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. 5, Issue. 11, November 2016, pg.21 – 28

A Methodology to Analyze Objects in Digital Image using Matlab

Mazen A.Hamdan[1], Prof. Ziad A.Alqadi[2], Bassam M.Subaih[3]

[1, 2, 3] Albalqa Applied University Jordan, Faculty of Engineering Technology
Department of Computer Engineering

Abstract: A simple, useful, and effective methodology will be proposed, implemented and tested. This methodology deals with the objects within the image, and it will be used for object analyzing and object manipulation, which is an important process of digital images and it can suit deferent human applications.

Key words: Digital image, Object, Connectivity, Object area, Object center.

I- Introduction

A large number of images in digital format are generated by deferent users and institutions every day. Consequently, how to make use of this huge amount of images effectively becomes a challenging problem [8]. In different fields of digital image processing applications, it still remains a challenging task to segment objects from its background and count them automatically [9]. The differences between the objects within a digital image lie on the texture, color, size, location and morphology of objects. Many digital image processing applications require object treatment such as : object counting, object indexing and labeling, object detection and extraction, object specification such as object size, object grouping object depletion and so on. So the need of a simple, easy, swift and effective process for object counting is Very important and necessary process of life and need to human applications.

II- Theoretical background

Digital image is a two or three dimensional matrix [1],[2], and can be classified to: binary, gray, or RGB color images. The matrix element is called a pixel and some pixels may have the same features, or some of the pixel form an object which is defined as a set of connected pixels [3].

© 2016, IJCSMC All Rights Reserved

In binary valued digital imaging, a pixel can either have a value of 1 -when it's part of the pattern- , or 0 -when it's part of the background- i.e. there is no grayscale level. (We will assume that pixels with value 1 are black while zero valued pixels are white) [4]. In order to identify *objects* in a digital pattern, we need to locate groups of black pixels that are "connected" to each other. In other words, the *objects* in a given digital pattern are the *connected components* of that pattern.

In general, a *connected component* is a set of black pixels, P, such that for every pair of pixels p_i and p_j in P, there exists a sequence of pixels $p_i, ..., p_j$ such that[5]:

- all pixels in the sequence are in the set P i.e. are black, and
- every 2 pixels that are *adjacent in the sequence* are "neighbors"

When can we say that a given set of black pixels is *4-connected?* First, we have to define the concept of a *4-neighbor* (also known as a *direct-neighbor*):

Definition of a 4-neighbor:

A pixel, Q, is a *4-neighbor* of a given pixel, P, if Q and P share an edge. The 4-neighbors of pixel P (namely pixels P2,P4,P6 and P8) are shown in *Figure 1* below.

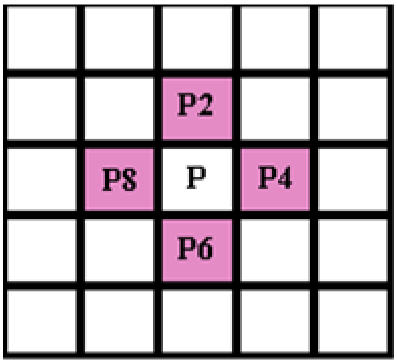


Figure 1: 4-connected pixels

Definition of an 8-connected component:

A set of black pixels, P, is an *8-connected component* (or simply a *connected component*) if for every pair of pixels p_i and p_j in P, there exists a sequence of pixels p_i, ..., p_j such that[6],[7]:

- all pixels in the sequence are in the set P i.e. are black, and
- every 2 pixels that are *adjacent in the sequence* are *8-neighbors*

Object counting algorithms [5], [6],[7]usually take two consecutive images as input and return the locations where differences are identified. The aim of such an algorithm is to locate only the changes that are due to structural changes in the scene, i.e. a object counting.
The change detection algorithms implemented in these video systems provide low-level information that can be used by higher level algorithms to determine the information desired (the trajectory of an object, the control of traffic flow, etc).

Object counting and extraction from the fixed background in the analyzed scene is mostly done by simple subtracting the current image and background image.

III- Matlab functions

The main matlab functions which can be used to manipulate the objects within the image are:

- rgb2gray: converts RGB color image to gray.
- im2bw: Converts image to binary image, based on threshold.
- imfill: Fill image regions and holes.
- bwareaopen: removes small objects from the image.
- bwlabel: Label connected components in 2-D binary image.
- regionprops: Measure properties of image regions such as area and coordinates.

IV- Objects manipulation methodology

The methodology of manipulating objects within a digital image can be implemented applying the following steps:

1. Get the original input image.
2. If the image is color image, convert it to gray image, then to binary image.
3. Remove noise from the image by removing small objects.
4. Apply the function bwlabel for Label connected components to retrieve the number of objects and a label matrix which points to the objects.
5. Measure properties of image regions to get objects information such as area and coordinates.
6. Extract objects or individual objects.

The following matlab code can be used to implement this methodology and it can be executed in various forms.

```
close all,clear all,clc

%% Object mannipulation

%% Get the image

imagen=imread('C:\Users\oday\Desktop\countobjects\im4.png');

figure(1)

imshow(imagen);

title('INPUT IMAGE WITH NOISE')
```

© 2016, IJCSMC All Rights Reserved

```
%% Convert to gray scale
if size(imagen,3)==3 % RGB image
  imagen=rgb2gray(imagen);
end
%% Convert to binary image
threshold = graythresh(imagen);
imagen =~im2bw(imagen,threshold);
%% Remove all object containing fewer than 30 pixels
imagen = bwareaopen(imagen,30);
%% Show image binary image
figure(2)
imshow(~imagen);
title('INPUT IMAGE WITHOUT NOISE')
%% Label connected components
[L Ne]=bwlabel(imagen);
%% Measure properties of image regions
propied=regionprops(L,'BoundingBox');
prop=regionprops(L,'Area','Centroid');
hold on
%% Plot Bounding Box
for n=1:size(propied,1)
  rectangle('Position',propied(n).BoundingBox,'EdgeColor','g','LineWidth',2)
end
hold off
pause (1)
%% 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
```

V- Implementation

The methodology of object analyzing can be useful to analyze the objects within a color image, or a gray image, or a binary image, and it can be implemented in deferent ways in order to get the necessary information needed to a selective analysis.

As an example we took a color image shown in figure 2 as an input image for the methodology, applying this methodology we can get the image shown in figure 3.

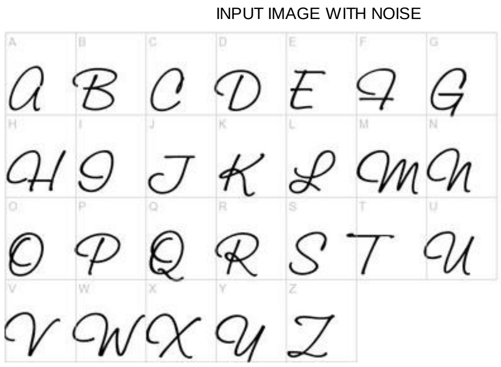


Figure 2: Input color image (Example)

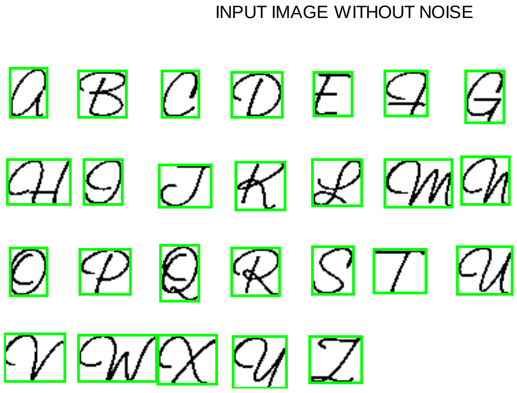


Figure 3: Image after removing noise.

As a results of implementation we can retrieve the necessary information such as :number of object(for this example the number=26), area in pixels for each object, center coordinates for each object, such information are listed in table 1.

We can also extract any object from the image and the corresponding information as shown in figure 4.

Table 1: Objects information

Object number	Area(pixel)	X coordinate	Y coordinate
1	213	20.2911	237.6808
2	271	25.2472	111.2915
3	194	19.0722	47.4742
4	262	15.6832	176.3817
5	269	73.1487	44.3048
6	347	81.8242	240.9942
7	227	73.8987	173.4714
8	198	72.9293	111.6717
9	258	132.1434	240.0659
10	177	132.2373	113.1073
11	316	127.4778	177.9525
12	172	130.0233	46.7151
13	235	185.4809	45.9447
14	261	184.5670	175.3448
15	280	187.2821	242.0429
16	221	185.8778	113.1403
17	230	242.6391	242.4304
18	238	243.5336	113.6387
19	178	240.9551	176.1348
20	176	235.9659	45.0909
21	140	288.2429	169.9286
22	210	295.7810	45.3476
23	390	305.2769	112.0487
24	268	353.5261	175.4552
25	263	354.1103	110.6806
26	241	352.4523	48.5187

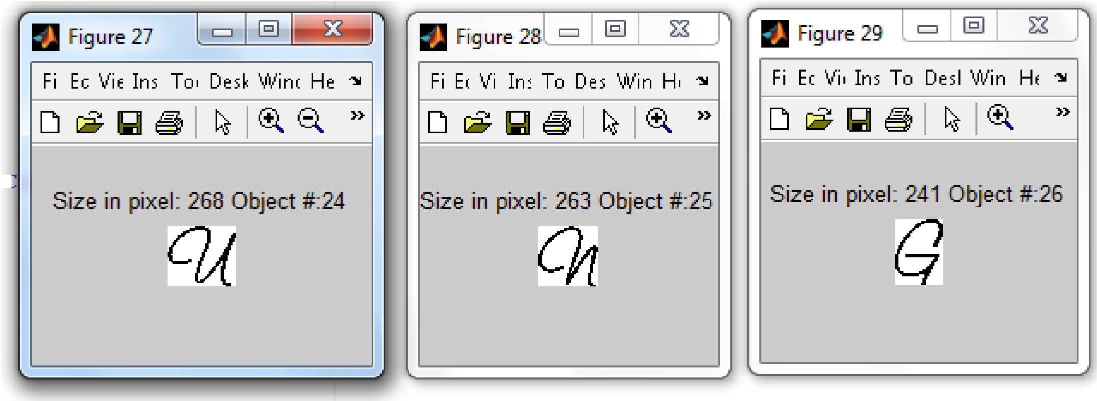


Figure 27
Fi Ec Vie Ins To Desk Winc He
Figure 28
Fi Ec Vi Ins To Des Win H
Figure 29
Fi Ec Vi Ins To Desl Win He

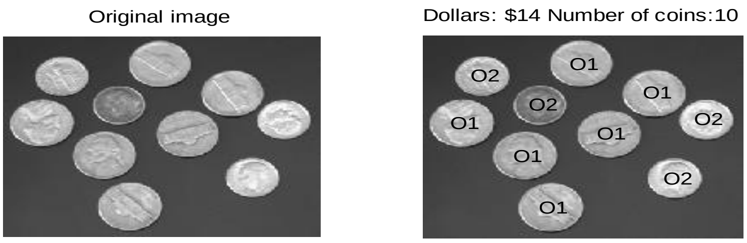
Size in pixel: 268 Object #:24

Size in pixel: 263 Object #:25

Size in pixel: 241 Object #:26

Figure 4: Objects 24, 25, and 26(example).

As it was mentioned before the methodology can be implemented in various ways, figure 5 shows the results of labeling equal objects using the image coins.png as an input image.


Original image

Dollars: $14 Number of coins:10

Figure 5: Labeling the objects.

VI-Conclusions

The proposed methodology is simple , easy, swift and effective process to be used in various image processing applications and it can treat the objects within the image deferent ways supplying the user with the following:

- Object identification
- Counting the number of objects.
- Grouping the same objects.
- Labeling the objects.
- Retrieving needed information for each object

- Extracting objects
- Deleting object.

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] G. Toussaint, Course Notes: Grids, connectivity and contour tracing (PostScript)

[4] T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, Maryland, 1982.

[5] Sumeet Chourasiya, G Usha Rani, Automatic Red Blood Cell Counting using Watershed Segmentation, Sumeet Chourasiya et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 4834-4838.

[6] Jayme Garcia Arnal Barbedo, “Method for Counting Microorganisms and Colonies in Microscopic Images,” 12th Int. Conf. Computer Science and Its Applications, June 2012. pp. 84-87.

[7] Sharif, J. MISWAN, et al. "Red blood cell segmentation using masking and watershed algorithm: A preliminary study." Biomedical Engineering (ICoBE), 2012 International Conference on. IEEE, 2012.

[8] Lehmann T.M., Wein B., Dahmen J., Bredno J., Vogelsang F. & Kohnen M. : Content based image retrieval in medical applications : a novel multi step approach. International Society for Optical Engineering (SPIE), 3972, pp.312-320.(2000).

[9] Choi H, Baraniuk R., Multiscale : Image segmentation using wavelet-domain hidden Markov models, IEEE Transaction on image processing, 10(9), pp.1309-1321 (2001).