



ANALYSIS OF AN IMAGE USING IMAGE SEGMENTATION METHODS IN IMAGE PROCESSING

Poornachander V¹, M.Nagaraj²

^{1,2}Research Scholar, Dept.Of Computer Science,Osmania University, Hyd.

ABSTRACT: *The Process of Image segmentation is defined as the technique, that a given photograph is divided into many elements so as to more analyze each of those parts present within the photo. Many image segmentation ways are developed by the researchers so as to create pictures sleek and simple to judge. The Prime task of the researchers operating within the field is to develop a way for economical and better image segmentation. No such algorithmic program has been developed until date that might keep a glance in any respect the factors that area unit present in image segmentation may be avoided. The aim of the researchers is to seek out the issues of digital image segmentation represent nice challenges for computer vision and for rising their accuracy and performance and conjointly for the most objective, that planning new algorithms. Here we tend to mentioned and valuate many various approaches to perform image segmentation as well as Thresholding methodology like Otsu's methodology, Colour primarily based segmentation known as K-Means clump, remodel methodology like watershed segmentation and Texture strategies like Texture Filters*

Key Words: *ImageProcessing, K-mean clustering, Texture Filters, Colour Segmentation, Thresholding.*

1. INTRODUCTION

The division of an image into significant structure is termed image segmentation, is commonly necessary to step in image analysis, object illustration, visual image, and plenty of alternative image process tasks. Here, we tend to focussed on the way to Associate in analyze and represent an object, however we tend to assumed the cluster of pixels that known that object was known beforehand and that we can target methods that notice the actual pixels that compose an object.

A great style of segmentation strategies has been projected within the past decades, and a few categorization is critical to gift the strategies properly here. A disjunct categorization doesn't appear to be attainable although, as a result of even terribly totally different segmentation approaches might share properties that defy singular categorization. The categorization bestowed during this, rather a categorization relating to the stress of an approach than a strict division.

The following categories are used:

- **Threshold based segmentation.** Histogram thresholding and slicing techniques are used to phase the image. they will be applied on to a picture, however can also be combined with pre- and post-processing techniques.
- **Edge based mostly segmentation.** With this method, detected edges in an image area unit assumed to represent object boundaries, and wont to identify these objects.
- **Region based segmentation.** wherever an edge based technique may plan to noticethe object boundaries so find the item itself by filling them in,

a vicinity based mostly technique takes the other approach, by (e.g.) beginning in the middle of an object so "growing" outward until it meets the item boundaries.

- **clustering techniques.** Although clustering is typically used as a synonym for(agglomerative) segmentation techniques, we tend to use it here to denote techniques that area unit primarily employed in exploratory information analysis of high-dimensional measure patterns. during this context, clump strategies plan to cluster along patterns that area unit similar in some sense. This goal is extremely just like what we tend to try to try and do once we section a picture, and so some clump techniques will promptly be applied for image segmentation.

- **Matching.** When we tend to apprehend what Associate in Nursing object we want to spot in a picture (approximately) sounds like, we will use this information to find the item in a picture. This approach to segmentation is termed matching.

Perfect image segmentation –i.e., every component is assigned to the right object segment– is a goal that can't sometimes be achieved. Indeed, as a result of the approach a digital image is nonheritable, this may be not possible, since a component might straddle the "real" boundary of objects such it partly belongs to 2 (or even more) objects. Most strategies bestowed here –indeed most current sectionation methods– solely plan to assign apixel to one segment, that is Associate in Nursing approach that's quite adequate for many applications.

Methods that assign a section chance distribution to every component area unit known as

probabilistic. This category of strategies is in theory a lot of correct, and applications wherever a probabilistic approach is that the solely approach correct enough for specific object measurements will simply be named. However, probabilistic techniques add extensive

complexity to segmentation –both within the sense of idea and implementation–and in and of itself area unit still very little used. Perfect image segmentation is additionally typically not reached as a result of the incidence of over segmentation or under segmentation. Within the 1st case, pixels happiness to a similar object area unit classified as happiness to totally different segments. one object is also diagrammatic by 2 or a lot of segments. within the latter case, the other happens: pixels happiness to totally different objects area unit classified as happiness to a similar object. one section might contain many objects.

2. Threshold Based Segmentation:

This is the foremost frequently used technique to section an image and it's also known as like “Otsu’s Method”. The thresholding operation is grey value remapping operation g defined by:

$$g(v) = \begin{cases} 0 & \text{if } v < t \\ 1 & \text{if } v \geq t, \end{cases}$$

Where v represents a grey value, and t is that the threshold value. Thresholding maps a grey-valued image to binary image. when the thresholding operation, the image has been segmented into 2 segments, known by the pixel values zero and one respectively. If we've got an image that contains bright objects on a dark background, thresholding may be used to section the image. Since in several types of pictures the gray prices of objects area unit terribly totally different from the background value, thresholding is commonly a well-suited methodology to section a picture into objects and background. If the objects don't seem to be overlapping, then we will produce a separate section from every object by running a labelling rule on the thresholded binary image, therefore assignment a novel constituent price to every object.

Many strategies exist to pick out an appropriate threshold price for a segmentation task. maybe the foremost common methodology is to line the edge price interactively; the user manipulating the worth and reviewing the thresholding result till a satisfying segmentation has been obtained. The bar chart is commonly a valuable tool in establishing an appropriate threshold price.

Here we've got a procedural approach to correcting heterogeneous Illumination in an image to create it easy to spot individual grains of rice within the image. we will learn the characteristics of the grains and easily reason statistics for all the grains within the image.

Step 1: Read Image

```
I=imread('rice.png');
Imshow(I).
```

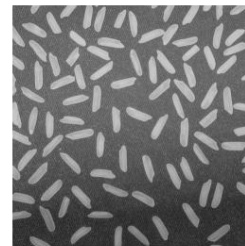


Fig 2.1. Original Image

Step 2: Use Morphological Opening to Estimate the Background

```
background = imopen(I,strel('disk',15));
surf(double(background(1:8:end,
1:8:end))),zlim([0 255]);
ax=gca;
ax.YDir= 'reverse';
```

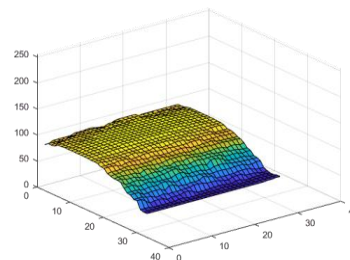


Fig 2.2. Background Image

Step 3: Subtract the Background Image from the Original image

```
I2 = I - background;
imshow(I2)
```

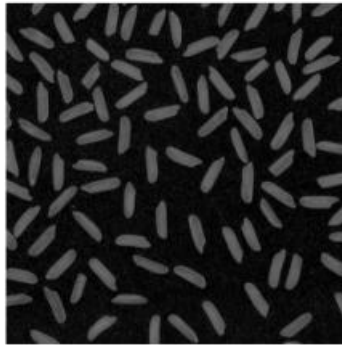


Fig 2.3 Subtracted image

Step 4: Increase the Image Contrast

```
I3 = imadjust(I2);
imshow(I3);
```



Step 5: Thresholdg the Image

Create a new binary image by thresholding the adjusted image. Remove background noise with *bwareopen*

```
bw = imbinarize(I3);
bw = bwareaopen(bw, 50);
imshow(bw)
```

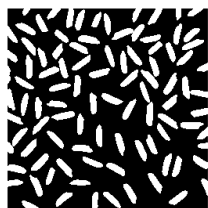


Fig 2.4 Binary Image

Step 6: Identify Objects in the Image

The function `bwconncomp` finds all the connected components (objects) in the binary image. The accuracy of the results depends on the size of the objects, the connectivity parameter (4,8, or arbitrary), and whether or not any objects are touching (in which case they may be labeled as one object). Some of the rice grains in `bw` are touching.

```
cc = bwconncomp(bw, 4)
cc = struct with fields:
Connectivity: 4
ImageSize: [256 256]
NumObjects: 95
PixelIdxList: {1x95 cell}
```

Step 7: Examine One Object

Each distinct object is labeled with the same integer value. Show the grain that is the 50th connected component.

```
grain = false(size(bw));
grain(cc.PixelIdxList{50}) = true;
imshow(grain);
```



Fig 2.5. Examine individual object

Step 8: View All Objects

One way to visualize connected components is to create a label matrix and then display it as a pseudo-colour indexed image.

Use `labelmatrix` to create a label matrix from the output of `bwconncomp`. Note that `labelmatrix` stores the label matrix in the smallest numeric class necessary for the number of objects.

```
labeled = labelmatrix(cc);
whos labeled
```

In the pseudo-colour image, the label identifying each object in the label matrix maps to a different colour in the associated colourmap. Use `label2rgb` to choose the colourmap, the background colour, and how objects in the label matrix map to colours in the colourmap.

```
RGB_label = label2rgb(labeled, @spring, 'c',
'shuffle');
imshow(RGB_label)
```

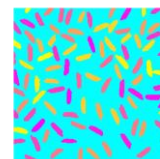


Fig 2.6. viewing objects

Step 9: Compute Area of Each Object



Each rice grain is one connected component in the cc structure. Use regionprops on cc to get the area.

```
graindata = regionprops(cc,'basic')
```

```
graindata = 95x1 struct array with fields:
```

```
Area
Centroid
BoundingBox
```

To find the area of the 50th component, use dot notation to access the Area field in the 50th element of graindata structure array.

```
graindata(50).Area
```

```
ans = 194
```

Step 10: Compute Area-based Statistics

Create a new vector grain_areas, which holds the area measurement for each grain.

```
grain_areas = [graindata.Area];
```

Find the grain with the smallest area.

```
[min_area, idx] = min(grain_areas)
```

```
min_area = 16
```

```
idx = 16
```

```
grain = false(size(bw));
grain(cc.PixelIdxList{idx}) = true;
imshow(grain);
```

Step 11: Create Histogram of the Area

```
figure
histogram(grain_areas)
title('Histogram of Rice Grain Area');
```

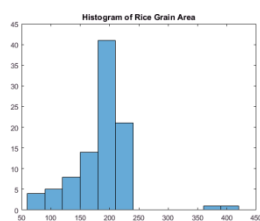


Fig 2.7 Histogram of Rice Grain Area

3. Colour Based Segmentation Using K-Means Clustering

It's a segmentation of an image supported colours in it. It consistent with this system, RGB image is first of all regenerate into hsv image. HSV colour home is used as a results of it 5 change step with human perception. Then the image is regenerate to 3 red planes then the pixels happiness to same colour area unit. Then total numbers of colours area unit displayed

and finally the entire no. of colours areaunit calculated.

The L*a*b colour house by victimization the following procedure.

Step 1: Acquire Image

Read in the fabric.png image, which is an image of colourful fabric.

```
fabric = imread('fabric.png');
figure(1), imshow(fabric), title('fabric');
```



Fig 3.1. Original fabric image

Step 2: Calculate the sample colours in L* a*b colour space for Each Region:

Here we can see six major colours in the image, the background colour red, green, purple, yellow and magenta. The L*a*b colour space enables we to quantify these visual differences. The L*a*b* colour space is derived from the CIE XYZ tristimulus values. The L*a*b* space consists of a luminosity 'L*' or brightness layer, chromaticity layer 'a*' indicating where colour falls along the red-green axis, and chromaticity layer 'b*' indicating where the colour falls along the blue-yellow axis.

Our approach is to choose a small sample region for each colour and to calculate each sample region's average colour in 'a*b*' space. We will use these colour markers to classify each pixel.

```
load regioncoordinates;
```

```
nColours = 6;
sample_regions = false([size(fabric,1) size(fabric,2)
nColours]);
```

```
for count = 1:nColours
sample_regions(:, :, count) =
roipoly(fabric, region_coordinates(:, 1, count), ...
region_coordinates(:, 2, count));
end
imshow(sample_regions(:, :, 2)), title('sample region
for red');
```

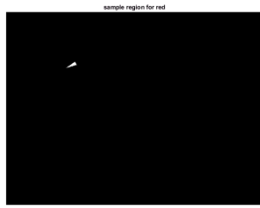


Fig 3.2. Sample region for RED

Convert our fabric RGB image into an L*a*b*

```
lab_fabric = rgb2lab(fabric);
Calculate the mean 'a*' and 'b*' value for each area
that we extracted with roiopoly. These values serve
as wer colour markers in 'a*b*' space.
```

```
a = lab_fabric(:,:,2);
b = lab_fabric(:,:,3);
colour_markers = zeros([nColours, 2]);
```

```
for count = 1:nColours
    colour_markers(count,1) =
    mean2(a(sample_regions(:,:,count)));
    colour_markers(count,2) =
    mean2(b(sample_regions(:,:,count)));
end
```

Step 3: Classify each pixel Using the Nearest Neighbour Rule:

Each colour maker has an 'a' and a 'b' value. We can classify each pixel in the lab_fabric image by calculating the Euclidean distance between that pixel and each colour marker. The smallest distance will tell that the pixel most closely matches that colour marker. For example, if the distance between a pixel and the red colour marker is the smallest, then the pixel would be labeled as a red pixel.

Create an array that contains colour labels, i.e., 0 = background, 1 = red, 2 = green, 3 = purple, 4 = magenta, and 5 = yellow.

```
colour_labels = 0: nColours-1;
Initialize matrices to be used in the nearest
neighbor classification.
```

```
a = double(a);
b = double(b);
distance = zeros([size(a), nColours]);
```

Perform classification

```
for count = 1:nColours
    distance(:,:,count) = ((a -
    colour_markers(count,1)).^2 + ...
    (b - colour_markers(count,2)).^2
    ).^0.5;
end
```

```
[~, label] = min(distance,[],3);
label = colour_labels(label);
```

```
clear distance;
```

Step 4: Display Results of Nearest Neighbor Classification

The label matrix contains a colour label for each pixel in the fabric image. Use the label matrix to separate objects in the original fabric image by colour.

```
rgb_label = repmat(label,[1 1 3]);
segmented_images = zeros([size(fabric),
nColours],'uint8');
```

```
for count = 1:nColours
    colour = fabric;
    colour(rgb_label ~= colour_labels(count)) = 0;
    segmented_images(:,:,count) = colour;
end
```

```
imshow(segmented_images(:,:,2)), title('red
objects');
```

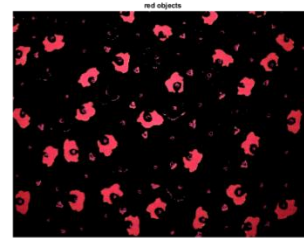


Fig 3.3. Red Object

```
imshow(segmented_images(:,:,3)), title('green
objects');
```

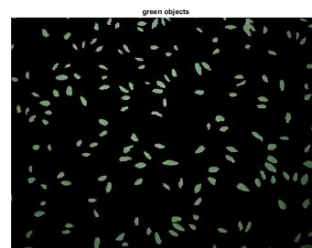
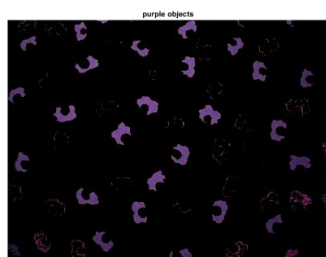


Fig 3.4. Green Object

```
imshow(segmented_images(:,:,4)), title('purple
objects');
```

Purple Object



```
imshow(segmented_images(:,:,5), title('magenta objects'));
```

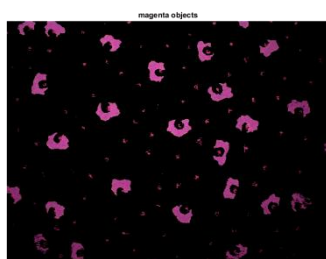


Fig 3.5. Magenta object

```
imshow(segmented_images(:,:,6), title('yellow objects'));
```

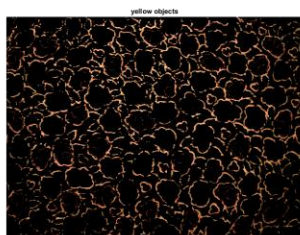


Fig 3.6. Yellow Object

Step 5: Display 'a*' and 'b*' values of the Labeled colours:

We can see the nearest neighbor classification separated the different colour populations by plotting the a* and b* values of pixels that were classified into separate colours. For display purposes, label each point with its colour label.

```
purple = [119/255 73/255 152/255];  
plot_labels = {'k', 'r', 'g', 'purple', 'm', 'y'};
```

```
figure  
for count = 1:nColours  
    plot(a(label==count-1),b(label==count-1),'MarkerEdgeColour', ...  
        plot_labels{count}, 'MarkerFaceColour',  
        plot_labels{count});  
    hold on;  
end
```

```
title('Scatterplot of the segmented pixels in "a*b*" space');  
xlabel("a*" values);
```

```
ylabel("b*" values');
```

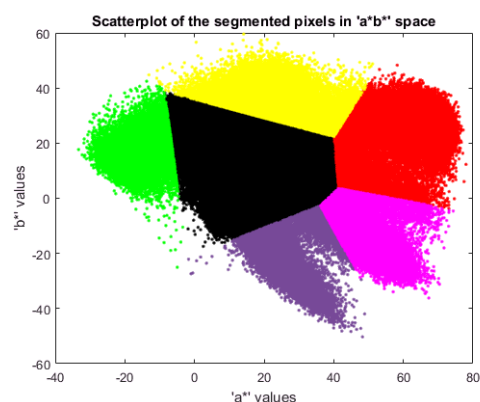


Fig 3.7. Scatterplot of the segmented pixels in 'a*b' space

4. CONCLUSION

In this paper, we discuss and evaluate many different approaches to perform image segmentation including Thresholding method such as Otsu's method, Colour based segmentation called K-Means clustering, Transform method such as watershed segmentation and Texture methods such as Texture Filters. And there is an effective approach to performing image segmentation includes using algorithms, tools and a comprehensive environment for data analysis, visualization and algorithm development. We need to develop algorithms for effective image segmentation is still a big research that will take place in the area of image processing.

5. REFERENCES

- [1] <https://in.mathworks.com/discovery/image-segmentation.html>
- [2] Gudrun J. Klinker, Steven A. Shafer and Takeo Kanade, "A Physical Approach to Colour Image Understanding," International Journal of Computer Vision, Vol. 4, No. 1, pp. 7- 38, 1990
- [3] Anil Z Chitade "Colour based image segmentation using kmeans clustering" International Journal of Engineering science and Technology, Vol. 2(10), 2010, 5319-5325
- [4] R. C. Gonzalez and R. E. Woods, Digital Image Processing. Addison-Wesley, 1993
- [5] YZang "A review on image segmentation techniques with remote sensing perspective" IAPRS Vienna, Austria, July 5-7, 2010, IAPRS, Vol. XXXVIII
- [6] Navneet Kaur et.al " Bacteria foraging based image segmentation "An International Journal of Engineering Sciences ISSN: 2229-6913 Issue July 2012, Vol. 6

- [7] Rajeshwar Dass et.al "Image Segmentation Techniques" IJECT Vol. 3, Issue 1, Jan. - March 2012
- [8] Y. Zhong, K. Karu et.al, "Locating Text in Complex Colour Images," Pattern Recognition, vol. 28, no. 10, pp. 1523-1535, October 1995
- [9] C.Yuan and S.Liang, "Segmentation of colour image based on partial differential equations" in Proc. Fourth International Symposium on Computational Conference on Intelligent Computing and Cognitive Informatics (ICICCI), pp. 300-303,2010.
- [10] R.Patil and K.Jondhale, "Edge Based technique to estimate number of clusters in k-means colour image segmentation", in Proc. 3rd IEEE International Conference on Computer Science and Information technology (ICCSIT), pp 117-121, 2010.
- [11] H.Hedberg, "a survey of various image segmentation techniques", Dept.of Electroscience, Box, Vol .118, 2010
- [12] V.Dey, "A Review on Image segmentation techniques with Remote Sensing Perspective", IAPRS, Vol XXXVIII, Part &A, pp 31-42
- [13] M.Jogendra Kumar, "Review on Image Segmentation Techniques" IJSRET, Volume 3, Issue 6, September 2014, pp 992-997.
- [14]. Rajiv kumar (December, 2011) "Image Segmentation using Discontinuity-Based Approach", IJMIP, volume 1, pp 72-78.
- [15] Amanpreetkaur, "A Review paper on Image Segmentation and its various techniques in Image Processing", IJSR, Volume 3 Issue 12, December 2014, pp 12-14.