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. 2, February 2017, pg.60 – 66

Window Averaging Method to Create a Feature Victor for RGB Color Image

Dr. Ziad A.AlQadi, Dr. Hussein M.Elsayyed

Albalqa Applied University

Abstract: Various applications now are dealing with huge data bases containing various images with different kinds and their own semantics. Color images usually have a huge size, thus they need a lot of time for processing tasks such as image retrieval, image transmission and image recognition. In order to reduce color image processing time and memory space size needed to accomplish the processing cycle an efficient method of features extraction will be proposed. This method will allow the user to create a feature victor for each color image; this victor will be used as key for color image retrieval or recognition. This key will optimize the processing task because it will have a very small size comparing with the color image size. The proposed method will be tested, implemented and the results of implementation will compared with other method in order to show the efficiency of the proposed method.

Keywords: Color image, window, feature victor, efficiency

1- Introduction

In machine learning process data is recognized using their meaningful features victor and extracted using the similarity between these victors. To find the recognizable victors among the data required to reduce the amount of data and extract the actual relationship or difference between two data instances. These relationships or differences are computed using the content of the data. Therefore that is a complex domain; where uncertainty and randomness nature of the data can be misguide the actual decision or recognition pattern. The proposed method here is an evaluation of techniques by which the optimal properties between the data can be evaluated to find and form the optimum properties by which the nature of data and pattern of the data can be recognize. The proposed method is evaluation of the image data and finding the most appropriate feature extraction method, in order to utilize them in various applications. For proper understanding of the relation between the data processing and image processing, first we take an example, suppose we have a set of random documents, for categorizing or proper arrangement of these documents according to their domain, required to find some knowledge about the document contents, therefore first required to read a document and then evaluate the domains and topic reside in the given document. In the same way for finding the appropriate feature victor over the given data, pre-processing, data model construction and implementation in problem is required. Image is a different kind of data which

includes a huge amount of information, such as color information, objects, edges, pixel definition, dimensions and others [1], [2]. Therefore the treatment of image data is a sensitive concern to preserve the complete information. This paper addresses an efficient method of extracting a feature victor which can be used as a key features and properties of image data by which the information from the image is extracted and utilized for different applications of face recognition, image retrieval and others.**[3], [4].**

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.[5], [6]

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.[7-9].

2- Proposed Method

In [10] and [11] a method of feature extraction based on converting RGB color image to HSV color image and image quantization was proposed and from the results of implementation it was seen that this method provides a unique feature victor(signature) for each RGB color image. Here we will introduce a window averaging method and we will perform a requirement analysis between the two methods to find the efficiency of the new proposed method,

The proposed window averaging method which can be used to create color image features victor(signature) required an implementation of the following steps:

a) Get the original RGB color image.

b) Retrieve the image size(S =rows X columns X 3).

c) Select the size of the features victor(L+ number of elements in the victor), here we choose L+32.

d) Find the widow size (WS = floor(S/32).

e) Victories the original image (reshape the original image to one column array).

f) Segment the victimized image, each segment size equal window size.

g) For each segment find the average value and store it in the features victor.

3- Proposed Method Implementation

The proposed method was implemented using matlab 7 and a PC with the following specification:

Processor:	Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz 2.50 GHz
Installed memory (RAM):	4.00 GB
System type:	64-bit Operating System

The following code was written and implemented using various RGB color images:

```
Clc, clear all
a=imread('C:\Users\win 7\Desktop\faces\f10.jpg');
tic
[n r c]=size(a);
nn=n*r*c;
b=reshape(a,nn,1);
k=floor(nn/32);
for i=1:32
   r1=1+k*(i-1);
   r2=k*i;
   s(i)=mean(b(r1:r2,1));
end
toc
s'
```

!6 RGB color images were selected, each of them represents an image of a human face, one of these images is shown in figure 1.

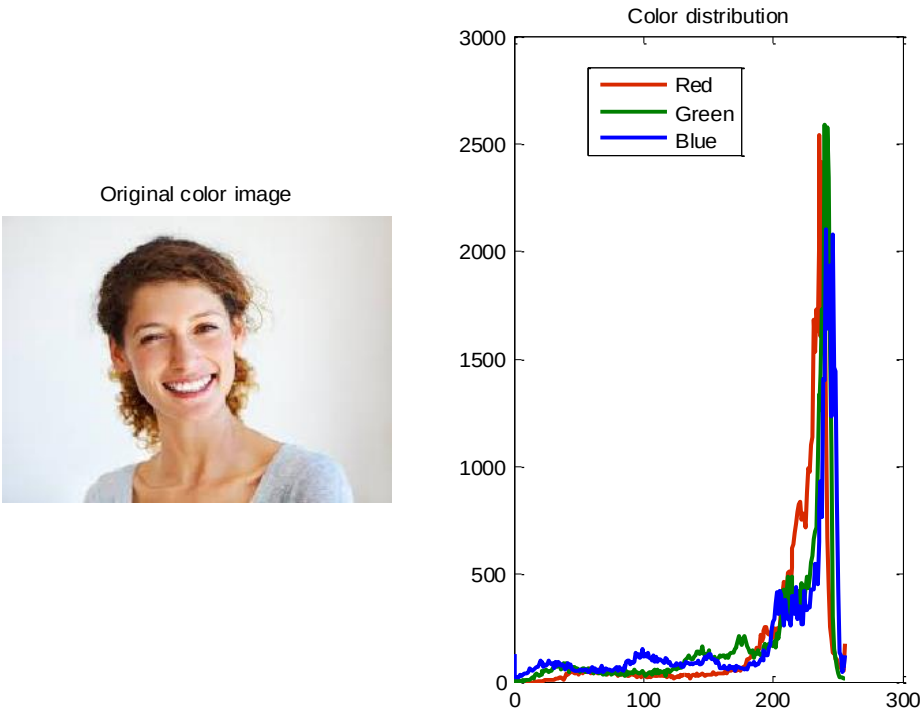

Figure 1: One of the selected images

Table 1 shows the basic information of the selected images.

Table 1: Selected color images basic information.

Image	Red Av, Mean and STD			Green Av, Mean and STD			Blue Av, Mean and STD		
F1	212.5288	228	24.2180	200.6530	228	27.7851	193.8089	225	30.4230
F2	227.6444	252	18.4895	216.2494	251	16.7532	207.6244	250	17.4949
F3	184.6877	209	8.3559	168.3994	161	11.3501	166.8652	147	11.0625
F4	144.0426	156	12.9833	120.7365	114	12.3251	108.5101	89	10.7149
F5	211.0658	213	19.1387	206.5601	222	18.3209	202.6197	228	19.0055
F6	225.4545	242	26.0239	205.2174	233	28.8704	193.8315	224	30.6818
F7	223.9638	254	17.7747	209.4392	254	19.5147	204.4932	253	20.5385
F8	165.7101	181	11.3478	123.7887	123	18.3776	101.3088	92	23.1630
F9	90.7098	101	12.0513	65.9978	74	7.7879	60.4305	65	6.3468
F10	139.0339	136	19.7780	119.2209	130	14.7805	114.4426	120	14.5844
F11	156.5855	174	16.4685	113.2107	136	9.2081	93.4137	109	5.2284
F12	148.1778	144	11.6391	124.8269	128	12.6157	113.1295	111	14.7976
F13	124.8817	127	21.4427	109.0315	118	12.3991	100.7285	111	12.2700
F14	183.0859	203	18.8768	169.9813	174	20.2094	162.0353	161	22.6303
F15	197.5098	215	15.8863	188.9327	211	18.6908	178.4102	193	18.4424
F16	107.2306	77	15.8722	81.6794	62	15.3249	61.6940	50	15.7954

The proposed method was implemented using the selected images, each time we created a feature victor for each of the selected images and table 2 shows the features victor for 10 of the selected images:

Table 2: Images feature victors

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
109.2434	31.7332	221.3488	164.6261	239.7594	254.9114	252.7760	164.5572	92.6467	132.6657
114.3355	45.4461	213.8396	158.1824	236.6559	245.3481	240.6016	138.2354	92.0745	126.3812
124.3862	73.9074	160.2656	110.2786	228.9132	205.2101	229.7543	150.5460	82.6405	109.9857
120.6220	97.3665	160.9681	133.7240	210.7917	209.7829	219.1489	160.2905	91.0052	143.7212
130.3966	108.0017	170.0239	140.1635	182.5031	217.9250	206.6249	169.6009	92.0880	186.1905
122.9108	108.8350	175.2358	143.6556	174.0489	210.5970	195.8411	187.7735	101.6056	193.5541
111.3214	99.3766	173.5050	143.3185	178.5831	196.4851	197.2925	177.7457	93.3492	171.4752
122.6055	81.6348	165.6024	142.5762	214.9767	203.7691	205.7815	171.3264	85.4739	123.6279
128.1709	48.7879	187.7133	122.9741	222.1661	235.8891	232.9650	165.0792	85.9526	90.8431
135.7095	31.4176	205.2616	164.4191	218.2366	254.8760	238.1540	150.8464	90.4855	115.9947
81.4318	41.9754	211.1122	163.8234	223.5913	254.9230	254.9028	198.4719	85.3959	133.4916
56.0465	63.4600	222.5542	150.8809	242.2580	254.9235	248.9735	125.3106	73.6183	135.6479
63.7541	63.3595	198.6366	131.6864	234.9721	225.4931	237.6428	120.3197	69.0825	118.6511
78.0055	74.3103	136.4570	91.1361	230.7765	171.9640	211.5501	104.5676	57.2320	101.3299
65.8725	92.3060	131.5282	109.6391	195.4383	171.5813	183.8950	108.9858	60.7651	113.2698
90.2006	93.1733	137.7853	105.0217	141.1189	172.8739	170.0108	113.7199	63.8586	145.2770
87.9982	90.3038	138.5020	105.8221	141.8133	160.9678	161.8087	120.2131	64.1555	135.4080
62.0116	79.1331	137.5134	107.1308	184.6224	165.0979	172.8882	114.8808	63.0470	119.6753
56.2999	63.9479	145.6945	107.9987	226.3200	196.3511	194.8125	114.4552	58.3796	87.0379
55.4404	54.5932	196.9941	120.8246	227.4174	244.7074	237.8277	120.0244	70.8433	94.5777
65.8418	70.2710	207.9465	154.3571	224.7410	254.9939	244.5838	138.5226	73.2199	130.3673
52.8653	79.8376	219.2538	150.0036	233.7287	254.9163	254.9007	183.4254	70.1934	140.1823
47.6154	79.7233	224.1722	143.5864	240.7524	253.0829	245.0123	103.0376	67.2891	140.1780
55.4587	79.8255	175.2768	105.4469	234.3101	193.5974	231.1406	89.3600	58.2588	112.3197
67.2573	94.8173	128.8730	81.4796	226.5835	149.2309	194.7013	78.9209	53.8364	92.0448
64.1097	106.8069	128.4082	92.5040	156.1979	146.6084	169.1803	82.7224	53.8702	108.3503
55.7348	104.1227	134.6546	89.0028	112.3247	145.8669	155.2904	93.9177	60.9326	131.8689
48.4621	99.9828	129.7315	84.7076	138.9325	138.1965	151.9584	90.8444	58.5624	111.7066
54.5738	87.2143	126.9596	88.7293	203.2326	155.8923	163.7461	86.0677	55.8508	92.8199
51.1153	68.6659	163.4137	89.6168	231.6667	206.3445	211.5390	83.8627	55.7903	86.8477
69.4879	74.4358	206.9624	134.7335	229.8042	253.6663	237.9088	102.1073	67.7068	108.4191
62.0365	91.7710	210.3596	150.1304	228.7164	254.9898	250.7601	158.7807	66.9482	141.5284

From table 2 we can see that each victor is a unique and differs from other victors, thus we can use this victor as a signature or a key to retrieve or recognize the needed color image, this is shown in figure 2:

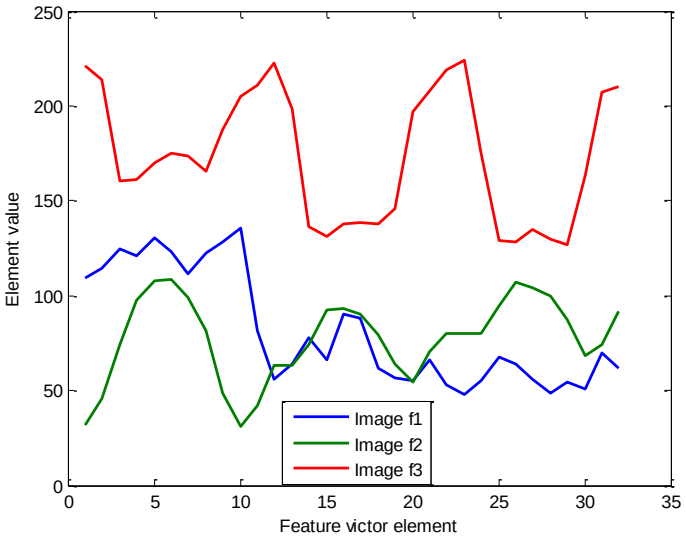

Figure 2: Feature victors for images f1, f2 and f3

4- Requirements Analysis

For performance issues, here we will implement the proposed method and the quantization method proposed in [10[and [11] to find the processing time and the memory space size needed to create a futures victor, We have to notice that the processing time for both methods will vary depending on the PC specification.

Table 3 shows the results of implementation, and from this table we can see that the processing time and memory spaces time for the proposed method are less than those for the quantization method and for all the selected images.

To calculate the efficiency of using this method we have first have to calculate the averages of the processing time and memory space size and from table 3 we can get the efficiency of the proposed method:

Processing time efficiency=:

Average processing time for quantization method/ average processing time for proposed method

=1.5918/0.0084=189.5000

This means that the proposed method is faster than quantization method in 189.5000 times.

Memory space efficiency=:

Average memory space for quantization method/ average memory space for proposed method

=23230630/1452200=15.9969

This means that the proposed method smaller memory space size than quantization method in 15.9969 times.

Table 3: Implementation results

image	size	Quantization processing time(sec.)	Averaging processing time(sec.)	Memory space size(byte) for quantization	Memory space size(byte) for averaging
F1	198x254x3	0.077000	0.004000	4828384	302072
F2	191x264x3	0.056000	0.004000	4841056	302864
F3	255x198x3	0.091000	0.004000	4847392	303260
F4	259x194x3	0.084000	0.005000	4823968	301796
F5	168x300x3	0.084000	0.004000	4838752	302720
F6	275x183x3	0.104000	0.004000	4831552	302270
F7	254x198x3	0.089000	0.004000	4828384	302072
F8	604x499x3	1.426000	0.009000	28934368	1808696
F9	609x499x3	1.483000	0.009000	29173888	1823666
F10	640x640x3	2.138000	0.012000	39321952	2457920
F11	366x550x3	0.610000	0.007000	19325152	1208120
F12	221x221x3	0.091000	0.004000	4689088	293366
F13	1200x1600x3	18.140000	0.046000	184320352	11520320
F14	507x337x3	0.620000	0.008000	16402816	1025474
F15	183x275x3	0.084000	0.004000	4831552	302270
F16	398x284x3	0.292000	0.006000	10851424	678512
Mean		1.5918	0.0084	23230630	1452200
Efficiency		1	189.5000	1	15.9969

Conclusions

A method of crating features victor for any type of color images was proposed, tested and implemented. The experimental results showed that the propose method creates a unique features victor which can suits different applications, such as image retrieval and image recognition.

The proposed method was compared with other method and the experimental results showed that the proposed method is more efficient in saving processing time and memory space size,

References

[1] Gaurav Mandloi, A Survey on Feature Extraction Techniques for Color Images, Gaurav Mandloi / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 4615-4620.
[2] KRYSTIAN MIKOLAJCZYK,TINNE TUYTELAARS, "Local Image Features", Universiteit Leuven, KasteelparkArenberg 10, Leuven,Belgium
[3] Minsu Cho, Jungmin Lee, and Kyoung Mu Lee,"Reweighted Random Walks for Graph Matching", Computer Vision–ECCV 2010, 2010 – Springer
[4] PritiMaheswary, Dr.NamitaSrivastava, "Retrieval of Remote Sensing Images Using Color & Texture Attribute",(IJCSIS) International Journal of Computer Science and Information Security,Vol. 4,No. 1&2, 2009.
[5] 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/.
[6] 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.
[7] Akram Mustafa and Ziad AlQadi, Color image, reconstruction using a new model. J. Computer. Sci., 5: 250-159, 2009.
[8] 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
[9] 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.
[10] Dr. Rushdi S. Abu Zneit, Dr. Ziad AlQadi, Dr. Mohammad Abu Zalata, A Methodology to Create a Fingerprint for RGB Color Image, IJCSMC, Vol. 6, Issue. 1, January 2017, pg.205 – 212.
[11] Dr. Ghazi. M. Qaryouti, Prof. Ziad A.A. Alqadi, Prof. Mohammed K. Abu Zalata, A Novel Method for Color Image Recognition, IJCSMC, Vol. 5, Issue. 11, November 2016, pg.57 – 64.

