# Image Indexing and Retrieval Based on Human Perceptual Color Clustering*

Yihong Gong, Guido Proietti† and Christos Faloutsos‡

Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
{ygong, proietti, christos}@cs.cmu.edu

## Abstract

*We propose a new image retrieval method based on human perceptual clustering of color images. This color clustering produces for each image a small set of representative colors which captures the color properties of the image, and a small set of sizable contiguous regions which captures the spatial/geometrical properties of the image. The proposed method outperforms the traditional histogram and its improved methods not only with its richer image retrieval capabilities which cover a wider spectrum of user requirements, but also with its powerful indexing scheme which is essential to cater for large scale image databases.*

## 1 Introduction

The rapid progress of multimedia computing and applications has brought about an explosive growth of digital images in computer systems and networks. This development has remarkably increased the need for image retrieval systems that are able to effectively index a large amount of images and to efficiently retrieve them based on their visual contents.

In developing a visual content-based image retrieval system, the first critical decision to be made is to determine what image feature, or combination of image features are to be used for image indexing and retrieval purposes. Image feature selection is critical because it largely affects the remaining aspects of the system design, and greatly determines image retrieval capabilities of the eventual system.

Color histograms are commonly used as image features by many image retrieval systems. Color histograms are easy to compute, and have few constraints when applied to images; hence, the task of feature capturing is very simple. However, color histograms have many inherent problems in indexing and retrieving images. First, they are big in size — common color histograms consist of from 64 to 256 bins. This large histogram size makes it rather difficult to create an effective database indexing scheme. Second, they do not include any spatial information; hence are prone to false positives. Third, they are sensitive to small brightness changes, and therefore are liable to false negatives as well. Finally, they are basically incompetent to support partial matching of image contents. Partial matching is essential to many image retrieval requests. User queries such as retrieving images that contain a green lawn while ignoring the rest part of the images can not be accommodated without partial image matching abilities.

Recently research studies have been made to improve the problems associated with color histograms. Some approaches exploit derived histogram features to facilitate the creation of an effective database indexing scheme, while others strive to incorporate spatial information into color histograms to increase image discrimination powers. Most of them make improvements on one or two aspects of color histograms, while leaving the rest of the problems untouched.

In this paper, we propose a new image retrieval method based on human perceptual clustering of color images. The features of the proposed method include: (1) it enables partial matching among images; (2) it captures both the global color distribution and the prominent regions in the images, so that users are able to form database queries using both the color and shape properties of images; (3) it provides an effective database indexing scheme, so that no round-robin matching is required to retrieve desired images; (4) the speed of image retrieval is fast, and is insensitive to the increase of database size. This method outperforms the traditional histogram and its improved methods not only with its richer image retrieval abil-

*This work was sponsored by the National Science Foundation under grant No. IRI-9411299, the National Space and Aeronautics Administration, and the Advanced Research Projects Agency.

†On leave from University of L'Aquila, Italy.

‡On leave from University of Maryland.

ity that covers a wider spectrum of user requirements, but also with its powerful indexing scheme which is essential to cater to large scale image databases.

## 2 Related Work

Recently, several approaches have been proposed to overcome the problems associated with the traditional histogram method. Stricker and Dimai [1] divided an image into five fixed overlapping blocks and extracted the first three color moments of each block to form a feature vector for the image. The use of color moments largely reduces the size of the feature vector and simplifies the creation of an effective database indexing scheme. The image division and feature extraction in local blocks roughly brings spatial information about colors into the feature vector. However, matching images based on a small number of derived color features could further decrease image discrimination powers of the system, and dividing an image into five fixed regions is not sufficient to represent spatial properties of the image.

Pass and Zabih [2] incorporated the spatial coherence feature into color histograms. A pixel is considered coherent if it is a part of some sizable contiguous regions, and incoherent otherwise. Based on this definition, pixels in each histogram bin are partitioned into either the coherent class or the incoherent class, and histogram matching is performed by comparing the corresponding classes in the corresponding bins.

Huang, et al. [3] proposed the correlogram to take into account the local color spatial correlation as well as the global distribution of this spatial correlation. Since a complete color spatial correlation matrix requires a larger storage than for a color histogram, the authors used the auto-correlogram instead, where each bin $i$ gives the probability that two pixels with distance $k$ have color value $i$.

Both the color coherence method and the correlogram method have reported a remarkable improvement on the image retrieval accuracy compared to traditional histogram-based method. However, the database indexing problem remains unsolved, and available means of retrieving desired images are still very limited.

Contrary to the above approaches, our method captures both the color and the spatial/geometrical properties of an image using color clustering based on human color perceptions. These two kinds of image features are used to form indexes of each image in the database. Image retrieval using a sample image is performed by a search for the feature sets that are the nearest to the given reference feature set in the feature space, which can be performed fast and efficiently on a very large image database. Moreover, our method is very effective to support partial matching of image contents, and to absorb large appearance changes caused by zooming, panning, brightness changes, etc.

## 3 Perceptual Clustering of Color Images

For a retrieval system to be successful, images must be represented by features that capture the major image properties and meanwhile remain compact in size. Traditional color histograms are created by equally subdividing a color space (e.g. the RGB color space) into a number of small bins, and then counting the number of pixels each bin contains. However, as most images do not possess a uniform color distribution, a typical histogram usually contains many empty or nearly empty bins. Retrieving images by matching histograms of a large size containing many empty bins is neither productive nor efficient. With a very large image database, this waste could become prohibitive. Therefore, by no means are histograms a good representation of color images.

To solve the above problem, we propose to adaptively cluster color images based on human color perceptions. A 24-bit color image contains up to 16 million colors. Most of the colors can not be differentiated by human beings, because human eyes are relatively less sensitive to colors. Research [4] has found that in the HVC color space, the space that represents colors along human color perceptual dimensions, colors with the NBS color distance below 3.0 are perceived to be almost the same color by human beings. The NBS color distance is devised through a number of subjective color evaluation experiments to better approximate human color perception. Given a pair of colors $A = (H_1, V_1, C_1)$ and $B = (H_2, V_2, C_2)$, the NBS color distance $E_{NBS}$ is defined as follows:

$$E_{NBS}(A,B) = 1.2 \cdot \sqrt{2C_1C_2\left\{1-\cos\left(\frac{2\pi}{100}\Delta H\right)\right\} + (\Delta C)^2 + (4\Delta V)^2} \quad (1)$$

where $\Delta H = |H_1 - H_2|$, $\Delta V = |V_1 - V_2|$ and $\Delta C = |C_1 - C_2|$.

There is a close relation between the human color perception and the NBS color distance, which is shown in Table 1.

Taking advantage of the above research studies, we transform the input image from RGB to HVC, cluster the image in the HVC color space by merging perceptually indistinguishable colors together, and separating colors with remarkable differences to form different clusters. This clustering method makes use of

Table 1: The Correspondence between the human color perception and the NBS color distance

| NBS Value | Human Perception |
|---|---|
| $0 \sim 1.5$ | Almost the same |
| $1.5 \sim 3.0$ | Slightly different |
| $3.0 \sim 6.0$ | Remarkably different |
| $6.0 \sim 12.0$ | Very different |
| $12.0 \sim$ | Different color |

three thresholds: the maximum number of initial seeds $T_s$(=20), the minimum NBS color distance between individual seeds $T_d$(=6.0), and the maximum cluster radius that allows the merge operation $T_r$(=3.0). The operation detail is as follows:

**Seed Initialization:**

1. Project the input image into the HVC color space. Initialize the seed set to empty **SEEDS** $= \emptyset$, and set $k = 1$.

2. In the HVC color space, find an unmarked color $\mathbf{Z}_k$ that has the largest pixel count. If **SEEDS** is empty, add $\mathbf{Z}_k$ to **SEEDS**; otherwise, compare $\mathbf{Z}_k$ with each $\mathbf{Z}_j \in$ **SEEDS**, where $j = 1, 2, \ldots, k-1$. If $\mathbf{Z}_k$ satisfies
$$\min_{1 \le j \le k-1} \{E_{NBS}(\mathbf{Z}_k, \mathbf{Z}_j)\} > T_d$$
add $\mathbf{Z}_K$ to **SEEDS**. In the case that $\mathbf{Z}_k$ is added to **SEEDS**, mark $\mathbf{Z}_K$ in the color space, and increment $k$ by 1.

3. If $k > T_s$, or no colors with a non-zero pixel count are left in the color space, go to **Clustering**; otherwise go to Step 2.

**Clustering:**

1. For each seed $\mathbf{Z}_j \in$ **SEEDS**, form a empty cluster $\mathbf{S}_j$ with radius $R(\mathbf{S}_j) = 0.0$.

2. In the HVC color space, for each color $\mathbf{P}_r$ with a non-zero pixel count, find a seed $\mathbf{Z}_x \in$ **SEEDS** that satisfies
$$E_{NBS}(\mathbf{P}_r, \mathbf{Z}_x) = \min_{1 \le j \le k} \{E_{NBS}(\mathbf{P}_r, \mathbf{Z}_j)\}$$
and add $\mathbf{P}_r$ to cluster $S_x$.

3. For each cluster $S_x$, update its seed $\mathbf{Z}_x$ and calculate its radius $R(S_x)$ as follows:
$$\mathbf{Z}_x = \frac{1}{N(S_x)} \sum_{\mathbf{P}_r \in S_x} \mathbf{P}_r \quad (2)$$
$$R^2(S_x) = \frac{1}{N(S_x)} \sum_{\mathbf{P}_r \in S_x} E^2_{NBS}(\mathbf{P}_r, \mathbf{Z}_x) \cdot \pi(\mathbf{P}_r) \quad (3)$$
where $N(S_x)$ is the total number of pixels included in cluster $S_x$, and $\pi(\mathbf{P}_r)$ is the pixel count of color $\mathbf{P}_r$.

4. For each $\mathbf{Z}_i \in$ **SEEDS**, if there is a $\mathbf{Z}_x \in$ **SEEDS** that satisfies
$$E_{NBS}(\mathbf{Z}_i, \mathbf{Z}_x) = \min_{1 \le j \le k} \{E_{NBS}(\mathbf{Z}_i, \mathbf{Z}_j)\} \le T_d \quad \text{and}$$
$$R(S_i) < T_r, \quad R(S_x) < T_r$$
merge the cluster $S_i$ and $S_x$ to form a new cluster $S'_i$ with the seed defined as:
$$\mathbf{Z}'_i = \frac{1}{N(S_i) + N(S_x)} \sum_{\mathbf{P}_r \in S_i, S_x} \mathbf{P}_r$$
Decrement the cluster counter $k$ by 1. Repeat the same operation for all the cluster $S'_i$s.

5. If no merge occurs in Step 4, terminate the operation; otherwise go to Step 2.

The above color clustering operation has the following characteristics: (1) the NBS distance of any pair of cluster centers is no less than 6.0; (2) the final number of clusters is no more than 20, and this number varies depending on the color distribution of the input image; (3) the maximum radius increase occurs when two clusters of radius 3.0 with a center distance 6.0 are merged to form a new cluster (the worst case merging). If Euclidean distance is used to define the cluster radius instead of the NBS color distance in Equ.(3), it can be calculated that the maximum radius obtained from the worst case merging is no more than 4.3 (we have to omit the mathematical derivation because of the space limitation). Since many mergings occur between smaller clusters with a shorter center distance, the resultant radius will be less than this maximum radius. However, with the NBS distance being used in Equ.(3), we are unable to mathematically calculate the maximum radius obtained from the worst case merging. Our experiments have shown that the average cluster radius is around 2.8, with the maximum radius not more than 5.1. This cluster size is acceptable when judged from Table 1.

Another thing to be noted is that the above clustering method has set the number of initial seeds to 20. As all of our image data are obtained by digitizing the CNN TV news broadcasting using an MPEG encoder, the images have a relatively narrow color bandwidth. Our experiments have revealed that 20 initial color seeds are sufficient for most images, and that the majority of the final cluster numbers fall into the range of 5 through 15.

## 4 Database Indexing

The color clustering produces a small set of clusters for each input image. This cluster set collectively represents color properties of the original image. Another effect of the color clustering is that the image is segmented into a set of contiguous regions. These contiguous regions can be easily detected by labelling adjacent pixels with the same cluster color (region labelling). These regions certainly capture the information of the object regions contained in the image.

To exploit both the color and the shape properties of an image, we construct two kinds of indexes — one representing each color cluster and one representing each sizable, contiguous region.

For each color cluster $S_x$, a five-dimensional feature vector is created to include the following features:

1. The average H, V, and C values (three elements).
2. The pixel count normalized by the image size.
3. The cluster aggregation degree $\mathcal{A}$, which is defined as $\mathcal{A} = Pr(\Theta(p) \in S_x | \ p \in S_x)$, where $\Theta(p)$ is the 8-neighbor of pixel $p$. This aggregation degree is a feature that tells whether pixels belonging to cluster $S_x$ form a contiguous region or scatter across the image.

The cluster feature vectors obtained from all the input images are stored as indexes in a SR-Tree $SRT_C$ [5]. SR-Tree is an enhanced version of R-Tree that is very efficient for storing medium and high dimensional vectors, as well as for searching nearest neighbors based on multi-dimensional vectors.

For each region $R_x$ with a size greater than 1/36 of the image, a 24-dimensional feature vector is created as follows:

1. The $X$- and $Y$-axis profiles of the region, which is calculated as (20 elements):
   (a) Find the the minimum bounding box $B$ of the region. Stretch $B$ along its short side into a square in the direction of increasing $X$ or $Y$ value. After this, three sides of $B$ remain touching the region boundary.
   (b) Equally subdivide $B$ along both the $X$ and $Y$ axes into 10 intervals.
   (c) For each cell $(u, v)$ obtained from the above subdivision, calculate the percentage $p(u, v)$ of the region that cell $(u, v)$ contains.
   (d) Define the profile of the region along the $X$-axis as a 10-element array $\Phi_x$ with the $i$-th element $\Phi_x(i) = \sum_{v=1}^{10} p(i, v)$.
   (e) Similarly, define the region profile along the $Y$-axis as a 10-element array $\Phi_y$ with the $j$-th element $\Phi_y(j) = \sum_{u=1}^{10} p(u, j)$.
2. The average H, V, and C values (three elements).
3. The region area normalized by the image size.

In each region feature vector, the $X$- and $Y$-axis region profiles are stored to measure the shape of the region. This shape measure is translation and size invariant, but is rotation dependent. It effectively differentiates primitive geometrical shapes such as squares, rectangles, circles, and ellipses, while absorbs certain variations between similar shapes. Hence, it is an appropriate shape measurement for similarity-based image retrieval.

Again, all the feature vectors are stored as indexes in a SR-Tree $SRT_R$, which is independent of $SRT_C$.

In summary, assume that a total of $m$ clusters and $n$ image regions have been acquired from the image; then, the image will be indexed by $m$ cluster vectors and $n$ region vectors in the database.

## 5 Image Retrieval

Image retrieval is initiated by using a sample image. The system clusters the sample image, and then detects contiguous, sizable regions in the image.

Let $\{S_x\}$, $\{R_y\}$ be the cluster and the region set obtained from the sample image. Users are enabled to form a database query by specifying: (1) a subset of clusters (prominent colors) $S' \subset \{S_x\}$, or (2) a subset of regions $R' \subset \{R_y\}$, or (3) combinations of (1) and (2).

Figure 1(1a) shows the user interface of the image retrieval system. The sample image is displayed in the middle of the window, and the prominent colors obtained from the image are displayed by the toggle buttons in the bottom of the window. The colors are shown in descending order of the pixel count.

The user selects the desired colors by pressing the corresponding toggle buttons, and the system reverses colors of the pixels belonging to the selected prominent colors in the sample image. Then, the user can either select weights for the five color features — H, V, C, the pixel count, and the aggregation degree, or use the default weights set by the system (which are 1.0, 1.0, 1.0, 0.5, 0.5, respectively).

Assume that the user has selected color $S_x$, and wants to retrieve all the images that contain a color similar to $S_x$. The system calculates the feature vector for $S_x$ as described in Section 4, and then performs a nearest-neighbor (NN) search on SR-Tree $SRT_C$. The similarity score used by NN search is the weighted $L_1$

distance between two feature vectors, which is defined as:

$$sim(S_i, S_j) = w_H \frac{|H_i - H_j|}{\sigma_H} + w_V \frac{|V_i - V_j|}{\sigma_V} + w_C \frac{|C_i - C_j|}{\sigma_C} + w_N \frac{|N_i - N_j|}{\sigma_N} + w_{\mathcal{A}} \frac{|\mathcal{A}_i - \mathcal{A}_j|}{\sigma_{\mathcal{A}}}$$

where $w_H, w_V, w_C, w_N, w_{\mathcal{A}}$ are the weights for, and $\sigma_H, \sigma_V, \sigma_C, \sigma_N, \sigma_{\mathcal{A}}$ are the standard deviation of, the corresponding features, respectively. The standard deviations are computed over all the cluster feature vectors in the database.

For region-based retrieval, the user can select the desired image region $R_y$ by clicking on any part of $R_y$. The color of the entire region will be reversed by the system to confirm the selection. This time, the user can set his/her desired weights for H, V, C, the area, and the shape profile of the region. Similarly, the system performs a NN-search on SR-Tree $SRT_R$, and uses the following similarity score:

$$sim(R_i, R_j) = w_H \frac{|H_i - H_j|}{\sigma_H} + w_V \frac{|V_i - V_j|}{\sigma_V} + w_C \frac{|C_i - C_j|}{\sigma_C} + w_N \frac{|N_i - N_j|}{\sigma_N} + w_\Phi \frac{\sum_{k=1}^{20} |\Phi_i(k) - \Phi_j(k)|}{20\sigma_\Phi}$$

where $\Phi_i(k), \Phi_j(k)$ is the $k$-th element of $R_i, R_j$'s region profile, respectively, and $\sigma_\Phi$ is the standard deviation of the region profile over the entire database.

Assume that the user has selected a total of $u$ objects $\{\mathcal{O}_i\}$ from the sample image, and that $\mathcal{O}_i$ could be either a cluster or an image region. Let $E_i$ be the image set retrieved from the database using object $\mathcal{O}_i$. Then the final retrieved image set presented to the user becomes $\bigcap_{i=1}^{u} E_i$. The final similarity score of each image is the summation of its similarity score against each object $\mathcal{O}_i$.

## 6 Experiments and Discussions

Our method was implemented on an SGI workstation. The test database currently consists of about 5,000 color images of size $352 \times 240$. These images are obtained from six 30-minute CNN Headline News video programs. As a NTSC news video has a frame rate of 30 fps, most consecutive frames have a very similar content. To eliminate redundancies from the video data, we detect scene breaks using the method in [6], and select only the frames before, and after each of the scene breaks. Therefore, our test database is very heterogeneous and diversified, containing images of indoor and outdoor scenes, portrait faces, landscapes, etc.

To simplify quantitative evaluations of our image retrieval system, we extracted 80 different images from the same six CNN Headline News videos to form a sample image set. For each image in this sample image set, we chose exactly 3 additional images that belong to the same video segment, but containing variations such as translation, scaling, zooming, panning, different brightness, a different background, etc. Then, we inserted these resulting 320 additional images into the test database in addition to the above 5000 color images.

Afterwards, we performed database queries using each of the 80 images in the sample image set. For each database query, the system returns up to 12 images from the database. Figure 1 shows three representative image retrieval examples, which our comparison studies have proven difficult for histogram-based methods. Figure 1 1(a) shows the user interface that displays the first sample image. The 16 color clusters obtained from the sample image are represented by the toggle buttons at the bottom of the window. The database query was formed by selecting the color cluster that corresponds to the blue background of the image (the second toggle button). As a result, 12 images in Figure 1 1(b) are retrieved from the database. The first image is the sample image itself, and the subsequent three images belong to the corresponding image triplet. Although these images contain anchor persons wearing different suits in different positions, and contain the weather maps with different appearances, they are all retrieved from the database. The rest of the images are retrieved because they all contain a similar blue color background. In Figure 1 2(b), the image retrieval corresponds to the user query that uses the bright green lawn in the middle of the sample image 2(a). Again, the second to the fourth retrieved images are from the corresponding image triplet, which contain large appearance changes caused by the camera zooming. Figure 1 3(b) shows shape-based image retrieval, where the human face in the sample image 3(a) is selected to form the database query. The result consists of 12 images that contain a skin color region of a shape similar to that of the sample image.

Our quantitative evaluation consists of two measures: the percentage that the desired images are retrieved from the database; and the average rank at which the desired images are retrieved. Let $\mathcal{Q}_1$, $\mathcal{Q}_2$ and $\mathcal{Q}_3$ be the image triplet of the sample image $\mathcal{I}$, these two measures are defined as: (1) **Recall Rate:** $\mu(\mathcal{I}) = |\{\mathcal{Q}_i \mid rank(\mathcal{Q}_i) \leq 12, i = 1, 2, 3\}|$; and (2) **Precision:** $\rho(\mathcal{I}) = \frac{1}{466} \sum_{i=1}^{3} \phi(rank(\mathcal{Q}_i))$, where function $\phi(rank(\mathcal{Q}_i)) = 0$ when $rank(\mathcal{Q}_i) > 12$,

Table 2: Performances of the two methods

| Measure | **Cluster** | **Histogram** |
|---|---|---|
| $\overline{\mu}$ | 2.84 | 1.76 |
| $\overline{\rho}$ | 0.90 | 0.55 |

$\phi(12) = 1$, $\phi(11) = 2$ and $\phi(i) = \phi(i+1) + \phi(i+2)$ for $i = 10, \ldots, 1$. The precision measure is based on the classical Fibonacci sequence which is frequently used to evaluate ranking problems. The denominator is a normalization factor, which is the score for the best retrieval where the matching triplets of the sample images are retrieved in the first 3 positions (i.e., $466 = \phi(1) + \phi(2) + \phi(3)$. We eliminate the sample image when counting the ranks for the retrieved images). Essentially, this function gives an equal score to the retrieval of an image at rank $i$ and the retrieval of two images at ranks $i+1$ and $i+2$; hence, it reflects the precision of the retrieving process.

Finally, we average the above two measures over all the 80 queries, resulting in the two values shown in Table 2. For comparison, the same table also includes the two values for the traditional histogram-based image retrieval method using the same test database and sample image set. Note that $0 \leq \overline{\mu} \leq 3$ and $0 \leq \overline{\rho} \leq 1$, and that higher $\overline{\mu}$, $\overline{\rho}$ values mean higher retrieval capability of desired images, and higher precision of retrieving the desired images. As both $\overline{\mu}$ and $\overline{\rho}$ of our method are much closer to their upper bounds than the histogram-based method, it can be concluded that our proposed method is much more effective and accurate to retrieve user desired images.

Now we are testing our image retrieval method with a larger data corpus containing tens of thousands of images. Our future work includes application of the human perceptual color clustering to video segmentation as well as key frame extractions.


Figure 1: Three image retrieval examples; 1(a)∼3(a) the sample images; 1(b)∼3(b) the retrieved images.

## References


[1] M. Stricker and A. Dimai, "Color indexing with weak spatial constraints," in *SPIE Proceedings*, vol. 2670, pp. 29–40, 1996.

[2] G. Pass and R. Zabih, "Histogram refinement for content-based image retrieval," in *Proceedings of the Third IEEE Workshop on Applications of Computer Vision*, (Sarasota, Florida), Dec. 1996.

[3] J. Huang, S. Kumar, and et al., "Image indexing using color correlograms," in *Proceedings of CVPR'97*, (Puerto Rico), pp. 762–768, 1997.

[4] The Japanese Society of Chromatology, *The Handbook of Chromatology*. University of Tokyo, 1980.

[5] N. Katamaya and S. Satoh, "The SR-tree: An index structure for high-dimensional nearest neighbor queries," in *Proceedings of ACM SIGMOD*, (Tuscon, Arizona), May 1997.

[6] M. A. Smith and T. Kanade, "Video skimming and characterization through the combination of image and language understanding techniques," in *Proceedings of CVPR'97*, (Puerto Rico), pp. 775–781, 1997.