Publication of Little Lion Scientific R&D, Islamabad PAKISTAN
Journal of Theoretical and Applied Information Technology
15th April 2011. Vol. 26 No.1
© 2005 - 2011 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MODIFIED NEW ALGORITHM FOR SEED FILLING

[1]MALIK SIKANDAR HAYAT KHAYAL, [2]AIHAB KHAN, [3]SABA BASHIR, [4]FARHAN HASSAN KHAN, [5]SALMA ASLAM,

[1,2,5]Computer Science Department, Fatima Jinnah Women University, Rawalpindi, Pakistan.
[3,4]Computer Engineering Department, National University of Science & Technology, Islamabad, Pakistan.

E-mail: aihabkhan@yahoo.com , m.sikandarhayat@yahoo.com , saba.bashir3000@gmail.com , mrfarhankhan@gmail.com , salma_kashmir12@yahoo.com

ABSTRACT

Seed filling is basic technique of image morphology. The goal of the seed filling operations is to find the seed point and fill the object contours starting from that seed point. In image processing area, the main and most important problem is region filling. This technique is used for image analysis. There are many algorithms for filling but the proposed algorithm can be used to find seeds dynamically. Moreover, proposed algorithm is more efficient as compared to existing techniques. Seeds are found very fast. During filling process the whole region is filled inside the contour and the proposed algorithm does not cross the boundary of contour.

Key words: *Contour filling, **i**mage processing, image morphology, seed filling.*

1. INTRODUCTION

In image processing, region filling is used for image analysis and in computer graphics, a specific area of image can be painted out by using this technique.

There are two parts for seed filling algorithm. First part of this algorithm finds starting seed and connected pixels are discovered in second part. Seed points can be searched automatically or they can be chosen by the user. A working memory is needed to keep the record of all seed filling algorithms.

A seed fill is the operation of image processing where pixels in some region of an image are assigned a label. Seed fill process has multiple names. In graphics community, the terms seed fill and flood fill are used for this process. In image morphology this operation is called binary reconstruction and grayscale reconstruction. In image analysis this operation is most important technique.

The division of this paper is as follows: Related work is given in Section 2. Framework has been described in Section 3. Section 4 describes technique. Results have been shown in section 5. Finally Section 6 gives the Conclusion and future work.

2. RELATED WORK

There are many algorithms for filling the image and one of them is linear-time algorithm [1]. Seed filling and edge filling are two main types among many region filling algorithms. A recursive search method is used to implement seed fill or flood fill methods.

A linear time algorithm is proposed to overcome the deficiencies of previous algorithms. This algorithm fills regions inside contour in raster form, the contour pixels or the 8- connected chain-code. In Fig (1) 8-connectivity is described. Linear time algorithm relies on contour labeling and filling. For contour tracing two techniques are used. One is edge-following and the second is border-following. [1]

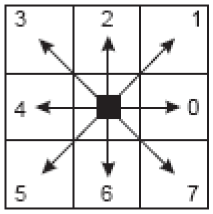

Fig. 1: Adjacency of neighbors

This algorithm has low computation cost and fills correctly. These types of algorithm are dependent on the 8-connected contour tracing for their operations. The correctness of the algorithm is still not proven and it is hard to find the failure cases of the tracing algorithm. The overlapping of the object of interest can be handled by this algorithm. [1]

Another algorithm is O $(p \times d)$-time algorithm for region filling [2]. Given a contour, represented by p bin codes with their corresponding neighbor information and denotes the depth of the corresponding bin tree of the given bin codes. There are many region filling algorithms such as region filling algorithms on raster graphics, region filling on binary raster images and algorithm for filling the region represented by a quad tree. But these region filling algorithms are only used for simple region filling and do not include the holes during filling. Proposed $O(p \times d)$-time algorithm overcome this deficiency. An efficient algorithm for region filling on the bin codes is presented in [2]

For region count, another algorithm is presented in [3]. This algorithm improves the deficiencies of seed fill algorithm such as column method and recursive method. The draw back of algorithm is that the processing time depends on the size and geometry of the input image.

3. PROPOSED FRAMEWORK

Keeping in view the deficiencies of the algorithm mentioned in related work, a new algorithm is proposed. It describes that once image is acquired then contour beginning point is found which is basically seed, from where filling is to be started. Then we find contour of an image which is starting point of the seed. Classify contour point to filling starting point and ending point. After classifying, filling is start from starting point to ending point. The steps involved in the implementation phase are shown in figure 2.

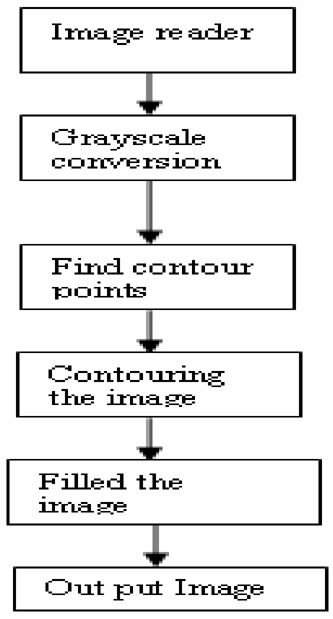

Fig 2: Filling process

4. PROPOSED TECHNIQUE

The proposed technique is given as follows:

4.1 Image Acquisition

The most important aspect of proposed technique is image acquisition

uigetfile function is used to retrieve more than one files .

[FileName, PathName] =uigetfile ('*.fmt');

Concatenation function strcat is used and store in variable tltPath.

tltPath=strcat (PathName,FileName);

image is read by using function imread.

```
Im_org=imread (tltPath)
If  im_org is RGB
Convert RGB to gray
else
Take im_org
```

To convert grayscale rgb2gray function is used.

Im_org=rgb2gray(im_org)

4.2 Find Seed Point

To find the seed point automatically and efficiently, proposed algorithm does not go outside of the region. The scanning process is only bound inside the filling regions. This algorithm scans the region once. In this algorithm, none of the pixels out of the region needed to be scanned and seeds are found very fast.

4.3 Contour points

To implement proposed algorithm for seed filling, first of all contour points are find. Identify the starting, skipping and ending points for filling. Filling starting point is marked by start while ending points are marked by nonc and skipping points are marked by offc. For this purpose sum_dalte_y is calculated. Sum_dalta_y is the sum of offsets in y direction. Consider (x_i,y_i) a contour point then there are following three conditions:

a) If sum_delta_y>0; (xi,yi) filling starting point,
b) If Sum_delta_y<0; (xi,yi) filling ending point
c) If sum_delta_y =0; (xi,yi) filling skipping point.

This algorithm also describes new method to find contour points. New method gives some formulas for finding starting, ending and skipping point.

Filling starting point is marked by start ending points are marked by nonc and skipping points are marked by offc.

initialization of number of bit per pixels.

num_bits_per _pixel = 8;

offc range

offc=2^num_bits_per_pixel-3

'for loop' 2 to 2^num_bits_per_pixel-3.

Values are assigned to offc one by one.

Publication of Little Lion Scientific R&D, Islamabad PAKISTAN
Journal of Theoretical and Applied Information Technology
15th April 2011. Vol. 26 No.1
© 2005 - 2011 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.4 Find outer and inner contour

After finding out starting, ending and skipping point, inner and outer contours are found. For an outer contour, filling starts at the filling starting point and finishes at the filling ending point that is from left to right. For inner contour filling also starts at filling starting point and finishes at the filling ending point but at this time from right to left. For finding inner and outer contour, sum of all angles are calculated. If sum_angle>0, contour is an outer contour otherwise it is inner contour.
contouring the input image imcontour function is used.

[im_cntr]=imcontour (im_gray,255)

which take input image and number of levels as an input.
Classify into starting point and ending point:

'For loop' 1 to n

conditions for classifying

```
If image is offc+1|| offc+2
 image= startc;
else if image== offc-1 || offc-2)
image= nonc;
```

4.5 Classify into starting point and ending point

After finding out outer contour and inner contour classify contour points to filling starting point and filling ending point. Starting point is marked by start and ending point is marked by nonc.
ending point

nonc=offc+2+1

starting point

Start= offc-2+1

4.6 Filling from starting to end point:

After classifying, execute filling from filling starting point to ending point. For outer contour filling is done from left to right and for inner contour filling is done from right to left.
contourf function is used for filling

[im_cntrf,h]=contourf(im_gray)

Output image

Filling image obtained

5. EXPERIMENTAL RESULTS

Proposed algorithm is applied on the original image for contour filling. The algorithm depends on the shape and size of an image. After contouring process, image is filled from starting point to ending point. Results show that original

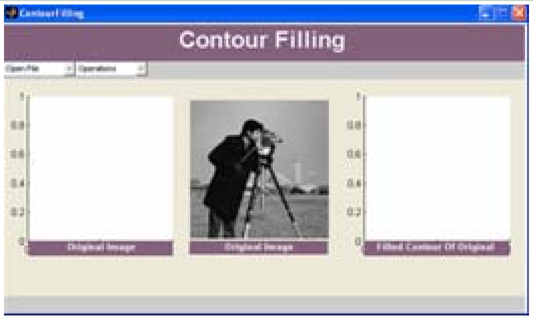

Fig. 3: Original image

image and filled image are nearly same. The screen shot in Fig. 3 shows the original image cameraman.tif. MATLAB provides special functions for reading image data from graphics file formats. After reading an original image, the contouring process finds the contours of the original image according to its axis and object shape. To find the contour of the image, first of all contour points are find. New algorithm finds the contour point automatically. The resultant image after contouring process shows only the contours of input image. Fig. 4 shows the results after contouring process.

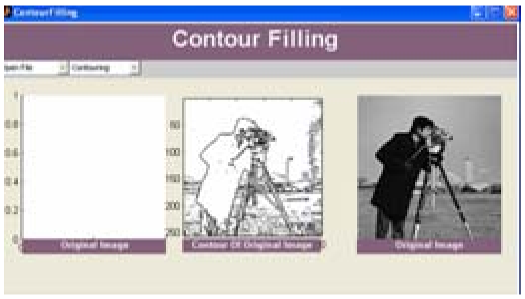

Fig. 4: Finding contours of image

After finding the contour of input image, filling process is start. Contour filling is applied to the contoured image. Filled image is approximately equal to the input image. Fig 5 shows the filled image.

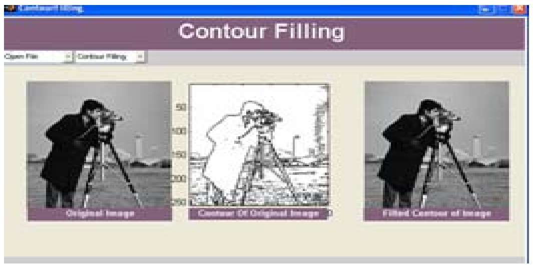

Fig. 5: Original and filled image

Publication of Little Lion Scientific R&D, Islamabad PAKISTAN
Journal of Theoretical and Applied Information Technology
15th April 2011. Vol. 26 No.1
© 2005 - 2011 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The result of JPG image is shown below. This image is firstly converted into grayscale by using rgb2gray function. I = rgb2gray (RGB) converts the true-color image RGB to the grayscale intensity image. Fig. 6 shows the original image, its contouring and filled image. In this image the result of filled image also approximately similar to filled image.

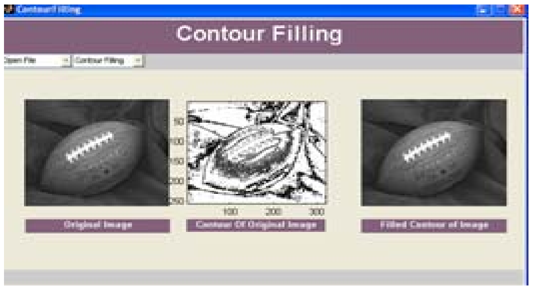

Fig. 6: Original image, its contouring and filled image

Fig. 7 shows contouring and filling of input image. Filled image is same like as input image.

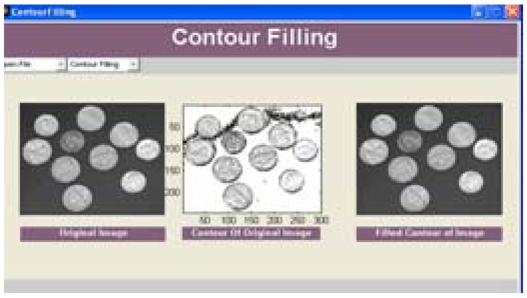

Fig. 7: Contouring and filling of input image

Fig. 8 shows original image, contoured image and filled image. The result of original image and filled image is nearly same.

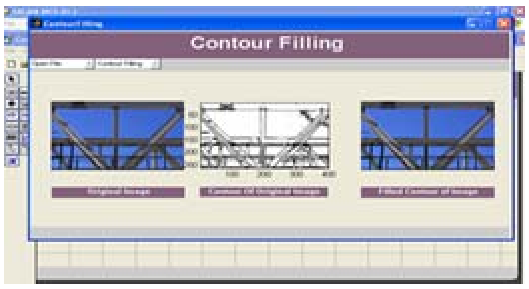

Fig. 8: Result of original and contoured image

6. CONCLUSION & FUTURE WORK

The goal of seed filling operation is to find the seed point and fill the object contours starting from that seed point. In proposed system, seed point is basically the starting point. Seed point can be found by the user manually in most of the existing techniques. To find the seed is difficult process in most of the applications. The proposed algorithm finds the seed automatically and fills the contours starting from that point. Seed points can be found very fast. After finding seed point, the contouring process starts which contours the object boundaries according to its shape and size. After contouring process, the filling of the contour is performed in which the whole region is filled inside the contour. During filling, this algorithm does not cross the boundary of contour. Results show that filled images are nearly equal to the input images as well as proposed algorithm is more efficient as compare to existing techniques.
The proposed algorithm depends upon the shape and size of images. For future work this can be made independent of shape and size.

REFERENCES:

[1] Marius C. Codrea*, Olli S. Nevalainen, “An algorithm for contour-based region filling” Turku Center for Computer science and Department of Information Technology, University of Turku, Lemminkdisenkatu 14 A, FIN-20520 Turku, Finland. Volume 29, Issue 3, Pages 441-450, June 2005.

[2] Yao-Hong, Tsai,Kuo-Liang Chung, “Region filling algorithm on bincode-based contour and its implementation” Department of Information Management and Institute of Information Engineering, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei 10672, Taiwan, ROC, Volume 24, Number 1, pp. 529-537, 6 September 2000

[3] W.G.M Geraets*, A.N. van Daatselaar, J.G.C. Verheij, “An efficient algorithm for counting region” Academic Center for Dentisty Amsterdam Radiology Department, Louwesweg 1, 1066 EA Amsterdam, The Netherlands. Computer Methods and Programs in Biomedicine, Volume 76, Issue 1, Pages 1-11, Number 1, October 2004.

[4] Mingwu Ren*, Wankou Yang, Jingyu Yang , "A new and fast contour-filling algorithm" , Department of Computer science, Nanjing University of Science and Technology,Nanjing 210094, China Received 14 September 2004; accepted 27 April 2005. Volume 38, Issue 12, December 2005, Pages 2564-2577.

[5] David Lowe, The computer vision industry, Computer Science Department 2366 Main Mall University of British Columbia Vancouver, B.C., V6T 1Z4, Canada, October 3, 1996.

[6] Andrew Mehnert * , Paul Jackway, **"**An improved seeded region growing algorithm**"** Cooperation Research Center for sensor signal and Information processing, Department of Electrical and Computer Engineering, The University of Queensland Brisbane, Qld 4072,Australia Received 20 January 1997; revised August 1997, Volume 18, Issue 10, Pages 1065-1071, 12 May, 1998.