Toward Perception-Based Image Retrieval

Edward Y. Chang and Beitao Li
Electrical & Computer Engineering
UC Santa Barbara
echang@ece.ucsb.edu

Chen Li
Department of Computer Science
Stanford University
chenli@cs.stanford.edu

Abstract

Since a content-based image retrieval (CBIR) system services people, its image characterization and similarity measure must closely follow perceptual characteristics. In this study, we enumerate a few psychological and physiological invariants and show how they can be considered by a CBIR system. We propose distance functions to measure perceptual similarity for color, shape, and spatial distribution. In addition, we believe that an image search engine should model after our visual system, which adjusts to the environment and adapts to the visual goals. We show that we can decompose our visual front-end into filters of different functions and resolutions. A pipeline of filters can be dynamically constructed to meet the requirement of a search task and to adapt to individuals' search objectives.

Keywords: *content-based image retrieval, perception-based image retrieval, personalization, relevance feedback, triangle similarity.*

1 Introduction

Much research has been conducted on content-based image retrieval (CBIR) in the past decade [12]. A content-based image retrieval system returns images that are "similar" to a query image. To measure similarity, most image database systems characterize images using perceptual features (e.g., color, shape and texture) and define how similarity can be quantified using these features.

Not many attempts [10], however, have been made to characterize images and to quantify similarity based on the characteristics of the human visual process in the CBIR community. For instance, while most traditional systems model our response to "brightness" as a linear function, human eyes respond to brightness in a non-linear fashion. In addition, most traditional systems treat color as a continuous spectrum of wavelength, while we give simple names to only a limited number of colors (red, green, etc). Also, most systems treat all pixels in an image equally, but our vision tends to pay less attention to the boarder pixels, and it distributes effort unevenly by paying closer attention to ambiguous regions. Many other discrepancies exist.

Moreover, our visual system adjusts to the environment and adapts to the visual goals [13]. We can think of our visual system as being divided into two parts: our eyes (the front-end) perceive and our brain (the back-end that is equipped with a knowledge database and an inference engine) recognizes images. The front-end collects visual data for the back-end to allow high-level processing. The back-end instructs the front-end to collect visual data with different *filters*. (A filter can be regarded as a particular way of perceiving an image.) The front-end responds flexibly in perceiving visual data by selecting, ordering and adjusting visual filters differently. For instance, we may not be able to tell if a figure is a real person or a statue at first glance. If we pay closer attention to the figure's surface, we may be able to identify it as a person or statue. The front-end and back-end may interact many times to complete a visual task.

We believe that since an image search engine services people, it should be modeled after our visual system. In this paper, we propose a perception-based image retrieval architecture that can dynamically construct a pipeline of filters to meet the requirement of a search task and to adapt to individuals' search objectives. We first focus on designing individual image filters (e.g., color masks, pixel masks, shape filters, etc.). We investigate how some of the psychological and physiological invariants mentioned above affect our perception and how they can be considered in characterizing visual data and in measuring similarity for a filter. Once filters are designed, we show how they can be selected and adjusted to support queries that have different goals both effectively and efficiently.

The contributions of this study are as follows:

- We enumerate some of our perceptual invariants and show how to apply them in image representation (Sections 2 and 3).
- We propose distance functions, e.g., Hamming, Gaussian and Triangle functions, to measure similarity for color and spatial distribution of colors (Section 3).
- We suggest to decompose our visual front-end into filters and show how filters can be selected and adjusted to support personalizable queries (Section 4).

2 Perceptual Characteristics

Many researchers in content-based image retrieval have assumed that the perceptual color space is a Euclidean metric space. Based on this assumption, clustering and indexing schemes that use Euclidean distance to measure similarity are employed. In addition, most studies treat all pixels equally in characterizing an image. We first describe two perceptual characteristics to argue that using Euclidean distance may not be appropriate in all circumstances and suggest alternatives (Sections 2.1 and 2.2). We then show that not all pixels in an image attract equal perceptual attention (Section 2.3). In Section 3, we apply these perceptual characteristics to image filter design.

2.1 JND and JNS

The *Just Noticeable Difference* (JND) is the smallest difference between two stimuli that a person can detect. Goldstein [4] uses the following example to illustrate the JND: A person can detect the difference between a 100-gram weight and a 105-gram weight but cannot detect a smaller difference, so the JND for this person is 5 grams. For our purpose, we introduce a new term, called *Just Not the Same* (JNS). Using the same weight example, we may say that a 100-gram weight is just not the same as a weight that is more than 120-gram. So the JNS is 20 grams. When the weight is between 105 and 120 grams, we say that the weight is similar to a 100-gram weight to a degree.

Now, let us apply JND and JNS to our color perception. We can hardly tell the difference between *deep sky blue* (whose RGB is 0,191,255) and *dodger blue* (whose RGB is 30,144,255). The perceptual difference between these two colors is below JND. On the other hand, we can tell that blue is different from green and yellow is different from red. In both cases, the colors are perceived as JNS.

For an image search engine, JND and JNS indicate that using Euclidean distance for measuring color difference may not be appropriate. First, JND reveals that when the difference between two colors exists but is small enough, then the two different colors are perceived the same. Second, JNS reveals that when the difference is significant, we say two colors are not the same. Taking both JND and JNS into consideration, we thus quantify color similarity in three discrete regions:

- When the color difference is above JNS: the similarity is defined zero. (We discuss JNS colors below.)
- When the color difference is between JND and JNS: the similarity is between zero and 100%.
- When the color difference is below JND: the similarity is 100%.

2.1.1 JNS Colors

We define a term *JNS colors* in what follows. Although the wavelength of visible light is 400 meters to 700 meters, research [5] shows that the colors that can be named by all cultures are limited to eleven. In addition to *black* and *white*, the discernible colors are *red*, *yellow*, *green*, *blue*, *brown*, *purple*, *pink*, *orange* and *gray*. Since it is difficult to quantify the difference between two JNS colors, we simply define the similarity between two JNS colors zero.

2.1.2 Benefits

Quantifying color similarity in three regions gives us the flexibility to design color filters of different resolutions. We describe color masks, color histograms, color-average and color-variance filters in Section 3. Picking different combinations of filters can accomplish different search objectives (examples are shown in Section 5).

2.2 Response Compression

The second important characteristic of our vision is our response to "brightness." When we double the intensity of a light, the light does not look twice as bright. According to Stevens's experiment in 1957, doubling the light intensity causes only a small change in perceived brightness. Figure 1 shows that the magnitude of response decreases when the stimulus intensity increases. Stevens's power law states that the perceived magnitude, P, equals a constant, K, times the stimulus intensity, S, raised to a power, n, or

$$P = K \times S^n. \tag{1}$$

The exponent for brightness is about 0.6.

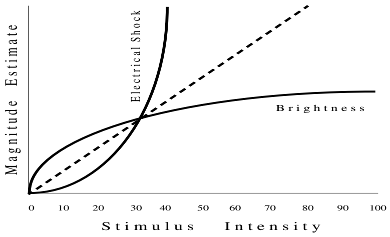

Figure 1. Stevens's Power Law.

According to Stevens's power law (shown in Figure 1), the visual sensitivity is a non-linear function with respect to the intensity of the stimulus. Using this law as our starting point, we first transfer colors from RGB space to a component color space with intensity and perceived contrasts [2, 14]. We define the new values of a color pixel based on the RGB values of an original pixel as follows:

$$\begin{cases} S = (R+G+B)/3 \\ C_2 = (R+(max-B))/2 \\ C_3 = (R+2*(max-G)+B)/4 \end{cases} \tag{2}$$

Here max is the maximum possible value for each color component in the RGB color space. For a standard 24-bit color image, $max = 255$. The intensity is isolated in S.

Once we isolate S, we can characterize intensity using the perceived magnitude, P, which can be obtained by plugging S into the right-hand side of Equation 1. We also normalize the values of P, C_2 and C_3 to be between 0 and 1. When two colors belonging to the same culture color are compared, we use this normalized Euclidean distance in P, C_2 and C_3 space to quantify their similarity.

2.2.1 Benefits

Isolating intensity from colors can achieve two benefits. First, we can normalize intensity so that we can characterize color similarity more accurately. Second, we can omit the intensity factor if we want to remove the effect of light sources from measuring image similarity. One can design two filters, one takes and one does not take light sources into consideration. A search task employs an appropriate filter for its goal.

2.3 Pixel Location

Not all pixels in an image are equally important to our perception. For instance, since we tend to look at the middle of an image, the pixels there tend to fire more neurons in the part of the brain devoted to perception than those at the boarder do. (One reason we tend to look at the middle of an image is that we tend to place important objects in the middle when we take a picture.) To take this factor into consideration, we can weight the pixels based on their location.

2.3.1 Benefits

Weighting pixels can remove the information that is not essential for similarity matching. For example, discounting the pixels on the image boarder is helpful for finding similar images that have been framed. Multi-resolution image analysis can also benefit from pixel weighting. For example, we may weight the background pixels lighter and the foreground ones heavier.

3 Image Filters

There could be thousands of useful image filters. To illustrate our concept of composing filters for a visual task, we enumerate a few color-based image filters in this section. We consider the perceptual characteristics described in Section 2 in our filter design. We treat shape and texture as attributes of a color set. (One can build filters by treating color and texture as attributes of a segmented shape or color and shape as attributes of a texture, e.g., [9].) We show in Sections 4 and 5 that selecting different filters can support queries that have different objectives.

To characterize an image, we divide it into *color sets*. A color set is the collection of pixels in an image that have the same color. We divide the visible color space into eleven JNS colors (discussed in Section 2.1). An image thus consists of at most eleven color sets, where a color set is characterized by three attributes of the image: its color, moments and centroid. These attributes are defined as follows:

- Color: The culture color of the pixel collection. Each culture color spans a range of wavelengths. We keep the mean and variance of the color set on the three axes, P, C_2 and C_3. We also record the number of pixels belonging to the color set.
- Moments: The moment invariants [6] of the color set. Moment invariants are properties of connected regions in binary images that are invariant to translation, rotation and scaling.
- Centroid: The center of gravity of the color set, $\hat{x}$ and $\hat{y}$.

Next, we describe how these attributes can be used to build image filters and how each filter measures similarity. Table 1 summarizes the filters discussed in this section. We roughly divide the filters into coarse- and fine-resolution ones. One may want to use a coarse filter at some occasions and fine filters at others. For instance, if one wants to search wall-papers of a given pattern, using color masks of different Hamming distance can be a good choice.

Filter Name	Representation
Color Masks	Number of identical JNS colors
Color Histograms	Distribution of colors
Color Average	Similarity comparison within the same JNS color
Color Variance	Similarity comparison within the same JNS color
Spread	Spatial concentration of a color
Elongation	Shape of a color
Spatial Relationship	Spatial relationship between major color sets

Table 1. Filter Summary.

3.1 Color Masks

Each image has an 11-bit color mask, and each bit represents a culture color. If a culture color is present, the corresponding bit in the mask is set. To filter out noise, we say that a color is present only if it covers at least α percent (e.g., one percent) of the pixels.

To compare two color masks (of two different images), we employ the *Hamming distance*. The Hamming distance of two color masks is the number of colors by which the two masks differ. For instance, the hamming distance between $\{black, red, blue\}$ and $\{green, red, blue\}$ is two. Let $H_{A,B}$ denote the hamming distance between color masks M_A and M_B, or $H_{A,B} = M_A \oplus M_B$.

The color masks plus the percentage of pixels each present color covers form a color histogram. To compare the difference between two color histograms, we can employ traditional measures proposed by [3, 11].

3.2 Refined Color Matcher

Suppose two color sets belong to the same JNS color (e.g., navy blue and sky blue). We perform a refined comparison step to see how similar the colors are. Traditionally, the similarity measure of two colors is done by comparing the Euclidean distance between their P, C_2 and C_3 (or HVS etc.) averages. (We call this an color-average filter.) We believe that employing color variance in the similarity measure adds the texture dimension to the consideration. For instance, the color of a building is typically uniformly distributed along the C_2 and C_3 axes. (The value of P may vary due to the lighting condition.) A JNS color on a scenic image is usually perceptually non-uniform. For instance, a green pasture may have green of different saturation and intensity.

The distribution of a color set can be characterized as a Gaussian function[1]. To measure similarity based on the mean and variance of a culture color, we compare the Gaussian functions represented by two sets of the mean and the variance in C_2 and C_3. We define similarity as the intersecting area of the two Gaussian functions. Given two normalized Gaussian

---

[1] The central limit theorem states that the sum of a large number of independent observations from the same distribution has, under certain general conditions, an approximate Gaussian distribution. One can assume that the pixels in a color set are samples of some objects, which colors follow some unknown distribution. A reasonable large number of samples of the objects thus follow a Gaussian distribution.

functions $f_1(x)$ and $f_2(x)$ and their means ($\bar{x}_1$ and $\bar{x}_2$) and variances (σ_1 and σ_2), the statistics community has shown that the following characteristic distance can be used to depict the difference between $f_1(x)$ and $f_2(x)$ [8]: $\Psi = \frac{|\bar{x}_1 - \bar{x}_2|}{\sqrt{\sigma_1 \sigma_2}}$. Notice that the larger the variances, the harder it is to distinguish one Gaussian from the other.

3.3 Pixel Masks

A pixel mask weights the pixels on an image by their location. We use $w_{i,j}$ to denote the weight of the pixel at the i, j coordinate. The value of $w_{i,j}$ is between zero and one.

3.4 Spatial Relationship: Triangle Filter

For some image applications, capturing the spatial relationship between the dominant objects can be helpful. A sunset picture, for example, can be roughly depicted as an orange circle above a dark horizon. A firework show can be characterized as sparks of colors on a black background. A tree should be beneath a blue or gray sky and above the land.

We propose using a triangle to capture the spatial relationship between the major pixel sets on an image. We use triangles for two reasons. First, the similarity of two triangles immunes from many geometrical transformations such as translation, rotation and scaling. Second, any geometrical shape can be composed of triangles, so triangle filters provide scalability for capturing spatial relationship. If we want to capture finer spatial relationship between more colors, we can form more triangles.

The three nodes of the triangle are the centroids of the three selected pixel sets. When comparing the similarity of two triangles, we first mark the three vertices of one triangle as ABC, then we mark the vertex of the other triangle with the same features (color, shape, texture or any combination of the three) as vertex A as A', the vertex of the same features as B as B', and the vertex of the same color as C as C'. Then we stretch or shrunk the triangle A'B'C' so that $\overline{A'B'} = \overline{AB}$.

Next, we put these two triangles together so that A' is on A, B' is on B as shown in Figure 2. We calculate the ratio of the overlapping area of these two triangles to the average area of them, which gives a good idea about how similar these two triangles are.

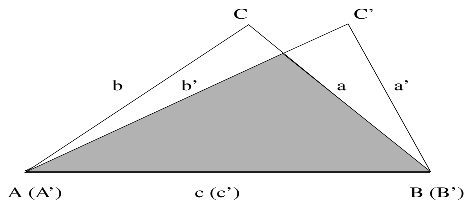

Figure 2. Triangle Similarity.

3.5 Shapes

Moment invariants [6] are properties of connected regions in binary images that are invariant to translation, rotation and scaling. They are useful because they define a simply calculated set of properties that can be used for shape classification and part recognition inside a region. One can identify basic shapes by examining the first moment of a pixel collection. For example, many experiments have confirmed that the first moment of a circle, regardless of its size and location on the image is a constant. In addition, several high-level shape characteristics can be derived from low-level moments. The study of [7] defines "spreadness" (which we call spread) as a normalized average distance of pixels to the center of the shape. It shows how much the shape is spread. Another important index is elongation, which reflects how much the shape is elongated. Moment invariants have been studied in many papers and we discuss the subject briefly here so that we could use the spread and the elongation filter in our experiments.

4 Filter Pipeline Model

Suppose an image set Ω consists of M images, denoted as y_i, where $i = 1, 2, \ldots, M$. Let y_q be the query image. A top-K query finds the K most *similar* images to y_q in Ω.

Suppose each image is depicted by N features, denoted as x_j, where $j = 1, 2, \ldots, N$. For feature x_j, we implement a filter F_j for screening out unwanted images to narrow down the search space. Let $S_j(y_i, y_q)$ be the function that measures the similarity between image y_i and the query image y_q for feature x_j and T_j the filter threshold. We can express filter F_j quantitatively as

$$F_j(\Omega, T_j) = \{y_i | S_j(y_i, y_q) \geq T_j \ and \ y_i \in \Omega\}. \qquad (3)$$

A filter selects images which similarity to the query image is above the filter threshold.

The strength of this filter-pipeline model lies in three areas: its expressiveness, efficiency and flexibility.

- Expressiveness: Filters can be combined using boolean operators such as and, or, $nand$, nor, etc. For example, a traditional linear model that combines κ filters can be expressed as

  $$\omega = F_1(F_2(...F_\kappa(\Omega, T_\kappa)..., T_2), T_1).$$

  It is obvious that being able to use boolean operators to combine filters improves the expressiveness drastically beyond the traditional linear model.
- Efficiency: Filters can be ordered judiciously to improve search speed. Many traditional query optimization techniques (e.g., index join, hash join, sort join, etc.) can be employed to minimize search space, reduce intermediate memory use, and parallelize query processing.
- Flexibility: Different filters can be selected based on different search goals. More importantly, a filter's threshold can be adjusted to make the filter coarser or finer for supporting relevance feedback and personalization. Having a coarser threshold (lower T_j) makes a filter less selective; having a finer threshold (higher T_j) makes a filter more scrupulous.

5 Preliminary Results

To achieve personalizable queries, a query processor should be adaptive to user preferences and relevance feedback. When a user expresses that some returned results are not satisfactory, the query processor learns from the feedback and then adds, removes or replaces filters, or changes filter thresholds to achieve better results. Because of the space limitation, we only show three query examples of progressive refinement. (For additional experimental results, please refer to [1].) Our experiments were conducted on a collection of $500,000$ randomly crawled Web images. In the figures where the results are presented, the query image is on the left-hand side of the figure and the top five similar images are presented next to the query image.

Figure 4. Color- and Shape-Based Query.

Figure 4 shows the results of queries on a dress. First, we used color histograms and color average as the filters. The result presented in the first row returns also a pair of shoes. Apparently, the shape of the dress is different from that of the shoes. We thus added shape filters to the pipeline and the result in the second row eliminates the shoes. (If one wants to search for shoes that can go with the dress, the first pipeline is the desired one.)

6 Conclusions and Future Work

In this study we enumerate several perceptual characteristics and show how to model and quantify them. We also show that our visual system perceives images by combining filters of different functions and resolutions. By decomposing our visual system into filters, we can dynamically construct a high-level pipeline of filters to perform a specific visual task, e.g., categorizing an image or identifying an object in an image. By considering a user's relevance feedback, this technique of dynamic visual pipeline construction can support a customizable/personalizable image search engine that can be tailored to meet the goal of a particular visual task. Our research will continue to explore perception-based image characterization and to formalize the principles and mechanisms for constructing a visual pipeline and for measuring its cost and performance (e.g., recall and precision).

References

[1] E. Chang, B. Li, and C. Li. Toward perception-based image retrieval (extended version). *UCSB Technical Report*, February 2000.

[2] E. Chang, J. Wang, C. Li, and G. Wiederhold. RIME - a replicated image detector for the www. *Proc. of SPIE Symposium of Voice, Video, and Data Communications*, November 1998.

[3] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, and et al. Query by image and video content: the QBIC system. *IEEE Computer*, 28(9):23–32, 1995.

[4] E. Goldstein and S. Fink. Selective attention in vision. *Journal of Experimental Psychology*, 7, 1981.

[5] E. B. Goldstein. *Sensation and Perception (5^{th} Edition)*. Brooks/Cole, 1999.

[6] L. M. Hurvich and D. Jameson. An opponent-process theory of color vision. *Psychological Review*, 64:384–90, 1957.

[7] J.-G. Leu. Computing a shape's moments from its boundary. *Pattern Recognition*, pages Vol.24, No.10,pp.949–957, 1991.

[8] C. Li. Introduction to experimental statistics. *McGraw-Hill Book Company*, pages 403–413, 1964.

[9] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, August 1996.

[10] T. V. Papathomas, T. E. Conway, I. J. Cox, J. Ghosn, M. L. Miller, T. P. Minka, and P. N. Yianilos. Psychophysical studies of the performance of an image database retrieval system. *IS&T/SPIE Conf. on Human Vision and Elec. Imaging III*, 1998.

[11] Y. Rubner, C. Tomasi, and L. Guibas. Adaptive color-image embedding for database navigation. *Proceedings of the the Asian Conference on Computer Vision*, January 1998.

[12] Y. Rui, T. S. Huang, and S.-F. Chang. Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, March 1999.

[13] B. Wandell. *Foundations of Vision*. Sinauer, 1995.

[14] J. Z. Wang, G. Wiederhold, O. Firschein, and S. X. Wei. Content-based image indexing and searching using daubechies' wavelets. *Journal of Digital Libraries*, 1(4):311–28, 1998.