VISUAL INSPECTION OF PHARMACEUTICAL COLOR TABLETS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

DENİZ AKTÜRK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

APRIL 2006

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. İsmet Erkmen
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Kerim Demirbaş
Supervisor

Examining Committee Members

Assoc. Prof. Dr Aydın Alatan (METU, EE)

Prof. Dr. Kerim Demirbaş (METU, EE)

Assist. Prof. Dr A. Özgür Yılmaz (METU, EE)

Assist. Prof. Dr Çağatay Candan (METU, EE)

Assoc. Prof. Dr Fahrettin Arslan (Ankara Unv, STAT)

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Deniz Aktürk

Signature :

ABSTRACT

VISUAL INSPECTION OF PHARMACEUTICAL COLOR TABLETS

Aktürk, Deniz
M.Sc., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Kerim Demirbaş

April 2006, 103 pages

In this work a machine vision system for inspecting pharmaceutical color tablets is presented and implemented. Nonparametric clustering based segmentation is faster and thus more appropriate for real-time applications. Two nonparametric clustering based methods, Nearest Neighbor algorithm and MaxShift algorithm are worked in RGB and HSV color spaces as the segmentation step. The implemented algorithm allows the system to detect the missing and broken tablets, tablet fragments, and the color, size, and shape of individual tablets in pharmaceutical blisters, in real-time. System has two operation modes called ‘‘training’’ and ‘‘inspection’’ mode, respectively. Operator selects one point on any tablet in a defect-free training captured image in the ‘‘training’’ mode. In the correction step an optimization algorithm is required, for which Powell and Downhill Simplex methods are used. Captured image is then corrected for spatial color nonuniformity,

segmented, and the position, size, shape, and color of each tablet are extracted in the “training” mode. The correction and segmentation models; the extracted features generated in the “training” mode is saved with the user defined values to form the model. Each acquired image in the “inspection” mode is corrected and segmented according to the blister model and then the blisters are classified as ‘‘good’’ or ‘‘bad’’ by comparing the extracted feature values with the user defined tolerances stored in the blister model.

Keywords: Visual Inspection, Color Tablets, Nonuniformity correction, Segmentation

ÖZ

RENKLİ MEDİKAL TABLETLERİN GÖRSEL DENETİMİ

Aktürk, Deniz
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Kerim Demirbaş

Nisan 2006, 103 sayfa

Bu çalışmada, renkli medikal tabletlerin denetlenmesi için bir makine görüşü sistemi gerçekleştirilmiş ve sunulmuştur. Parametrik olmayan küme tabanlı dilimleme yöntemi hızlı olduğu için gerçek zamanlı uygulamalar için daha uygundur. İki parametrik olmayan küme tabanlı dilimleme yöntemi, En Yakın Komşu ve MaksimumKaydırma algoritmaları, RGB ve HSV renk uzaylarında çalıştırılmış ve dilimleme aşaması olarak kullanılmıştır. Uygulanan algoritma sistemin eksik ve kırık tabletleri; tablet çatlaklarını; her bir tabletin rengini, boyutlarını ve biçimini gerçek zamanlı olarak denetlemesine olanak tanımaktadır. Sistem, “öğretme” ve “izleme” adlarında iki çalışma moduna sahiptir. “Öğretme” modunda operatör, hatasız tabletlerden oluşan bir görüntüde herhangi bir tablet üzerinde bir nokta seçer. Daha sonra çekilen resim uzaysal renk düzensizliği için düzeltilir; dilimlenir; her bir tabletin pozisyon, boyut, biçim ve renk bilgileri “öğretme” modunda elde edilir.

Düzeltme aşamasında gerekli olan optimizasyon algoritması için Powell ve Downhill Simplex yöntemleri kullanılmıştır. Modeli oluşturmak üzere kullanıcının girdiği değerlerle birlikte, “öğretme” modunda üretilen düzeltme, dilimleme modelleri ve elde edilen tablet özellikleri kaydedilir. “İzleme” modunda, alınan her bir görüntü modele göre düzeltilir, dilimlenir ve tablet özellikleri, modelde bulunan kullanıcının tanımladığı tolerans değerleriyle karşılaştırılarak, tablet grupları “iyi” veya “kötü” olarak sınıflandırılır.

Anahtar Kelimeler: Görsel Denetim, Renkli Tabletler, Düzensizlik Düzeltmesi, Dilimleme

To My Family

ACKNOWLEDGMENTS

The author wishes to express his deepest gratitude to his supervisor Prof. Dr. Kerim Demirbaş for his guidance, advice, criticism, encouragements and insight throughout the research.

TABLE OF CONTENTS

PLAGIARISM.....iii

ABSTRACT.....iv

ÖZ.....vi

ACKNOWLEDGEMENTS.....ix

TABLE OF CONTENTS.....x

LIST OF TABLES.....xii

LIST OF FIGURES…….....xiii

CHAPTER

1 INTRODUCTION..... 1

1.1 INTRODUCTION 1

1.2 SCOPE OF THE THESIS 4

1.3 ORGANIZATION 5

2 VISUAL INSPECTION SYSTEM..... 6

3 TRAINING MODE 10

3.1 SPATIAL NONUNIFORMITY CORRECTION..... 11

3.1.1 THEORY 12

3.1.1.1 FORMULATION 12

3.1.1.2 STRATEGY 13

3.1.2 LINEAR MODELING 16

3.1.2.1 MODELING 17

3.1.2.2 CORRECTION 24
3.1.2.2.1 Minimization or Maximization of Functions..... 25
3.1.2.2.1.1 Bracketing a Minimum 29
3.1.2.2.1.2 Brent's One Dimensional Optimization Algorithm .. 29
3.1.2.2.1.3 Powell's Multidimensional Direction Set Method ... 34
3.1.2.2.1.2 Downhill Simplex Method 36
3.1.2.3 IMPLEMENTATION 39
3.2 IMAGE SEGMENTATION…….....……....47
3.2.1 RGB & HSV COLOR SPACE 48
3.2.1.1 RGB TO HSV CONVERSION 50
3.2.1.2 HSV TO RGB CONVERSION……...51
3.2.2 MAXSHIFT ALGORITHM 52
3.2.3 NEAREST NEIGHBOR ALGORITHM 56
3.3 FEATURE EXTRACTION..... 58
4 INSPECTION MODE 65
4.1. SPATIAL NONUNIFORMITY CORRECTION…….65
4.2. IMAGE SEGMENTATION……..66
4.3. DEFECTIVE TABLET DETECTION 67
5 EXPERIMENTS…….68
6 CONCLUSIONS…….98
REFERENCES……..102

LIST OF TABLES

Table 5.1 Process time comparison table for different algorithms using 100 sample images..... 95

Table 5.2 Error Performance table for different algorithms using 100 sample images.....96

LIST OF FIGURES

Figure 1.1 A simplified diagram of a typical machine vision system.....2

Figure 1.2 A defect free sample image and the corresponding RGB color space.....3

Figure 2.1 GUI of the implemented software.....6

Figure 2.2 Training mode operation, inspection mode operation…….....7

Figure 3.1 A defect-free training image…….....10

Figure 3.2 Image degradation and retrospective correction of spatial color nonuniformity based on information minimization….....15

Figure 3.3 Retrospective correction of spatial color nonuniformity based on information minimization.....16

Figure 3.4 Extrema of a function in an interval…….....26

Figure 3.5 Convergence to a minimum by inverse parabolic interpolation.....31

Figure 3.6 Powell's conjugate direction set method.....36

Figure 3.7 Simplex with Pr…….....38

Figure 3.8 Simplex with Pr and Prr.....38

Figure 3.9 Simplex with Pr, Prr' and Prr''.....39

Figure 3.10 HSV hexacone.....49

Figure 3.11 RGB color cube along its principal diagonal…….....50

Figure 3.12 Obtainable HSV color from RGB color space.....50

Figure 3.13 Mode seeking with max shift algorithm using a cube-kernel….....54

Figure 3.14 Determination of clusters in the feature space…….....…….54

Figure 3.15 Clusters of different colors separated by an optimal 3D boundary.....56

Figure 5.1 The acquired defect free image of pink tablets with spatial intensity nonuniformity, in training mode and its components….....……69

Figure 5.2 The corrected image with one broken tablet and its components in the inspection mode.....72

Figure 5.3 The corrected image with one missing tablet and a tablet having a surface defect; and its components in the inspection mode.....74

Figure 5.4 The corrected defect free image of orange tablets with spatial intensity nonuniformity, in training mode and its components….....76

Figure 5.5 The corrected image with one missing tablet and and its components in the inspection mode…….....78

Figure 5.6 The corrected image with one broken tablet and its components in the inspection mode…….....80

Figure 5.7 The corrected image with one tablet with a surface defect and its components in the inspection mode…….....81

Figure 5.8 The corrected image with a broken tablet, a missing tablet and one tablet with a surface defect and its components in the inspection mode.....82

Figure 5.9 Training mode result of a defect free image using MaxShift algorithm in RGB color space.....84

Figure 5.10 Inspection mode result of a blister of defected tablets using MaxShift algorithm in RGB color space.....84

Figure 5.11 Training mode result of a defect free image using Nearest Neighbor algorithm in RGB color space.....85

Figure 5.12 Inspection mode result of a blister of defected tablets using Nearest Neighbor algorithm in RGB color space.....85

Figure 5.13 Training mode result of a defect free image using Nearest Neighbor algorithm in HSV color space.....86

Figure 5.14 Inspection mode result of a blister of defected tablets using Nearest Neighbor algorithm in HSV color space.....86

Figure 5.15 Inspection mode results for tablet images.....88

Figure 5.16 Inspection mode results for screw image.....97

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Machine vision systems, whose goal is to create a model of the real world from images, are emerging as more and more popular solutions for industrial production, either as robot vision systems or as automated inspection systems. The main task of a machine vision system is to provide computer understandable descriptions of objects from either single image or whole array of images.

Machine vision systems have many advantages. They can perform some tasks more effectively than people, especially tasks that involve highly repetitive inspection of products. Machine vision systems are consistent. Unlike people, machine vision systems apply the same rules time after time, they do not get bored or distracted. Another advantage is their preciseness, depending on where the camera is placed -on top of a microscope or on a satellite- a machine vision system can measure anything from micrometers to kilometers. Machine vision systems are through, they can inspect every single product. Often, it is too costly and time-consuming to have people do this task. There are systems that operate faster than video cameras can generate images, i.e. process more than 25 images per second. These systems can be used in hazardous environments, in chemical factories for example or in places where temperatures are extremely high.

A simplified diagram of a typical machine vision system is shown in Figure 1.1. Essentially; the image from the camera is digitized and stored in the computer where it is automatically analyzed to extract the required information. The camera can either be a standard video camera, or in the cases of products on a conveyor, a linear array sensor which generates a high resolution image on a line-by-line basis. It is usually not necessary to store an ideal image of the item being inspected. Rather, certain features of a correct item are combined to form a model of an ideal product. The same features are extracted from each inspected item and compared with the model. In this way factors such as positional alignment, changes in illumination levels and tolerances in specifications are accommodated. Often a conventional computer is not fast enough for real applications. In such cases special hardware must be used for parts of the problem. This can take the form of custom designed preprocessing hardware or commercially available high-speed vision architectures.

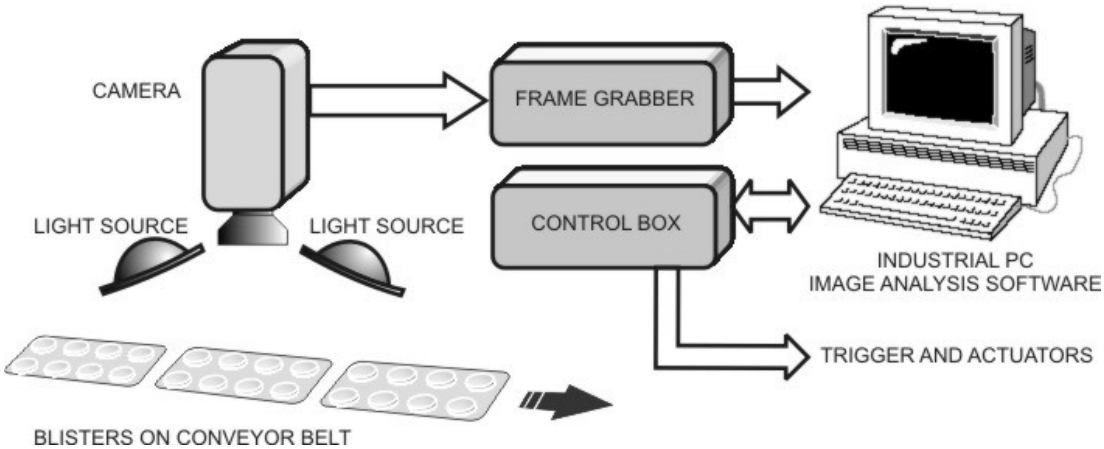


Figure 1.1. A simplified diagram of a typical machine vision system

Roughly, two different approaches, namely template matching and design-rule verification can be taken to perform automated visual inspection. By

template matching every pixel in the inspected image is compared with the pixels of the reference image. The design-rule verification approach checks for the violation of a set of generic rules [5]. All these methods have some advantages and disadvantages, i.e. the template matching approach requires very precise alignment of the reference and inspected images so which is not trivial if moving objects are inspected. The design-rule verification approach does not need a precise alignment but it requires accurate, robust, and fast segmentation which determines the quality of the final inspection [5].

The methods for object inspection make use of the image of objects captured by cameras and for which colored images are frequently preferred since they contain more information than gray level images. Generally images are captured in RGB mode by the electronic devices, each pixel being represented as a point in the RGB color space as shown in Figure 1.2 Experimentally defined features XYZ and I1 I2 I3 are derived from RGB through linear transformation. Human color perception is mathematically modeled by HSV, Lab or Luv features, and each is determined by a nonlinear transformation of the RGB color system [2].

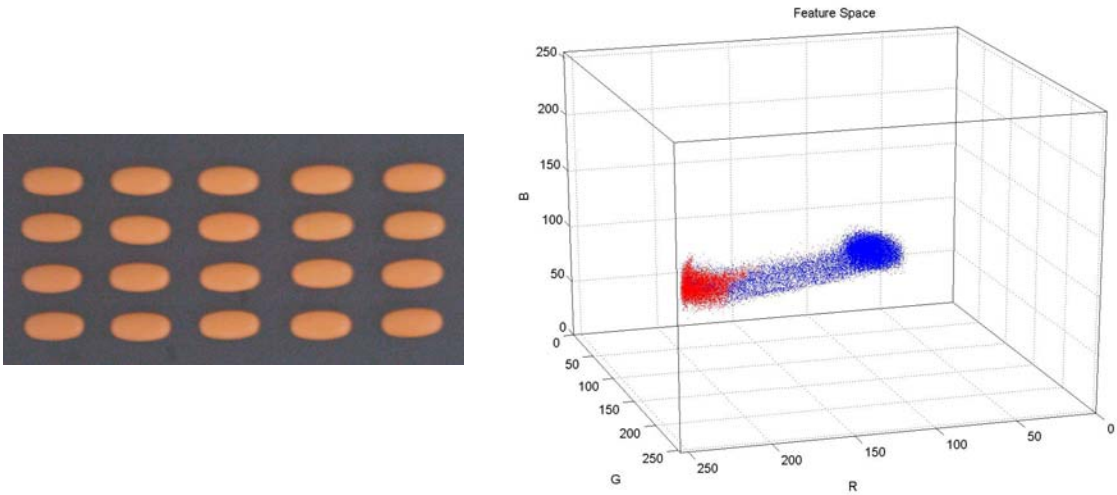


Figure 1.2. A defect free sample image and the corresponding RGB color space

Segmentation methods can be categorized into the following classes; edge detection, region based, methods based on physical reflectance models and statistical methods in some color feature space [2, 3].

Edge detection alone can not segment an image [20]. For the region based methods, a recursive algorithm is required [19]. It is possible to improve the segmentation by using human perception based models [11].

Statistical approaches are based on thresholding of color feature histograms or on clustering in some color feature space [14]. The statistical segmentation approach is very effective even the clusters in the feature space are not compact. The spatial color nonuniformity causes these clusters to be overlapped. Since the RGB space includes both the color value and the intensity value unseperated, RGB feature space is inherently more complex compared with the other color spaces that provides color and intensity values separated. A nonparametric clustering-based segmentation approach is required for the complex cluster shapes, like the nearest neighbor method [19], multivariate histogram or kernel estimation [13].

1.2 SCOPE OF THE THESIS

In this work, a machine vision system software is presented and implemented for inspecting color tablets in pharmaceutical blisters. Two nonparametric clustering based algorithms are used and compared for the inspection; Nearest Neighbor method and Kernel estimation. Nearest Neighbor method is utilized in both RGB and HSV color spaces, while Kernel estimation method is utilized only in RGB color space. Kernel estimation is based on a novel nonparametric clustering-based segmentation method that has been derived in [5] from a recently proposed mean shift procedure [3] but is faster and thus more appropriate for real-time applications. The segmentation allows the system to detect missing and broken tablets, tablet fragments, and

the color, size, and shape of individual tablets in pharmaceutical blisters automatically in real-time. Because spatial color nonuniformity may influence the accuracy and robustness of automated inspection, it is reduced prior to segmentation. In this correction step a multivariable optimization is required. Downhill Simplex method [12] is utilized as well as Powell's method, which is used in [10], and the performances of the methods are compared with each other.

1.3 ORGANIZATION

In Chapter 2 the components of the inspection system are described and the flowchart of the software algorithm, which consists of two main operation modes; the training mode and the inspection mode, is defined.

The training mode, consisting of the three main steps: spatial color nonuniformity correction, segmentation and feature extraction is discussed in detail, in Chapter 3.

Chapter 4 covers the details of the inspection mode, which consists of three main steps: spatial color nonuniformity correction, segmentation and defective tablet detection.

In Chapter 5, experiments done are discussed; the results of all steps of some experiments are presented.

Method's features are outlined and the obtained results are discussed in the conclusions part, in Chapter 6.

CHAPTER 2

VISUAL INSPECTION SYSTEM

The machine vision system is composed of a digital USB camera, illumination unit, and desktop PC (a frame grabber and control box is not used in this work). The scene is illuminated by diffuse front lighting obtained by diffusive glass in front of white LED sources. The software was implemented in MATLAB 7.0.

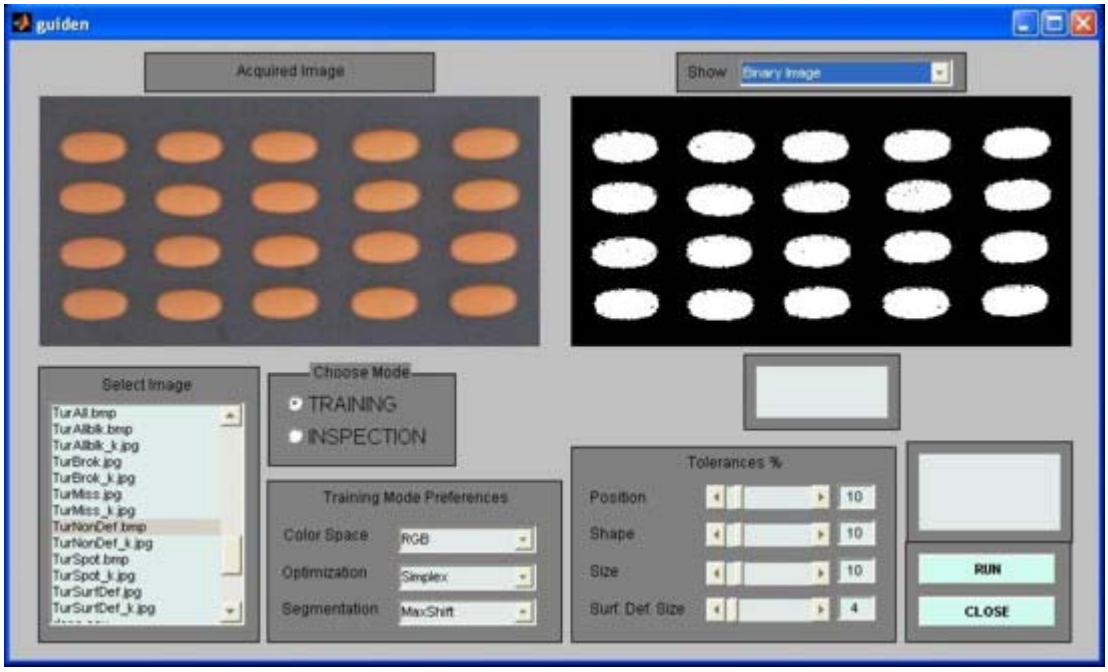


Figure 2.1. GUI of the implemented software

Software operates either in the ‘‘training’’ or ‘‘inspection’’ mode. Prior to the training mode, optimization algorithm for the spatial color nonuniformity correction is selected as Powell or Downhill Simplex method; color space is selected as RGB or HSV for the segmentation step; segmentation method is selected as MaxShift (Kernel estimation) or Nearest Neighbor algorithm.

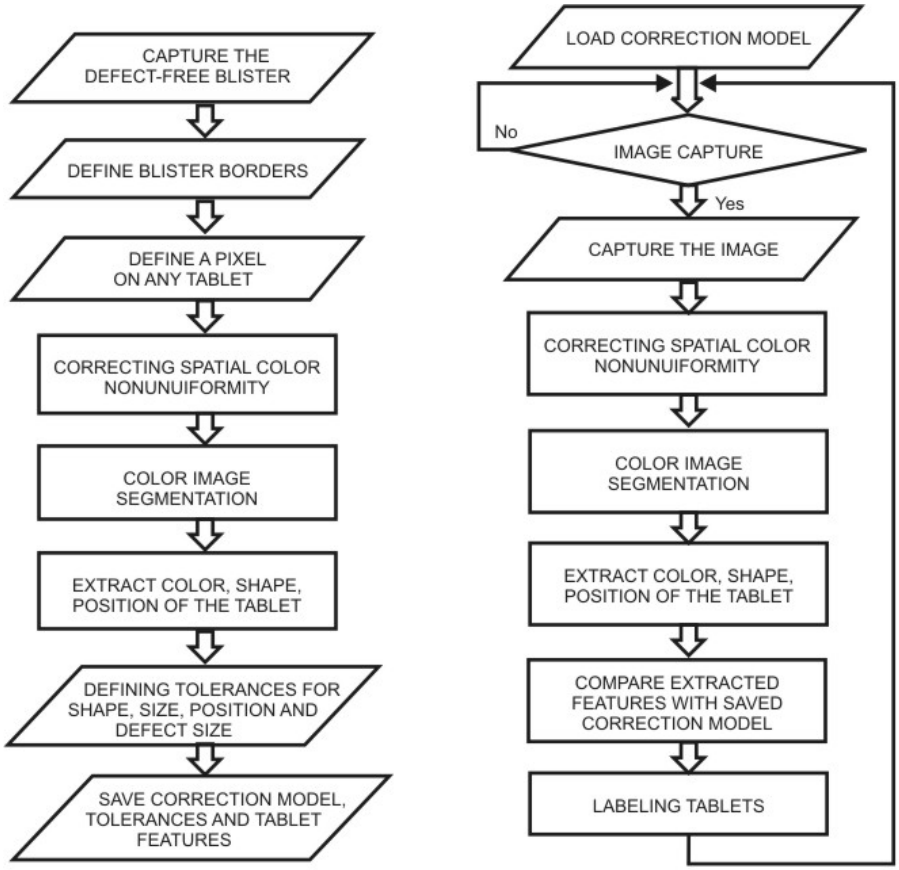


Figure 2.2. Training mode operation (left), inspection mode operation (right).

In the training mode firstly a defect free image is captured (Figure 2.2.). After capturing the image, the operator;

a. defines the blister borders. Training tablets are assigned to the corresponding blisters. This information is used in inspection mode to reject blisters containing defective tablets.
b. selects a point on any tablet, which is required for nonparametric clustering based color image segmentation. If the segmentation method is Nearest Neighbor algorithm, he also selects a point on background, too.
c. sets tolerances for:
    - Position of each tablet
    - Size of each tablet
    - Shape of each tablet
    - Size of surface defects

The acquired image is then corrected for spatial color nonuniformity, segmented and its extracted features are stored in the resulting blister model, which is composed of:

a. Spatial color nonuniformity correction function
b. Positions of blisters
c. Nonparametric color image segmentation model of tablets
d. Feature values describing:
    - Position of each tablet
    - Size of each tablet
    - Shape of each tablet
e. User defined tolerances of feature values describing:
    - Position of each tablet
    - Size of each tablet
    - Shape of each tablet
    - Size of surface defects

After using the training mode just once, the tablets are inspected in the inspection mode. In the inspection mode, spatial color nonuniformity of every acquired image is corrected by the correction model which is generated in the training mode and stored in the blister model. The corrected image is

than segmented according to the segmentation model which is also saved in the blister model, in the training mode. The features of the segmented image is extracted in the same way as the training mode and the obtained results are than compared with the feature values and the user-defined tolerances to detect the defective tablets. The defective tablets detected are marked and the blisters containing any defective tablet are determined.

CHAPTER 3

TRAINING MODE

In the training mode, blister model is extracted by using an image of a defect-free blister scene.

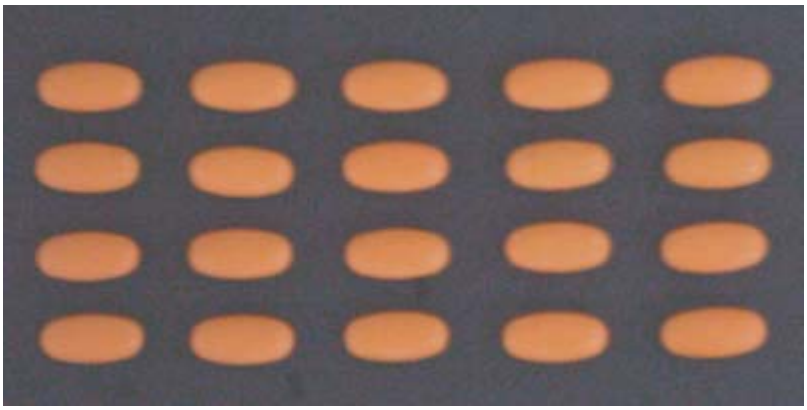


Figure 3.1. A defect-free training image

Training mode is performed when the packing machine is not operating, and there is no time constraint for blister model extraction which is a time consuming process.

3.1 SPATIAL NONUNIFORMITY CORRECTION

Image color intensity inhomogeneity is an adverse phenomenon which manifests itself by slow color intensity variations over the image domain. Since perfect uniform illumination is practically hard to achieve, smooth spatial color variations may be present in the image. Spurious intensity variations, which may reach up to 30% of the image color intensity amplitude, usually do not effect the visual impression of the image significantly, but may have serious implications for image analysis, e.g. in segmentation, registration, quantification. Consequently, the correction of the adverse spatial color uniformities is a necessary preprocessing step in many image analysis tasks.

Methods for correction of intensity inhomogeneities may be prospective or retrospective. The prospective methods require acquisition protocol tuned to inhomogeneity correction, while the retrospective methods can be applied to any image, since they only use the information naturally occurring in an image.

Likar et al. [10] described a retrospective correction method based on entropy minimization, designed for correcting the intensity inhomogeneity. The intensity nonuniformity correction method they proposed for gray scale images is extended to color images in [5]. The derivation of the algorithm, which is used in this thesis, is based on the assumption that an image corrupted by color intensity inhomogeneity contains more information than the corresponding uncorrupted image. Color nonuniformity correction requires:

- no pre-processing
- no special tuning
- no parameter setting
- no pre-specification of the number of expected colors in a scene
- no user interaction

This method is general, flexible and proved to be fast, efficient and fully automatic. Images of multicolor objects, such as capsules, are corrected with this method without any modification.

3.1.1THEORY

3.1.1.1 FORMULATION

Let $\mathbf{v}(\mathbf{x})=[v_1(\mathbf{x}),v_2(\mathbf{x}),v_3(\mathbf{x})]$ be a color image, where $\mathbf{x}$ is a vector in spatial domain X and $[v_1(\mathbf{x}),v_2(\mathbf{x}),v_3(\mathbf{x})]$ is a three-dimensional vector providing the respective color intensities. Suppose an image is captured and digitized with n_v levels for each color component. For $\mathbf{v}(\mathbf{x})$ denote the acquired image and $\mathbf{u}(\mathbf{x})=[u_1(\mathbf{x}),u_2(\mathbf{x}),u_3(\mathbf{x})]$ denote the "true" image of the imaged object, the two images are related by:

$$v(\mathbf{x}) = f(u(\mathbf{x})) \quad \text{or} \quad u(\mathbf{x}) = f^{-1}(v(\mathbf{x})) \tag{1}$$

with f denoting the image degradation model (bias field) that introduces a spatially dependent color intensity degradation to the true image component $u(\mathbf{x})$ and f^{-1} representing the inverse of the degradation model. The problem of retrospective correction of intensity inhomogeneity is to find the true image $\mathbf{u}(\mathbf{x})$ from the acquired image $\mathbf{v}(\mathbf{x})$. A model-based solution involves two steps:

a. A parametric correction model f^{-1}, the estimate of the inverse of the degradation model, has to be selected. This can be done,
   - theoretically; by mathematically describing the process of image degradation
   - experimentally; by analyzing the acquired images of well-defined calibration objects.
b. The parameters defining the optimal correction model f_o^{-1} have to be found, so that the image $\mathbf{u_o}(\mathbf{x})=[u_{o1}(\mathbf{x}),u_{o2}(\mathbf{x}),u_{o3}(\mathbf{x})]$ with minimal color

intensity nonuniformity is obtained by the application of f_{o1}^{-1}, f_{o2}^{-1} and f_{o3}^{-1} to the acquired image components $v_1(\mathbf{x})$,$v_2(\mathbf{x})$ and $v_3(\mathbf{x})$, respectively. This step requires a quantitative color intensity uniformity criterion and a strategy for finding the corrected image $\mathbf{u_o}(\mathbf{x})$.

3.1.1.2 STRATEGY

The proposed correction strategy is based on the assumption that because of the image degradation process, the information content of each color component $v(\mathbf{x})$ of the acquired nonuniform image $\mathbf{v}(\mathbf{x})$ will be higher than the information content of the corresponding component $u(\mathbf{x})$ of the ideal color uniform image $\mathbf{u}(\mathbf{x})$ [10].

$$I\,[v(\mathbf{x})] = I\,[\,f(u(\mathbf{x}))] > I\,[u(\mathbf{x})] \qquad (2)$$

The image degradation process is, thus, treated as an additional source of information that increases the information of the ideal color uniform image $\mathbf{u}(\mathbf{x})$.

The above assumption should hold when spurious color intensity variations induced in the degradation process are spatially different, i.e., smoother than the natural intensity variations in the true image. This is certainly true for tablet images and enables retrospective correction of spatial color nonuniformity by using the information of the acquired image and an appropriate correction model f^{1}. The correction model should describe only the color intensity variations due to degradation process and not the color intensity variations of the true image.

Besides, the correction model f^{1} must be constrained so that:

- It can not change the global intensity characteristics of the input image, such as brightness or contrast.

- It can only introduce smooth, spatially dependent intensity variations that are needed for correction of color nonuniformity.

The information I of any image component $v(\mathbf{x})$ is expressed by the Shannon entropy H as:

$$I[v(\mathbf{x})] = H[v(\mathbf{x})] = -\sum_{n} p(n) log p(n) \tag{3}$$

where $p(n)$ is the probability that a point in image component $v(\mathbf{x})$ has value n. The magnitude of the entropy H is a measure of the uncertainty described by a set of probabilities. The entropy is nonnegative and takes on its maximum value when all gray levels are equally likely. By changing the parameters of the correction model f^{1} and transforming the acquired image component $v(\mathbf{x})$ (1) in such a way that the information I of the transformed image is minimized, i.e., that the correction model best reduces additional information induced in the degradation process, leads to the optimal correction model f_o^{-1}

$$f_o^{-1} = \arg\min_{f^{-1}} \{ I[f^{-1}(v(\mathbf{x}))] \} \tag{4}$$

which defines the transformation of the acquired image component $v(\mathbf{x})$ to the corrected image component $u_o(\mathbf{x})$

$$u_o(\mathbf{x}) = f^{1}{}_{o}\,(v(\mathbf{x})) \tag{5}$$

The flowchart in Figure 3.2., illustrates the image degradation process and the proposed correction strategy for ith image component. The image degradation process increases the information of the actual color uniform image component $u_{act}(\mathbf{x})$ due to the color intensity degradation induced by the corresponding actual f_{act}.The information of the acquired image component $v(\mathbf{x})$ is then minimized by the correction model f^{1}, which leads to the corrected image component $u_o(\mathbf{x})$.

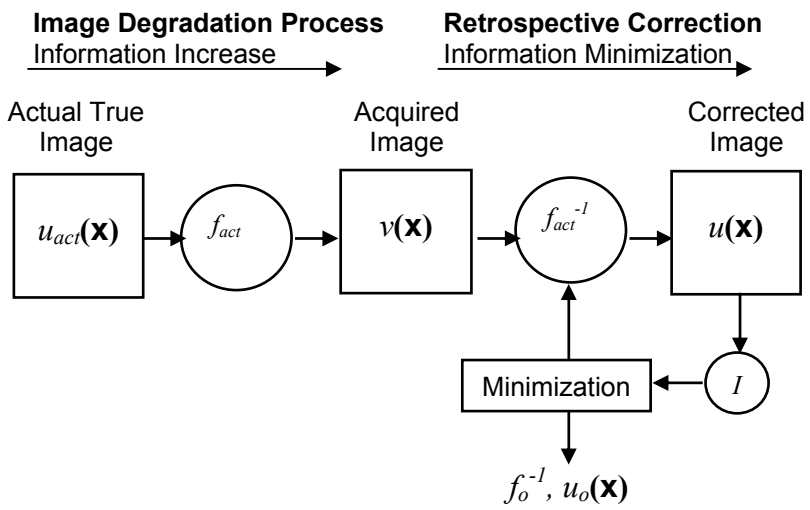


Figure 3.2. Image degradation and retrospective correction of spatial color nonuniformity based on information minimization

The image $\mathbf{v}(\mathbf{x})$ is separated into components $v_1(\mathbf{x})$, $v_2(\mathbf{x})$ and $v_3(\mathbf{x})$, representing the Red, Green and Blue components, respectively. The components are independently corrected for spatial color nonuniformity and the obtained corrected components $u_{o1}(\mathbf{x})$,$u_{o2}(\mathbf{x})$ and $u_{o3}(\mathbf{x})$ are merged into a corrected color image $\mathbf{u_o}(\mathbf{x})$. The flowchart in Figure 3.3., illustrates the entire procedure, which is applicable for a fixed illumination and a fixed geometry situation. This requirement is achieved in a well-controlled industrial environment which is the case for the automated visual inspection system to be installed on a blister packing machine.

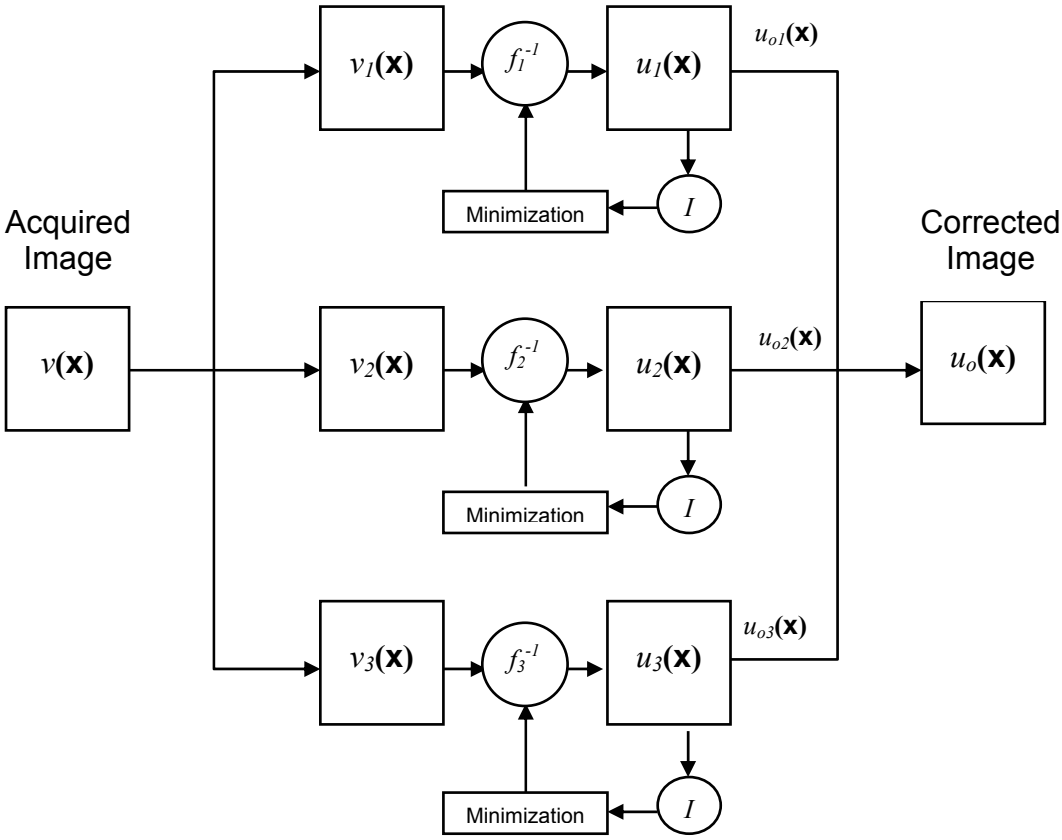


Figure 3.3. Retrospective correction of spatial color nonuniformity based on information minimization

3.1.2LINEAR MODELING

The proposed retrospective correction strategy is implemented by a linear image degradation model f. In this section, the parametric correction model f^{-1} is defined in the modeling step, while the procedures for finding the optimal correction model f_o^{-1} and the corrected image $\mathbf{u_o}(\mathbf{x})$ are explained in the correction step. Implementation details are described in the last step.

3.1.2.1 MODELING

Since it is a relatively general model that can characterize the image degradation processes in many visual inspection images, image degradation is modeled as linear which consists of a multiplicative $m(x)$ and an additive $a(x)$ color intensity degradation component [10].

$$v(\mathbf{x}) = u(\mathbf{x})m(\mathbf{x}) + a(\mathbf{x}) \tag{6}$$

The estimation of the true image is obtained by the inverse of the degradation model as:

$$u(\mathbf{x}) = v(\mathbf{x})m^{-1}(\mathbf{x}) + a^{-1}(\mathbf{x}) \tag{7}$$

where

$$m^{-1}(\mathbf{x}) \stackrel{\Delta}{=} \frac{1}{m(\mathbf{x})} \qquad \text{and}$$

$$a^{-1}(\mathbf{x}) \stackrel{\Delta}{=} -\frac{a(\mathbf{x})}{m(\mathbf{x})} \tag{8}$$

$m^{-1}(\mathbf{x})$: multiplicative component
$a^{-1}(\mathbf{x})$: additive component

These two components are described by a linear combination of K smoothly varying basis functions $s_i(\mathbf{x})$ that are uniquely defined by parameters ${}^{m}b$ and ${}^{a}b$, respectively, as:

$$m^{-1}(\mathbf{x}) = \sum_{i=1}^{K} {}^{m}b_i \, {}^{m}s_i(\mathbf{x}) \qquad \text{and}$$

$$a^{-1}(\mathbf{x}) = \sum_{i=1}^{K} {}^{a}b_i \, {}^{a}s_i(\mathbf{x}) \tag{9}$$

Considering that the correction model f^{1} can not change the global intensity characteristics of the input image and it can only introduce smooth, spatially dependent color intensity variations, three conditions are defined:

a) Mean Preserving Condition:

In order to ensure that the mean intensity values of $v(\mathbf{x})$ and $u(\mathbf{x})$ to be the same, the mean preserving condition is introduced in [10]:

$$\frac{1}{\Theta}\int_{\Omega} v(\mathbf{x})d\Omega = \frac{1}{\Theta}\int_{\Omega} u(\mathbf{x})d\Omega \tag{10}$$

$$\frac{1}{\Theta}\int_{\Omega} v(\mathbf{x})d\Omega = \frac{1}{\Theta}\int_{\Omega} (v(\mathbf{x})m^{-1}(\mathbf{x}) + a^{-1}(\mathbf{x}))d\Omega \tag{11}$$

defined over the domain Ω which comprises the relevant part of the image, of size Θ:

$$\Theta = \int_{\Omega} d\Omega \tag{12}$$

The global transformation effect of the correction components $m^{-1}(\mathbf{x})$ and $a^{-1}(\mathbf{x})$ with respect to the given image is neutralized by this condition.

b) Normalization of the Parameters:

For the equal change of any parameter ${}^{m}b_i$ or ${}^{a}b_i$ of the basis function $s_i(\mathbf{x})$ to produce intensity transformation of the same order, the parameters are normalized:

$$u(\mathbf{x}) = v(\mathbf{x})\sum_{i=1}^{K} {}^{m}b_i \, {}^{m}s_i(\mathbf{x}) + \sum_{i=1}^{K} {}^{a}b_i \, {}^{a}s_i(\mathbf{x}) \tag{13}$$

$$u(\mathbf{x}) = \sum_{i=1}^{K} {}^{m}b_i (v(\mathbf{x}) {}^{m}s_i(\mathbf{x})) + \sum_{i=1}^{K} {}^{a}b_i \, {}^{a}s_i(\mathbf{x}) \tag{14}$$

$$\frac{1}{\Theta}\int_{\Omega} d\Omega = 1 \tag{15}$$

$$\frac{1}{\Theta}\int_{\Omega} \left| v(\mathbf{x}) {}^{m}s_i(\mathbf{x}) \right| d\Omega = 1 \quad \text{and}$$

$$\frac{1}{\Theta}\int_{\Omega} \left| {}^{a}s_i(\mathbf{x}) \right| d\Omega = 1 \qquad \text{for all } i \tag{16}$$

c) Smooth Variation Condition:

Polynomial terms $q_i(\mathbf{x})$ are used as the smoothly varying basis function $s_i(\mathbf{x})$ to achieve smooth color intensity variations introduced by the correction model f^1[10]. For example, for a first order polynomial K is 3 and the polynomial terms $q_1(\mathbf{x})$ to $q_3(\mathbf{x})$ are the elements of the matrix [1 x y]; while for a second order polynomial, K is 6 and the polynomial terms $q_1(\mathbf{x})$ to $q_6(\mathbf{x})$ are the elements of the matrix [1 x y x^2 xy y]. In the same manner for a fourth order polynomial, $q_1(\mathbf{x})$ to $q_{15}(\mathbf{x})$ are the elements of the matrix [1 x y x^2 xy y^2 x^3 x^2y xy^2 y^3 x^4 x^3y x^2y^2 xy^3 y^4]. The basis function is described as:

$${}^{m}s_i(\mathbf{x}) = \frac{q_i(\mathbf{x}) - {}^{m}c_i}{{}^{m}d_i} \qquad \text{and}$$

$${}^{a}s_i(\mathbf{x}) = \frac{q_i(\mathbf{x}) - {}^{a}c_i}{{}^{a}d_i} \tag{17}$$

c_i: neutralization constant, needed to fulfill the mean preserving condition (11)

d_i: normalization constant, needed to normalize the parameters (16)

- **Derivation of ${}^{a}c_i$ and ${}^{m}c_i$:**

Rewriting the mean preserving condition (11) [10]:

$$\frac{1}{\Theta}\int_{\Omega} v(\mathbf{x})d\Omega = \frac{1}{\Theta}\int_{\Omega} v(\mathbf{x})m^{-1}(\mathbf{x})d\Omega + \frac{1}{\Theta}\int_{\Omega} a^{-1}(\mathbf{x})d\Omega \tag{18}$$

the mean value of the additive component is set to zero

$$\int_{\Omega} a^{-1}(\mathbf{x})d\Omega = 0 \tag{19}$$

so

$$\int_{\Omega} v(\mathbf{x})d\Omega = \int_{\Omega} v(\mathbf{x})m^{-1}(\mathbf{x})d\Omega \tag{20}$$

for the additive component, using (9) and (19) leads to the condition

$$\int_{\Omega}\left(\sum_{i=1}^{K} {}^{a}b_i^{a}s_i(\mathbf{x})\right)d\Omega = 0 \tag{21}$$

$$\int_{\Omega} {}^{a}b_1\,{}^{a}s_1(\mathbf{x})d\Omega + \int_{\Omega}\sum_{i=2}^{K} {}^{a}b_i\,{}^{a}s_i(\mathbf{x})d\Omega = 0 \tag{22}$$

the basis function ${}^{a}s_1(\mathbf{x})$ is the uniform polynomial term and is set to a neutral value, which is 0 for the additive term:

$${}^{a}s_1(\mathbf{x}) = 0 \tag{23}$$

by substituting (23) and (17) in (22):

$$\int_{\Omega}\left(\sum_{i=2}^{K} {}^{a}b_i \frac{q_i(\mathbf{x})-{}^{a}c_i}{{}^{a}d_i}\right)d\Omega = 0 \tag{24}$$

rewriting (24) yields

$$\sum_{i=2}^{K} \frac{{}^{a}b_i}{{}^{a}d_i}\left(\int_{\Omega}(q_i(\mathbf{x})-{}^{a}c_i)d\Omega\right) = 0 \tag{25}$$

which has a nontrivial solution (${}^{a}b_i \neq 0$)

$$\int_{\Omega}(q_i(\mathbf{x})-{}^{a}c_i)d\Omega = 0 \tag{26}$$

$$\int_{\Omega} {}^{a}c_i d\Omega = \int_{\Omega} q_i(\mathbf{x})d\Omega \tag{27}$$

$${}^{a}c_i \int_{\Omega} d\Omega = \int_{\Omega} q_i(\mathbf{x})d\Omega \tag{28}$$

so ${}^{a}c_i$ is defined as

$${}^{a}c_i = \frac{1}{\Theta}\int_{\Omega} q_i(\mathbf{x})d\Omega \tag{29}$$

In the same manner, for the multiplicative component, using (9) and (20) leads to the condition

$$\int_{\Omega} v(\mathbf{x})d\Omega = \int_{\Omega} v(\mathbf{x})\left(\sum_{i=1}^{K} {}^{m}b_i^{m} s_i(\mathbf{x})\right)d\Omega \tag{30}$$

$$\int_{\Omega} v(\mathbf{x})d\Omega = \int_{\Omega} v(\mathbf{x})\left({}^{m}b_{1}\,{}^{m}s_{1}(\mathbf{x}) + \sum_{i=2}^{K} {}^{m}b_{i}\,{}^{m}s_{i}(\mathbf{x})\right)d\Omega \tag{31}$$

the basis function ${}^{m}s_{1}(\mathbf{x})$ is the uniform polynomial term and is set to a neutral value, which is 1 for the multiplicative term:

$${}^{m}s_{1}(\mathbf{x}) = 1 \tag{32}$$

Substituting (32) in (31):

$$\int_{\Omega} v(\mathbf{x})d\Omega = \int_{\Omega} v(\mathbf{x})\left(1 + \sum_{i=2}^{K} {}^{m}b_{i}\,{}^{m}s_{i}(\mathbf{x})\right)d\Omega \tag{33}$$

$$\int_{\Omega} \sum_{i=2}^{K} {}^{m}b_{i}\,{}^{m}s_{i}(\mathbf{x})d\Omega = 0 \tag{34}$$

by substituting (17) in (34)

$$\sum_{i=2}^{K} {}^{m}b_{i} \int_{\Omega} v(\mathbf{x})\left(\frac{q_{i}(\mathbf{x}) - {}^{m}c_{i}}{{}^{m}d_{i}}\right)d\Omega = 0 \tag{35}$$

$$\sum_{i=2}^{K} \frac{{}^{m}b_{i}}{{}^{m}d_{i}}\left(\int_{\Omega} v(\mathbf{x})(q_{i}(\mathbf{x}) - {}^{m}c_{i})d\Omega\right) = 0 \tag{36}$$

the nontrivial solution (${}^{m}b_{i} \neq 0$) yields

$$\int_{\Omega} v(\mathbf{x})(q_{i}(\mathbf{x}) - {}^{m}c_{i})d\Omega = 0 \tag{37}$$

$$\int_{\Omega} v(\mathbf{x})\,{}^{m}c_{i}d\Omega = \int_{\Omega} v(\mathbf{x})q_{i}(\mathbf{x})d\Omega \tag{38}$$

so ${}^{m}c_i$ is defined as

$$ {}^{m}c_i = \frac{\int_{\Omega} v(\mathbf{x}) q_i(\mathbf{x}) d\Omega}{\int_{\Omega} v(\mathbf{x}) d\Omega} \tag{39} $$

- **Derivation of ${}^{a}d_i$ and ${}^{m}d_i$:**

by rewriting (16) for the multiplicative component [10]:

$$ \frac{1}{\Theta} \int_{\Omega} \left| v(x) \frac{q_i(x) - {}^{m}c_i}{{}^{m}d_i} \right| d\Omega = 1 \tag{40} $$

so ${}^{m}d_i$ is defined as

$$ {}^{m}d_i = \frac{1}{\Theta} \int_{\Omega} \left| v(x)(q_i(x) - {}^{m}c_i) \right| d\Omega \tag{41} $$

by rewriting (16) for the additive component:

$$ \frac{1}{\Theta} \int_{\Omega} \left| \frac{q_i(x) - {}^{a}c_i}{{}^{a}d_i} \right| d\Omega = 1 \tag{42} $$

so ${}^{a}d_i$ is defined as

$$ {}^{a}d_i = \frac{1}{\Theta} \int_{\Omega} \left| q_i(x) - {}^{a}c_i \right| d\Omega \tag{43} $$

The two correction components $m^{-1}(\mathbf{x})$ and $a^{-1}(\mathbf{x})$ are modeled by the linear combination of neutralized and normalized polynomials:

$$m^{-1}(\mathbf{x}) = 1 + \sum_{i=2}^{K} {}^{m}b_i \frac{q_i(\mathbf{x}) - {}^{m}c_i}{{}^{m}d_i}$$ and

$$a^{-1}(\mathbf{x}) = \sum_{i=2}^{K} {}^{a}b_i \frac{q_i(\mathbf{x}) - {}^{a}c_i}{{}^{a}d_i} \tag{44}$$

3.1.2.2 CORRECTION

Considering the linear image degradation model (6) from the information perspective, the ideal true image component $u(\mathbf{x})$ contains solely the information about the imaged object. After $u(\mathbf{x})$ is first multiplied by the multiplicative color intensity degradation image $m(\mathbf{x})$ and then the additive intensity degradation image $a(\mathbf{x})$ added to it, the information of the two color intensity degradation images, $m(\mathbf{x})$ and $a(\mathbf{x})$, is merged with the information of $u(\mathbf{x})$. Information content of the resulting image component $v(\mathbf{x})$, which is the acquired image, will be higher than the information content of the true image $u(\mathbf{x})$, since the true image $u(\mathbf{x})$ and the two degradation images $m(\mathbf{x})$ and $a(\mathbf{x})$ contain distinct spatial variations.

The information of $u(\mathbf{x})$ is minimized by using the inverse of the image degradation model (7) and changing the parameters of the correction components $m^{-1}(\mathbf{x})$ and $a^{-1}(\mathbf{x})$ defined in (44). The correction steps are:

i. The optimal parameters ${}^{m}b_o$ or ${}^{a}b_o$ are found by Brent's one dimensional optimization algorithm and Powell's multidimensional directional set method [15]:

$$\left\{{}^{m}b_o, {}^{a}b_o\right\} = \underset{\left\{{}^{m}b, {}^{a}b\right\}}{\arg\min}\left\{I\left[u(\mathbf{x})\right]\right\} \tag{45}$$

$$\left\{{}^{m}b_o, {}^{a}b_o\right\} = \underset{\left\{{}^{m}b, {}^{a}b\right\}}{\arg\min}\left\{I\left[v(\mathbf{x})m^{-1}(\mathbf{x}) + a^{-1}(\mathbf{x})\right]\right\} \tag{46}$$

ii. The optimal components $m_o^{-1}(\mathbf{x})$ and $a_o^{-1}(\mathbf{x})$ are defined by the optimal parameters ${}^m b_o$ or ${}^a b_o$, respectively:

$$m_o^{-1}(\mathbf{x}) = 1 + \sum_{i=2}^{K} {}^m b_{oi} \frac{q_i(\mathbf{x}) - {}^m c_i}{{}^m d_i} \qquad \text{and}$$

$$a_o^{-1}(\mathbf{x}) = \sum_{i=2}^{K} {}^a b_{oi} \frac{q_i(\mathbf{x}) - {}^a c_i}{{}^a d_i} \tag{47}$$

iii. The optimal components $m_o^{-1}(\mathbf{x})$ and $a_o^{-1}(\mathbf{x})$ transform the acquired image component $v(\mathbf{x})$ into the optimally corrected image component $u_o(\mathbf{x})$:

$$u_o(\mathbf{x}) = v(\mathbf{x}) m_o^{-1}(\mathbf{x}) + a_o^{-1}(\mathbf{x}) \tag{48}$$

3.1.2.2.1 Minimization or Maximization of Functions

Given a single function f that depends on one or more independent variables, in order to calculate what value of f is achieved at the maximum or minimum, firstly the value of those variables where f takes on a maximum or a minimum value must be found. The tasks of maximization and minimization are trivially related to each other, since one person's function f could just as well be another's $-f$.

Since the computational effort is dominated by the cost of evaluating f (and also perhaps its partial derivatives with respect to all variables, if the chosen algorithm requires them), f must be evaluated as few times as possible to reduce the computational cost.

An extremum (maximum or minimum point) can be either global (truly the highest or lowest function value) or local (the highest or lowest in a finite neighborhood and not on the boundary of that neighborhood).

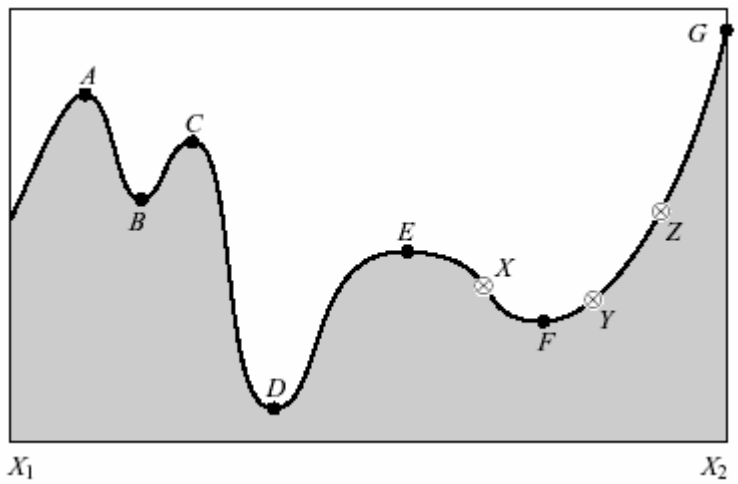


Figure 3.4. Extrema of a function in an interval. *A*, *C*, and *E* are local maximas; *B* and *F* are local minimas. The global maximum is at *G;* the global minimum is at *D*. *X*, *Y* and *Z* are said to "bracket" the minimum *F*, since *Y* is less than both *X* and *Z* [15].

Finding a global extremum is, in general, a very difficult problem. Two standard algorithms are widely used:

(i) find local extrema starting from widely varying starting values of the independent variables (perhaps chosen quasi-randomly), and then pick the most extreme of these (if they are not all the same)

(ii) perturb a local extremum by taking a finite amplitude step away from it, and then see if your routine returns you to a better point, or "always" to the same one

The usual name of this large field of numerical research is optimization. A selection of the best established algorithms in unconstrained minimization is submitted in [15]. Unfortunately, there is no perfect optimization algorithm; more than one method must be tried in comparative fashion. Initial choice must be made between methods that need only evaluations of the function to be minimized and methods that also require evaluations of the derivative of that function. Algorithms using the derivative are somewhat more powerful

than those using only the function, but not always enough so as to compensate for the additional calculations of derivatives.

For the one-dimensional case:

• Without calculation of the derivative, firstly minimum is bracketed and then Brent's method is used. If the function has a discontinuous second or lower derivative, then the parabolic interpolations of Brent's method are of no advantage, and the simplest form of golden section search, can be used.

• With calculation of the derivative, a variant of Brent's method which makes limited use of the first derivative information can be used.

For the multidimensional case choice must be made between methods that require storage of order N^2 and those that require only of order N, where N is the number of dimensions, considering the storage memory size.

The methods without calculation of the gradient are:

• Downhill simplex method just crawls downhill in a straightforward fashion that makes almost no special assumptions about the function. This can be extremely slow, but it can also, in some cases, be extremely robust. This method is mostly useful when the minimization calculation is only an incidental part of your overall problem. The storage requirement is of order N^2.

• For the direction-set methods, e.g. Powell's method, although derivatives are not needed, the method does require a one-dimensional minimization sub-algorithm such as Brent's method. The storage requirement is of order N^2.

There are two major families of algorithms with calculation of first derivatives. Both families require a one-dimensional minimization sub-algorithm, which can either use or not use the derivative information, depending on your choice. Either family does not dominate the other in all applications:

• Conjugate gradient methods are typified by the Fletcher-Reeves algorithm and the closely related and probably superior Polak-Ribiere algorithm. These methods require only of order a few times *N* storage, and one-dimensional sub-minimization.

• Quasi-Newton or variable metric methods are typified by the Davidon-Fletcher-Powell (DFP) algorithm (sometimes referred to just as Fletcher-Powell) or the closely related Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm. These methods require of order N^2 storage and one dimensional sub-minimization.

A multidimensional optimization method is required in order to minimize the information function I that depends on the parameters ${}^{m}b$ and ${}^{a}b$. The minimization calculation is one of the most essential parts of the inspection problem that affects the segmentation part, and the storage is not important since this algorithm will be processed only once in the training mode. Also, the algorithms using the derivative are not enough powerful to compensate for the additional calculations of derivatives. Considering these facts, Downhill Simplex method and Powell's direction-set method are decided to be used among the methods for optimization of a function and they are compared. As mentioned, Pwell's method, which needs storage of order N^2, does not require the calculation of derivatives, but requires a one-dimensional minimization sub-algorithm. Since the information function depending on the variables ${}^{m}b$ and ${}^{a}b$ is sufficiently smooth – nicely parabolic near the minimum-, has one minima and wide capturing ranges; the advantage of the parabolic interpolations of Brent's method as the one-dimensional minimization sub-algorithm can be taken. Brent's method is used without calculation of the derivative after bracketing the minimum of the function.

3.1.2.2.1.1 Bracketing a Minimum

A minimum is known to be bracketed only when there is a *triplet* of points, $a < b < c$ (or $c < b < a$), such that $f(b)$ is less than both $f(a)$ and $f(c)$. In this case, if the function is nonsingular, it has a minimum in the interval (a, c).

Initial bracketing is an essential part of any one-dimensional minimization. Although, there are some one-dimensional algorithms that do not require initial bracketing, it is better to bracket the minima before isolating it by knowing that the minimum is certainly bracketed. The range is subdivided to iteratively isolate the minimum.

A very simple iterative minimum bracketing procedure is as follows:
After starting from any initial point, one gets a new point in each iteration by moving a step in the downhill direction. From the third iteration on, the last three points are stored. New one in each iteration substitutes the oldest one in the triplet. Larger and larger steps are taken until the downhill trend stops and one gets the last point, where the function value is greater than in the previous one. The new step is computed by simply multiplying the previous one by a constant factor or obtained as a result of a parabolic interpolation applied on a previous triplet of points.

3.1.2.2.1.2 Brent's One Dimensional Optimization Algorithm

After bracketing the minimum in the first place, it remains to decide on a strategy for choosing the new point x, given (a, b, c). A golden section search is designed to handle, in effect, the worst possible case of function minimization. For the golden section search, given, at each stage, a bracketing triplet of points (a, b, c); the next point to be tried is that which is a fraction 0.38197 into the larger of the two intervals measuring from the central point of the triplet. In other words, the optimal bracketing interval (a, b, c) has its middle point b a fractional distance 0.38197 from one end (say, a),

and 0.61803 from the other end (say, b). These fractions are those of the so-called golden mean or golden section, mentioned by Pythagoreans for its aesthetic qualities. The golden section search guarantees that each new function evaluation will bracket the minimum to an interval just 0.61803 times the size of the preceding interval after self-replicating ratios have been achieved. This is comparable to, but not quite as good as, the 0.50000 that holds when finding roots by bisection.

The golden section search method can be easily used with any continuous function since it is designed to handle, in effect, the worst possible case of function minimization. However, its convergence is only linear, meaning that successive significant figures are won linearly with additional function evaluations. If the function has at least a continuous second derivative, and is nicely parabolic near to the minimum which is the case for the sufficiently smooth information function I, then the parabola fitted through any three points ensures finding the minimum quicker since its convergence is superlinear, where the rate at which successive significant figures are liberated increases with each successive function evaluation.

Assuming the function is parabolic, the abscissa of the minimum of the function is found by fitting the parabola to the three bracket points. The procedure is technically called inverse parabolic interpolation, since an abscissa is found rather than an ordinate.

The formula for the abscissa x that is the minimum of a parabola through three points $f(a)$, $f(b)$, and $f(c)$ is:

$$x = b - \frac{1}{2}\frac{(b-a)^2\big(f(b)-f(c)\big)-(b-c)^2\big(f(b)-f(a)\big)}{(b-a)\big(f(b)-f(c)\big)-(b-c)\big(f(b)-f(a)\big)} \tag{49}$$

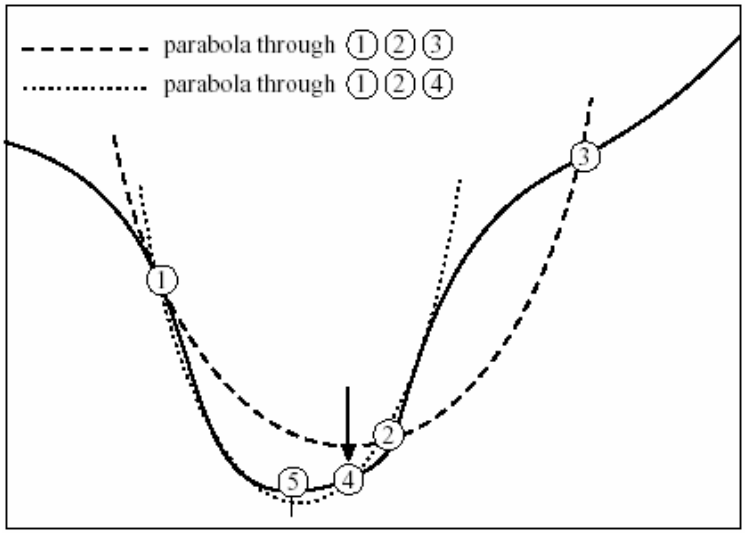


Figure 3.5. Convergence to a minimum by inverse parabolic interpolation. A parabola (dashed line) is drawn through the three original points 1, 2, 3 on the given function (solid line). The function is evaluated at the parabola's minimum, 4, which replaces point 3. A new parabola (dotted line) is drawn through points 1, 4, 2. The minimum of this parabola is at 5, which is close to the minimum of the function [15].

For the derivation of this formula, the following approach is chosen:

$$f(x) = f(a)\frac{(x-b)(x-c)}{(a-b)(a-c)} + f(b)\frac{(x-c)(x-a)}{(b-c)(b-a)} + f(c)\frac{(x-a)(x-b)}{(c-a)(c-b)} \tag{50}$$

The idea is that the right hand side is of grade 2 in x, and is identical to the left hand side, i.e. $f(x)$ for the three x-values a, b and c. Since a parabola with given orientation is uniquely defined by three points, this is the parabola wanted. As searching for a minimum of the parabola, the derivative is computed and set to zero:

$$f'(x) = f(a)\frac{(x-b)+(x-c)}{(a-b)(a-c)} + f(b)\frac{(x-c)+(x-a)}{(b-c)(b-a)} + f(c)\frac{(x-a)+(x-b)}{(c-a)(c-b)} \tag{51}$$

Multiplying with the denominator yields:

$$0 = f(a)(c-b)(2x-b-c) + f(b)(a-c)(2x-c-a) + f(c)(b-a)(2x-a-b) \qquad (52)$$

Hence:

$$0 = 2x\left[f(a)(c-b) + f(b)(a-c) + f(c)(b-a)\right] f(a)(c^2-b^2) + f(b)(a^2-c^2) + f(c)(b^2-a^2) \qquad (53)$$

and

$$x = \frac{1}{2}\frac{f(a)(c^2-b^2) + f(b)(a^2-c^2) + f(c)(b^2-a^2)}{f(a)(c-b) + f(b)(a-c) + f(c)(b-a)} \qquad (54)$$

$$x = b + \frac{1}{2}\frac{f(a)(c-b)(c+b-2b) + f(b)(a-c)(a+c-2b) + f(c)(b-a)(b+a-2b)}{f(a)(c-b) + f(b)(a-c) + f(c)(b-a)} \qquad (55)$$

$$x = b + \frac{1}{2}\frac{f(a)(c-b)^2 + f(b)(a-c)(a+c-2b) + f(c)(b-a)^2}{f(a)(c-b) + f(b)(a-c) + f(c)(b-a)} \qquad (56)$$

$$x = b + \frac{1}{2}\frac{(f(a)-f(b))(c-b)^2 + (f(b)-f(c))(b-a)^2}{(f(a)-f(b))(c-b) + (f(b)-f(c))(b-a)} \qquad (57)$$

This formula fails only if the three points are collinear, in which case the denominator is zero. However, it must be noted that, (49) is as likely to jump to a parabolic maximum as to a minimum. No minimization scheme that depends solely on (49) is likely to succeed in practice because:

- It is important to minimize the number of function calls.
- When a and c are separated by a small x, $f(a) \cong f(b) \cong f(c)$ near the minimum, so round-off error will be a concern.
- Before localization has occurred sufficiently, the function may not be easily approximated by a parabola.

The exacting task is to invent a scheme that relies on a sure-but-slow technique, like golden section search, when the function is not cooperative,

but that switches over to (49) when the function behaves well. The task is nontrivial because:

- The housekeeping needed to avoid unnecessary function evaluations in switching between the two methods can be complicated.
- Careful attention must be given to the “endgame,” where the function is being evaluated very near to the round-off limit of equation.
- The scheme for detecting a cooperative versus noncooperative function must be very robust.

Brent’s method combines parabola fitting code with the golden section method. It is up to the task in all particulars. At any particular stage, it is keeping track of six function points that are not necessarily all distinct:

- a, one side of the bracket
- b, the other side of the bracket
- u, most recent point of evaluation
- x, the point with the current least value found so far (or the most recent one in case of a tie)
- w, the point with the current second-least value
- v, previous value of w

Also the point x_m appears in the algorithm as the midpoint between a and b, where the function is not evaluated.

Parabolic interpolation is attempted, fitting through the points x, v, and w. To be acceptable, the parabolic step must;

- fall within the bounding interval (a, b),
- imply a movement from the best current value x that is *less* than half the movement of the *step before last*.

This second criterion insures that the parabolic steps are actually converging to something, rather than, bouncing around in some nonconvergent limit cycle. In the worst possible case, where the parabolic steps are acceptable but useless, the method will approximately alternate between parabolic steps

and golden sections, converging in due course by virtue of the latter. The reason for comparing to the step before last seems essentially heuristic: Experience shows that it is better not to “punish” the algorithm for a single bad step if it can make it up on the next one.

Another principle exemplified in the code is never to evaluate the function less than a distance `tol` from a point already evaluated or from a known bracketing point. The reason is that, there is simply no information content in doing so, the function will differ from the value already evaluated only by an amount of order the round-off error. Therefore there are several tests and modifications of a potential new point, imposing this restriction, in the code. This restriction also interacts subtly with the test for “doneness”, which the method takes into account.

A typical ending configuration for Brent’s method is that a and b are $2 \times x \times$`tol` apart, with x, the best abscissa at the midpoint of *a* and *b*, and therefore fractionally accurate to ±`tol`, which is taken as the square root of the machine’s floating-point precision.

So, summarizing the things that Brent's Line Minimization Routine does:

- initially brackets the minimum
- uses a parabolic interpolation to find the minimum
- avoids unnecessary function evaluations
- changes algorithms to maintain efficiency near the minimum
- is robust for cooperative and noncooperative functions

3.1.2.2.1.3 Powell’s Multidimensional Direction Set Method

Direction set methods minimizes a function of many variables by iterating through direction sets. The difference between the various direction set methods is the choice of the direction set. These methods do not require the calculation of gradients but need utilization of one-dimensional line

minimization techniques. If started at a point **P** in N-dimensional space, and proceed from there in some vector direction **n**, then any function of N variables f(**P**) can be minimized along the line **n** by one-dimensional methods, by performing a line search that finds the λ that minimizes f(**P**+λ**n**) after which, **P** is replaced by (**P** + λ**n**) and **n** is replaced by λ**n**. As stated, different methods differ only in the way they choose the next direction **n** to try.

Powell's method produces N mutually conjugate directions along which to minimize. First, set of directions is initialized with the basis functions:

$$\mathbf{u}_i = \mathbf{e}_i \qquad i=1,\ldots, N \tag{58}$$

Next the following steps are repeated until the function stops decreasing:

i. The starting position is saved as $\mathbf{P}_0$
ii. For $i=1,\ldots, N$, $\mathbf{P}_{i-1}$ is moved to the minimum along direction $\mathbf{u}_i$ and this point is called $\mathbf{P}_i$
iii. For i=1,..., N-1, $\mathbf{u}_i \leftarrow \mathbf{u}_{i+1}$ is set
iv. $\mathbf{u}_N \leftarrow \mathbf{P}_N - \mathbf{P}_0$ is set
v. $\mathbf{P}_N$ is moved to the minimum along direction $\mathbf{u}_N$ and this point is called $\mathbf{P}_0$

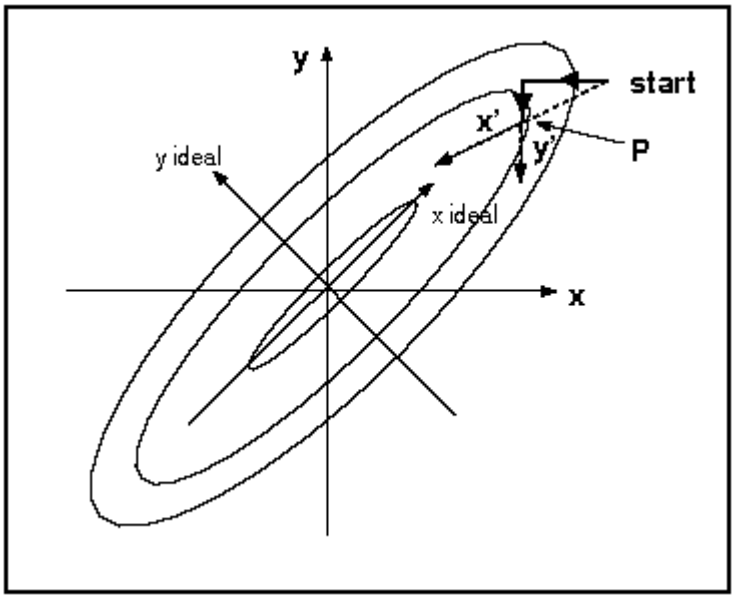


Figure 3.6. Powell's conjugate direction set method [15]

The first steps move to a new point, **P**, and line minimization continues from **P** along the new set of directions **x'** and **y** which are **x'** is the resultant direction after our first step. In this case, it is clear that ideally we should be minimizing along the axes $\mathbf{x}_{\mathbf{ideal}}$ and $\mathbf{y}_{\mathbf{ideal}}$, but even after one step, our new set of directions is far closer to these than originally, which is promising.

It can be demonstrated that N iterations of the basic procedure i.e. N (N +1) line minimizations will exactly minimize a quadratic form.
In order to implement line minimization, one dimensional methods must be utilized so that, rather than scalar-valued abscissas x, vector-valued points **P**, all lying along a given direction **n** are used.

3.1.2.2.1.4 Downhill Simplex Method

Nelder & Mead (1965) or downhill optimization method is a numerical method for minimising an objective function in a many-dimensional space. The

method uses the concept of a simplex, which is a polytope of N + 1 vertices in N dimensions; a line segment on a line, a triangle on a plane, a tetrahedron in three-dimensional space and so forth. The method approximately finds a locally optimal solution to a problem with N variables when the objective function varies smoothly.

Most of the image processing operations or sub-algorithms used require a set of operating parameters which directly affect overall performance. In order to optimize performance, an adaptation or optimization procedure which generates an optimum parameter set for each algorithm is required. Downhill Simplex algorithm is based upon a geometric figure (called the simplex) consisting of these n+1 points, together will all interconnected line segments. The algorithm starts by defining the initial simplex then takes a series of steps, in which the points of the simplex corresponding to the "worst" case is moved, or reflected, through the opposite face of the simplex. The reflection is made such that the volume of the simplex is preserved. After each reflection, the result at the new point is compared with those of the old point. If the new point is "better", the algorithm expands the simplex by moving the point further out. If it is "worse", it contracts the simplex. If no improvement is provided by the contraction, the algorithm resets the simplex to its last values and then shrinks it about the best point. The algorithm terminates when the volume of the simplex is smaller than some user-defined preset value.

If there is one initial point $\mathbf{P}_0$ then the other N points can be expressed by $\mathbf{P}_i = \mathbf{P}_0 + a_i\ \mathbf{e}_i$ where $\mathbf{e}_i$ are N unit vectors, and a_i are constants that characterize the length scale for each vector direction.

The downhill simplex method takes a series of steps, most steps just moving the highest point (the point of simplex where the function is largest) through the opposite face of the simplex to a supposed lower point (**Pr**). These steps are called reflections, and they are constructed to conserve the volume of the simplex.

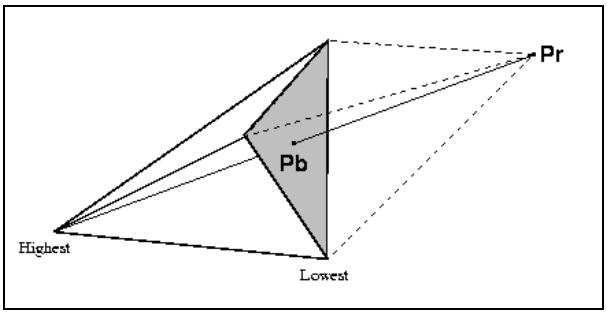


Figure 3.7. Simplex with Pr

In case when the value of the function in point **Pr** is between the second highest and lowest value of in the points of the simplex then the highest point is changed to **Pr**. When the value of function in point **Pr** is lower than or equal to in the lowest point of the simplex the value is checked in point **Prr** to see if the function drops further in direction of **Pr**.

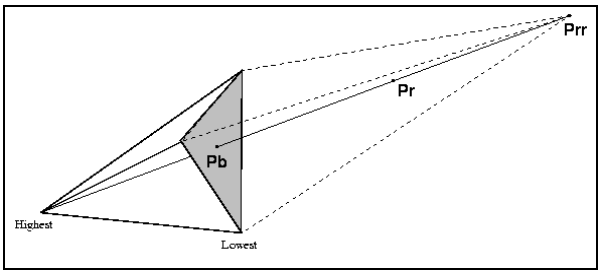


Figure 3.8. Simplex with Pr and Prr

The lower of points **Pr** and **Prr** replaces then the highest point of the simplex. If point **Pr** is greater then the simplex expands in direction to **Prr**, its volume grows. When the value of function in point **Pr** is higher than or equal to the value in the highest point of the simplex the value is checked in point **Prr'** to

see if the function is lower between the highest point and the point **Pb**. The point **Pb** is the average of simplex points except the highest point. When the value of function in point **Pr** is higher than or equal to the value in the second highest point of the simplex and lower than the value in the highest point the value is checked in point **Prr"** to see if the function is lower between the point **Pr** and the point **Pb**.

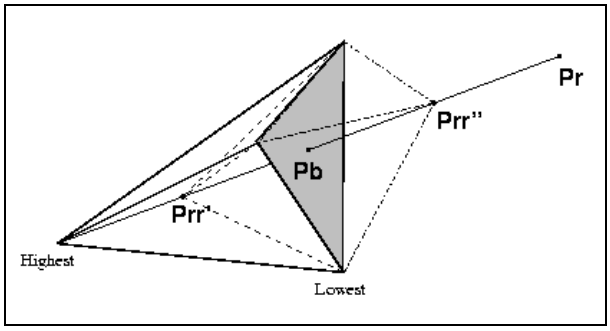


Figure 3.9. Simplex with Pr, Prr' and Prr''

When the value of function in point **Prr'**/**Prr"** is lower than in the highest point of the simplex / in the point **Pr** then **Prr'**/**Prr"** replaces the highest point of the simplex.

3.1.2.3 IMPLEMENTATION

Among three variations of information minimization methods;

MA2: the additive component $a^{-1}(\mathbf{x})$ and the multiplicative component $m^{-1}(\mathbf{x})$ are modeled by a second order polynomial

M2: consists of solely the multiplicative component $m^{-1}(\mathbf{x})$ modeled by a second order polynomial

M4: consists of solely the multiplicative component $m^{-1}(\mathbf{x})$ modeled by a fourth order polynomial

fourth order polynomial correction model M4, proved to be efficient, robust, and fast in execution in [10]. It is more appropriate to use fourth order polynomial, since the parametric correction model f^{1} must perform smooth spatially dependent color nonuniformity correction. So, the transformation of the acquired image component $v(\mathbf{x})$ to the corrected image component $u_o(\mathbf{x})$, assuming that the color nonuniformity is multiplicative represented by a fourth order polynomial, is:

$$u_o(\mathbf{x}) = f_o^{-1}(\mathbf{x})\, v(\mathbf{x}) \tag{59}$$

In general, a two dimensional fourth-order polynomial $p(\mathbf{x})$ is defined as:

$$p(\mathbf{x}) = p(x, y) = \mathbf{r}^{\mathbf{T}}\mathbf{a} \tag{60}$$

where,

$$\mathbf{r}^{\mathrm{T}}=[1 \;\; x \;\; y \;\; x^2 \;\; xy \;\; y^2 \;\; x^3 \;\; x^2y \;\; xy^2 \;\; y^3 \;\; x^4 \;\; x^3y \;\; x^2y^2 \;\; xy^3 \;\; y^4] \tag{61}$$

and,

$$\mathbf{a}^{\mathrm{T}}=[a_{00} \;\; a_{10} \;\; a_{01} \;\; a_{20} \;\; a_{11} \;\; a_{02} \;\; a_{30} \;\; a_{21} \;\; a_{12} \;\; a_{03} \;\; a_{40} \;\; a_{31} \;\; a_{22} \;\; a_{13} \;\; a_{04}] \tag{62}$$

$$\begin{aligned} p(\mathbf{x}) &= p(x, y) \\ &= a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{30}x^3 + a_{21}x^2y + \\ &\quad a_{12}xy^2 + a_{03}y^3 + a_{40}x^4 + a_{31}x^3y + a_{22}x^2y^2 + a_{13}xy^3 + a_{04}y^4 \end{aligned} \tag{63}$$

Considering K=15, the correction steps becomes to:

i. The optimal parameter ${}^{m}b_o$ is found by Brent's one dimensional optimization algorithm and Powell's multidimensional directional set method [15]:

$$\{{}^m b_o\} = \arg\min_{\{{}^m b\}}\{I[u(\mathbf{x})]\} = \arg\min_{\{{}^m b\}}\{I[v(\mathbf{x})m^{-1}(\mathbf{x})]\} \tag{64}$$

Estimations of the true image are examined in order to find the optimal parameter ${}^m b_o$,:

$$u(\mathbf{x}) = v(\mathbf{x})m^{-1}(\mathbf{x}) \tag{65}$$

the multiplicative component with fourth-order polynomial is defined as:

$$m^{-1}(\mathbf{x}) = 1 + \sum_{i=2}^{15} {}^m b_i \frac{q_i(\mathbf{x}) - {}^m c_i}{{}^m d_i} \tag{66}$$

The fourth order polynomial terms $q_i(\mathbf{x})$s are the elements of the matrix:

$$\mathbf{q}^T(\mathbf{x}) = \mathbf{q}^T(x, y) = [1 \;\; x \;\; y \;\; x^2 \;\; xy \;\; y^2 \;\; x^3 \;\; x^2y \;\; xy^2 \;\; y^3 \;\; x^4 \;\; x^3y \;\; x^2y^2 \;\; xy^3 \;\; y^4] \tag{67}$$

where,

$q_1(\mathbf{x}) = q_1(x, y) = 1$ $\quad$ $q_2(\mathbf{x}) = q_2(x, y) = x$

$q_3(\mathbf{x}) = q_3(x, y) = y$ $\quad$ $q_4(\mathbf{x}) = q_4(x, y) = x^2$

$q_5(\mathbf{x}) = q_5(x, y) = xy$ $\quad$ $q_6(\mathbf{x}) = q_6(x, y) = y^2$

$q_7(\mathbf{x}) = q_7(x, y) = x^3$ $\quad$ $q_8(\mathbf{x}) = q_8(x, y) = x^2y$

$q_9(\mathbf{x}) = q_9(x, y) = xy^2$ $\quad$ $q_{10}(\mathbf{x}) = q_{10}(x, y) = y^3$

$q_{11}(\mathbf{x}) = q_{11}(x, y) = x^4$ $\quad$ $q_{12}(\mathbf{x}) = q_{12}(x, y) = x^3y$

$q_{13}(\mathbf{x}) = q_{13}(x, y) = x^2y^2$ $\quad$ $q_{14}(\mathbf{x}) = q_{14}(x, y) = xy^3$

$q_{15}(\mathbf{x}) = q_{15}(x, y) = y^4$ (68)

ii. The optimal component $m_o^{-1}(\mathbf{x})$ is defined by the optimal parameter ${}^m b_o$:

$$m_o^{-1}(\mathbf{x}) = 1 + \sum_{i=2}^{15} {}^m b_{oi} \frac{q_i(\mathbf{x}) - {}^m c_i}{{}^m d_i} \tag{69}$$

iii. The optimal component $m_o^{-1}(\mathbf{x})$ transforms the acquired image component $v(\mathbf{x})$ into the optimally corrected image component $u_o(\mathbf{x})$:

$$u_o(\mathbf{x}) = v(\mathbf{x})m_o^{-1}(\mathbf{x}) \tag{70}$$

In the training mode, a defect free RGB image $V(j, k)$ of size MxNx3x8 bits is acquired and separated into components $V_1(j, k)$, $V_2(j, k)$, $V_3(j, k)$ representing Red, Green and Blue, respectively. Each image component is composed of MxN pixels whose values representing the corresponding color intensity are stored in 8 bits, taking values between 0 and 255. So, the size of the components is MxNx8 bits, where M is the number of the rows and N is the number of the columns of the image component. j is used as indice for rows and k is used as indice for columns for each image component which are independently corrected for spatial color nonuniformity to get the optimally corrected components $U_{1o}(j, k)$, $U_{2o}(j, k)$ and $U_{3o}(j, k)$ to be merged into the corrected color image $\mathbf{U}_o(j, k)$.

Prior to the optimization process which is the first correction step, the neutralization constant matrix C composed of elements C_i, which corresponds to ${}^m c_i$ of the continuous domain, and the normalization constant matrix D composed of elements D_i, which corresponds to ${}^m d_i$ of the continuous domain for the multiplicative component, are calculated for each image component, by using the discrete versions of (39) and (41), respectively, as follows:

$$C_i = \frac{\sum_{k=1}^{N}\sum_{j=1}^{M} V(j,k)Q_i(j,k)}{\sum_{k=1}^{N}\sum_{j=1}^{M} V(j,k)} \qquad i=2,\ldots\ldots, 15 \tag{71}$$

makes up the neutralization constant matrix of size 15x1:

$$C=[0\ C_2\ C_3\ C_4\ C_5\ C_6\ C_7\ C_8\ C_9\ C_{10}\ C_{11}\ C_{12}\ C_{13}\ C_{14}\ C_{15}] \tag{72}$$

$$D_i = \frac{1}{\Theta}\sum_{k=1}^{N}\sum_{j=1}^{M}\left|V(j,k)\left[Q_i(j,k) - C_i\right]\right| \qquad i=2,\ldots\ldots, 15 \tag{73}$$

makes up the normalization constant matrix of size 15x1:

$$D=[1\ D_2\ D_3\ D_4\ D_5\ D_6\ D_7\ D_8\ D_9\ D_{10}\ D_{11}\ D_{12}\ D_{13}\ D_{14}\ D_{15}] \tag{74}$$

where, Θ is the size of the image domain, calculated as:

$$\Theta = \sum_{k=1}^{N}\sum_{j=1}^{M} 1 \tag{75}$$

yielding the number of pixels in the whole image for the discrete version:

$$\Theta = MN \tag{76}$$

Also, the fourth-order polynomial terms are used in the discrete domain version $Q_i(\mathbf{x}) = Q_i(j, k)$, which are the elements of the matrix of size 15x1:

$$Q(\mathbf{x}) = Q(j, k) = [1\ \ j\ \ k\ \ j^2\ \ jk\ \ k^2\ \ j^3\ \ j^2k\ \ jk^2\ \ k^3\ \ j^4\ \ j^3k\ \ j^2k^2\ \ jk^3\ \ k^4] \tag{77}$$

where,

$$\begin{array}{lll}
Q_1(\mathbf{x}) = Q_1(j, k) = 1 & Q_2(\mathbf{x}) = Q_2(j, k) = j & Q_3(\mathbf{x}) = Q_3(j, k) = k \\
Q_4(\mathbf{x}) = Q_4(j, k) = j^2 & Q_5(\mathbf{x}) = Q_5(j, k) = jk & Q_6(\mathbf{x}) = Q_6(j, k) = k^2 \\
Q_7(\mathbf{x}) = Q_7(j, k) = j^3 & Q_8(\mathbf{x}) = Q_8(j, k) = j^2k & Q_9(\mathbf{x}) = Q_9(j, k) = jk^2 \\
Q_{10}(\mathbf{x}) = Q_{10}(j, k) = k^3 & Q_{11}(\mathbf{x}) = Q_{11}(j, k) = j^4 & Q_{12}(\mathbf{x}) = Q_{12}(j, k) = j^3k \\
Q_{13}(\mathbf{x}) = Q_{13}(j, k) = j^2k^2 & Q_{14}(\mathbf{x}) = Q_{14}(j, k) = jk^3 & Q_{15}(\mathbf{x}) = Q_{15}(j, k) = k^4
\end{array} \tag{78}$$

The correction steps are achieved in the discrete domain is as follows:

i. The optimal parameter matrix B_o, which corresponds to ${}^{m}b_o$ for the multiplicative component in the continuous domain, is composed of

elements B_{oi} for i=1,…….,15. B_o of size 15x1, is found by Powell's multidimensional directional set method utilizing Brent's one dimensional optimization algorithm:

$$\{B_o\} = \arg\min_{\{B\}}\{I[U(j,k)]\} \tag{79}$$

B is composed of B_is for i=1,…….,15 as follows:

$$B=[1\ B_2\ B_3\ B_4\ B_5\ B_6\ B_7\ B_8\ B_9\ B_{10}\ B_{11}\ B_{12}\ B_{13}\ B_{14}\ B_{15}] \tag{80}$$

$$U(j,k) = V(j,k) \otimes M(j,k) \tag{81}$$

where $V(j,k)$ is the acquired image component matrix and $U(j,k)$ is the estimated corrected image component matrix both with size MxN. $M(j,k)$ is the multiplicative correction component matrix in the discrete domain, which corresponds to the multiplicative correction component $m^{-1}(\mathbf{x})$ in the continuous domain, which also has size MxN. The corrected image component estimation matrix is obtained by the element-wise multiplication ($\otimes$) of the acquired image matrix and the correction component matrix. The multiplicative correction component matrix $M(j,k)$ is defined as:

$$M(j,k) = 1 + \sum_{i=2}^{15} B_i \frac{Q_i(j,k) - C_i}{D_i} \tag{82}$$

ii. The optimal multiplicative correction component matrix $M_o(j,k)$ is defined by the optimal parameter matrix B_o:

$$M_o(j,k) = 1 + \sum_{i=2}^{15} B_{oi} \frac{Q_i(j,k) - C_i}{D_i} \tag{83}$$

iii. The optimal multiplicative correction component matrix $M_o(j,k)$ transforms the acquired image component matrix $V(j,k)$ into the optimally corrected image component matrix $U(j,k)$ by utilizing a element-wise multiplication ($\otimes$):

$$U_o(j,k) = V(j,k) \otimes M_o(j,k) \tag{84}$$

Since the correction is achieved depending on the information function I, calculation of the image information is important. The information of the estimated corrected image component is expressed by (3) is as follows:

$$I[U(j,k)] = H[U(j,k)] = -\sum_{n=1}^{256} Pr(n) \log Pr(n) \tag{85}$$

where $Pr(n)$ is the nth element of the probability matrix Pr, which corresponds to the probability that a pixel on the image $U(j,k)$ has value (n-1). The variation of the parameters of the correction components results in smooth changes of the entropy, which is calculated by using the probability matrix obtained from the intensity histogram of the current estimated corrected image component $U(j,k)$.

There is a compromise between the speed of the entropy calculation and statistical power. Although subsampling the image data and using approximately 5000 image samples is sufficient to form a histogram that is statistically enough powerful and enables efficient calculation of entropy; all the image pixels were used to form the histograms due to the high process power.

Since the intensity transformation, which is applied to the acquired image component $V(j,k)$ to obtain $U(j,k)$, may in general transform the gray values out of the image gray-level range [0, 255], the histograms are formed in a range of gray levels [-255, 511].

Because of the intensity transformation, the integer pixel intensity value g is transformed to a new real value g', lying between two integer values:

$$k \le g' < k+1 \tag{86}$$

A matrix, hpe, is formed by scanning the subsample image window of about 10000 pixels of $U(j,k)$ with two columns. The first column is composed of the real g' values of the pixel values in the window, and the second column is composed of the number of the pixels, the frequency distribution, with value g' in the same row. The g' values take value in a range of gray levels [-255, 511], and are not in order.

In order to update the corresponding histogram entries, an intensity interpolation is used to convert the real g' values to integer values and a new matrix hei wih size 767x1 is formed. The 1st row of hei corresponds to the frequency of the pixels with value -255; the 2nd row of hei corresponds to the frequency of the pixels with value -254; the 3rd row of hei corresponds to the frequency of the pixels with value -253;…….; the 767th row of hei corresponds to the frequency of the pixels with value 511. hei is formed by the quantization of hpe. So hei is formed by the histogram entries $h(k)$ updated by interpolating the elements of hpe as follows:

If $g'_i - k \le 0.5$ then $h(k) \leftarrow h(k)+1$

If $g'_i - k > 0.5$ then $h(k+1) \leftarrow h(k+1)+1$

In order to reduce the effects of imperfect intensity interpolation the histogram entries $h(n)$ are slightly blurred and heib matrix is formed, before calculating the probabilities $Pr(n)$:

$$h(n) \leftarrow \sum_{i=-t}^{t} h(n+i)(t+1-|i|) \tag{87}$$

where, the parameter t defines the size ($2t$+1) of a triangular window and is set to 2 yielding a triangular window of size 5:

$$h(n) \leftarrow \sum_{i=-2}^{2} h(n+i)(3-|i|) \tag{88}$$

$$h(n) \leftarrow h(n-2)+2h(n-1)+3h(n)+2h(n+1)+h(n+2) \tag{89}$$

The blurring process achieves weighted summation in a window of size five, the inner elements weighted more than the outer elements. heib is normalized by dividing its elements by 1+2+3+2+1=9, and normheib matrix is obtained which is formed up with weighted average of the interpolated histogram entries.

The elements of the probability matrix Pr, $Pr(n)$'s are calculated by the division of the elements of the normheib histogram matrix by the total number of the pixels used in the image.

The correction steps are achieved after obtaining the probability matrix Pr for each image component and the optimal spatially corrected color image $\mathbf{U}_o(j,k)$ is obtained.

Spatial color nonuniformity correction function, $f_{oi}^{-1}(\mathbf{x})$, is saved in the blister model, to correct the inhomogeneities of the acquired images in the inspection mode.

3.2 IMAGE SEGMENTATION

Reducing the adverse effect of the color inhomogeneities of the acquired image by using spatial color nonuniformity correction, reduces the overlapping of the probability densities in the feature space and results in

less complex clusters [5]. More compact clusters can be obtained by transforming the RGB image to the other color spaces such as HSV that separate color and intensity values. On the other hand, when a color space transformation is used, color features become unstable at low intensity values [8], and it is a time consuming process which would result in an unfeasible inspection system. Representing the distribution in the feature space with a parametric model (e.g. Gaussian mixture) introduces severe artifacts. On the other hand, nonparametric cluster analysis uses the modes of the underlying probability density to define the cluster centers and the valleys in the density to define the boundaries separating the clusters. So, a nonparametric clustering based image segmentation method is required because complex clusters are inherently encountered in the color space. Some of the nonparametric techniques are multivariate histogram, the nearest neighbor method and kernel estimation [3]. Since the number of bins exponentially increases with the space dimension and artifacts are introduced by quantization, multivariate histograms are less useful for higher dimensional feature spaces.

The nearest neighbor method and Kernel estimation method were used and compared in this thesis work. The nearest neighbor method is utilized for both HSV and RGB color spaces, while Kernel estimation method is utilized only for RGB color space. The nearest neighbor method is highly affected by the local noise making the localization of the modes difficult. Kernel estimation is a good practical choice for low to medium data sizes because it is simple, uniformly consisted in probability and consistent in a mean-square sense.

3.2.1. RGB & HSV COLOR SPACE

The HSV stands for the Hue, Saturation, and Value (Tint, Shade, and Tone). The coordinate system in a hexacone is illustrated in Figure 3.10. The Value represents intensity of a color, which is decoupled from the color information

in the represented image. The hue and saturation components are intimately related to the way human eye perceives color resulting in image processing algorithms with physiological basis.

As hue varies from 0 to 1.0, the corresponding colors vary from red, through yellow, green, cyan, blue, and magenta, back to red, so that there are actually red values both at 0 and 1.0. As saturation varies from 0 to 1.0, the corresponding colors (hues) vary from unsaturated (shades of gray) to fully saturated (no white component). As value, or brightness, varies from 0 to 1.0, the corresponding colors become increasingly brighter.

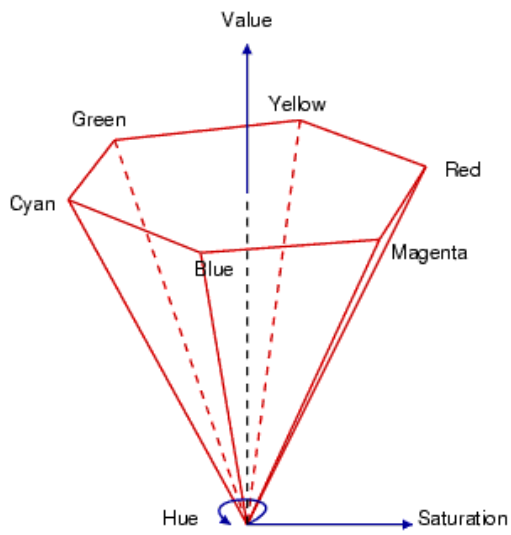


Figure 3.10. HSV hexacone

RGB color space is also three-dimensional orthogonal coordinate systems, meaning that there are three axes (in this case the Red, Green, and Blue color intensities) that are perpendicular to one another. This color space is illustrated in Figure 3.11. The red intensity starts at zero at the origin and increases along one of the axes. Similarly, green and blue intensities start at

the origin and increase along their axes. Because each color can only have values between zero and some maximum intensity (255 for 8-bit depth), the resulting structure is the cube. We can define any color simply by giving its red, green, and blue values, or coordinates, within the color cube.

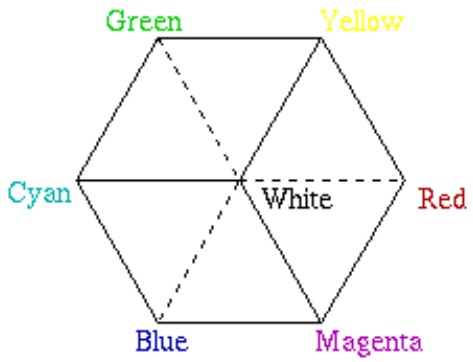


Figure 3.11. RGB color cube along its principal diagonal

3.2.1.1. RGB TO HSV CONVERSION

The obtainable HSV colors lie within a triangle whose vertices are defined by the three primary colors in RGB space:

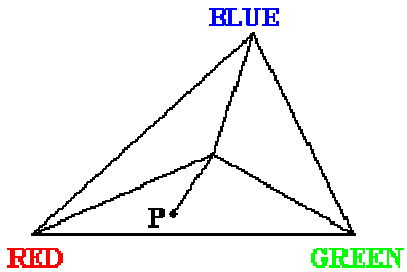


Figure 3.12. Obtainable HSV color from RGB color space

The hue of the point P is the measured angle between the line connecting P to the triangle center and line connecting RED point to the triangle center. The saturation of the point P is the distance between P and triangle center. The value (intensity) of the point P is represented as height on a line perpendicular to the triangle and passing through its center. The grayscale points are situated onto the same line. And the conversion formula is as follows :

$$H = \cos^{-1}\left\{\frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right\} \tag{90}$$

$$S = 1 - \frac{3}{R+G+B}\left[\min(R,G,B)\right] \tag{91}$$

$$V = \frac{1}{3}(R+G+B) \tag{92}$$

3.2.1.2. HSV TO RGB CONVERSION

Conversion from HSV space to RGB space is more complex. And, given to the nature of the hue information, we will have a different formula for each sector of the color triangle.

<u>Red-Green Sector:</u>

$$for \quad 0° < H < 120°$$

$$b = \frac{1}{3}(1-S), \quad r = \frac{1}{3}\left[1+\frac{S\cos H}{\cos(60°-H)}\right], \quad g = 1-(r+b) \tag{93}$$

Green-Blue Sector:

$$for \quad 120° < H \leq 240°$$

$$r = \frac{1}{3}(1-S), \quad g = \frac{1}{3}\left[1 + \frac{S\cos H}{\cos(60° - H)}\right], \quad b = 1-(r+b) \tag{94}$$

Blue-Red Sector:

$$for \quad 240° < H \leq 360°$$

$$g = \frac{1}{3}(1-S), \quad b = \frac{1}{3}\left[1 + \frac{S\cos H}{\cos(60° - H)}\right], \quad r = 1-(r+b) \tag{95}$$

3.2.2. MAXSHIFT ALGORITHM

Kernel estimation based clustering essentially relies on two techniques:

- The underlying density is estimated and a hierarchical data structure is derived, based on which the data is decomposed.
- Density gradient estimation is used, the modes being detected with the hill climbing mean shift procedure

Both the density and the density gradient estimation require the search for the data points falling in the neighborhood inside the employed kernel, which is called multidimensional range searching. However, for applications involving large data sets (e.g. multi-spectral image segmentation, image restoration, image coding), since the complexity of both the kernel estimation and density gradient estimation is proportional to the square of the number of data points, they become computationally expensive. Inaccurate results are got if the data is subsampled. [3] Presents a practical algorithm based on the mean shift procedure for unsupervised nonparametric clustering of large data sets solving the stated problem, and a novel efficient method derived from [3] is proposed in [5] which works faster and easy to implement.

Color intensities $U_{1o}(j, k)$, $U_{2o}(j, k)$, $U_{3o}(j, k)$ of the optimally corrected color image $\mathbf{U}_o(j, k)$ are mapped into a color space Y, and there they form a multivariate and multimodal probability distribution $P(\boldsymbol{y})$, which is estimated by normalizing the frequency distribution of the feature vector $\boldsymbol{y}=[U_{1o}, U_{2o}, U_{3o}] \in Y$.

The task of clustering based segmentation is to separate the modes of the underlying probability distribution, where the clusters, highest density regions, are centered on in a multivariate color space. In order to locate the dense regions in the feature space, corresponding to homogeneous regions in the image, a search kernel can be used. The algorithm to estimate the density gradient for color image segmentation application used in [3] is based on shifting the sphere-shaped kernel Sy by the mean shift vector $\boldsymbol{\eta}$, the vector between the local mean of $P(\boldsymbol{y})$ inside Sy and the centre of Sy:

$$\boldsymbol{\eta}(\boldsymbol{y}) = \arg\left[\underset{y \in Sy}{mean}(P(\boldsymbol{y})) \right] - \boldsymbol{y} \tag{96}$$

Until a mode $\boldsymbol{c_i}$ is reached, i.e. local maximum for the probability distribution, the translation of the search window is repeated. Since the shifts are always in the $\boldsymbol{c_i}$ direction, $\boldsymbol{\eta}$ becomes equal to approximately zero when the end positions; $\boldsymbol{c_i}$'s i=1,……., n are reached, starting from locations $\boldsymbol{y}$, where $\boldsymbol{c_i}$'s also represent the n significant colors of the image.

Keeping the robustness, two modifications are made in [5], to make the algorithm work faster, which is also employed in this thesis:

1. Since for the digital images the feature space is dense and discrete, the kernel is shifted to the maximum inside the kernel, instead of the mean.

2. The shape of the kernel is changed to a cube.

$$\boldsymbol{\eta}(\boldsymbol{y}) \rightarrow \boldsymbol{\mu}(\boldsymbol{y}) = \arg\left[\max_{y \in Sy}(P(\boldsymbol{y})) \right] - \boldsymbol{y} \tag{97}$$

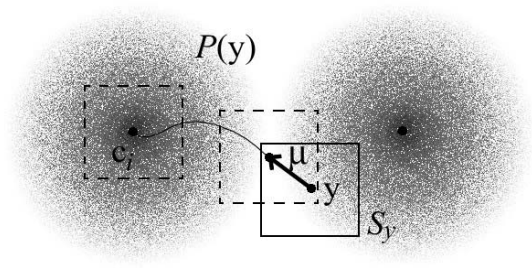


Figure 3.13. Mode seeking with max shift algorithm using a cube-kernel

In the training mode, an operator must select one pixel for each of the different desired colors. If the pill to be partially segmented is of one color, the operator selects only one pixel at location (j_o, k_o) with the desired color $\boldsymbol{V}(j_o, k_o) = [V_{1o}, V_{2o}, V_{3o}]$, on any tablet. The corresponding feature vector is $\boldsymbol{y_o}=[V_{1o}, V_{2o}, V_{3o}] \in Y$. The mode $\boldsymbol{c_o}$ of the tablets' color probability density is obtained by the max shift algorithm started from the initial location $\boldsymbol{y_o}$:

$$\boldsymbol{c_o} = MaxShift(\boldsymbol{y_o}) \tag{98}$$

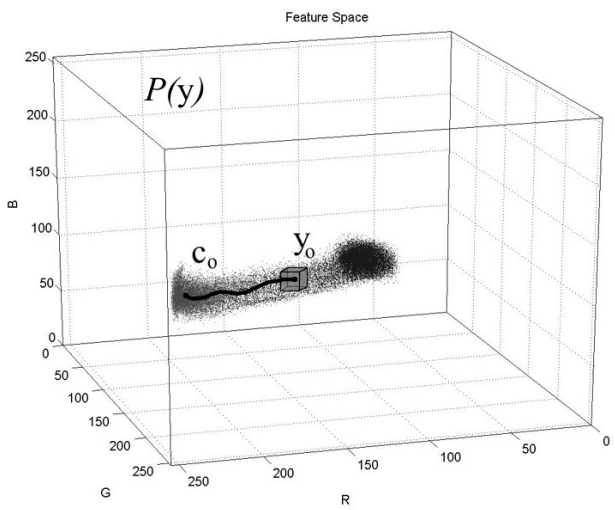


Figure 3.14. Determination of clusters in the feature space

The initial value of the labeling function for all $\boldsymbol{y} \in Y$, is set to 1, indicating the background:

$$L(\boldsymbol{y}) = 1 \qquad \forall \boldsymbol{y} \in Y \tag{99}$$

After obtaining $\boldsymbol{c_o}$ and labeling its location with 0, indicating the tablet:

$$L(\boldsymbol{c_o}) = 0 \tag{100}$$

the cluster growth algorithm is used to determine the corresponding object feature points forming a cluster. Growth begins from the mode by the application of the MaxShift algorithm to the points, $\boldsymbol{y_n}$ in the neighborhood of $\boldsymbol{c_o}$:

If $L(\boldsymbol{y_n} + \boldsymbol{\mu}(\boldsymbol{y_n})) = 0 \qquad \forall \boldsymbol{y_n} \in Y$

then $\quad L(\boldsymbol{y_n}) = 0 \qquad (101)$

Until there are no points in the temporal cluster neighborhood satisfying (101), the cluster growth algorithm is repeated. So, a nonparametric 3D boundary, which is optimal with regard to the probability distribution, is formed between the object and the background in the color space.

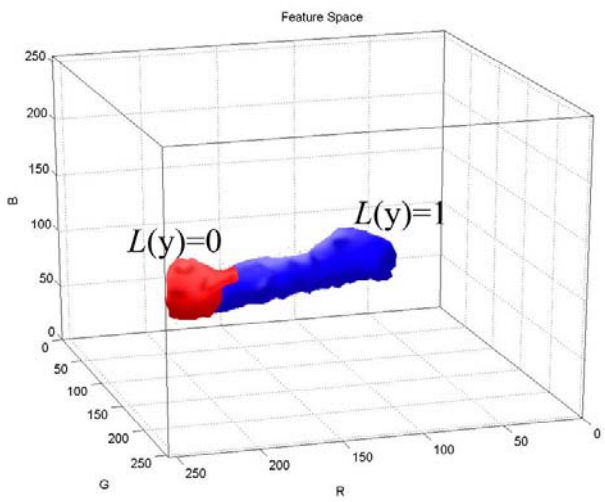


Figure 3.15. Clusters of different colors separated by an optimal 3D boundary

A binary image $BW(j, k)$ is obtained directly by the labelling matrix:

$$BW(j,k) = L(\boldsymbol{U}_o(j,k)) \qquad \forall (j,k) \in X \tag{102}$$

where X is the spatial image domain. The intensity of the pixels of the binary image is set equal to the label value (0 for the tablet and 1 for the background) of the corresponding image color intensity value.

Nonparametric color image segmentation model is saved in the blister model, to segment the corrected images in the inspection mode.

3.2.3. NEAREST NEIGHBOR ALGORITHM

In object recognition task classification can be achieved by representing the features of an object in the feature space as a point and using a particular classification method. In nearest neighbor method, having ideal objects each representing a class *i* in an N-dimensional feature space as o_{ij}, *j*=1,...N and

representing the features of the detected object as s_j; to determine the class of the detected object, its distance from the feature points is calculated. The object is classified as the model object for which the distance is the minimum among all model objects. Any distance metric can be used, Euclidian,etc.

The distance d_j of the detected object to the model objects is calculated by [9]:

$$d_j = \sqrt{\left[\sum_{i=1}^{N}\left(s_j - o_{ij}\right)^2\right]} \tag{103}$$

The detected object is classified as C according to:

$$d_C = \min_{j=1}^{N}(d_j) \tag{104}$$

Due to to the difficulty of finding a model object in practice, rather than using a model object point in the feature space, the class can be represented by a cluster of many model object points known to belong to that class. The nearest-neighbor algorithm is extremely simple to implement, but the performance of this method is not low for overlapping clusters.

K-Nearest Neighbor classifier performs a method for classifying phenomena based upon observable features, similar to the nearest neighbor classification method. Nearest neighbor is calculated by using a distance measure. When we need a prediction about a known point, nearest neighbor algorithm finds the closest point to training point according to some distance metric such as Euclidian distance. K-Nearest Neighbor algorithm selects a set of k nearest neighbors and assigns the class label to the new data point based upon the most numerous classes with the set.

Computing the nearest neighbor [18] of an input can be challenging for large data sets. Several different types of nearest neighbor finding algorithms

including binning methods such as k-d tree [1] and Elias [16], and those which make use of the Triangle Inequality [6, 17, 7]. Implementations of the ICP typically employ one of these general methods, with the k-d tree likely being the most widely used method.

In this thesis work Euclidian distance measure is used to calculate the distances. The number of classes is the number of of the main colors on tablets plus one for the background. For monocolored tablet case the number of classes is two. When using RGB color space as the feature space in the segmentation, N is 3, while N is 1 for the HSV case since the color information H component is 1-D.

3.3 FEATURE EXTRACTION

The binary image of the defect free tablet image, which is obtained by the nonparametric color image segmentation of the spatially corrected optimal image, is used to extract the feature values, size, position and shape to complete the blister model.

By applying connected component labeling algorithm as defined in [9] to the binary image BW, the size and the position of each tablet are extracted and saved in the blister model.

The sequential connected component algorithm using 4-connectivity, defined in [9] is extended to 8-connectivity and used in this thesis. The points in a connected component form a candidate region to represent a tablet. The connected component algorithms are sequential in nature since finding connected components is a global operation. In order to find the object locations and properties, connected components must be determined when there is more than one object in the image, which is the case for tablet inspection. The algorithm finds every connected component in the image, assigns a unique label to all points of the same component and results in X

matrix of the same size with the acquired image. Two passes over the image are required by the sequential algorithm which works only with two rows of the image each time. This algorithm looks the 8-neighborhood of a pixel which means that, the two pixels share at least one corner while two pixels must share at least one boundary for 4-neighborhood. Already used labels are tried to be assigned to the pixels. In case of different labels in the neighborhood of a pixel, an equivalence table is formed to keep track of all labels that are equivalent. In the second pass, this table is used to assign a unique label to all pixels of a component.

The steps are:

1) Scan the image left to right, top to bottom
2) If the pixel is 0 (tablet), then
    a. If at least one of its upper-left, or upper, or upper-right, or left neighbors has a label, and all the labels are the same, then copy the label.
    b. If upper-left and at least one of its upper, upper-right or left neighbors has a label, and there are different labels, then copy the label of the upper-left, enter the labels in the equivalence table as equivalent labels.
    c. If upper-left does not have a label and at least two of its upper, upper-right or left neighbors has a label, and there are different labels, then copy the label of the upper, enter the labels in the equivalence table as equivalent labels.
    d. If upper-left and upper does not have a label and its upper-right and left neighbors has different labels, then copy the label of the upper-right, enter the labels in the equivalence table as equivalent labels.
    e. Otherwise assign a new label to the pixel and enter its label to the equivalence table.
3) Repeat step 2 until there is no pixel remaining
4) Find the lowest label for each equivalent set in the equivalence table.

5) Replace each label in its equivalent set found in step 4 by scanning the image second time.

The two characteristics, size (area) and the position of the each individual component are computed as defined in [9].

Area of a region is defined as the number of pixels in that region for a binary image. Then, for K components, CC_1, $CC_{2,.....,}$ CC_K, found after connected component labeling algorithm, the area of each component is the number of pixels in that component. Area can be calculated for the ith component as follows:

$$Area_i = \sum_{(j,k)\in CC_i}\sum 1 \qquad i=1, 2,\ldots, K \qquad (105)$$

There are different ways to specify the position of an object, such as its enclosing rectangle or centroid. In industrial applications, a controlled environment is used; objects appear on a known surface, the position of the camera is known with respect to table etc., which is the case, the position of every object determines its spatial location. Although other methods, such as the enclosing rectangle can be used, the positions of the objects are defined using the centers of the corresponding area, because the centre of the area is a point and less sensitive to noise in the image.

The centre of the area is the centre of mass considering the object points as the mass at that point. Since the object points in the binary image BW are labeled with 0, while the background is labeled with 1; its logical complement CBW is used in the calculation of the centroids to make the weight of the object points 1, and the weight of the background 0. The x and y coordinates of the centroid of the ith component are:

$$Centroid_i^x = \frac{\sum_{(j,k)\in CC_i}\sum kCBW(j,k)}{Area_i} = \frac{\sum_{(j,k)\in CC_i}\sum k}{Area_i} \qquad i=1, 2,\ldots, K \qquad (106)$$

$$Centroid_i^y = \frac{\sum_{(j,k)\in CC_i}\sum jCBW(j,k)}{Area_i} = \frac{\sum_{(j,k)\in CC_i}\sum j}{Area_i} \qquad i=1, 2,\ldots, K \qquad (107)$$

The positions calculated are not necessarily integers and lie between the integer values of the image indices which does not imply that the calculated position is better than the pixel resolution of pixel coordinates.

Area, projection, orientation and circularity implying the shape of each region are used as the shape information and calculated according to [9].

The projection of each image region corresponding to tablets are obtained by the number of 1 pixels in CBW that are on lines perpendicular to each bin, which are the partitions of the line. The horizontal projections along the rows and the vertical projections along the columns are found by finding the number of 1 pixels in CBW for each bin in the horizontal and vertical directions, respectively as follows:

$$\Pr ojection_i^H(j) = \sum_{k\in CC_i} CBW(j,k) = \sum_{k\in CC_i} 1 \qquad i=1, 2,\ldots, K \qquad (108)$$

$$\Pr ojection_i^V(k) = \sum_{j\in CC_i} CBW(j,k) = \sum_{j\in CC_i} 1 \qquad i=1, 2,\ldots, K \qquad (109)$$

Since the first moments of an image region equal to the first moments of its projections, and the area -the zeroth-order moment- can easily be calculated by using either horizontal or vertical projection, the area and the centroid for each image region can also be calculated as follows:

$$Area_i = \sum_{k\in CC_i} \Pr ojection_i^V(k) = \sum_{j\in CC_i} \Pr ojection_i^H(j) \qquad i=1, 2,\ldots, K \qquad (110)$$

$$Centroid_i^x = \frac{\sum_{k\in CC_i} k \Pr ojection_i^V(k)}{Area_i} \qquad i=1, 2,\ldots, K \qquad (111)$$

$$Centroid_i^y = \frac{\sum_{j \in CC_i} j \, \Pr ojection_i^H(j)}{Area_i} \qquad i=1, 2,\ldots, K \qquad (112)$$

In order to define a unique orientation for an image region, the component must be elongated [9]. So shapes like circles do not have unique orientations. The orientation of an object is defined as the orientation of the axis of elongation, for which the axis of least second moment is used. The axis of second moment for an object region is the line for which the sum of the squared distances between object region points and the line is minimum. For all object points of the connected component, minimize the sum of distances from the line:

$$D_i^2 = \sum_{(j,k) \in CC_i} \sum r_{jk}^2 CBW(j,k) = \sum_{j,k \in CC_i} \sum r_{jk}^2 \qquad i=1, 2,\ldots, K \qquad (113)$$

where r_{jk} is the distance from an object point [j,k] to the line. Representing the line in polar coordinates, letting ρ be the distance of the line from the origin and θ be the angle between the line and the x-axis, increasing in the counterclockwise direction:

$$\rho_i^2 = x \cos\theta_i + y \sin\theta_i \qquad i=1, 2,\ldots, K \qquad (114)$$

The distance r of a point (x,y) is obtained by substituting the coordinates in the line equation:

$$r^2 = (x \cos\theta_i + y \sin\theta_i - \rho_i)^2 \qquad i=1, 2,\ldots, K \qquad (115)$$

Substituting (1115) in (113) yields the regression problem for fitting a straight line to the object points. The aim is to find the model parameters ρ and θ by minimizing:

$$D_i^2 = \sum_{(j,k)\in CC_i}\sum (x_{jk}\cos\theta_i + j_{jk}\sin\theta_i - \rho_i)^2 CBW(j,k)$$

$$D_i^2 = \sum_{(j,k)\in CC_i}\sum (x_{jk}\cos\theta_i + j_{jk}\sin\theta_i - \rho_i)^2 \qquad i=1, 2,\ldots, K \qquad (116)$$

In order to minimize D^2, its derivative with respect to ρ is set to 0 and ρ is obtained as:

$$\rho_i = (\bar{x}_i \cos\theta_i + \bar{y}_i \sin\theta_i) \qquad i=1, 2,\ldots, K \qquad (117)$$

Since this is the line equation for $(x,y) = (\bar{x},\bar{y})$ given by (114), it is proven that the regression line passes through the center of mass points;

$$\bar{x} = Centroid_i^x \quad \text{and} \quad \bar{y} = Centroid_i^y \qquad i=1, 2,\ldots, K \qquad (118)$$

Substituting (117) in (116) and replacing:

$$x' = x - centroid_i^x \quad \text{and} \quad y' = y - centroid_i^y \qquad (119)$$

$$D_i^2 = a_i \cos^2\theta_i + b_i \sin\theta_i \cos\theta_i + c_i \sin^2\theta_i \qquad i=1, 2,\ldots, K \qquad (120)$$

The second order moments are the parameters:

$$a_i = \sum_{(j,k)\in CC_i}\sum (x'_{jk})^2 CBW(j,k) = \sum_{(j,k)\in CC_i}\sum (x'_{jk})^2 \qquad i=1, 2,\ldots, K \qquad (121)$$

$$b_i = 2\sum_{(j,k)\in CC_i}\sum x'_{jk} y'_{jk} CBW(j,k) = 2\sum_{(j,k)\in CC_i}\sum x'_{jk} y'_{jk} \qquad i=1, 2,\ldots, K \qquad (122)$$

$$c_i = 2\sum_{(j,k)\in CC_i}\sum (y'_{jk})^2 CBW(j,k) = \sum_{(j,k)\in CC_i}\sum (y'_{jk})^2 \qquad i=1, 2,\ldots, K \qquad (123)$$

The minimization problem is written as:

$$D_i^2 = \frac{1}{2}(a_i + c_i) + \frac{1}{2}(a_i - c_i)\cos 2\theta_i + \frac{1}{2} b \sin 2\theta_i \qquad i=1, 2,\ldots, K \qquad (124)$$

The derivative of (124) with respect to θ and setting to zero yields:

$$\tan 2\theta_i = \frac{b_i}{a_i - c_i} \qquad i=1, 2,\ldots, K \qquad (125)$$

The orientation is given by:

$$\sin 2\theta_i = \pm \frac{b}{\sqrt{b_i^2 + (a_i - c_i)^2}} \qquad i=1, 2,\ldots, K \qquad (126)$$

$$\cos 2\theta_i = \pm \frac{a_i - c_i}{\sqrt{b_i^2 + (a_i - c_i)^2}} \qquad i=1, 2,\ldots, K \qquad (127)$$

For objects with $b=0$ and $a=c$, there is no unique axis for the orientation which is found for the minimum value of D^2,

Elongation which is the inverse of the circularity is the largest value of D to the smallest value of D. The maximum and the minimum values of D^2 is determined by the signs of (126) and (127) expressions. The minimum value of elongation is 1 which is the case for a circle:

$$Elongation_i = \frac{1}{Circularity_i} = \frac{D_{\max_i}}{D_{\min_i}} \qquad i=1, 2,\ldots, K \qquad (128)$$

CHAPTER 4

INSPECTION MODE

In the inspection mode, which is performed in real time when the packing machine is operating; the generated blister model, composed of the correction model and tablet features generated in the training mode; and user defined tolerances, is used in order to speed up the process of the images and get a feasible inspection system. When an image to be inspected is acquired, it is segmented after correction for spatial color inhomogeneity, by using the correction and the segmentation models obtained in the training mode. The features are extracted in the same way as defined in the training mode and they are compared with the extracted features of the defect free tablets and the user defined tolerances, to determine the defective tablets.

4.1. SPATIAL NONUNIFORMITY CORRECTION

For every acquired image $\mathbf{v}(\mathbf{x})=[v_1(\mathbf{x}),v_2(\mathbf{x}),v_3(\mathbf{x})]$, the color intensity inhomogeneity of the image components are corrected by using the spatial color nonuniformity correction model, f_{oi}^{-1}, which is generated in the training mode and stored in the blister model.

$$u_{oi}(\mathbf{x}) = f_{oi}^{-1}(\mathbf{x})\, v_i(\mathbf{x}) \tag{129}$$

The uniform color components are merged into a uniform image component $\mathbf{u_o}(\mathbf{x})$. Since the correction model is chosen as M4, a fourth order multiplicative model:

$$u_{oi}(\mathbf{x}) = m_{oi}^{-1}(\mathbf{x})\, v_i(\mathbf{x}) \tag{130}$$

So, for every component of the acquired color image matrix $\boldsymbol{V}(j,\ k) = [V_1(j,\ k),\ V_1(j,\ k),\ V_1(j,\ k)]$, the optimal correction model matrix $M_{oi}(j,k)$ stored in the blister model is used to transform the acquired image components to the components of the optimal uniform color image $\boldsymbol{U_o}(j,\ k) = [U_{o1}(j,\ k),\ U_{o2}(j,\ k),\ U_{o3}(j,\ k)]$ by employing:

$$U_{oi}(j,k) = V(j,k)_i \otimes M_{oi}(j,k) \tag{131}$$

4.2. IMAGE SEGMENTATION

The uniform image component $\boldsymbol{U_o}(j,\ k)$ is segmented directly by using the labeling function that is stored in the blister model after being generated in the training mode like the correction model. A segmented binary image $BW(j,\ k)$ is obtained, whose pixels with value 0 corresponds to the tablet and, pixels with value 1 corresponds to the background for the blisters containing tablets of one color:

$$BW(j,k) = L(\boldsymbol{U}_o(j,k)) \qquad \forall (j,k) \in X \tag{132}$$

where, X is the spatial image domain, RGB or HSV.

4.3. DEFECTIVE TABLET DETECTION

Unlike the correction and the segmentation steps, no blister model information is used for the feature extraction in the inspection mode. Rather, the blister model including the feature values extracted in the training mode and the user defined tolerances for these features and the surface defect size are used to be compared with the currently obtained feature values, in order to detect the defective tablets. The feature values stored in the blister model to be considered to determine the defects are:

i. Position of each tablet
ii. Size of each tablet
iii. Shape of each tablet
iv. Size of the surface defects

For each tablet position in the model, the nearest region in the inspected image is found. There are three cases for a tablet that it is said to *fail*; the tablet may be “missing”, or “defective”, or it may have a “surface defect” detected by the procedures:

i. A tablet is marked as “missing” if the centre of the region lies out of the position value stored in the model.
ii. If the size and shape of the region, labeled as a tablet in the inspection mode is larger than the size and shape tolerance values compared to the corresponding tablet stored in the blister model; it is marked as “defective”.
iii. A tablet is said to have a “surface defect” if the size of the region labeled as background inside the region corresponding to a tablet is greater than the tolerance value.

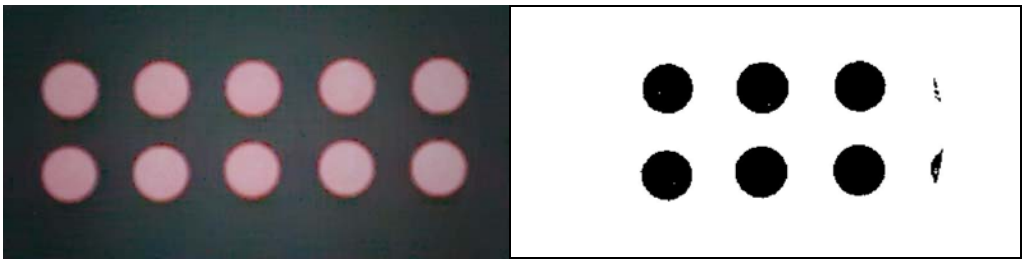
If a tablet does not fail, it is said to *pass.* All blisters containing any tablet that fails are determined in order to produce a signal in the inspection system to eject these blisters.

CHAPTER 5

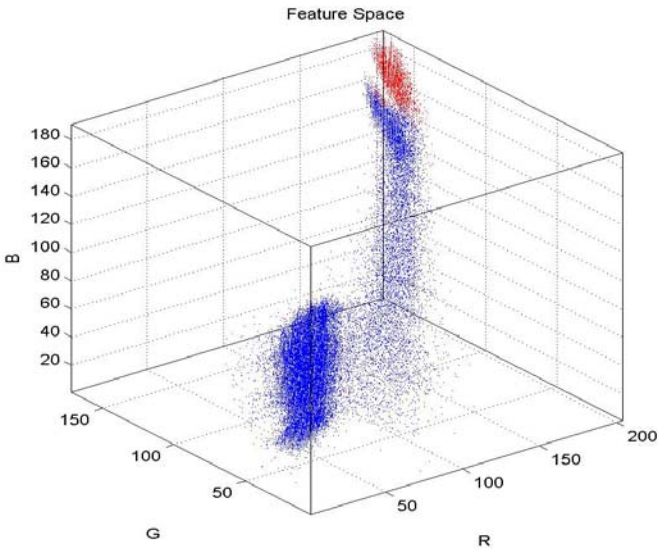
EXPERIMENTS

In order to emphasize the importance of the spatial color nonuniformity correction, a defect free image (Figure 5.1.a.) with high spatial color nonuniformity, was acquired in the training mode. Using the GUI, optimization algorithm for the correction was selected as Powell; color space was selected as RGB for the segmentation step which was also selected as MaxShift. A point was selected on any of the tablets as the starting point of the MaxShift algorithm used in the segmentation mode. The blister borders; the tolerances for the tablet features, position, size and shape; and the tolerance for the size of the surface defect were also set by the user, prior to the training mode process.

The clusters in the feature space corresponding to the probability density distribution which can also be named as 3D histogram, of the nonuniform image (Figure 5.1.c) are not as compact as the clusters (Figure 5.1.f) formed for the spatially uniform image (Figure 5.1.d). Since segmentation process aims to find a boundary in the 3D histogram space, its efficiency is proportional with the compactness of the clusters. If the image is segmented without correcting the color intensity inhomogeneity, this process is adversely affected by the overlapping of the clusters of the acquired image and results in a binary image, from which tablet features can not be extracted. Reduction of the complexity and the overlapping of the clusters results in an efficient segmentation which uses a cube-shaped kernel of size 21x21x21 as can be seen from Figure 5.1.e.

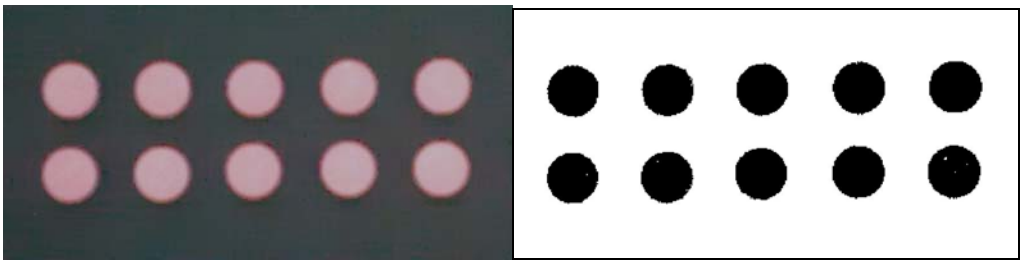


(a) (b)

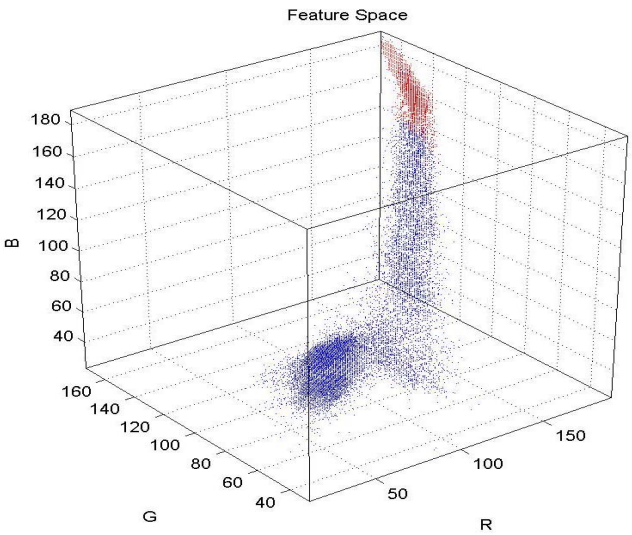


(c)

Figure 5.1. (a) The acquired defect free image of pink tablets with spatial color nonuniformity, in training mode (b) the segmented image (c) its 3D histogram, red points labeled as “tablet” and the blue points labeled as “background”



(d) (e)



(f)

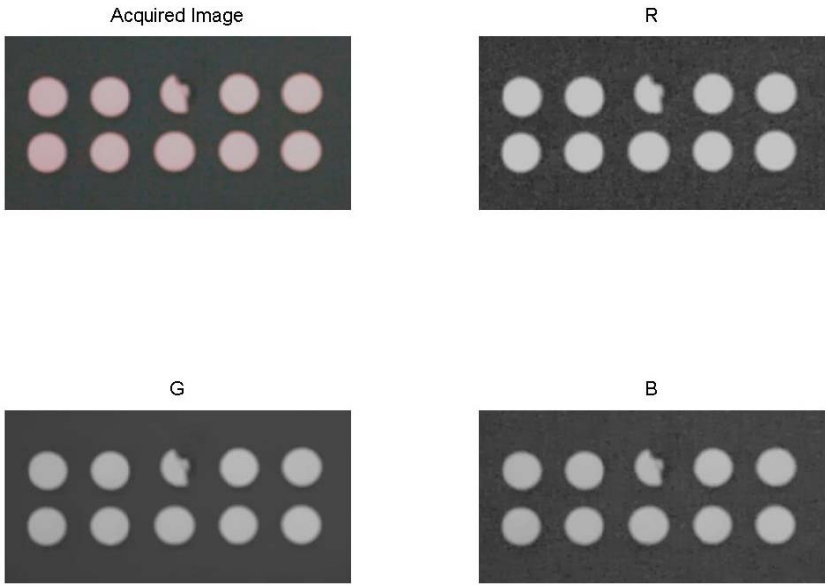
Figure 5.1.(continued) (d) The spatially color uniform image of pink tablets (e) the segmented image (f) Its 3D histogram, red points labeled as “tablet” and the blue points labeled as “background”

After capturing the defect free image in the training mode, its color nonuniformity was corrected by a robust method, which requires no input argument and produces the correction model to be stored in the blister model. The minimization algorithm to be used in the correction step was firstly selected as Powell’s method, and then selected as Downhill Simplex method and their performances were compared. Considering that the aim of these methods were to minimize the Shannon information of the estimated

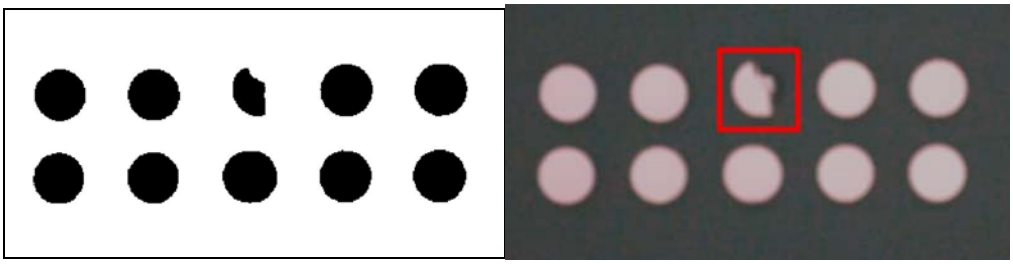
image, Powell's method reduced the information more than Downhill Simplex method did in much shorter time as expected. So, Powell's method yielded more uniform images in a smaller process time, since a simplex with fifteen vertices was required for the Downhill Simplex method which in turn resulted in about forty five minutes on the average. The corrected image was then segmented by the selection of a point on a tablet by the operator, which in turn determines the tablet cluster. A labeling function was generated by the segmentation process in the training mode, defining a 3D boundary between the tablet and the background clusters, in the feature space to be used in the inspection mode. The binary image of the defect free tablet obtained by the segmentation was used to describe the feature values; position, size and shape for each tablet according to their position. At the end of the training mode, the blister model was totally formed to speed up the inspection mode for which time is one of the major parameters determining the feasibility of inspection systems.

The following inspection mode experiments, whose results are shown in Figure 5.2.a-c and Figure 5.3.a-c, were achieved using the model generated for the acquired defect free image.

A broken tablet was captured (Figure5.2.a), segmented (Figure5.2.b), the defective tablets were found successfully by the employment of the algorithm and the tablets with defects exceeding the user defined tolerances were marked (Figure5.2.c).



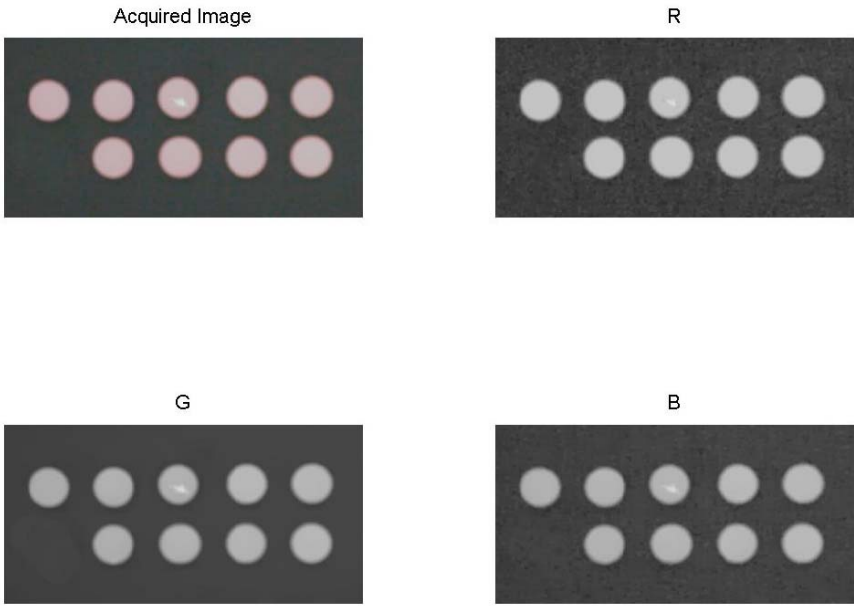
(a)



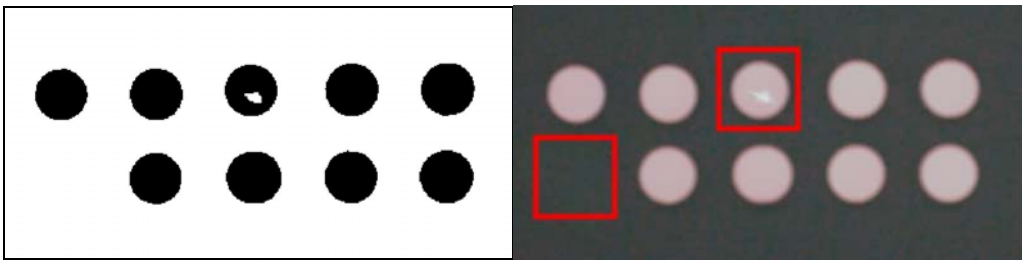
(b) (c)

Figure 5.2. (a) The acquired image with one broken tablet and its components in the inspection mode (b) The segmented image (c) The acquired image with marked defects

An image containing one missing tablet and one tablet with surface defect was captured in the following experiment (Figure5.3.a), segmented (Figure5.3.b) and the defective tablets are marked (Figure5.3.c).



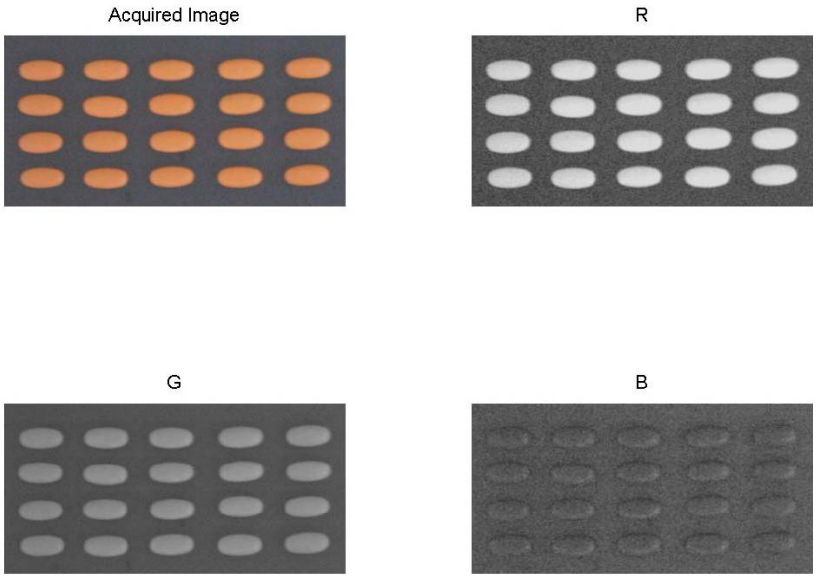
(a)



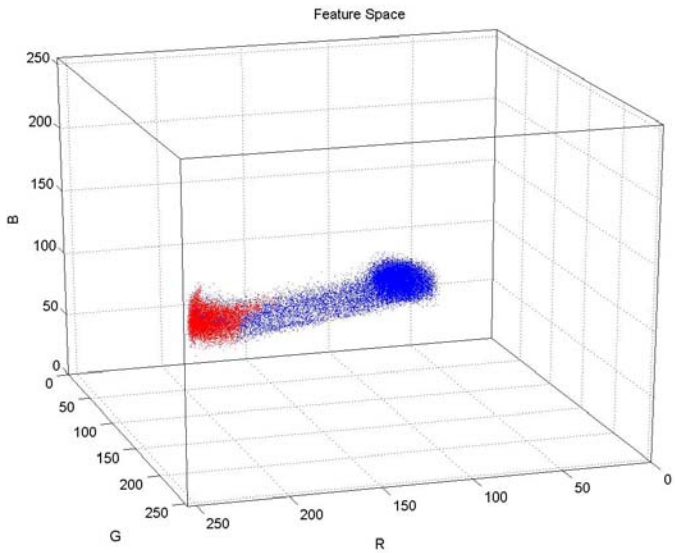
(b) (c)

Figure 5.3. (a) The acquired image with one missing tablet and a tablet having a surface defect; and its components in the inspection mode (b)Tthe segmented image (c) The acquired image with marked defects

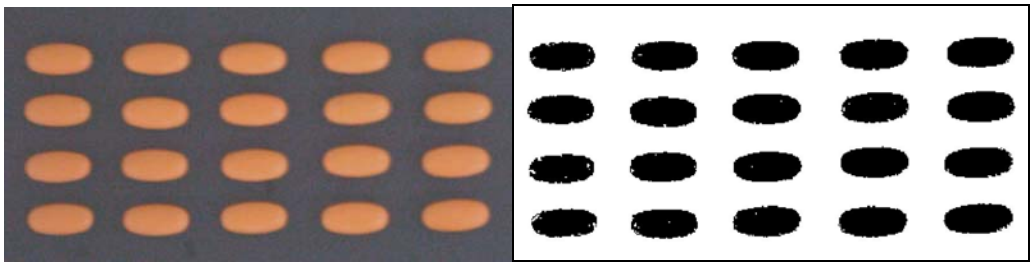
The results of the training mode and the inspection mode employed for orange tablets are presented also through Figure5.4. - Figure5.8. A defect free image was acquired in the training mode and the blister model is formed.



(a)



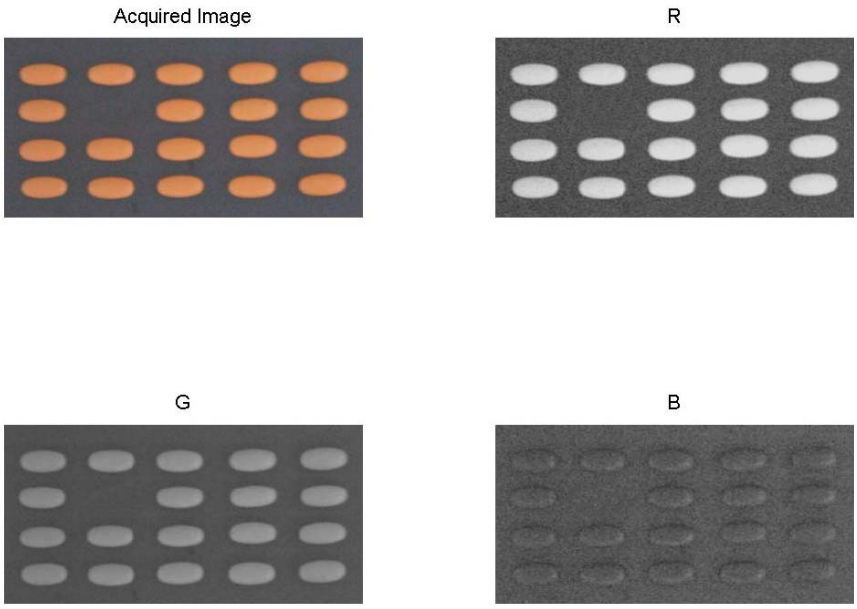
(b)



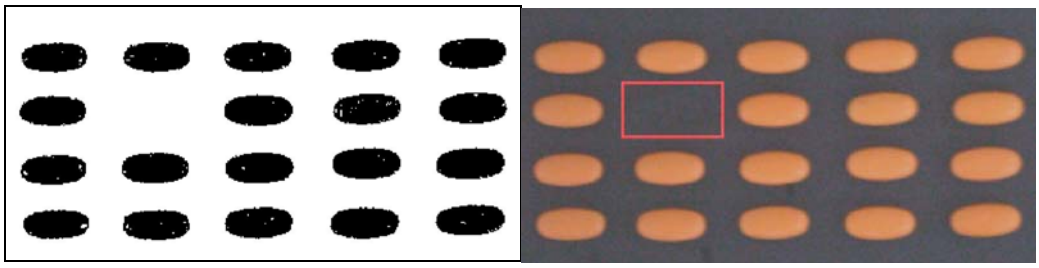
(c) (d)

Figure 5.4. (a) The acquired defect free image of orange tablets with spatial color nonuniformity, in training mode and its components (b) Its 3D histogram, red points labeled as “tablet” and the blue points labeled as “background” (c) The acquired image compared with (d) the segmented image after correction.

An image containing one missing tablet was captured in the following experiment and the defective tablets were detected by utilizing the inspection mode (Figure5.5.a-c).



(a)



(b)

Figure 5.5. (a) The acquired image with one missing tablet and its components in the inspection mode (b) The segmented image (c) The acquired image with marked defects

One broken tablet and a tablet with surface defect are marked in Figure 5.6. and Figure 5.7., respectively, while three tablets are marked in Figure 5.8.; one broken, one missing and one with a surface defect.

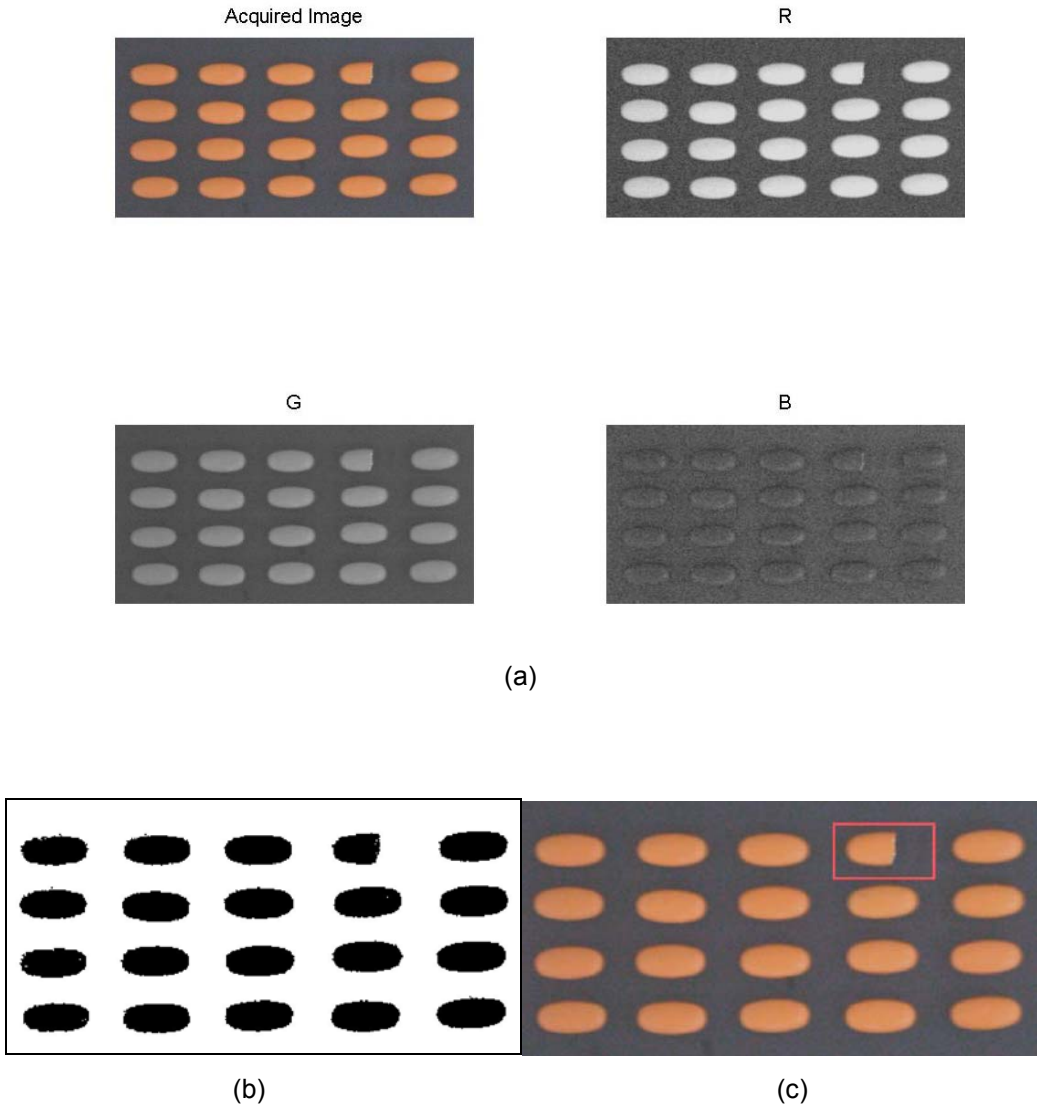
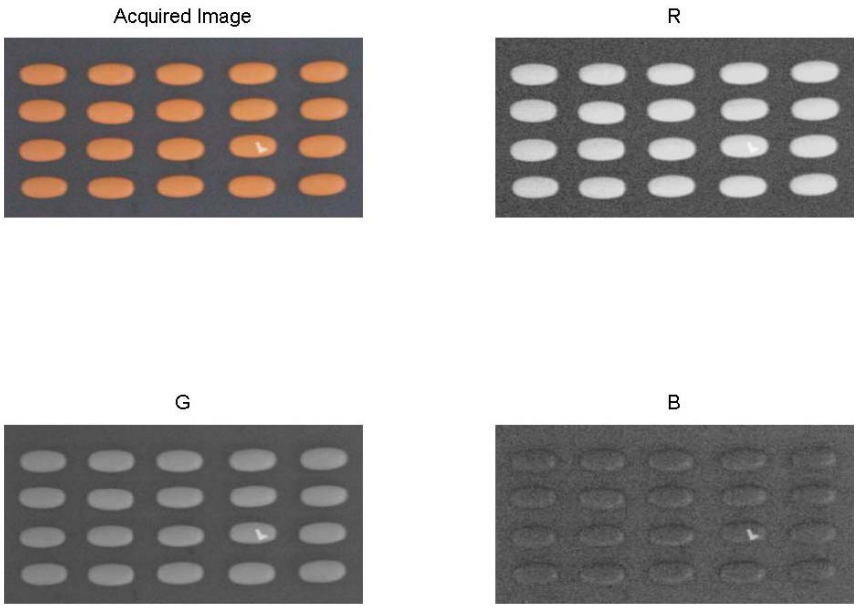
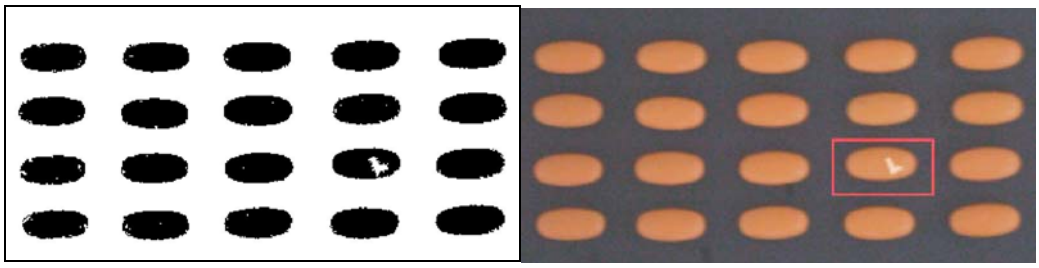


Figure 5.6. (a) The acquired image with one broken tablet and its components in the inspection mode (b) The segmented image (c) The acquired image with marked defects

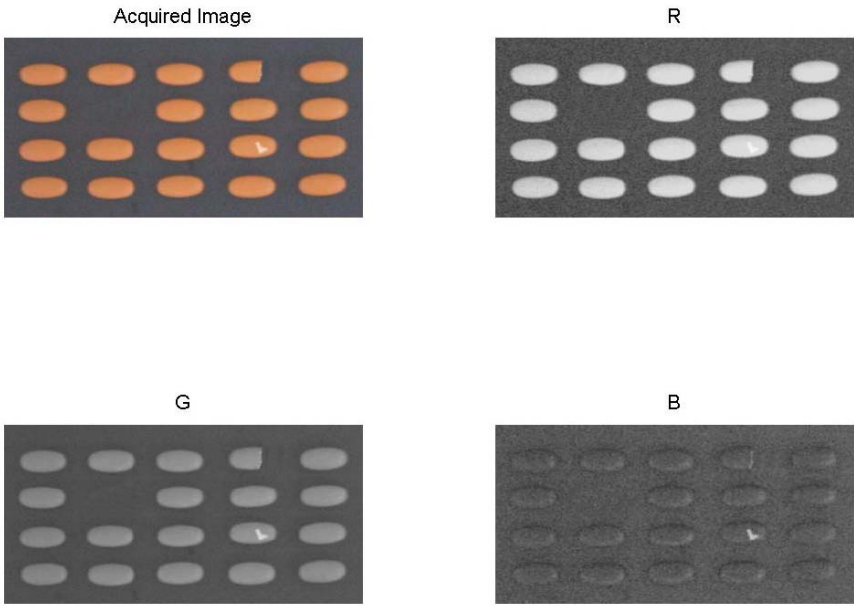


(a)

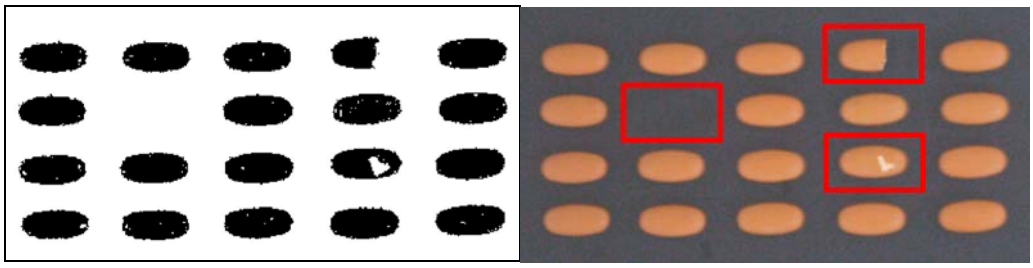


(b) (c)

Figure 5.7. (a) The acquired image with one tablet with a surface defect and its components in the inspection mode (b) The segmented image (c) The acquired image with marked defects



(a)



(b) (c)

Figure 5.8. (a) The acquired image with a broken tablet, a missing tablet and one tablet with a surface defect and its components in the inspection mode (b) The segmented image (c) The acquired image with marked defects

The pink and orange tablet images were used for every preference case, the results for a few defective images are presented in Figure 5.9. through Figure 5.14.

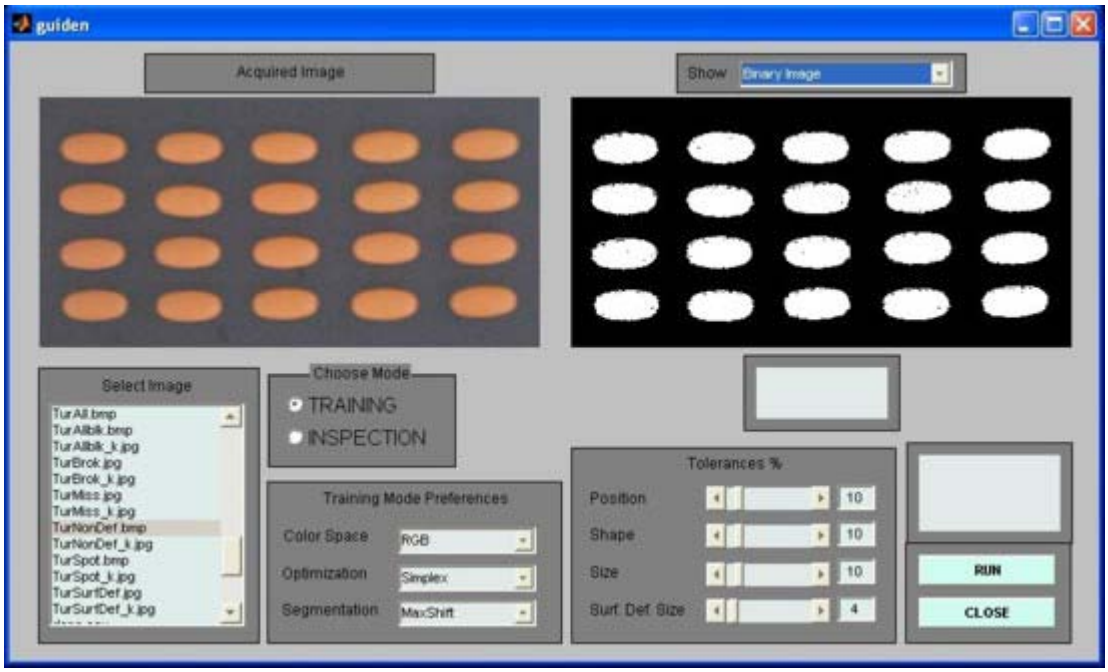


Figure 5.9. Training mode result of a defect free image using MaxShift algorithm in RGB color space

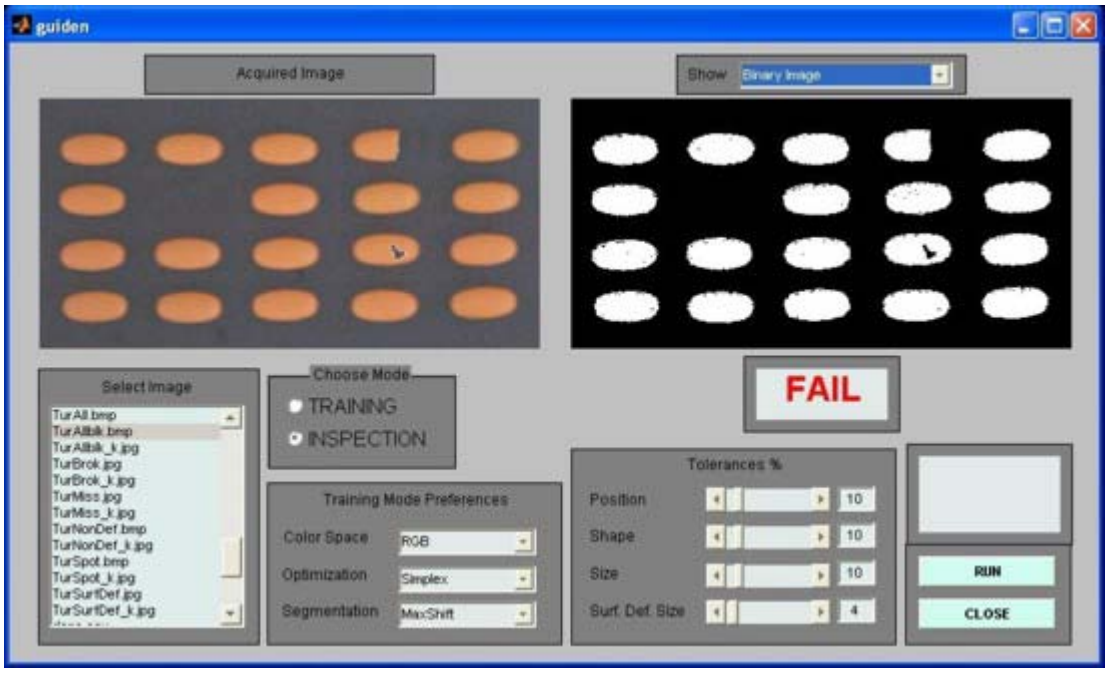


Figure 5.10. Inspection mode result of a blister of defected tablets using MaxShift algorithm in RGB color space

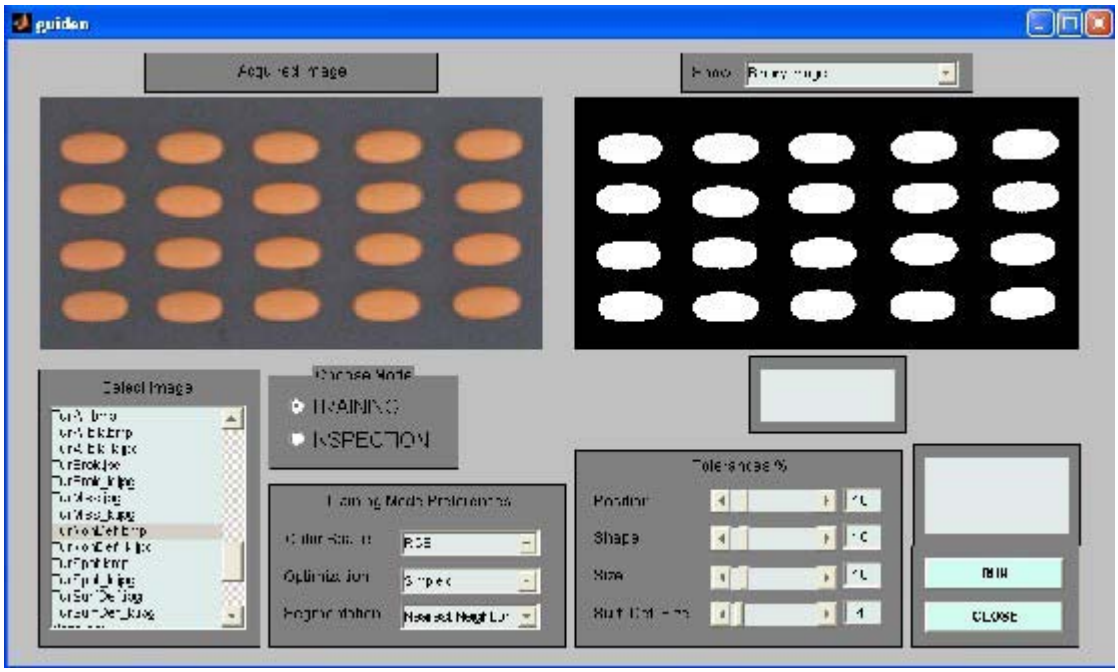


Figure 5.11. Training mode result of a defect free image using Nearest Neighbor algorithm in RGB color space

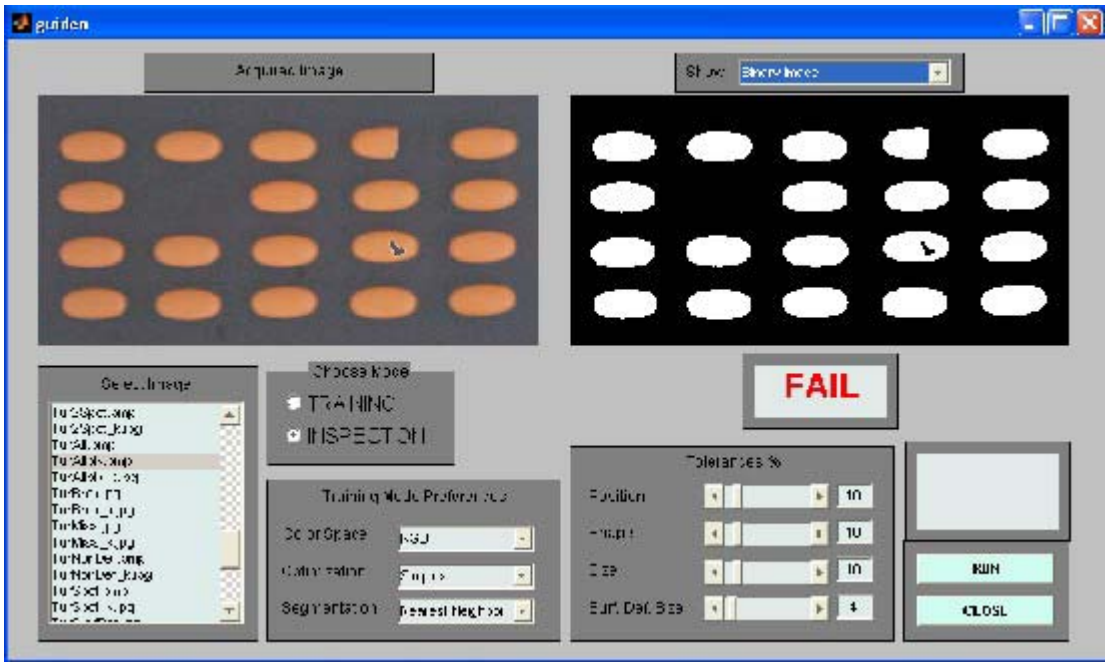


Figure 5.12. Inspection mode result of a blister of defected tablets using Nearest Neighbor algorithm in RGB color space

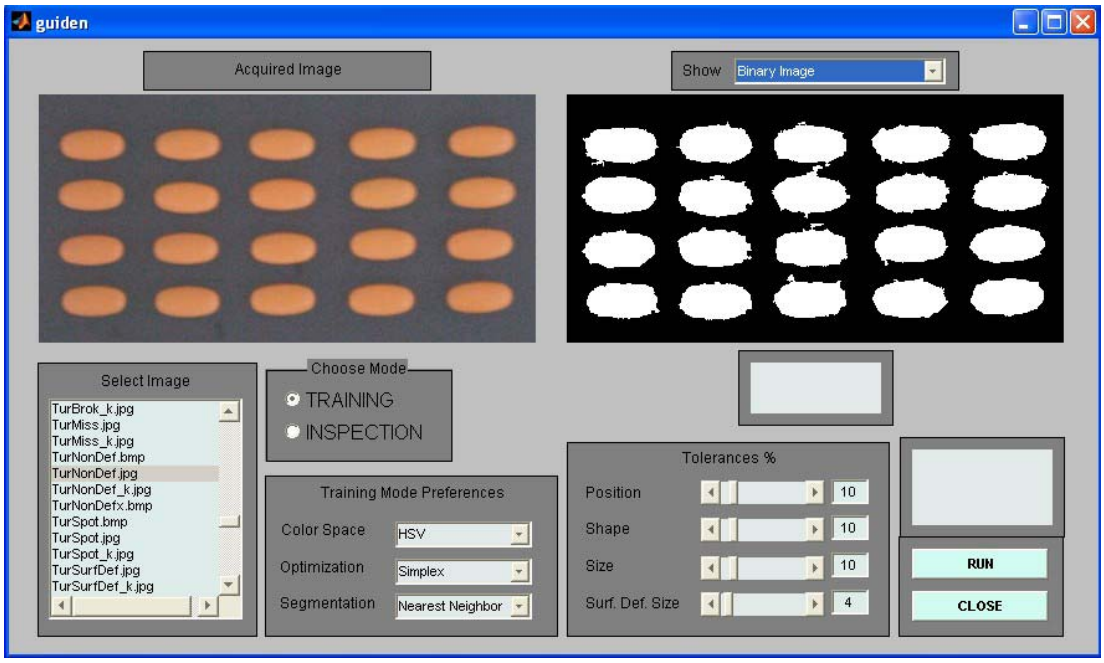


Figure 5.13. Training mode result of a defect free image using Nearest Neighbor algorithm in HSV color space

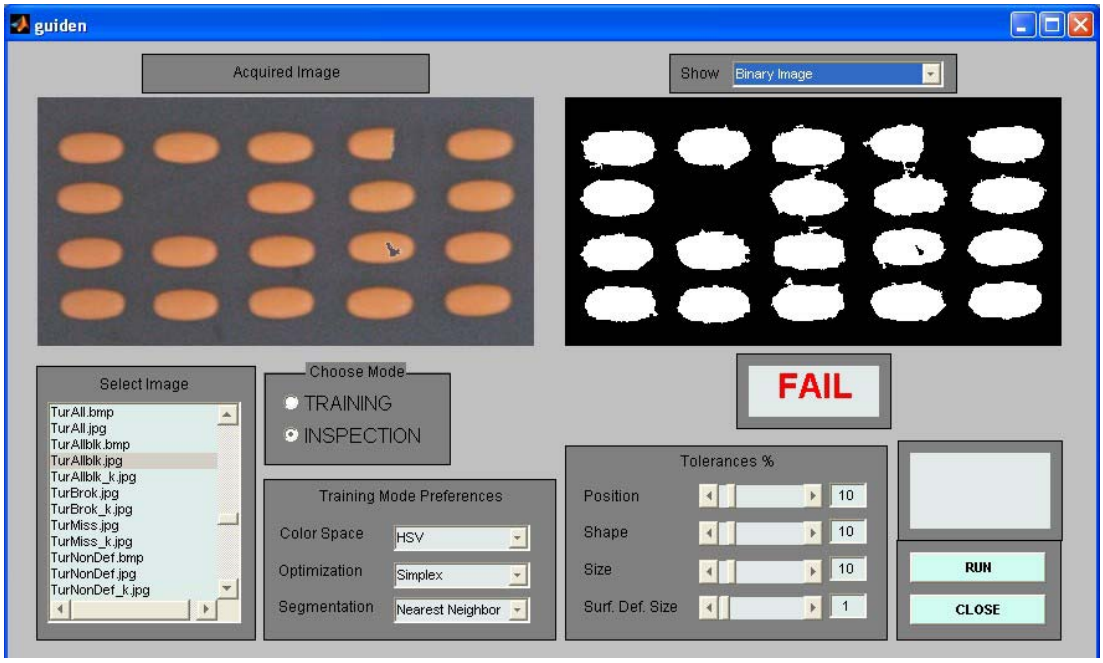


Figure 5.14. Inspection mode result of a blister of defected tablets using Nearest Neighbor algorithm in HSV color space

As well as the binary image; corrected image, connected component labeled image or the acquired image with the defects marked can also be viewed on the GUI according to the selection.

After the pink and orange tablet images were used for every preference case, the implemented program was applied to other types of tablets with different, shape, color and sizes in various illumination circumstances, with different background whose colors. All combinations of selections were used for each individual experiment. Optimization algorithm for the spatial color nonuniformity correction was selected as Powell or Downhill Simplex method; color space was selected as RGB or HSV for the segmentation step for which MaxShift or Nearest Neighbor method was preferred.

For each circumstance defined by the tablet properties, the environment and the algorithm preference case, firstly a defect free tablet was captured in the training mode. In the inspection mode, nine types of tablet sets composed of 100 different tablet images were used; each set was containing all types of defects. The tablet sets were including different sized tablets; different shaped tablets such as broken tablets; tablets with surface defects, and missing tablets. All the user defined tolerances were set to 20% for all the sets. The error performance is closely related to the tolerances. For example for surface defects, there is a compromise between the detectable defect size and the tolerance for the surface defect; the system is defined reliable for defect sizes greater than the tolerance selected for the surface defect. The results for some of the images are presented in Figure 5.15.. The resultant process time and the error performance tables are Table 5.1. and Table 5.2., respectively.

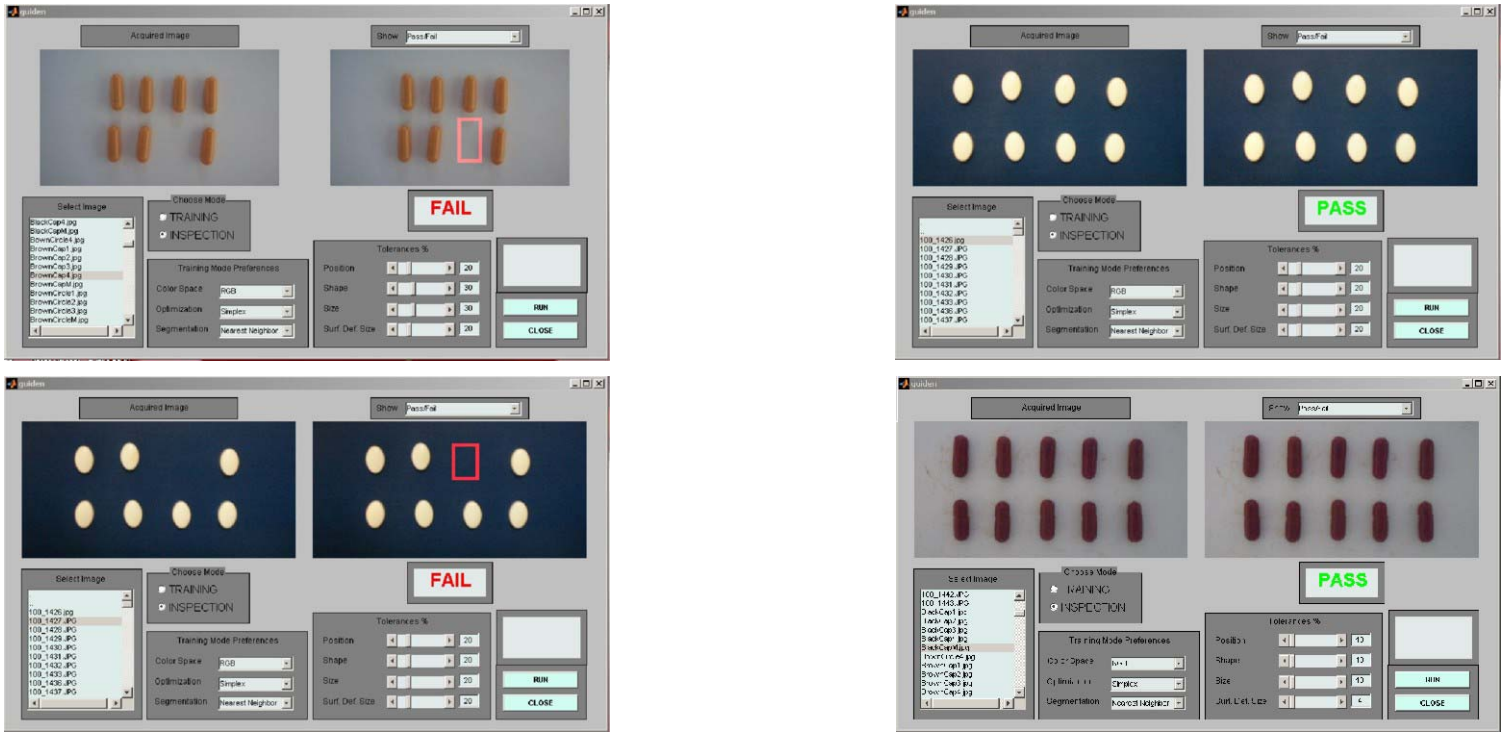


Figure 5.15. Inspection mode results for tablet images

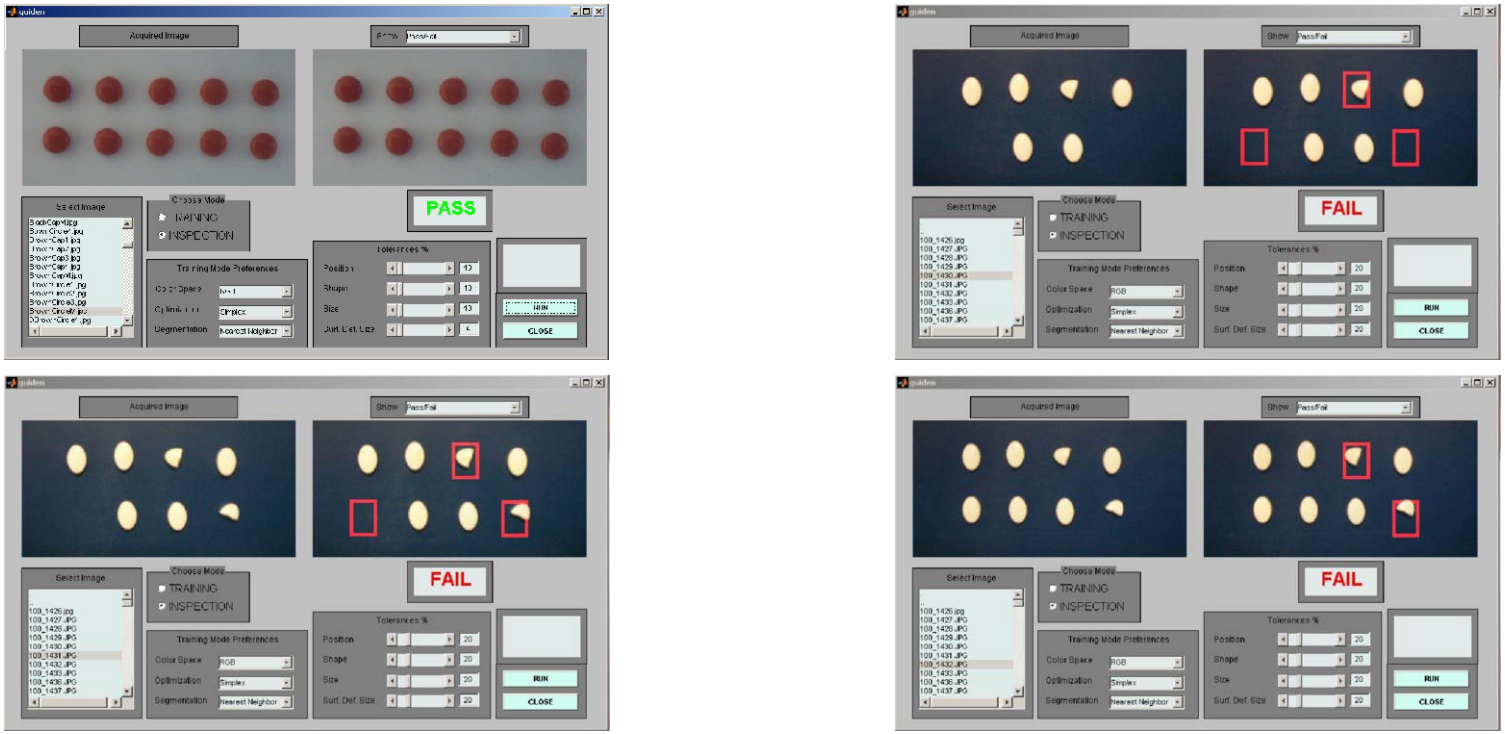


Figure 5.15. (continued) Inspection mode results for tablet images

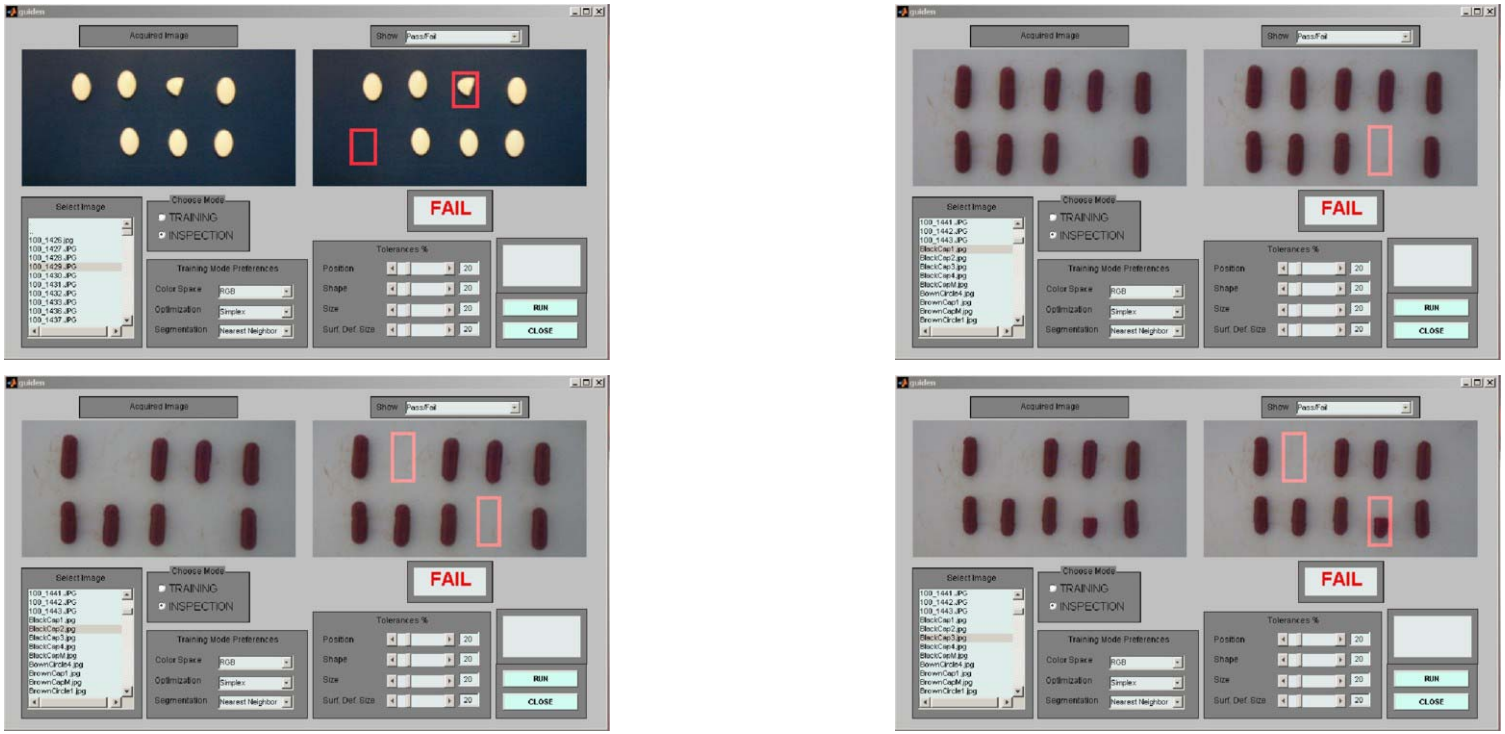


Figure 5.15. (continued) Inspection mode results for tablet images

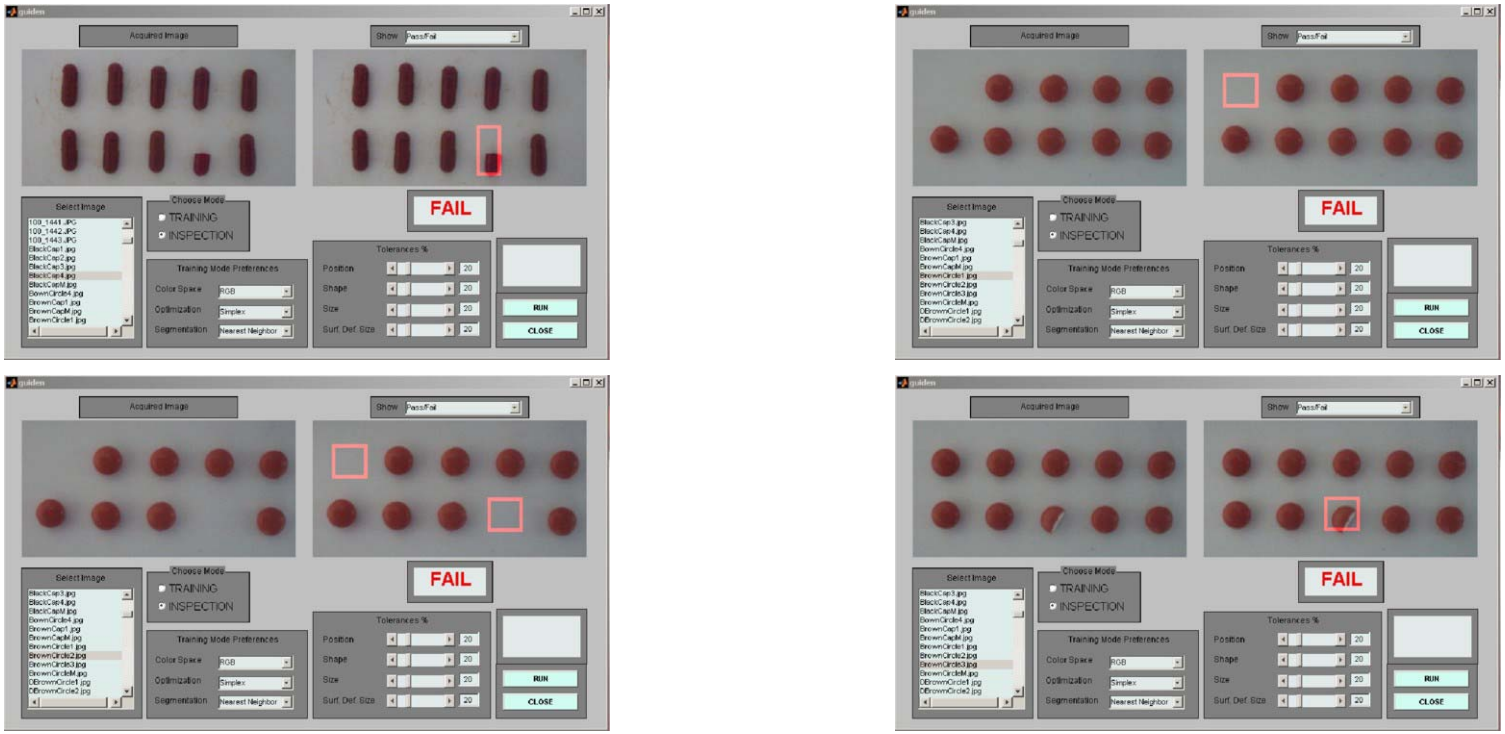


Figure 5.15. (continued) Inspection mode results for tablet images

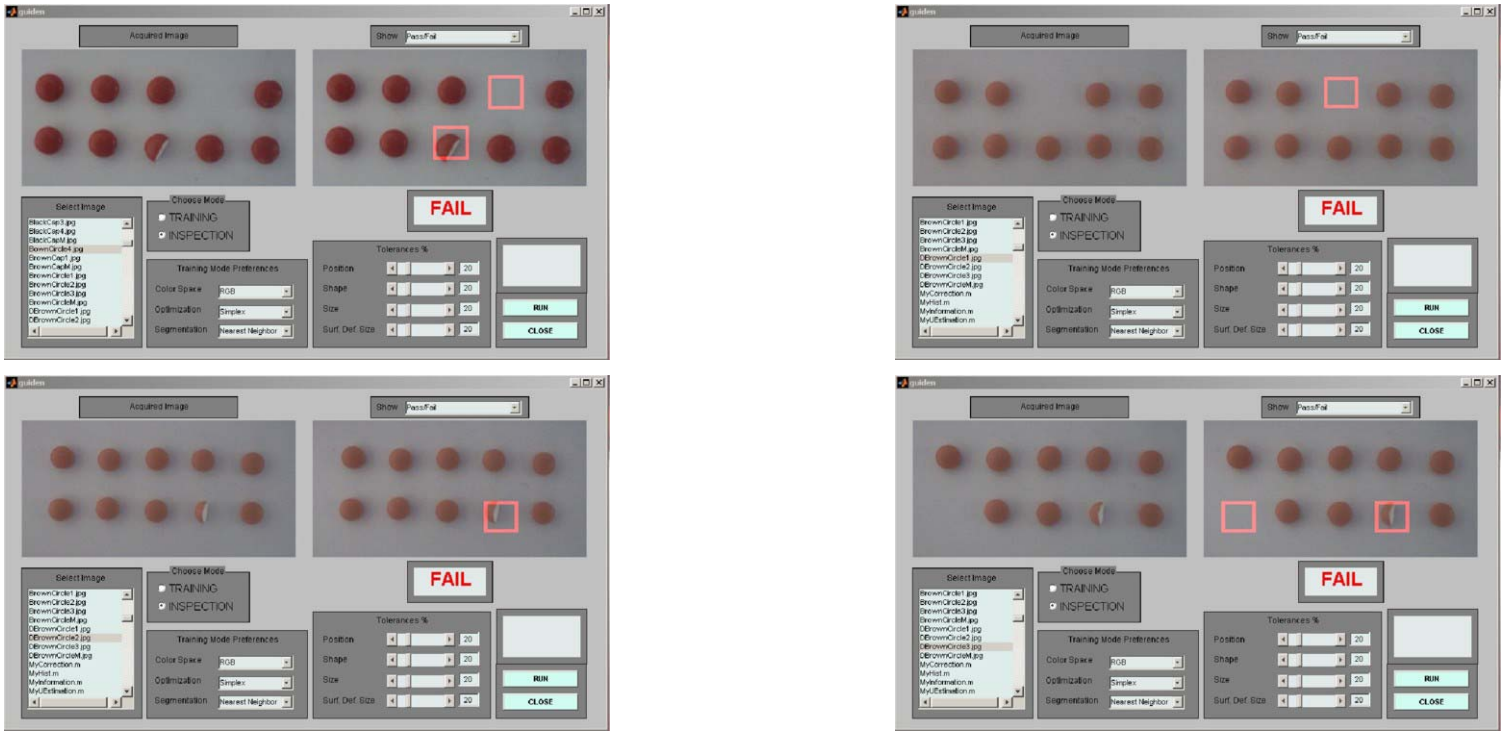


Figure 5.15. (continued) Inspection mode results for tablet images

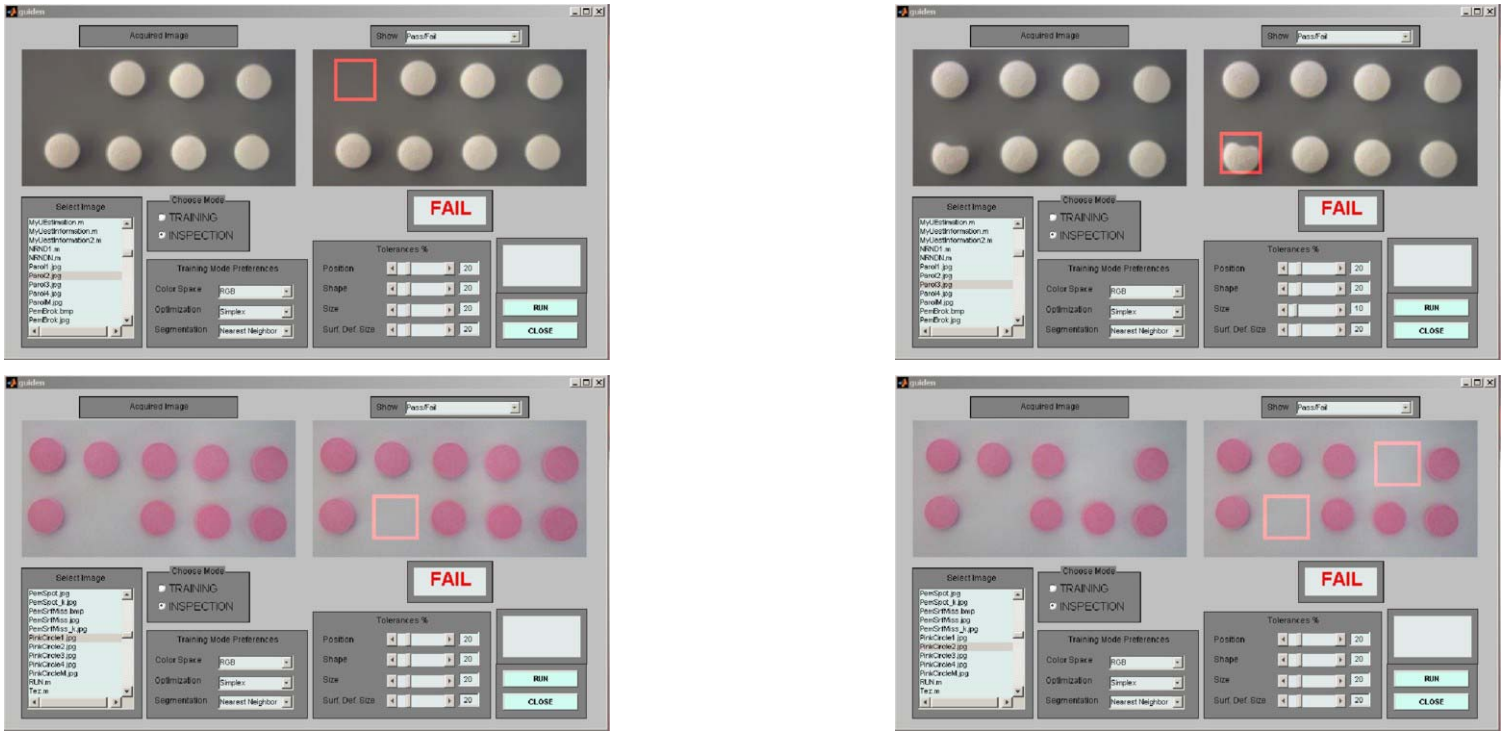


Figure 5.15. (continued) Inspection mode results for tablet images

Figure 5.15. (continued) Inspection mode results for tablet images

Table 5.1. Process time comparison table for different algorithms using 100 sample images

TOLERANCES				COLOR SPACE	SEGMENTATION METHOD	TRAINING MODE PROCESS TIME (ms)				INSPECTION MODE PROCESS TIME (ms)			
POSITION (%)	SHAPE (%)	SIZE (%)	SURFACE DEFFECT (%)			CORRECTION	SEGMENTATION	FEATURE EXTRACTION	TOTAL TIME	CORRECTION	SEGMENTATION	DEFECT DETECTION	TOTAL TIME
20	20	20	20	RGB	MAXSHIFT	13512	375956	1537	391005	49	65	785	899
20	20	20	20	RGB	NEAREST NEIGHBOR	13024	813	1073	14910	43	74	826	943
20	20	20	20	HSV	NEAREST NEIGHBOR	12790	944	1259	14993	46	236	949	1231

Table 5.2. Error Performance table for different algorithms using 100 sample images

TOLERANCES				COLOR SPACE	SEGMENTATION METHOD	Error Performance Percentage (%)
POSITION (%)	SHAPE (%)	SIZE (%)	SURFACE DEFFECT (%)			
20	20	20	20	RGB	MAXSHIFT	88
20	20	20	20	RGB	NEAREST NEIGHBOR	86
20	20	20	20	HSV	NEAREST NEIGHBOR	67

In order to show that the implemented program is applicable to other types of industrial products, the inspection result for a screw image is presented as an example in Figure 5.16..

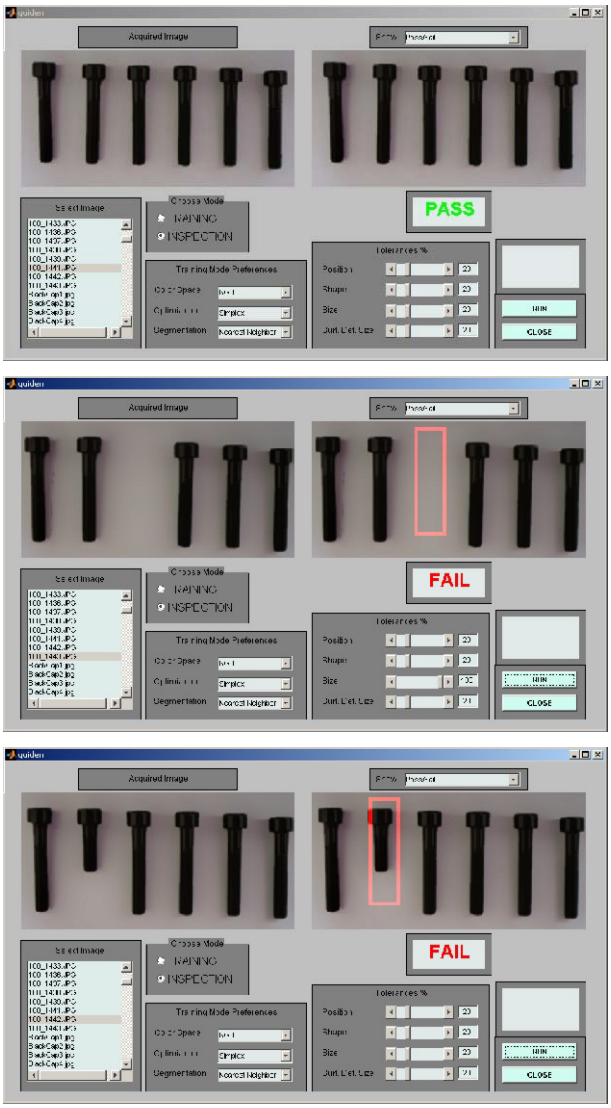


Figure 5.16. Inspection mode results for screw image

CHAPTER 6

CONCLUSIONS

Different algorithms were used and compared for the segmentation of pharmaceutical color tablets. The nonparametric clustering based methods Nearest Neighbor and MaxShift, which is a kernel estimation algorithm, were implemented in RGB and HSV color spaces. For the multidimensional optimization required in the spatial color nonuniformity correction step, Powell's method or Downhill Simplex method was selected. The model for the three main steps of the method: spatial color nonuniformity correction, nonparametric clustering based segmentation and feature extraction, are generated in the training mode without a time limitation. The model is directly used in the inspection mode, speeding up this mode where time is one of the major criterions that determine if a proposed algorithm is feasible or not.

In this thesis, area, projection, orientation, circularity, major and minor axis lengths implying the shape were used as the shape information, yielding a sufficient discrimination.

For the segmentation step to produce correct binary images, the clusters in the probability density feature space must be as discrete and dense as possible for both of the segmentation methods, since both are nonparametric clustering based algorithms. RGB color space, which includes both the color and the intensity information, is used without a transformation to the other color spaces that include only the color information; so it has more complex feature space than the color spaces containing only the color information like

HSV. Because of the stated reason it is more crucial to make the clusters discriminative prior to the segmentation especially for RGB color space. High contrast and spatial color uniformity are the required conditions to provide a more compact feature space. For the multivariate minimization required by the spatial color nonuniformity correction, two methods that do not require gradient calculation were used; Powell's method and Downhill Simplex method. Powell's method yielded more uniform images in a smaller process time, because Downhill Simplex method is extremely slow for higher order number of variables, which was fourteen in this work. Downhill Simplex method required a simplex with fifteen vertices which in turn resulted in very long process time and low minimization performance.

High contrast is highly related with the illumination system. Since the background is generally reflective and the tablets have angled surfaces, the continuous diffuse illumination, which minimizes shadows and specular reflections by providing a uniform lighting of 170° solid angle, obtained by on-axis and off-axis light sources to match the geometry of the tablets, should be used. In order to increase the difference between the reflections of the light on the different colors on the tablet and between the background; the ideal way is to change the wavelength of the light depending on the colors, which is achieved by using color filters, or changing the light source, or using a programmable light source. Even when using good illumination hardware, there may be still spatial color nonuniformities that should be corrected to gain a more compact 3D histogram, that can be effectively obtained by the correction algorithm used in this thesis.

If a controlled environment with a constant illumination can not be implemented perfectly, the detection performance becomes low when the color nonuniformity for the inspection mode and the training modes differed much, because the nonuniformities of the inspection mode images are corrected according to the illumination model generated in the training mode.

If the clusters are still highly overlapping after the correction step, which is the case for the images with high noise or the tablets that are highly textured; the color information is not sufficient to extract the features. Although for tablet inspection implemented using a controlled environment, high noise is not encountered, additional features are needed in such cases. For example, spatial noise removal algorithms may be used using features related with the noise; or additional features for the texture can be used, in order to extend the feature space.

Since the statistical segmentation approach is very effective even when the clusters in the feature space are not compact; two statistical nonparametric clustering based algorithms, MaxShift (kernel estimation) and Nearest Neighbor method were used. The MaxShift algorithm was derived in [5] from the MeanShift algorithm [3]. Since for tablet inspection, the number of main colors encountered is small, where clusters in the probability density feature space are discriminative assuming spatial color uniformity; the usage of MaxShift rather than MeanShift, has some advantages. MaxShift algorithm keeps the robustness, accurateness and repeatability properties of MeanShift algorithm; while it is two times faster, and it avoids oscillations in the vicinity of the mode and avoids trapping in local maxima. Although more compact clusters can be obtained by transforming the RGB image to the other color spaces such as HSV, that separate color and intensity values; color features become unstable at low intensity values and it is a time consuming process. Using Nearest Neighbor algorithm, process time of the inspection mode for HSV color space is longer than the process time of the inspection mode for RGB color space, because of the color space transformation. The error performance for the HSV color space case is lower than the error performance for the RGB color space case, since converting from RGB color space, information is lost; hue is not reliable for low intensity values. Using RGB color space, the inspection mode process time performances for MaxShift and Nearest Neighbor algorithms are similar since correction and feature extraction steps are common and in the segmentation step a labeling function obtained at the training mode is used in both. On the other hand,

process time of the training mode of the MaxShift method is much longer than the Nearest Neighbor method, but time is not critical in the training mode, since this mode is executed once for each tablet set. Error performance of MaxShift method using RGB color space is better than the error performance of Nearest Neighbor method using RGB color space, while the error performance of Nearest Neighbor method using HSV color space is the worst. As a result, segmentation in RGB color space using MaxShift method performed best considering the process time in the inspection mode and the error performance, while having the longest process time in the training mode.

The extension of the segmentation algorithm to multicolored tablets like capsules, as a future work, can be achieved by the operator selecting a point on each color occurring on the tablet in the training mode, resulting in more number of clusters than two, which is the case for one-colored tablets; one for the background and each of the remaining corresponding to each color that the tablet has on it. This modification makes the software implemented to be used in a much wider range, since multi-colored tablets are frequently encountered in pharmaceutics.

The implemented program with GUI is applicable to other types of industrial products and so it can be used as a general purpose real time automated visual inspection software.

REFERENCES

[1] Bentley JL. Multidimensional binary search trees used for associative searching. Communications of the ACM, 18(9):509–517, September 1975.

[2] Cheng HD, Jiang XH, Sun Y, Wang JL. Color Image segmentation: advances and prospects. Pattern Recognition 2001; 34: 2259-81

[3] Comaniciu D, Meer P. Distribution free decomposition of multivariate data. Pattern Analysis and Applications 1999; 2: 22–30.

[4] Dasarathy B. Nearest Neighbor Pattern Classification Techniques. IEEE Computer Society Press, 1990.

[5] Derganc J, Likar B, Bernard R, Tomaževič D, Pernuš F. Real-time automated visual inspection of color tablets in pharmaceutical blisters. Science Direct, Real-time Imaging 9, 2003; 113-124.

[6] Fukunaga K, Narendra PM. A branch and bound algorithm for computing k-nearest neighbors. IEEE Transactions on Computers, pages 750–753, July 1975.

[7] Greenspan M, Godin G, Talbot J. Acceleration of binning nearest neighbor methods. Vision Interface 2000, pages 337–344, Montreal, Canada, May 14-17 2000.

[8] Healey GE. Segmenting images using normalized color. IEEE Transactions on Systems, Man and Cybernetics 1992; 22: 64–73.

[9] Jain R, Kasturi R, Schunk B. Machine vision. New York: McGraw-Hill Inc, 1995.

[10] Likar B, Viergever MA, Pernuš F. Retrospective correction of MR intensity inhomogeneity by information minimization. IEEE Transactions on Medical Imaging 2001; 20: 1398–410.

[11] Maxwell BA, Shafer SA. Segmentation and interpretation of multicolored objects with highlights. Computer Vision and Image Understanding 2000; 77 : 1-24

[12] Nelder JA, Mead R. Downhill Simplex Method. 1965, Computer Journal 1965; 7 : 308

[13] Pal NR, Pal SK. A review on image segmentation techniques. Pattern Recognition 1993; 26: 1277-94

[14] Park SH, Yun ID, Lee SU. Color image segmentation based on 3-D clustering: morphological approach. Pattern Recognition 1998; 31: 1061-76

[15] Press WH, Flannery BP, Teukolosky SA, Vetterling WT. Numerical Recepies in C. 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[16] Rivest RL. On the optimality of Elias's algorithm for performing best-match searches. In Information Processing 74, pages 678–681, 1974.

[17] Ruiz EV. An algorithm for finding nearest neighbors in (approximately) constant time. Pattern Recognition Letters, 4:145–157, July 1986.

[18] Shakhnarovish, Darrell, Indyk. Nearest-Neighbor Methods in Learning and Vision. The MIT Press, 2005.

[19] Tremeau A, Borel N. A region growing and merging algorithm to color segmentation. Pattern Recognition 1997; 30: 1191-203

[20] Zugaj D, Lattuati V. A new approach of color images segmentation based on fusing region and edge segmentation outputs. Pattern Recognition 1998; 31: 105-13