An analysis of the Seeded Region Growing algorithm

Thavi Tennakoon

ABSTRACT

Traditionally when scanning documents in a computer, a high quality copy isn't always generated. Medical scans don't always produce a clear image for the doctor to use. Image binarization is method in which we can segment any unwanted noise from an image so that any human can clearly understand what's going on. There are many types of algorithms used to binarize. This report will focus on one image binarization technique; seeded region growing. This algorithm permits users to pick an initial seed for the algorithm to start from and will branch away by comparing neighbouring pixels to the original. After the process is done, a binarized image will be produced. Changing the tolerance level on the parameters allows you to adjust the intensity threshold of the algorithm. This report will also compare multiple images before and after the algorithm has been used on it to illustrate how it works.

Keywords: seeded region growing, image segmentation, neighbouring pixels, algorithm.

Introduction

When you scan in a real-life document into a computer, the file tends to include some unwanted visual noise which makes it difficult for the computer to read. Image binarization is a technique of image segmentation that can help clear up the image. This is done by separating the image into two classes, text and non-text.

Three different types of image binarization are seeded region growing, automatic thresholding, and topographic analysis using a water flow model. Seeded Region Growing (SRG) lets the user pick a starting point for the seed and the algorithm will check neighbouring pixels for similar and determines the value of it. Automatic thresholding looks at the image and determines a thresholding level that all the pixels will look at, then binarizes the image accordingly. Topographic analysis simulates water droplets falling on the image in a 3d plane to determine the thresholding point.

The aim of the report is to analyse one image binarization algorithm which has been tested on multiple images with unwanted visual noise.

Algorithm

The algorithm that this report will be focusing on will be the seeded region growing algorithm. The idea of seeded region growing came about when L. Bischoff said "Humans will always need to place the seeds."[1] This means that humans should be able to select the initial pixel value for the algorithm to branch from.

Segmentation is a big advantage of seeded region growing as it lets you distinguish between multiple objects in a single image. This algorithm prevents over and under segmentation in the image[2] and as it is "robust, rapid and free of tuning parameters" it provides a more effective use for document image binarization[3].

This algorithm can also be used with cellular images for feature extraction and classification of cells. Usually a biological image of cells is very unclear to the human eye so seeded region growing can help distinguish each cell through segmentation.[4]

A drawback of this algorithm is that is requires a manually selected seed. This means that depending on the seed you can get a completely different solution because the pixels will lead to different regions each time.[5]

How the algorithm works

- First, if there are no values selected for the initial seed, a figure window will open up for the user to manually select a seed.
- Create a seed so we can add pixels to a black image.
- Evaluate the image intensity at the seed points as well as calculating the mean intensity.

- Grow 1 pixel seed, and remove previous seed (so only a new pixel perimeter is created).
- Evaluate image intensity over the new perimeter.
- If the image intensity over the new perimeter is greater than the mean intensity of the previous perimeter (minus tolerance), then this perimeter is part of the segmented object.
- Repeat while there's new pixel in seed, stop if no new pixel were added.

Results

A comparison of four images before and after applying the seeded region growing algorithm are shown in the figures below. These images will also be compared with their respective 'ground truths'. The tolerance level is at 0.2 (For figure 3 it is 0.4). The seed is manually selected by clicking the middle of the image.

The Peak Signal to Noise Ratio (PSNR) will be used to objectively compare each image with its ground truths. Using the following formulas, you can calculate the Mean Squared Error (MSE) and PSNR:

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[f(i,j)-g(i,j)]^2}{M*N}$$

$$PSNR = 10log_{10}(\frac{R^2}{MSE})$$

(a) HW1 Source Image

(b) HW1 Image: Seeded Region Growing 0.2

(c) HW1 Ground Truth Image

Figure 1. HW1 Image

(a) HW4 Source Image

(b) HW4 Image: Seeded Region Growing 0.2

(c) HW4 Ground Truth Image

Figure 2. HW4 Image

(a) PR4 Source Image

(b) PR4 Image: Seeded Region Growing 0.4

(c) PR4 Ground Truth Image

Figure 3. PR4 Image

(a) PR7 Source Image

(b) PR7 Image: Seeded Region Growing 0.2

(c) PR7 Ground Truth Image

Figure 4. PR7 Image

Figure 1 shows that most of the image has been binarized. Pixel values close to black has remained unchanged. Also, you can see that pixel values with a black barrier, like the loops in some letters, are not affected. The right side of the image has been shifted to black because of the sudden pixel intensity change compared to the neighbouring pixels. The PSNR of figure 1 is 1.7516. This is a significantly low number. This is due to the black smudges on the image.

For figure 2, a lot of the coffee stains have been removed. However, the darker stains managed to bypass the algorithm. This will be because those stains show a sudden intensity change that the algorithm thinks it's wanted text. The PSNR of figure 2 is 0.4846. This is also a very low number because of the smudges on the image.

For figure 3, all the background has been removed. However, parts of the background within the letter loops remain. This is because of the limitation of this algorithm that doesn't allow the pixel neighbouring system to bypass the black text. The PSNR of figure 3 is 56.5654. This is very good compared to the previous figures. A lot of the noise has been removed which makes it very similar to the ground truth.

For figure 4, the algorithm has managed to fully binarize the image. It has lost a little detail on the text, but the noisy background has been removed. The PSNR of figure 3 is 63.3387. This is the best output compared to all the other images. It is very close to the ground truth

Looking at the PSNR values, we can objectively say that figure 3 and figure 4 show the best solutions for binarization using seeded region growing.

All these images are like their respective ground truths. The similarity with all the solutions is that sudden changes in pixel intensity makes the algorithm think that the value is for a letter. Also, when some noise within a darker barrier, the seeded region growing algorithm won't be able to reach it.

(a) Stem Cells Source Image

(b) Stem Cells Image: Seeded Region Growing 0.1

Figure 5. Stem Cells Image

Figure 5 shows two images of a bunch of stem cells in an organism before and after the seeded region growing algorithm has been applied. The tolerance level was set to 0.1 because of the low intensity levels. You can see how well the algorithm did to separate the cells with from the original image. Referring back to the 'Algorithm' section, this is why seeded region growing is used in the medical scene when studying cells.

Conclusion

In conclusion, seeded region growing is a great algorithm for segmenting objects in a single image. It's also good for general image binarization from important legal documents to cellular images for medical use.

The main limitation to this algorithm is that when there's a sudden intensity change, the algorithm passes it as a value to keep. Meaning the background has not been completely filtered out. The other limitation is the algorithm cannot reach the pixels that are surrounded by darker pixels in its perimeter.

The results prove that this algorithm is a viable solution for stem cell image research because it can clearly segment each cell from the dark background.

References

1. Rolf Adams and Leanne Bischof "Seeded Region Growing" IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 16, NO. 6, JUNE 1994
2. Shweta Kansal, Pradeep Jain “AUTOMATIC SEED SELECTION ALGORITHM FOR IMAGESEGMENTATION USING REGION GROWING” International Journal of Advances in Engineering & Technology, June, 2015, ISSN: 22311963
3. Savneet Dhaliwal, Abhilasha Jain “A Survey on Seeded Region Growing based Segmentation Algorithms” International Journal of Computer Science and Management Research. Vol 2 Issue 6 June 2013 ISSN 2278-733X.
4. Mohammed. M. Abdelsamea “An Automatic Seeded Region Growing for 2D Biomedical Image Segmentation” Mathematics Department, Assiut University, Egypt
5. Prof. R.K.Krishna2, Shilpa Dantulwar (Kamdi)1 "PERFORMANCE ANALYSIS USING SINGLE SEEDED REGION GROWING ALGORITHM" International Journal of Innovative Research in Advanced Engineering (IJIRAE). Volume 1 Issue 6 (July 2014) ISSN: 2349-2163 2/2

Appendix A

seeded_region_growing.m

```
function selected_pixels = seeded_region_growing(tolerance,img,x,y)

% If there's no point, select one from image

if(x == 0 || y == 0)
    imshow(img,[0 255]);
    [x,y] = ginput(1);
end

%Create seed with by adding point in black image

selected_pixels = false(size(img,1),size(img,2));
old_selected_pixels = selected_pixels;
selected_pixels(uint8(x),uint8(y)) = 1;
while(sum(selected_pixels(:)) ~= sum(old_selected_pixels(:)))
    old_selected_pixels = selected_pixels;
% Evaluate image intensity at seed/line points
    intensity_value = img(selected_pixels);
% Calculate mean intensity at seed/line points
    mean_intensity = mean(intensity_value);
% Grow seed 1 pixel, and remove previous seed (so you'll get only new pixel perimeter)
    neighbour = imdilate(selected_pixels,strel('disk',1,0)) - selected_pixels;
% Evaluate image intensity over the new perimeter
    neighbours = find(neighbour);
    neighbour_values = img(neighbours);
% If image intensity over new perimeter is greater than the mean intensity of previous perimeter (minus tolerance), than this perimeter is part of the segmented object
    selected_pixels(neighbours(neighbour_values > mean_intensity - tolerance & neighbour_values < mean_intensity + tolerance)) = 1;
% Repeat while there's new pixel in seed, stop if no new pixel were added
end
```

generate_results.m

```
%It will take several seconds for the output images to pop up because the
%algorithm is being ran on four images.

%Read in the image and convert to double values
HW1 = im2double(imread('HW1.png'));
HW4 = im2double(imread('HW4.png'));
PR4 = im2double(imread('PR4.png'));
PR7 = im2double(imread('PR7.png'));
Stem_Cells = im2double(imread('Stem_Cells.png'));

%If you want to select the seed manually on the image, then set the x and y
    coordinates to 0
HW1_SRG = seeded_region_growing(0.2,HW1,0,0);
HW4_SRG = seeded_region_growing(0.2,HW4,0,0);
PR4_SRG = seeded_region_growing(0.4,PR4,0,0);
PR7_SRG = seeded_region_growing(0.2,PR7,0,0);
Stem_Cells_SRG = seeded_region_growing(0.1,Stem_Cells,0,0);

%Show the binarised image
figure, imshow(HW1_SRG);
figure, imshow(HW4_SRG);
figure, imshow(PR4_SRG);
figure, imshow(PR7_SRG);
figure, imshow(Stem_Cells_SRG);

colormap gray(255);
```

calculatePSNR.m

```
function psnr = calculatePSNR(img1,img2)
%CALCULATEPSNR Summary of this function goes here
%   Detailed explanation goes here
    mse = sum(sum((double(img1)-double(img2)).^2))/(size(img1,1)*size(img2,2));
    psnr = 10*log10(255^2/mse);

end
```