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.190 – 196

Extracting Individual Objects from RGB Color Image

Dr. Ziad AlQadi, Dr. Rushdi S. Abu Zneit, Dr. Mohammad Abu Zalata
Albalqa Applied University
Jordan-Amman

Abstract: Deferent applications that are dealing with color digital image processing require retrieving information about the objects included in the RGB color image such as object size, object center. This information is very valuable to a user interested in image processing to extract an object needed to the user or a group of objects by eliminating other groups from the image.

This paper will propose a matlab method to process a color image in order to extract the necessary object based on the information retrieved from the method implementation. The method will be implemented and tested using real RGB color images.

Key words: RGB color image, connectivity, object, area, center.

1- Introduction

Digital image is a two (gray image, binary image) or three dimensional matrix (RGB color image) [1], [2]. The matrix element is called a pixel and some pixels may have the same features, or some of the pixels form an object which is defined as a set of connected pixels [3]. In General an object within an RGB color image is a set of pixels which have the same values from a selected rang and are *4-neighbor* or 8- *neighbor* connected.[3][4],[5].

The notation of pixel connectivity describes a relation between two or more pixels [6]. For two pixels to be connected they have to fulfill certain conditions on the pixel brightness and spatial adjacency.

First, in order for two pixels to be considered connected, their pixel values must both be from the same set of values *V*. For a grayscale image, *V* might be any range of gray levels, *e.g. = {22, 23,...40}*, for a binary image we simple have *V={1}*.

To formulate the adjacency criterion for connectivity, we first introduce the notation of *neighborhood*. For a pixel *p* with the coordinates *(x, y)* the set of pixels given by:

$$N_4(p) = \{(x+1, y), (x-1, y), (x, y+1), (x, y-1)\}$$

is called its *4-neighbors*. Its *8-neighbors* are defined as

$$N_8(p) = N_4 \cup \{(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)\}$$

From this we can infer the definition for *4*- and *8*-*connectivity*:

Two pixels *p* and *q*, both having values from a set *V* are *4*-connected if *q* is from the set $N_4(p)$ and *8*-connected if *q* is from $N_8(p)$.

General connectivity can either be based on *4*- or *8*-connectivity; for the following discussion we use *4*-connectivity [7] and [8].

A pixel *p* is connected to a pixel *q* if *p* is *4*-connected to *q* or if *p* is *4*-connected to a third pixel which itself is connected to *q*. Or, in other words, two pixels *q* and *p* are connected if there is a path from *p* and *q* on which each pixel is *4*-connected to the next one[9].

A set of pixels in an image which are all *connected* to each other is called a *connected component*. Finding all connected components in an image and marking each of them with a distinctive label is called connected component labeling.

Figure 1 illustrates an example of a binary image with two connected components which are based on *4*-connectivity can be seen in. If the connectivity were based on *8*-neighbors, the two connected components would merge into one.

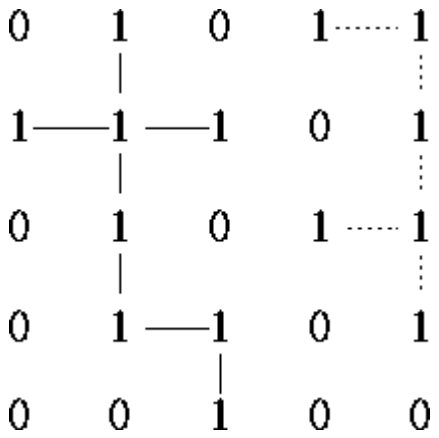

Figure 1 Two connected components based on *4*-connectivity.

2- The proposed method

Color is a powerful descriptor that simplifies object identification, and is one of the most frequently used visual features for content-based image retrieval [10] and [11].

The proposed method can be implemented in 2 phases:

- First phase: Retrieving objects information.
- Second phase> Using the retrieved information to extract a necessary object
  - ✓ Retrieving object information

This phase can be implemented applying the following steps:

1. Get the input RGB color image.
2. Convert the color image to gray image.
3. Convert the gray image to binary image.
4. Remove noises and unnecessary small objects from the image.
5. Label connected components.
6. Measure the properties of each object.
7. Determine the objects.
8. Retrieve the information for each object

Extracting the necessary object: This phase can be implemented applying the following steps:

1. From the obtained information select the object properties such as area range or center.
2. Label the object.
3. Apply the selected range to get the object.

3- Implementation and results

The following matlab code was written to implement the proposed method:

```
% 1- Get The input image
imagen=imread('C:\Users\win 7\Desktop\im2.jpg');
% 2- Convert to gray scale
if size(imagen,3)==3 % RGB image
imagen=rgb2gray(imagen);
end
% 3- Convert to binary image
threshold = graythresh(imagen);
imagen =~im2bw(imagen,threshold);
% 4- Remove all object containing fewer than 30 pixels
imagen = bwareaopen(imagen,30);
% 5- Set the range
imagen1 = bwareaopen(imagen,1199);
imagen2=bwareaopen(imagen,1201);
imagen=imagen1-imagen2;
% 6- Label connected components
[L Ne]=bwlabel(imagen);
Ne
% 7- Measure properties of image regions
propied=regionprops(L,'BoundingBox');
prop=regionprops(L,'Area','Centroid','Extrema','Image');
hold on
% 8= Plot Bounding Box
for n=1:size(propied,1)
rectangle('Position',propied(n).BoundingBox,'EdgeColor','g','LineWidth',2)
end
hold off
pause (1)
% 9= Objects extraction
figure
for n=1:Ne
    [r,c] = find(L==n);
n1=imagen(min(r):max(r),min(c):max(c));
dd=prop(n).Area;
figure,imshow(~n1);
title(['Size in pixel: ',num2str(dd),' ', 'Object #:',num2str(n)])
pause(0.5)
end
Ne
```


The following image shown in figure 2 was taken as an example

INPUT IMAGE WITH NOISE

Figure 2: Input color image

Applying phase 1 of the proposed method we can obtain the following results:
Figure 3 shows the image after removing noise and labeling the objects within the image.
Figure 4 shows the 20 objects in the original image.
Table 1 and 2 show properties for each object, in table 1 we add a column to indicate the rage of areas which can be used to extract each individual object.

INPUT IMAGE WITHOUT NOISE

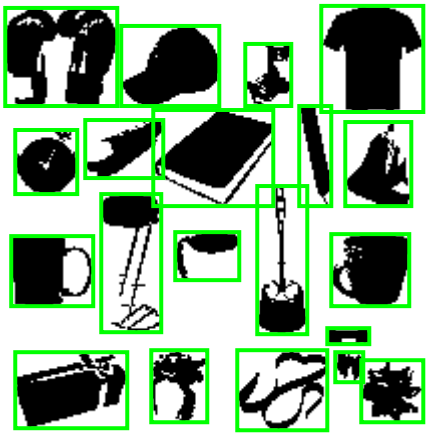

Figure 3: Labeling the objects.

Figure 4: All objects within the image.

Table 1: Area property for each object

Object number	Area in pixels	Extraction range Lower bound<area<upper bound
1	1742	17421<area<1743
2	963	962<area<964
3	661	660<area<662
4	1190	1189<area<1191
5	475	474< area<476
6	644	643< area<645
7	1200	1199< area<1201
8	444	443< area<445
9	1389	1388< area<1390
10	292	291< area<293
11	542	541< area<543
12	315	314< area<316
13	565	564< area<566
14	347	346< area<348
15	1678	1677< area<1679
16	138	137< area<139
17	1041	1040< area<1042
18	123	122< area<124
19	693	692< area<694
20	521	520< area<522

Table 2: center property for each object

Object number	X coordinate of the object center	Y coordinate of the object center
1	32	28
2	22	136
3	25	84
4	35	197
5	64	76
6	67	120
7	89	36
8	90	191
9	109	78
10	104	125
11	144	196
12	136	40
13	142	148
14	159	77
15	187	29
16	175	170
17	188	136
18	175	184
19	191	89
20	196	197

Taking the area range (for example for object 7) from table 1(1199< 1200<1201) and applying phase 2 of the method we can label and extract object 7 from the image as shown in figures 5 and 6.

Figure 5: Labeling object 7

Figure 6: Extracted object 7

Conclusions

A simple method for extracting objects from color image was propose, implemented and tested,
The obtained results shows that the proposed method is valuable to users dealing with digital image processing application, and it is very efficient because it can be used in various ways and for various application and it suits objects extraction, group of object extraction, individual object extraction and object features extraction.

References

[1] *Ziad Alqadi*,A Novel Methodology for Repairing a Torn Image, International Journal on Communications Antenna and Propagation - August 2011 (Vol. 1 N. 4).
[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 ISSN 1818-4952.
[3]: Akram A. Moustafa and Ziad A. Alqadi, Color Image Reconstruction Using A New R'G'I Model, Journal of Computer Science 5 (4): 250-254, 2009 ISSN 1549-3636, 2009 Science Publications.
[4] G. Toussaint, Course Notes: Grids, connectivity and contour tracing (PostScript)
[5] T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, Maryland, 1982.
[6]: A. Dumitrescu and J. Pach. Pushing squares around. In Proceedings of the 20th Annual ACM Symposium on Computational Geometry (SoCG2004), pages 116–123, 2004.
[7] Fingerprint minutiae extraction based on principal curves, Pattern Recognition Letters vol. 28, issue 16, 1 December 2007, pp. 2184-2189.
[8] Employing Generic Algorithms for precise Fingerprint Matching based on Line Extraction, GVIP journal, vol. 7, issue 1, April, 2007.
[9] Nitin Jain & Dr. S. S. Salankar, Color & Texture Feature Extraction for Content Based Image Retrieval, IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) e-ISSN: 2278-1676, p-ISSN: 2320-3331 PP 53-58,2014.
[10] Choi H, Baraniuk R., Multiscale : Image segmentation using wavelet-domain hidden Markov models, IEEE Transaction on image processing, 10(9), pp.1309-1321 (2001).
[11] Mazen A.Hamdan, Prof. Ziad A.Alqadi, Bassam M.Subaih, A Methodology to Analyze Objects in Digital Image using Matlab, *IJCSMC, Vol. 5, Issue. 11, November 2016, pg.21 – 28*.

