Volume : 2 | Issue : 4 | April 2013 ISSN - 2250-1991

Research Paper Engineering

Review on Various Image Retrieval Techniques

* Vinky

* M.TECH, Student, Dept. of CSE, S.G.G.S.W.U., Fatehgarh Sahib(Punjab), India

ABSTRACT

Color based image retrieval is an important research area in digital image processing and Color is the useful property of the image. Finding good similarity measures between images based on some feature set is a challenging task. To find efficient and accurate color image similar values is a complicated and exigent task because there are limitations in matching two different types of images. In this paper, various techniques are discussed to find similar values between query and database color image by using histogram, spatiogram and other methods. RGB and HSV and other color spaces are used.

Keywords : Image retrieval, color space, histogram, spatiogram, bins, texture discriptor.

INTRODUCTION

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually Image Processing system includes treating images as two dimensional signals while applying already set signal processing methods to them. It is among rapidly growing technologies today, with its applications in various aspects of a business. Image Processing forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps.

- Importing the image with optical scanner or by digital photography.
- Analyzing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs.
- Output is the last stage in which result can be altered image or report that is based on image analysis.

Purpose of Image processing

The purpose of image processing is divided into 5 groups. They are:

1. Visualization - Observe the objects that are not visible.
2. Image sharpening and restoration - To create a better image.
3. Image retrieval - Seek for the image of interest.
4. Measurement of pattern – Measures various objects in an image.
5. Image Recognition – Distinguish the objects in an image.

Image Processing Toolbox in Matlab provides a comprehensive set of reference-standard algorithms and graphical tools for image processing, analysis, visualization, and algorithm development. You can perform image enhancement, image deblurring, feature detection, noise reduction, image segmentation, geometric transformations, and image registration. Many toolbox functions are multithreaded to take advantage of multicore and multiprocessor computers. Image Processing Toolbox supports a diverse set of image types, including high dynamic range, gigapixel resolution, embedded ICC profile, and tomographic. Graphical tools let you explore an image, examine a region of pixels, adjust the contrast, create contours or histograms, and manipulate regions of interest (ROIs), spatiogram. With toolbox algorithms you can restore degraded images, detect and measure features, analyze shapes and textures, and adjust color balance.

literature survey

In [1], the author has presented properties of an image retrieval system based on color matching. The implemented procedure is presented as a simplified pseudo code presentation of the system. The RGB components are analyzed pixel by pixel and their real combination is the most important thing. Adaptive histogram methods are used which allow the comparison of histograms with different numbers of bins, but this increases computational complexity and it is not a desired property when similarity analysis will be made on-line over a large image database. Grayscale and color images cannot be compared together, because it is not possible to build a color histogram from a grayscale image.

Various steps used [1] by author to evaluate the color based matching:

1. If a query image has already been analyzed and data stored into a database, jump to 7.
2. Read the query image from the disk.
3. Make a color space conversion if needed.
4. Analyze the query image pixel by pixel and store the values in a quantized histogram.
5. Change bin hit values to relative values (percentage).
6. Save the histogram data with additional information (average color, dominant colors) and image name in the database.
7. Calculate color differences and histogram difference on a higher level (8 - 64 bins) between query image and database images.
8. Select the best candidates to closer evaluation (64 - 512 bins).
9. Select matching images.

Author has done the MATLAB simulations for all color spaces. Histograms have been built for 2041 images (average size 240×320) and Histogram build times (Total time and Time/image) is calculated for all color spaces. These times were proportional to the computational complexity analysis. HSV takes much more CPU-time than other methods. Then Histogram comparisons have been made between histograms and analysis times and histogram comparison tomes are calculated which shows histogram analyses are 972 - 6333 times faster than the histogram build part. System analyzing software has been used during this last step.

When natural images were used as a query image usually the best result was got either from RGB 125 or HSV 166. In case of a constructed query image RGB 64 produced the best results In [2], author has proposed a method for image mining based on analysis of color Histogram values and texture descriptor of an image. Author used entropy, local range and standard deviation for texture descriptor analysis and histogram values are used to extract the color properties of an image.

The author proposed the following algorithm[2]:
Step 1: Load database in the Mat lab workspace.
Step 2: Resize the image for [128. 128].
Step 3: Convert image from RGB to Gray.
Step 4: Normalize the gray image for fixed mean.
Step 5: Generate the histogram of RGB.
Step 6: Find entropy, standard deviation and local range of Gray.
Step 7: Combine the image feature.
Step 8: Load the test image.
Step 9: Apply the procedure 2-7 to find combine feature of test image.
Step 10: Determine the normalized Euclidean distance of test image with stored image of database.
Step 11: Sort the normalized Euclidean distance values to perform indexing.
Step 12: Display the result on GUI.

The images having closest value compared to query image are retrieved from database and displayed on GUI as result.

In [3], the author proposed a new matching technique to find the similar value between query color image and database color image using histogram, spatiogram and bins using RGB and HSV color space. Author has used Histogram Intersection (HI), Histogram Euclidean Distance (HED) and Histogram Quadratic Distance Measures (HQDM) as the similarity measures.

The author has proposed the color matching algorithm[3]:
(1) Read the database image d(I) and query image q(I) and both image are RGB color images.
Where d(I) and q(I) are variables

(2) Convert d(I) and q(I) RGB image into HSV image.
RGB query images [q(I)] àHSV query images [q'(I)]
RGB database images [d(I)] à HSV database images [d'(I)]

(3) Extract a color histogram from each image h1 and h2.
Bins = {4, 8, 12, 32 …}

(4) Compare their histogram, similarity (h1, h2).
S = sum [sum {sum (√h1*√h2)}]
If result = 0, very low similarity
Result = 0.9, good similarity
Result = 1, perfect similarity.

(5) Start the process of re-matching.

(6) Extract a color Spatiogram from each image s1 and s2.
Bins = {4, 8, 12, 32 …}

(7) Compare their spatiogram, similarity (s1, s2).
C = 2*sqrt(2*pi)
C2 = 1/(2*pi)
q = sigma1+sigma2
q = C * (q(1,1,:) .* q (2,2,:)) .^ (1/4)
sigmai = 1./(1./ (sigma1+ (sigma1==0)) + 1./ (sigma2+(sigma2==0)))
Q = C * (sigmai(1,1,:) .* sigmai(2,2,:)) .^ (1/4)
q = permute (q, [1 3 2])
Q = permute(Q, [1 3 2])
x = mu1(1,:) - mu2(1,:)
y = mu1(2,:) - mu2(2,:)
sigmax = 2 * (sigma1+sigma2)
isigma = 1./(sigmax+(sigmax==0))
detsigmax = permute(sigmax(1,1,:) .* sigmax(2,2,:), [1 3 2])
isigmaxx = permute(isigma(1,1,:), [1 3 2])
isigmayy = permute(isigma(2,2,:), [1 3 2])
z = C2 ./ sqrt(detsigmax) .* exp(-0.5 * (isigmaxx.*x.^2 + isigmayy.*y.^2))
Dist = q * Q * z
S = sum {sum (√h1*√h2*dist.)}
If result = 0, very low similarity
Result = 0.9, good similarity
Result = 1, perfect similarity.

Author has presented color image matching methods, which can be used to find the similar value between query image and database image based on histogram and spatiogram whose accuracy will depend on the histogram bin and segmentation methods. This proposed technique also finds the similarity between compressed color image, 90 degree rotate color image and RGB and HSV color image.

conclusion
Various methods for image retrieval are existing which reduces the time for image retrieval. There are many techniques which are used to solve the general problem of image retrieval some uses the color matching using histogram values and other uses texture descriptor analysis and spatiogram and bins. After studying different issues with the algorithms discussed and the different solutions provided by different authors, there is still a scope of improvement. We can improve the algorithms by considering the vagueness of the different parameters used. In our future work we will try to propose a new matching algorithm to improve the performance of existing system.

REFERENCES
[1] Petteri Kerminen, Moncef Gabbouj, "Image Retrieval based on color matching" | [2] Wasim Khan, Shiv Kumar. Neetesh Gupta, Nilofar Khan, "A Proposed Method for Image Retrieval using Histogram values and Texture Descriptor Analysis", International Journal of Soft Computing and Engineering (IJSCE), Volume-I Issue-II, pp. 33-36, May 2011. | [3] Krishna Kumar Pandey, Nishchol Mishra, Hemant Kumar Sharma[1], "Enhanced of color matching algorithm for image retrieval", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, pp. 529-532, May 2011. | [4] Eric Boyer, Aldo Morales, "A Color Merging Algorithm using MATLAB", 2008. | [5] N Ganeshwara rao,Dr. V. Vijaya Kumar, V Venkata Krishna, "Texture Based Image Indexing and Retrieval ". | [6] Rafael C. Gonzalez and Richard E. Woods, "Digital image processing Using MATLAB", 2010. |