Image Segmentation based Quality Analysis of Agricultural Products using Emboss Filter and Hough Transform in Spatial Domain

Mamta Juneja[1], Parvinder Singh Sandhu[2]
1 & 2: RBIEBT, Kharar under Punjab Technical University, Punjab India

er_mamta@yahoo.com

Abstract: Very few technologies developed less than 30 years ago have permeated our lives as much as the image processing. Development in the field of image processing especially in its field image segmentation which is used to extract regions of interest has proven wonders in various applications like Signature verification, Face recognition, Thumb impression verification, Automatic character recognition, Industrial machine vision for assembly and inspection etc. But the potential of image segmentation in the field of agriculture is yet to be exploited for the daily use. In this paper efforts are focused on the role of image segmentation in the field of agriculture to analyze the fruit quality on the basis of its color, size and weight. In the proposed algorithm, the desired conclusions/analysis can be made by comparing various inputs received (color, size, weight) with predefined parameters, which can be further used for grading, packaging and chopping of fruits in agriculture. [Researcher. 2009;1(5):62-68]. (ISSN: 1553-9865).

Keywords: Spatial domain, Frequency domain, Linear spatial filtering, Color models, Histogram, segmentation, Edge detection, Prewitt and Emboss filter, Hough transform

1. Introduction

Digital Images are electronic snapshots taken of a scene or scanned from documents, such as photographs, manuscripts, printed texts, and artwork. The digital image is sampled and mapped as a grid of dots or picture elements (pixels). Each pixel is assigned a tonal value (black, white, shades of gray or color), which is represented in binary code (zeros and ones). The binary digits ("bits") for each pixel are stored in a sequence by a computer and often reduced to a mathematical representation [1]. An image may be considered to contain sub-images sometimes referred to as regions-of-interest, ROIs, or simply regions. [2]. In searching the region of interest, the Edge detection is one of the most commonly used technique in image analysis, and there are probably more algorithms in the literature for enhancing and detecting edges than any other single subject [3] and [4].To extract features from digital images, it is useful to be able to find simple shapes - straight lines, circles, ellipses and the like - in images. In order to achieve this goal, one must be able to detect a group of pixels that are on a straight line or a smooth curve [5] [6] and [7].

1.2. Color Models

Each color model is oriented towards either specific hardware (RGB, CMY, YIQ), or image processing applications (HSI).

1.1.1 RGB Color Model

The RGB (Red, Green, Blue) color model is an especially important one in digital image processing because it is used by most digital imaging devices (e.g., monitors and color cameras). In the RGB model, a color is expressed in terms that define the amounts of Red, Green and Blue light it contains.

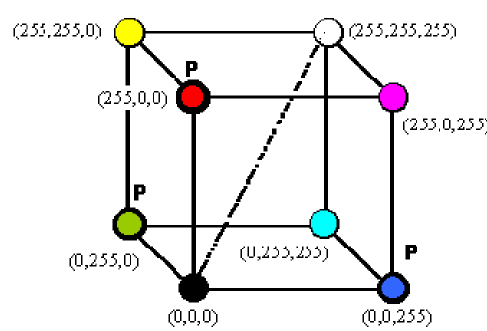


Figure 1: RGB Color Model [8]

1.1.2 HSI Color Model

The HSI (Hue, Saturation, Intensity) color model describes a color in terms of how it is perceived by the human eye. This is useful when processing images to compare two colors, or for changing a color from one to another. The HSI model is also a more useful model for evaluating or measuring an object's color characteristics, such as the "redness" of a berry.

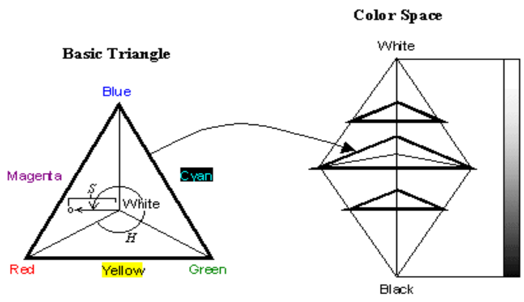


Figure 2: HSI Color Model

1.1.3 RGB to HSI Conversion

First, we convert RGB color image to HSI space beginning with normalizing RGB values:

r= R/R+G+B, g=G/R+G+B, b=B/R+G+B (2)

Each normalized H, S and I components are obtained by,

$$h=\cos^{-1}\left\{\frac{0.5\left[(r-g)+(r-b)\right]}{\left[(r-g)^2+(r-b)(g-b)\right]^{1/2}}\right\}$$

$$h\in\left[0,\Pi\right] \quad \text{for } b\le g \qquad (3)$$

$$h=2\Pi-\cos^{-1}\left\{\frac{0.5\left[(r-g)+(r-b)\right]}{\left[(r-g)^2+(r-b)(g-b)\right]^{1/2}}\right\} \quad h\in\left[\Pi,2\Pi\right] \quad \text{for}$$

$$b>g \qquad (4)$$

$$s=1-3.\min(r,g,b)\ ; \quad s\in[0,1] \qquad (5)$$

$$i=(R+G+B)/(3\cdot 255)\ ; \quad i\in[0,1] \qquad (6)$$

For convenience h, s and *i values* are converted in the ranges of [0,360], [0,100], [0,255] respectively by:

$$H=h\times 180/\Pi\,,\ S=s\times 100\,,\ I=i\times 255 \quad (7)$$

1.2. Segmentation

The techniques that are used to find the objects of interest are usually referred to as *segmentation techniques*. The most commonly used techniques of segmentation are Thresh holding, Edge Detection, Binary Mathematical Morphology and Region Growing and Splitting [9].

1.2.1 Edge Detection

Edges often occur at points where there is a large variation in the luminance values in an image, and consequently they often indicate the edges, or occluding boundaries, of the objects in a scene. **First Order Differential Methods of Edge Detection**

Most edge detection methods work on the assumption that an edge occurs where there is a discontinuity in the intensity function or a very steep intensity gradient in the image. If we take the derivative of the intensity values across the image and find points where the derivative is a maximum, we will have marked our edges (refer Figure 4).

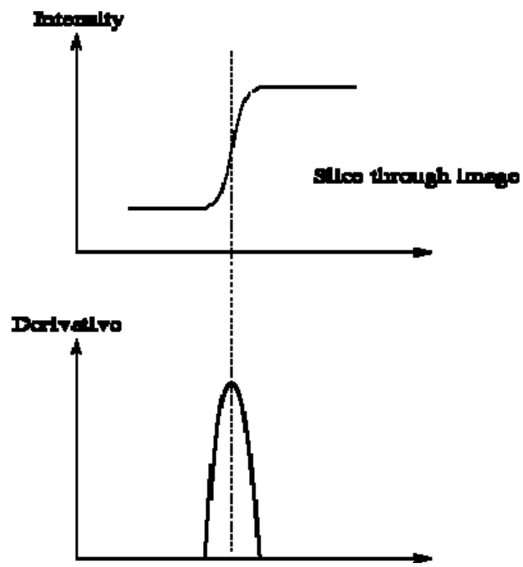


Figure 4(a): An ideal step edge and its derivative profile.

In a discrete image of pixels we can calculate the gradient by simply taking the difference of grey values between adjacent pixels.

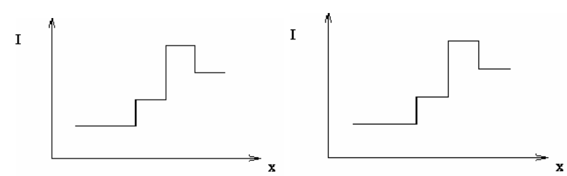


Figure 4(b): An ideal step edge and its derivative profile [10].

This is equivalent to convolving the image with the mask [-1, 1]. The gradient of the image function *I* is given by the vector

$$\nabla I=\left[\frac{\partial I}{\partial x},\frac{\partial I}{\partial y}\right] \qquad (8)$$

The magnitude of this gradient is given by $\sqrt{\left(\frac{\partial I}{\partial x}\right)^2+\left(\frac{\partial I}{\partial y}\right)^2}$ and its direction by $\tan^{-1}\left(\frac{\partial I}{\partial y}/\frac{\partial I}{\partial x}\right)$.
Note, one can use any pair of orthogonal directions to compute this gradient, although it is common to use the x and y directions.
The simplest gradient operator is the *Robert's Cross operator* and it uses the masks

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \& \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad (9)$$

Instead of finding approximate gradient components along the x and y directions it approximates gradient components along directions at 45^o and 135^o to the axes respectively. Thus the Robert's Cross operator uses the diagonal directions to calculate the gradient vector. An 3×3 approximation to $\frac{\partial I}{\partial x}$ is given by the convolution mask

-1	0	1
-1	0	1
-1	0	1

This defines $\frac{\partial I}{\partial x}$ for the *Prewitt operator*, and it detects vertical edges. The *Sobel operator* is a variation on this theme giving more emphasis to the centre cell. The $\frac{\partial I}{\partial x}$ is given by

-1	0	1
-2	0	2
-1	0	1

Similar masks are constructed to approximate $\partial I/\partial y$,thus detecting the horizontal component of any edges. Both the Prewitt and Sobel edge detection algorithms convolve with masks to detect both the horizontal and vertical edge components; the resulting outputs are simply added to give a gradient map. The magnitude of the gradient map is calculated and then input to a routine that suppresses (to zero) all but the local maxima. This is known as non-maxima suppression. The resulting map of local maxima is thresholded (small local maxima will result from noise in the signal) to produce the final edge map. It is the non-maxima suppression and thresholding that introduce non-linearities into this edge detection scheme. Moreover, the output of the thresholding stage is extremely sensitive and there are no automatic procedures for satisfactorily determining thresholds that work for all images [11].

1.2.1.1 Gradient Based Methods

An edge point can be regarded as a point in an image where a discontinuity (in gradient) occurs across some line. A discontinuity may be classified as one of these types (see figure).

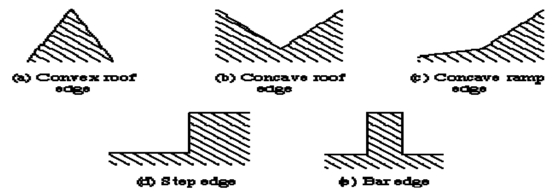


Figure 5: Points of Discontinuity in an Image [12]

Roberts

It provides simple approximation

$$G[f[i,j]]=|f[i,j]-f[i+1,j+1]|+|f[i+1,j]-f[i,j+1]| \qquad (10)$$

Which becomes this after applying convolution mask:

$$G[f[i,j]]=|G_x|+|G_y| \qquad (11)$$

where

$$G_x=\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, G_y=\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \qquad (12)$$

Sobel

A way to avoid having the gradient calculated about an interpolated point between pixels is to use a 3 x 3 neighborhood for the gradient calculations. Consider the arrangement of pixels about the pixel [i, j]. The Sobel operator is the magnitude of the gradient computed by:

$$M\sqrt{s_x^2+s_y^2} \qquad (13)$$

Where the partial derivatives are computed by

$$s_x=(a_2+ca_3+a_4)-(a_0+ca_7+a_6) \qquad (14)$$

$$s_y=(a_0+ca_1+a_2)-(a_6+ca_5+a_4) \qquad (15)$$

with the constant c = 2.

Like the other gradient operators, S_x and S_y can be implemented using convolution masks:

$$S_x=\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, S_y=\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad (16)$$

Prewitt

The Prewitt operator uses the same equations as the Sobel operator, except that the constant c = 1. Therefore:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \; S_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad (17)$$

Note that, unlike the Sobel operator, this operator does not place any emphasis on pixels that are closer to the center of the masks.

1.2.1.2 Laplacian-Based Edge Detection

A maximum of the first derivative will occur at a zero crossing of the second derivative. To get both horizontal and vertical edges we look at second derivatives in both the *x* and *y* directions. This is the *Laplacian* of *I*

$$\nabla^2 I = \frac{\partial^2 I}{\partial^2 x} + \frac{\partial^2 I}{\partial^2 y} \quad (18)$$

The Laplacian is linear and rotationally symmetric.

1.2.1.3 The Emboss Filters

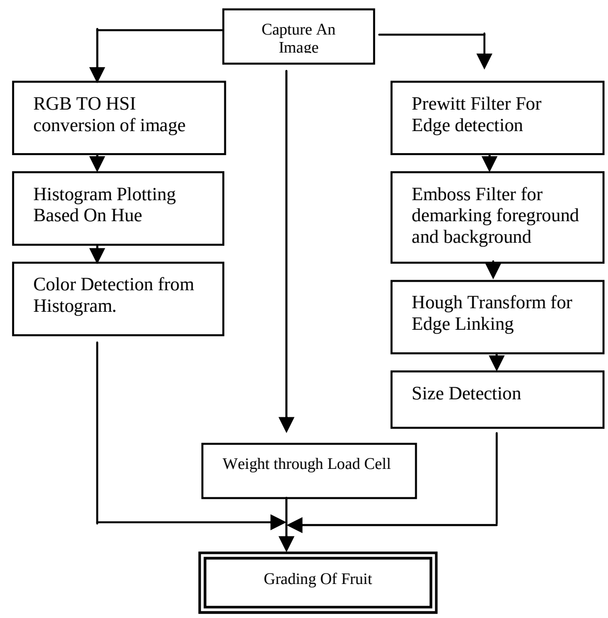
Emboss filters work somewhat like edge detection filters, with the difference that embossed images are entirely gray. Areas in the picture are detected and are given a certain "height level", which is made visible by using grayscale borders, making the image look like it is three-dimensional.The embossing effect gives the optical illusion that some objects of the picture are closer or farther away than the background, making a3D or embossed effect. This filter is strongly related to the previous one, except that it is not symmetrical, and only along a diagonal. Mask used is:

$$\begin{pmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{pmatrix} \quad (19)$$

Two dimensional convolution works just like one dimensional discrete convolutions, the output image y of an entering bitmap x, through a filter system of bitmap impulse response h (width and height M) is given by formula

$$y[r,c] = \frac{1}{\sum_{i,j} h[i,j]} \sum_{j=0}^{M-1} \sum_{i=0}^{M-1} h[j,i]\, gx[r-j, c-i] \quad (20)$$

2. Proposed Algorithm



2.1 Color Detection:

RGB to HSI Conversion: Captured RGB image is converted to HSI (Hue, Saturation & Intensity) (Refer eqn3-7), which helps to analyze the dominant color content in the image as hue provides the exact type of color dominance.

(a) Histogram Analysis: The hue value of each pixel value of the image is extracted using function HSI (r, g, b) which returns the value of Hue, Saturation and Intensity and plotted in a graph (no of pixel vs. hue) to get the graphical color dominance in an image.
The conclusion can be made from the graphs as follows:
If 0< Hue <40 red color is dominant with a small component of green color but no component of blue color.
If 40< Hue < 80 red green color is dominant with the contribution from red color too but no blue color component.
If 80 < Hue < 120 green color is dominant and a bit contribution from blue is present but no red color is absent.
If Hue > 120 Blue color is dominant with a bit contribution from green color
If the histogram is spread in all the regions this reflects the mixture of RGB is present then the density of the graph represents the color dominance.

2.2 Size Detection

This is done while applying edge detection method, which is further, followed by edge linking to get the size.

a) **Edge Detection** to define boundary is carried in following steps:

1) **Prewitt Filter** (gradient method based on first derivative operator) is used to mask with the captured bitmap of image. Result for the same is stored in convolution matrix in eqn (20).

2) Output of step 1 is then applied with **Emboss Filter** (Laplacian method based on second derivative operator) to get more refined, thin and better results immune to noise. This even depicts position of pixel (whether in foreground or background) as given in eqn 22.

b) **Edge Linking** is implemented using global method named Hough Transform.

Hough Transform for size calculation:

1. Find all of the desired points in the range
2. For each feature point
3. For each possibility i in the accumulators that passes through the feature point.
4. Increment that position in accumulator
5. Find local maxima in the accumulator
6. If desired map each maxima in the accumulator back to image space.

Above steps are implemented by changing bitmap to bitmap data, locking it and defining scan0, stride for it and rotation of 360,100,100 is given to h,s,i respectively which further checks variance in reference image and given image for R, G, B. This variance is accumulated and changed to centimeters.

Size=variance in terms of no. of pixels /13
(to change result in cms).

Conversion of variance in terms of size depends on height of camera and the luminosity on the plane of interest

2.3 Weight Approximation

Weight parameter is passed by user in category A-heavy, B-average, C-light. Any one of the switches representing A/B/C which symbolizes weight is passed as input parameter.

2.4 Grading of Fruit

Accumulative sum of results (2.1), (2.2) and (2.3) decides the quality of captured image.
Example of the results:

Weight	Color	Size	Result
A	> 80	6-8 unit	Very good quality
B/C	>80	6-8 unit	Good quality
A	50-80	6-8 unit	Average
B/C	50-80	6-8 unit	Poor
A/B/C	<50	6-8 unit	Average
A/B/C	Any	< 6 unit	Average
A/B/C	< 50	<6 unit	Poor

3. Experimental Evaluation and Discussion

The software of the prototype had been tested using the standard 1.3 Mega pixel camera (of creative make) with suitable hardware i.e. micro controller. The camera is placed at USB port and the controller is interfaced at COM port available at the machine. Initially the standard tests with standard images had been performed. This test helps to check the plotting of the RGB Color in HSI domain and helps to locate the region of interest.

The distributed histogram is obtained from the RGB image under test. The area obtained in the histogram analysis helps us to restrict ourselves in the particular region of histogram if any particular dominance of color is to be checked in the image analysis that helps to make desired selection. The objective of color segmentation for color dominance is achieved which is stated in our problem.

Edge detection plays the vital role for any image processing software, which leads to get location of the object in particular region of the interest under the camera scanning area. The simple edge detection and the laplacian edge detection technique had tried in the system by implementing suitable convolution matrix, the results are overwhelming and the detailed outputs are shown below for the reference.Hough transformation plays the important role in calculating the size of the object in the area of interest. The image scanned (known size) with the camera is put under standard algorithm of Hough transform and we are able to get the approximate size, which is placed under the camera. This even worked successfully as per desired in the starting of the discussion. The various experimentation outputs are explained in various steps as follows:

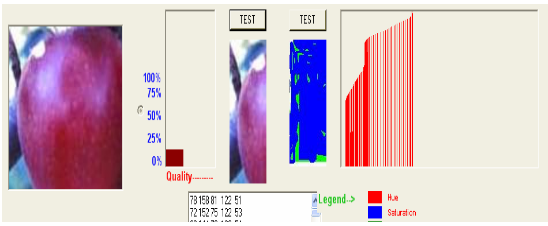


Figure 7 Histogram Analysis of an Apple.

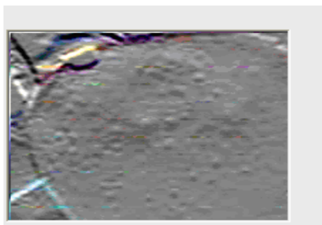


Figure 8: Edge Detection using Prewitt Filter

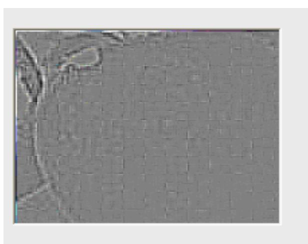


Figure 9: Edge detection using Emboss Filter

Figure 10: Size Detection using Hough Transform

6. Conclusion

The Algorithm has been designed and implemented for the image segmentation based quality analysis using Hough transform in RGB vector space. In this a new algorithm is proposed and implements the quality analysis. The captured image of the fruit is transformed from RGB to HSI vector space, which is further utilized for to get the exact color content by plotting its histogram based on hue.The boundary of the image is found using the Edge Detection techniques by operating the same captured image with Prewitt Filter (Gradient based linear special filter) and Emboss Filter (Laplacian based linear special filter). The size of the image is calculated by applying the Hough Transform (a Global Edge Linking technique) and calculating the variance in color of an image in RGB vector space with some pre defined reference. The various algorithms viz. RGB to HSI conversion, Edge detection using Prewitt Filter/Emboss Filter followed by Edge Linking using Hough transform which helps to calculate the size of the image.

The outputs of the software using the stated algorithm are satisfactory when put the apple under test, the size and color outputs are in acceptable limits. Thus, the said algorithm can play an important role in grading, packaging and chopping of fruits in agriculture.

References

1. Pratt, W.K., Digital Image Processing, John Wiley & Sons, New York 1978.
2. Grimson, W.E.L., From Images to Surfaces, MIT Press, Cambridge, MA. 1981.
3. Marr, D. and Hildreth, E., Theory of Edge Detection, Proceedings of the Royal Society of London, Series B, Vol. 207, 1980. Pp. 187-217.
4. Canny, J., A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 6, November 1986. Pp. 679-698.
5. Shen, J. and Castan, S., An Optimal Linear Operator for Step Edge Detection, Computer Vision, Graphics, and ImageProcessing:Graphical Models and Understanding, Vol. 54, No. 2, March 1992. Pp. 112-133.
6. Torre, V. and Poggio, T. A., On Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 2, March, 1986. Pp. 147-163. Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," Comm. ACM, Vol. 15, pp. 11–15 (January, 1972).
7. Digital Image Processing by Gonzalez and Woods, 2nd Edition Prentice Hall.
8. Abdou, I.E, and Pratt, W.K., Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors, Proceedings of the IEEE, Vol. 67 No. 5, Mat 1979. Pp. 753-763.

9. Deutsch, E. S., and Fram, J.R., A Quantitative Study of Orientation Bias of Some Edge Detector Schemes, IEEE Transactions on Computers, Vol. C-27, No. 3, March, 1978. Pp. 205-213.
10. Kitchen, L. and Rosenfeld, A., Edge Evaluation Using Local Edge Coherence, IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-11, No. 9, Sept. 1981. Pp. 597-605.
11. Shah, M., Sood, A., and Jain, R., Pulse and Staircase Edge Models, Computer Vision, Graphics, and Image Processing, Vol. 34, 1986. Pp.321-343.
12. Steven M. Kay and Gerald J. Lemay "Edge Detection using Linear Model" IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-34, NO. 5, October 1986 PP 1221-1227.

6/5/2009