

A THESIS FOR THE MASTER DEGREE OF COMPUTER
ENGINEERING

**LEAF DAMAGE DETECTION AND
PENCIL DRAWING BASED ON IMAGE
PROCESSING TECHNIQUES**

FEBRUARY 21, 2014

**GRADUATE SCHOOL OF
CHONBUK NATIONAL UNIVERSITY**

**DIVISION OF ELECTRONICS AND INFORMATION
ENGINEERING (COMPUTER ENGINEERING)**

**BY
HITIMANA ERIC**

**LEAF DAMAGE DETECTION AND
PENCIL DRAWING BASED ON IMAGE
PROCESSING TECHNIQUES**

**이미지 처리 기술 기반의 잎손상 탐지 및
연필 드로잉**

FEBRUARY 21, 2014

**GRADUATE SCHOOL OF
CHONBUK NATIONAL UNIVERSITY**

**DIVISION OF ELECTRONICS AND INFORMATION
ENGINEERING (COMPUTER ENGINEERING)**

**BY
HITIMANA ERIC**

**LEAF DAMAGE DETECTION AND
PENCIL DRAWING BASED ON IMAGE
PROCESSING TECHNIQUES**

ADVISOR OUBONG GWUN

SUBMITTED TO THE GRADUATE SCHOOL IN PARTIAL
FULFILLMENT OF THE REQUIREMENTS FOR THE MASTER
DEGREE OF COMPUTER ENGINEERING

NOVEMBER 6, 2013

**GRADUATE SCHOOL OF
CHONBUK NATIONAL UNIVERSITY
DIVISION OF ELECTRONICS AND INFORMATION
ENGINEERING (COMPUTER ENGINEERING)**

**BY
HITIMANA ERIC**

**TO APPROVE THE SUBMITTED THESIS FOR THE
MASTER DEGREE OF COMPUTER ENGINEERING BY
HITIMANA ERIC**

COMMITTEE CHAIR

Dr. Sooncheol Park
Chonbuk National University

COMMITTEE

Dr. Gihwan Cho
Chonbuk National University

COMMITTEE

Dr. Oubong Gwon
Chonbuk National University

DECEMBER 11, 2013

**GRADUATE SCHOOL OF
CHONBUK NATIONAL UNIVERSITY**

Contents

Contents	i
List of Figures	iv
Abstract	vi
1.Introduction	1
1.1 Background	1
1.2 Motivation and Objectives	1
1.3 Outline of the Thesis	4
2.Leaf damage detection and estimation	5
2.1 Literature review / related works	5
2.1.1 Comparative survey of leaf image processing	5
2.1.1.1 Leaf disease detection and classification	5
2.1.1.2 Background removal	6
2.1.1.3 Leaf Area Measurement	8
2.1.2 Image Processing Techniques	11
2.1.2.1 Image enhancement	11
2.1.2.2 Image Segmentation.....	14

2.2	Proposed Leaf Damage Detection and Estimation	16
2.2.1	Contrast enhancement.....	17
2.2.2	Background removal.....	20
2.2.3	Leaf damage clustering.....	22
2.2.4	Leaf damage estimation.....	23
2.3	Leaf damage detection implementation and results.....	25
2.3.1	Implementation.....	25
2.3.2	Experimental Results and Analysis	26
3.	Pencil Sketch with Hashing	31
3.1	Literature review / related works	31
3.1.1	Applications of NPR.....	31
3.1.2	Image Based Pencil Drawing	33
3.1.2.1	Image Based Pencil Drawing by Rendering Strokes	33
3.1.2.2	Image Based Pencil Drawing by Decoupling Strokes	34
3.1.2.3	Image Based Pencil Drawing by LIC	35
3.1.2.4	Image Based pencil drawing by tone and stroke combination.....	39
3.1.2.5	Image Based Color Sketch by Interactive Segmentation	41
3.2	Proposed Color Pencil Sketch with Hatchings	43

3.2.1 Hatching Line Generation	44
3.2.2 Shading Image with Boundary	46
3.2.3 Sketch Colorization	48
3.3 Color Pencil Sketch implementation and results	49
3.3.1 Implementation.....	49
3.3.2 Experimental Results and Analysis	49
4. Conclusion and Future Works	53
4.1 Conclusion	53
4.2 Future Works.....	54
Reference	56
요약	Error! Bookmark not defined.
Acknowledgement.....	66

List of Figures

[Figure 1] Various Gamma Curves	13
[Figure 2] Gamma Vs Gamma Correction	14
[Figure 3] Proposed framework of leaf damage detection and estimation	17
[Figure 4] Image enhancement using LUT gamma correction.....	18
[Figure 5] Enhancement of high contrast image	19
[Figure 6] Enhancement of low contrast image	20
[Figure 7] Image background removal	21
[Figure 8] Result for background removal	22
[Figure 9] Segmentation result of coffee rust damage infected at 26.25%.	24
[Figure 10] Detection with an estimation of 6%	27
[Figure 11] Detection with an estimation of 22%	27
[Figure 12] Detection with an estimation of 26.25%	27
[Figure 13] Graphical comparison of damage detection	29
[Figure 14] Comparative results of background method.....	29
[Figure 15] Comparative results of background method.....	30
[Figure 16] Infected leaf detection comparison.....	30
[Figure 17] Appearance of the speed lines can be varied using line styles	31
[Figure 18] Three variants of rendering a design	32
[Figure 19] Interactive rendering using 6H pencil followed by interactive rendering with strokes	34

[Figure 20] Basic features of Durand’s interactive drawing system	35
[Figure 21] Flow mechanism of LIC convolution.....	35
[Figure 22] Mechanism of the LIC proposed by Brian Cabral and Casey	36
[Figure 23] Colored pencil drawing resulting from the same reference image.....	38
[Figure 24] Framework of proposed hatched pencil sketch image generation.....	43
[Figure 25] Hatching lines generation in the direction of 45°	45
[Figure 26] Shaded image with boundary	47
[Figure 27] Pencil sketch with hatching lines	48
[Figure 28] Results of the proposed color pencil sketch method	50
[Figure 29] Results show the flexibility to change angle of the hatchings (-45°)	51
[Figure 30] The obvious shadings depend on the brightness of the surface.....	51
[Figure 31] Indoor image example	51
[Figure 32] Outdoor image example	52
Figure 33] Comparison of our method with Cewu Lu method [45].....	52

Abstract

Leaf Damage Detection and Pencil Drawing Based on Image Processing Techniques

Hitimana Eric

Division of Electronics and Information Engineering

(Computer Engineering)

Advised by Oubong Gwun, PhD

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of the proposed solutions to a given problem. The given problem to deal with can be either in computer vision or computer graphics applications.

This thesis proposes two methods to handle problems in computer vision application for pest damage detection and estimation from leaf image and from Computer graphics, for rendering an automatic color pencil sketch with hatching lines from an input image.

Image segmentation is the most challenging issue in computer vision applications, and most difficulties of crops management in agriculture are the lack of appropriate and accurate methods for detecting the leaf damage for pests' treatment.

An automatic method for leaf damage detection and severity estimation by avoiding defoliation is proposed. After enhancing the contrast of the original image using LUT based gamma correction, the image is processed to remove the background, and the output leaf is clustered using Fuzzy c-means segmentation method in V channel of YUV color space to maximize all damages detection, and finally, the severity damage of leaf is measured in terms of ratio of leaf pixel distribution between the normal and the detected leaf damage. The results are compared to the current researches and the accuracy is obvious either in the background removal or damage detection. From the point of view, the proposed method can be recommended to famers as a solution of the current naked eye detection method resulting to errors and time costing.

On the other hand, Non-Photorealistic Rendering (NPR) techniques in computer graphics to render an image can be created by making use of image editing and authoring tools. However, these tools are known to bring about high production cost along with increasing the time and labor charge.

This thesis also proposes an automatic generation of pencil sketch rendition with hatching lines algorithm. From the input color image, the hatching lines are

generated where their directions depend on the choice of the orientation by motion blurring; shaded image with contour generated using Kirsch operator is produced to control the hatching texture on the final pencil sketch result. The algorithm also can come up with a colored pencil sketch where the color components are got from the original image.

The method can be used as an editing tool for computer arts, advertising industrial or technical design, etc.

Keywords: Coffee rust, Image segmentation, color and luminance, Gamma correction, Hatching, shading, kirsch operator, NPR, color pencil sketch.

Student ID Number: 201250572.

1. Introduction

1.1 Background

A computer vision system is an attempt to replicate the human eye to brain assessment process, whereby the human eye is replaced by a digital camera and the human brain is replaced by a learning algorithm. The camera can record objective and consistent image data without substantial confounding noise [1]. And image processing has been proved to be an effective tool for analysis in various fields and applications [2].

The main focus in computer graphics has always been the rendering of photorealistic imagery. While a tremendous amount of research has been done in this area, non-photorealistic rendering is a relatively young field of research and in many areas; it can be more efficient than photorealism, since it can focus on the information to be conveyed. For that reason this thesis proposed two automatic algorithms to solve computer vision and non-photorealistic problems by adopting image processing techniques.

1.2 Motivation and Objectives

In one hand, an evolution towards sustainable agriculture system, it was clear that important contributions can be made by using emerging techniques. Precision agriculture was new and developing technology which leads to incorporate the

advance techniques to enhance farm output and also enrich the farm inputs in profitable and environmentally sensible manner. With these techniques/ tools it was now possible to reduce errors, costs to achieve ecological and economical sustainable agriculture.

Coffee rust is the most economically important coffee disease in the world, and in monetary value, coffee is the most important agricultural product in international trade. Even a small reduction in coffee yields or a modest increase in production costs caused by the rust has a huge impact on the coffee producers, the support services, and even the banking systems in those countries whose economies are absolutely dependent on coffee export [3].

Infections occur on the coffee leaves. Coffee rust and other coffee pests cause premature defoliation, which reduces photosynthesis capacity and weakens the tree [4]. The most techniques used to avoid coffee rust and other pests that destroy the coffee leaves are using pesticides and fungicides, but if not controlled well, they can cause ecosystem problems.

The detection of severity of infected leaves have been done by the farmers using naked eyes, which can contribute to many errors, and the precise ways are needed to be sure the amount of pesticides or fungicides to apply while preserving the ecosystem. And most leaf diseases destroy the leaf, so that it can be easy to detect the damage using image processing techniques, but in the case of coffee rust, there is

only color change which acts as a special case.

In this thesis for the first work, we design and develop system for detecting the pests' attacks infection on image of leaf using image segmentation techniques.

The novelty of this application is that the images used are captured from the tree, to avoid defoliation. Most researches about leaf disease detection [5], [6] have been done, but they cut off the leaf and put it to the white background to be able to process it, but our algorithm considers leaves' images regardless the type of background.

In other hand, Non-Photorealistic computer graphics are used to imitate the eyes and minds of artists and designers to create, view, and depict the graphical world. These computer-generated graphics are used instead of achieving the illusion of photorealism via an optical camera.

Pencil sketch drawings are a very popular form of arts that act as most fundamental pictorial languages to abstract human perception of natural scenes. In a typical sketch image, only the most characteristic lines of the underlying subject are drawn, using a dark colour (pencil) on a white background (paper). In that context, certain degree of variation in the darkness of the pencil is typically used to depict various types of transitional boundaries (edges) and shadows in the original scene.

There are two different mechanisms used in computer-generation of pen-and-ink

drawings depending on the input object. The first one is based on 3D geometry [33], which generates pen-and-ink images from 3D geometry or through human interaction. The main advantage of that method is that they place the main strokes along the contours of the surfaces; they can also convey the 3D forms of the surfaces because they have full access to the 3D geometry and viewing transformation. The second one is based on image [34]. It offers some advantages such as reducing the tasks of geometric modelling and of specifying surface reflectance properties. Furthermore it allows more complicated models to be illustrated because obtaining high-quality pictures is much easier than constructing 3D models of the scene due to the popularity of digital cameras and internet sharing.

The second work for this thesis, we implemented an algorithm of automatically generating a color pencil sketch from an input image. Especially the method can add hatching lines to the drawings like the known expertise drawings, i. e. the Alphonse Legros sketch [35].

1.3 Outline of the Thesis

The thesis is organized as follows. Chapter 2 introduces the related works, the proposed method for the leaf damage detection and estimation, implementation and experimental results analysis. Chapter 3 introduces the related works, proposed method for color pencil sketch with hatchings, implementation and experimental results analysis. Finally, conclusion and future works are presented in chapter 4.

2. Leaf damage detection and estimation

2.1 Literature review / related works

This section surveys the related researches of image processing in agricultures, such as background removal, infected leaf segmentation and area of leaf measurement.

2.1.1 Comparative survey of leaf image processing

2.1.1.1 Leaf disease detection and classification

As a rapid, nondestructive and objective method, image processing technology has been widely used in determination of some quality characteristics of agricultural products. Leaf disease detection and classification is a hot research in plant managements and taxonomy.

Gloria D. et al. [5] proposed semi-automatic approach based on an initial pixel based classification according to the chrominance feature from the YCrCb color space. It requires user-intervention to select a sample of pixels for training the color space classifiers. Thiago L. G. Souza et al. [6] automatically classified the main agents that cause damages to soybean leaflets. After extracting the contour of the damages, they are taken as a complex network, and trained using SVM.

C.P. Wijekoon, et al. [8] used Scion Image software to quantify a wide variety of

fungus interactions with plant leaves. This software is responsible for measuring the change in leaf color caused by fungal sporulation or tissue damage. But it only deals with the detached, well placed and shadow free leaves. Qinghai He et al. [9] proposed the damage detection method by measuring the damage ratio in different color model after enhancing the leaf image, but the algorithm fails to handle the outdoor leaves for contrast and other noises. A.C. Nazare-JR. et al. [10] automatically quantified the damaged leaf area, by handling the noises and recovering the leaf contour using computational geometry, they recovered only the line segments but not the curved edges and they only detected the damaged/ destroyed parts, not the damage in terms of color change, as we are considering the healthy leaf as the one responsible for photosynthesis process.

All of those methods only handle the detached, well placed, shadow free, with simple background leaves images, because they only did the simple image thresholding to remove the background noises. On the other hand, our proposed algorithm considers leaves regardless of any background, and can also detect both damaged parts (in terms of destroying and color change).

2.1.1.2 Background removal

In still image object detection, many researchers proposed different algorithms for image segmentation purpose. Jeong-In Park et al. [11] suggested the variable order $N \times M$ dimensional vector, where the vectors are applied to the reduced objective

image to remove the background. This method does not remove the actual background; rather it regenerates the filtered replica after replicating the background. It may look like the background is actually removed after it is applied to the image, but it is reconfigured with white color lines which are smoothly processed while retaining the background. This method have been proposed to overcome the computational time of code book, but compare to our method, this is still expensive in terms of computation depend on the size of objective image as it is dealing with image reduction and $N \times M$ dimensional vector processing.

In object extraction based on detecting salient regions [12], [13], [14], [15] they ended by segmenting the image to remove the background, but sometimes this method failed to detect the object in question. Guanqun Cao et al. [15] proposed a salient object extraction with opponent color boosting, the method is based on emphasis on color in an iso-salient color space and filtering by a DoG filter afterwards, but the detection is not accurate as it includes even other non objects.

In most agriculture image processing applications, they applied a simple threshold or otsu's image segmentation [16] [10], [6] to separate the background and the foreground because the background is not complicated. Our proposed method for background removal provides an obvious accuracy to their papers as well.

2.1.1.3 Leaf Area Measurement

Accurate and rapid non-destructive leaf area measurement/estimation is important in plant understanding and modeling ecosystem function. In this section, some methods for leaf area measurement are surveyed.

Utilizing leaf area instruments, it is reliable and convenient to estimate leaf area using mechanical, digital or portable scanning planimeters [17], but the method is expensive and destructive i.e., means we need to cut off the leaf.

a. *Photoelectric scanning*, it requires scanning the reference object by specific dot per inch, calculating the area of one pixel. Then scanning the leaf by the same dot per inch (dpi), and calculating the leaf area by the equation:

$$LA = PN \times RAP \quad (1)$$

Where LA is leaf area, PN is the number of leaf scanning image pixels and RAP is area of one reference object scanning image pixel [18]. Still this method is destructive.

b. *Weighting*, copying the shape of the leaf on a piece of paper and weighing the copy. Then estimate the leaf area using equation:

$$LA = W / c \quad (2)$$

Where LA is leaf area, W is weight of the paper and c is the coefficient of the paper (weight of unit area) [20]. But it's expensive in terms of computational time.

c. *Counting grid.* Projecting leaf on a piece of grid paper and counting the number of panes. Then estimate the leaf area with equation:

$$LA = GN \times GA \quad (3)$$

Where LA is the leaf area, GN is the number of grid, and GA is area of one grid [18]. Time cost is expensive.

d. *Regression equation,* it includes two steps: Firstly one specific species of plant leaf of different size is selected for study. The length and width of the leaf is measured and the leaf area is also measured using method b) and c). And the leaf area prediction model was proposed as follow:

$$LA = f(L, W) \quad (4)$$

Where LA is leaf area, L is the leaf length and W is leaf width.

Secondly, when you want to know the specific plant leaf area, you can calculate the leaf area using the equation only by measuring the length and width of the leaf [19]. This method requires many different equations and each is applicable for one specific species of plant.

e. *LI-3000*, is a digital device manufactured by LI-COR, it is a hand-held meter for measuring leaf area, width, maximum width, length and height. It is accurate but slow and expensive. The main limitations are: It cannot handle large leaves. A large leaf has to be cut into smaller pieces and the area for each piece will have to be measured individually so that one can later obtain the area of the whole leaf, may result to the measurement errors [21].

This thesis also surveyed the current proposed method using image processing techniques. Mahdi M. Ali et al. [19], after preprocessing the leaf and detecting the edge, they used digital vernier and compared the results with Li-Cor 3100 leaf area meter. The method is not destructive but still they used some devices to measure the area. Chaohui Lu et al. [22], captured the leaf and put on a hand normal panel, i.e square (known area) drawn on a white paper, the image is processed using image processing techniques, and then the area is calculated through pixel number statistic. The accuracy can be affected by the geometric distortion of the panel. Sanjay B. Patil et al. [23], after processing the leaf image, the area is calculated by estimating the pixel statistic referencing to piece of coin (known area). The pixel count of the processed image depends on the distance between the camera and the object when the picture is taken.

From the surveyed papers, we can see that all of them have advantages and drawback. This thesis proposes to estimate the severity of the infected leaf area, the proposed algorithm handles leaf images regardless where and how the image was

taken (background, environmental condition that can be affecting the contrast of image, etc), we ended by finding that to adopt one of the proposed method above, can lead us to many errors, because the algorithm did not care how far or close the image leaf was captured/shot.

To overcome those challenging, the proposed method estimate the severity of the damaged area in percentage by calculating the ratio between the normal leaf pixel distributions to the infected pixel distributions.

2.1.2 Image Processing Techniques

In this section, the concepts of some image processing techniques that have been adopted in this thesis were discussed deeply.

2.1.2.1 Image enhancement

Image enhancement is a process of improving quality of a digitally stored image. The commonly-used enhancement techniques include contrast enhancement, histogram equalization, edges sharpening, and variety of filtering and so on [24]. The leaf image can be having low contrast or high contrast depend on the weather condition of shot, or other conditions that can cause the negative impacts on the output image results. Daniel R. Tretter et al. [25] proposed image histogram to enhance image contrast. In [26], they used histogram Normalization techniques for palm leaf Manuscript image. In this thesis, the leaf image is enhanced by using LUT based

gamma correction.

Image enhancement by Gamma Correction

Gamma refers to the brightness of a monitor or computer display. It is a setting that determines how bright the output of the display will be. Therefore, ‘*Gamma correction*’ is used to alter the output levels of a monitor.

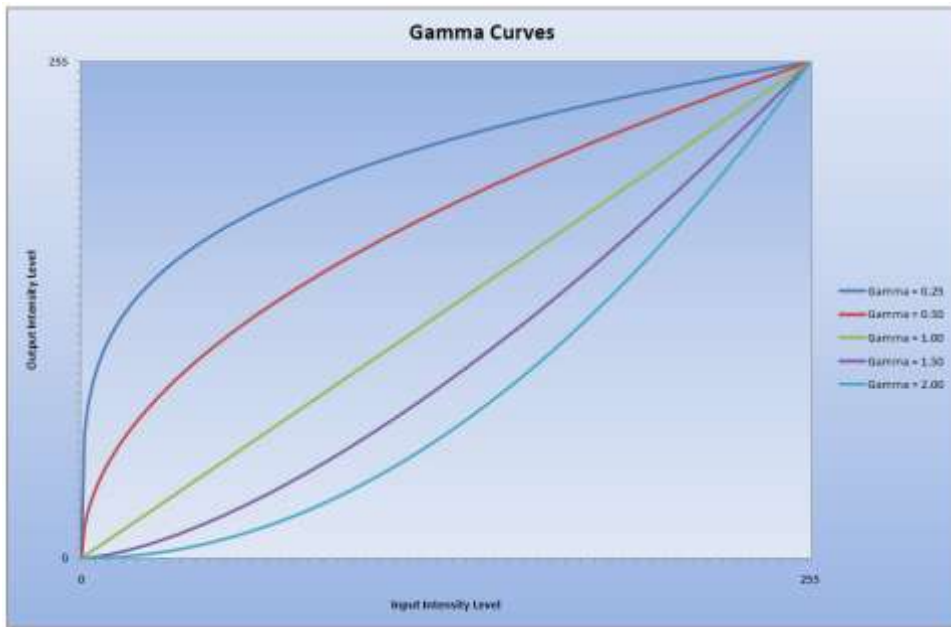
The reason for gamma correction usage is because the input signal, or voltage, sent to a monitor is not high enough to create a bright image. Therefore, if the gamma is not altered, the images on your screen would be dark and difficult to see. By applying gamma correction, the brightness and contrast of the display are enhanced, making the images appear brighter and more natural looking [27].

Gamma, represented by the Greek letter γ , can be described as the relationship between an input and the resulting output. For the scope of this thesis the input will be the RGB intensity values of an image.

The relationship in this case between the input and output is that the output is proportional to the input raised to the power of gamma. The formula for calculating the resulting output is as follows:

$$I' = 255x\left(\frac{I}{255}\right)^\gamma \quad (5)$$

Where I denotes the original input image and I' describes the output result. To illustrate here are a few gamma curves:



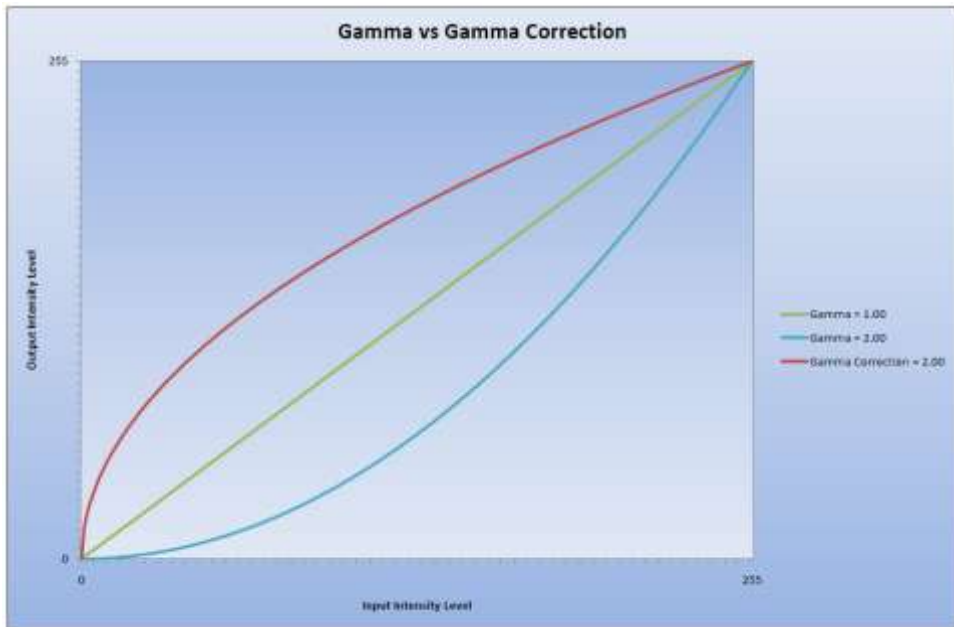
[Figure 1] Various Gamma Curves

Note that with a gamma of 1 the input equals the output producing a straight line.

For calculating gamma correction the input value is raised to the power of the inverse of gamma. The formula for this is as follows:

$$I' = 255 \times \left(\frac{I}{255} \right)^{1/\gamma} \quad (6)$$

The following graph shows a comparison between the gamma curve and the gamma correction curve:



[Figure 2] Gamma Vs Gamma Correction

The range of values used for gamma will depend on the application [28].

2.1.2.2 Image Segmentation

Image segmentation is one of the key techniques in image understanding and computer vision. The task is to divide the image into a number of non-overlapping regions, which have the same characteristics. There are different types of image segmentation (Region based, Boundary based, Edge based, clustering based ...), but

in this thesis we adopted the segmentation by clustering.

Clustering is a process whereby a data set is replaced by clusters, which are collections of data points that belong together. It is natural to think of image segmentation as clustering; we would like to represent an image in terms of clusters of pixels that belong together. The specific criterion to be used depends on the application. Pixels may belong together because they have the same color, they have the same texture, they are nearby, and so on [29].

In this section, the introductory concept of image segmentation by clustering pixels is described.

Segmentation by Fuzzy C-means clustering

Fuzzy c-means (FCM) algorithm is one of the most widely used fuzzy clustering algorithms in image segmentation. FCM algorithm was first introduced by Dunn and later extended by Bezdek. Although the conventional FCM algorithm works well on most noise-free images, it fails to segment images corrupted by noise, outliers and other imaging artifacts [30].

This section describes Fuzzy c-means clustering algorithm adopted to implement leaf image segmentation.

The idea is to represent the similarity a point shares with each cluster with a function (membership function) whose values (memberships) are between zero and

one. Each sample will have a membership in every cluster. Memberships close to unity signify a high degree of similarity between the sample and a cluster. The memberships close to zero imply little similarity between the sample and the cluster.

The equation (7) defines the objective function for partitioning a dataset $\{x_k\}_{k=1}^N$ (in the gray-level space) into c clusters.

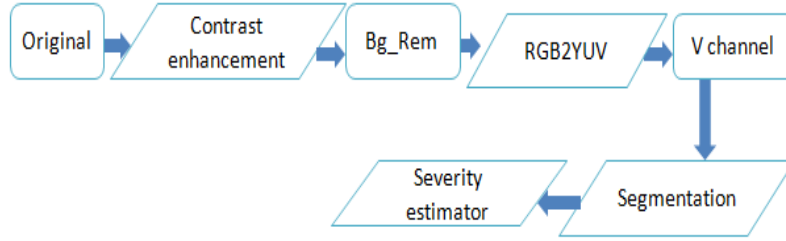
$$J_{FCM} = \sum_{k=1}^n \sum_{i=1}^c (u_{ik})^q d^2(x_k, v_i) \quad (7)$$

Where d^2 denotes the Euclidean norm, $\{v_i\}_{i=1}^c$ stands for the centers or prototypes of the clusters. The parameter q is a weighting exponent of each fuzzy membership and determines the amount of fuzziness of the resulting classification.

2.2 Proposed Leaf Damage Detection and Estimation

As far as the system is concerned, the algorithm is made by three processes. Firstly, the captured leaf is processed for contrast enhancement using LUT gamma correction algorithm. Secondly, the enhanced image is processed to remove the unwanted background. Lastly, the recovered leaf is segmented using fuzzy C means clustering to detect the infected part; the area of infected leaf is calculated by the ratio of the damaged pixel distribution to the normal leaf pixel distribution.

The figure below shows the framework of the method.



[Figure 3] Proposed framework of leaf damage detection and estimation

2.2.1 Contrast enhancement

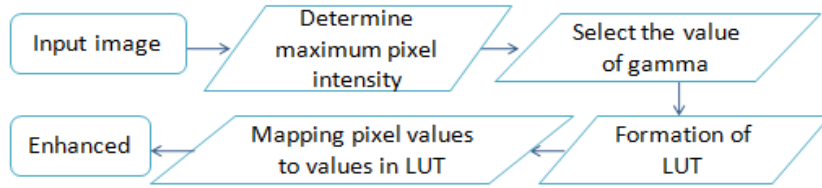
The contrast enhancement process adjusts the relative brightness and darkness of objects in the scene to improve their visibility. As the method can handle any image regardless of its condition of shot, we decided to use LUT based gamma correction algorithm to deal with the image contrast and brightness enhancement.

$$LUT = Max_int \times \left(\frac{0:Max_int}{Max_int} \right)^{1/\gamma} \quad (8)$$

$$I' = LUT(double(I)) \quad (9)$$

Look-Up Table is formulated using the maximum intensity (Max_int), and the value of gamma depends on the input image.

The original brightness value of image I is mapped to I' by using the formulated LUT as shown on figure below.



[Figure 4] Image enhancement using LUT gamma correction

This thesis proposes a method to get the value of gamma automatically from the input gray intensity value as follow.

- ① Analyze the histogram of an input gray image,
- ② Calculate the mean average intensity value

$$I_{avg} = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H I(i, j) \quad (10)$$

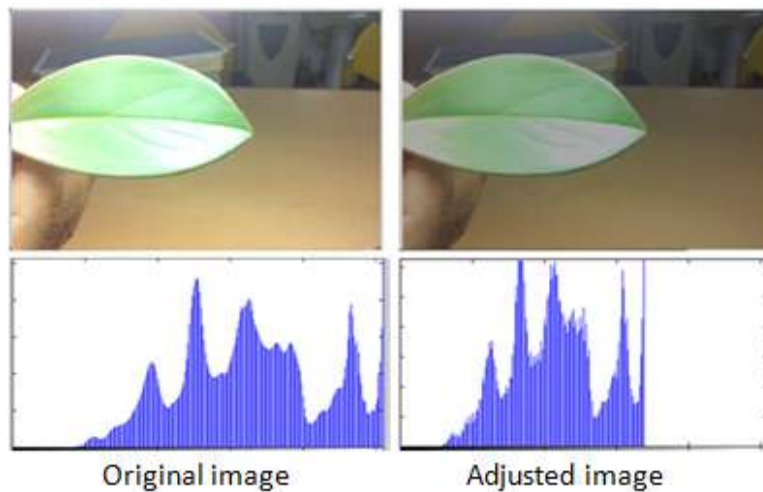
- ③ Normalize the value in range [0,1]

$$r = \frac{I_{avg}}{I_{max}} \quad (11)$$

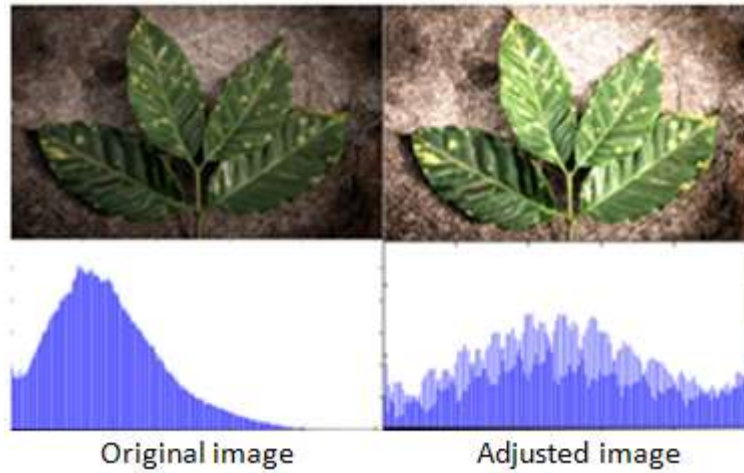
- ④ Calculate the gamma value by using equation

$$\gamma = \begin{cases} 10r - 4 & \text{for } r > 0.5 \\ 1 & \text{for } r = 0.5 \\ r^{0.1} - 0.4 & \text{otherwise} \end{cases} \quad (12)$$

In the experimental results, there is no doubt to say that in both contrast conditions (low and high); the gamma value is selected efficiently. And this proposed automatic gamma value always enhances the image based on its average gray intensity value, this method is different from the adaptive gamma correction proposed by Shi-Chia Huang et al. [31] which requires some adjustable parameters and it always enhances the low contrast, because the value of gamma is always equal or less than 1 according to their equation $\gamma = 1 - cdf_w(l)$. The figure 5 and 6 show the results for contrast enhancement.



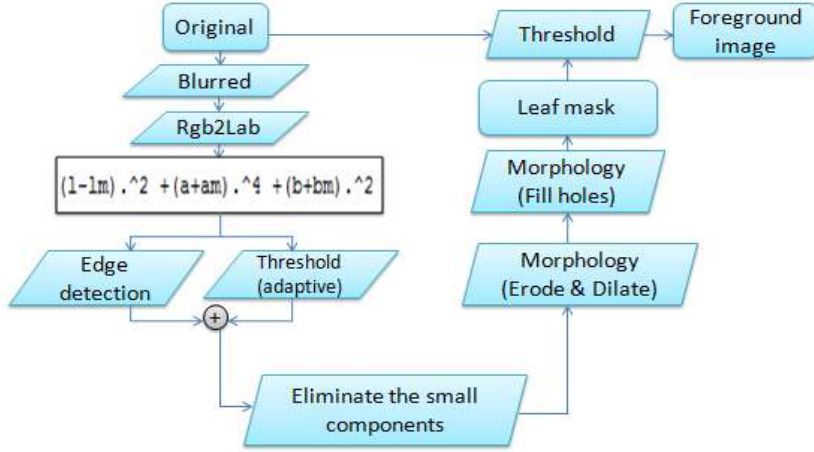
[Figure 5] Enhancement of high contrast image



[Figure 6] Enhancement of low contrast image

2.2.2 Background removal

The leaf image processed maybe having some surrounding noises that can affect the accuracy of leaf damage detection. We propose a method that can only keep the foreground object, i.e., leaf only.



[Figure 7] Image background removal

The original image is blurred using Gaussian kernel to suppress the noise, and the image is converted to CIELab color space that have been proved as the most color space to detect the object based on the salient properties [32].

$$O(x, y) = (\tilde{I}_l - I_{\mu l})^2 + (\tilde{I}_a + I_{\mu a})^4 + (\tilde{I}_b + I_{\mu b})^2 \quad (13)$$

Where I_{μ} is the arithmetic mean value of the image in each channel, \tilde{I} is the corresponding image pixel vector value in the Gaussian blurred version (using a 5x5 separate binomial kernel) of the original image. The above proposed equation can highlight the foreground object and suppress the background. The resulting image is threshold using the following adaptive threshold.

$$Th_r = \frac{1}{W \times H} \sum_{k=1}^W \sum_{i=1}^H O(x, y) \quad (14)$$

The threshold image is combined by the boundary features detected using canny edge detector to adjust the overall structure of the object. The resulting image with different separate objects is judged to remain with the biggest object among the arrays using labeling method. Image erosion and dilation algorithms are applied to adjust the objects by using the disk element of fixed size. The output image is filled to recover the internal holes. The final image is a mask of the whole object within an image, and it is used as a threshold to segment the original image. The resulting output image at this stage is the image with background free as shown in the figure below.



[Figure 8] Result for background removal

2.2.3 Leaf damage clustering

At this stage, the leaf in question is available; the only problem is to detect the damage. In most proposed methods [10], [6], they tried to detect damage in gray image because their methods only captured the destroyed part of the leaf as an infected leaf part. But our method considers a damaged part as all leaf areas that

cannot contribute to the photosynthesis process.

In our cases we detected the damage leaf area in YUV color model, and our results show a good efficiency compare to other methods. And the other advantage for using YUV color model is that, the leaf veins are not mistaken as the damage. The V channel is clustered using Fuzzy C-Mean algorithm as discussed in the introductory concept of image clustering, where only two clusters are used.

2.2.4 Leaf damage estimation

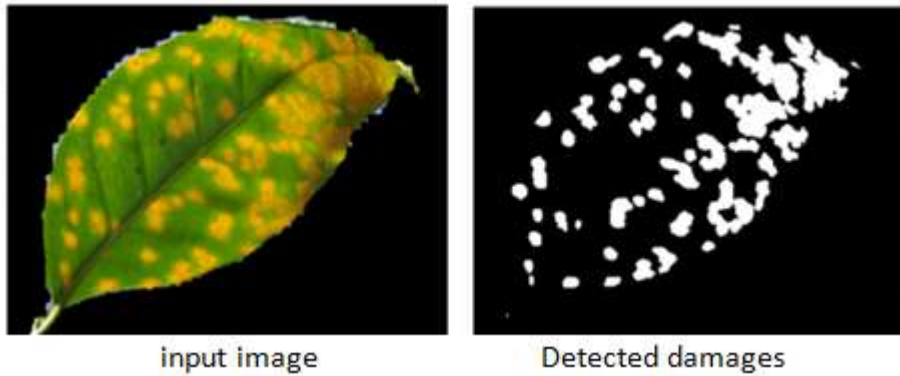
The estimation of the severity of the leaf damage was decided, to allow the farmers being able to take into account their plant management (for pesticides or fungicides utilization).

After surveying different methods used for estimating leaf damages, it is obvious that most of them cannot give good results on the sample image database used in this thesis. We decided to estimate infection by calculating the percentage of the damaged pixels statistic to the normal real leaf pixels distribution by using the following equation.

$$I_{severity} = \frac{I_{infected}}{I_{normal}} \times 100 \quad (15)$$

Where $I_{severity}$ defines the approximated value of leaf infection severity in percentage,

$I_{infected}$ shows the infected image pixels and I_{normal} specifies the normal pixel in leaf image. The figure below shows the coffee leaf with its damage segmentation results.



[Figure 9] Segmentation result of coffee rust damage infected at 26.25%.

2.3 Leaf damage detection implementation and results

2.3.1 Implementation

The algorithm was implemented with MATLAB[®]. MATLAB[®] is defined as a Matrix Laboratory which is a numerical computing environment and fourth-generation programming language developed by Math Works. Besides the matrix manipulation as basic function of MATLAB[®], it also provides many other functions: plotting of functions and data, creation of user graphical interfaces, and interfacing with programs written in other languages.

The advantage of MATLAB[®] is the multifarious toolboxes containing a number of sizeable functions that can be used directly. Most of the functions are the implementation of the classical algorithms. In the implementations, image processing and Fuzzy Logic Toolboxes were adopted.

Image Processing Toolbox [47] supports a range of image processing operations. In this thesis, we used functions such as image read and display, edge, rgb2lab, imdilate, imerode... included in Image Processing Toolbox. By the means of this toolbox, MATLAB[®] can replace OpenCV in some extent, and more accessible than OpenCV. But the weakness of MATLAB[®] is the slow running speed which limits the efficiency of the implementation.

Fuzzy Logic Toolbox [7] was also used that provides tools to create and edit fuzzy

inference systems within the framework of MATLAB[®]. The Fuzzy C-means clustering function was used to segment the leaf damage, its implementation is provided by function $[\text{center}, U, \text{objFcn}] = \text{fcm}(\text{fcmdata}, c)$, where c is number of clusters and fcmdata is input image. The variable center contains the cluster centers, U contains the membership grades for each of the data points, and objFcn contains the history of the objective function across the iterations.

2.3.2 Experimental Results and Analysis

The leaf damage detection and estimation method proposed is tested to leaves image database and the results are accurate. The method is also compared to other proposed algorithms on the same application for damage detection purpose.

From figure 10 to figure 12, the input images were taken without considering the lighting conditions and the type of background even some of the leaves are still on their trees just to avoid defoliation. After removing the background, the method tries to estimate the damage severity of the leaf.



[Figure 10] Detection with an estimation of 6%



[Figure 11] Detection with an estimation of 22%



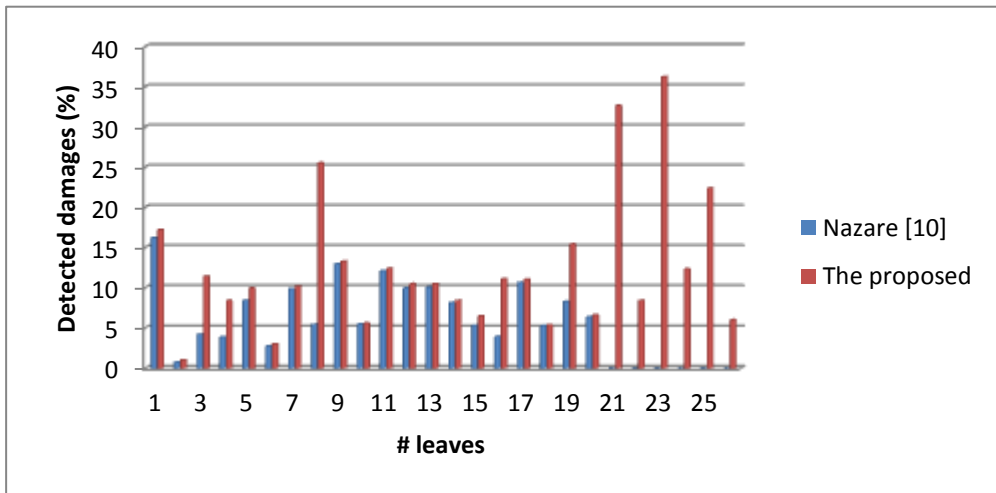
[Figure 12] Detection with an estimation of 26.25%

The background removal method shows the obvious performance on figure 14, and 15. On the other hand the damage detector's results are compared, figure 16 shows how accurate and precise the method is compared to others. From the point of view, on the third row, second column, Nazare's method [10] took the leaf as the health one, while it was 26.25% damaged. From the other leaves in first row and second row, the red mark from the input image shows the damage that were not detected by Nazare's method. Whereas the proposed method detects both damages either destroyed leaf or the color change leaf based on the anatomical process of photosynthesis mechanism.

We apply our method to Nazare's image database of soybeans and other coffee leaves. From the 27 images we tested, we can say that our method can detect both destroyed and color change damages at the same time. Nazare et al. evaluated their method by comparing with the manual segmented data by the expert in the area of Plant Science, and other proposed method of Mura. And according to their results at that time, their method was better from others.

The graph below shows how accurate our method is compared by Nazare's method. From the same 27 tested images, our mean average detection is two times than Nazare's method, i.e. 12.36 ± 8.43 and 5.76 ± 4.65 respectively (mean value and standard deviation). The strong point for our method is that we can handle all leaf damages (destroyed and color change), it can be seen from the graph that for the last six leaves, which are infected by coffee rust as described in the introduction, Nazare's method considered them healthy, while they are damaged already.

On other hand, it is also obvious that for the same destroyed leaf damages, the estimated values are almost the same, which shows that, our method works like their method and beyond for color change leaf damages.



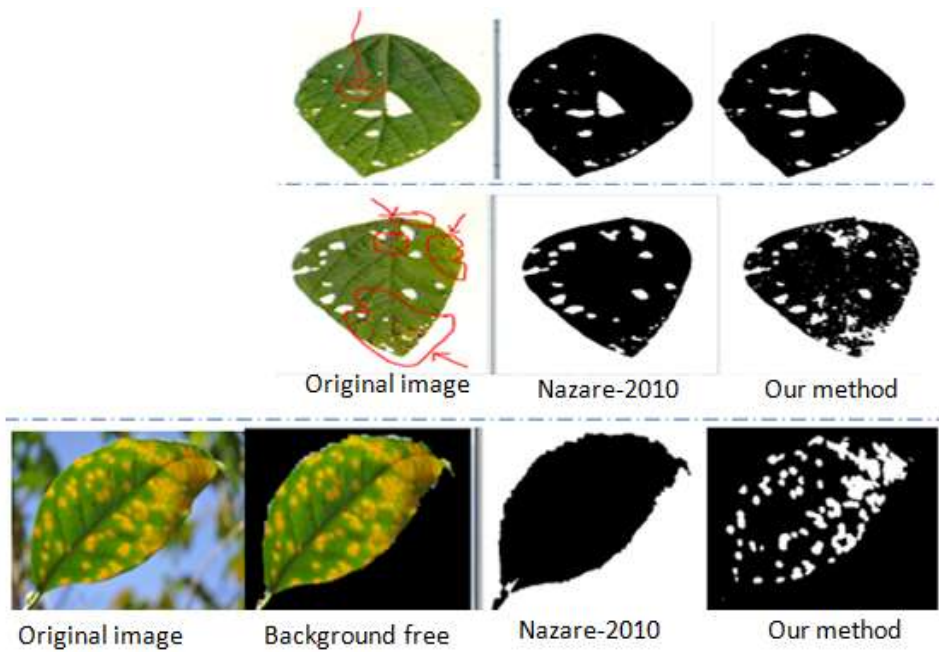
[Figure 13] Graphical comparison of damage detection



[Figure 14] Comparative results of background method



[Figure 15] Comparative results of background method



[Figure 16] Infected leaf detection comparison

3. Pencil Sketch with Hashing

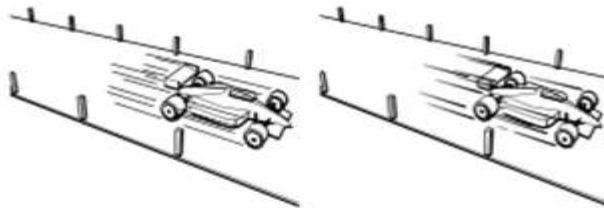
3.1 Literature review / related works

3.1.1 Applications of NPR

Non-photorealism have been proven as a potential and useful research, it can be helpful in real life applications which act as a motivation. This part describes some of its applications.

a. Non-Photorealistic Animation

Animation data can be visualized either as a moving images using NPR or even as still image with some additional markings to indicate movement.

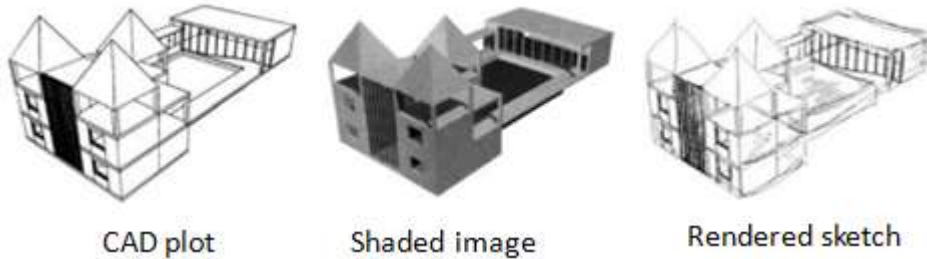


[Figure 17] Appearance of the speed lines can be varied using line styles

b. Architectural Illustrations

Most of cases, the architects prefer to use handmade sketches to show the results of their work to their clients rather than polished, photo-like images. Nowadays, some rendering software packages already contain a certain amount of functionality with

respect to NPR.



[Figure 18] Three variants of rendering a design

c. Rendering Plants

Plants in particular, trees are objects that are often needed in rendered scenes, they may be necessary because they viewers know of their existence or they can be used to convey size relationships to viewers who have a general idea of how big trees are and can thus judge the sizes of other objects with respect to them.

d. Illustrating Medical and Technical Texts

Another highly important motivation for working on NPR pertains to medical illustrations. In digital books, such illustrations can be ultimately rendered from three-dimensional geometric models so as to provide the added value users expect from images on computer screens [36].

3.1.2 Image Based Pencil Drawing

This section surveys the previous work on pencil sketching and color pencil sketching based on reference image. There are two typical approaches to create pencil drawings from reference images. One is to implicitly recover or specify the pencil drawing strokes in terms of tone values of the reference image, and then render the final pencil drawing based on the simulation model of interactive drawing. The other one is to employ the image processing and visualization techniques to directly simulate the visual effect of the pencil drawing in terms of the empirical conventions.

3.1.2.1 Image Based Pencil Drawing by Rendering Strokes

The important aspects of a pencil are first to what extent and how it has been sharpened, and second, how much pressure the user applies. Sousa and Buchanan [37] tried to develop a graphite pencil 3D renderer by simulating the drawing materials (graphite pencil and drawing paper). They employed the blender and eraser model to render the tone values using pencil hardness (degree).



[Figure 19] Interactive rendering using 6H pencil followed by interactive rendering with strokes

The problem is that its strokes are limited to the simple point shape, and it is difficult to generate the sophisticated pencil drawings, and the method required user interaction.

3.1.2.2 Image Based Pencil Drawing by Decoupling Strokes

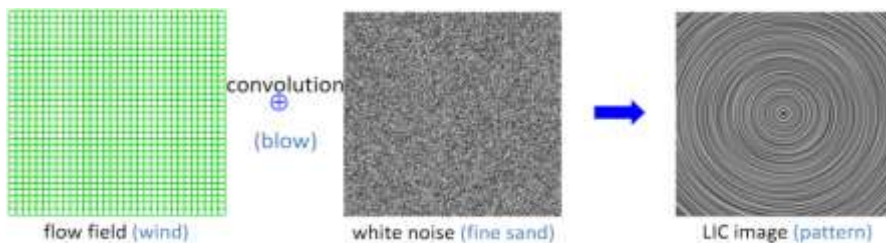
This case was adopted by Durand et al. [38], where they proposed a more general stroke model for the image-based stroke drawings. The strokes are modeled using a local threshold structure (threshold texture), which has a flat histogram and is represented as a grey-level texture map. Threshold textures can be either acquired by scanning real strokes or modeled directly. A stroke skeleton is represented by a poly-bezier curve, along which the threshold texture is warped. The equilibration of threshold structures is performed by the merged probability of the histogram of overlapping strokes using the max mode. The user can specify smudging and control the amount of detail over each part of the drawing.



[Figure 20] Basic features of Durand's interactive drawing system

3.1.2.3 Image Based Pencil Drawing by LIC

Line integral convolution (LIC) was proposed by Brian Cabral and Casey Leedom [39] as a well-known texture synthesis technique. It employs a low-pass filter to convolve an input noise texture along pixel-centered symmetrically bi-directional streamlines to exploit spatial correlation in the flow direction. LIC provides a global dense representation of the flow, analogous to the resulting pattern of a tract of wind-blown sand.



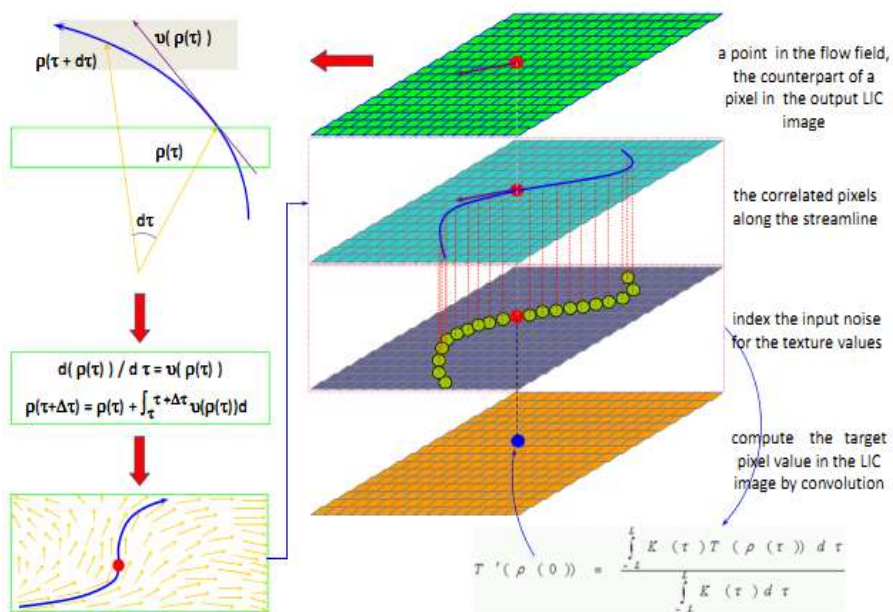
[Figure 21] Flow mechanism of LIC convolution

The mechanism below is used to create the flow field as a filter for texture synthesis. The local behavior of the vector field can be approximated by computing a local stream line that starts at the center of pixel (x, y) and moves out in the positive and negative directions. The forward coordinate advection is given by equation.

$$P_0 = (x + 0.5, y + 0.5)$$

$$P_i = P_{i-1} + \frac{V(\lfloor P_{i-1} \rfloor)}{\|V(\lfloor P_{i-1} \rfloor)\|} \Delta s_{i-1} \quad (16)$$

where $V(\lfloor P \rfloor)$ is the vector from the input vector field at lattice point $(\lfloor P_x \rfloor, \lfloor P_y \rfloor)$.



[Figure 22] Mechanism of the LIC proposed by Brian Cabral and Casey

Huang and Li, [40], used the LIC (Line Integral Convolution) algorithm to generate a pencil drawing effects. They used the following procedures.

- [1] Use color-based image segmentation to subdivide the source image into different regions,
- [2] Each region is considered as a feature region, whose boundary is extracted,
- [3] Apply image moment functions and texture analysis to obtain the feature's geometric attributes,
- [4] Generate white noise image from original image,
- [5] Apply the feature's geometric attributed to generate the vector field,
- [6] Use LIC to generate the final rendering.

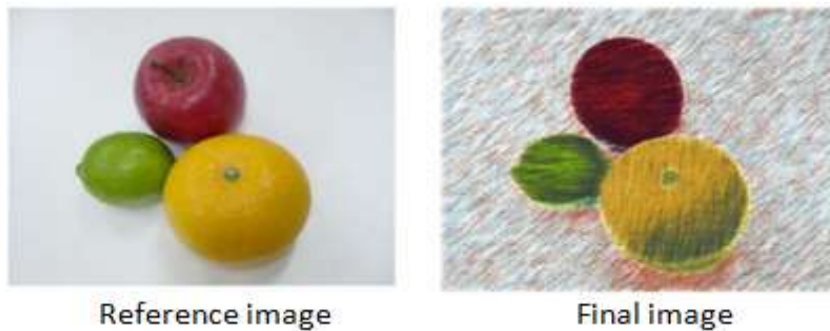
Yamamoto et al. [41] extended the LIC method to generate the colored pencil drawing. It reproduces color images with custom links to automatically select the best color set for individual regional in a source image.

The major algorithmic steps are below:

- [1] Segment the source image into different regions and decide two best colors for each region,
- [2] For each of the two colors chosen, calculate its density required for building the target color. The density is calculated in a pixel-by-pixel manner. The result consists of two density values for each pixel. Two layers grey scale images are subsequently generated with the density of each color,

- [3] Generate noise images for the two layers respectively,
- [4] For each layer, define a vector field representing the stroke directions,
- [5] Apply LIC to the vector fields and noise images to generate two layers of stroke images.

The users are allowed to specify regions and to customize the color set for a specified region interactively.



[Figure 23] Colored pencil drawing resulting from the same reference image

The drawing is obvious, but there is some lack of boundaries (contours), and the result is having some textures that are not proved by the pencil properties. And the method is interactive that means it is time consuming, and require some knowledge about painting techniques.

3.1.2.4 Image Based pencil drawing by tone and stroke combination

Based on the basis of pencil sketch visual concept; Cewu Lu et al. [45] proposed a pencil sketch approach by combining generated pencil stroke and tone drawing together.

1. Line drawing with strokes; they took an approach to put a set of less lengthy edges to approximate sketchy lines. After computing the gradients on the grayscale version of the input image, yielding magnitude

$$G = ((\partial x I)^2 + (\partial y I)^2)^{1/2} \quad (17)$$

They choose eight reference directions at 45 degrees apart and denote the line segments as $\{\mu_i\}, i \in \{1, \dots, 8\}$ and compute the response map for a certain direction as

$$G_i = \mu_i * G \quad (18)$$

Where G_i is taken as a filter response map. Then the classification is performed by selecting the maximum value among the response in all directions and is written as

$$C_i(p) = \begin{cases} G(p) & \text{if } \arg \min_i \{G_i(p)\} = i \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

To shape the lines and link edge pixels that are not connected, given the map set

$\{C_i\}$, they generated lines at each pixel by convolution.

$$S' = \sum_{i=1}^8 (\mu_i \oplus C_i) \quad (20)$$

2. To emphasize shadow shading and dark objects; they used tone drawing concept.

They proposed a parametric model to represent the target tone distribution

$$p(v) = \frac{1}{Z} \sum_{i=1}^3 w_i p_i \quad (21)$$

Where v is the tone value and $p(v)$ gives the probability that a pixel in a pencil drawing is with value v , Z is the normalization factor to make $\int_0^1 p(v) dv = 1$,

$p_i(v)$ accounts for the three tonal layers in pencil drawing, and w_i s are the weights coarsely corresponding to the number of pixels in each tonal layer.

To model the tonal distribution, they used the Laplacian distribution with its peak at the brightest value because pixels concentrate at the peak and number decrease sharply. They got color sketch by taking a sketch as luminance in YUV color model and change back to RGB.

This rendition is precise but it requires a lot of parameters for tone adjustment and the resulting sketches show the weak boundaries and even its resulting color sketches

look like oil painted.

3.1.2.5 Image Based Color Sketch by Interactive Segmentation

Given an input image, Fang Wen et al. [44] proposed a system that provided interactive segmentation option and the drawing is outlined to get the abstract of the image. The image thus becomes a set of regions $\{\Omega_i\}$ with vectorized outlines. In addition, the user needs to manually specify which region is background/foreground. Then, a two-stage stylization algorithm is used to simulate properties of the artists' free-hand drawing style.

In the first stage, they used a boundary shrinking algorithm to shrink and smooth the boundary of each region. The blank areas are based on the luminance distribution in the region and make the picture look like a freehand painting.

In the second stage, they did the color shift. Rather than simply selecting the most visually similar color from the database or giving a global transformation to the background/foreground region colors, they formulate the color shift as an optimization problem and define an energy function on candidate color pair's database, artist drawing rules and reference information from the input image to obtain a globally optimal color shift solution.

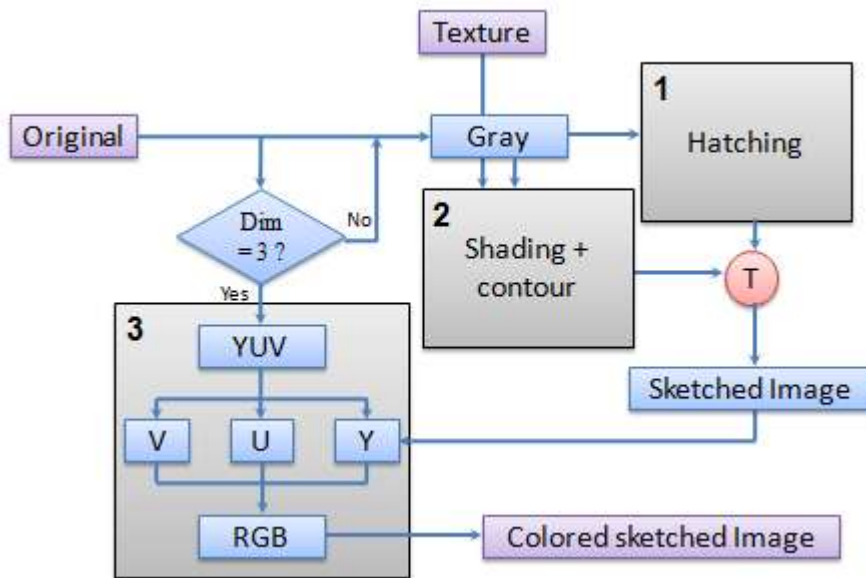
Jin Zhou and Baoxin Li's work [42] applied a gradient computation on an input image followed by some proper transformation to achieve the best effects. In [43],

Ling Xu et al. manually selected the facial region from portrait input image, to shade and enhance the areas, they used a bilateral filtering to get the base and detail layer and then they applied the stroke to non-face regions to render the silhouette, creases and some dark regions.

3.2 Proposed Color Pencil Sketch with Hatchings

Based on the motivation of Non-Photorealistic Rendering applications, and after being motivated by the handmade drawing based on different pencil style, this thesis proposes a method of automatically processing a still color input image and come up with an output result looking like a man-made hand drawing by using image processing techniques.

The following diagram shows the framework of the proposed method.



[Figure 24] Framework of proposed hatched pencil sketch image generation

As far as the overall structure is concerned, the system is made by three phases:

Hatching line generation, Shading and contour generation and coloring process where the color of the sketch is automatically got from the original input image.

The following section provides a deeply and detailed description of the proposed method to come up with a non-realistic images.

3.2.1 Hatching Line Generation

The hatching lines are used to depict tone or shading by drawing dark and parallel strokes in different regions of an image. The input image of our algorithm can be a gray value image or a colorful image, but in case of input gray value image, based on the framework of the proposed method, the resulting drawing will be having a look of a pencil sketch without color, because the third phase will not be performed.

After the conversion of the original color image to gray image, the gray image is degraded by the Gaussian noise base on its distribution. The added Gaussian noises are distributed on the gray image by replacing a certain percentage of pixel value with a uniform distribution between 0 and 255.

The resulting image is noisy, and the idea is to come up with a drawing having some hatching line mimicking the man-made drawing style, we decided to apply a motion blurred algorithm at any angle to the noisy image to decide the direction of the hatching line by using the equation (22) below.

$$h(x, y, L, \theta) = \begin{cases} \frac{1}{L} & \text{if } \sqrt{x^2 + y^2} \leq \frac{L}{2} \text{ and } \frac{x}{y} = -\tan \theta \\ 0 & \text{Otherwise} \end{cases} \quad (22)$$

This function shows how spread the points are in the image $I(x, y)$, where L – the length of the motion determines the length of hatching lines, and α – angle of direction, identifies the orientation of the hatching lines, the angle can be set depends on the user need.

The resulting image is convolved by Laplacian of Gaussian (LOG) filter to detect the edges. The hatching line image is produced by combining original gray image and the detected edge image using equation (23) -Multiplication of darkness.

$$hatching(x, y) = 1 - (1 - gray(x, y))(1 - edged_im(x, y)) \quad (23)$$

The following image shows the results of the described method for hatching line image generation.



[Figure 25] Hatching lines generation in the direction of 45°

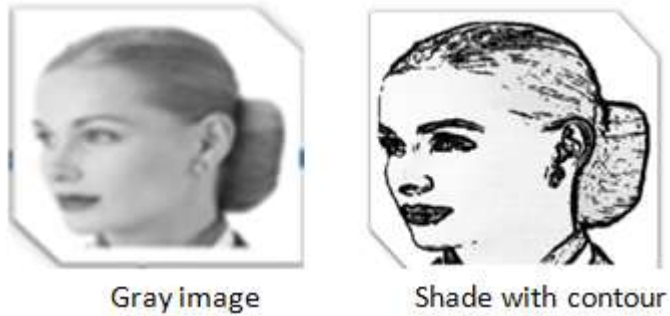
3.2.2 Shading Image with Boundary

From Non-Photorealistic rendering techniques, one of the main properties which have been taken into account is the image shading by tone or pencil graphite as we saw in the Sousa and Buchanan simulation proposal [37].

To mimic the expertise pencil graphite drawings, this thesis proposed the use of texture-based rendering style to define shading of an image. The algorithm as shown on the figure (24) takes the second input image as texture image, and combines it with an input gray image by using multiplication of darkness equation (23).

The input gray image is smoothed using Gaussian filter to suppress the noise got during the acquisition or elsewhere, and then its contour map is drawn using Kirsch mask. Kirsch edge module detects edges using eight compass filters. All eight filters are applied to the image with the maximum being retained for the final image. The threshold can be varied to fine tune the output image [48].

The resulting image is got by image fusion between the resulted image from multiplication of darkness and contour map from the Kirsch filter. The shaded image with boundary results from this stage is shown below.



[Figure 26] Shaded image with boundary

The resulting shaded image with contour is combined by the hatching lines using the following threshold method: the algorithm defines two thresholds a and b belonging in $[0, 1]$ range where $a > b$, to control the shading and hatching based on illumination and shadow of the input image. The pixels having values above a , will be white, the other below b will be given the value of shaded image with boundary. Then the pixel values between a and b will be assigned the values got by linear interpolating the two input images (shaded image with contour and hatching line image). Here the value of r as weight of linear interpolation is set depending on the behavior of the input image. The figure below shows the resulting pencil sketch image with hatching lines in the direction of 45° .



[Figure 27] Pencil sketch with hatching lines

3.2.3 Sketch Colorization

As far as the proposed pencil sketch algorithm is concerned, we can easily get the gray-scale pencil sketch with hatching effect from an input image. Beside the shown method in the section 3.1.2.3 using LIC method, other researches have been done to mimic a colored pencil sketch by using image processing techniques.

In this thesis, in order to get the colored pencil drawing effect, we adopt the method proposed by Cewu Lu [45]. After converting the original color image to YUV color model as shown on the main framework of the proposed method, where Y is the monochromatic luminance channel that we will refer to simply as intensity, while U and V, respectively, known as the chroma, describing the properties of the color saturation. The advantages of YUV color space is the brightness signal Y and chromance signals (U,V) independent of each other, and the color information is just encoded with U and V.

The trick in this thesis is to replace our resulting pencil sketch image by the current luminance and keep U and V unchanged. So the image content especially the pencil texture is preserved. The conversion formula between RGB color space and YUV color space was used. Dang-en Xie et al. [46] proposed color pencil sketch based on interactive colorization, whereas ours is automatic.

3.3 Color Pencil Sketch implementation and results

3.3.1 Implementation

The proposed method was implemented with MATLAB[®], where we adopted some of its important Toolboxes. Image Processing Toolbox [47] supports a range of image processing operations. In this thesis, we used functions such as color to gray conversion, Gaussian Blur, imnoise, edge, image read and display,... included in Image Processing Toolbox. By the means of this toolbox, MATLAB[®] can replace OpenCV in some extent, and more accessible than OpenCV. But the weakness of MATLAB[®] is the slow running speed which limits the efficiency of the implementation.

3.3.2 Experimental Results and Analysis

The proposed color pencil sketch was tested to an image database and its implementation results have been compared with the physical appearance of the

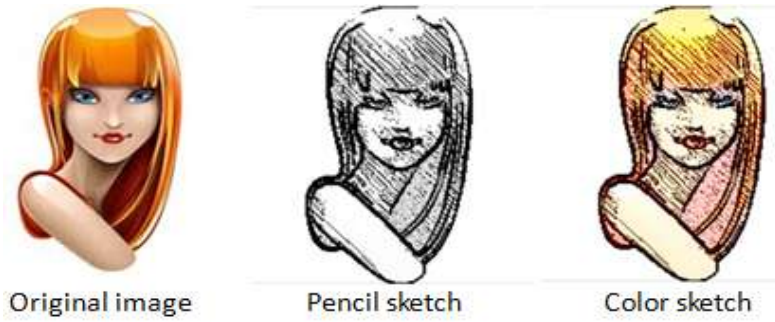
expertise based free hand drawing as we saw in the literature review, and also compared to those produced by several representative image based pencil sketching approaches.

According to the observation, it is obvious that the resulting image in Figure 28, (b) looks like a free hand drawing from expertise sketchers which is the purpose of the NPR rendering. The hatching lines are added according to the shading and luminance of the original image and the boundaries are also highlighted. The colored sketch shows the same color as the original image different from the rendering of Yamamoto et al. [41], where the textures are not oriented and the color was not exactly the same as the original input image.

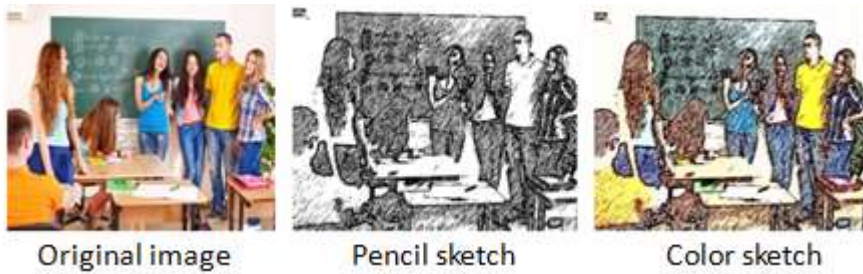


[Figure 28] Results of the proposed color pencil sketch method

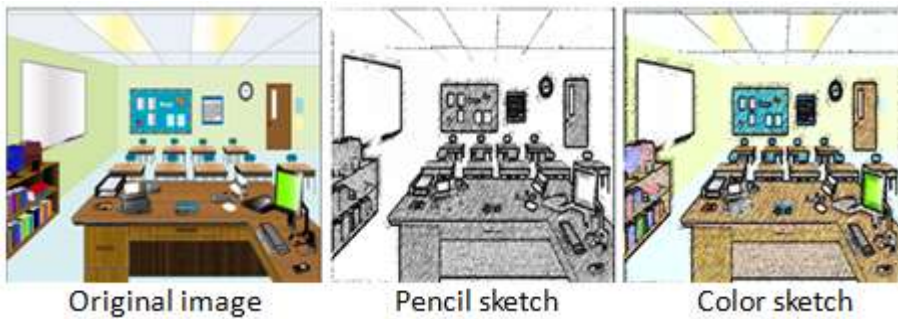
The method was tested on images of different kind, and the results show that it is a mimicked drawing.



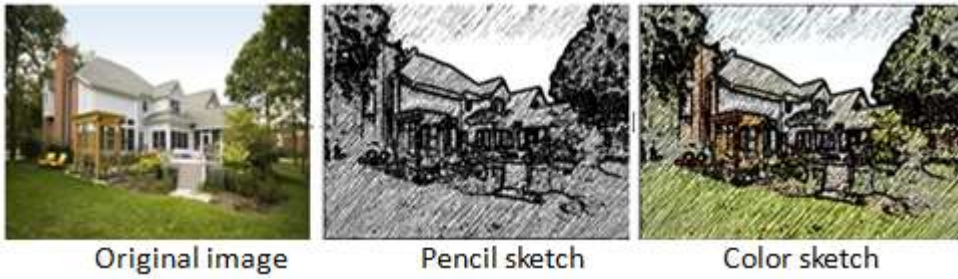
[Figure 29] Results show the flexibility to change angle of the hatchings (-45°)



[Figure 30] The obvious shadings depend on the brightness of the surface



[Figure 31] Indoor image example



[Figure 32] Outdoor image example

Figure 33 gives the comparison with the method of Cewu Lu et al. [45]. From the input image (a), our result shown in (c), presents a different drawing style –not only the contours is highlighted by sketchy lines, but also the hatchings are added based on the illumination and shadow. Whereas the shown result in (b), the appearance of the texture of background and the cloth of the man looks like an oil drawing style.



Figure 33] Comparison of our method with Cewu Lu method [45]

4. Conclusion and Future Works

4.1 Conclusion

In this thesis, two methods were proposed based on image processing techniques. Firstly, we implemented an automatic infected leaf detection and estimation algorithm that combines three processes: Image enhancement, background removal, and infected leaf detection and estimation. After enhancing the contrast by getting the value of gamma automatically in gamma correction process, the system processes the original leaf image to keep the real leaf (foreground) by using background removal method which is based on luminance and color. The background free image is then segmented in YUV color model, i.e. on V channel, to maximize the detection of the leaf damage using the Fuzzy C-means Clustering.

The estimation of the severity of infected leaf can be made very first and accurate compared to the necked eye process used by the farmers. And this measure can help them to be sure which quantity of pesticides or fungicides their fields (coffee) require for easy and quick crops management resulting to environmental good treatment.

The proposed method was compared with other researches, and it is obvious that it can over perform the other methods either in background removal or in infection detection even the method is fast; and avoids defoliation done by all other methods surveyed.

Secondly, the algorithm of automatically generating a color pencil hatching sketch from a photo was implemented. The method is based on blurring technique to control the direction of the hatching lines and input texture image to deal with shading and shadow. The experimental results show that the proposed method can generate pencil sketch hatching image from photos of any object. The method can be used as an editing tool for computer arts, advertising industrial or technical illustration design, etc.

4.2 Future Works

Some results of the proposed methods are not good for the specific purposes depend on the application.

In case of the leaf pest damage detection and estimation application, some leaves images were taken directly from trees, the method can detect the branch of tree as damage, and for the overlapped leaves, the method counts it as one leaf. We are planning in the future work to extend the algorithm for detecting the pure single leaf to be able to estimate the better accuracy of the pest damage. And we are planning to extend the method to the real-time application for more flexibility for the famers.

For the case of the automatic color pencil sketch method, our results show only the parallel hatching lines in one direction on the drawing, the plan is to extend the algorithm to mix different hatching style based on free hand drawing rules (parallel

hatching lines, cross hatching lines, etc.). The method is accurate on the still images; our future work will be concentrating on the real-time sketching for animation.

Reference

- [1] Hasan M. Velioglu, Ozgur S. 'Evaluation of Insect Damage on Beans using Image Processing Technology', 5th International Workshop on Computer Image Analysis in Agriculture CIGR-Agency2012, Spain, 2012.
- [2] Anup Vibhute and S K Bodhe, 'Applications of Image Processing in Agriculture: A Survey', International Journal of Computer Application (IJCA), New York, USA, 2012, pp.34-40.
- [3] International Coffee Organization, 'Benefits of Membership of the International Coffee Agreement', 2007.
- [4] Thurston, 'Tropical Plant Diseases', H.D American phytopathological Society, St. Paul, Minnesota, 1998, p123-127.
- [5] Gloria Diaz, Eduardo R. Juan R. B. Norberto M., 'Recognition and Quantification of Area Damaged by Oligonychus Perseae in Avocado Leaves', CIARP 2009, Mexico, 2009, pp.677-684.
- [6] Thiago L., G,Souza, Eduardo S. M., Kayran Dos S, David M., 'Application of Complex Networks for Automatic Classification of Damaging Agents in Soybean Leaflets', IEEE International Conference in Image Processing, 2011.
- [7] J.-S. Roger Jang and Ned Gully. 'MATLAB[®] Fuzzy Logic Toolbox'.

- [8] C.P. Wijekoon, P.H. Goodwin, T. Hsiang, 'Quantifying Fungal Infection of Plant Leaves by Digital Image Analysis Using Scion Image Software', *Journal of Microbiological Methods*, 2008, pp.94-101.
- [9] Qinghai He, Benxue Ma, Duanyang Qu, Qiang Zhang, Xinmin Hou, Jing Zhao, 'Cotton Pests and Diseases Detection Based on Image Processing', *TELKOMNIKA*, June 2013, pp.3445~3450.
- [10] A.C. Nazare-JR., D. Menotti and J.M.R. Neves and T. Sedyima, 'Automatic Detection of the Damaged Leaf Area in Digital Images of Soybean', *IWSSIP2010*, Rio de Janeiro, Brazil, 2010.
- [11] Jeong-In Park and Jin-Tak Choi, 'A Background Removal Algorithm using the Variable Order $n \times m$ dimensional Vector', *Proceedings, The 3rd International Conference on Circuits, Control, Communication, Electricity, Electronics, Energy, System, Signal and Simulation (SERSC)*, 2013.
- [12] Yiqun Hu, Xing Xie, Wei-Ying Ma, Liang-Tien Chia and Deepu Rajan, 'Salient Region Detection using Weighted Feature Maps based on the Human Visual Attention Model', *5th Pacific Rim Conference on Multimedia*, Japan, 2004. *Proceedings, Part II*, pp. 993-1000.
- [13] Ming-Ming Cheng, Gup-Xi Xiang, Niloy J. Mitra, Xiaolei Huang, Shi-Min Hu, 'Global Contrast based Salient Region Detection', *CVPR 2011*, pp.409-416.

- [14] Jiang H., Wang J., Yuan Z., Liu T., Zheng N., Li S. 'Automatic Salient Object Segmentation Based on Context and Shape Prior', In proceedings : BMVC 2011, pp.110.1-110.12.
- [15] Guanqun Cao, Faouzi Alaya Cheikh, 'Salient Region Detection with Opponent Color Boosting', Visual Information Processing (EUVIP), 2nd European Workshop on, 2010, pp.13-18.
- [16] N. Otsu, 'A threshold Selection Method from Gray-level Histogram', IEEE Transactions on System Man Cybernetics, vol. SMC-9, no.1, 1979, pp.62-66,
- [17] Daughtry C, 'Direct Measurements of Canopy Structure'. Rem. Sens. Rev. 5(1):45-60.
- [18] Zhichen Li, Changying Ji, Jicheng Liu, 'Leaf Area Calculating based on Digital Image, IFIP, 2008, pp.1427-1433.
- [19] Mahdi M. Ali, Ahmedi Al-Ani, Derek Eamus and Daniel K.Y. Tan, 'A New Image Processing based Technique for Measuring Leaf Dimensions', American-Eurasian J.Agric. & Environ. Sci, 2012, pp 1588-1594.
- [20] Li, Z., C. Ji and J. Liu, 2008, 'Leaf Area Calculation Based on Digital Image'. Computer and Computing Technologies in Agriculture, 259:1427-33.
- [21] You-Wen, T. and W. Xiao-Juan, 'Analysis of Leaf Parameters Measurement of Cucumber Based on Image Processing', paper presented to the world congress on Software Engg, Xiamen, China, 2009.

- [22] Chaohui, Yibin Zhang, Yinhua Shen, 'Leaf Area Measurement Based on Image Processing', International Conference on Measurement Technology and Mechatronics Automation', China, 2010, pp. 580-582.
- [23] Sanjay B. Patil and Shrikant K. Bodhe, 'Betel Leaf Area Measurement Using Image processing', IJCSE, 2011, vol.3, pp.2656-2660.
- [24] Zhang Lili, Hao Tianyao, Wu Jiansheng, Wang Jialin, 'Application of Image Enhancement Techniques to Potential Field Data', Applied Geophysics, 2005.
- [25] Daniel R. Tretter et al. 'System and Method for Histogram Based Image Contrast Enhancement', United States Patent, 1995.
- [26] Zhixin Shi, Srirangaraj S., Venu G., 'Digital Enhancement of Palm Leaf Manuscript Image using Normalization Techniques', CEDAR, 2005.
- [27] S. Jayaraman, S. Esakkirajan, and T. Veerakumar, 'Digital Image Processing', Book 2009, page 258-259.
- [28] Francis G. Loch's, 'Image processing',
(<http://www.dfstudios.co.uk/articles/image-processing-algorithms-part-6/>).
- [29] Guy B. C. and Harry C. Andrews, 'Image Segmentation by Clustering', Proceedings of the IEEE 1979, pp.773-781.
- [30] Maoguo Gong, Yan Liang, Jiao Shi, Wenping Ma, and Jingjin Ma. 'Fuzzy C-Means Clustering with Local Information and Kernel Metric for Image

- Segmentation', IEEE Transactions on Image Processing, 2013, vol.22, pp.573-584.
- [31] Shi-Chia Huang, Fan-Chieh Cheng, and Yi-Sheng Chiu, 'Efficient Contrast Enhancement using Adaptive Gamma Correction with Weighting Distribution', IEEE Transaction on Image processing, 2012, vol.22, no. 3 pp:99.
- [32] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk, 'Frequency-Tuned Salient Region Detection', Computer Vision and Pattern Recognition, CVPR 2009, pp.1597-1604.
- [33] Georges Winkenbach and David H. Salein. 'Computer-Generated Pen-and-Ink Illustration'. Processings of SIGGRAPH 94, pp 91-100, 1994.
- [34] P. Salisbury, Michael T. Wong, John F. Hughes, and David H. Salein. 'Orientable Textures for Image-Based Pen-and-Ink Illutration'. Proceedings of SIGGRAPH 97, pp 401-406, 1997.
- [35] Betty Edwards 'The New Drawing on the Right Side of the Brain', New York, p. 208, 1999.
- [36] Thomas Strothotte and Stefan Schlechtweg, 'Non-Photorealistic Computer Graphics Modeling, Rendering and Animation', 2002.
- [37] Sousa and Buchanan, 'Computer Generated Graphite Pencil Rendering of 3D Polygonal Models', EuroGraphics '99, 1999.

- [38] Durand F, Ostromoukhov V, Miller M, Duranleau F, Dorsey, ‘Decoupling Strokes and high-level Attributes for Interactive Traditional Drawing Rendering Techniques’, Springer-Verlag/ Wion, New York, 2001.
- [39] Brian Cabral and Casey Leedom, ‘Imaging Vector Fields Using Line Convolution’, proceedings of ACM SigGraph 93, Anaheim, California, pp.263-270, 1993.
- [40] Huang and Li, ‘A method of Feature-based Pencil Drawing’, In: Proceedings of the 1st International Conference on Computer Graphics and Interactive Techniques on Australasia and South East Asia, 2003, pp.135-140.
- [41] Yamamoto S., Mao X, Imamiya, ‘Colored Pencil filter with Custom Colors’, In: Proceedings of 12th Pacific conference on Computer Graphics and Applications, 2004, pp. 329-338.
- [42] Jin Zhou and Baoxin Li, ‘Automatic generation of pencil-sketch like drawings from personal photos’, ICME 2005, pp. 1026-1029.
- [43] Ling Xu and David Mould, ‘A hybrid Image-Based Method to Generate Sketching Portrait’, Graphics Interface 2010.
- [44] Fang Wen Qing Luan†* Lin Liang Ying-Qing Xu Heung-Yeung Shum, ‘Color sketch generation’, NPAR’06, pp. 47-54, 2006.
- [45] Cewu Lu, Li Xu and Jiaya Jia, ‘Combining Sketch and Tone for Pencil Drawing Production’, NPAR 12, pp. 65-73, 2012.

- [46] Dang-en Xie, 'A Colored Pencil-drawing Generating Method Based on Interactive Colorization', CCIE'10, pp. 166-169, 2010.
- [47] 'Image Processing Toolbox for use with MATLAB®'.
- [48] Muthukrishnan and Radha, 'Edge Detection techniques for image Segmentation', International Journal of Computer Science & Information Technology (IJCSIT), 2011, pp.259-267.

요 약

이미지 처리 기술 기반의 앞손상 탐지 및 연필 드로잉

전북대학교 전자정보공학부 컴퓨터공학

히티마나 에릭

지도 교수: 권 오 봉

디지털 이미지 프로세싱은 주어진 문제에 대해서 제안된 해법의 타당성을 검증하는데 많은 실험이 필요한 분야이다. 주어지는 문제 또한 컴퓨터 비전이나 컴퓨터 그래픽스와의 매우 밀접한 관계를 가지고 있다.

본 논문에서는 컴퓨터 비전과 관련이 있는 “앞 영상으로부터 병충해 부분 검출 및 추정” 과 컴퓨터 그래픽스와 관련이 있는 “입력영상으로부터 해칭 라인을 특징으로 하는 색연필 스케치 영상의 자동렌더링” 문제를 디지털 이미지 프로세싱 기법과 융합하여 해결하는 방법을 제시하였다.

농작물 관리에서 가장 중요한 부분은 병충해를 치료하는 것인데 치료에

필수적인 잎의 손상 정도를 정밀하고 정확하게 검출할 수 있는 방법을 쉽게 찾을 수 없다는 점이다. 본 논문의 전반부에서는 잎을 파괴하지 않고 자동으로 손상 부위를 검출하고 추정하는 방법을 제안한다.

처음에 감마보정 기반의 룩업테이블을 이용하여 원래 이미지의 콘트라스트를 강화한 후 배경을 제거하였다. 다음에 YUV 칼라공간의 V 채널 값을 Fuzzy c-means 세그멘테이션에 적용하여 병든 부분 전체를 검출하였다. 마지막으로 잎의 손상 부위를 정상 픽셀 부분과 병든 픽셀 부분과의 비율로서 검출하였다. 배경부를 제거하고 손상부위를 검출하는데 있어서 최근의 연구결과와의 비교는 제안한 방법이 다른 연구 결과 보다 우수함을 보였다. 이러한 연구결과로부터 판정에 기복이 심하고 경우에 따라 부정확한 농부의 육안으로 병충해를 검사하는 방법으로 대체 가능성을 확인 할 수 있었다.

한편 스케치화나 유화같은 비사실적 렌더링(Non-Photorealistic Rendering)을 영상편집기 또는 웹 저작도구를 이용하여 수작업으로 만들기에는 노력과 비용이 많이 든다. 본 논문의 후반부에서는

해칭라인을 갖는 연필 스케치화를 자동으로 생성하는 방법을 제안하였다.

입력 칼라 영상으로부터 해칭 라인을 생성하는데 해칭의 방향은 모션 블러링의 방향에 맞추었다. 최종 연필 스케치화의 결과에 해칭 텍스처를 반영하기 위하여 Kirsch 연산자로 생성한 윤곽선을 갖는 음영 영상을 생성하였다. 마지막으로 원래의 영상으로부터 색 성분을 얻어 색 연필 스케치 화를 완성하였다. 제안된 방법은 순수 미술, 광고용 영상, 산업용 영상을 편집하는 도구로 활용이 기대된다.

키워드: 커피 앞마름병, 영상 분할, 색상 및 휘도, 감마보정, 해칭, 음영, 키르쉬 연산자, 비사실적 렌더링, 색연필 스케치.

Acknowledgement

My special thanks to Almighty God who protected me until now and from whom all knowledge comes. I would like to express my sincere gratitude to my advisor Professor Ou-bong Gwun for his guidance, support, patience, motivation, enthusiasm and immense knowledge throughout all the time of research and thesis writing.

Besides that, I would like to thank the rest of my thesis committee: Professor Soon-cheol Park and Professor Gi-hwan Cho for their encouragement, insightful comments and suggestions.

I thank my fellow members of Computer Graphics Lab who made this venture more enjoyable in the past two years. What's more, I wish to say "Thanks!" to my R-wonder team for their accompanying which made me not lonely.

Last but not the least; I greatly appreciate my parents, brother and sisters for their encouragement and endless love. My heart swells with gratitude to all people who helped me.