SEGMENTATION AND HISTOGRAM GENERATION USING THE HSV COLOR SPACE FOR IMAGE RETRIEVAL

Shamik Sural, Gang Qian and Sakti Pramanik

Dept. of Computer Science and Engineering,

3115 Engineering Building, Michigan State University, East Lansing, MI 48824, USA.

shamik@ieee.org, {qiangang, pramanik}@cse.msu.edu

ABSTRACT

We have analyzed the properties of the HSV (Hue, Saturation and Value) color space with emphasis on the visual perception of the variation in Hue, Saturation and Intensity values of an image pixel. We extract pixel features by either choosing the Hue or the Intensity as the dominant property based on the Saturation value of a pixel. The feature extraction method has been applied for both image segmentation as well as histogram generation applications – two distinct approaches to content based image retrieval (CBIR). Segmentation using this method shows better identification of objects in an image. The histogram retains a uniform color transition that enables us to do a window-based smoothing during retrieval. The results have been compared with those generated using the RGB color space.

1. INTRODUCTION

We have done in-depth analysis of the visual properties of the HSV color space and its usefulness in content based image retrieval applications. In particular, we have developed image segmentation and histogram generation applications using this color space – two important methods in CBIR [5,7].

Segmentation is done to decompose an image into meaningful parts for further analysis, resulting in a higher-level representation of the image pixels like the foreground objects and the background. In region-based CBIR applications, segmentation is essential for identifying objects present in a query image and each of the database images. Wang et al [12] have used the LUV values of a group of 4X4 pixels along with three features obtained by wavelet transform of the L component for determining regions of interest. Region-based retrieval has also been used in the NeTra system [4] and the Blobworld system [1]. We segment color images using features extracted from the HSV space as a step in the region-based matching approach to CBIR. The HSV color space is fundamentally different from the widely known RGB color space since it separates out the Intensity (luminance) from the color information (chromaticity). Again, of the two chromaticity axes, a difference in Hue of a pixel is found to be visually more prominent compared to that of the Saturation. For each pixel we, therefore, choose either its Hue or the Intensity as the dominant feature based on its Saturation. We then segment the image by grouping pixels with similar features using the K-means clustering algorithm [3].

A standard way of generating a color histogram of an image is to concatenate 'N' higher order bits for the Red, Green and Blue values in the RGB space [11]. The histogram then has 2^{3N} bins, which accumulate the count of pixels with similar color. It is also possible to generate three separate histograms, one for each channel, and concatenate them into one [2]. Smith and Chang [8] have used a color set approach to extract spatially localized color information. Ortega et al [6] have used the HS co-ordinates to form a two-dimensional histogram where each bin contains the percentage of pixels in the image that have corresponding H and S colors for that bin. We generate a one-dimensional histogram from the HSV space where a perceptually smooth transition of color is obtained in the feature vector. This enables us to use a window-based smoothing of histograms so that similar colors can be matched between a query and each of the database images.

We explain the HSV-based feature extraction and image segmentation method in the next section and the histogram generation technique in section 3. We then present experimental results in section 4 and draw conclusions from our work in the last section of the paper.

2. IMAGE SEGMENTATION USING FEATURES FROM THE HSV COLOR SPACE

2.1. Analysis of the HSV Color Space

A three dimensional representation of the HSV color space is a hexacone, where the central vertical axis represents the Intensity [9]. Hue is defined as an angle in the range $[0,2\pi]$ relative to the Red axis with red at angle 0, green at $2\pi/3$, blue at $4\pi/3$ and red again at 2π. Saturation is the depth or purity of the color and is measured as a radial distance from the central axis with value between 0 at the center to 1 at the outer surface. For S=0, as one moves higher along the Intensity axis, one goes from Black to White through various shades of gray. On the other hand, for a given Intensity and Hue, if the Saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the

color represented by its Hue. Looked from a different angle, any color in the HSV space can be transformed to a shade of gray by sufficiently lowering the Saturation. The value of Intensity determines the particular gray shade to which this transformation converges. When Saturation is near 0, all pixels, even with different Hues, look alike and as we increase the Saturation towards 1, they tend to get separated and are visually perceived as the true colors represented by their Hues as shown in figure 1. Thus, for low values of Saturation, a color can be approximated by a gray value specified by the Intensity level while for higher Saturation, the color can be approximated by its Hue. The Saturation threshold that determines this transition is once again dependent on the Intensity. For low intensities, even for a high Saturation, a color is close to the gray value and vice versa. Saturation gives an idea about the depth of color and human eye is less sensitive to its variation compared to variation in Hue or Intensity. We, therefore, use the Saturation value of a pixel to determine whether the Hue or the Intensity is more pertinent to human visual perception of the color of that pixel and ignore the actual value of the Saturation. It is observed that for higher values of intensity, a saturation of 0.2 differentiates between Hue and Intensity dominance. Assuming the maximum Intensity value to be 255, we use the following threshold function to determine if a pixel should be represented by its Hue or its Intensity as its dominant feature.

$$\text{th}_{\text{sat}}(V) \; = 1.0 - \frac{0.8V}{255} \qquad\qquad (1)$$

In the above equation, we see that for V=0, th(V) = 1.0, meaning that all the colors are approximated as black whatever be the Hue or the Saturation. On the other hand, with increasing values of the Intensity, Saturation threshold that separates Hue dominance from Intensity dominance goes down.

2.2. Feature Generation using the HSV Color Space

We generate features by utilizing the above properties of the HSV color space for clustering pixels into segmented regions. Figure 2(a) shows an image and figure 2(b) shows the same image using the approximated pixels after Saturation thresholding. Pixels with sub-threshold Saturation have been represented by their gray values while the other pixels have been represented by their Hues. The feature generation used by us makes an approximation of the color of each pixel in the form of thresholding. On the other hand, features generated from the RGB color space approximate by considering a few higher order bits. In figures 2(c) - (d) we show the same image approximated with the six lower-order bits all set to 0 and all set to 1, respectively. It is seen that the approximation done by the RGB features blurs the distinction between two visually separable colors by changing the brightness. On the other hand, the HSV-

based approximation can determine the intensity and shade variations near the edges of an object, thereby sharpening the boundaries and retaining the color information of each pixel. This phenomenon is exhibited in detail in figure 3. Figure 3(a) shows a number of solid colors with varying intensities. Figure 3(b)-(c) shows the result of approximation using the RGB color space taking the 2 higher order bits. It is seen that some of the colors with high intensities cannot be recognized, as they are inseparable from the background. Also, we see that the background of white and gray are considered equivalent due to approximation. The HSV features used by us retain the identity of the colors even at these intensity levels as seen in figure 3(d). This makes the HSV-based features very useful in running segmentation algorithms like clustering on the approximated pixels.

2.3. Pixel Grouping by K-means Clustering Algorithm

The RGB value of a pixel is first transformed to the HSV value using a method suggested in [9]. The feature is next extracted from each image pixel. After extraction, the pixel features are clustered using the K-Means clustering algorithm to group them into regions of similar color. Since the Hue and the Intensity values belong to the same number space, the two sets of data are clustered separately so that the color and the gray value pixels are not considered in the same cluster. In the K-means clustering algorithm, we start with K=2 and adaptively increase the number of clusters till the improvement in error falls below a threshold or a maximum number of clusters is reached. We set the maximum number of clusters to 12 and an error improvement threshold over number of clusters to 5 %.

3. HISTOGRAM GENERATION

We also use the HSV color space for histogram generation where each pixel contributes either its Hue or its Intensity as explained in the last section. We extract the color histogram as the feature vector having two parts: (i) A representation of the Hue between 0 and 2π quantized after a transformation and (ii) A quantized set of gray values as shown in figure 4(a). The number of components in the feature vector generated based on Hue is given by:

$$N_h = \lceil 2\pi\,\text{MULT\_FCTR} \rceil \; + 1 \qquad\qquad (2)$$

Here MULT_FCTR determines the quantization level for the Hues. We typically choose a value of 8. The number of components representing gray values is:

$$N_g = \; \left\lceil \frac{\text{Imax}}{\text{DIV\_FCTR}} \right\rceil + 1 \qquad\qquad (3)$$

Here I_{max} is the maximum value of the Intensity, usually 255, and DIV_FCTR determines the number of quantized gray levels. We, typically, choose DIV_FCTR =

16. The quantized values of Hue may be considered circularly arranged since Hue varies between 0 to 2π, both the end points being red. The feature vector is thus a combination of two independent vectors as shown in figure 4(b).

It has been observed that, when color histograms are extracted from two similar images, often two neighboring components in the histograms have high values. This is due to the fact that two colors that appear close to the human eye may have a small difference in shade and map to two neighboring components in the histogram. When a standard measure like the Euclidean distance is used to order such feature vectors, the result shows a high distance value. To overcome this drawback, we compare two histograms through smoothing windows instead of comparing the vector components directly. All the conventional color histograms fail to provide the requisite perceptual gradation of colors in the feature vectors as required by such a comparison. The histogram generated by us retains this property in the feature vector. For the jth component ($j \in [0, N_h+N_v-1]$), with a smoothing window size of $2N+1$, the average value is calculated as follows:

$$\text{Hist}_w[j] = \sum_{i=j-N}^{j+N} w(i-j)\text{Hist}[i] \text{ where } w(i-j) = 2^{-|i-j|} \quad (4)$$

Since we derive both the Hue and the gray level features using the HSV space, there are two independent color continuums in the histogram, one from Red→Green→Blue→Red and the other from Black→Gray→White. The circular nature of the Hue components and the discontinuity at the Hue-Intensity component boundary are taken care of programmatically. The use of the new histogram shows an improved performance over conventional histograms generated from the RGB color space. The smoothing window further tunes the result of retrieval.

4. EXPERIMENTAL RESULTS

4.1. Segmentation Results
We have tested the algorithm on a large number of natural scene images. In this paper we demonstrate results that represent our findings from these experiments. In figures 5(a)-(c), we show three images, their HSV-based segmentation results and RGB-based segmentation results. For RGB, we consider the higher order 2 bits to generate the feature vectors. In the images, we have painted the different regions using the color represented by the centroid of the clusters to give an idea about the differentiation capabilities of the two color spaces. Although exact segmentation of unconstrained color images is still a difficult problem, we see that the object boundaries can be identified in a way more similar to human perception of the same. The RGB features, on the other hand, fail to determine the color and Intensity

variations and come up with clusters that put neighboring pixels with similar color but small difference in shade to different clusters. Often, two distinct colors are merged together. For the first image in figure 5, it is seen that RGB clustering could not detect the object boundary at all with the color of the man's body merged with the color of the river due to high brightness. In the second image, we see that the sky has been identified as three regions and also the bush in front of the castle is interspersed with the color of the brick wall. In the third image, the faces of the people could not be identified distinctly. In some cases, a face was clustered along with the color of the dress of the subject. In the HSV-based approach, better clustering was achieved in all the cases with proper segmentation. The clustered image pixels may be further processed to merge small image regions into larger blocks for marking the exact object boundaries.

4.2. Histogram based Image Retrieval Results
We have developed an interface using Java applet that displays images similar to a query image from a database of about 14,500 images obtained from the web and IMSI master clips. Figures 6(a)-(b) show the recall and precision of image retrieval using a standard RGB histogram and the new histogram. For the new histogram, we show results for different widths of the smoothing window. From the figures, it is seen that the new histogram performs much better than an RGB histogram based system. In most cases, recall and precision values are higher for the same number of nearest neighbors. It is also observed that application of a window with small width improves the result set. Again, for a very large window width, different distinct colors tend to get added up and hence we do not see better results anymore. Such a comparison cannot be done with the RGB features due to the lack of color continuity in the generated histogram as explained in the last section. Some of the preliminary results in terms of actual retrieved images are available in [10].

5. CONCLUSIONS

We have studied some of the important properties of the HSV color space and have developed a framework for extracting features that can be used both for image segmentation and color histogram generation – two important approaches to content based image retrieval. Our approach makes use of the Saturation value of a pixel to determine if the Hue or the Intensity of the pixel is more close to human perception of color that pixel represents. The K-means clustering of features combines pixels with similar color for segmentation of the image into objects. We are also able to generate a histogram that enables us to perform a window-based smoothing of the vectors during retrieval of similar images. While it is well

established that color itself cannot retain semantic information beyond a certain degree, we have shown that retrieval results can be considerably improved by choosing a better histogram.

Figure 1. Variation of color perception with saturation (Decreasing from 1 to 0 left to right) for a fixed value of Intensity and Hue = 0 (Red), Hue = $2\pi/3$ (Green), Hue = $4\pi/3$ (Blue).
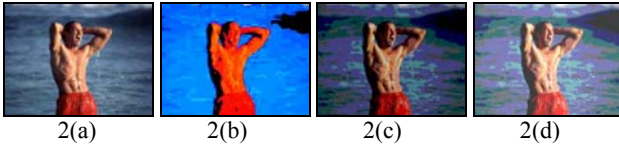

2(a) 2(b) 2(c) 2(d)

Figure 2. (a) Original Image (b) HSV Approximation (c) RGB approximation with all low order bits set to 0 and (d) RGB approximation with all low order bits set to 1.

3(a) 3(b) 3(c) 3(d)
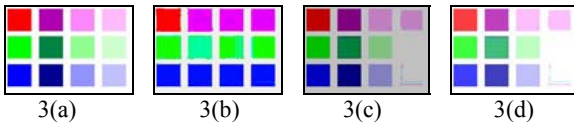
Figure 3. (a) Original Colors (b) HSV Approximation (c) RGB approximation with all low order bits set to 0 and (d) RGB approximation with all low order bits set to 1.
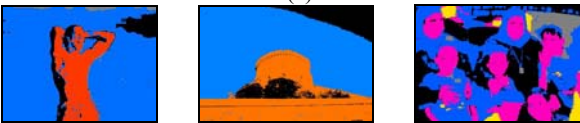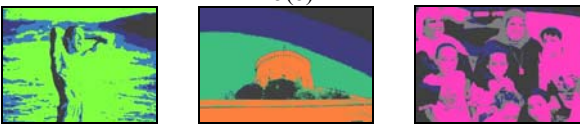

4(a)
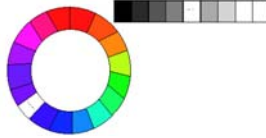
4(b)

Figure 4. (a) Representation of colors in the histogram and (b) Circular representation of hue and linear representation of gray values in the histogram.
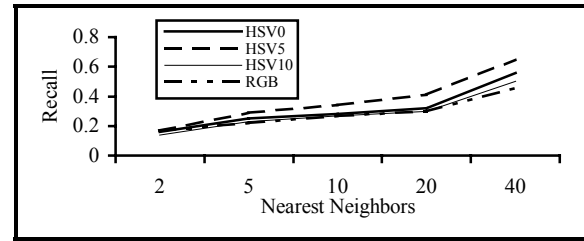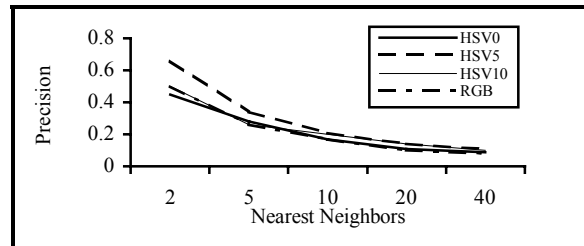

5(a)

5(b)

5(c)

Figure 5. (a) Original Images (b) Segmentation using HSV features and (c) Segmentation using RGB features.

6(a)

6(b)

Figure 6. (a) Recall and (b) Precision variation of the new histogram and a standard RGB histogram.

References

[1] C.Carson et al, "Blobworld: A System for Region-based Image Indexing and Retrieval", Proc. Third Int. Conf. on Visual Information Systems, June 1999.

[2] A.Jain and A.Vailaya, "Image Retrieval using Color and Shape, Pattern Recognition, vol. 29, no. 8, pp. 1233-1244, 1996.

[3] L.Kaufman and P.J.Rousseeuw, "Finding Groups in Data: An Introduction to Cluster Analysis", John Wiley & Sons, 1990.

[4] W.Y.Ma and B.Manjunath, "NeTra: A Toolbox for Navigating Large Image Databases", Proc. IEEE Int. Conf. on Image Processing, pp. 568-571, 1997.

[5] W.Niblack et al, "The QBIC Project: Querying Images by Content using Color Texture and Shape", Proc. SPIE Int. Soc. Opt. Eng., in Storage and Retrieval for Image and Video Databases, vol. 1908, pp. 173-187, 1993.

[6] M. Ortega et al, "Supporting Ranked Boolean Similarity Queries in MARS", IEEE Trans. on Knowledge and Data Engineering, vol. 10, no. 6, pp. 905-925, 1998.

[7] A.W.M. Smeulders et al, "Content Based Image Retrieval at the End of the Early Years", IEEE Trans. on PAMI, vol. 22, no. 12, pp. 1-32, December, 2000.

[8] J.R.Smith and S.-F. Chang, "VisualSeek: A Fully Automated Content based Image Query System", Proc. ACM Multimedia Conf., Boston, MA, 1996.

[9] G.Stockman and L.Shapiro, "Computer Vision", Prentice Hall, 2001.

[10] S.Sural, G.Qian and S.Pramanik, "A Histogram with Perceptually Smooth Color Transition for Image Retrieval", Proc. Fourth Int. Conf. on CVPRIP, Durham, 2002 (to appear).

[11] M.Swain and D.Ballard, "Color Indexing", Int. Journal of Computer Vision, vol.7, no. 1, pp. 11-32, 1991.

[12] J.Z.Wang, Jia Li, Gio Wiederhold, "SIMPLIcity: Semantics-sensitive Integrated Matching for Picture LIbraries," IEEE Trans. on PAMI, vol. 23, no. 9, pp., 2001.