A Comparative Analysis of Watershed and Color based segmentation for Fruit Grading

P.Deepa, S.N. Geethalakshmi

Abstract: ***In this paper, we presented two segmentation methods. Multi-scale edge detection with watershed segmentation and color based segmentation using K-means. Color based segmentation is based on fruit color and its difference. Mostly the damage part of the fruit will be of different color and that will be segmented by our algorithm very correctly. Second the watershed segmentation also segments the fruit based on color, shape and size of the damage. We compared the results of both segmentation results and the watershed segmentation outperforms the color based segmentation in all aspects.MATLAB image processing toolbox is used for the computation and Comparison results are shown with the segmented images.***

Keywords: ***Fruit grading, Multiscale edge detection, watershed segmentation, Region merging, Kmeans segmentation.***

I. INTRODUCTION

At present, fruits are graded manually by a conventional grading system of Fresh Fruit Bunches (FFB). The most difficult is to classify and sort the oil palm fruit bunches since it is labour intensive, slow and can be inconsistent due to fatigue. Different human graders classify the fruits differently and an expert grader may fail to record the grading criteria properly. The local past research focused more on the correlation of oil content with the images of the fruits captured using a CCD RGB or digital camera. They also programmed in traditional programming language to produce the algorithm which can be very time consuming and tedious.

Image segmentation is an important domain in digital image processing [2]. In recent years, in order to free the employees from the tedious work of grading fruits, many scientists work hard on the computer with powerful data processing capability, trying to make initial screening from a large number of fruits, so that employees can focus on those suspected fruits, and make timely diagnosis.

The clustering approaches can be categorized into two general groups: partitional and hierarchical clustering algorithms (for details, please refer to [1]). Partitional clustering algorithms such as K-means and EM clustering are widely used in many applications such as data mining [3], compression, [4] image segmentation [4], [5] and machine learning [6].Therefore, the advantage of clustering algorithms is that the classification is simple and easy to implement. Similarly, the drawbacks are of how to determine the number of clusters and decrease the numbers of iteration, [7].

Watersheds may also be defined for image segmentation. There are also many different algorithms to compute watersheds. Watershed algorithm is sensitive to weak edges. Although it overcame the shortcomings of traditional segmentation method for losing the weak edge, it could detect low contrast change in the region of symmetrical area, resulting in over segmentation problems. Therefore, in the application of watershed image segmentation algorithm, image preprocessing or post-processing are often made to limit the number of the regions that allowed existing.

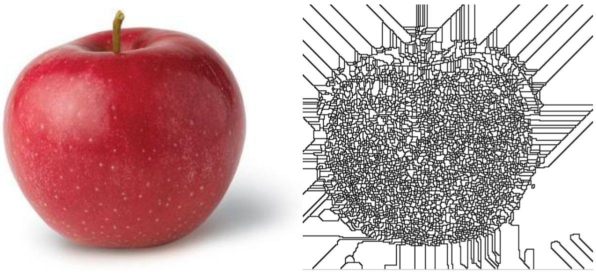


Fig 1.Original And Watershed Image

The image shown above is a normal watershed image and it is resulted in over segmentation. To reduce this we need multi scale edge detection technique. The paper is organized as follows 1. Multi-scale edge detection 2.Watershed Segmentation 3. Region merging.

II. K-MEANS IMAGE SEGMENTATION

Two K-Means algorithms have been implemented. The first clusters the pixel information from an input image based on the RGB color of each pixel, and the second clusters based on pixel intensity. The algorithm begins with the creation of initial partitions for the data. The clustering based on pixel color will be considered first. In order to try to improve the runtime and results of the algorithm, partitions which are equally spaced were chosen. Initially, eight partitions were chosen. The partitions represented red, green, blue, white, black, yellow, magenta, and cyan or the corners of the “color cube”.

With the initial partitions chosen, the algorithm can begin. Every pixel in the input image is compared against the initial partitions and the nearest partition is chosen and recorded. Then, the mean in terms of RGB color of all pixels within a given partition is determined. This mean is then used as the new value for the given partition. If a partition has no pixels associated with it, it remains unchanged.

In some implementations, a partition with no pixels associated with it would be removed; however, these partitions are simply ignored in this implementation. Once the new partition values have been determined, the algorithm returns to assigning each pixel to the nearest partition.

Manuscript received on July, 2012
P.Deepa Department of computer science, Avinashilingam University for Women, Coimbatore, India..
S.N. Geethalakshmi Department of computer science, Avinashilingam University for Women, Coimbatore, India.

The algorithm continues until pixels are no longer changing which partition they are associated with or, as is the case here, until none of the partition values changes by more than a set small amount. In order to gather more results, the initial partitions used were varied by adding and removing partitions.

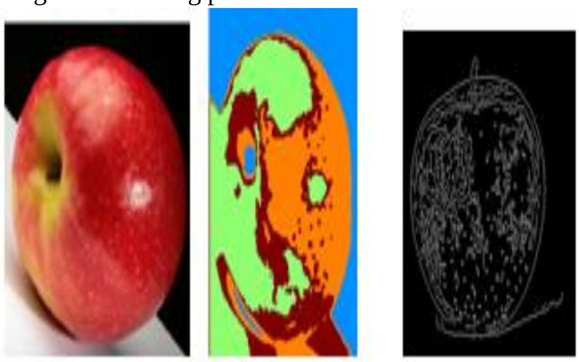


Fig 2. Original, Kmeans And Canny Edge Detected

III. MULTI-SCALE EDGE DETECTION WITH BILATERAL FILTERING IN SPIRAL ARCHITECTURE

In this research, edge detection is accomplished by applying a new bilateral filtering technique specifically designed for Spiral Architecture, integrating the multi-scale approach to edge detection. The kernel coefficients of a bilateral filter are determined by the combined closeness and similarity function.

Let f: $\Re^2 \rightarrow \Re$be the original brightness function of an image which maps the coordinates of a pixel (x, y) to a value in light intensity. Then for any given pixel a at (x, y) within a neighborhood of size n, which has *a0* as its centre, its coefficient assigned by the range filter $r(a)$ is determined by the following function:

$$r(a_i) = e^{-\frac{[f(a_i)-f(a_0)]^2}{2\sigma_r^2}} \qquad (1)$$

Similarly, its coefficient assigned by the domain filter $g(a)$is determined by the closeness function below:

$$g(x,y;t) = e - \frac{x^2+y^2}{2t} \qquad (2)$$

Where t is the scale parameter.

For the central pixel of the neighborhooda_0, its new value, denoted by $h(a_0)$,

$$h(a_0) = k^{-1}\sum_{i=0}^{n-1} f(a_i) * g(a_i) * r(a_i) \qquad (3)$$

k is the normalization constant and is defined as follows

$$k = \sum_{i-0}^{n-1} g(a_i) * r(a_i) \qquad (4)$$

The normalize k is necessary because the average image intensity should not be affected by multiplying the mask with the original image.

Application of the new bilateral smoothing filter produces, for each pixel in the image, a weighted average such that the central pixel $a_0$contribute more significantly to the result than its neighboring pixels. Pixels with more similar intensity value or closer to the central pixel contribute more than those with more different value or further away. Level of smoothness depends on the value of the scale parameter chosen and σ_r . The stronger the scale and σ_r , the higher level of smoothness will be achieved.

Following bilateral smoothing the edge detection algorithm performs edge point marking based on the method developed by Him [11]. First-order derivatives are obtained using adjacent pixel value difference along x and y directions. Definition of edge points is based on the gradient magnitude at the given location and the gradient direction.

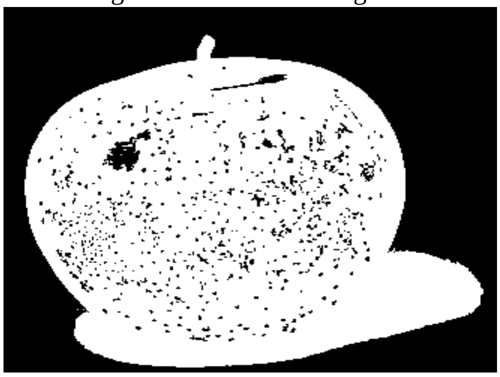


Fig 3. Muti-scale edge detected image

This research attempted to construct several edge maps of the given image, each representing intensity changes when a different and systematically controlled smoothing strength is applied. Edge map from a less smoothed image will have edge locations closer to the centre of the original edge map before any smoothing. Edge map from a stronger smoothing strength will have fewer noisy or false edge points. Therefore to ensure precise localization of edge points and detection of real edge points, these edge maps should be systematically compared. Major steps of edge detection algorithm developed by this research with bilateral smoothing for Spiral Architecture are presented below:

1. **Initial edge detection**. Edge map of the original input image gives the most precise locations of edge points because it is not affected by smoothing. Set this edge to E;
2. **Edge detection for newly smoothed image**. Edge points are marked according to the method defined in [8]. Denote this edge by E_s
3. **Edge Map Update**. Edge map$E$and$E_s$arecompared. Edge map E is depended on for its more precise edge locations and edge map of the smoothed image E_s is used to eliminate some false edge points for the smoothed image has less noise than the original one. The new edge map after comparing will contain those edge points from the initial edge map that have at least one immediate neighboring pixel from the edge map of the smoothed image.
4. **Repeat** Step 2, 3 and 4 with systematically

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

increased smoothing strength until the new current edge map is not significantly different from the one obtained at the end of last iteration.

IV. WATERSHED SEGMENTATION

Watershed segmentation is a mathematical morphology segmentation method based on topography, the basic idea is to consider the image as a topographical surface, gray levels of pixels in the image correspond to altitude values, local minima and its influence zones are defined as catchment's basins, and the watershed will be defined by the lines that separate adjacent catchment's basins.

By analogy, we can figure that we have pierced holes in each regional minimum of the image which are being regarded as a topographic surface. We then slowly immerse our surface into a lake. Starting from the minima of lowest altitude, the water will progressively fill up the different catchment basins. At each pixel where two or more catchment's basins meet, an imaginary 'dam' is built. At the end of a recursive process, each minimum is surrounded by dams, which delimit the associated catchment basins. These dams correspond to the watersheds [5] of the topographical

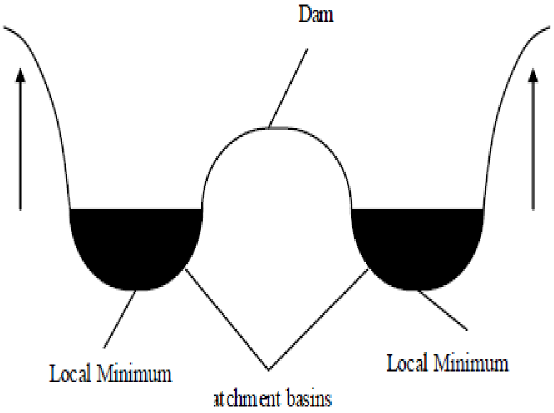


Fig 4. Watershed segmentation

L. Vincent and Pierre Soille presented a precise and efficient implementation of watershed transform by immersion simulation [6]. In this implementation there are two steps: an initial sorting and a flooding step.

The first step consists in an initial sorting of the pixels in the increasing order of their gray values. It's very efficient since it exploits the particular data structure. It runs in linear time with respect to the number of pixels to be sorted. In the second step, a fast computation of geodesic influence zones is enabled by a breadth-first scanning of each threshold level. This particular scanning is implemented via the use of a queue of pixels, i.e., a first-in-first-out data structure.

V. REGION MERGING

The watershed computation of the image mostly results in an over-segmentation, i.e., the correct contours are lost in a mass of irrelevant ones; this can be solved by performing region merging on segmented image. We mainly carried on merging to the small regions and the adjacent regions, two merging methods were given below.

1) Small regions merging

There are a number of small regions in the image which is usually caused by noise pixels. Removal of such small regions can not only further reduce the number of regions, but also reduce the cost of merging the adjacent similar regions.

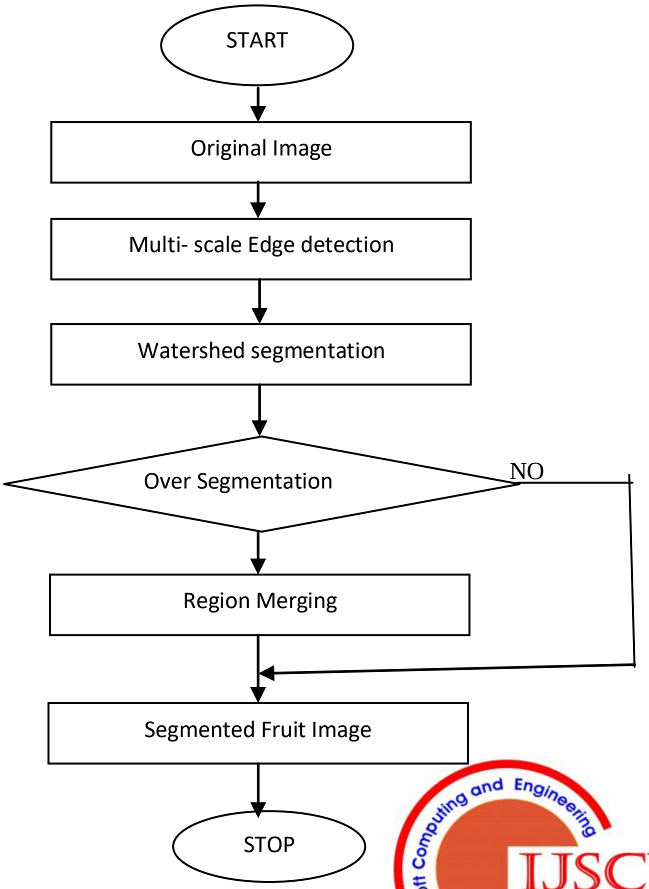
Therefore, first step of region merging is the initial merging according to the area of the region. The area of a small area can be limited by a threshold, the area of the region are defined as the number of pixels within the region, definition of the region number *Gm* is generally not more than 8. Although number of pixels in these small regions are small, but the variance of the region are usually large, if taking merge regions as a whole, the merged regions often cannot truly reflect the characteristics of the region, and may affect the surrounding characteristics. So we used such merging criteria: searching for the nearest gray-scale neighbors pixel-by-pixels. With each pixel merging, the properties of similar regional update, markers of corresponding pixels should be updated too.

2) Adjacent regions merging

In case of weak edges between adjacent regions, also consider the region merging. Because the merged area must have similar characteristics, so take the mean gray value of the region as references [7]. Merging method is: let $\mu_i i=1, 2..., m$ denotes the mean gray values of pixels in the region, the merging formula is: $|\mu_1 - \mu_2| \le Ta$.

5.1 Algorithm Flow

We introduced Multi-scale Edge Detection algorithm in the pretreatment process, and simultaneously achieved the effects of noise-removing and edge-strengthening. Then the preprocessed image was selected as input of watershed segmentation, thus we got a more accurate segmentation results. Finally we performed region merging to the images in order to remove the over-segmentation effect. We implemented algorithms in Matlab7.6; the specific process is shown in Figure.



Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

VI. EXPERIMENTAL RESULTS AND DISCUSSION

Kmeans image segmentation is applied to fruit image. It segments the image using the color. The results are shown below

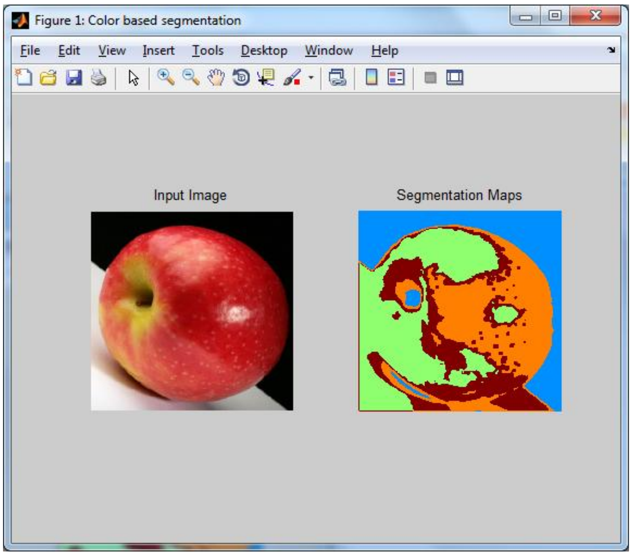


Fig 5. Kmeans color based segmentation

Our method is applied to Fruit segmentation; Measure the standard deviation with the area, the formula

$$E = \left|1 - \frac{sa}{sb}\right| * 100\% \quad (5)$$

S_a is the area of the fruit which to be assessed, S_b is manually segmented results. Calculate the effect according to the above equation above, evaluation results are shown in Figure.

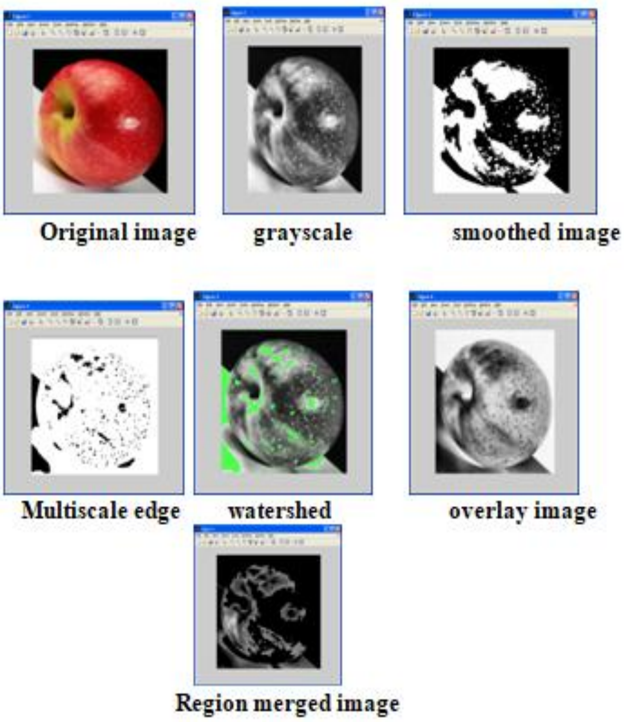


Fig 2. Segmentation results

Table 1 shows the evaluation of the effect on fruit grading. Histogram segmentation's error: 4.21%, method in this paper's error: 1.15%. Due to the fruits damage part, segmentation by histogram may form empty holes within the fruit, we can use the opening and closing operation to eliminate holes; however, in order to eliminate holes, there would be slight changes in morphology, and this is the major factor which affect segmentation effect.

Table 1: Evaluation Results

Fruit Image	Manual Grading(area-mm^2)	Histogram Grading(area- mm^2)	Watershed grading(area- mm^2)	Kmeans image seg
1	458	375	439	405
2	378	312	367	347
3	1578	1345	1598	1404
4	2548	1801	2436	1981
5	3659	2945	3510	3268
6	3854	3054	3739	3478
Mean error	-	10.41%	3.09%	6.85%

III. CONCLUSION

In this paper, the fruit image segmentation method based on color segmentation results more color space and accurately detects the damage part of the image. It outperforms the results of histogram based segmentation. But it consider only the color of the image, it didn't consider the damage size.

Method on watershed algorithm combines the advantages of multi-scale edge detection and strengthens the image edge while filtering; then make watershed segmentation. The method overcomes the shortcomings of traditional segmentation method for missing the weak edge; finally, the method makes region merging in the segmented image to limit the number of the regions that allowed existing. It can be safely concluded that this method can achieve closed and accurate results.

REFERENCES

1. YQ Zhao, JX Liu, J Liu. Medical image segmentation based on morphological reconstruction operation [J], Computer Engineering and Applications, 2007,43 (10) :238-240.
2. J Tian. Medical Image Processing and Analysis [M], Beijing: Electronic Industry Press, 2003, 9: 35-43.
3. X Liu, ZS You. Multi-scale morphological edge detection method [J], Optical Engineering, 2003, 3 (30): 56-58.
4. Y Wei, XH Xu, T Jia, DZ Zhao. CT images of suspected pulmonary nodules extraction based on multiscale morphology filtering[J],
5. Northeastern University Journal, 2008, 3 (29): 91-93.
6. Patino. Fuzzy relations applied to minimize over segmentation in watershed algorithms [J], Pattern Recognition Letters, 2005, 26 (6): 819-828.
7. Vincent, L., Soille,P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans. Patt. Anal. Mach. Intell. 13,6(1991), 583-598
8. HB Liu, XQ He, XD Liu. Image segmentation method based on watershed and region merging[J], Computer Applications Research, 2007, 9(24): 307-30.

9. Hongshe Dang, Jinguo Song, Qin Guo. A Fruit Size Detecting and Grading System Based on Image Processing. 2010 Second International Conference.
10. John B. Njoroge. Kazunori Ninomiya. NaoshiKondo and IlidekiToita. Automated Fruit Grading System using Image Processing. *2002* Auk 5-7.2MI2, *osa*s.*