IMPLEMENTATION OF CONTENT-BASED IMAGE RETRIEVAL USING THE CFSD ALGORITHM

Nagaraja. G. S[1], Samir Sheriff[2], Raunaq Kumar[3]

[1] Associate Professor, [2,3] VII Semester B.E., Dept of Computer Sci. Eng., R.V. College of Engineering, Bangalore, India
nagarajags@rvce.edu.in, samiriff@gmail.com, devilronny@gmail.com

Abstract

Content-based image retrieval (CBIR) is the application of multimedia -processing techniques to the image retrieval problem, that is, the problem of searching for digital images in large databases. There is a growing interest in CBIR because of the limitations inherent in metadata-based systems. The most common method for comparing two images in CBIR is using an image distance measure based on color, texture, shape and others. In this paper, we describe an object-oriented graphical implementation of a system, backed by a My SQL database, that computes distance measures using the CFSD algorithm, based on color similarity

----------*****----------

1. INTRODUCTION

"Content-based" means that the search will analyze the actual contents of the image rather than the metadata such as keywords, tags, and/or descriptions associated with the image. The term 'content' in this context might refer to colors, shapes, textures, or any other information that can be derived from the image itself.

The system we have implemented uses color-based feature extraction. Computing distance measures based on color similarity is achieved by computing a color histogram for each image that identifies the proportion of pixels within an image holding specific values. Examining images based on the colors they contain is one of the most widely used techniques because it does not depend on image size or orientation.

Our program allows the user to choose between two algorithms

_ Color Histogram Method: In image processing andphotography, a color histogram is a representation of thedistribution of colors in an image. For digital images, acolor histogram represents the number of pixels that havecolors in each of a fixed list of color ranges, that spanthe image's color space, the set of all possible colors.

_ Color Frequency Sequence Difference Method: Thehigh dimensionality of feature vectors of the Color Histogrammethod results in high computation cost and spacecost. CFSD expresses the color image in terms of numericalvalues for each color channel, greatly improvingcomparison time and computation costs.

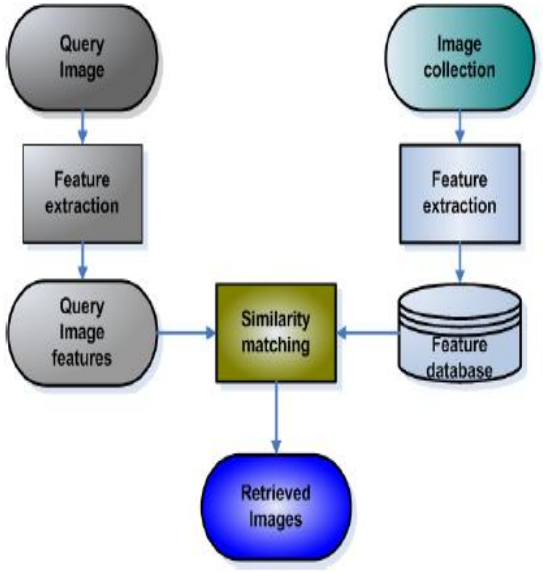

Fig1. Content-Based Image Retrieval System

2. THE COLOR HISTOGRAM METHOD

1) Color Quantization: For every image in the database,colors in the RGB model are quantized, to make latercomputations easier. Color quantization reduces thenumber of distinct colors used in an image. In ourapplication, the user can select the quantization bucketsize for red, green and blue separately.

2) Compute Histogram: For every quantized image: Calculatea color histogram, which is a frequency distributionof quantized RGB values of each pixel of animage. In our application, we have used hashmaps tostore histogram values as <string, double >pairs.

3) Query Image Feature Extraction: Compute the quantizedcolor-values and the resulting histogram of thequery image.

4) Compute Distance Measure: Compare the histogramof the query image with the histograms of every imageof the retrieval system. Let h and g represent two colorhistograms. If n = 2B □ 1, where B is the maximumnumber of bits used for representing quantized colorcomponents a, b and c, then the euclidean distancebetween the color histograms h and g can be computed
as:

Xn
a=0
Xn
b=0
Xn
c=0
j(h(a; b; c) □ g(a; b; c))j (1)

In this distance formula, there is only comparison betweenthe identical bins in the respective histograms.Two different bins may represent perceptually similarcolors but are not compared cross-wise. All bins contributeequally to the distance.

5) Find Similar Images: The similarity between twoimages is inversely proportional to the (color histogram)distance between them.

3. THE COLOR FREQUENCY SEQUENCE DIFFERENCE METHOD

1) Color Space Conversion: Translate the representationof all colors in each image from the RGB space to theHSV space. HSV color space has two distinct characteristics:one is that lightness component is independentof color information of images; the other is that hueand saturation component is correlative with manner ofhuman visual perception.

2) Color Quantization: For every image in the database,colors in the HSV model are quantized, to make latercomputations easier. Color quantization reduces thenumber of distinct colors used in an image. In ourapplication, the user can select the quantization bucketsize for each color component separately.

3) Compute Histogram (Sequence): For every quantizedimage, a color histogram is calculated, which is afrequency distribution of quantized HSV values of eachpixel of an image. In our application, we have usedtreemaps to store histogram values as <string, double>pairs. The values are ordered alphabetically by theirHSV strings. For instance, (H, S, V) string (0, 0, 0) isfollowed by (0, 0, 1).

4) Compute Sorted Histogram (Frequency): Take thecolor histogram of each image and sort the keys indescending order of frequency values. Each image nowhas 2 histograms:
_ One sorted by color sequence Call this P
_ One sorted by color frequency Call this Q
However, in our application, we maintain only thesorted histogram as the other histogram can be derivedimplicitly.

5) Query Image Feature Extraction: Convert and quantizethe color space of the query image. Then, computethe sequence histogram as well as the frequency histogramof the query image.

6) Compute SFD: In this algorithm, SFD is an alias fordistance measure, which is calculated as follows, foreach component of the HSV model:
For every color combination x in Histogram P,

sfd =
XN
i=1
w(i)h(i) (2)

where
_ i is the index of color combination x in histogram
Q and i is the index of x in histogram P
_ w(i) = 1
ji□i0j+1.w is a function of the difference
between the indices i and i.
_ h is a function of the frequency of color combination
x in the image

7) Compute Distance with SFD (Difference): Let h andg represent two images which have SFD values foreach component of the HSV model. The SFD distancebetween h and g can be computed as:
Dsfd(a; b) = (sfdH(a; b)+sfdS(a; b)+sfdV (a; b))=3
(3)
where
_ sfdT (a; b) = asfdT □ bsfdT which is the distancebetween the corresponding SFD values of the componentT of the HSV model of the two images aand b.
_ T_fH; S; V g
_ asfdT is the corresponding SFD value of componentT of the HSV model representing image a

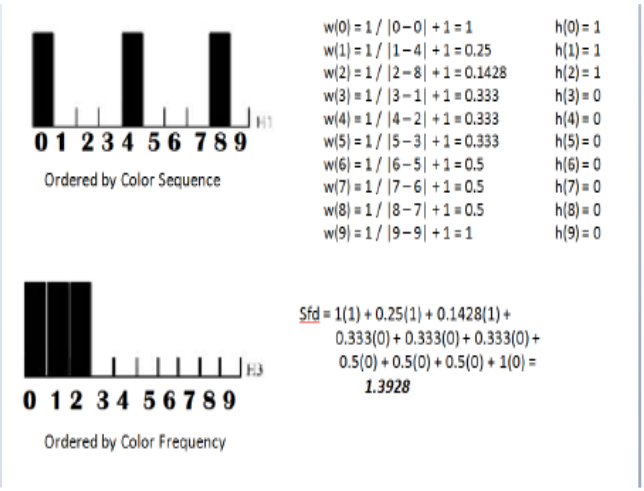

Fig2. SFD Calculation for one component of a given image

8) Find Similar Images: Compare the SFDs of each HSVcomponent of the query image with the correspondingSFDs of every image of the retrieval system. The similaritybetween two images is inversely proportional tothe SFD distance between them.

4. OBJECT-ORIENTED DESIGN

The Java Programming Language has been used to createour Content-Based Image Retrieval System, with MySQL forthe back-end. For displaying graphical objects, we have usedthe ACM package (created by the Java Task Force (JTF)),which provides a set of classes that support the creation ofsimple, object-oriented graphical displays. The object-orienteddesign of our program below:

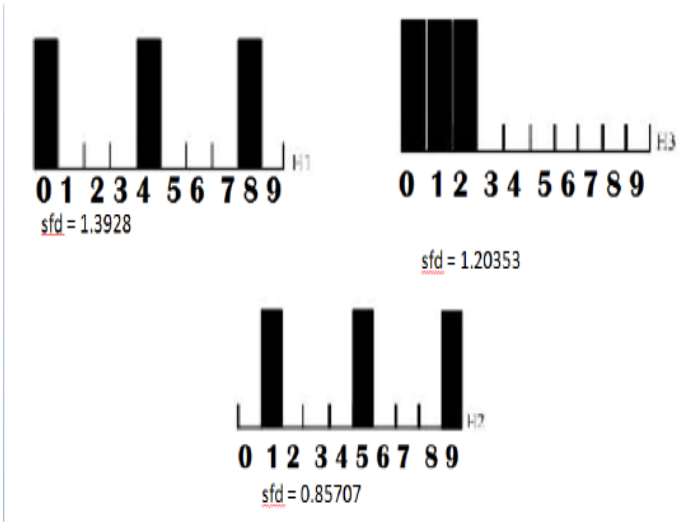

Fig3. Finding Similar Images based on SFD Distance. The two images on top are similar since their sfd values are quite close. This similarity cannotbe detected by the normal color histogram method.

A. Class Design

We have segregated the code into two packages for easymanagement:

_ retriever - This package contains classes that providemethods to read a database of images, store status information.The implementation of the algorithms mentionedin the previous sections can be found in this package.

_ gui - As the name suggests, this package contains allthe code necessary for the graphical user interface of ourretrieval system. It makes use of methods provided by theACM library as well as the Swing library.

Now that you know the organization of code, let us divedeeper into the actual object-oriented design. The retrieverpackage consists of the following classes:

1) ColorFrequency - This class is used while storing colorhistograms in hashmaps/treemaps, to store< value, frequency > tuples for different color combinations.It implements the Comparable Interface so thatthe Collections.sort() method provided by the java.utilpackage to sort histograms in decreasing order of frequency,as required by the CFSD algorithm.

2) ImageData - This class keeps track of the status of asingle image of the database. It stores information andprovides methods to:

_ store the two-dimensional representation of animage using the GImage class of the ACM package

_ store the path of this image

_ quantize every color model component of the givenpixel of this image

_ create color histograms with ColorFrequency mappingsfor individual components as well as a combinationof all components of the color space of thisimage_ calculate SFD values for every component of thecolor space of this image_ establish or disable a connection with a databaseand retrieve status information if feature values havealready been calculated for this image during aprevious run of the program. This is done to preventunnecessary computation of SFD values wheneverthe program is run repeatedly

3) ImageCollection - This class keeps track of the Image-Data of a number of images in the database. In essence,an ImageCollection Object is a collection of ImageDataobjects.It provides methods to:

_ initialize an ImageData collection with images froma specified directory, which is traversed to a folderdepth of 1

_ find images that are most similar to a query image,computed based on the Color Histogram method andEuclidean distance

formula. A list of images, sortedbased on decreasing order of similarity is returned
_ find images that are most similar to a query image,computed based on the CFSD method and SFDdistance formula. A list of images, sorted based ondecreasing order of similarity is returned.

4) UserVariables - This class only maintains the values ofall the variables that the user can set in the GUI window.The ImageData and ImageCollection objects read thesevalues during computation. It provides methods to getand set:
_ the number of red, green and blue bins for colorquantization
_ the number of similar images to be returned byImageCollection
_ aboolean variable denoting whether the Color Histogram Algorithm or CFSD algorithm should be
Used

The gui package consists of only one class, namely:
1) MainGraphics - This class uses the power of the widgetsprovided by the ACM graphics package and the Javaswing library to create a user-friendly interface for ourImage Retrieval System. It creates a window consistingof a white area where the query image selected by theuser, and the three most similar images are displayed,in their normal forms as well as their quantized forms.

The northern portion of the window consists of:
_ Two Buttons - One to open a query image and anotherto start the execution of the selected algorithm,which is the Color Histogram Algorithm, by default.
_ Three sliders - each of which can be varied from 1to 60. The sliders are used to choose the bin sizesof each component of the RGB(HSV) model for theColor Histogram (CFSD) algorithm
_ One checkbox - If checked, then the CFSDalgorithm is run to find similar images. Otherwise,the Color Histogram method is used

For each of these widgets, the corresponding variablesof the UserVariable class are set.

B. Schema Design

Since comparisons in the CFSD algorithm are done using afew SFD values instead of comparing entire histograms as inthe normal color histogram method, the CFSD algorithm canbe made to operate more efficiently by calculating the SFDvalues of an image only once (when it is encountered for thevery first time) and storing them in a database which can beaccessed for all future references. Please note that wheneverwe say red, green or blue in the CFSD algorithm, we actuallyrefer to the H, S or V values of the image respectively. Thiswas done for compatibility.

The database that we constructed for our system consists oftwo tables:

1) imagedata - It consists of the following attributes:_ Image ID - which is the primary key of this table.
It is an auto-incremented variable
_ Image Path - A string with a variable number ofcharacters that stores the path of the given image,relative to the directory of the application
_ redSFD - The SFD calculated using the red (H)histogram of the image
_ greenSFD - The SFD calculated using the green (S)histogram of the image
_ blueSFD - The SFD calculated using the blue (V)histogram of the image

2) quantizedvalues- It consists of the following attributes:
_ Image ID - which is the foreign key of this tablethat references the primary key of the imagedatatable.
_ qRed - The value of the bin used to quantize red(H)
_ qGreen - The value of the bin used to quantize green(S)
_ qBlue - The value of the bin used to quantize blue(V)

C. Summary and Results

In order to test out our image retrieval system, we have provideda sample set of images, organized in seven folders withfive images each. Feel free to add more folders if necessary,but ensure that each folder contains exactly five images at anygiven instant of time. Using our sample database, it was foundthat this application could retrieve at least three similar imagesto a supplied query image, chosen from the database itself. Thespeed of retrieval varied depending on the algorithm selected,with the CFSD algorithm running faster than the naive ColorHistogram algorithm.

5. FUNCTIONALITY

First ensure that all the above requirements are met. Then,to start the application, go to the installation folder and clickon file named CBIR.jar. In the new window that pops up,select the appropriate query image using the open button. Afterselecting the appropriate number of bins for quantization ofeach component of the color space, and checking/uncheckingthe CFSD box, click on the Retrive Similar Images button toretrieve the three most similar images that will be displayedin the lower portion of the window. Figures 4, 5 and 6 showthe state of the window at different instances of time.

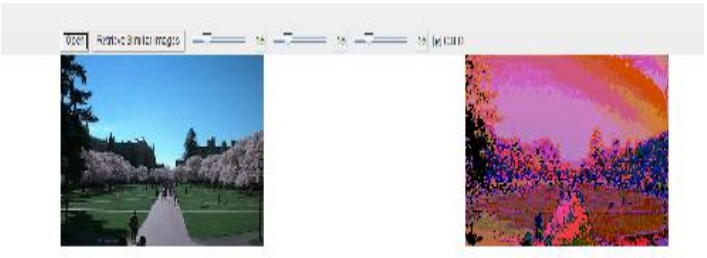

Fig4. Application after the query image has been selected by the use

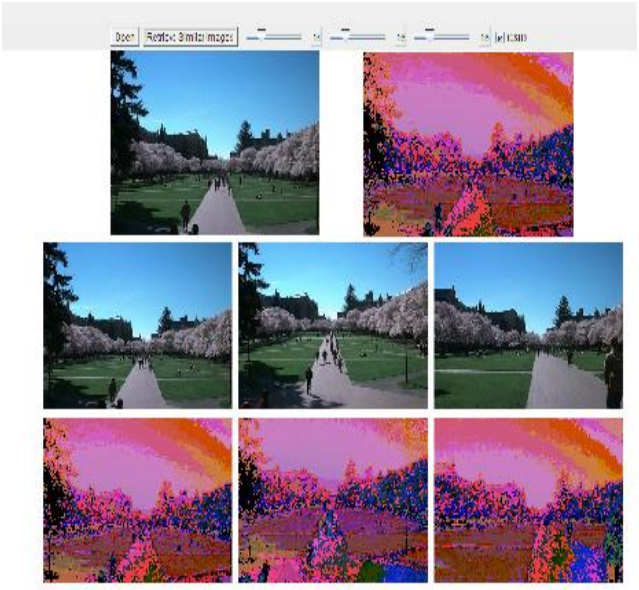

Fig5 Application after images similar to the query image have been retrieved using the CFSD algorithm

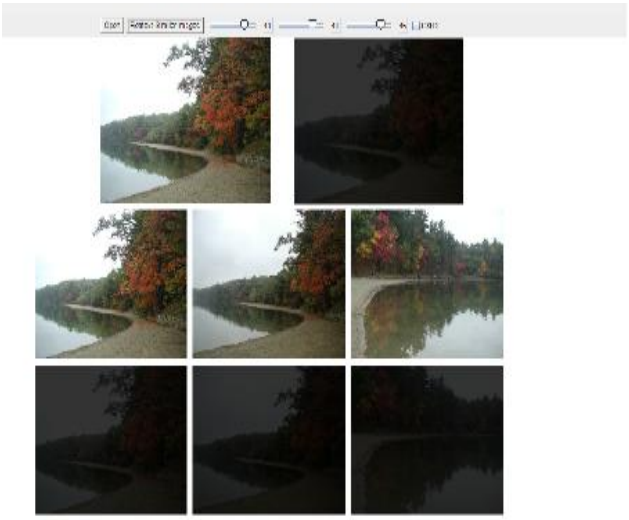

Fig6 Application after images similar to the query image have been retrieved using the Color Histogram algorithm

CONCLUSIONS

This paper discussed briefly the software implementation of a Content-Based Image Retrieval System using the CFSD algorithm as well as a naive Color Histogram method. This software tool was implemented in Java, relying heavily on Object-Oriented concepts without which code management wouldn't have been easy at all. We also had the opportunity to find out that the naive Color Histogram algorithm is not as fast as the CFSD algorithm. In fact, the CFSD algorithm, coupled with a database, forms a highly efficient system which retrieves at most three similar images quite accurately. However, it was found that the accuracy diminished when more than three similar images were sought. This leads us to conclude that we should also consider other features such as shape, etc., while comparing images in a content-based retrieval system.

ACKNOWLEDGMENTS

The authors wish to express their gratitude to Dr. NagarajaG.S., who offered invaluable assistance, support and guidance throughout the development of this tool.

REFERENCES

[1] RishavChakravarti, XiannongMeng, "A Study of Color HistogramBased Image Retrieval", Sixth International Conference on InformationTechnology: New Generations, 2009. ITNG '09., pp. 1323□1328, 2009

[2] Zhenhua ZHANG, Yina LU, Wenhui LI and Wei LIU, "Novel ColorFeature Representation and Matching Technique for Content-based ImageRetrieval", International conference on Multimedia Computing and Systems (ICMCS 09), pp. 118□122, 2009

[3] M. J. Swain and D. H. Ballard, "Color indexing", International Journalof Computer Vision, vol. 7, no. 1, pp. 11□32, 1991.

[4] H. Lewkovitz and G. H. Herman, "A generalized lightness hue andsaturation color model", Graphical Models and Image Processing, pp.271□278, 1993

[5] Smeulders A.W.M., Worring, M., Santini, S., Gupta, A., and Jain, R.,"Content-based image retrieval at the end of the early years", IEEE transactionson Pattern Analysis and Machine Intelligence, pp. 1349□1380,2000

[6] M. Stricker and M. Orengo, Similarity of color images, Proceeding ofSPIE Storage and Retrieval for Image and Video Databases III, vol. 2420,pp. 381□392, 1995.

[7] JaumeAmores, NicuSebe, and PetiaRadeva, "Context-Based Object-Class Recognition and Retrieval by Generalized Correlograms", Transactionson Pattern Analysis and Machine Intelligence, vol. 29, no. 10, pp.1818□1833, 2008