and they show that it agrees more with human perception. The idea is to weight different subbands according to the Contrast Sensitivity Function (CSF) of HVS and the frequency responses of wavelet filters.

The advantage of SSIM, and, extended to complex wavelet domain, CW-SSIM, is that it is not a point-by-point comparison, but more of a structural metric. On the other hand, the structure term, from which it got its name, is indeed a point-by-point comparison [**28**]. Therefore, Zhao et al. propose using broader subband statistics to account for texture characteristics. Even though the subbands are computed with different orientations, the argument is that we can exploit directional information within each subband in order to improve the performance of the metric. By removing the structure term and adding the first order autocovariance in the horizontal and vertical directions, we get better statistics computed in each subband.

The autocovariance term in horizontal direction is defined as:

$$\rho_x(0,1) = \frac{E\{(x_{i,j} - \mu_x)(x_{i,j+1} - \mu_x)\}}{\sigma_x^2} \tag{2.10}$$

and analogous, vertical direction:

$$\rho_x(1,0) = \frac{E\{(x_{i,j} - \mu_x)(x_{i+1,j} - \mu_x)\}}{\sigma_x^2} \tag{2.11}$$

The values for autocorrelation are bounded and lay in the interval [-1,1], and they are compared for two images as:

$$C_{0,1}(x,y) = 1 - 0.5|\rho_x(0,1) - \rho_y(0,1)| \tag{2.12}$$

$$C_{1,0}(x,y) = 1 - 0.5|\rho_x(1,0) - \rho_y(1,0)| \tag{2.13}$$

For each sliding window in each subband, the previously defined L (Eq. 2.4) and C (Eq. 2.5) terms are combined with the new ones into the Structural Texture Similarity Metric (STSIM) as:

$$STSIM(x,y) = L^{\frac{1}{4}}(x,y)C^{\frac{1}{4}}(x,y)C_{0,1}^{\frac{1}{4}}(x,y)C_{1,0}^{\frac{1}{4}}(x,y) \tag{2.14}$$

Since each point in the subband image has its associated STSIM coefficient, the question is how to combine them for all subbands. One approach is the so-called ”additive” where the total resulting STSIM is calculated by first taking the mean over spatial locations in each subband, and then taking the mean across frequencies. The other is ”multiplicative” approach, where corresponding STSIM values for each spatial location get multiplied across the subbands, and then the final metric is the spatial mean of these multiplied coefficients. This is depicted in Figure 2.4: if we have N different subbands, at each location the corresponding STSIM values are multiplied, then the N-th root is taken from each point (so that the numbers do not become too small) and in the end they are spatially averaged.

The STSIM has shown to perform better, i.e. closer to human judgements of texture similarity, than SSIM and CW-SSIM.

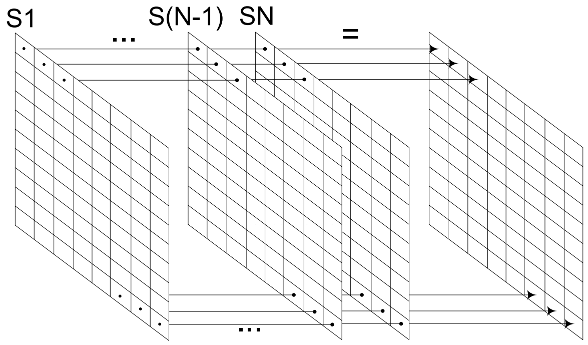


Figure 2.4. Multiplicative computation of STSIM

To conclude the texture descriptors review, SSIM and its variations have shown to incorporate most of HSV properties in an implicit way, and we will see in Chapter 3 how they can be further explored.

2.2. Color Description

Color is perhaps the most expressive of all the visual features and has been extensively studied in the image retrieval research during the last decade [**29**]. For describing the color composition of images, the simplest approach has been to use color histograms with static color space [**30**], accompanied by simple L_p, histogram intersection metrics [**30**] or more sophisticated color quadratic distance [**31**]. However, as shown in [**29**] and [**32**], human vision system is not designed to distinguish well between similar colors, and also the studies show that humans cannot simultaneously perceive a large number of colors present in one image. People, in fact, see only few prominent colors from the image, and those are so-called "dominant colors". Thus, the color descriptors and comparing methods have moved away from direct histogram acquisitions and comparisons to more sophisticated techniques that should account better for the HVS properties.

One approach to describing colors in an image has found its application in the popular MPEG-7 standard. The color descriptors, developed by Manjunath et al. [**29**], are classified in 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 [**33**]. First, image is segmented using edgeflow algorithm [**34**], then colors are clustered within each segment by using modified generalized Lloyd algorithm proposed in [**35**]. The clustering algorithm consists of pre-processing of the images to remove noise and to 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 according percentages of pixels belonging to the clusters. The distance between two descriptors is similar to the quadratic color distance [**31**].

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 (8x8) and the average color for each block is computed. This results in an 8x8 matrix of local means. Then, 8x8 DCT is applied, and 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, histogram representation lack discriminatory power in retrieval of large image databases and do not match human perception very well [**32**]. 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 [**5**]. This is the basis for extraction algorithm of perceptually important colors, as developed by Mojsilovic et al. [**32**].

The chosen colorspace for use is L*a*b*. CIELAB (or L*a*b*) and CIELUV (L*u*v*) colorspaces exhibit perceptual uniformity, meaning that the Euclidian distance separating two similar colors is proportional to their visual difference [**36**]; however, the Euclidian distances in these colorspaces are not linearly proportional to the visual judgement when the colors are dissimilar. This is suitable for image retrieval applications, when we're interested in the proximity of the two color compositions, but it may not be helpful in assessing the absolute similarity/dissimilarity.

The colors in the image are firstly quantized into m colors according to the 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 NxN 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 DxD 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 [**37**]. It differs in the sense that 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 newly proposed method by Birinci et al. [**38**] combines the dominant color approach and correlogram calculations. They named their approach as 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 also 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. [**37**] utilize simple L_1 distance measure. Manjunath et al. [**29**] are using quadratic color histogram to compute distances between dominant colors. However, as shown in [**39**], the metric that has superior classification and retrieval results with compact representation is the Earth Mover's Distance.

The Earth Mover's Distance [**40**] is based on the minimal cost that must be paid to transform one distribution into the other. Informally speaking, the Earth Mover's Distance (EMD) measures how much work needs to be applied to move earth distributed in piles x so that it turns into the piles y.

This can be formalized as linear optimization problem: denote two images as X and Y, and their representative color compositions C_X, C_Y, as $C_X = \{(c_{x1}, p_{x1}), ...(c_{xm}, p_{xm})\}$ and $C_y = \{(c_{y1}, p_{y1}), ...(c_{yn}, p_{yn})\}$, where c-elements denote the colors and p-elements their respective percentages within the image. 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* mapping between colors (c_{xi}, c_{yj}) (how much of color c_{xi} gets "transported" to color c_{yj}), the problem can be stated as:

$$\min_{\mathbf{F}} \frac{\sum_{i,j} d_{i,j} f_{i,j}}{\sum_{i,j} f_{i,j}} \tag{2.15}$$

subject to:

$$f_{i,j} \geq 0 \qquad 1 \leq i \leq m,\ 1 \leq j \leq n \tag{2.16}$$

$$\sum_{j=1}^{n} f_{i,j} \leq p_{xi} \qquad 1 \leq i \leq m \tag{2.17}$$

$$\sum_{i=1}^{m} f_{i,j} \leq p_{yj} \qquad 1 \leq j \leq n \tag{2.18}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} f_{i,j} = \min\left(\sum_{i=1}^{m} p_{xi}, \sum_{j=1}^{n} p_{yj}\right). \tag{2.19}$$

These conditions can be explained if we look at the informal problem of earth transportation between centers C_X and C_Y. Let's 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 by Eq. 2.16 means we can't have "negative transportation", i.e. transportation between c_{xi} and c_{yj} can only go $c_{xi} \rightarrow c_{yj}$. Condition in Eq. 2.17 means that we can't take out of c_{xi} more than there's inside; condition in Eq. 2.18 means that we can't put in c_{yj} more than it can receive; the last condition (Eq. 2.19) 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 2.5. Color composition of the first image (black bordered circles) is given as:

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

and of the second image (gray bordered circles):

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

The L_2 distances between colors (normalized to have maximum distance 1) are given in the following table:

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

The computed matching for EMD is given in the image; on the arrows between the circles are noted the amounts transported and distances. 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 $EMD(C_X, C_Y) = 0.1494$.

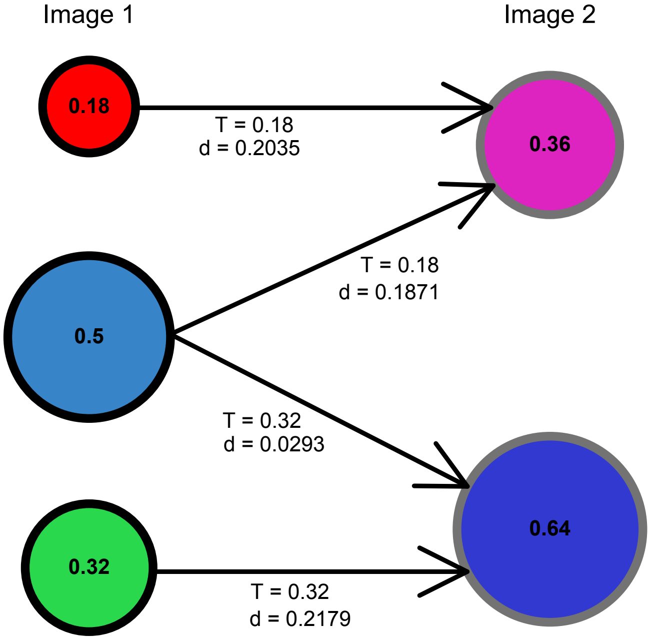


Figure 2.5. EMD example of color matching

An approach that follows the same philosophy as EMD is Optimal Color Composition Distance (OCCD) developed by Mojsilovic et al. [**32**]. 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. $n \cdot p = 100$; now, each image is represented with n units, each unit is labeled with the color value; percentages are not needed anymore since the number of units with the same color is proportional to its percentage. The problem is now transformed into minimum cost graph matching problem - the bijective matching between two sets of n units. Let the units from one image be denoted as $C_X = \{c_{x1}, ..., c_{xn}\}$ and $C_Y = \{c_{y1}, ..., c_{yn}\}$ and let m_{XY} be a bijective function that maps set C_X onto set C_Y, $\{m_{XY} : C_X \rightarrow C_Y\}$; 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_{XY}:

$$\min_{m_{XY}} \sum_{i=1}^{n} d(c_{xi}, m_{XY}(c_{xi})) \tag{2.20}$$

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_X: 10 units of c_{x1}, 6 units of c_{x2}, 4 units of c_{x3}
- For C_Y: 13 units of c_{y1}, 7 units of c_{y2}

Now, OCCD tries to find the best match between the units s.t. the sum of distances is minimized. The problem is depicted in Figure 2.6.

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:

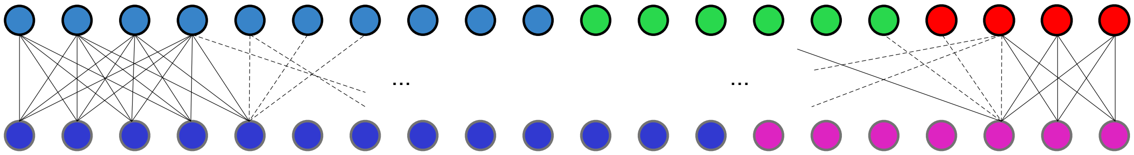


Figure 2.6. OCCD problem statement

$OCCD(C_X, C_Y) = 0.1444$ (compared to $EMD(C_X, C_Y) = 0.1494$). On the other hand, if we use 1% units, we will yield 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 the weighted graph matching that can be solved by deterministic algorithms, unlike linear-programming based EMD calculation.

The final result for units matching is given in Figure 2.7.

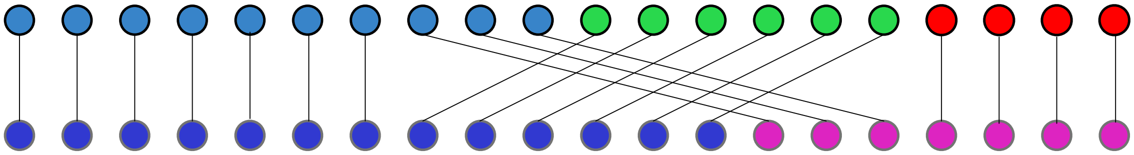


Figure 2.7. OCCD problem solution

This analysis leads to the assumption that best results in color matching are obtained if we utilize the dominant color descriptors, followed by the metric that incorporates in a sophisticated way the properties of HSV, like EMD and OCCD do.

2.3. Combining Texture and Color Information

In the previous sections, the emphasis was on extracting texture and color information from images, and developing techniques for comparing them. Since we are interested in both features, we need also to review possible ways of combining them.

One of the possible composite descriptors is presented in [**41**]. Park et al. define a simple 14-point vector (6 basic colors, 5 shades of gray, and 3 edge descriptors) as image feature, and the distances are computed using histogram intersection measure. This approach, despite being very memory and CPU time efficient, doesn't apply any HSV properties and as such may be highly inadequate for comparing two images on structural basis.

There has been more research in data fusion algorithms in text retrieval than in image retrieval and comparison. Algorithms that combine multiple searches like CombMNZ [**42**] are popular for their good performance, and they may be utilized for CBIR systems. On the other hand, they cannot be applied in image similarity assessment context since the goal is not to rank images but to determine similarity pair-by-pair, regardless of the remaining images in the database.

A common approach is to linearly combine color and texture descriptors, i.e. their distances, because of the simplicity of the algorithm. In some recent works, Lu et al. [**24**] propose using the quadratic distance, which would in effect be linear combination of texture and color distances. However, in most of the cases, the weights are tuned for each image or set of images, as in the works of Guerin-Dugue et al. [**43**] or Markov et al. [**44**]. As Markov and al. have shown [**45**], there are optimal weights for combining texture and color information, but those optimal weights are unique for each image or group of images and they need to be "learned", just like in [**43**]. This is not an acceptable solution, if the application in matter is similarity measurement, since we cannot assume we will have all possible images at our disposal for training the system.

A more elaborate technique is presented in [**5**] by Mojsilovic et al. In their algorithm, image similarity is determined according to the grammar rules, which are developed based on human judgements. It is shown that people are strongly influenced by the pattern (texture) similarity; if the pattern similarity is not very emphasized, next step is determining how similar the dominant colors and directionalities of patterns are; third level is similarity of directionality of textures, regardless of the color, and the last step is color similarity.

Thus, the methods for efficient combining of color and texture descriptors are to be more explored in the future. Some possible solutions will be presented in Chapter 3.

CHAPTER 3

Proposed Method

In the previous chapter, numerous possible solutions to the problem of image similarity have been examined and described. In this work, the accent is on textures, and determining texture similarity. This is an important step in both developing general image similarity/quality assessment algorithms, as well as developing intuition about human perception of images.

Since the proposed STSIM metric [**28**] has outperformed the basic SSIM and its extensions like CW-SSIM and WCW-SSIM, it will be used as the basis for improvements in texture similarity measurements.

The proposed method for determining image similarity attempts to address all the aspects of human perception, and also to fuse the best algorithms for determining texture and color composition similarity of images. Its major contribution is in incorporating new and effective additions to the texture similarity metric: exploiting the overcompleteness of wavelet subband decompositions and calculating the correlations between subbands.

The method consists of three major steps:

(1) Determining texture similarity
(2) Determining color similarity
(3) Fusing the results.

They will each be discussed in detail in the following sections.

3.1. New Texture Similarity Metric

For the simplicity, the following notation will be utilized throughout this section:

- size of images is NxN
- size of neighborhood is WxW, W being an odd number
- total number of scales is N_S, while current scale number is $n_S \in \{1 \dots N_S\}$
- total number of orientations is N_O, while current orientation number is $n_O \in \{1 \dots N_O\}$
- a filtered image, at scale n_S and orientation n_O, is denoted as I^{n_S,n_O}

For developing texture similarity metrics, the choice was to use Steerable Filter Decomposition on grayscale part of the images. They are, like Gabor filters, inspired by biological visual processing, and also have nice properties, such as translation-invariance and rotation-invariance, as claimed by Portilla and Simoncelli [**46**]. In this work, the 3-scales 4-orientations decomposition is used, and the steerable pyramids' Fourier spectra are given in Figure 3.1. Note that, apart from 12 steerable subband filters, we also have a low-pass and a high-pass subband.

The new texture similarity metric uses all the terms previously described by Equations 2.4, 2.5, 2.12 and 2.13 in Section 2.1. On top of that, we propose using the new terms that would compare similarities of cross-correlation properties between subbands. This is motivated by the work of Portilla and Simoncelli [**46**], where focus is on texture synthesis. The reasoning is, if a statistic is good for texture generation, it would also be suitable as a feature for texture comparisons.

Justification of using the cross-correlations between coefficients in different subbands lies in the fact that the image representation by steerable filter decomposition is overcomplete

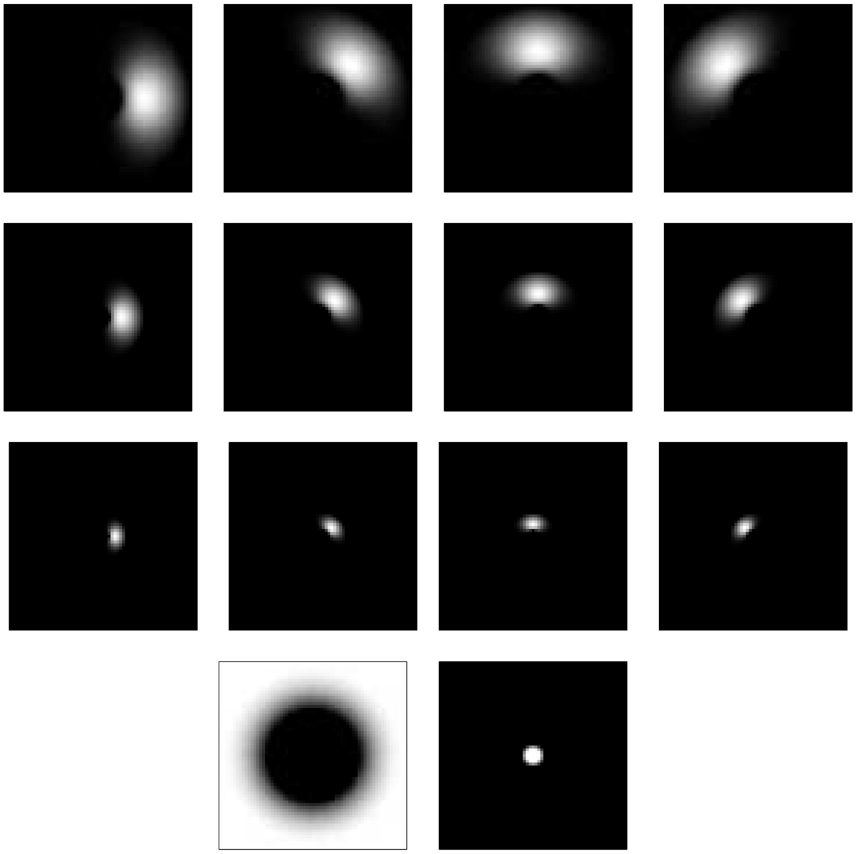


Figure 3.1. Frequency responses of Steerable Filters in 3 scales and 4 orientations; the axes ranges are $[-\pi, \pi]$ in both vertical and horizontal direction

and thus the coefficients will be correlated. This can easily be seen in Figure 3.1, where it is clear that the subbands overlap. Another reason to use these statistics is that covariances of subband coefficients can arise from spectral peaks (i.e., periodicity) in a texture [**46**], which also can be a good comparison point.

It should be noted that the luminance, contrast and autocorrelation terms are calculated on the *raw* subband coefficients. Since the subband decomposition (apart from the low-pass filtering) does not include the origin of the frequency plane, the filtered images will be *zero-mean* over the *whole* image; however, within small windows of size WxW, e.g. 7x7, this does not have to be true; thus, we need to compute the L-term for each sliding window, despite the band-pass filtering. Variances describe the spectral power within the sliding window, for given subband decomposition, and may be good descriptors for natural images. The autocovariances give us more directionality information.

More importantly, the cross-correlation statistics are computed on *magnitudes* of subband coefficients. The raw coefficients may be uncorrelated, since the phase information can lead out to cancellations. As shown by Simoncelli [**47**], the wavelet coefficients magnitudes are *not* statistically independent, and large magnitudes in 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 [**46**].

We propose computing the correlations between subbands at adjacent scales, for a given orientation, and between all orientations, for a given scale. For the $N_S = 3$, $N_O = 4$ example, we would have 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 terms. We will discuss in Chapter 4 the effects of utilizing all terms, or a subset of the 26 possible coefficients.

To compare two (grayscale) images I_X and I_Y, the proposed algorithm can be summarized in the following steps:

(1) filter the images with steerable pyramid bank of filters with N_S scales and N_O orientations; this gives us two sets of $N_S \cdot N_O + 2$ images;

(2) for each pair of corresponding subbands, $I_X^{n_S,n_O}$ and $I_Y^{n_S,n_O}$, for each sliding window of size WxW centered at (i_0, j_0), take W^2 (complex) coefficients from $I_X^{n_S,n_O}$, $c_x(i,j)$, W^2 coefficients from $I_Y^{n_S,n_O}$, $c_y(i,j)$, and compute the local statistics:

$$\mu_x = \frac{1}{W^2} \sum_{i,j} c_x(i,j) \tag{3.1}$$

$$\mu_y = \frac{1}{W^2} \sum_{i,j} c_y(i,j) \tag{3.2}$$

$$\sigma_x = \sqrt{\frac{1}{W^2-1} \sum_{i,j} |c_x(i,j) - \mu_x|^2} \tag{3.3}$$

$$\sigma_y = \sqrt{\frac{1}{W^2-1} \sum_{i,j} |c_y(i,j) - \mu_y|^2} \tag{3.4}$$

$$\rho_x(0,1) = \frac{E\{(c_x(i,j) - \mu_x)(c_x(i,j+1) - \mu_x)^*\}}{\sigma_x^2} \tag{3.5}$$

$$\rho_y(0,1) = \frac{E\{(c_y(i,j) - \mu_y)(c_y(i,j+1) - \mu_y)^*\}}{\sigma_y^2} \tag{3.6}$$

$$\rho_x(1,0) = \frac{E\{(c_x(i,j) - \mu_x)(c_x(i+1,j) - \mu_x)^*\}}{\sigma_x^2} \tag{3.7}$$

$$\rho_y(1,0) = \frac{E\{(c_y(i,j) - \mu_y)(c_y(i+1,j) - \mu_y)^*\}}{\sigma_y^2} \tag{3.8}$$

The expected values are computed by taking the empirical mean.

Now, calculate:

$$L(i_0,j_0) = \frac{2|\mu_x||\mu_y| + c_0}{|\mu_x|^2 + |\mu_y|^2 + c_0} \tag{3.9}$$

$$C(i_0,j_0) = \frac{2\sigma_x\sigma_y + c_1}{\sigma_x^2 + \sigma_y^2 + c_1} \tag{3.10}$$

$$C_{0,1}(i_0,j_0) = 1 - 0.5|\rho_x(0,1) - \rho_y(0,1)| \tag{3.11}$$

$$C_{1,0}(i_0,j_0) = 1 - 0.5|\rho_x(1,0) - \rho_y(1,0)| \tag{3.12}$$

It is important to note that, for an NxN image, these calculations will lead to matrices of size (N-W+1)x(N-W+1). This is because the only the 'valid' values are taken into account, i.e. the values computed entirely on the image coefficients.

This leads to having, for each subband, 4 matrices of local statistic similarities. Since the values in matrices are bound to lay between [0,1], we can combine them as follows and have one statistics matrix per subband (matrix multiplications done on a point-by-point basis):

$$S^{n_S,n_O} = L^{1/4} \cdot C^{1/4} \cdot C_{0,1}^{1/4} \cdot C_{1,0}^{1/4} \tag{3.13}$$

The power scaling by 0.25 is necessary to avoid having values that are too small - when multiplying values that are between 0 and 1, we risk to encounter errors due to precision of representation of such small numbers.

(3) Let us denote by $\mathbf{T} = \{(n_{S1}^1, n_{O1}^1, n_{S2}^1, n_{O2}^1), ..., (n_{S1}^M, n_{O1}^M, n_{S2}^M, n_{O2}^M)\}$ the set of all possible combinations of scales and orientations, such that it contains pairs of subbands at adjacent scales, for a given orientation, and pairs of subbands at all orientations, for a given scale. Then, for each combination $t \in \{1, ...M\}$ in $\mathbf{T}$, we take the subband images' *magnitudes*: $|I_X^{n_{S1}^t,n_{O1}^t}|$, $|I_X^{n_{S2}^t,n_{O2}^t}|$, $|I_Y^{n_{S1}^t,n_{O1}^t}|$, $|I_Y^{n_{S2}^t,n_{O2}^t}|$. Within each sliding window centered at (i_0, j_0), we take the (real) coefficients, respectively, c_{x_1}, c_{x_2}, c_{y_1}, c_{y_2}, and compute the following cross-correlation terms:

$$\mu_{x_1} = \frac{1}{W^2} \sum_{i,j} c_{x_1}(i,j) \tag{3.14}$$

$$\mu_{x_2} = \frac{1}{W^2} \sum_{i,j} c_{x_2}(i,j) \tag{3.15}$$

$$\mu_{y_1} = \frac{1}{W^2} \sum_{i,j} c_{y_1}(i,j) \tag{3.16}$$

$$\mu_{y_2} = \frac{1}{W^2} \sum_{i,j} c_{y_2}(i,j) \tag{3.17}$$

$$\sigma_{x_{11}} = \sqrt{\frac{1}{W^2 - 1} \sum_{i,j} |c_{x_1}(i,j) - \mu_{x_1}|^2} \tag{3.18}$$

$$\sigma_{x_{22}} = \sqrt{\frac{1}{W^2 - 1} \sum_{i,j} |c_{x_2}(i,j) - \mu_{x_2}|^2} \tag{3.19}$$

$$\sigma_{y_{11}} = \sqrt{\frac{1}{W^2 - 1} \sum_{i,j} |c_{y_1}(i,j) - \mu_{y_1}|^2} \tag{3.20}$$

$$\sigma_{y_{22}} = \sqrt{\frac{1}{W^2 - 1} \sum_{i,j} |c_{y_2}(i,j) - \mu_{y_2}|^2} \tag{3.21}$$

$$\rho_x = \frac{E\{(c_{x_1}(i,j) - \mu_{x_1})(c_{x_2}(i,j) - \mu_{x_2})\}}{\sigma_{x_{11}} \cdot \sigma_{x_{22}}} \tag{3.22}$$

$$\rho_y = \frac{E\{(c_{y_1}(i,j) - \mu_{y_1})(c_{y_2}(i,j) - \mu_{y_2})\}}{\sigma_{y_{11}} \cdot \sigma_{y_{22}}} \tag{3.23}$$

Since ρ_x and ρ_y are cross correlations, their values are bounded by [-1,1], thus we can combine them similarly to the autocovariance terms. The statistic that describes the similarity between the cross-correlations at location (i_0, j_0) is given by:

$$C_R^t(i_0, j_0) = 1 - 0.5|\rho_x - \rho_y| \tag{3.24}$$

For each combination t we get a (N-W+1)x(N-W+1) matrix of similarity measurements, thus we have total of M of these matrices.

With this proposed method, we have $N_S \cdot N_O + 2$ matrices of STSIM statistics, S^{n_S, n_O}, and M matrices with new statistics, C_R^t. The next question is how to combine those computed terms to obtain a single similarity measure value. Also, the question remains which new statistics to use and which to discard.

For a set of chosen matrices, we can follow the ways proposed in [**28**]:

Additive approach: In this case, the overall similarity measurement is taken as the average of all values in all matrices; the spatial averages are computed within each matrix, so each matrix votes for a point, and the final statistic is the mean of the voted points.

Multiplicative approach: This approach requires that all the matrices within the chosen set are multiplied point-by-point, as depicted in Figure 2.4. Again, we need power scaling by the inverse of number of matrices multiplied, to avoid dealing with small numbers. Once we have a final similarity matrix, we take the mean value of its elements as the final statistic.

Either of these approaches will produce one number, T_x, as a final texture similarity measurement.

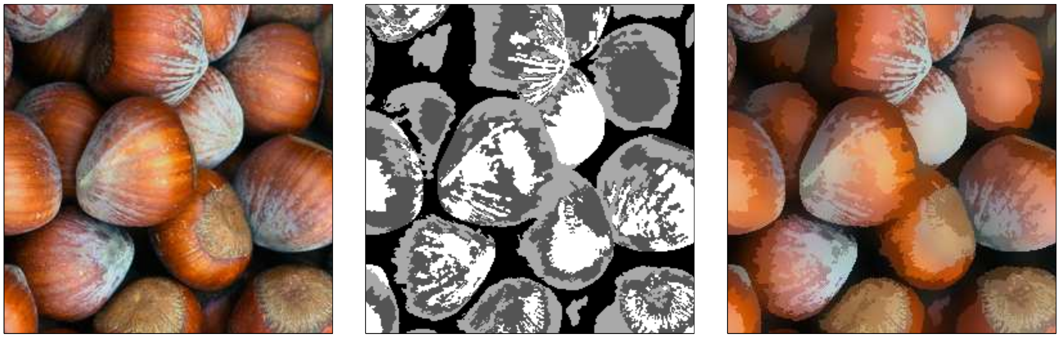
The issue of choosing the subset of matrices, and also different ways of combining them, will be discussed in Chapter 4.

3.2. New Color Similarity Metric

From the given background in Chapter 2, we can make some conclusions about how to develop a good similarity metric:

- First of all, the image should be processed as to remove the noise and smooth the color levels
- The colorspace in use should comply with HVS and human perception, to make the Euclidean distances meaningful
- The image should be represented by a compact, yet descriptive feature
- The features should be compared in a way that agrees with human perception

Given these statements, the first step in extracting the color composition would be image filtering. For this matter, we chose to segment the image using the ACA algorithm [**3**]. This algorithm gives good segmentation results, but also gives us the ”local averages” image as a by-product: that is, we get the smoothed image, where smoothing (spatial averaging) is done within small local windows (typically 7x7). The good characteristic of this method is that the averaging is performed only on pixels that belong to the same segment, which avoids possible blurring along the borders of two segments. The importance of this step lies in the fact that people don't perceive a lot of different colors at the same time [**5**], and they perform local averaging as opposed to noticing all the detailed variations of colors within small segments. An example of Local Averages image is shown in Figure 3.2.



(a) Original image (b) Segmentation map (c) Image of local averages

Figure 3.2. Example of ACA segmentation results

The next step is the choice of colorspace. Since CIELAB has perceptual uniformity [**36**], the smoothed images are represented in L*a*b* values.

For determining color similarity between two images, Rubner et al. [**39**] have found that the best results are obtained by using the multivariate histograms. Unfortunately, operating on these histograms is extremely computationally exhaustive, and their representation is not compact. That is why most of the approaches rely on dimensionality reduction, like calculating dominant colors [**32**], [**29**], [**38**].

On the other hand, in texture similarity algorithm, we opted for measurements within sliding windows that span a small local neighborhood. It would be reasonable to follow the same logic for the color comparisons, and to calculate color similarity on a sliding window. This works to our advantage, as now, since we are operating on a relatively small number of pixels, we can use the complete information about the colors contained within the windows, without having to worry about complexity.

As for the comparison method, Earth Mover's Distance [**40**] and OCCD [**32**] are the well-known procedures that perform in accordance with human judgements and perception. We chose to use OCCD for two reasons: first, unlike EMD, it uses ”units” of colors; since we are operating on small sliding windows, it is very easy to simply use each pixel within the window as one color unit and to compute OCCD distance between two sets of W^2 units; second, OCCD problem be solved by a deterministic algorithm, and its implementation is fairly straight-forward.

The color comparison algorithm between two (color) images I_X and I_Y can be summarized as follows:

(1) Segment the images and get the images of local averages, $I_{X,lav}$ and $I_{Y,lav}$

(2) For each sliding window of size WxW centered at (i_0, j_0), take W^2 pixels from each local average image, make the sets of color units, $C_X = \{c_{x1}, ..., c_{xW^2}\}$, $C_Y = \{c_{y1}, ..., c_{yW^2}\}$, and compute:

$$OCCD(i_0, j_0) = \min_{m_{XY}} \sum_{i=1}^{n} d(c_{xi}, m_{XY}(c_{xi})), \tag{3.25}$$

where $\{m_{XY} : C_X \to C_Y\}$ is a previously defined bijection function that maps set C_X to set C_Y, and $d(c_X, c_Y)$ is the Euclidean distance between two colors (c_X, c_Y).

Example 3. Here is an example of the proposed algorithm. Let's take two images of similar colors, like a leaf and patch of grass (Fig. 3.3). First, we need their local averages images, as given in Figure 3.4.

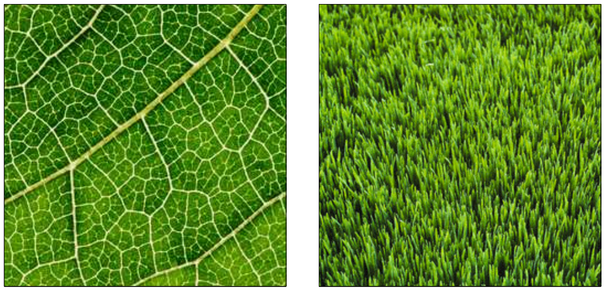


Figure 3.3. OCCD Example: Original images

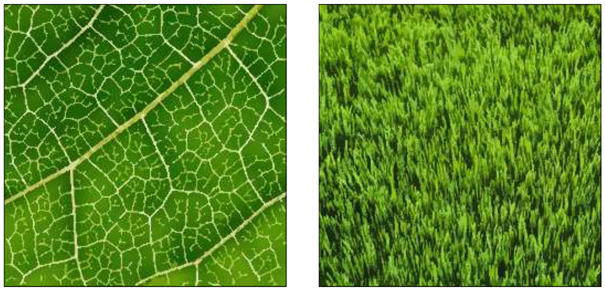


Figure 3.4. OCCD Example: Local Averages images

Then, we compute OCCD on sliding windows. For example, for a window centered at (164,113), the corresponding leaf and grass patches are given in Figure 3.5:

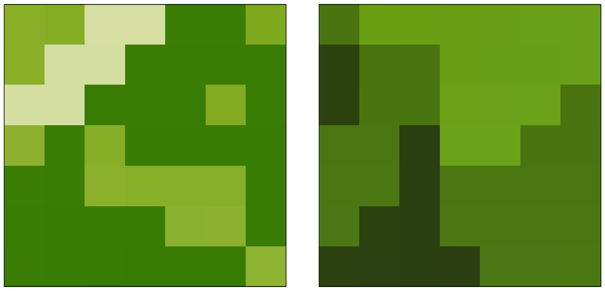


Figure 3.5. OCCD Example: Sliding window patches taken at location [164,113]

For this particular window, OCCD is 0.0753. In Figure 3.6, we are giving the result for the color units matching, sorted by ascending distances. If we take an image with completely different colors e.g. pink, as in Figure 3.7, take the window at the same location, and compare it with the ”leaf” image window, we get OCCD equal to 0.5579.

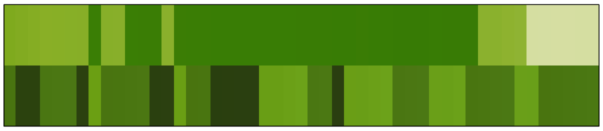


Figure 3.6. OCCD Example: Matching units of color, sorted by ascending distances

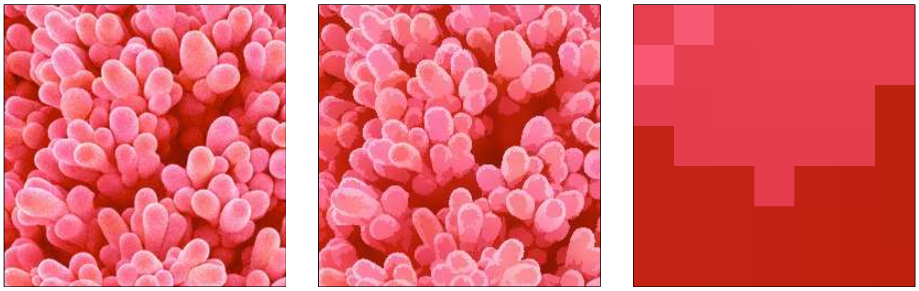


Figure 3.7. OCCD Example: Original, Local Average and sliding window image with different color composition

The color similarity metrics are given in one (N-W+1)x(N-W+1) matrix that we shall call $OCCD$ matrix, and the range of values is [0,1]. Scaling the matching costs to be within these boundaries is very important, since the texture similarity metrics are in this range, therefore we need to have comparable units.

Since the OCCD computes the *distance* between two colors, their *similarity* can be rated as $1 - distance$. Thus, as a similarity measure, we use the matrix $OCCD_S = 1 - OCCD$. We can either calculate the mean of $OCCD_S$ matrix as the color similarity measurement C_l, or we can combine the matrix in a different way with the texture similarity matrices. This will be addressed in the following section.

3.3. Combining Texture and Color Similarity Matrices

After performing similarity calculations on sliding windows, we have in the end matrices for texture similarities, $S^{ns,no}$ and C_R^t, as well as the color similarity matrix $OCCD_S$. The widely used solution is to linearly combine texture similarity measure, T_x, and color similarity measure, C_l, with appropriate weights, w_T and $w_C = 1 - w_T$:

$$SIMILARITY = w_T \cdot T_x + w_C \cdot C_l \tag{3.26}$$

Another possible way is to combine the $OCCD_S$ matrix as if it was another texture similarity matrix, i.e. to multiply it point-by-point with the texture similarity matrices. This was explored in our experiments, however, the linear combination yielded better results.

We evaluated our method by comparing our results to human judgements. The experiment set-up and discussion will be given in the following chapter.

CHAPTER 4

Experiments and Results

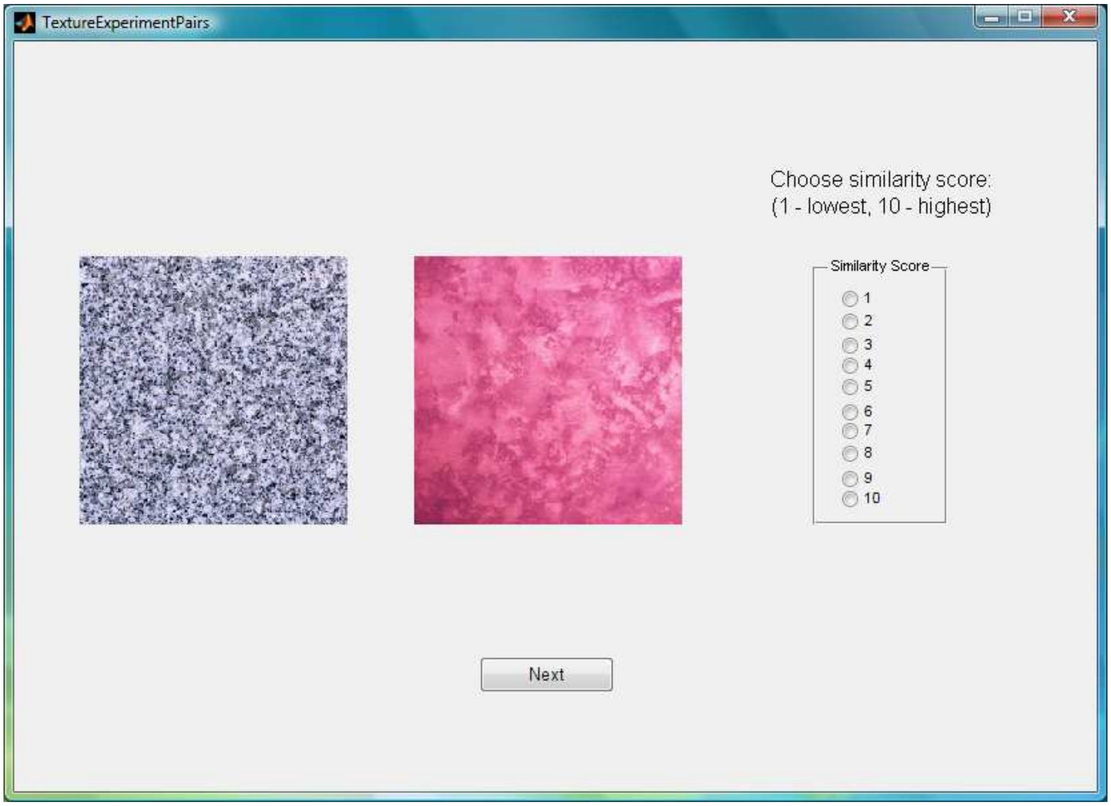
The new texture image similarity assessment method was proposed in Chapter 3. It combines what are believed to be the best features for texture aspect and color portion - the metric utilizes the methods that incorporate human vision properties.

We conducted an informal subject test, where people were asked to rate similarity between two images. Then, we applied our method with varying parameters. In the end, the results were compared to the human judgements, and the best method (among the proposed ones) is found.

4.1. Experiment Set-up: The Informal Test

The informal test was conducted using Matlab GUI. We carefully picked 50 pairs from a pool of 30 textures, extracted from Corbis online database [**48**]. Each human subject was asked to grade 50 pairs of textures according to their similarity, with the lowest grade being 1 (”completely dissimilar”) and the highest grade being 10 (”identical/almost identical”). A snapshot of the user interface is given in Figure 4.1, and the selected pairs are given in Figure 4.2.

People were asked to rank similarity between two images *as they perceive it*, so there was no guidance as for how they should rank textures that are of similar structure, similar color, both or neither. This works both towards our advantage but also disadvantage: since there were no strict rules, people were free to make judgements according to their own perception



Choose similarity score:
(1 - lowest, 10 - highest)

Figure 4.1. Snapshot of Matlab GUI for subjects test

and thinking, which we are trying to reach as an ultimate goal. On the other hand, people cannot separate their visual stimuli from the context they associate with it; thus, e.g. pairs of textures that both represent grass were automatically ranked as ”highly similar” even though at the grayscale level they are quite different, while some pairs, like asphalt and grass, are ranked very low, even though their grayscale representations appear to be very similar, i.e. they have very similar structure pixel-wise.

The final score assigned to a pair of textures is the mean value of all the scores human subjects assigned to it. Pairs, sorted according to the descending ranking of similarity, are given in Figure 4.3.

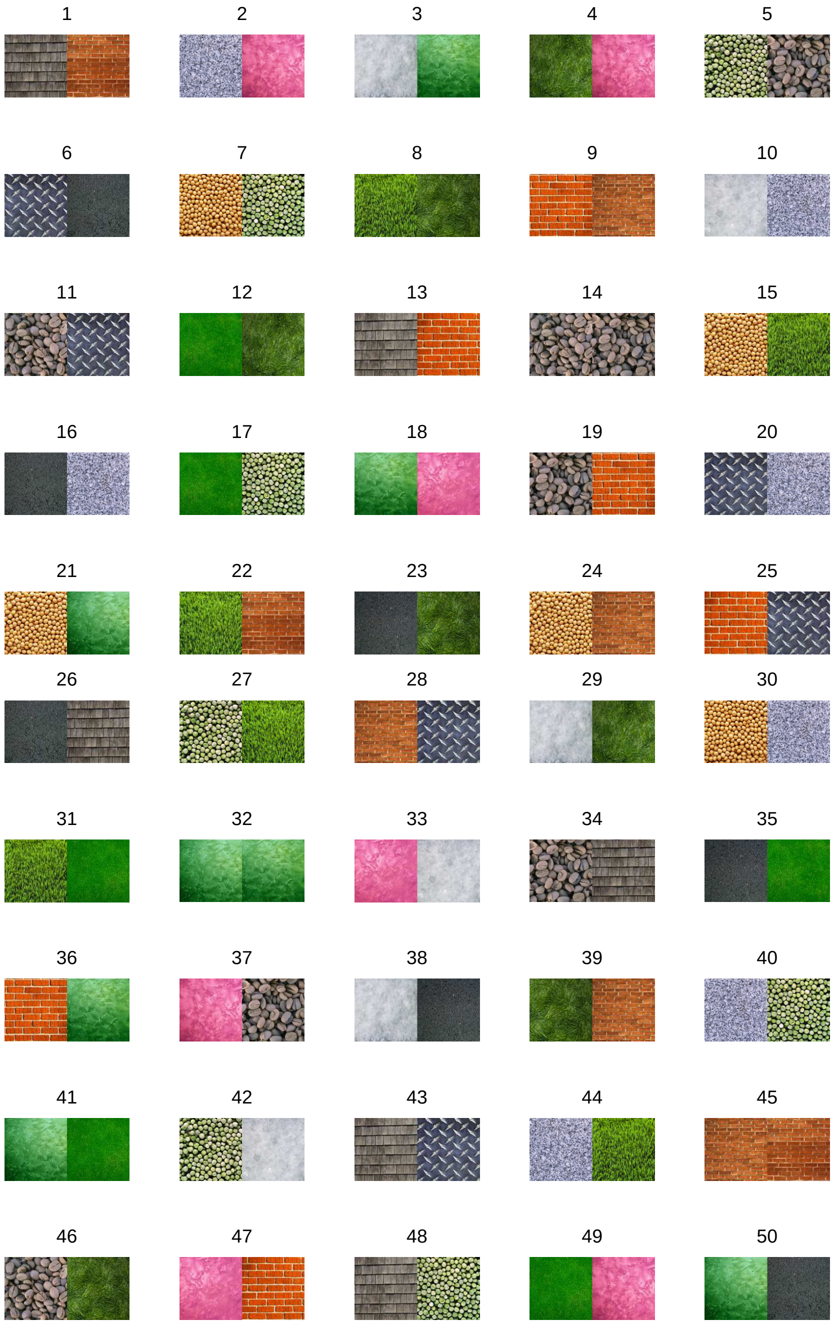


Figure 4.2. Texture pairs for testing the metric

4.1.1. Inter-Human Agreement on the Scores

To illustrate how people ranked the textures and how consistent they are among themselves, we are giving a boxplot in Figure 4.4. For each pair, the horizontal red line is the median

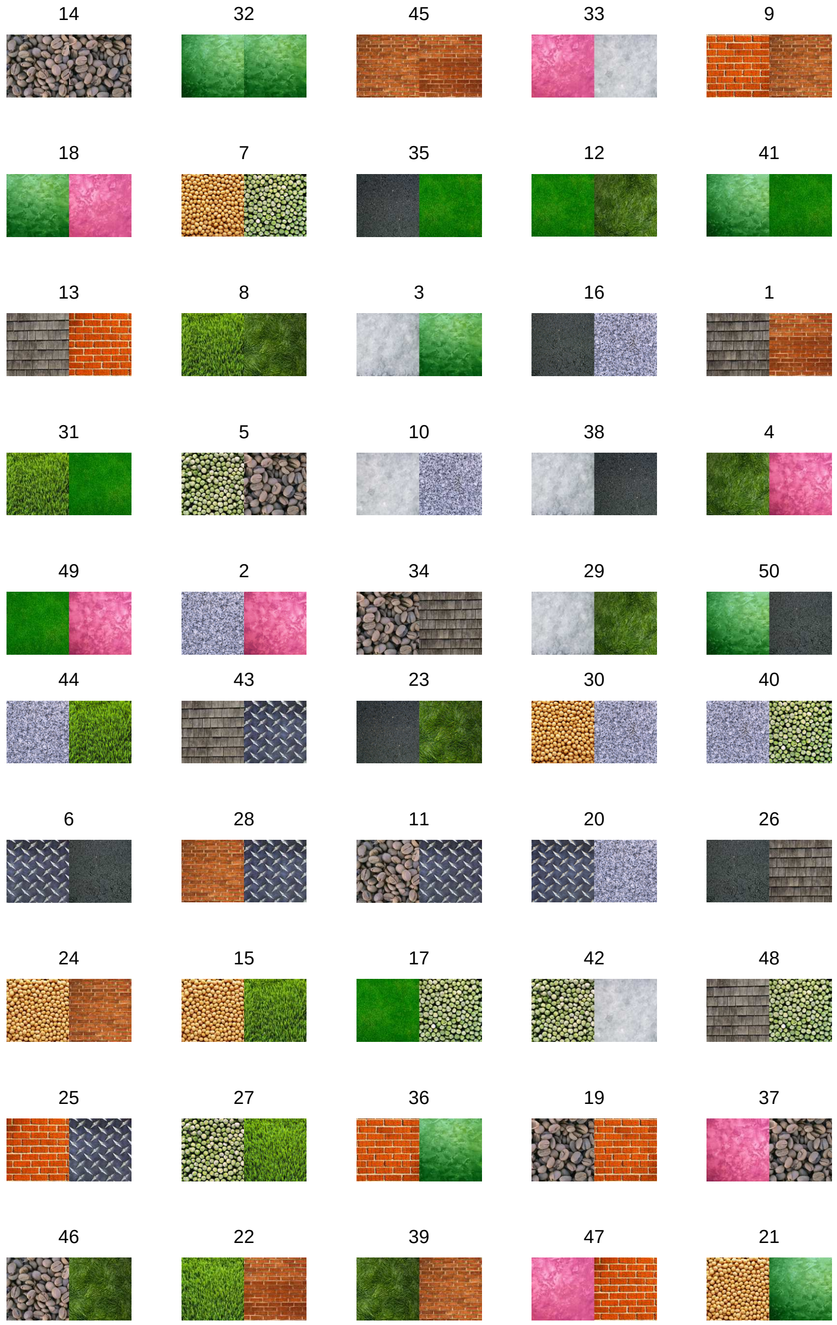


Figure 4.3. Texture pairs sorted in descending order according to human judgements of similarity

value, the blue boxes represent the limits of the upper and lower quartile range; ”whiskers” extend to the most extreme values within 1.5 times the interquartile range from the ends of the box; outliers are marked with red ”+” signs. We can see that humans agree fairly well for the two extreme cases, when textures are very similar or very dissimilar, while for the middle ground there is moderate disagreement.

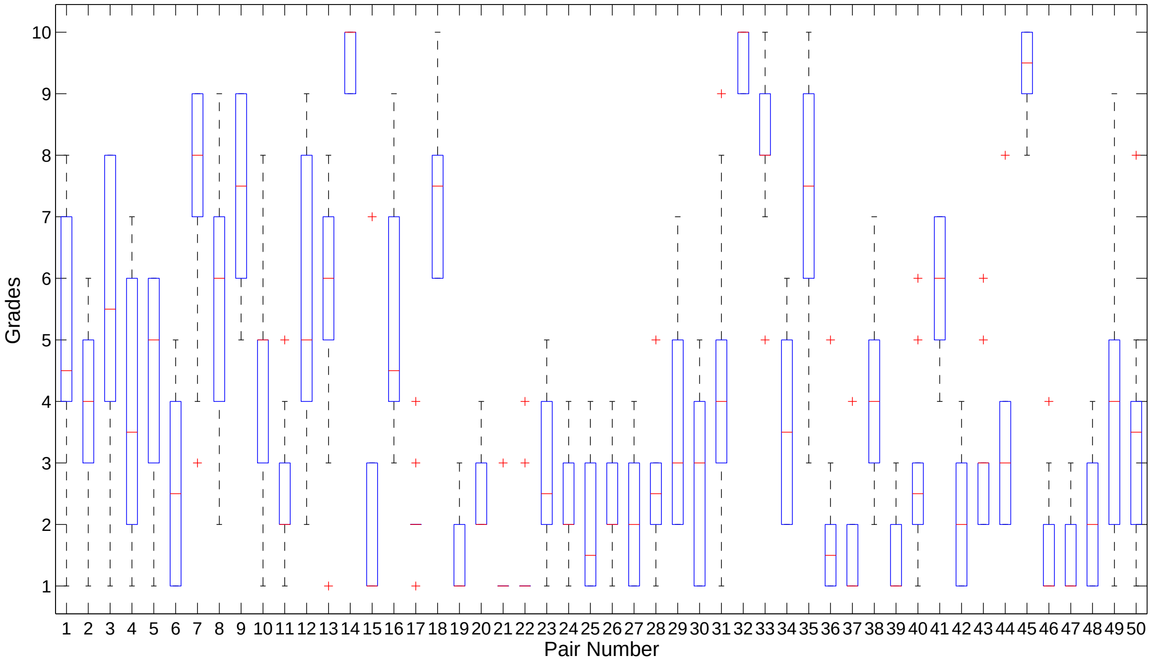


Figure 4.4. Whisker plot for human grades

4.2. Computing the Texture Image Similarity Score

As noted earlier, for steerable pyramid subbands, we chose the 3-scales, 4-orientations decomposition, which is given in Figures 2.3 and 3.1. The images are of size 128x128, and the sliding windows are 7x7, which gives us similarity matrices of size 122x122.

To get the texture similarity scores, we have tried various approaches. They can be divided in two major categories:

Linear Combination: in this case, we used the linear combination of texture similarity T_x and color similarity C_l with varying weights; here, we can apply both Additive and Multiplicative approach for calculating T_x, as described in Section 3.1

Multiplicative Combination: in this case, we treat the $OCCD_S$ matrix as an equal similarity measurement like the texture similarity matrices, thus we can only apply the Multiplicative approach.

For each of these categories, we have varied the subset of the newly introduced C_R^t matrices. If we label the subbands by numbers as given in Figure 4.5, we have tested the following subsets:

(1) No new matrices used

(2) All of the new matrices used

(3) Using only correlations between scales for a given orientation, e.g. correlation between subbands 1 and 2, 2 and 3, 4 and 5 and so on

(4) Using only correlations between different orientations for a given scale, e.g. between subbands 1 and 4, 1 and 7, 1 and 10, 2 and 5 and so on

(5) Using correlation between orthogonal bands at highest scale, i.e. 1 and 7, 4 and 10

(6) Using Using correlation between orthogonal bands at highest scale and between the diagonal bands at lower scales, i.e. 1 and 7, 4 and 10, 5 and 11, 6 and 12

(7) Using only correlations between subbands with same orientation at highest and medium scale, i.e. between subbands 1 and 2, 4 and 5, 7 and 8, 10 and 11

(8) Using only correlations between subbands with same orientation at medium and lowest scale, i.e. between subbands 2 and 3, 5 and 6, 8 and 9, 11 and 12.

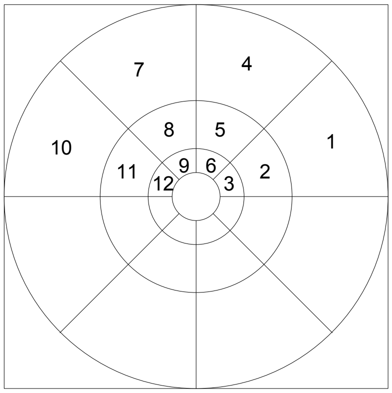


Figure 4.5. Numbered subbands of steerable filter decomposition

These combination are not chosen randomly. Apart from the first four cases that are self-explanatory, the latter four are chosen by examining properties of a large number of textures. In this particular experiment we are using 30 images of textures, but we have a database of over 300 textures, where most of them are grouped in duplets or triplets of textures that are extracted from the same, bigger image. This leads to having over 100 of sets of textures, where we know that the ones within the same set should be similar (since they're taken from the same image). Looking at the cross-correlations between subbands, the attempt was to find features that will have low intra-set variances, and high inter-set variances. That would suggest a feature is discriminative between different textures, but also has similar values for (almost) identical textures.

For the Linear Combination approach, we varied the weights from 0 to 1 (for w_T; corresponding values of w_C are 1 to 0) with 0.1 steps.

4.3. Performance Evaluation

Evaluating performance of similarity evaluation systems is difficult. Since we are using a relatively small set of pairs, we are mostly interested in how does our ranking of textures, based on similarity, compare to the human rankings (given in Fig 4.3).

For this purpose, we used two tests: Kendall tau rank correlation coefficient (developed by Maurice Kendall) which is ”a non-parametric statistic used to measure the degree of correspondence between two rankings and assessing the significance of this correspondence” (from Wikipedia, [**49**]); and Spearman's rank correlation coefficient, which is ”a non-parametric measure of correlation that is, it assesses how well an arbitrary monotonic function could describe the relationship between two variables, without making any assumptions about the frequency distribution of the variables” (from Wikipedia, [**50**]).

In our case, these two tests are used to measure the performance of our metric, compared to the human judgements. We submit the values assigned to the textures by our method, along with the mean human evaluated grades. The tests give us two values, correlation coefficient and significance. The correlation coefficient varies between [-1,1]: negative correlation means we have opposite rankings, correlations around zero means that the rankings are uncorrelated, while coefficients close to 1 mean we have good agreement between rankings. Significance value (p-value) describes how likely we would be to score that correlation coefficient, if one of the rankings was assigned randomly. For example, result of one simulation was $\rho = 0.60053$ and $p_{val} = 4.0191e-06$, which means that if we drew random rankings, we would get this score about four times in a million trials.

4.3.1. Linear Combination performance evaluation

Here are given the tables with Kendall tau and Spearman rank correlation coefficients and p-values. The "Bands" column corresponds to the previously numbered list of used subset of correlation matrices. The weights for texture similarity are given by w_T and for color similarity by w_C; Kendall tau rank correlation coefficient and its p-value are denoted as ρ_K, $p_{val,K}$, while Spearman's rank correlation coefficient and its significance value are given by ρ_S, $p_{val,S}$. We tested two approaches, additive and multiplicative, and we have separate tables for these two methods of combining texture information. To represent the results, we will use multiple tables, grouped in two subset combinations.

Table 4.1. Linear Combination, Additive Approach, Bands 1 and 2

Bands	w_T	w_C	ρ_K	$p_{val,K}$	ρ_S	$p_{val,S}$
1	1.0	0.0	0.4374	9.1090e-006	0.5962	4.9091e-006
1	0.9	0.1	0.4457	6.1539e-006	0.5984	4.4435e-006
1	0.8	0.2	0.4358	9.8443e-006	0.5874	7.3235e-006
1	0.7	0.3	0.4391	8.4264e-006	0.5902	6.4502e-006
1	0.6	0.4	0.4358	9.8443e-006	0.5855	7.9426e-006
1	0.5	0.5	0.4391	8.4264e-006	0.5918	6.0049e-006
1	0.4	0.6	0.4143	2.6353e-005	0.5681	1.6798e-005
1	0.3	0.7	0.3962	5.8521e-005	0.5450	4.2737e-005
1	0.2	0.8	0.3501	3.8545e-004	0.4899	3.0474e-004
1	0.1	0.9	0.3007	2.3075e-003	0.4137	2.8222e-003
1	0.0	1.0	0.1491	1.3181e-001	0.2193	1.2589e-001
2	1.0	0.0	0.4753	1.4182e-006	0.6385	6.0592e-007
2	0.9	0.1	**0.4885**	**7.1815e-007**	0.6543	2.5506e-007
2	0.8	0.2	0.4852	8.5270e-007	0.6568	2.2204e-007
2	0.7	0.3	0.4852	8.5270e-007	0.6584	2.0229e-007
2	0.6	0.4	0.4803	1.1010e-006	**0.6587**	**1.9898e-007**
2	0.5	0.5	0.4687	1.9800e-006	0.6379	6.2796e-007
2	0.4	0.6	0.4457	6.1539e-006	0.6036	3.4800e-006
2	0.3	0.7	0.3979	5.4500e-005	0.5399	5.1909e-005
2	0.2	0.8	0.3468	4.3751e-004	0.4863	3.4265e-004
2	0.1	0.9	0.2694	6.3457e-003	0.3759	7.1405e-003
2	0.0	1.0	0.1491	1.3181e-001	0.2193	1.2589e-001

Table 4.2. Linear Combination, Additive Approach, Bands 3 and 4

Bands	w_T	w_C	ρ_K	$p_{val,K}$	ρ_S	$p_{val,S}$
3	1.0	0.0	0.4704	1.8222e-006	0.6261	1.1548e-006
3	0.9	0.1	0.4687	1.9800e-006	0.6272	1.0915e-006
3	0.8	0.2	0.4704	1.8222e-006	0.6240	1.2855e-006
3	0.7	0.3	0.4720	1.6766e-006	0.6269	1.1132e-006
3	0.6	0.4	0.4753	1.4182e-006	0.6212	1.4788e-006
3	0.5	0.5	0.4555	3.8098e-006	0.6061	3.0993e-006
3	0.4	0.6	0.4374	9.1090e-006	0.5967	4.8019e-006
3	0.3	0.7	0.4028	4.3949e-005	0.5515	3.3132e-005
3	0.2	0.8	0.3484	4.1071e-004	0.4759	4.7761e-004
3	0.1	0.9	0.2924	3.0388e-003	0.4095	3.1438e-003
3	0.0	1.0	0.1491	1.3181e-001	0.2193	1.2589e-001
4	1.0	0.0	0.4424	7.2049e-006	0.6084	2.7767e-006
4	0.9	0.1	0.4489	5.2505e-006	0.6144	2.0769e-006
4	0.8	0.2	0.4457	6.1539e-006	0.6142	2.0965e-006
4	0.7	0.3	0.4440	6.6596e-006	0.6087	2.7386e-006
4	0.6	0.4	0.4473	5.6850e-006	0.6120	2.3290e-006
4	0.5	0.5	0.4473	5.6850e-006	0.6102	2.5378e-006
4	0.4	0.6	0.4275	1.4454e-005	0.5926	5.7993e-006
4	0.3	0.7	0.3798	1.1752e-004	0.5186	1.1423e-004
4	0.2	0.8	0.3270	9.1504e-004	0.4590	7.9962e-004
4	0.1	0.9	0.2760	5.1685e-003	0.3812	6.3113e-003
4	0.0	1.0	0.1491	1.3181e-001	0.2193	1.2589e-001

Table 4.3. Linear Combination, Additive Approach, Bands 5 and 6

Bands	w_T	w_C	ρ_K	$p_{val,K}$	ρ_S	$p_{val,S}$
5	1.0	0.0	0.4440	6.6596e-006	0.6060	3.1135e-006
5	0.9	0.1	0.4555	3.8098e-006	0.6109	2.4568e-006
5	0.8	0.2	0.4489	5.2505e-006	0.6043	3.3638e-006
5	0.7	0.3	0.4555	3.8098e-006	0.6075	2.8939e-006
5	0.6	0.4	0.4539	4.1295e-006	0.6085	2.7640e-006
5	0.5	0.5	0.4473	5.6850e-006	0.5983	4.4633e-006
5	0.4	0.6	0.4275	1.4454e-005	0.5824	9.1341e-006
5	0.3	0.7	0.4061	3.8025e-005	0.5588	2.4668e-005
5	0.2	0.8	0.3567	2.9824e-004	0.4909	2.9529e-004
5	0.1	0.9	0.3056	1.9500e-003	0.4258	2.0513e-003
5	0.0	1.0	0.1491	1.3181e-001	0.2193	1.2589e-001
6	1.0	0.0	0.4358	9.8443e-006	0.5990	4.3168e-006
6	0.9	0.1	0.4457	6.1539e-006	0.6054	3.1925e-006
6	0.8	0.2	0.4457	6.1539e-006	0.6084	2.7767e-006
6	0.7	0.3	0.4440	6.6596e-006	0.6054	3.1925e-006
6	0.6	0.4	0.4374	9.1090e-006	0.5987	4.3749e-006
6	0.5	0.5	0.4325	1.1488e-005	0.5942	5.3841e-006
6	0.4	0.6	0.4242	1.6823e-005	0.5913	6.1367e-006
6	0.3	0.7	0.3995	5.0742e-005	0.5581	2.5449e-005
6	0.2	0.8	0.3468	4.3751e-004	0.4818	3.9557e-004
6	0.1	0.9	0.2875	3.5734e-003	0.4030	3.7113e-003
6	0.0	1.0	0.1491	1.3181e-001	0.2193	1.2589e-001

Table 4.4. Linear Combination, Additive Approach, Bands 7 and 8

Bands	w_T	w_C	ρ_K	$p_{val,K}$	ρ_S	$p_{val,S}$
7	1.0	0.0	0.4621	2.7525e-006	0.6128	2.2437e-006
7	0.9	0.1	0.4704	1.8222e-006	0.6186	1.6908e-006
7	0.8	0.2	0.4737	1.5422e-006	0.6220	1.4229e-006
7	0.7	0.3	0.4737	1.5422e-006	0.6235	1.3234e-006
7	0.6	0.4	0.4737	1.5422e-006	0.6206	1.5257e-006
7	0.5	0.5	0.4555	3.8098e-006	0.6025	3.6655e-006
7	0.4	0.6	0.4457	6.1539e-006	0.5907	6.3121e-006
7	0.3	0.7	0.4242	1.6823e-005	0.5740	1.3140e-005
7	0.2	0.8	0.3748	1.4410e-004	0.5012	2.0959e-004
7	0.1	0.9	0.2908	3.2083e-003	0.4090	3.1866e-003
7	0.0	1.0	0.1491	1.3181e-001	0.2193	1.2589e-001
8	1.0	0.0	0.4391	8.4264e-006	0.5925	5.8120e-006
8	0.9	0.1	0.4506	4.8479e-006	0.6014	3.8515e-006
8	0.8	0.2	0.4457	6.1539e-006	0.5931	5.6495e-006
8	0.7	0.3	0.4539	4.1295e-006	0.6016	3.8171e-006
8	0.6	0.4	0.4424	7.2049e-006	0.5936	5.5395e-006
8	0.5	0.5	0.4325	1.1488e-005	0.5843	8.3759e-006
8	0.4	0.6	0.4160	2.4470e-005	0.5704	1.5309e-005
8	0.3	0.7	0.3896	7.7593e-005	0.5459	4.1324e-005
8	0.2	0.8	0.3353	6.7597e-004	0.4629	7.1265e-004
8	0.1	0.9	0.2858	3.7698e-003	0.4015	3.8573e-003
8	0.0	1.0	0.1491	1.3181e-001	0.2193	1.2589e-001

Highlighted are the two maximum values for different tests; we see that the Kendall tau correlation coefficient is maximum for using all subband correlations, with weights $w_T = 0.9$ and $w_C = 0.1$, while the Spearman rank correlation coefficient is also maximal for using all subband correlations, but different weights, $w_T = 0.6$ and $w_C = 0.4$. This is in agreement with findings of Mojsilovic et al. [**5**], where the similarity of two textured images mainly depends on the structure of images, and less on color.

In the following tables are the results for using Linear Combination, Multiplicative Approach.

Table 4.5. Linear Combination, Multiplicative Approach, Bands 1 and 2

Bands	w_T	w_C	ρ_K	$p_{val,K}$	ρ_S	$p_{val,S}$
1	1.0	0.0	0.4341	1.0636e-005	0.5940	5.4434e-006
1	0.9	0.1	0.4407	7.7928e-006	0.5948	5.2442e-006
1	0.8	0.2	0.4358	9.8443e-006	0.5866	7.5785e-006
1	0.7	0.3	0.4407	7.7928e-006	0.5917	6.0179e-006
1	0.6	0.4	0.4407	7.7928e-006	0.5960	4.9635e-006
1	0.5	0.5	0.4391	8.4264e-006	0.5914	6.1234e-006
1	0.4	0.6	0.4209	1.9559e-005	0.5736	1.3329e-005
1	0.3	0.7	0.3979	5.4500e-005	0.5451	4.2578e-005
1	0.2	0.8	0.3435	4.9607e-004	0.4853	3.5455e-004
1	0.1	0.9	0.3023	2.1821e-003	0.4194	2.4296e-003
1	0.0	1.0	0.1491	1.3181e-001	0.2193	1.2589e-001
2	1.0	0.0	0.4770	1.3038e-006	0.6412	5.2476e-007
2	0.9	0.1	0.4835	9.2878e-007	0.6469	3.8683e-007
2	0.8	0.2	**0.4885**	**7.1815e-007**	**0.6553**	**2.4226e-007**
2	0.7	0.3	0.4704	1.8222e-006	0.6420	5.0351e-007
2	0.6	0.4	0.4770	1.3038e-006	0.6498	3.2827e-007
2	0.5	0.5	0.4654	2.3357e-006	0.6374	6.4416e-007
2	0.4	0.6	0.4424	7.2049e-006	0.6044	3.3562e-006
2	0.3	0.7	0.3929	6.7422e-005	0.5386	5.4542e-005
2	0.2	0.8	0.3468	4.3751e-004	0.4815	3.9984e-004
2	0.1	0.9	0.2776	4.9068e-003	0.3861	5.6186e-003
2	0.0	1.0	0.1491	1.3181e-001	0.2193	1.2589e-001

Table 4.6. Linear Combination, Multiplicative Approach, Bands 3 and 4

Bands	w_T	w_C	ρ_K	$p_{val,K}$	ρ_S	$p_{val,S}$
3	1.0	0.0	0.4737	1.5422e-006	0.6281	1.0443e-006
3	0.9	0.1	0.4720	1.6766e-006	0.6279	1.0572e-006
3	0.8	0.2	0.4753	1.4182e-006	0.6238	1.3012e-006
3	0.7	0.3	0.4687	1.9800e-006	0.6218	1.4402e-006
3	0.6	0.4	0.4737	1.5422e-006	0.6257	1.1834e-006
3	0.5	0.5	0.4539	4.1295e-006	0.6043	3.3638e-006
3	0.4	0.6	0.4407	7.7928e-006	0.6047	3.3033e-006
3	0.3	0.7	0.4127	2.8374e-005	0.5687	1.6430e-005
3	0.2	0.8	0.3583	2.7953e-004	0.4869	3.3631e-004
3	0.1	0.9	0.2974	2.5781e-003	0.4150	2.7319e-003
3	0.0	1.0	0.1491	1.3181e-001	0.2193	1.2589e-001
4	1.0	0.0	0.4275	1.4454e-005	0.6014	3.8689e-006
4	0.9	0.1	0.4457	6.1539e-006	0.6151	2.0050e-006
4	0.8	0.2	0.4374	9.1090e-006	0.6052	3.2217e-006
4	0.7	0.3	0.4292	1.3393e-005	0.5970	4.7281e-006
4	0.6	0.4	0.4374	9.1090e-006	0.6028	3.6163e-006
4	0.5	0.5	0.4374	9.1090e-006	0.6018	3.7829e-006
4	0.4	0.6	0.4176	2.2716e-005	0.5878	7.1682e-006
4	0.3	0.7	0.3666	2.0137e-004	0.5114	1.4762e-004
4	0.2	0.8	0.3237	1.0310e-003	0.4604	7.6815e-004
4	0.1	0.9	0.2809	4.4192e-003	0.3910	4.9874e-003
4	0.0	1.0	0.1491	1.3181e-001	0.2193	1.2589e-001

Table 4.7. Linear Combination, Multiplicative Approach, Bands 5 and 6

Bands	w_T	w_C	ρ_K	$p_{val,K}$	ρ_S	$p_{val,S}$
5	1.0	0.0	0.4473	5.6850e-006	0.6071	2.9541e-006
5	0.9	0.1	0.4522	4.4749e-006	0.6093	2.6517e-006
5	0.8	0.2	0.4506	4.8479e-006	0.6062	3.0782e-006
5	0.7	0.3	0.4555	3.8098e-006	0.6093	2.6517e-006
5	0.6	0.4	0.4588	3.2400e-006	0.6129	2.2333e-006
5	0.5	0.5	0.4489	5.2505e-006	0.5969	4.7596e-006
5	0.4	0.6	0.4308	1.2406e-005	0.5863	7.6927e-006
5	0.3	0.7	0.4045	4.0885e-005	0.5604	2.3126e-005
5	0.2	0.8	0.3567	2.9824e-004	0.4929	2.7587e-004
5	0.1	0.9	0.3040	2.0631e-003	0.4235	2.1823e-003
5	0.0	1.0	0.1491	1.3181e-001	0.2193	1.2589e-001
6	1.0	0.0	0.4358	9.8443e-006	0.5974	4.6451e-006
6	0.9	0.1	0.4424	7.2049e-006	0.6027	3.6245e-006
6	0.8	0.2	0.4424	7.2049e-006	0.6063	3.0572e-006
6	0.7	0.3	0.4424	7.2049e-006	0.6048	3.2883e-006
6	0.6	0.4	0.4391	8.4264e-006	0.6022	3.7238e-006
6	0.5	0.5	0.4275	1.4454e-005	0.5870	7.4500e-006
6	0.4	0.6	0.4176	2.2716e-005	0.5808	9.7900e-006
6	0.3	0.7	0.3995	5.0742e-005	0.5621	2.1588e-005
6	0.2	0.8	0.3468	4.3751e-004	0.4840	3.6910e-004
6	0.1	0.9	0.2957	2.7240e-003	0.4145	2.7629e-003
6	0.0	1.0	0.1491	1.3181e-001	0.2193	1.2589e-001

Table 4.8. Linear Combination, Multiplicative Approach, Bands 7 and 8

Bands	w_T	w_C	ρ_K	$p_{val,K}$	ρ_S	$p_{val,S}$
7	1.0	0.0	0.4654	2.3357e-006	0.6147	2.0479e-006
7	0.9	0.1	0.4687	1.9800e-006	0.6126	2.2595e-006
7	0.8	0.2	0.4687	1.9800e-006	0.6186	1.6908e-006
7	0.7	0.3	0.4704	1.8222e-006	0.6232	1.3428e-006
7	0.6	0.4	0.4671	2.1508e-006	0.6167	1.8548e-006
7	0.5	0.5	0.4638	2.5359e-006	0.6098	2.5912e-006
7	0.4	0.6	0.4440	6.6596e-006	0.5898	6.5769e-006
7	0.3	0.7	0.4242	1.6823e-005	0.5732	1.3549e-005
7	0.2	0.8	0.3732	1.5416e-004	0.5003	2.1616e-004
7	0.1	0.9	0.2941	2.8775e-003	0.4130	2.8719e-003
7	0.0	1.0	0.1491	1.3181e-001	0.2193	1.2589e-001
8	1.0	0.0	0.4555	3.8098e-006	0.6061	3.0923e-006
8	0.9	0.1	0.4572	3.5138e-006	0.6047	3.3033e-006
8	0.8	0.2	0.4522	4.4749e-006	0.5988	4.3555e-006
8	0.7	0.3	0.4522	4.4749e-006	0.6020	3.7575e-006
8	0.6	0.4	0.4407	7.7928e-006	0.5922	5.9013e-006
8	0.5	0.5	0.4259	1.5596e-005	0.5786	1.0755e-005
8	0.4	0.6	0.4127	2.8374e-005	0.5686	1.6496e-005
8	0.3	0.7	0.3896	7.7593e-005	0.5447	4.3299e-005
8	0.2	0.8	0.3419	5.2802e-004	0.4663	6.4269e-004
8	0.1	0.9	0.2891	3.3864e-003	0.4060	3.4423e-003
8	0.0	1.0	0.1491	1.3181e-001	0.2193	1.2589e-001

The best results are highlighted, and for both test they occur at $w_T = 0.8$ and $w_C = 0.2$. We can see that, according to Kendall tau correlation coefficient, there is a tie between the Additive approach with weights $w_T = 0.9$ and $w_C = 0.1$, and Multiplicative approach with weights $w_T = 0.8$ and $w_C = 0.2$. However, according to Spearman rank correlation coefficient, Additive approach is superior to the Multiplicative. We can also see that the results are statistically significant, since the p-values are of the order 10^{-7} which means we would randomly get these results in a few trials out of ten million.

Here, we are presenting the results for using the 8 different subsets of subbands with color information embedded in a multiplicative manner.

Table 4.9. Multiplicative Combination

Bands	ρ_K	$p_{val,K}$	ρ_S	$p_{val,S}$
1	0.4407	7.7928e-006	0.5948	5.2442e-006
2	**0.4786**	**1.1983e-006**	**0.6413**	**5.2206e-007**
3	0.4704	1.8222e-006	0.6251	1.2155e-006
4	0.4308	1.2406e-005	0.6022	3.7238e-006
5	0.4489	5.2505e-006	0.6055	3.1780e-006
6	0.4391	8.4264e-006	0.6005	4.0192e-006
7	0.4687	1.9800e-006	0.6173	1.7986e-006
8	0.4572	3.5138e-006	0.6032	3.5436e-006

We can see again that the maximum values occur if we're using all subband correlations. However, this way of combining information performs worse than the Linear Combination.

4.3.2. Evaluating the Significance of Tests

To evaluate the performance of our metric, we examined how well PSNR works, and the coefficient of correlation for Spearman's test was 0.283. Also, we examined how SSIM [**4**] performs, with Matlab implementation downloaded from the website [**51**]. We used the same varying weights, and the best result obtained was 0.5147. As can be seen from the tables above, ST-SIM has best performance of 0.5984.

Since raw numbers like $\rho = 0.65$ are not very descriptive if we do not know what to expect, we conducted the following experiment: let's denote the number of human subjects as N_s; for each one of the human subjects, we removed their judgements from the pool, and computed the mean grades of the remaining $N_s - 1$ subjects. Then, we conducted the Kendall tau and Spearman rank correlation tests, to see how well a human performs against the humans; to be fair, we computed again the correlation coefficients between our best performing method, and the same means of $N_s - 1$ subjects. In short, we are competing with one human in being close to the judgements of other humans.

When we perform this test for the Linear Combination, Additive Approach with weights $w_T = 0.8$ and $w_C = 0.2$, that correspond to the highest Spearman rank correlation, we get that the mean value of correlations of human judgements against one human is 0.794151, while mean value of correlations against our metric is 0.660734. This confirms the good performance of our texture image similarity metric.

4.4. Discussion

We have shown that our texture image similarity algorithm performs well and agrees with human perception. Our best performing ranking results are shown in Figure 4.6.

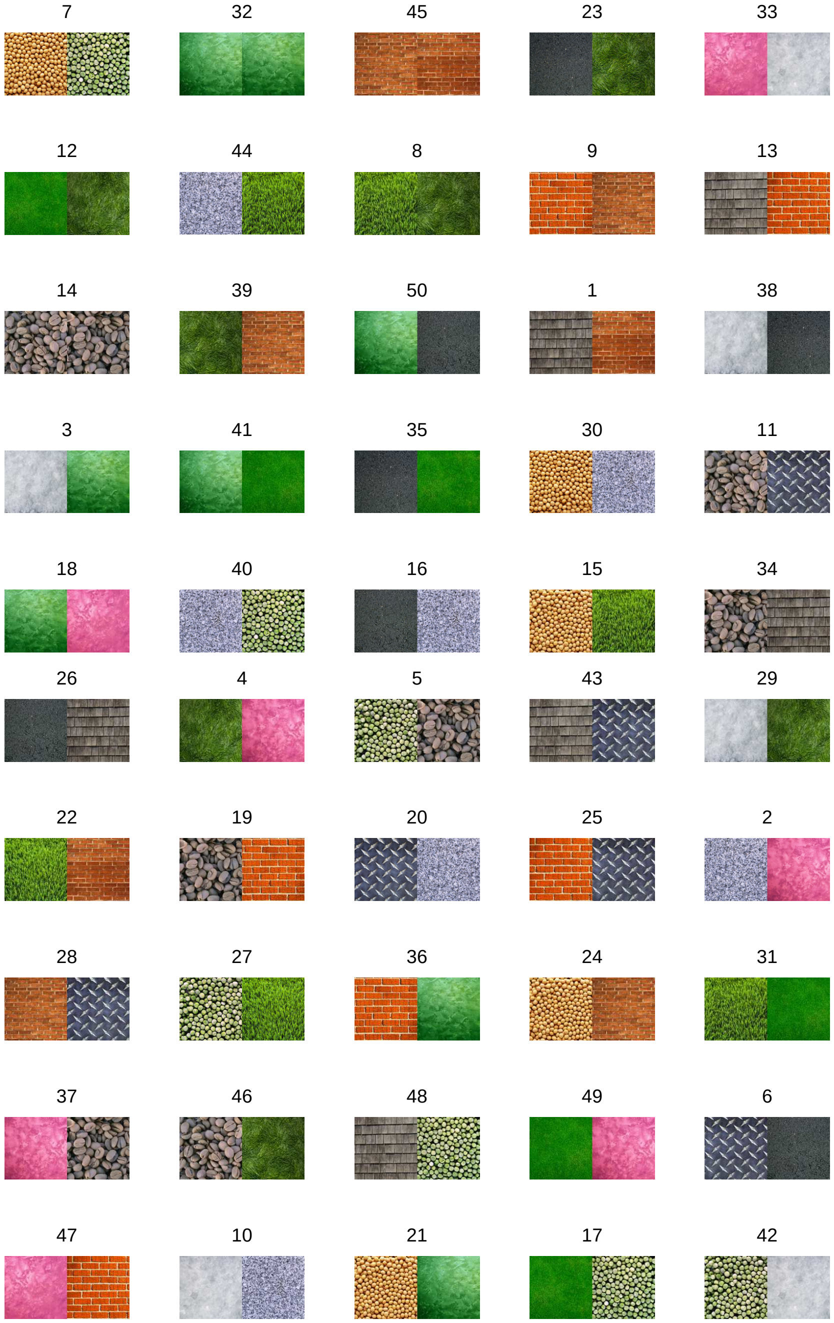


Figure 4.6. Texture pairs sorted in descending similarity order according to our metric

It is apparent that in our approach, the texture information dominates the results. For the number 1 ranked texture, pair number 7, our method feels so strongly about its similarity that even the different color compositions did not move it to a lower ranked spot. Some points at which our method fails is for example texture pair number 23. Its color and gray-scale versions are given in Figures 4.7 and 4.8. Our metric ranks it at the 4^{th} place, while humans put it at rank 28. We assume that this is due to the context association that humans inevitably exhibit, since they *know* they are comparing two different materials. Also, these two textures are very similar at lower scales, thus our metric ranks it high.

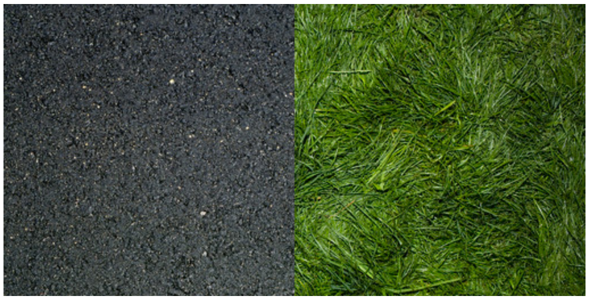


Figure 4.7. Point of failure: Texture pair no. 23

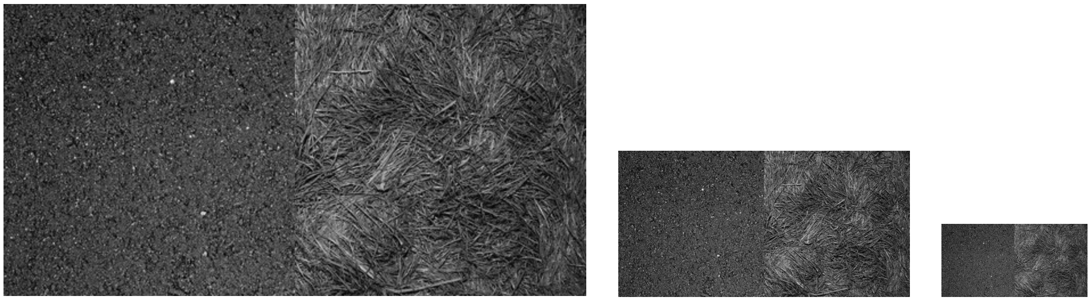


Figure 4.8. Point of failure: Texture pair no. 23, gray-level, at different scales

4.5. Additional Experiments

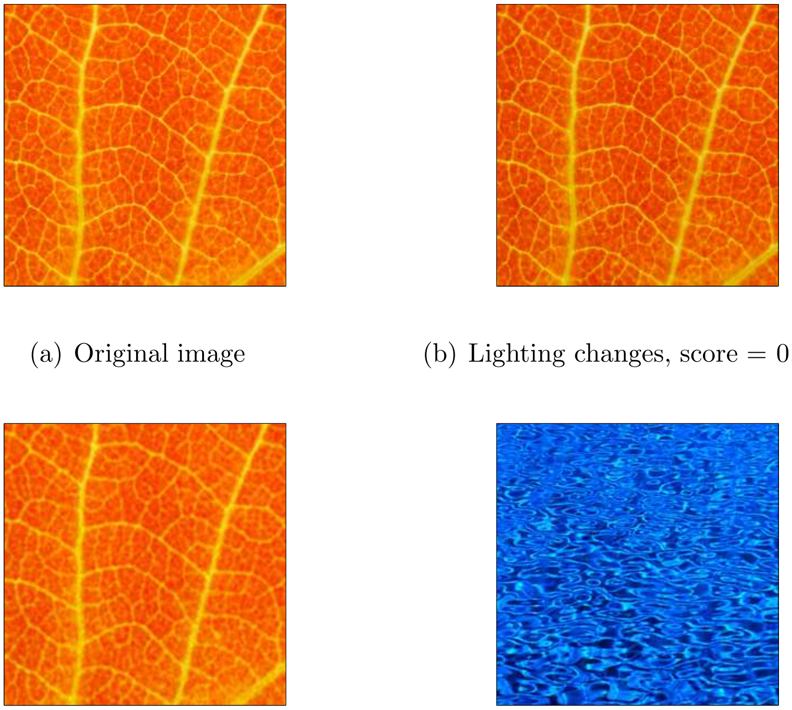
We also tested our metrics on other images and databases, and the experiments confirm good performance of the new proposed metric.

4.5.1. Evaluating the Leaf Example

In Section 2.1, we gave an example of three images of the same leaf, the original and two distorted ones in Figure 2.1.1. We ran our algorithm and compared the original image (2.1(a)) with DCT compressed (2.1(b)) and with image with lighting changes (2.1(c)). We used the method that was selected as the best in the previous experiment (all new similarity maps, additive method for combining grayscale results, linear combination with weights $w_T = 0.6$ and $w_C = 0.4$ for combining structure and color similarity). We also compared the original leaf image with an image of water. We can see that our metric ranked the images according to human perception, giving it highest score for the image with luminance changes, then a bit lower (but still high) score for the DCT compressed image, and a relatively low score for the water image. The images and their respective scores are given in Figure 4.9.

4.5.2. Evaluating the Zhao_Reyes Database

In [**28**], a database of 39 texture pairs was used; they are grayscale, thus we can't include the color information in this section. Again, we varied the subset of subband correlations, and the results for both Additive and Multiplicative approach are given in Tables 4.10 and 4.11. We can see that the performance now is better if we use the subset 3 (adjacent subbands for a given orientation). If we plot this optimal combination of parameters against ST-SIM, we can see in Figure 4.10 that with the new metric, there can be drawn a line in horizontal direction (dash-dot line) that separates the values of the metric for the similar and dissimilar pairs (separated by a vertical dashed line).



(a) Original image

(b) Lighting changes, score = 0.997

(c) DCT compressed, score = 0.926

(d) Water, score = 0.783

Figure 4.9. Leaf example similarity scores

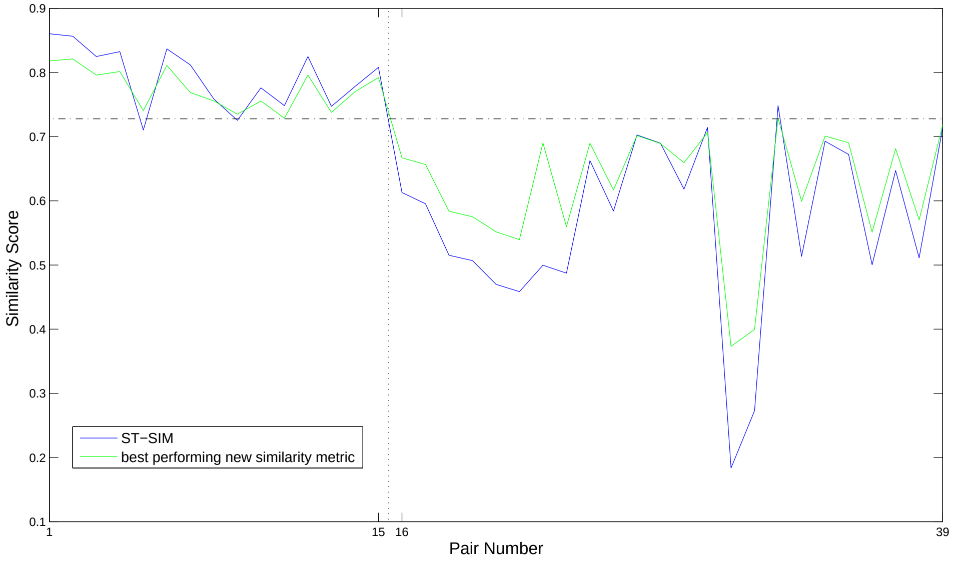


Figure 4.10. ST-SIM and new similarity metric for Zhao_Reyes database

Table 4.10. Zhao_Reyes database, Additive Approach

Bands	ρ_K	$p_{val,K}$	ρ_S	$p_{val,S}$
1	0.4581	4.3260e-005	0.6450	9.3127e-006
2	0.4554	4.8003e-005	0.6462	8.8458e-006
3	**0.4878**	**1.3276e-005**	**0.6881**	**1.2918e-006**
4	0.4554	4.8003e-005	0.6389	1.2004e-005
5	0.4662	3.1556e-005	0.6612	4.6022e-006
6	0.4662	3.1556e-005	0.6689	3.2436e-006
7	0.4662	3.1556e-005	0.6687	3.2740e-006
8	0.4635	3.5075e-005	0.6636	4.1253e-006

Table 4.11. Zhao_Reyes database, Multiplicative Approach

Bands	ρ_K	$p_{val,K}$	ρ_S	$p_{val,S}$
1	0.4527	5.3236e-005	0.6401	1.1415e-005
2	0.4689	2.8375e-005	0.6626	4.3182e-006
3	**0.4716**	**2.5500e-005**	**0.6665**	**3.6264e-006**
4	0.4500	5.9006e-005	0.6341	1.4649e-005
5	0.4608	3.8964e-005	0.6472	8.4732e-006
6	0.4608	3.8964e-005	0.6493	7.7707e-006
7	0.4527	5.3236e-005	0.6497	7.6367e-006
8	0.4554	4.8003e-005	0.6519	6.9372e-006

CHAPTER 5

Conclusion and Future Work

In this thesis, a new method to determine texture image similarity is presented. It combines the best features from the literature that are utilized for determining similarity of gray-level textured regions, and for comparing color compositions of images. The contribution of this work is multifold:

- We have improved the existing texture similarity algorithms by adding terms describing the cross-correlations between subbands of images; this finds justification in human perception and "visual features" that tend to have spatial, scale and orientation correlations;
- We have extended the existing OCCD color comparison method to sliding windows, which leads to greater complexity but achieves better results
- We have explored new methods of combining texture and color information.

This work will be extended in near future to include more subjective testing, with better systematically designed tests and with more examples. Also, the method can be extended to general image comparison. We will explore image comparisons at different scales, since we had encountered a few examples where the textures would be similar at one scale, but dissimilar at another.

References

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

[2] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

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

[4] Z. Wang, A. C. Bovik, H. R. Sheikh, S. Member, and E. P. Simoncelli, "Image quality assessment: from error measurement to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, 2004.

[5] 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.

[6] B. Girod, "What's wrong with mean-squared error?" pp. 207–220, 1993.

[7] Z. Wang, A. C. Bovik, and L. Lu, "Why is image quality assessment so difficult?" vol. 4, 2002, pp. 3313–3316.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] B. S. Manjunath and W. Y. Ma, “Texture features for browsing and retrieval of image data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, 1996.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] J. Mathiassen, A. Skavhaug, and K. B, “Texture similarity measure using kullback-leibler divergence between gamma distributions,” *Computer Vision ECCV 2002*, pp. 19–49, 2002.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.

[25] Z. Wang and E. Simoncelli, "Translation insensitive image similarity in complex wavelet domain," vol. 2, 2005, pp. 573–576.

[26] 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.

[27] A. C. Brooks and T. N. Pappas, "Structural similarity quality metrics in a coding context: exploring the space of realistic distortions," vol. 6057, no. 1. SPIE, 2006, p. 60570U.

[28] X. Zhao, M. G. Reyes, T. N. Pappas, and D. L. Neuhoff, "Structural texture similarity metrics for retrieval applications," *IEEE International Conference on Image Processing, 2008. ICIP 2008., to appear in*, 2008.

[29] 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.

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

[31] 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.

[32] A. Mojsilovic, H. Hu, and E. Soljanin, "Extraction of perceptually important colors and similarity measurement for image matching, retrieval and analysis," *IEEE Transactions on Image Processing*, vol. 11, no. 11, pp. 1238–1248, 2002.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

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

[38] 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.

[39] 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.

[40] 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.

[41] 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.

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

[43] 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.

[44] I. Markov and N. Vassilieva, “Image retrieval: Color and texture combining based on query-image,” 2008, pp. 430–438.

[45] 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.

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

[47] 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.

[48] “Corbis database.” [Online]. Available: www.corbis.com

[49] “Kendall tau rank correlation coefficient.” [Online]. Available: http://en.wikipedia.org/wiki/Kendall_tau_rank_correlation_coefficient

[50] “Spearman’s rank correlation coefficient.” [Online]. Available: http://en.wikipedia.org/wiki/Spearman’s_rank_correlation_coefficient

[51] H. R. Sheikh, Z. Wang, A. C. Bovik, and L. K. Cormack, “Image and video quality assessment research at live.” [Online]. Available: http://live.ece.utexas.edu/research/quality/