Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 1, January 2017, pg.205 – 212

A Methodology to Create a Fingerprint for RGB Color Image

Dr. Rushdi S. Abu Zneit, Dr. Ziad AlQadi, Dr. Mohammad Abu Zalata
Albalqa Applied University
Jordan_Amman

Abstract: Color image retrieval, dolor image recognition methods required to efficient in term of minimizing processing time and memory space to store deferent data structures needed in image processing,
Color Histograms, Histogram Distance Measurements, Color Spaces and Quantization play an important role in retrieving images and image recognition based on similarities.
In this paper a novel methodology is presented for determining color image fingerprint, which can be used for efficient color image processing. Color fingerprint for each color image is a unique one column array can be used instead of the image, thus saving time and memory spaces needed to process color image.
Keywords: RGB color image, HSV color image, fingerprint, and quantization.

1- Introduction

1-1 RGB and HSV color spaces

Color vision can be processed using RGB color space or HSV color space. RGB color space describes colors in terms of the amount of red, green, and blue present. HSV color space describes colors in terms of the Hue, Saturation, and Value. In situations where color description plays an integral role, the HSV color model is often preferred over the RGB model. The HSV model describes colors similarly to how the human eye tends to perceive color. RGB defines color in terms of a combination of primary colors, whereas, HSV describes color using more familiar comparisons such as color, vibrancy and brightness.[1], [2]

The RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue.

HSL and HSV are the two most common cylindrical-coordinate representations of points in an RGB color model. The two representations rearrange the geometry of RGB in an attempt to be more intuitive and perceptually relevant than the Cartesian (cube) representation.

Anyone with a monitor has probably heard of the RGB color space. If you deal with commercial printers, you know about CMYK, and you may have noticed **HSV** in the color picker of your graphics software. HSV is so named for three values—Hue, Saturation and Value. This color space describes colors (hue or tint) in terms of their shade (saturation or amount of gray) and their brightness value.[3-5].

The HSV color wheel is depicted as a cone or cylinder. Some color pickers use the acronym HSB, which substitutes the term "Brightness" for value, but HSV and HSB are the same color model. Photoshop uses HSB.

- Hue is expressed as a number from 0 to 360 degrees representing hues of red (which start at 0), yellow (starting at 60), green (starting at 120), cyan (starting at 180), blue (starting at 240) and magenta (starting at 300).
- Saturation is the amount of gray from zero percent to 100 percent in the color.
- Value (or brightness) works in conjunction with saturation and describes the brightness or intensity of the color from zero percent to 100 percent.

The R, G, B values are divided by 255 to change the range from 0...255 to 0...1:

R' = R/255

G' = G/255

B' = B/255

Cmax = max(R', G', B')

Cmin = min(R', G', B')

Δ = Cmax - Cmin

Hue calculation:

$$H = \begin{cases} 60° \times \left(\frac{G'-B'}{\Delta} mod 6\right) & , Cmax = R' \\ 60° \times \left(\frac{B'-R'}{\Delta} + 2\right) & , Cmax = G' \\ 60° \times \left(\frac{R'-G'}{\Delta} + 4\right) & , Cmax = B' \end{cases}$$

Saturation calculation:

$$S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases}$$

Value calculation: V = Cmax

Table 1 shows an example of RGB colors and there equivalents in HSV color space,

Table 1: RGB to HSV color

Color name	RGB	HSV
Black	(0,0,0)	(0°, 0%,0%)
White	(255,255,255)	(0° ,0%,100%)
Red	(255,0,0)	(0°, 100%, 100%)
Lime	(0,255,0)	(120°, 100%, 100%)
Blue	(0,0,255)	(240°, 100%,100%)
Yellow	(255,255,0)	(60°,100%,100%)
Cyan	(0,255,255)	(180°,100%,100%)
Magenta	(255,0,255)	(300°,100%,100%)
Silver	(192,192,192)	(0°,0%,75%)
Gray	(128,128,128)	(0°,0%,50%)
Maroon	(128,0,0)	(0°,100%,50%)
Olive	(128,128,0)	(60°,100%,50%)
Green	(0,128,0)	(120°,100%,50%)
Purple	(128,0,128)	(300°,100%,50%)
Teal	(0,128,128)	(180°,100%,50%)
Navy	(0,0,128)	(240°,100%,50%)

Figures 1 and 2 shows the RGB color image, RGB color histogram(color distribution) and the equivalents in HSV color space.

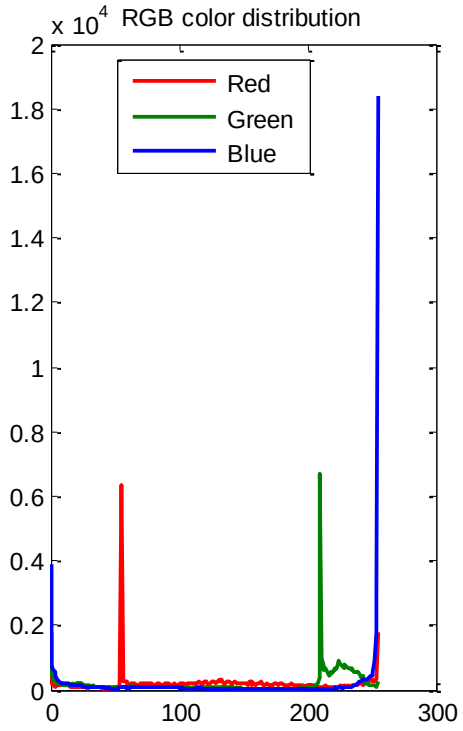
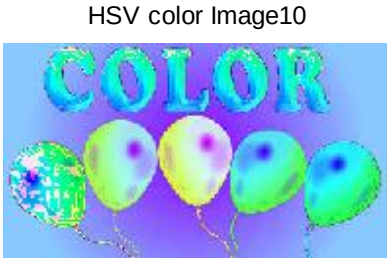


Figure 1: RGB color image



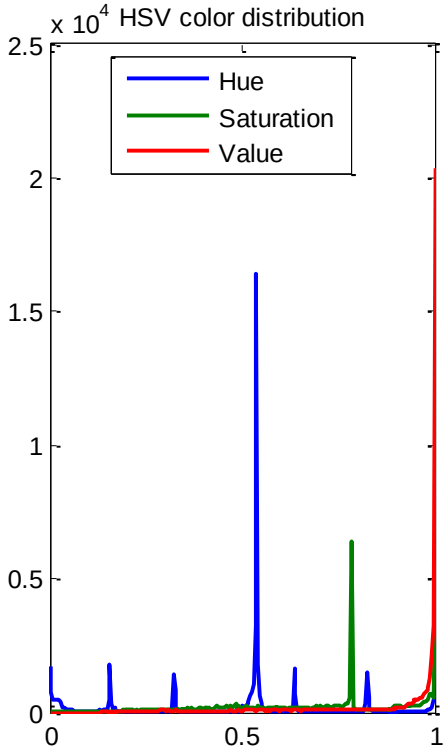


Figure 2: HSV color image

1-2 Color space quantization

A Color space quantization is a process that reduces the number of distinct colors used in an image. The intention of Color quantization is that the new image should be as visually similar as possible to the original image. For a true color image, the number of the kind of colors are up to 2^24 = 16777216, so the direct extraction of color feature from true color will lead to a large computation. In order to reduce the computation, without a significant reduction in image quality, some representative color is extracted, to represent the image, thereby reducing the storage space and enhancing the process speed [6].[7] The effect of color quantization on the performance of image retrieval has been reported by many authors in [8] [9] [10] and [11], with different quantization schemes, like RGB (8X8X8), Lab (4X8X8), HSV (16X4X4), Lu*v* (4X8X8).

The normality of color values in images is stored in an RGB model. The total amount of 24 bits is being used to store the color; 8 bit each for Red, Green, and Blue. It will require a profusely high amount of memory and computation speed. Therefore, quantization detaches the color layers and decreases the amount of data being used. Colors in RGB model are quantized into (8 × 8 × 8) = 512 bins. In HSV color space, quantization of hue requires the most attention. The hue circle consists of the primary color red, green, and blue separated by 120 degrees. A circular quantization at 20 degree steps sufficiently separated the hues such that the three primaries and yellow, magenta and cyan are represented each with three sub-divisions. Saturation and value are each quantized to three levels yielding greater perceptual tolerance along these dimensions. As a consequence, H is quantized to 8 levels and S and V are quantized to 2 levels. Therefore, the quantized HSV space has 8 × 2 × 2 =32 histogram bins which form the color image fingerprint.

2- The proposed methodology

The proposed methodology can be implemented applying the following steps:

1. Get the original RGB color image.
2. Convert RGB color image to HSV color image.
3. Extract H, S, V bands from HSV image,
4. For each band set the number of quantization level(8 for H band and 2 for S and V bands).
5. Quantize HSV bands,
6. Create the fingerprint vector for the selected image;
7. Add the fingerprint victor the database (matrix of victors: one column for each image with 32 values).
8. Save the data base matrix for later use.

3- Implementation and result discussion

The proposed methodology was implemented using matlab, the following function was used to create a fingerprint victor for each RGB color image:

```
function fingerprint = hsvquantize(image)
% input: image to be quantized in hsv color space into 8x2x2 equal bins
% output: 1x32 vector indicating the features extracted from hsv color
% space
[r, c, hsv] = size(image);
% totalPixelsOfImage = rows*cols*numOfBands;
image = rgb2hsv(image);
% split image into h, s & v planes
h = image(:, :, 1);
s = image(:, :, 2);
v = image(:, :, 3);
% quantize each h,s,v equivalently to 8x2x2
% Specify the number of quantization levels.
% quantize each h,s,v to 8x2x2
% Specify the number of quantization levels.
Hquanlevel = 8;
Squanlevel = 2;
Vquanlevel = 2;
% Find the max.
maxValueForH = max(h(:));
maxValueForS = max(s(:));
```



```
maxValueForV = max(v(:));
% create final histogram matrix of size 8x2x2
fingerprint = zeros(8, 2, 2);
% create col vector of indexes for later reference
index = zeros(r*c, 3);
% Put all pixels into one of the "numberOfLevels" levels.
count = 1;
for row = 1:size(h, 1)
 for col = 1 : size(h, 2)
 quantizedValueForH(row, col) = ceil(Hquanlevel * h(row, col)/maxValueForH);
 quantizedValueForS(row, col) = ceil(Squanlevel * s(row, col)/maxValueForS);
 quantizedValueForV(row, col) = ceil(Vquanlevel * v(row, col)/maxValueForV);
 % keep indexes where 1 should be put in matrix hsvHist
 index(count, 1) = quantizedValueForH(row, col);
 index(count, 2) = quantizedValueForS(row, col);
 index(count, 3) = quantizedValueForV(row, col);
 count = count+1;
 end
end
% put each value of h,s,v to matrix 8x2x2
% (e.g. if h=7,s=2,v=1 then put 1 to matrix 8x2x2 in position 7,2,1)
for row = 1:size(index, 1)
 if (index(row, 1) == 0 || index(row, 2) == 0 || index(row, 3) == 0)
 continue;
 end
 fingerprint(index(row, 1), index(row, 2), index(row, 3)) = ...
 fingerprint(index(row, 1), index(row, 2), index(row, 3)) + 1;
end
% normalize hsvHist to unit sum
fingerprint = fingerprint(:)';
fingerprint = fingerprint/sum(fingerprint);
end
```

A database of RGB color images was created, for each image a fingerprint victor was created and added to the fingerprint matrix data base,

Table 2 show some of the images in the \database of RGB color images used in the implementation stage.

Table 3 show the results of implementation, here each column identifies the fingerprint for each image, which has 32 values that form each fingerprint,

From the obtained results shown in table 3 we can raise the following facts:

- ✓ Each RGB color image has a fingerprint column which can be used as an identifier or a key to retrieve or recognize a color image,
- ✓ Each fingerprint has a unique value, thus it is simple to identify an image,
- ✓ The fingerprint victor has a small size comparing with the image size, so using this victor as a key leads to minimize the image processing time and the memory space required to manipulate the image.
- ✓ Fingerprint creation time is very small(as shown in table 4), the average required time for each image pixel is around 0,0008 millisecond.

Table 2: RGB color images treated by the methodology

Image	Size
Image1 	194 x 259 x 3
Image2 	159 x 318 x 3
Image3 	141 x 358 x 3
Image4 	168 x 300 x 3
Image5 	236 x 214 x 3
Image6 	157 x 320 x 3
Image7 	177 x 284 x 3
Image8 	168 x 300 x 3
Image9 	177 x 284 x 3
Image10 	168 x 300 x 3

Table 3: RGB images fingerprints database

Image1	Image2	Image3	Image4	Image5	Image6	Image7	Image8	Image9	Image10
0.0492	0.0203	0.004	0.1226	0.3415	0.1751	0.0071	0.002	0.0001	0.0001
0.0196	0.0274	0.0226	0.0069	0.092	0.0087	0.0044	0.0033	0.0001	0.0002
0.0023	0.0042	0.0271	0.0023	0.003	0.0079	0.004	0.0102	0.0001	0.0002
0.0007	0.0032	0.006	0.0716	0.001	0.0022	0.0037	0.0092	0	0.0002
0.0008	0.0049	0.002	0.0025	0.0001	0.0054	0.0166	0.0029	0	0.0001
0.0012	0.009	0.0036	0.0014	0.0005	0.0032	0.013	0.0054	0	0.0004
0.0024	0.0029	0.0012	0.0012	0.0021	0.006	0.004	0.0087	0	0.0004
0.0061	0.0036	0.0022	0.0027	0.0064	0.016	0.0066	0.002	0	0.0002
0.1927	0.1015	0.0193	0.006	0	0.406	0.0497	0.0091	0.0413	0.0065
0.0338	0.2283	0.3263	0.0039	0.0001	0.0053	0.0182	0.012	0.0102	0.0134
0.0007	0.0088	0.2434	0.0087	0.0007	0.0013	0.0068	0.0062	0.0002	0.0154
0.0001	0.0004	0.0157	0.0113	0.0003	0.0002	0.0084	0.002	0	0.0012
0.0002	0.0052	0.0017	0.005	0	0.0003	0.2878	0.0074	0	0.0011
0.001	0.0018	0.0012	0.0054	0	0.0001	0.0314	0.0119	0	0.0156
0.0007	0.0178	0.0005	0.0017	0	0.0005	0.005	0.3927	0	0.0156
0.0042	0.0052	0.0005	0.0035	0	0.0138	0.0135	0.0099	0.0134	0.0052
0.0605	0.0714	0.0032	0.0115	0.0422	0.1328	0.0614	0.001	0.0358	0.0062
0.008	0.0142	0.0432	0.0058	0.1386	0.0086	0.0063	0.0003	0.0265	0.0038
0.002	0.0042	0.0306	0.0028	0.0812	0.062	0.0006	0.0102	0.0145	0.0038
0.0027	0.0011	0.0041	0.0211	0.0165	0.0206	0.0003	0.006	0.0002	0.0108
0.0097	0.0015	0.0056	0.1559	0.0092	0.0138	0.0003	0.0001	0.0003	0.2037
0.0064	0.0026	0.0034	0.0026	0.0015	0.0103	0.0002	0	0.0003	0.0046
0.0034	0.0019	0.0084	0.0014	0.0086	0.005	0.001	0.2949	0.0016	0.0037
0.0457	0.003	0.0014	0.0042	0.0098	0.0598	0.0038	0.0002	0.034	0.0039
0.526	0.2129	0.0008	0.1927	0.087	0.0321	0.2673	0.0073	0.6099	0.1111
0.0163	0.1021	0.1578	0.0559	0.0336	0.0002	0.1142	0.0009	0.0456	0.061
0	0.0004	0.0625	0	0.0049	0	0	0.0007	0.0017	0.0441
0	0	0.0018	0.0084	0.0587	0	0	0.0001	0	0.0029
0	0.0533	0	0.2314	0	0.0003	0.0633	0.0116	0	0.3488
0	0.0007	0	0	0	0.0015	0.0002	0.0091	0	0.0439
0	0.071	0	0	0	0	0	0.1483	0	0.0444
0.0034	0.0152	0	0.0495	0.0604	0.0011	0.001	0.0146	0.1642	0.0275

Table 4: Time to create a fingerprint

Image	Fingerprint creation time(sec.)
Image1	0.077000
Image2	0.074000
Image3	0.074000
Image4	0.074000
Image5	0.094000
Image6	0.079000
Image7	0.075000
Image8	0.074000
Image9	0.074000
Image10	0.099000
Peppers with size 384x512x 3	0.632000
Football with size 256x320x 3	0.169000

4- Proposed methodology applications

The proposed methodology can be used in various image processing applications such as image retrieval and image processing and here are some recommendations to get the benefits of the proposed methodology:

- To add a new image to the image database perform the following steps:
  - ✓ Get the image and add it to image database
  - ✓ Apply the proposed methodology to create the image fingerprint victor.
  - ✓ Add the fingerprint to fingerprint matrix data file(add column to matrix).
- To retrieve the image from the image database perform the following steps:
  - ✓ Get the image fingerprint.
  - ✓ Search the fingerprint data file to get the column number.
  - ✓ Use a column number as an index to retrieve the image from the image database.
- To recognize the image perform the following steps:
  - ✓ Get the image.
  - ✓ Create the image fingerprint.
  - ✓ Use a recognition tool to get the image index.
  - ✓ Use the index as a reference to the image database.

Conclusions

A proposed methodology for image fingerprint creation was proposed, implemented and tested. It was shown that the fingerprint can be used as an image key, the process of fingerprint creation is very simple and required a small time. The image fingerprint has a unique value and it is possible to use it in image retrieval and image recognition. Using the image fingerprint leads to efficient image processing in term of minimizing processing time and minimizing memory space required for digital image processing.

References

[1] Shao, H., Svoboda, T., Van Gool, L.: ZuBuD — Z¨urich buildings database for image based recognition. Technical Report 260, Computer Vision Laboratory, Swiss Federal Institute of Technology (2003) Database downloadable from http://www.vision.ee.ethz.ch/showroom/.

[2] Majed O. Al-Dwairi, Ziad A. Alqadi, Amjad A. AbuJazar and Rushdi Abu Zneit, Optimized True-Color Image Processing, World Applied Sciences Journal 8 (10): 1175-1182, 2010.

[3] Akram Mustafa and Ziad AlQadi, Color image, reconstruction using a new model. J. Computer. Sci., 5: 250-159, 2009.

[4] Rafael C. Gonzalez and Richard Eugene Woods (2008). *Digital Image Processing*, 3rd ed. Upper Saddle River, NJ: Prentice Hall. ISBN 0-13-168728-X. pp. 407–413

[5] Monika Deswal1 , Neetu Sharma, A Fast HSV Image Color and Texture Detection and Image Conversion Algorithm, International Journal of Science and Research (IJSR), Volume 3 Issue 6, June 2014.

[6] S.Niranjanan, S.P.Raja Gopalan, Performance Efficiency of Quantization using HSV Colour Space and Vector Cosine Angle Distance in CBIR with Different Image Sizes, *International Journal of Computer Applications (0975 – 8887) Volume 64– No.18, February 2013*.

[7] Hafner, J and Sawhney, H. S.-1995 -. Efficient colorhistogram indexing for quadratic form distancefunctions.In *IEEE Transactions on Pattern Analysis and* Machine Intelligence, Intelligence, 17(7): pp.729-736.

[8] Smith, J. R. – 1997 -. Integrated spatial and feature image system: Retrieval, analysis and compression, Ph.D dissertation, Columbia University, New York

[9] Wan, X and Kuo, - 1996 - Image retrieval with multiresolution color space quantization. in Electron.Imaging and Multimedia Syst.

[10] Wan. X and Kuo. K. – 1996 -. Color distribution analysisand quantization for image retrieval. in SPIE Storageand Retrieval for Image and Video Databases IV, vol.SPIE 2670, pp. 9- 16

[11] Zhang, Z., Wenhui, Land Bo, L. – 2009 -. An Improving Technique of Col or Histogram in Segmentation-based Image Retrieval. At Fifih*International Conference on Information Assurance and Security*. IEEE.

