DISHWARE IDENTIFICATION AND INSPECTION

FOR AUTOMATIC DISHWASHING OPERATIONS

By

TRUNG HUY DUONG

Bachelor of Technology in Mechatronics

Hanoi University of Technology

Hanoi, Vietnam

2004

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2009

DISHWARE IDENTIFICATION AND INSPECTION FOR AUTOMATIC DISHWASHING OPERATIONS

Thesis Approved:

Dr. Lawrence L. Hoberock

Thesis Adviser

Dr. Jay C. Hanan

Dr. Guoliang Fan

Dr. A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to Dr. Lawrence L. Hoberock, my advisor, who has inspired and directed me in the fascinating research area of machine vision, always encouraging me to do my best, giving me a lot of flexibility, and generously providing me financial support. I would like to express my sincere thanks to Dr. Jay C. Hanan and Dr. Guoliang Fan for serving on my Graduate Committee.

I wish to express my special thanks to Ministry of Education and Training of Vietnam (MOET- www.moet.gov.vn) for supporting a 2-year full scholarship and to School of Mechanical and Aerospace Engineering, Oklahoma State University for supporting a research assistantship to complete this work. I would also thank to University of Communication and Transport for providing me an opportunity to study in USA.

I would like to extend my thanks to Mr. Venugopal Lolla, Mr. Lertrit Sabpipatana, Mr. Venkatesh Akella, my colleagues at Robotics Research Lab in Oklahoma State University, for their valuable ideas and discussions. Finally, I would like to thank my parents Mr. Lung H. Duong and Ms. Luc T. Vu for their love and encouragement. I specially express my appreciation for my wife, Ms. Ly H. Doan, for her love, understanding, and helping me overcome difficulties. I would like to dedicate my work to my daughter, Duong T. Giang, who is my joy, inspiration, and future.

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Background

The worldwide industrial vision market has recently experienced considerable growth. The international Vision 2008 Conference, held November 2008 in Stuttgart, Germany, attracted 282 exhibitors, the highest ever for this annual meeting and 31% more than the previous year (IMVE 2008a). Basler Co. reported incoming orders have increased 20 percent compared to the same period last year and reached €17.4m ($21.8m USD). Stemmer Imaging Group, including Firstsight Vision, Stemmer Imaging in Germany and Switzerland, and Imasys in France, announced total group revenue of approximately €43m ($54m USD) with an overall growth of 8.8 percent from the previous year. Instrument Systems also reported its biggest annual revenue growth since 1986, totaling €9.3m ($11.7m USD), which was a 22 percent increase over 2007 revenues (IMVE 2008b). Automation Tooling Systems Inc. announced its third quarter 2008 revenue of $221.7m, an increase of 27 percent over the same period in 2007 (AIA 2009b). These revenue increases are likely due in part to increases in automated production processes, which often require automated pattern recognition and inspection of surfaces for distortion, flaws, and/or textures. Such tasks can be difficult, especially for complex surface shapes or variation in surface reflectance with parts production at high rates.

The application of machine vision to automated surface inspection can be found in many quality control processes, including manufacturing and assembly operations, such as those found in the pharmaceutical, semiconductor, food processing, and floor covering industries. Because of its broad application base, machine vision has had some ambiguity in terminology and lack of an all-purpose approach. When machine vision algorithms used for pattern regconition, we find a number of descriptive parameters from simple area and perimeter caculations to more complex metrics, which may be used in a basis feature vector for statistical pattern recognition (Webb 2002). Surface inspection typically employs various techniques for the automated detection of surface blemishes and aberrations and extracts features from an acquired 2D image of intensity data. Numerous investigators report defect identification and classification techniques based on measurements of image intensity variation and a shape description of the segmented local irregularity (Smith 2001; Lolla 2005; Zhou 2007; AIA 2009a).

The study herein was motivated by operations of a commercial flight-type dishwasher in a private 700 bed hospital in the mid-western U.S., which operates 3 two-hour shifts per day with approximately 700 trays of dishes and silverware per shift. One tray typically consists up to six dishes of five different types, three or four silverware pieces, and a cup (Johnson 1993; Nagraj 2003; Yeri 2003; Lolla 2005; Peddi 2005; Zhou 2007). Such commercial dishwashing systems currently involve human loading, sorting, inspecting, and unloading dishes and silverware pieces before and after washing, in hot and humid environments. In such difficult working conditions, leading to high turn-over of low-paid employees, automation is desirable, especially in large scale kitchens of hospitals, navy ships, schools, hotels and other large dinning facilities. Our project is a part of developing an integrated machine vision sorting and inspecting system for dish pieces and silverware exiting a flight-type commercial dishwashing machine, coupled with automatic loading and unloading for large automated dishwashing operations.

Although the description of need seems straightforward, designing and building an efficient high speed and cost effective automatic system is challenging. Inspection difficulties are compounded by food particles (“dirt”) of different food types varying in color, size, shape, and position on a dish piece. In addition, glare and shadows increase the difficulty of discerning clean from dirty dishes, even for human manual inspection. Moreover, because of the non-flat geometry of the dish surface, the gray intensity of the image drops significantly at the dish wall, especially for a deep dish with steep sidewalls. Even the definitions of a “clean dish” and a “dirty dish” are subjective and ill-defined. Identification on the other hand is an easier task to perform, since it may be solved based only on the size and shape of a dishware or silverware piece. Therefore the accuracy of identification will typically be higher than the accuracy of inspection, whether done manually or automatically with a machine vision system (Lolla 2005; Zhou 2007). For manual labor, because of the tedious and repetitive task in a difficult working environment, efficiencies of both identification and inspection processes decrease as production rate increases. In manual operations, accuracy of inspection most likely declines faster than that of identification as production increases.

1.2 Literature Review

Johnson (1993), working on the same dish set with the one used in this project, employed area and radius of the corner of the dish in using machine vision to automatically identify dish pieces exiting commercial dishwashing machines. His method required an invariant position of a dish under the camera axis, which required a pre-location mechanism for each dish piece. Even with pre-location, he reported poor repeatability of results under small lighting variations, such as those due to fluctuations in power supply voltage. He claimed that using indirect ultra-violet (UV) lighting, from fluorescent lamps, together with UV band pass filters attached to camera lens, provided better contrast between defect spots and dish surface than incandescent lighting. For inspection of dishes for cleanliness, his algorithm used simple global thresholding, which was found to be insufficiently accurate for real implementation. Furthermore, the smallest detectable dirt spot size in his approach was too large compared with the size of real dirt found on dishes.

Hashimoto (1995) developed a prototype machine to singulate silverware pieces and process them for identification and inspection. Her mechanism used a vibrating feed-hopper storing a mixed batch of silverware pieces, together with a magnetic conveyer to pull single pieces from the bottom of the hopper. However, her constructed prototype had limited capability, providing only 14.9% successful alignment and 41% singulation of silverware pieces. She suggested that dividing a large batch of silverware pieces into several small batches before feeding them into the hopper could improve the singulating of the system. She also recommended that adding more magnet-carrying conveyor strips and passive-alignment channels to the conveyor could potentially solve the miss-alignment problem. Nagaraj (2003) devised a sorting and orientation turntable device to sort and orient silverware pieces. Her sorting machine was capable of processing a maximum of 34 silverware pieces per minute with an average sorting error of 1.87%. But if her sorting and orienting device was integrated with a machine vision system for prior identification of silverware pieces (Yeri 2003), the combined system production rate was reduced to a maximum processing rate of 21 pieces per minute with an increased average sorting and orienting error of 8.4 percent. She suggested that replacing Yeri's (2003) serial process identification with a much faster process handling multi-threaded programs, could improve the processing rate. Peddi (2005) avoided the difficulties encountered by Nagaraj by using a set of relay actuated fingers to selectively lift identified silverware off a magnetic conveyor. His prototype system had capability of sorting and orienting 55 silverware pieces per minute with 98% of accuracy. Yeri (2003) devoloped a machine vision system, including a frame grabber PCI card, camera, lenses, lighting equipment, lighting setup, camera trigger circuitry using optical sensors, and software applications, using blob analysis to recognize silverware pieces and their orientations. He set up indirect illuminaiton using a light tent to diminish speccular reflections. His identification algorithm, based on area of silveware piece image alone, was insufficiently accurate for real implementation. Lolla (2005) improved on Yeri's results and focused on inspection of silverware pieces, which Yeri did not address. Both Yeri and Lolla used NI-IMAQ Vision Software Libraries (Device Drivers, Hardware & Software Interfacing software) in a Microsoft Visual C++ environment in their studies. Lolla identified silveware objects by their perimeters, symmetric and asymmetric properties, and area moment of inertia measurements. He used edge detection algorithms together with template matching to inspect recognized silverware pieces. His algorithm produced 87% accurately classified clean silverware pieces and 91% accuracy inspection of dirty silverware pieces at average at speed of approximately 55 pieces per minute. He also sugguested several ways to increase the accuracy of his algorithms, such as using a color camera, a camera with higher resolution, or employing a thermal/Infra-red imaging technique that he expected would eliminate spurious specular reflections in silverware pieces, as long as the costs of these improvements were acceptable. Zhou (2008) proposed an algorithm to recognize silverware pieces even with incomplete (truncated) images and a fusion-based method for silverware inspection, producing very good results. For the identification problem, he developed a Complete-Pattern algorithm and a Part-Pattern algorithm, using respectively complete template images and imcomplete template images of silverware pieces. Both could be used to regconize incomplete images. A new image registration algorithm, intended to work with slim objects like silverware pieces, was developed for image fusion purposes For inspection, his key idea was based on the observation that shadows will move, but dirt will not move, between two images of a silverware piece captured at two different positions under fixed illumination. After fusion of two images of a single silverware piece, Zhou applied simple global thresholding to the three color (R, G and B) channels and searched for any difference in three channels which indicated dirt spots. While this approach worked well for silverware, it will not work for dish inspection, where the intensity of a dirt spot on the dish floor is much greater than the intensity of a clean portion on dish’s wall.

1.3 Objective and Problem Statement

The objective of this project is to develop fast, adaptable and efficient algorithms and procedures for on-line dishware identification and inspection of certain types of dish pieces exiting a commercial dishwasher at a minimum rate of 30 dish pieces per minute.

The dish set used in this project was commercially available and used by a large, 700 bed hospital in Oklahoma. It consisted of 5 types and 3 colors of dishes, and was selected not only because it was in wide commercial use, but also because the colors, shape, and size of different types of dishes are very similar. Each dish piece had uniform color with no decorative markings (i.e. they were “plain”). The identification and inspection algorithms should work with dish images captured at different positions and orientations under the camera axis.

1.4 Outline of This Study

This report includes 4 additional chapters beyond Chapter 1. Chapter 2 describes the experimential setup and perprocessing of images of dish pieces. In Chapter 3 we present several different attempts to solve the dish identification problem, including the final identification method using statistics of shape decriptors of dish pieces. Chapter 4 decribes several attempts to solve the inspection problem, including a proposed new technique using partitioning and adaptive thresholding, combined with global thresholding. Conclusions and recommendations are given in Chapter 5. Figure 1 illustrates the overall process, which we will describe in what follows.

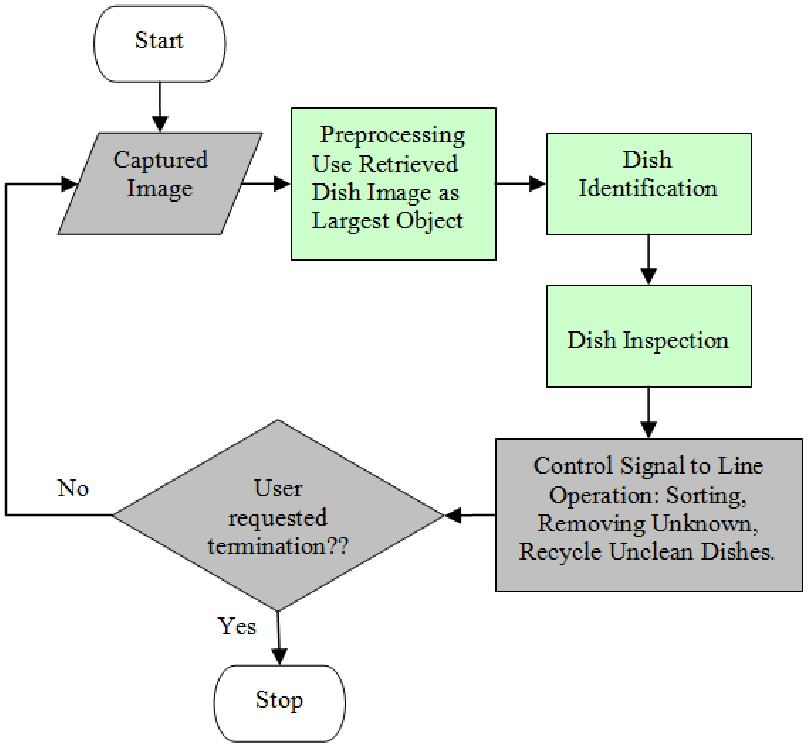

Start
Captured Image
Preprocessing Use Retrieved Dish Image as Largest Object
Dish Identification
Dish Inspection
Control Signal to Line Operation: Sorting, Removing Unknown, Recycle Unclean Dishes.
User requested temination??
No
Yes
Stop

Figure 1: Processing Flow Chart

CHAPTER 2

EXPERIMENTAL SETUP AND PRE-PROCESSING

Designing a lighting setup is a critical step for building a machine vision system for automated surface inspection applications. According to the Machine Vision Association (2000), “Lighting and optics are 80 percent of most vision applications”. In order to detect defects on product surfaces, a machine vision system must be able to differentiate the defects from the background. The ultimate purpose of lighting is enhancing those differences and/or eliminating specular reflections and shadows, which typically result in incorrect processing of the image. Unfortunately, a lighting setup that could minimize specular reflection and shadow effects usually depends upon the geometry of product surfaces. Therefore it is problematic to design a single lighting setup that works effectively for a wide variety of applications. In designing an imaging system, there is typically a trade off between accuracy and speed. An enlarged image could provide greater detail, but requires more time for processing. A reduced image can be processed quickly, but might not yield enough details for correct decisions. Most automated surface inspection applications, including our study herein, take advantage of prior knowledge about the product geometry to design the most suitable lighting setup and imaging system (Smith 2001; Lolla 2005; Zhou 2007). Machine vision systems for those applications are all domain-dependant and can be classified as subjective or viewer-centered approaches.

2.1 Camera Configuration

The camera used in this project was an area scan, color digital industrial camera, Basler Co. model A102kc, directly connected to an image processing board in a personal computer (Pentium dual core 3.0, 4GB RAM) for real time image processing. It offers features including RS-644 programming capability, high signal to noise ratio, electronically controllable sensor exposure time, individual electronically adjustable gains and offsets for red, green, and blue channels (for white balancing), electronic trigger capability, and a maximum frame rate of 75 frames/sec. The camera setting could be configured either by the Basler CCT+ Camera Configuration Tool in a Windows (™) environment or by binary programming commands which can be included in user application programs. The camera’s sensor, a Sony ICX285 Progressive Scan CCD Sensor, has a principal spectral response from 400nm to 700nm, peaking at approximately 470nm for the blue channel, 540nm for the green channel, and 620nm for the red channel, as shown in Figure 2.1.

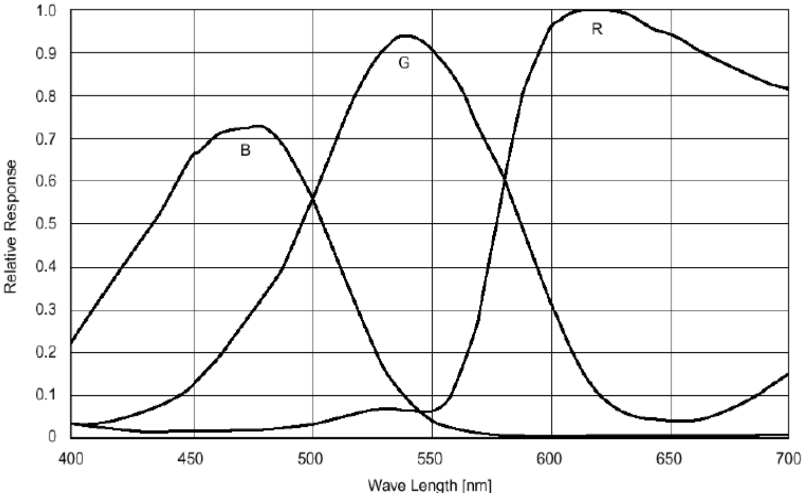


Figure 2.1 : Basler color camera model A102kc Spectral Response (Basler Co. 2007).

During the imaging process, the optical equipment focuses the reflected light from the object to be imaged onto the camera sensors, which then converts the light intensity signals to electrical signals. Those signals are then digitalized and interpreted as an image of the object. The criteria for selection of optical equipment includes: resolution, sensor size, field of view, working distance and depth of field, and are illustrated in Figure 2.2.

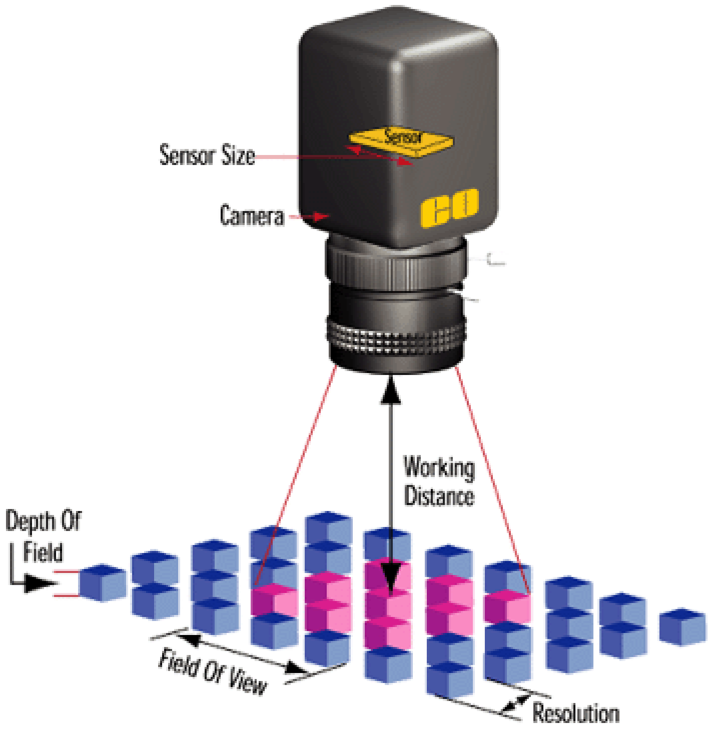


Figure 2.2 : Fundamental Parameters of an Imaging System (EIO 2009a).

Sensor size in the camera was approximately 17 mm square. Resolution was set to 1392 by 1040 pixels, which is sufficient to discern a small dirty spot of SFS (smallest feature size) mm square. The lens was a Fuji model CF35HA-1, 35mm focus length, with 14^{o}26'x10^{o}46' aperture view cone. Let sensor size SS=17mm, sensor resolution SR=1040 pixels, and focal length FL=35mm. Choose working distance WD=600mm, which is large enough to avoid distortion when the lens is focused to an object within the working distance (Zhou, 2008). Then we can calculate the smallest feature size, SFS, that the camera can detect (EIO 2009a) as:

$$SFS = \frac{2 \times FOV}{SR} \qquad (2.1)$$

where FOV= 200mm is the field of view, given by

$$FOV = \frac{SS \times WD}{FL} \qquad (2.2)$$

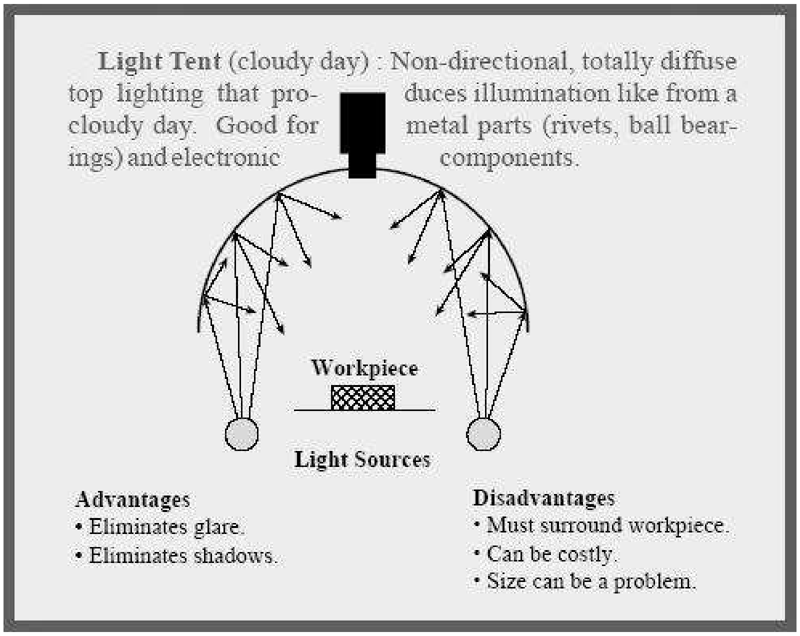
Subtituting FOV from (2) in (1) yields:

$$SFS = \frac{2 \times SS \times WD}{SR \times FL} \qquad (2.3)$$

Using the above numerical values in (3) gives SFS= 0.38mm, which is sufficiently small for detection of food particles in our project.

2.2 Lighting Design

Our experimental setup was inherited from Zhou (2008), Lolla (2005), and Yeri (2003). They all worked on silverware pieces, which were metallic and had complex geometric surfaces with a shiny finish. Such smooth surfaces would produce specular reflection, which resulted in "glare" being formed in the image, causing deterioration of the quality of the object image. Yeri (2003) designed a light tent to weaken specular reflections. Yeri's lighting setup was an adapted version of the "light tent" setup, shown in Fig 2.3, and had advantages of minimizing glare and shadows, which is a well-established lighting technique (MVA/ SME 2004). However, the images acquired by Yeri's system still revealed undesirable characteristics including shadows and ill-lighted regions on silverware pieces (Lolla 2005).



Light Tent (cloudy day) : Non-directional, totally diffuse top lighting that produces illumination like from a cloudy day. Good for metal parts (rivets, ball bearings) and electronic components.

Advantages

- Eliminates glare.
- Eliminates shadows.

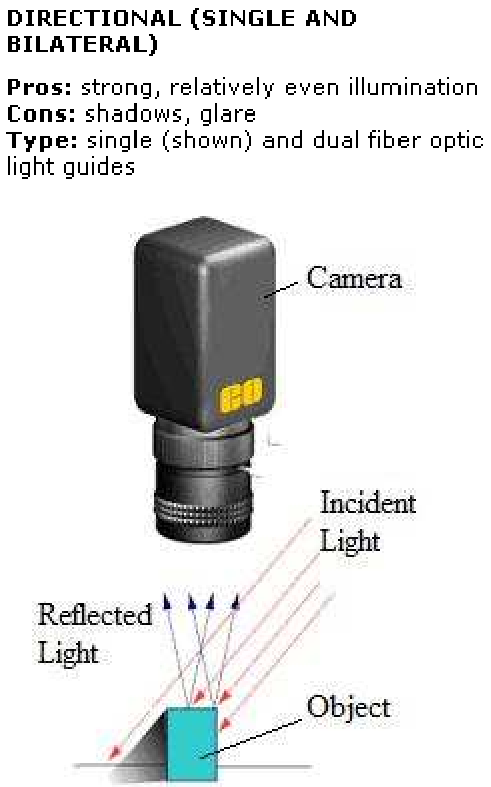
Disadvantages

- Must surround workpiece.
- Can be costly.
- Size can be a problem.

Figure 2.3: Light Tent Illumination Model (MVA/ SME 2004).

Yeri also experienced image shape distortion, which could be explained by the silverware pieces vibrating as they were conveyed by sliding beneath the camera. Peddi (2005) modified the conveying mechanism, eliminating sliding, which eliminated this distortion. Lolla (2005) improved Yeri’s imaging system by adding curtains outside the tent, which reduced shadows. Lolla also used DC lamps together with a Switch Mode Power Supply (SMPS), commonly used to power PCs, to eliminate noise in acquired images. Zhou (2007) attached the curtains inside the tent and made them capable of moving with the magnet holding the silverware pieces. With this modification, Zhou reported that almost all shadowing could be removed.

Although the lighting setup used by Zhou (2007) and Lolla (2005) worked well for silverware pieces, it would not work for dishware pieces because dish surfaces not only are less shinny than those of a silverware piece, but also have unique 3D geometric shapes. We needed stronger light sources than used for silverware to illuminate dish pieces better. For inspection purposes, we desired uniform illumination across the dish piece, as well as minimum specular reflections and shadows. Among the well-established lighting techniques, the “light tent” (Fig 2.3) and the “direct front illumination” (Fig 2.4) lighting setups appeared to offer the best compromise for the system under consideration.

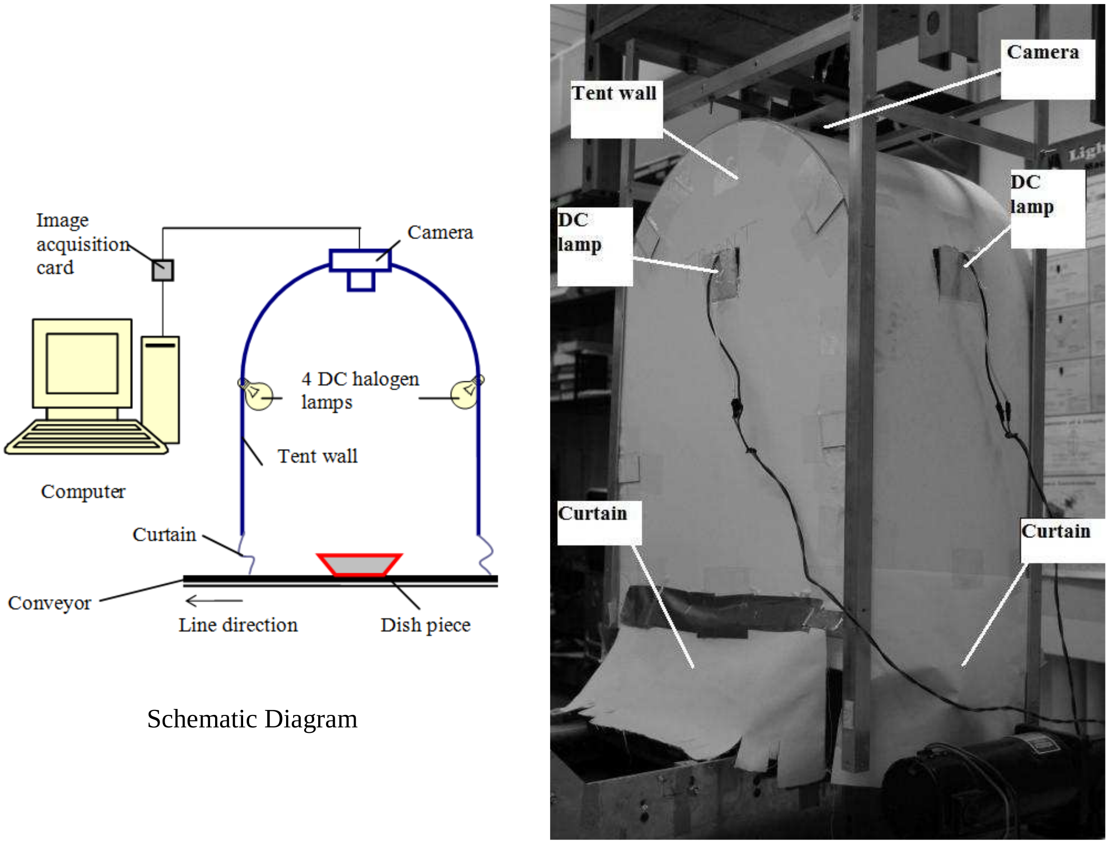


DIRECTIONAL (SINGLE AND BILATERAL)

Pros: strong, relatively even illumination
Cons: shadows, glare
Type: single (shown) and dual fiber optic light guides

Figure 2.4: Directional Front Illumination (EIO 2009b).

Our experimental setup, inherited from Zhou (2007), with modifications in lighting is shown in Figure 2.5.



Schematic Diagram

Photograph

Figure 2.5: Experimental Setup with Our Modifications in Lighting.

After washing, it is anticipated that dish pieces will be automatically placed on the conveyor. However, in this project, dried dishes were placed manually. A dish image was captured by the camera when the dish was inside the light tent in the camera field of view. In full implementation, image taking will be triggered by appropriately placed sensors, but in the work herein, this triggering was done manually. This image was transmitted to a computer for image processing. The tent wall and curtains are used to eliminate unexpected illumination from the outside environment. In actual implementation, after processing, a signal identifying the type of dish piece will be sent to a sorting mechanism to sort the dish into a stack of like dishes. Unidentified objects will be automatically sent to a bin for such objects, or if a dish is determined to be unclean by the vision system, a signal will be sent to convey said dish into a bin for re-washing.

The white mat finished cardboard forming the inside surfaces of our light tent provided some diffuse lighting of our dishpieces and reduced specular reflections, while the choice and placement of our lights reduced shadowing. After considerable experimentation, we selected as our light sources four 12V-20W DC halogen light bulbs surrounding the dish piece and placed as indicated in Figure 2.5, which provided sufficient illumination for both identification and inspection. These lamps were powered by a Switch Mode Power Supply (SMPS) fed by 110V 60Hz building supply. There were many ways to achieve desireable lighting, such as changing emitted energies of light sources by regulating the supplied power voltage, or changing exposure time and/or the shutter speed of the camera. To avoid the need of a voltage regulator, we chose tuning the exposure time of the camera, which could be programmed. Acceptable lighting was achieved by trial and error. Fig 2.6 shows examples of dish images captured under different exposure times. By subjectively visual assessment, we observed that the ceramic SC dish images (2.6a) exhibited shiny surfaces and provided good details without excessive glare at exposure times of 16 to 20 ms. We also observed that exposure times of 18 to 24 ms produced the best result for the spacer plastic SP dish images (2.6b), and 18 to 22 ms produced the best results for the plastic SX dish images (2.6c).

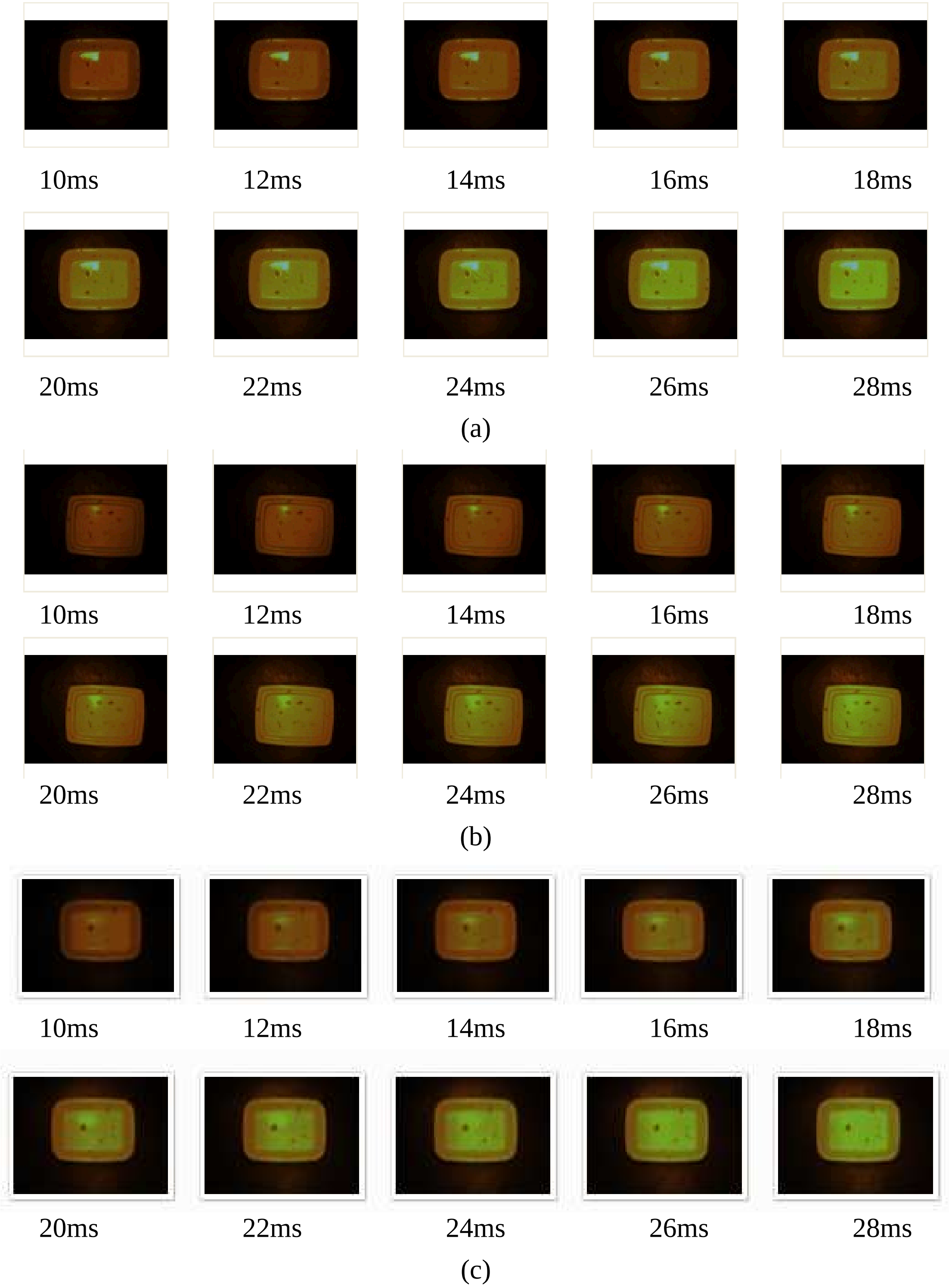


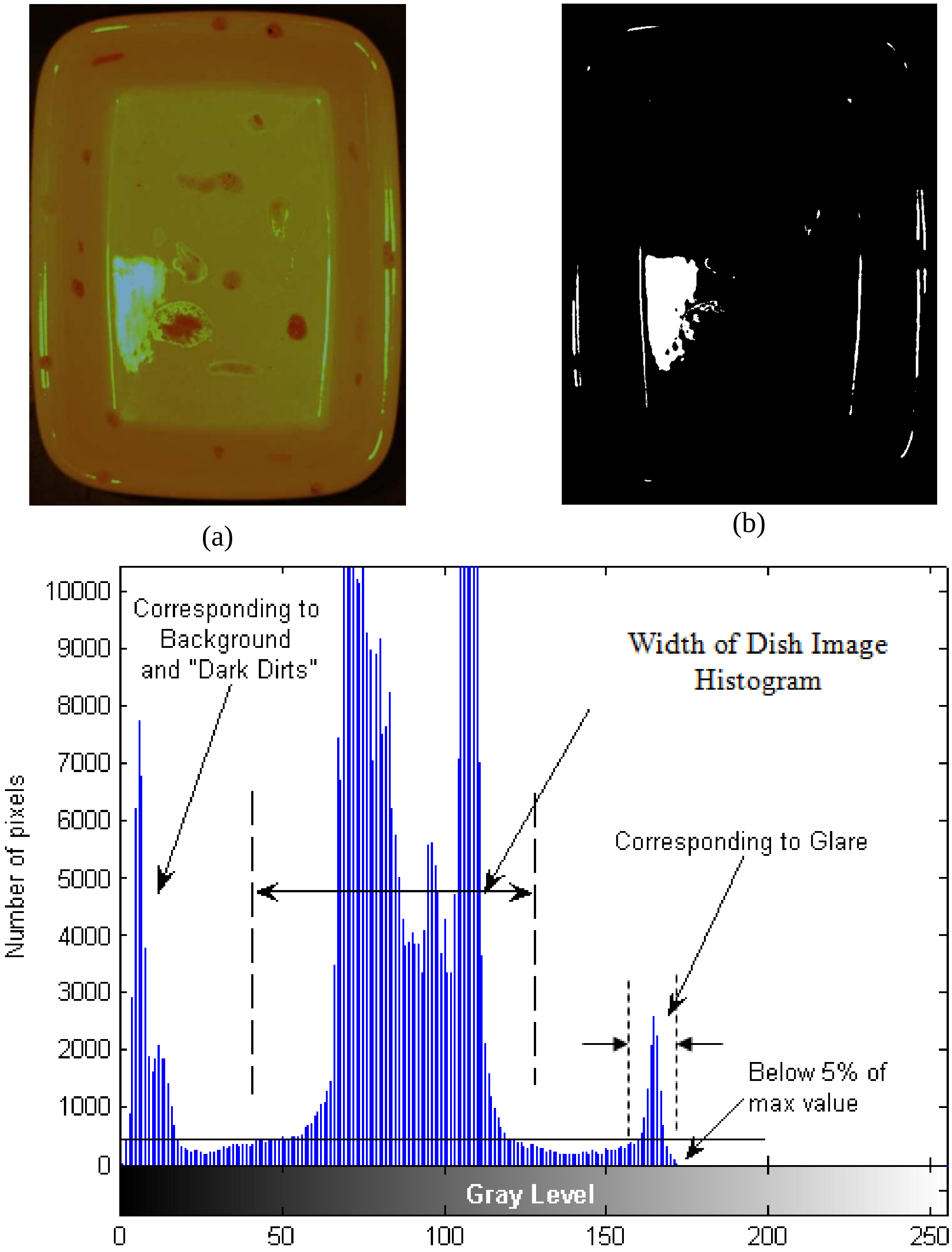
Figure 2.6: Dish Images Captured Under Different Exposure Times as Labeled.
A Ceramic SC (a), a Spacer Plastic SP Dish (b), and a Plastic SX Dish (c).

(As will be described later, our dish set had five different types of dishes, two of them ceramic and three of them plastic).

A more objective approach for choosing a reasonable camera exposure time employed the two following criteria:

- Width of histogram of dish image – desired to be large to maximize dirt detection.
- Area of glare region on dish images – desired to be small to minimize glare.

More illumination provided in the field of view (without over-exposure) produces more detail that can be dishtinguished on the dish images, and also produces a larger the width of the dish images histogram. At the same time, however, more illumination produces more glare on the dish image. Figure 2.7 (a) and (b) show color and binary images of a dirty dish with glare, while Figure 2.7 (c) shows the corresponding histogram for the gray level version of Fig 2.7 (a). We have also indicated on the histogram the correspondence with areas of the dish image in Fig 2.7 (a). A pixel is considered to belong to the glare region if its gray level is greater than 150 (gray level range is 0-255). The area of glare region on the histogram is defined by the number of pixels belonging to glare. The width of the dish image histogram is difined by the number of gray levels belonging to the dish image. We count only the gray levels in range 45 to 150 for the dish image, with gray levels below 45 corresponding to the dark background and gray levels greater than 150 corresponding to glare. We also ignore gray level values that lie below 5% of the maximum histogram value as representing noise.



(a)

(b)

(c)

Figure 2.7: Glare Area and Width of Histogram Illustrations
(a) A Ceramic SC Dish, (b) Binary Image Showing Glare Region as White, (c) Histogram of the Gray Image of (a)

Figure 2.8a shows that both the width of histogram and the glare area of SC Dish Images, captured under different exposure times (10 to 28 ms), increases with exposure time. Clearly, the exposure time of 20 ms provide a reasonable compromise between the two goals of having the width of histogram dish image large while having the width of the glare area small.

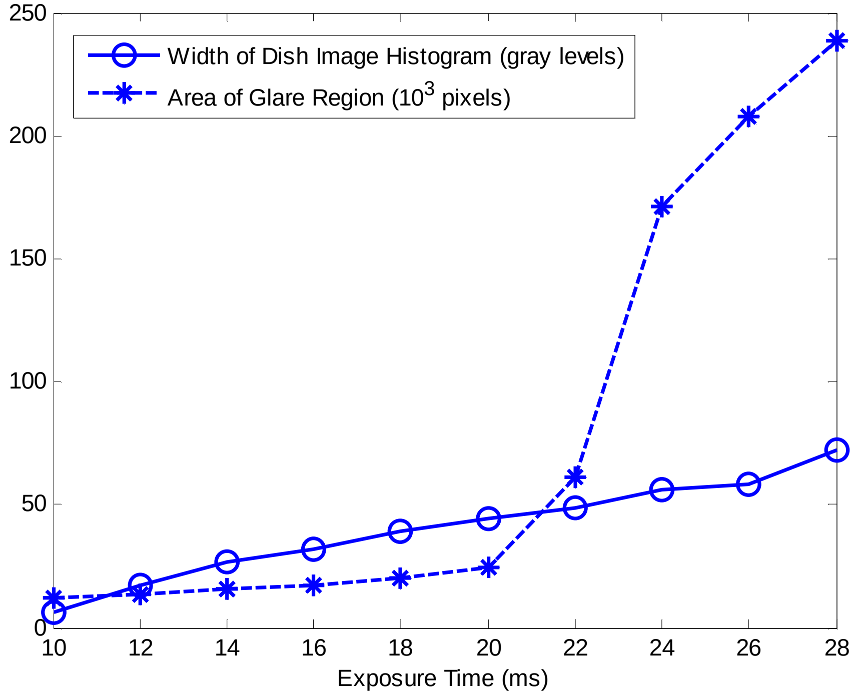


Figure 2.8a: Width of Dish Image and Glare Area of Ceramic SC Dish Images vs. Exposure Time

Similarly, for the plastic spacer SP dish image (see Fig 2.8b) and the plastic SX dish image (see Fig 2.8c), the compromise exposure times can be seen as 22 ms and 20 ms, respectively. Accordingly, because we wished to use a single exposure time, we selected 20 ms for all five dish types (The dishes LC and LP, will be described in Section 2.3, were made of the same materials as SC and SP, respectively, such that exposure time analysis would produce the same result for LC and LP). This result matches that determined by subjective visual inspection, discussed previously.

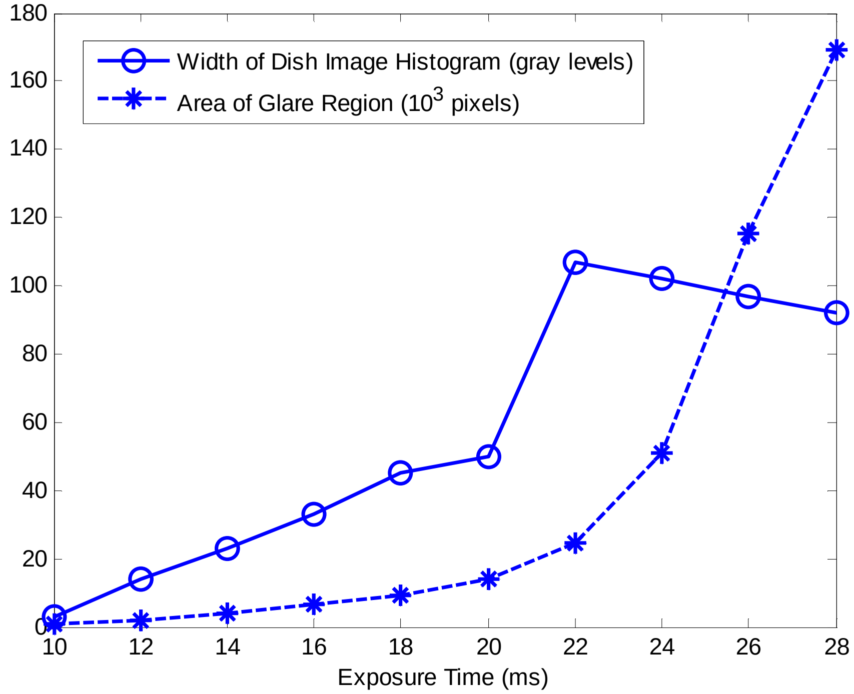


Figure 2.8b: Width of Histogram and Glare Area of Plastic Spacer SP Dish Images vs. Exposure Time

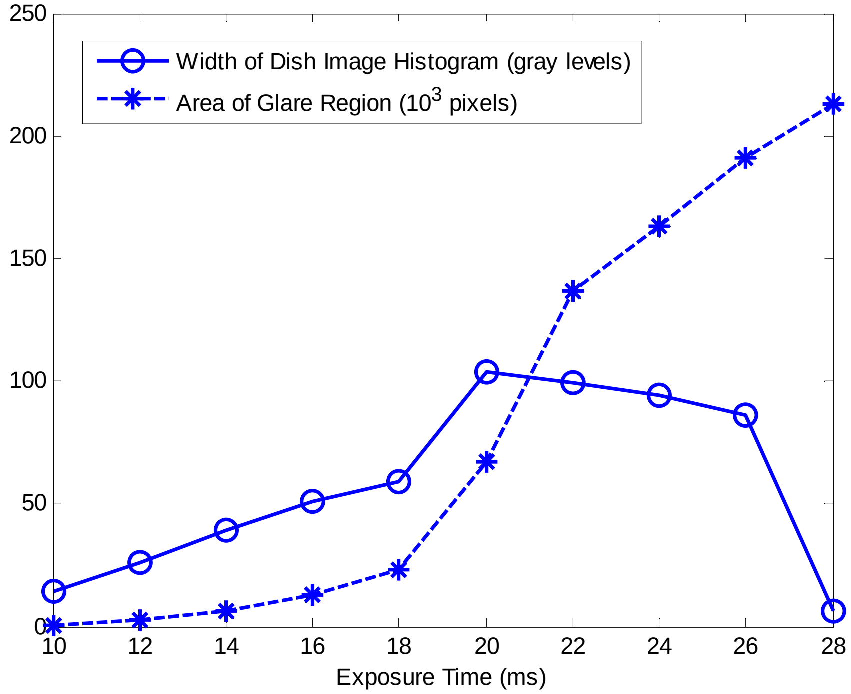


Figure 2.8c: Width of Histogram and Glare Area of Plastic SX Dish Images vs. Exposure Time

2.3 Pre-processing

Dish set

Our dish set, shown in Fig 2.9, was commercially available and used by a large, 700 bed hospital in Oklahoma. It consisted of 5 types and 3 colors of dishes. For easy reference, we named each dish piece using size (large or small), and its material or function (ceramic, plastic, or spacer). Then LC and LP represent respectively the large ceramic dish and the large plastic dish, while SC, SP and SX represent respectively the small ceramic dish, the small plastic dish, and the small spacer dish.

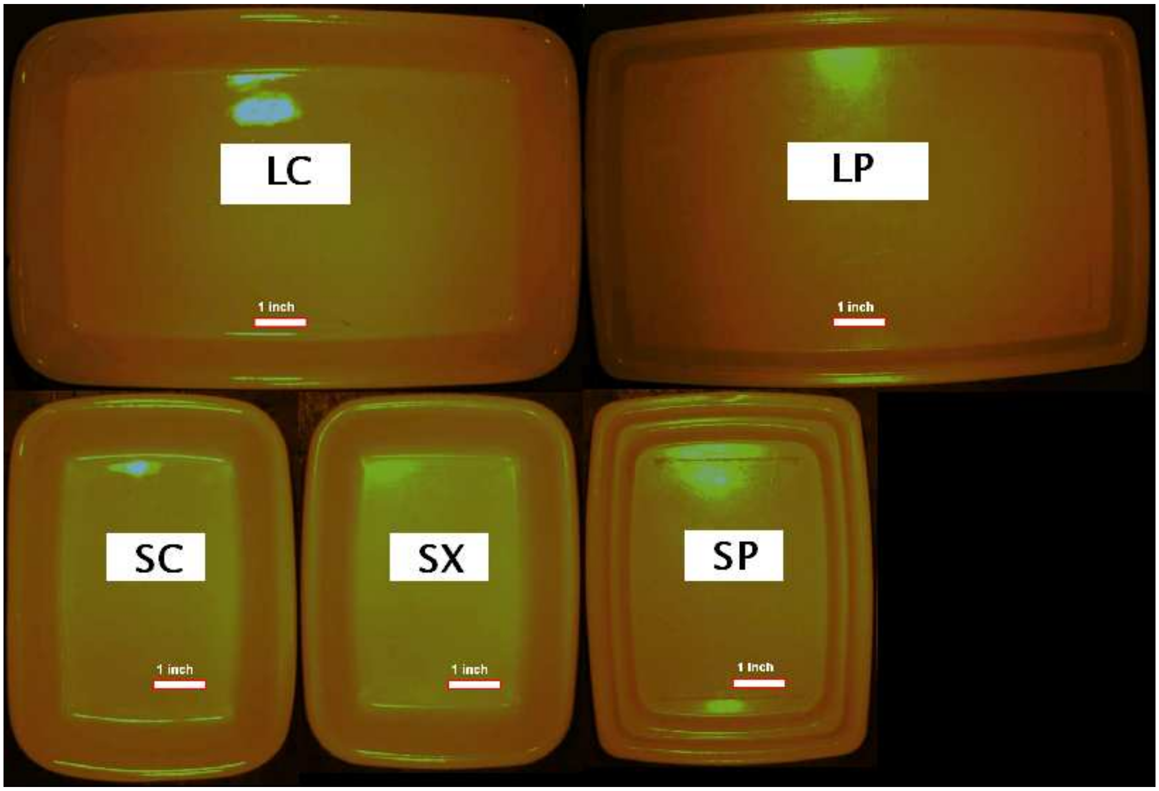


Figure 2.9: Our Dish Set.

The reason for selecting this dish set for our study is that not only it is in wide commercial use for institutional food service, but also the colors, shape, and size of its different types of dishes are very similar. However, each dish piece had uniform color, with no decorative markings (i.e. they were “plain”). Other commercially available “plain” dish sets present much lower challenges for the identification problem because their sizes are often easily distinguishable. Figure 2.10 shows other two commercial “plain” dish sets. Compared with our dish set, dish pieces in Fig 2.10 could be recognized by combining information from their sizes, shapes, and colors.

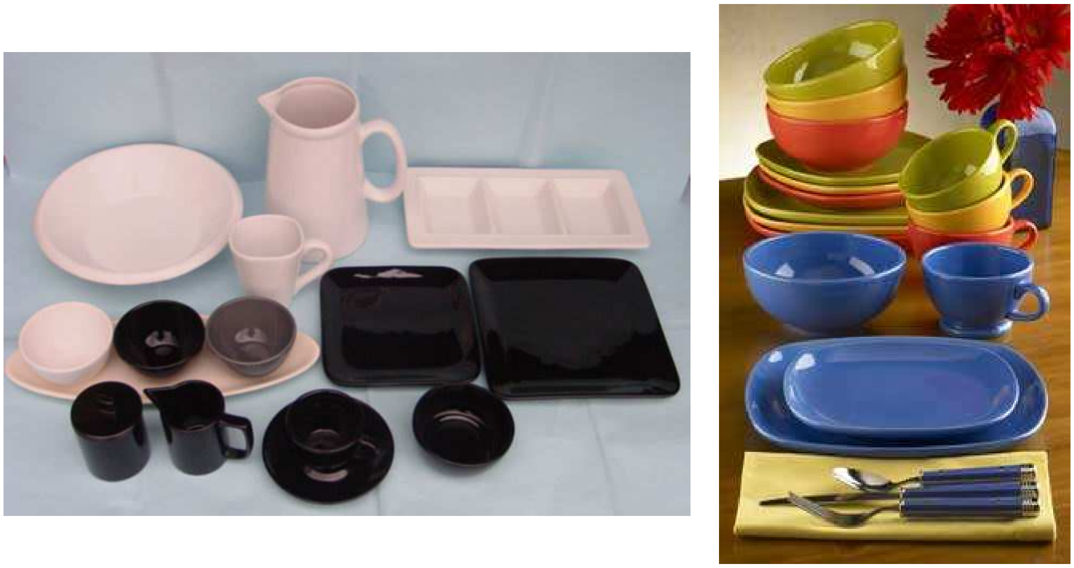


Figure 2.10: Example of Two Other Commercial Dish Sets.

Pre-processing

Images acquired from the camera could include images from objects other than a dish piece that might appear in the field of view. Furthermore, it was unnecessary to process the entire image (object plus background), which contained only a small region of interest around the dish. Hence pre-processing removed unnecessary features from the image and improved the quality of the data left in the image passed on for inspection and identification. We developed a pre-processing algorithm for thresholding, computing areas, and choosing the largest object (or particle) in a dish image, rejecting all other particles. This process removed noise and reliably retrieved a good dish image from the camera image, which was then ready for identification and inspection. It also reduced the amount of data presented for calculation in the identification and inspection processes, resulting in increased processing rate. The flowchart for our pre-processing algorithm is shown in Figure 2.11.

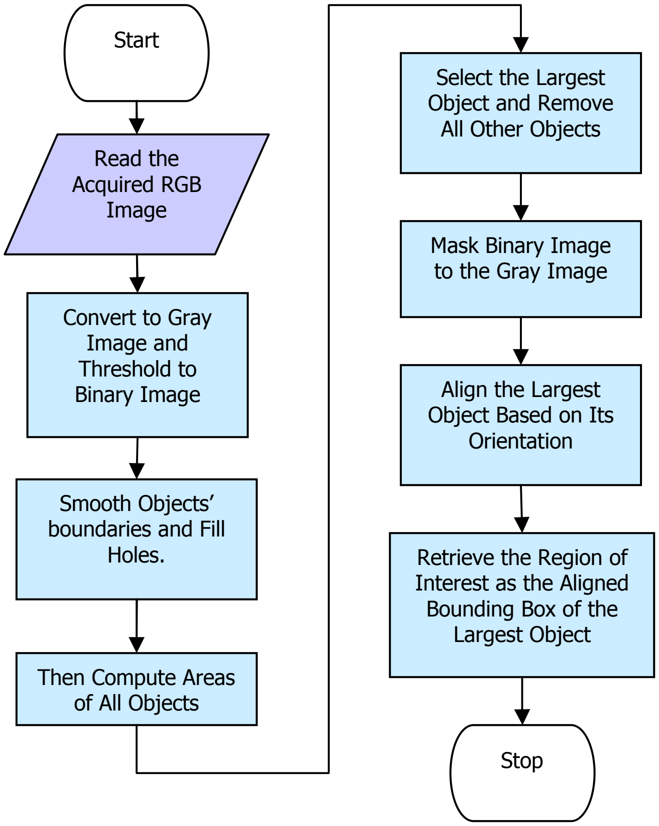
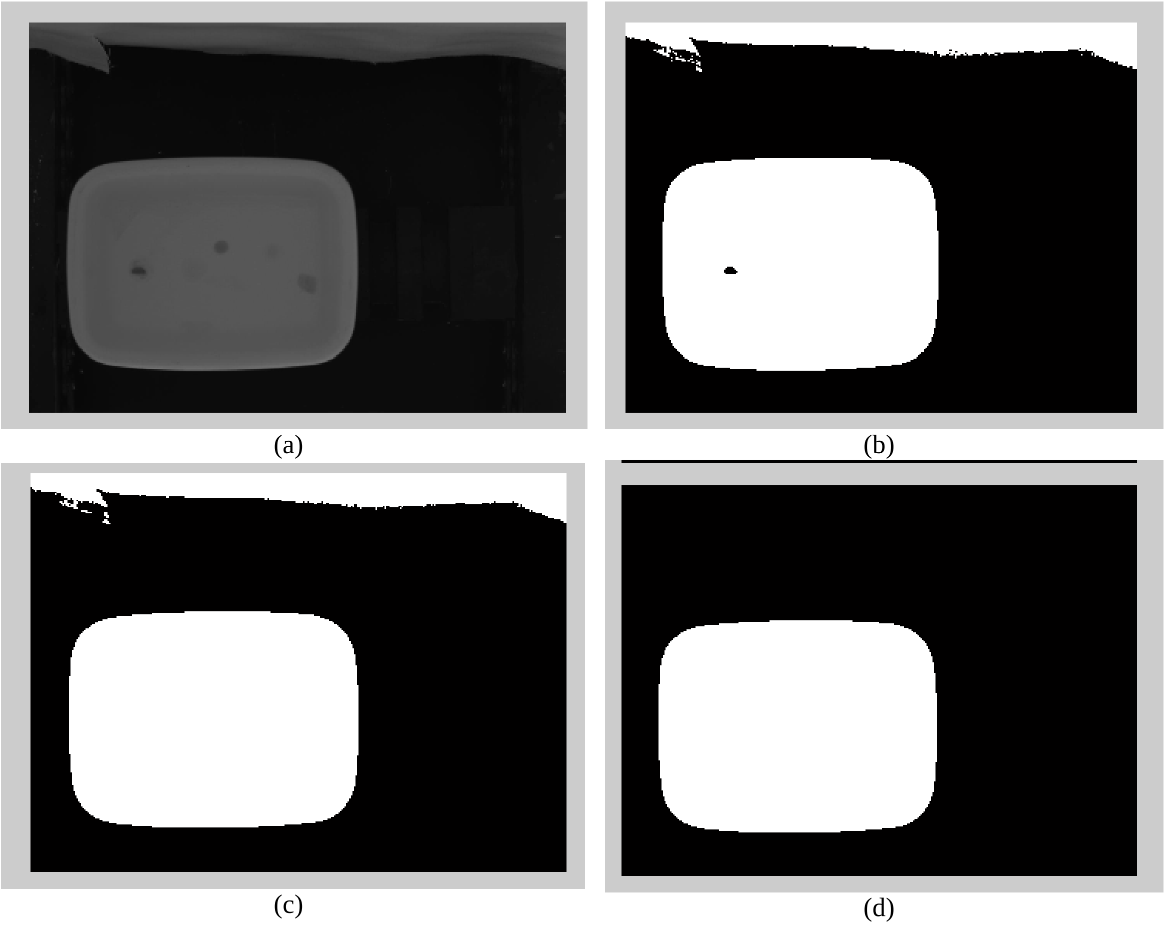

Start
Read the Acquired RGB Image
Convert to Gray Image and Threshold to Binary Image
Smooth Objects' boundaries and Fill Holes.
Then Compute Areas of All Objects
Select the Largest Object and Remove All Other Objects
Mask Binary Image to the Gray Image
Align the Largest Object Based on Its Orientation
Retrieve the Region of Interest as the Aligned Bounding Box of the Largest Object
Stop

Figure 2.11: Flowchart for Pre-processing.

As illustrated in Fig 2.12, the gray scale image (a), after being converted from the acquired color (RGB) image, was thresholded to the binary image (b). Thresholding is the process of segmenting an image into a binary image that has only two possible values for each pixel, 0 or 1, typically represented as two colors black and white. Traditionally, the thresholding process simply defines a range of intensity (or brightness) values in the original image that any pixels within this range are set to be the foreground (or objects), while other pixels are set to the background (Russ 2007, Chapter 6). While this is the simplest method of image segmentation, it is also a lossy process, in which some information about the image is irrecoverably lost during processing. Several thresholding techniques, including adaptive thresholding, will be discussed further in Chapter 4.


(a)

(b)

(c)

(d)

Figure 2.12: Example of Thresholding, Fill-hole operation and the mask.
(a) Gray-scale image; (b) After thresholding;
(c) After filling hole(s); (d) Mask of the largest object.

In our binary image, the black (0) region and the white (1) region, respectively, represent the background and the foreground. Each connected foreground region is considered as an object presented in the image. Our binary dish image could contain holes, or black regions inside dish object, as shown in Fig 2.12 b, corresponding to dirt spots (if present). Such holes were filled by applying morphological processing, which is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. By choosing the size and shape of the neighborhood, one constructs a morphological operation that is sensitive to specific shapes in the input image. The basic operations of binary morphology are dilation, erosion, closing, and opening. A dilation operation enlarges a region, while an erosion operation makes it smaller. A closing operation can close up internal holes in a region and eliminate bays along the boundary. An opening operation can eliminate of small portions of the region that jut out from the boundary into the background region (Vincent 1993; Russ 2007). From the binary image, after filling holes, we then labeled all connected components (or objects) and computed the area of each object by counting all pixels belonging to that object. The dish image was taken as the largest object presented in the image (Fig 2.12d). Therefore, after selecting the largest object and removing all other objects, by masking the binary image to the gray image, we produced a gray image that retained only the image of the dish itself, while setting other regions as the background. For efficient computation, and the need of template matching in the following inspection process, the dish image was aligned, and then only the region of interest was extracted, including the bounding box around the dish object.

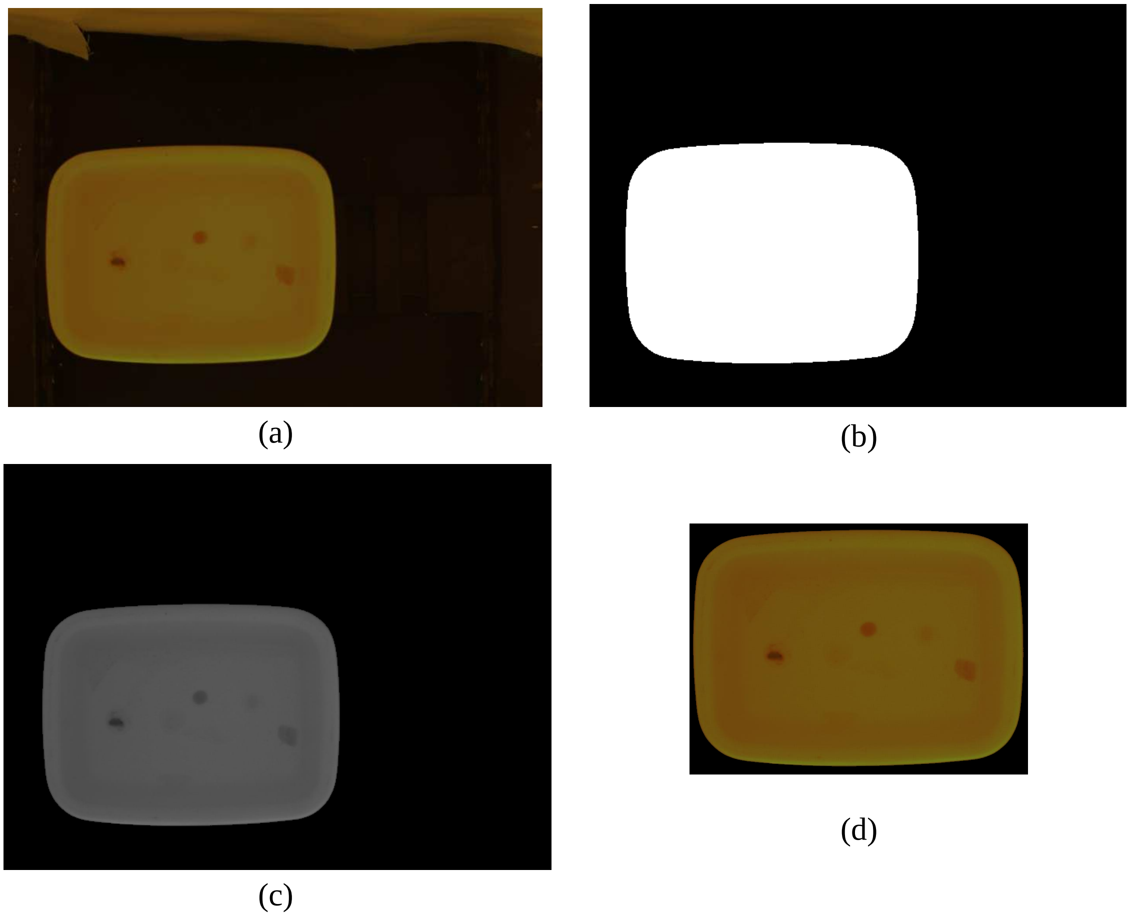


Figure 2.13: Example of Pre-processing.
(a) Original camera image (RGB); (b) Binary mask for largest object.
(c) Gray dish image; (d) Dish retrieved image.

Figure 2.13 illustrate the main steps of the pre-processing process. The captured image (a) included not only a dish piece but also a part of the moving curtain (top of image) and some small unwanted particles inside the field of view. After masking the largest binary object (b), the gray image (c) was reduced except for the dish image portion and its bounding box area. Then only the region of interest (the bounding box and the dish image) was extracted as a smaller gray scale or color image. This small image (d) was then ready for the following identification and inspection processes, discussed in Chapters 3 and 4, respectively.

CHAPTER 3

DISH IDENTIFICATION

3.1 Introduction

Our task using a machine vision system to identify 5 kinds of dishware pieces (including SC, SP, SX, LC and LP as described in Chapter 2) is a problem in pattern recognition. Pattern recognition involves observing the environment, learning to distinguish patterns of interest from their background, and categorizing the patterns. According to Zhou (2008), there are four principal approaches for pattern recognition suitable for machine vision applications, namely: template matching, statistical classification, structural matching, and neural networks. For practical purposes, the implemented automatic system should at least be comparable in performance to what can be achieved manually. Accordingly, from our observations in actual commercial dishwashing operations, our automatic system should be able to accurately recognize and inspect 5 types of dishes in real time at a minimum rate of 30 dishes per minute. While this task is easily accomplished manually, it poses a significant challenge for automation. We desire an algorithm that works flawlessly under varying dish positions beneath the camera and with varying illumination.

Zhou (2008) used a template matching method to identify silverware. He developed a Complete-Pattern algorithm and a Part-Pattern algorithm, using respectively complete template images and imcomplete template images of silverware pieces. These could be used to regconize both complete and incomplete images and produce very good results for silverware identification. Other investigators also studied identification of mixed silverware pieces (Yeri 2003, Lolla 2005) or dishware (Johnson 1993) exiting a commercial dishwashing machine. Johnson (1993) used dish image area and radius of the corner of the dish image, but his method could not work with varying dish positions and orientations under the camera axis. Yeri (2003) identified silverware pieces based only on area of their images. However, since this metric identifier does not provide any information about shape of the object, Yeri's identification algorithm could fail in cases where areas of two different pieces of silverware are the same, even though their shapes are significantly different. Lolla (2005) used three features, namely, the area of the silverware image (the largest particle), the area moment of inertia of the largest particle about an axis perpendicular to the image plane (z-axis), and the perimeter of the largest particle. Lolla's method worked well for silverware identification.

For a statistical approach, the problem is to choose a set of descriptors that is both good enough to accurately and consistently classify input objects and simple enough for effective computation. In this chapter, we present several different approaches to solve the dish identification problem, including our selected identification method using statistics of shape decriptors of dish pieces.

3.2 Trial Approaches

A human can quickly recognizes each type of dish based on the material, corner curvature, size, edge pattern, color, and/or a mix of those properties. In an attempt to imitate some of these capabilities, we experimented with several approaches based on edge detecting and statistics of shape descriptors. Our dish set, described in Section 2.3, is reprinted as Figure 3.1 for convenience.

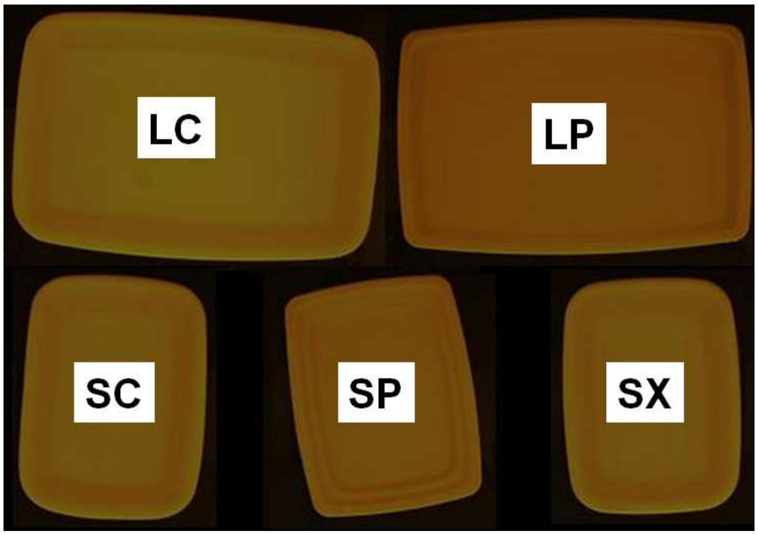


Figure 3.1: Reprint Our Dish Set.

Approach 1: Use Descriptor Set 1, having the two following metrics. The dish image must be aligned with a horizontal-vertical reference coordinate system before processing. We define "alignment" as the process of rotating the dish image to produce alignment of the major axes of the dish image with the horizontal reference coordinate.

1) The ratio of dish image length to width (called O_REC).

2) The ratio of the dish image area to the area of axis-aligned minimum

bounding box (called O_EXT)

Input: The binary image in which dish image pixels are ‘1’, and all other pixels are ‘0’ (background).

Output: Dish type code (SX, SP, SC, LC, LP, or Unknown).

The O_REC ratio can be easily calculated as the ratio of the two dimension of the axis-aligned bounding box (AABB) of the dish image and represents how similar the dish shape is to a square. The O_EXT ratio presents how nearly the dish image area approaches the area of a rectangle. In our dish images, O_EXT might be considered as indicative of the radius of the dish corner, which was used by Johnson (1993) for solving the same dishware identification problem.

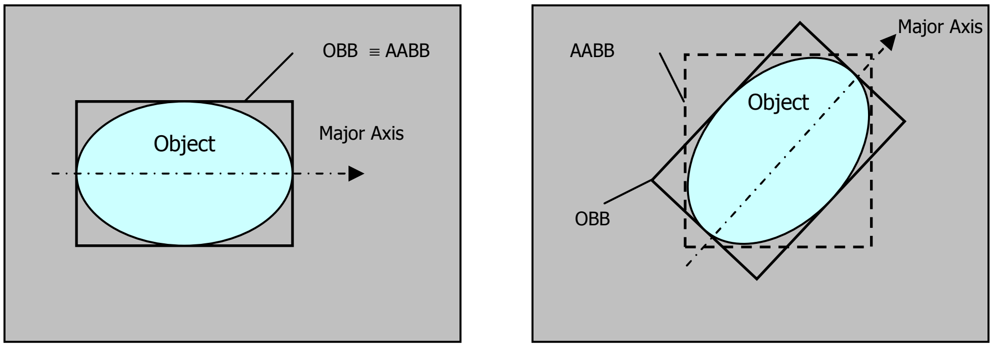


Figure 3.2: Axis-aligned Bounding Box (AABB) and Oriented Bounding Box (OBB) (Left) When the Object is Aligned; (Right) When the Object is Non-aligned.

Figure 3.2 illustrates the differences between the two common bounding boxes of an object, named axis-aligned bounding box (AABB) and oriented bounding box (OBB-which will be used later). Both these bounding boxes, as the names infer, are the minimum sized rectangles that enclose the object image. The four edges of the AABB are always parallel with the axes of the coordinates. On the other hand, the four edges of the OBB, which is, in general, different from the AABB, are parallel with the major axis orientation and the minor axis orientation (which is perpendicular to the major axis orientation) of the object image. (There is another definition of OBB (Jain 2001), which is more general than what we define in our study. According to Jain's definition, OBB is the minimum rectangle that contains the object, and its orientation could be any direction rather than the major orientation of the object). In our definition, when the object is aligned (left), the AABB and OBB are the identical. Otherwise, they are different (right).

The distributions of 84 dishes of 5 types in the two-property space (O_REC and O_EXT) are shown in Fig 3.3a and Fig 3.3b. The experimental results do not separate well, especially between the two small dishes SC and SX and the two large dishes LC and LP. The reason is that the rotation of a dish image to assure alignment produced pixel counting error when calculating the area of the bounding box, causing a reduction in accuracy of the computed O_EXT ratio.

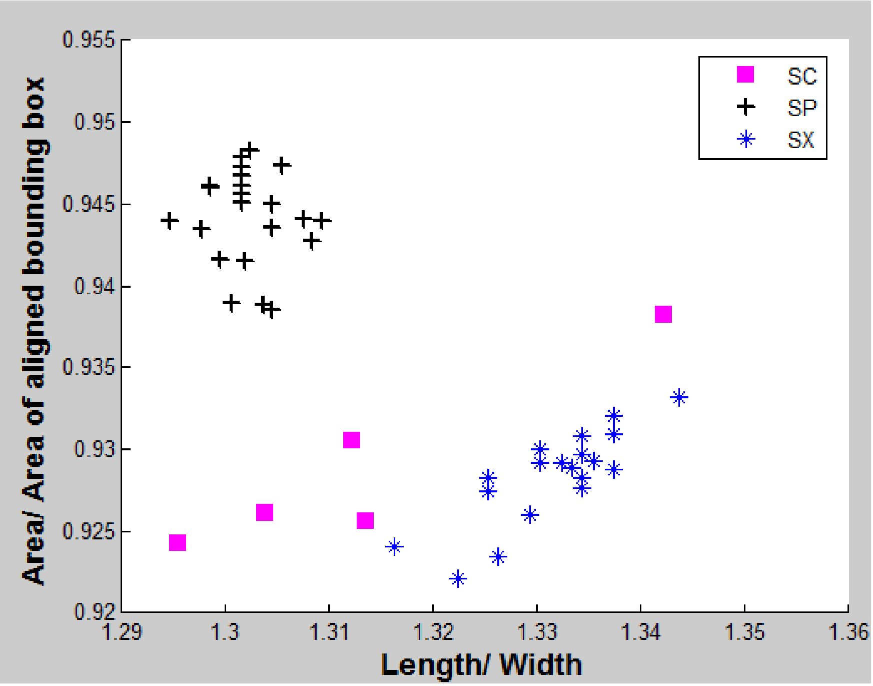


Figure 3.3a: Separating Dish Pieces by Descriptor Set 1 for the Three Small Dish Types

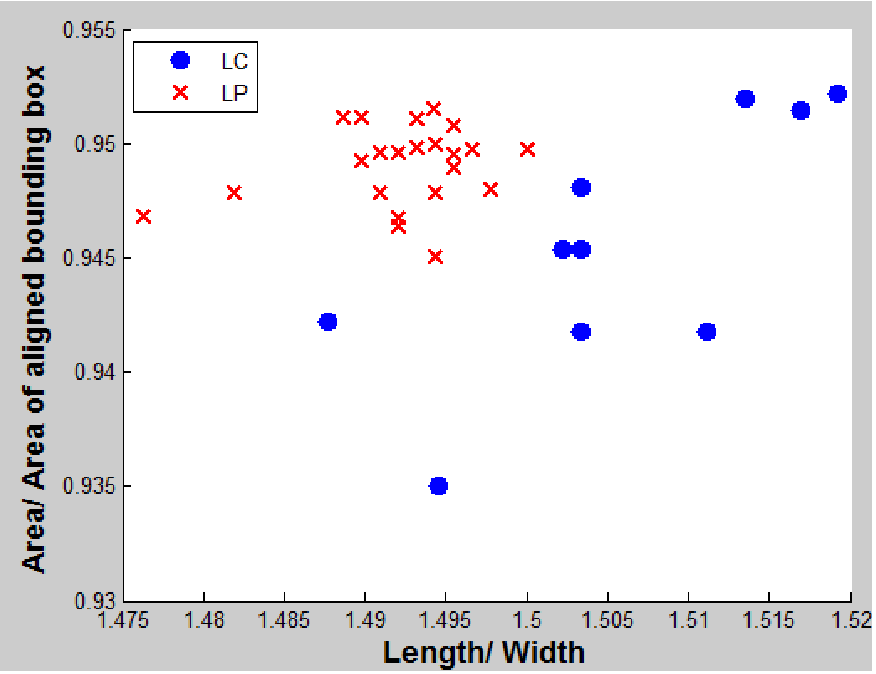


Figure 3.3b: Separating Dish Pieces by Descriptor Set 1 for the Two Large Dish Types.

Figure 3.4 illustrates how an error of a single pixel produced by the rotating operation could contribute a significant error in computing O_EXT. In general, several artifact pixels in the dish image after alignment are unavoidable. The error in counting the area of the dish object, which is typically on the order of 10^4 pixels, is acceptable. But the error in counting pixels to determine the area of the AABB of the dish is not negligible.

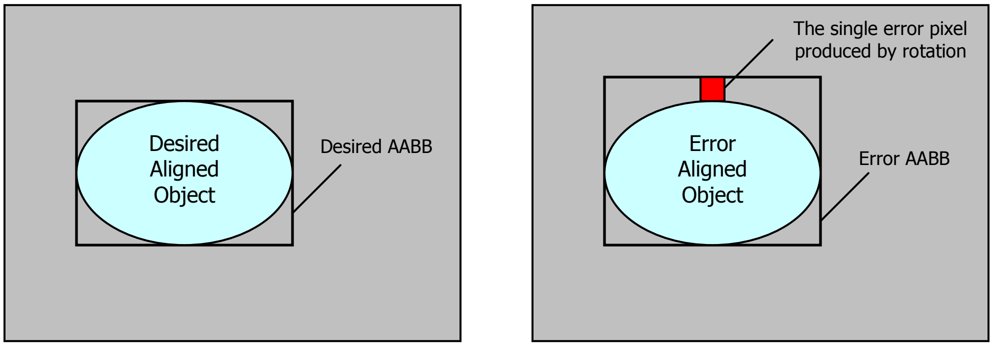


Figure 3.4: Pixel Counting Error Produced by Rotation Operation.
(Left) Flawless Rotation Results; (Right) Rotation Results with Error of a Single Pixel.

Humans can visually distinguish the slight difference in color between the two small dish types, SC and SX, which were not separated well in Fig 3.3. However, we were unable to consistently use color differences to discriminate, because the color of dish images changed significantly under varying lighting conditions. Possible solutions for improvement in this approach could be:

1) Use the oriented bounding box rather than axis-aligned bounding box to avoid rotating the dish image.
2) Develop algorithms to compensate for the pixel counting error in computing the O_EXT ratio of the dish image after rotation.
3) Consider other dish geometric parameters.

Approach 2: Use Descriptor Set 2, having the two following metrics:

1) Mean of the dish image gray level distribution.
2) Variance of the dish image gray level distribution.

Input: The gray scale dish image and its bounding box only (produced from pre-processing).

Output: Dish type code (SX, SP, SC, LC, LP, or Unknown).

The idea of this approach is separating dish pieces based on their gray level distributions (or histograms). With our assumption of uniform illumination, the dish image gray level distribution reflects mainly the dish geometry information. Therefore, with some basic indicators of this distribution, such as the mean and the variance, we speculated that the dish type could be recognized. The mean of the dish image histogram represents the size of the dish (large or small), while the variance, representing the width of the distribution, is basically a measurement of dish shape. Figure 3.5 shows examples of dish images and their histograms.

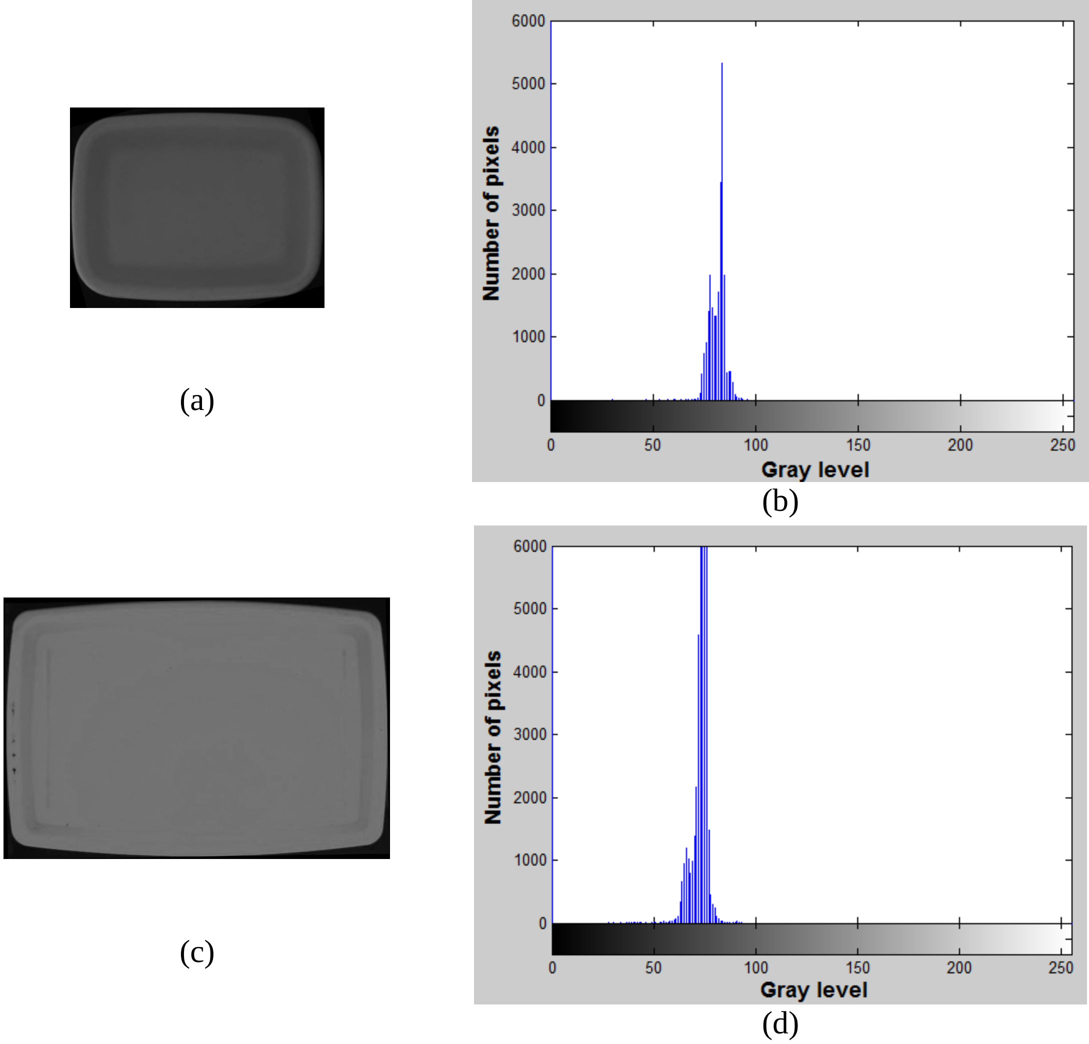


Figure 3.5: Example of Dish Images and Their Histogram.
(a): SX dish image; (c): LP dish image;
(b), (d): Histogram of (a) and (c), respectively.

Results of separating 600 dish images of all 5 types under different dish positions and varying lighting condition are shown in Fig 3.6a and Fig 3.6b. We note that the data from different dish types overlap, especially for the two large dish types, LC and LP. One of the reasons for this is that the assumption of uniform illumination was not realized. Furthermore, methods based on color or gray level intensity were found to be overly sensitive to small lighting variations. Notice that data for the SC dish type separates very well from data for other two small dish types, SP and SX. With Approach 1, the SP dish type was distinguished easily from the others. Therefore, we can take advantages of both approaches to completely identify the three small types of dishes. However, the two large dish types are still unidentifiable. One possible solution for improvement in this approach would be to employ data taken under identical lighting conditions.

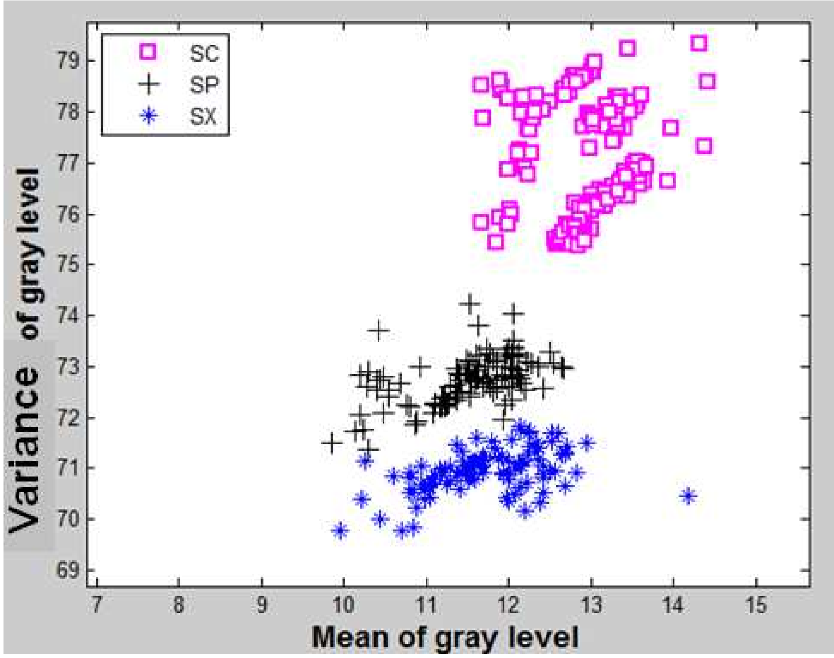


Figure 3.6a: Separating the Three Small Dish Types by Descriptor Set 2

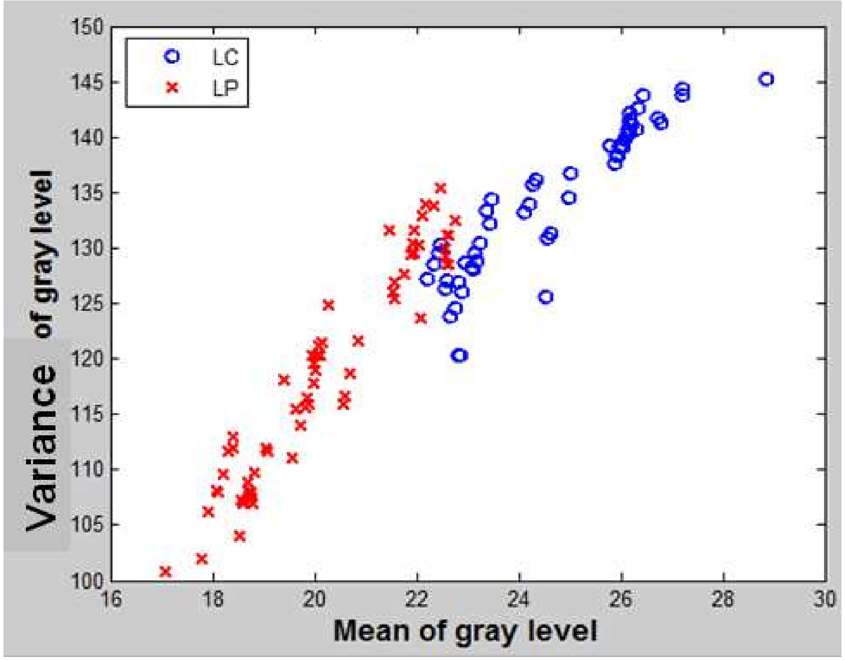


Figure 3.6b: Separating the Two Large Dish Types by Descriptor Set 2

Approach 3: Employ edge detection for separating the two small dish types, SX and SP, and the two large dish types, LC and LP. This could be one solution for the problems of Approach 2 with data overlap between SX and SP, as well as between LC and LP dish pieces. Among the three small dish pieces, SP is the most different, with the largest number of edges, such that it can be easily recognized (Figure 3.7).

LC and LP can be separated by other edge features, such as the curvature of edges, or the distances between edges. However, finding those features is time consuming and, even worse, is highly susceptible to noise. Edge detecting methods using Matlab and its Image Processing Toolbox was not only computationally expensive and slow, but also proved difficult in selecting appropriate threshold values (Duong, 2008).

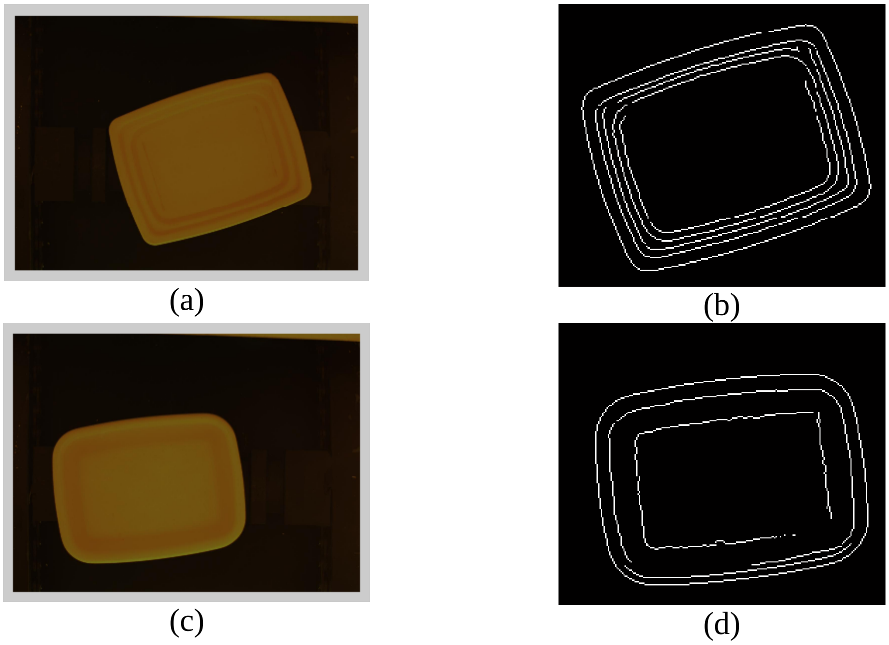


Figure 3.7: Separating SP from SC Dish Type Using Number of Edges
(a): SP dish image; (c): SC dish image;
(b), (d): After edge detection of (a) and (c), respectively.

3.3 Final Approach and Identification Algorithm

From our early experiments described above, we propose using statistics of shape descriptors of dish pieces to solve the identification problem. Three easily calculated statistics of shape descriptors are:

1) The dish image area (or the number of pixels belonging to the dish image).

2) The ratio of dish image length to width (called O_REC).

3) The ratio of the dish image area to the area of the oriented bounding box (called O_EXT).

This final approach is an improvement of Approach 1, described in Section 3.2. By adding one more metric, we were able to successfully identify all types of dishes in our set. The area of the dish image, which is already available from the pre-processing step, is a fairly good indicator of dish size. Recall that the O_REC ratio represents how similar the dish image shape is to a square. The O_EXT ratio presents how closely the dish image resembles a rectangle. All three shape descriptors are position invariant. Also from experiments with our pre-processing results, which produced binary dish images as the input of the identification phase, we found consistency under varying light conditions. Therefore, we postulated that our identification algorithm would produce consistent results with varying dish positions beneath the camera and with varying illumination.

In order to classify dish types, we used a training set of 500 images, with 100 images for each dish type, to estimate the distributions of dish image properties. Examining the dish image area property, we observe from Fig 3.8a and Fig 3.8b that two groupings readily appear: large areas representing LC and LP, and small areas representing, SC, SP and SX.

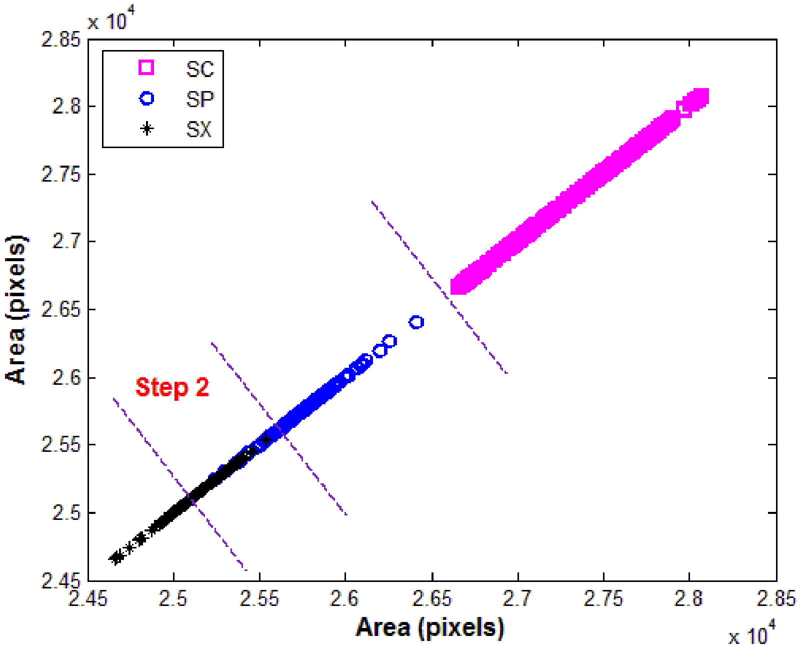


Figure 3.8a: Areas Distribution of Small Dish Images.

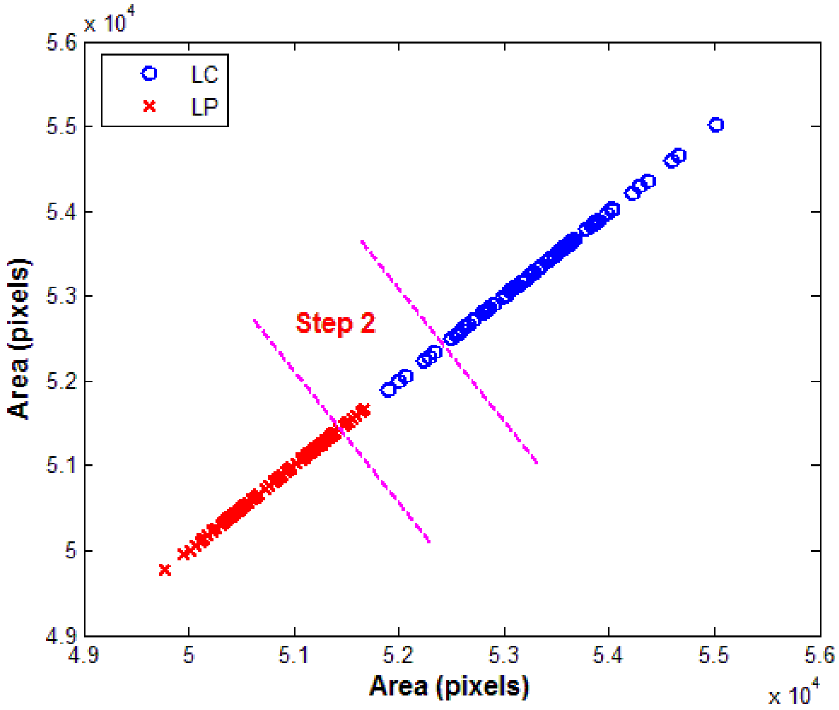


Figure 3.8b: Areas Distribution of Large Dish Images.

By considering only dish image area information, Table 3.1 shows that it is straightforward to identify SC, since none of the image areas of SC overlap with any other dish piece.

Table 3.1: Areas of Dish Images.

Dish type	SX	SP	SC	LP	LC
Area (10^4 pixels)	2.45- 2.58	2.53- 2.66	2.69- 2.83	4.96- 5.28	5.16- 5.51
Overlap region	2.53- 2.58 SX or SP?		x	5.16- 5.28 LP or LC?	

However, there clearly is overlap in image areas of LP and LC, and of SP and SX. Accordingly, we employ the other metrics, the O_REC ratio and O_EXT ratio, to distinguish between them. Using these two metrics together, as indicated in Fig 3.9a, we can easily distinguish SP from SX. Using area and O_REC ratio, Fig 3.9b illustrates how

LC is distinguished from LP.

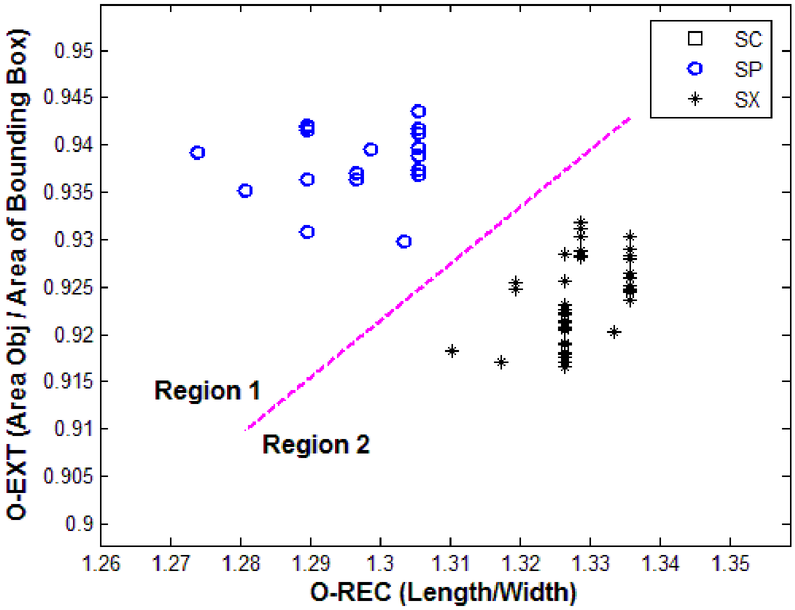


Figure 3.9a: Using the O_REC and the O_EXT to Distinguish SP from SX Dish Types

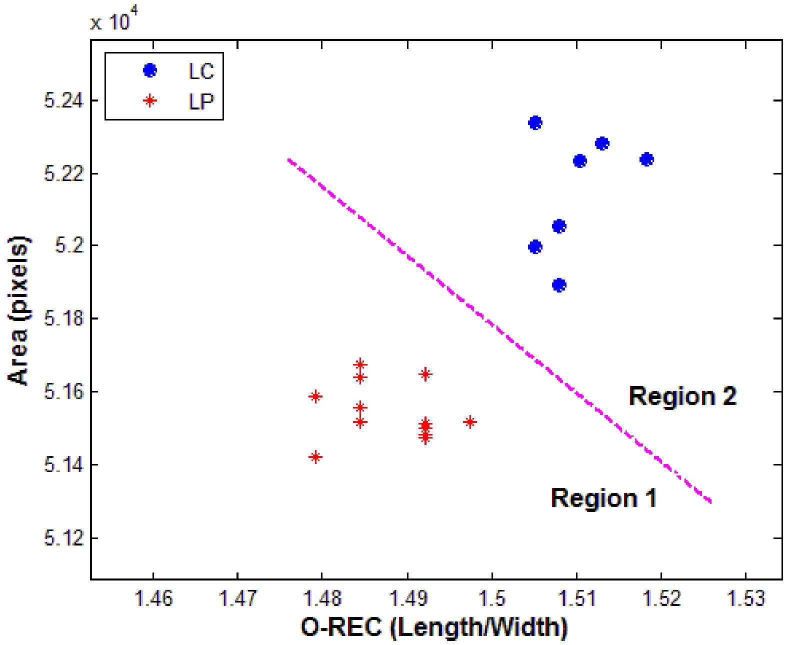


Figure 3.9b: Using the O_REC and the Area to Distinguish LP from LC Dish Types.

The “optimal lines” to separate SP and SX, and LC and LP are shown in Table 3.2. One could be able to employ a simple Single-layer Neural Network to automatically learn the optimal linear boundary of data points from two separable data groups, in the sense of minimum square error (Hagan 2004). However, it is simple enough in our case, with a two-dimensional space, that these “optimal lines” could be manually determined. Regardless of how these lines are determined, manually or automatically, they should be pre-calculated in a “training phase” as we did here, and then parametrically coded into the user program for real-time use.

Table 3.2: Linear Separation of SP and SX, LC and LP.

	Line to separate SP and SX		Line to separate LC and LP	
	Point 1	Point 2	Point 1	Point 2
Area (10^4 pixels)	x	x	5.22	5.13
O_REC	1.28	1.35	1.48	1.52
O_EXT	0.91	0.95	x	x

Hence, our identification algorithm is as shown in Table 3.3 and Figure 3.10:

Table 3.3: Main Steps of Identification Algorithm.

1. (Pre-processing): retrieve dish image as the largest object in camera image.
2. Step 1: Classify using dish area. Identify SC.
3. Step 2: Distinguish SP from SX using O_REC and O_EXT. Distinguish LC from LP using O_REC and area.

To save time, the two properties, O_REC and O_EXT, are computed only if the area property is insufficient to make a good decision. Otherwise, the algorithm stops at Step 1 shown in Table 3.3.

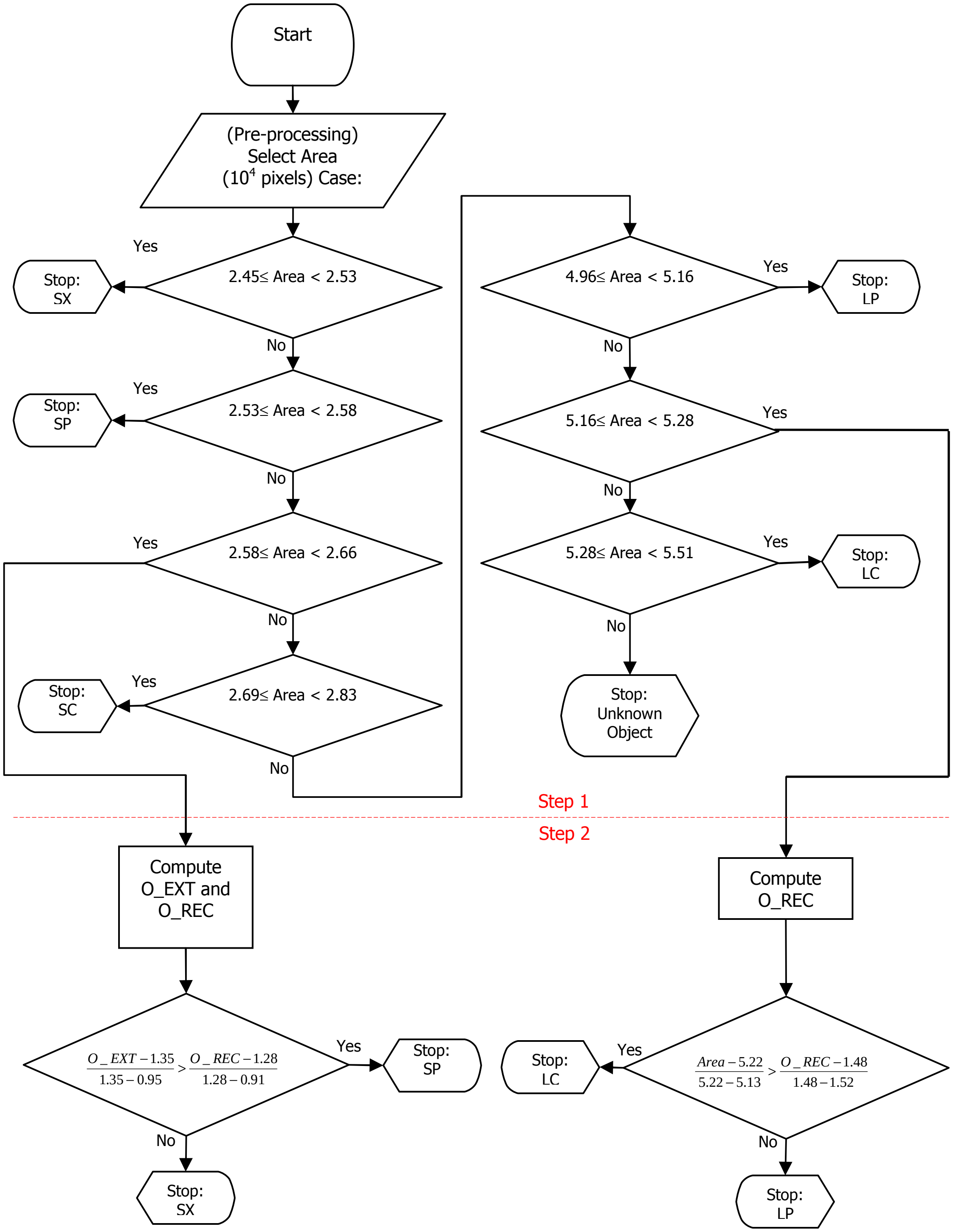


Figure 3.10: Identification Algorithm Flow Chart

3.4 Identification Results

Results were collected from 725 images of all types of dish pieces, not including any of the 500 training set images. All training and testing image sets were produced from 84 dishes, of all types, clean and dirty, under different dish positions and orientations under the camera axis. The camera exposure time varied between 16µs and 24µs.

Table 3.4: Results of Dish Identification.

	Number	Correct	Time* (sec)		
			Min	Average	Max
LC	85	100%	0.18	0.22	0.57
LP	120	100%	0.18	0.33	0.59
SC	200	100%	0.17	0.18	0.23
SP	167	100%	0.17	0.20	0.49
SX	153	100%	0.16	1.24	0.48
All dishes	725	100%	0.16	0.21	0.59

(*) Matlab® R14, Image Processing Toolbox V5.0, Window Vista, dual core 1.6GHz, 2GB RAM.

The results shown in Table 3.4 show accurate identification for all images, with an average computation time of 0.21 sec. This is deemed acceptable to allow identification and inspection of dishes at our target dish processing rate of 30 dish pieces per minute (2 secs/dish).

In the next chapter we present automatic methods to inspect dish pieces for cleanliness, given that they have already been identified using the algorithm in Fig 3.10.

CHAPTER 4

DISH INSPECTION

4.1 Introduction

Following commercial dishwashing, automated dish inspection using image processing presents some unique challenges. First, the intensity of dish images is sensitive to changes in lighting, power fluctuation, and camera sensitivity drift (Lolla, 2005). Second, even with reasonable attempts to establish uniform illumination of dish pieces, uneven illumination persists in the camera field of view. This causes non-uniformity in color and gray intensity across a clean dish as the position of the dish varies in the field of view. Third, because of the non-flat geometry of the dish surface, the gray intensity of the image drops significantly at the dish wall, especially for a deep dish with steep sidewalls, such as our dish pieces LC, SC and SX. Moreover, glare and shadows increase the difficulty of discerning clean from dirty dishes, even for human manual inspection. Fourth, food particle images vary in gray level, depending on food type, size, and location. Certain food particles, such as dried egg yolk, can be especially difficult to detect against the color of our dish. Fifth, the definitions of a “clean dish” and a “dirty dish” are subjectively ill-defined (Zhou, 2008). Comparing with many industrial identification tasks, automatic dish inspection can be much more interesting and challenging.

Zhou (2008) proposed a fusion based technique for silverware inspection. The most important part of his technique was pre-processing, in which he combined relevant information from two images of one silverware piece captured at different positions, to recover information in regions obscured by lighting glare and shadows. Zhou also applied simple global thresholding to the three color (R, G and B) channels, and his approach worked well for silverware. Lolla (2005) used template edge matching for silverware inspection. This approach is not only time consuming, but also suffers from lacking the ability to deal effectively with glare and shadows. We investigated several new approaches for dish inspection, as described in what follows.

4.2 Trial Approaches

Approach 1: Employ color segmentation with Mahalanobis distance and Cosine angular similarity.

The idea of this approach is detecting any spot (corresponding to dirt) in a dish image that is different in color from the dish surface's image color. Remember that dishes in our set are all of uniform color, with no decorative markings. The Mahalanobis distance method and the Cosine method were employed for measurement of color similarity in RGB space.

The Mahalanobis distance (M-distance), introduced by the Indian scientist and applied statistician Prasanta C. Mahalanobis (Gonzales 2007), is a distance measure using correlations between variables by which different patterns can be identified and analyzed. Applied to the RGB model, the M-distance $D_M(X)$ from a point $X = [X_{R,} X_G, X_B]^T$ in the RGB color space to a group of n sample points

$\tilde{X}_i = \left[\tilde{X}_{iR}, \tilde{X}_{iG}, \tilde{X}_{iB}\right]^T$, $i = \overline{1:n}$, represented by a mean $\mathrm{M} = \left[\mu_{R,} \mu_G, \mu_B\right]^T$ and a covariance matrix C, is defined as follows (Gonzalez 2007).

$$D_M(X) = \sqrt{(X - M)^T C^{-1} (X - M)} \qquad (4.1)$$

where $(.)^T$ means the transpose of a vector or a matrix, and $(.)^{-1}$ means the inverse of a (non-singular square) matrix. The covariance matrix C is a matrix of covariances between elements of the vector $(\tilde{X} - M)$.

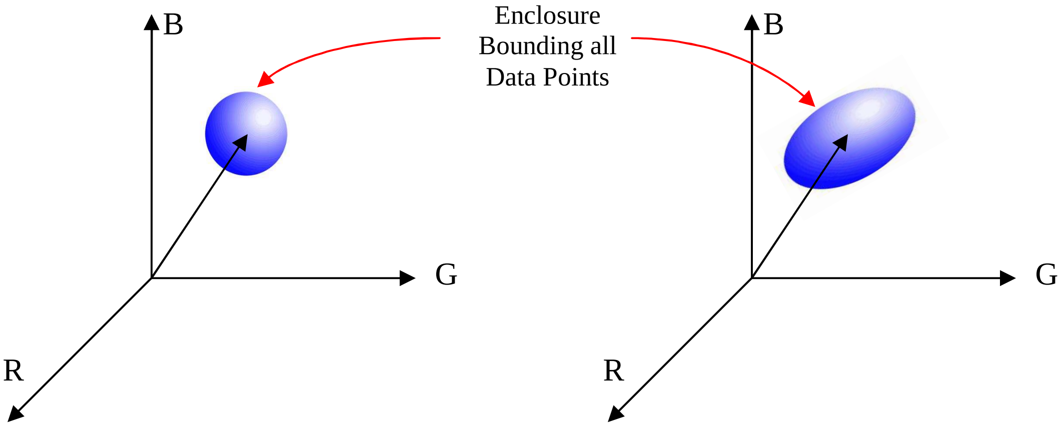


Figure 4.1: Enclosures of Data in RGB Space Using Euclidean (left) and Mahalanobis (right) Distances.

The M-distance differs from Euclidean distance in that it takes into account that data points (from one object) may have their values in one color channel correlated with the values in one or both of the other color channels (as illustrated in Figure 4.1). If the covariance matrix is the identity matrix, the M-distance reduces to the Euclidean distance.

Another method used between the two points X and M in RGB color space was the Cosine method, with the cosine defined by Lukac (2007) as:

$$\cos\theta = \frac{X^T M}{\|X\|.\|M\|} \tag{4.2}$$

where the operation ||.|| means the Euclidean norm of a vector, or the Euclidean length of the vector (in our implementation).

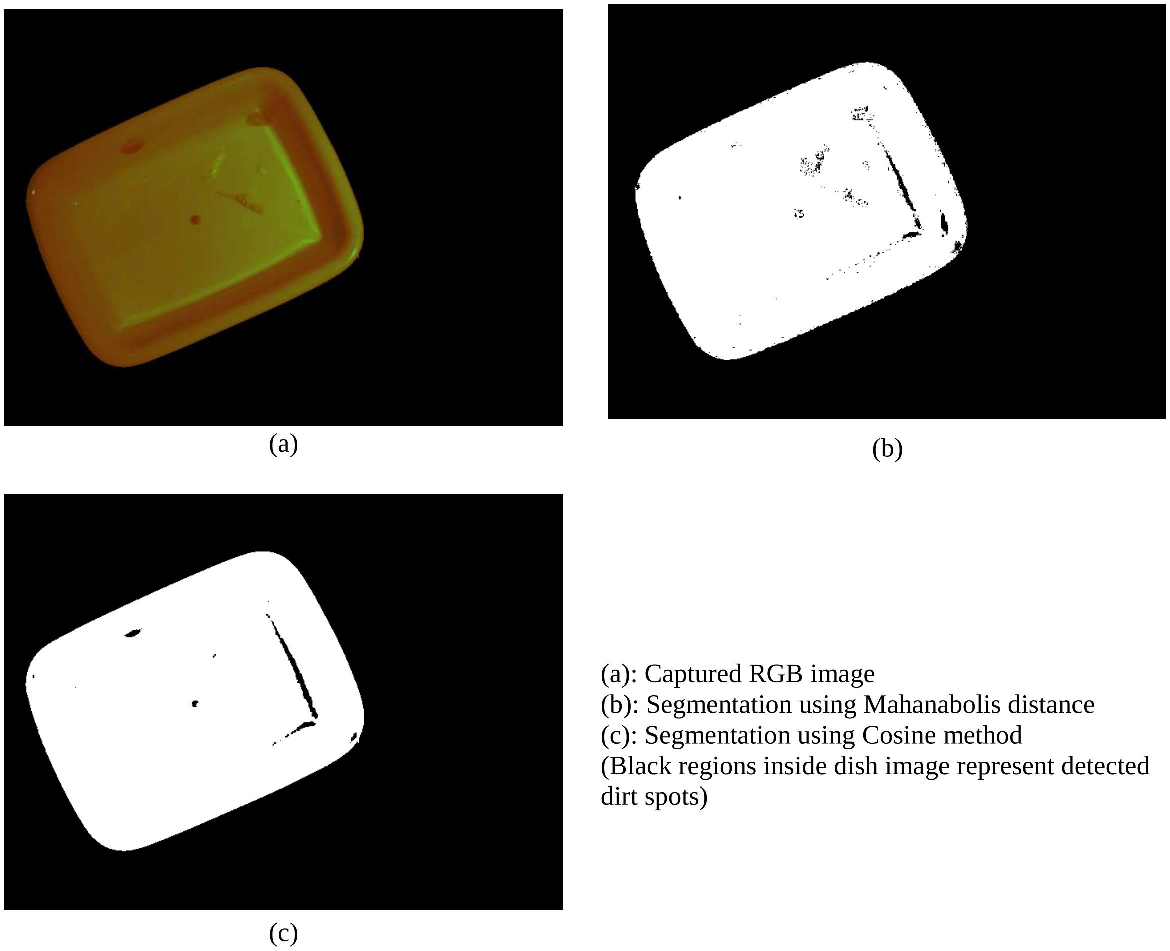
In our inspection process, a sample image of a clean dish of interest (of the same type that was earlier identified in the identification process) was used for sample points to calculate vector M and matrix C in (4.1) and (4.2). Then the dish image of interest was segmented based on M-distance and/or Cosine similarity using the following criteria:

$$[D_M(X)]^2 \leq d \text{ , for D-distance} \tag{4.3a}$$

$$255\sqrt{1-\cos^2\theta} \leq d \text{ , for Cosine} \tag{4.3b}$$

where d is a distance threshold value, normalized in the range [0,255].

Results of the color segmentation methods, illustrated in Fig 4.2 for an SX dish image, show that the M-distance result is more sensitive to color variations, as well as more susceptible to noise, compared with that of Cosine result. Notice that results of both methods miss-classify the glare, or high intensity region on the right of the dish image, as a dirt spot. Even worse, the results using these approaches varied with varying dish location as well as varying lighting.



(a)

(b)

(c)

(a): Captured RGB image
(b): Segmentation using Mahanabolis distance
(c): Segmentation using Cosine method
(Black regions inside dish image represent detected dirt spots)

Figure 4.2: Result of Approach 1 for a Dirty SX Dish Images.
(Dirt was made of BBQ sauce, egg yolk, coffee stain
which were manually placed onto dish surfaces and dried in place)

Approach 2: Employ a Fusion-based method for pre-processing combined with a Color-based edge detection method.

This approach was motivated by the method used by Zhou (2007), combining two images of a silverware piece captured at different positions for eliminating glare and shadows. However, applying a simple global thresholding operation to three color RGB channels of the fused dish image, as the second step of Zhou’s technique suggested, did not provide a good result. One reason is that the B channel of a dish image contains much less information comparing with the other two channels because: 1) the dish color itself lacks a blue component; and 2) the CCD camera sensor is always less sensitive to blue color (Siegwart 2002, p119). More importantly, the 3-dimensional geometry of our dishes caused significant variation in gray level of dish images even under "ideally" uniform illumination. Accordingly, dirt spots in the floor region of a dish image might have greater intensity than the clean wall region. Therefore, a single global threshold value could not successfully distinguish dirt from the whole dish image.

In this approach, after the fusion process, the color-based edge detection technique was employed to the fused dish image to discover regions where the color changed significantly. The variation of color was measured directly by the color-gradient, defined by Gonzalez (2004, section 6.6) as follows:

Define: $\nabla R = \begin{bmatrix} R_x \\ R_y \end{bmatrix}$, $\nabla G = \begin{bmatrix} G_x \\ G_y \end{bmatrix}$, and $\nabla B = \begin{bmatrix} B_x \\ B_y \end{bmatrix}$ as the gradient in R, G, and B channels, respectively. Define:

$$g_{xx} = R_x^2 + G_x^2 + B_x^2 \tag{4.4a}$$

$$g_{yy} = R_y^2 + G_y^2 + B_y^2 \tag{4.4b}$$

$$g_{xy} = R_x R_y + G_x G_y + B_x B_y \tag{4.4c}$$

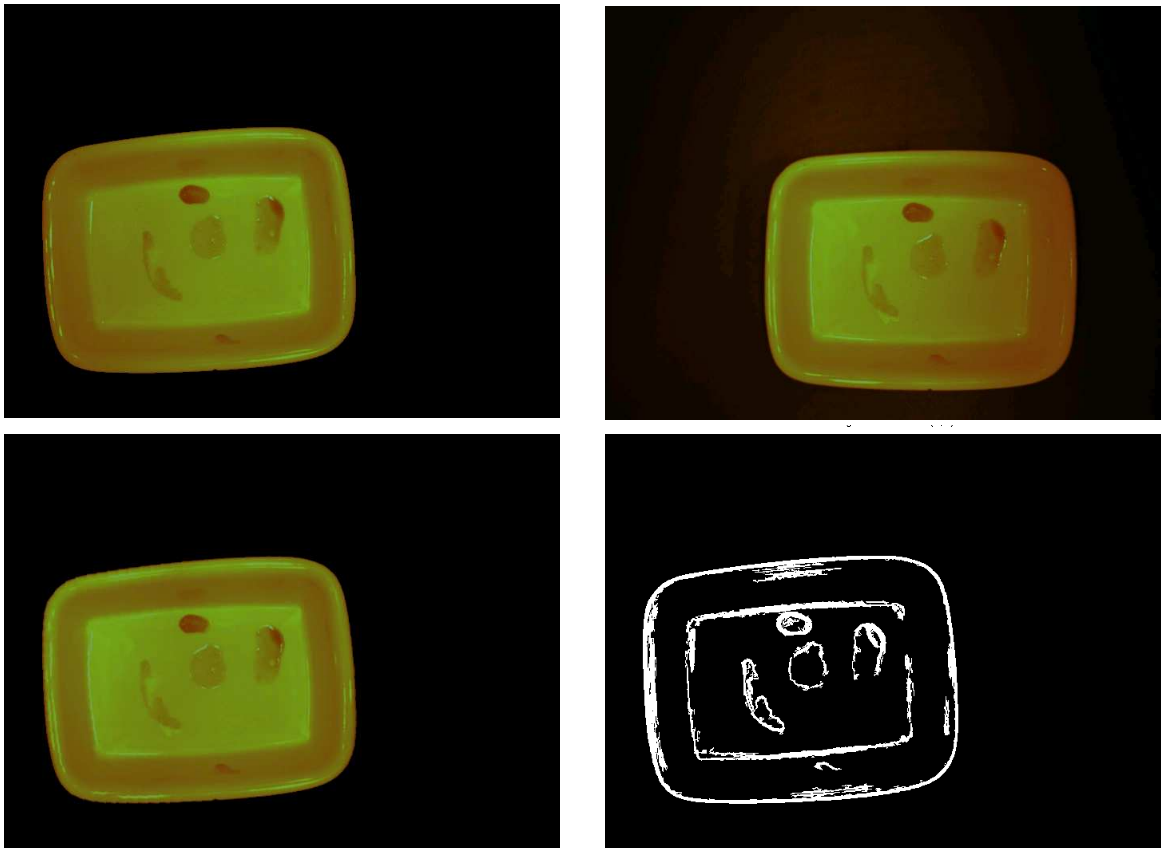
Then the angle ψ and gradient ∇ at each point (x,y) in the RGB input image are respectively calculated by:

$$\psi(x,y) = 0.5\tan^{-1}\left(\frac{2g_{xy}}{g_{xx} - g_{yy}}\right) \tag{4.5a}$$

$$\nabla = 0.5\lfloor g_{xx} + g_{yy} + (g_{xx} - g_{yy})\cos 2\psi + 2g_{xy}\sin 2\psi \rfloor \tag{4.5b}$$

The edge is then detected at locations whose gradient ∇ greater than some threshold value. The results of fusion for the two sample SC images and of color-based edge detection in the fused image are shown in Fig 4.3. Notice how the glare and shadows in the two original images (a) and (b) was reduced in the fused image (c), which approached more uniform color and intensity distribution. The edge detected image (d) contained edges of dirt spots and edges of dish geometry, as well as edges caused by glare. This result could be improved by employing addition processing to remove the dish geometry edges, which are known.

However, fusion and color-based edge detecting processes themselves were so computationally expensive that we did not develop this approach further.



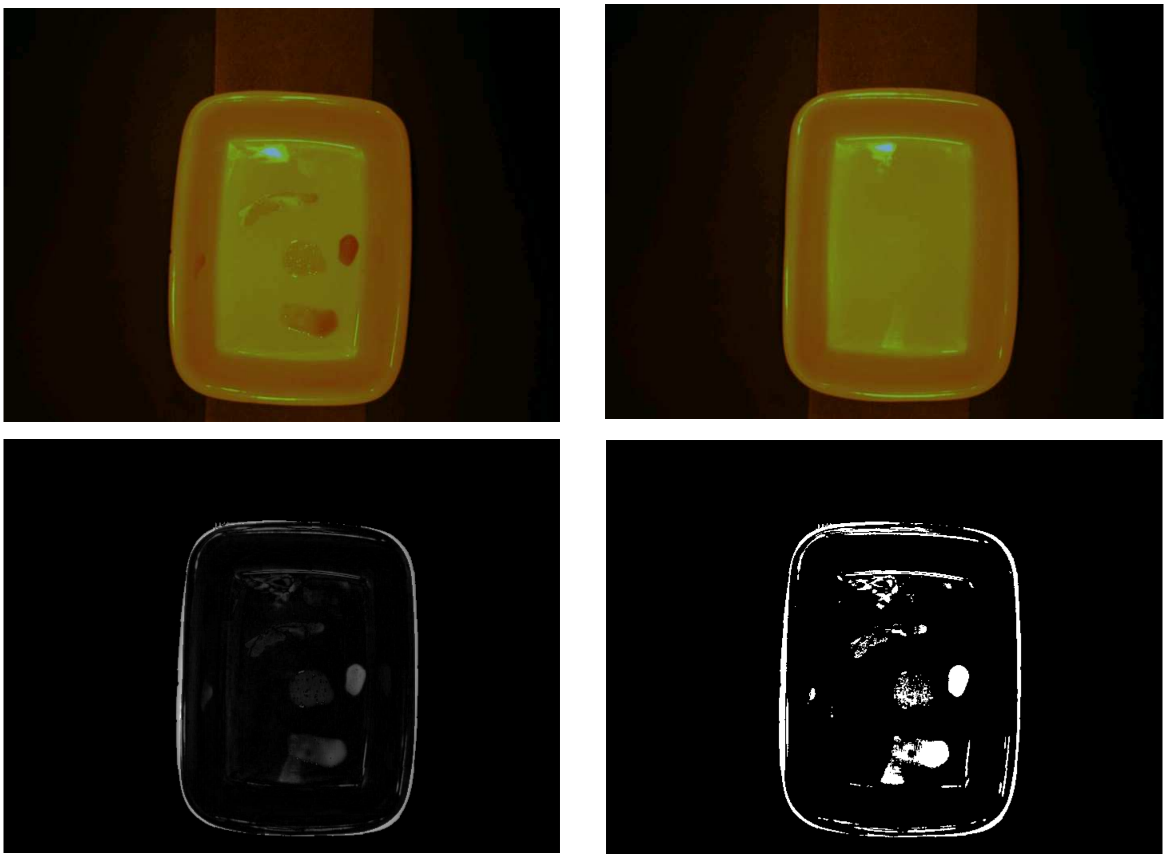
(a)	(b)
(c)	(d)

(a): Captured RGB image of a SC dish at position A
(b): Captured RGB image of the same SC dish at position B
(c): After fusion of two images to position A
(d): After color-based edge detecting

Figure 4.3: Result of Approach 2 for a Dirty SC Dish Images.
(Dirt was made of BBQ sauce and potato sauce,
which were manually placed onto dish surfaces and dried in place)

Approach 3: Use the template of a clean dish.

Similar to the fusion technique, this approach employed image registration technique between a dish image for inspection (called the target image) and a known clean dish (called reference image). Registration is the process of transforming the different sets of images into a single coordinate system to align the dish images. This process is necessary in order to be able to compare or integrate the images obtained from different scenes or sources. In our feature-based registration implementation, the centroids, or the geometric centers, and the orientations of the major axes of the two dish images were calculated and used as the key features. And then simple linear spatial transformations, including rotation and translation, were applied to align the target image with the reference image.



(a)	(b)
(c)	(d)

(a): Captured RGB Image of a Dirty SC (Target Image A)
(b): Captured RGB Image of Clean SC (Reference Image B) at Almost the Same Position as Target Image A.
(c): Differences between Image A and Image B after Registration.
(d): Results after Thresholding, with White Regions Interpreted as Dirt Spots

Figure 4.4: Results of Approach 3 for a Dirty SC Dish Image.
(Dirt was made of BBQ sauce and potato sauce,
which were manually placed onto dish surfaces and dried in place)

Conceptually, any differences between the two dish images, after registration, could be considered as dirt spots (as shown in Figure 4.4). However, the result was highly affected by noise and registration error. Even when the reference dish image was captured under the same lighting with nearly the same position as the target dish image, the differences between their ‘backgrounds’, or the clean portion of dish images, were significant. And those differences were indistinguishable from the true dirt spots in the target dish image. Finally, this approach (and any others that involved multiple images) is quite time consuming.

Approach 4: Multi-scale edge detection.

The most difficult problem for any inspection procedure is how to distinguish true defects from glare or shadows in the image. Approach 4 is based on the assumption that the variation of the intensity caused by defects, or dirt spots in our application, is sharper than the variation caused by glare and shadows. In other words, dirt spots are more inconsistent with dish background than are glare and shadows. Therefore, if we examine the detected edges of the image at different scales the edges of dirt spots should remain at the same relative position, while those of glare and shadows change (illustrated as Fig 4.5).

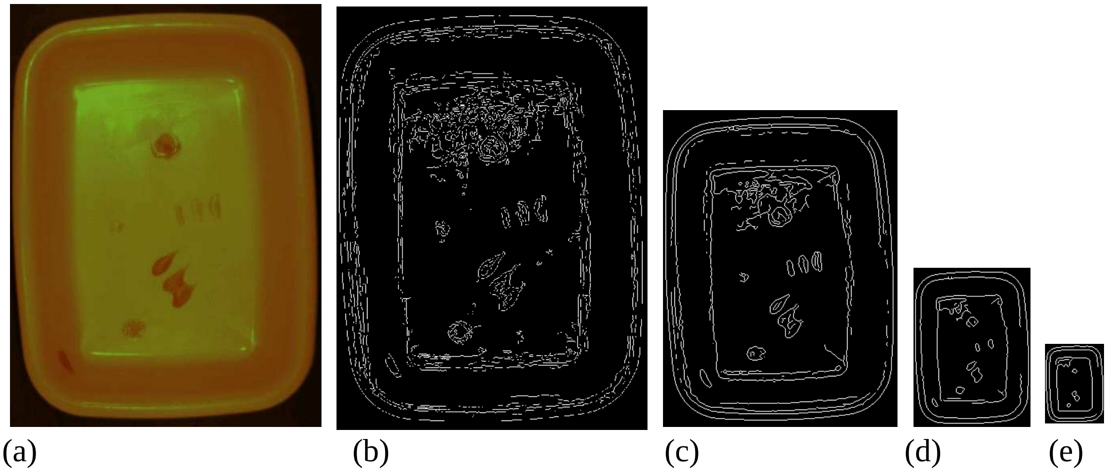


Figure 4.5: Results of Approach 4 for a Dirty SX Dish Image
(a) An Original Dirty SX Image; Its Detected Edge at Multi-scales
(b)=Full scale; (c)=50% Reduction; (d)=25% Reduction; (e)=12.5% Reduction
(Dirt was made of BBQ sauce, coffee stain, and potato sauce,
which were manually placed onto dish surfaces and dried in place)

The advantage of this approach is ease of developing an algorithm that could automatically determine the correct scale to examine at the image. However, the algorithm is so computationally time-consuming that it would be not suitable for real-time processing.

4.3 Final Approach and Inspection Algorithm

Our final approach, which leads to our inspection method, was inspired by observing how humans inspect a dish piece. Human eyes always focus locally, and are very sensitive to relative chances in color or intensity. The human visual system also easily learns, and appears to eliminate glare and shadows, as well as adaptively adjusts the background intensity.

In our earlier experiments, the most difficult problem arose from the unique 3D geometry of the dishes which caused significant differences in clean dish background gray level between the floor region and the wall region. One approach we considered was to create targeted illumination on the dish walls based on their incline angle, and then apply global thresholding to the entire modified image. The problem is that modifying illumination of the wall would be expensive and time consuming, and difficult to adapt to inside corners. A more promising approach was to treat the dish floor region and dish walls regions differently during image processing. This is the essence of our proposed method, in which we investigate partitioning and adaptive thresholding, which holds promise of simplicity and efficiency.

In our inspection algorithm, a “dirty spot” is defined as a connected component that is (1) dark enough, and (2) darker than the immediately surrounding area. The algorithm first separates the dish image into a dish floor region and a dish wall region. Then for each region, we automatically locate all components that are both darker than the near surrounding area (employing adaptive thresholding) and dark enough (employing global thresholding). Without global thresholding, regions next to glare in dish images could be considered as dirt spots, because they are darker than neighbors. After these steps, we combine the floor and wall regions and carry out post-processing to remove “tinny” spots, which are most likely produced by noise rather than true dirt spots. The steps of our inspection algorithm are as illustrated in Table 4.1 and Figure 4.6.

Table 4.1: Main Steps of Our Inspection Algorithm.

1. Identify the dish piece using the method presented in Chapter 3, so that we have a template for partitioning.

2. Start from a gray image of the identified dish piece which is the result of pre-processing.

3. Partitioning: Detect and separate floor region and wall region of a dish image by using the appropriate floor template image.

4. Adaptive and global thresholding: work with the floor region and wall region separately. Use adaptive thresholding to find potential spots; use global thresholding to retain only those with gray levels less than the global threshold.

5. Combine the two regions.

6. If total area of all spots greater than some value, the dish will be classified as dirty. Otherwise it will be classified as clean.

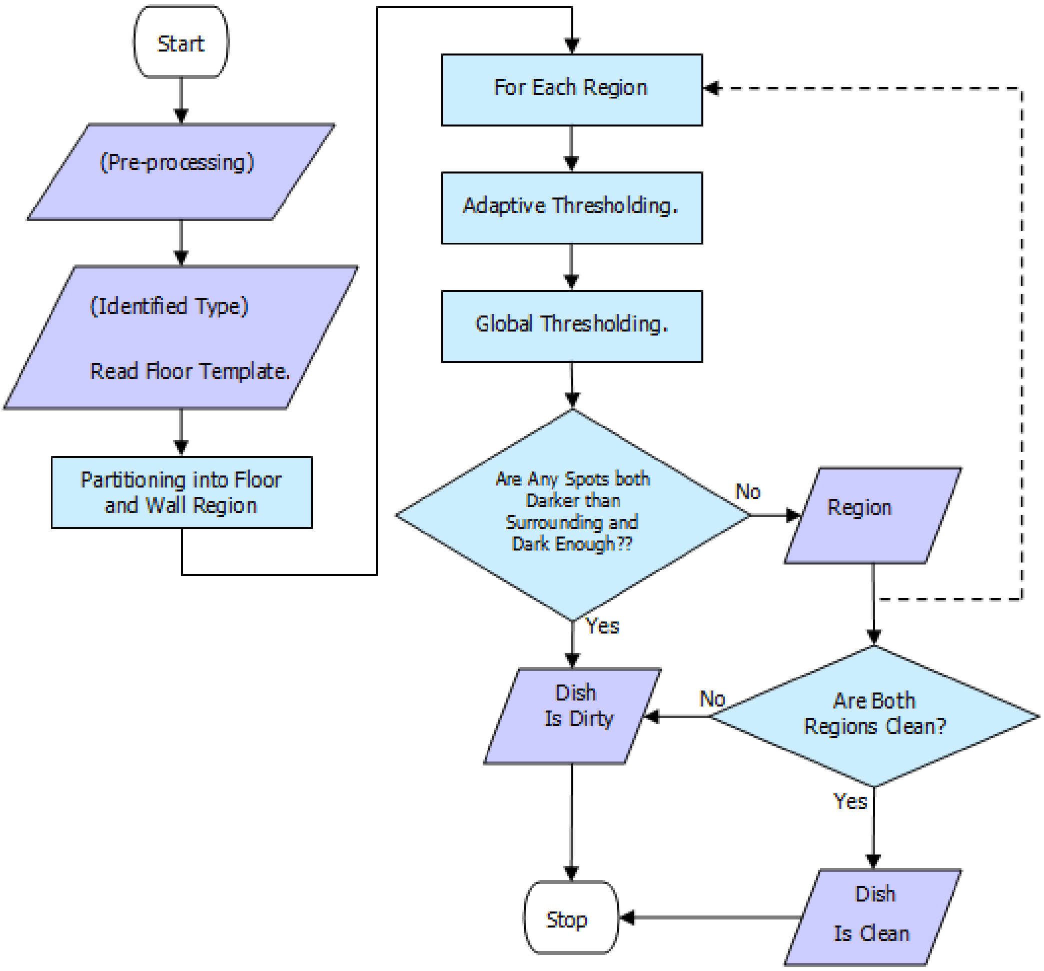


Figure 4.6: Inspection Algorithm Flow Chart

Thresholding

As defined in Chapter 2, thresholding is the simplest segmentation process that divides an image's (color or gray-scale) pixels into background pixels or object pixels. According to Sezgin (2004), thresholding methods can be categorized based on the information the algorithm manipulates, such as: histogram shape-based methods, entropy-based methods, clustering-based methods, and local or adaptive methods. In the following section, we consider a commonly used automatic global thresholding algorithm and a commonly used automatic adaptive thresholding algorithm.

Auto global thresholding

One simple way to achieve an auto global thresholding is using iterative technique, which is a clustering-based method. The iterative algorithm is described in Table 4.2 (Sharpio 2002).

Table 4.2: Iterative Auto Global Thesholding Algorithm (Sharpio 2002).

1. Choose an initial threshold (T), either randomly or according to any other methods.

2. Use T to segment the image into object region G_1 and background region G_2.

$G_1 = \{f(x,y):f(x,y)>T\}$, (object pixels)

$G_2 = \{f(x,y):f(x,y)\leq T\}$,(background pixels)

where f(x,y) is the intensity of the pixel at location (x,y).

3. Calculate M_1 = mean(G_1) and M2 = mean(G2).

4. Update new threshold value: $T = 0.5*(M_1+M_2)$.

5. Repeat step 2 to 4 until the change of T from one iteration to another is sufficiently small.

This iterative method was shown to converge to a local minimum, meaning that a different initial threshold might give a different final result (Sezgin 2004). In our implementation, the global thresholding value T was manually set as shown in Table 4.5, described later.

Adaptive thresholding

Unlike the conventional thresholding operator, which uses a single global threshold value for all image pixels, adaptive thresholding calculates threshold values dynamically over the image based on local information. Therefore, this method can accommodate changing background in the image, caused by, for examples, non-uniform illumination or shadows. There are two main approaches to finding the adaptive threshold values: 1) the Chow (1972) approach and 2) local thresholding (Fisher 2003). Both methods assume that smaller image regions are more likely to have approximately uniform illumination, in which the thresholding operation could be easily employed. Chow divides an image into sub-images and then employs an histogram-based method to find the threshold for each sub-image. While this method produces desirable results in general, it is computationally expensive such that it is unsuitable for real-time application. An alternative approach for finding an adaptive threshold, which is simpler and used in our implementation, is to statistically compare the intensity value of each pixel with each of its neighboring pixels. Among commonly used statistical indicators, we used the arithmetic mean of pixel gray levels for comparing. In searching for dark sports in a dish image, we set all pixels to the background intensity, whose intensities are greater than the mean C gray levels, where C is a constant in the range [0,1]. There are two parameters in our adaptive thresholding function, namely the kernel (or window) size and the constant C. A larger window yields results that are more adversely affected by the illumination gradient. The results of using a smaller window size are more easily corrupted by noise. Also note that as window size is increased, more computational time is required. Our adaptive thresholding algorithm is shown in Table 4.3 (Fisher 2003).

Table 4.3: Adaptive Thresholding Algorithm (Fisher 2003)

1. Set the window size w. For all image pixels, compute the arithmetic mean of gray levels of that pixel and its neighbors (which are inside the window with size w and centers at that pixel).

2. If a pixel gray level is less than the corresponding computed mean value minus a constant C, set that pixel to the foreground (1 value). Otherwise, that pixel belongs to the background (0 values).

Partitioning:

We define partitioning as a process that divides a dish image into two regions, namely the dish floor region and the dish wall region, based on the dish geometry. There are two approaches to achieve partitioning: 1) employ an intensity-based method, and 2) employ a manually-derived dish floor template image. The first method automatically divides the dish image by applying an appropriate threshold to pick out the floor region (having higher intensity), and then fill any holes inside that found region. The second method use a pre-captured or pre-calculated floor template (each dish type has a corresponding floor template) as a mask to pick out the floor region. We chose the second approach, using manual dish floor template, for our partitioning process not only because it requires less computation and is therefore faster, but also because it produces better results than the intensity-based method. Fig 4.7 illustrates the differences in results from using the two partitioning methods. Notice the dirt spot located in the lower left corner of the image on the boundary between the dish floor region and dish wall region (inside the artificial red circle superimposed on images (c) and (e)), was not detected when using intensity-based partitioning. The reason is that the zigzag of the floor boundary, resulting from intensity-based partitioning, classified the dirt spot as belonging to the wall region, whose background intensity was not very different from that of the dirt spot. On the other hand, using the manually-derived floor template caused that dirt spot stand out from the dish floor region.

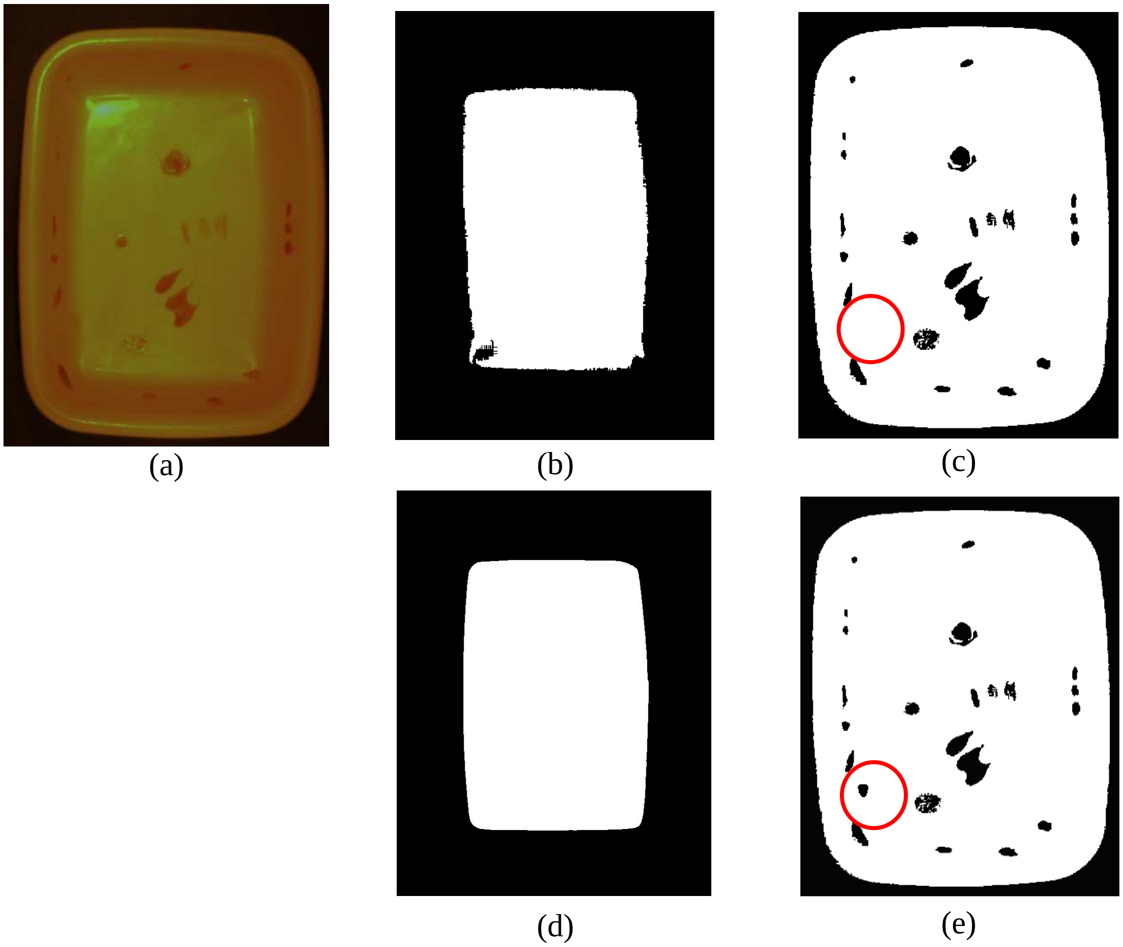
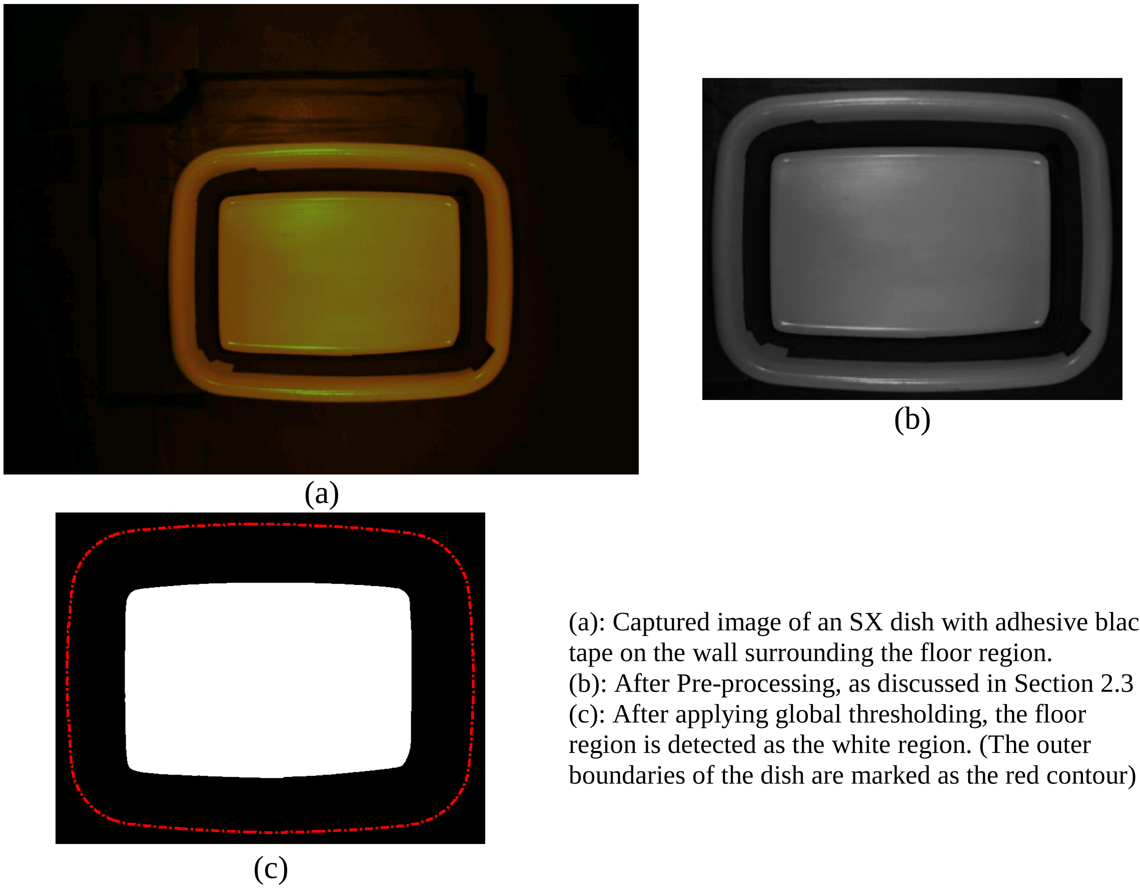

(a)
(b)
(c)
(d)
(e)

Figure 4.7: Comparing Results of the Two Partitioning Methods.
(Top): Using Intensity-based; (Bottom): Manual Floor Template Partitioning
(a): A dirty SX dish image; (b) Detected floor region and (c) Inspection results from intensity-based partitioning; (d) Manually-derived floor template region; (e) Inspection results of using (d).
(Dirt was made of BBQ sauce, coffee stain, and potato sauce,
which were manually placed onto dish surfaces and dried in place)

Manual partitioning was done by manually applying black adhesive tape on the wall surface of each dish type, and trimming the tape around the boundaries between the dish walls and the dish floors. Then the images of the "taped" dishes were captured, yielding good contrast between the floor and wall regions. Then, by employing the same pre-processing techniques as discussed in Section 2.3, the floor region was easily separated from the entire dish image. Notice that the images of the floor regions of each dish type have the same sizes and orientations (aligned with horizontal axis) with dishes images for inspection of the same type. Therefore, the floor templates (in binary images) could be directly used as masks to segment out the floor region of dish images for inspection. Figure 4.8 illustrates the process of creating the SX floor template.



(a): Captured image of an SX dish with adhesive black tape on the wall surrounding the floor region.
(b): After Pre-processing, as discussed in Section 2.3
(c): After applying global thresholding, the floor region is detected as the white region. (The outer boundaries of the dish are marked as the red contour)

Figure 4.8a: Process of Creating the SX Floor Template

Figure 4.8b shows the five floor template for each dish type of our set.

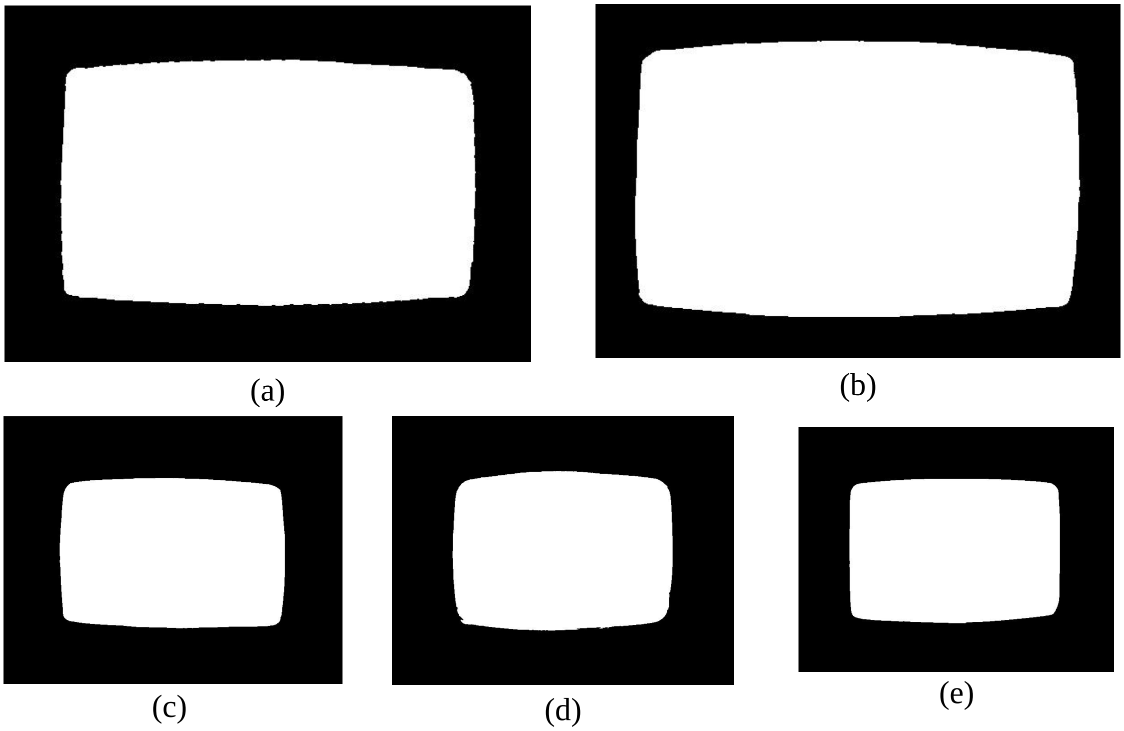
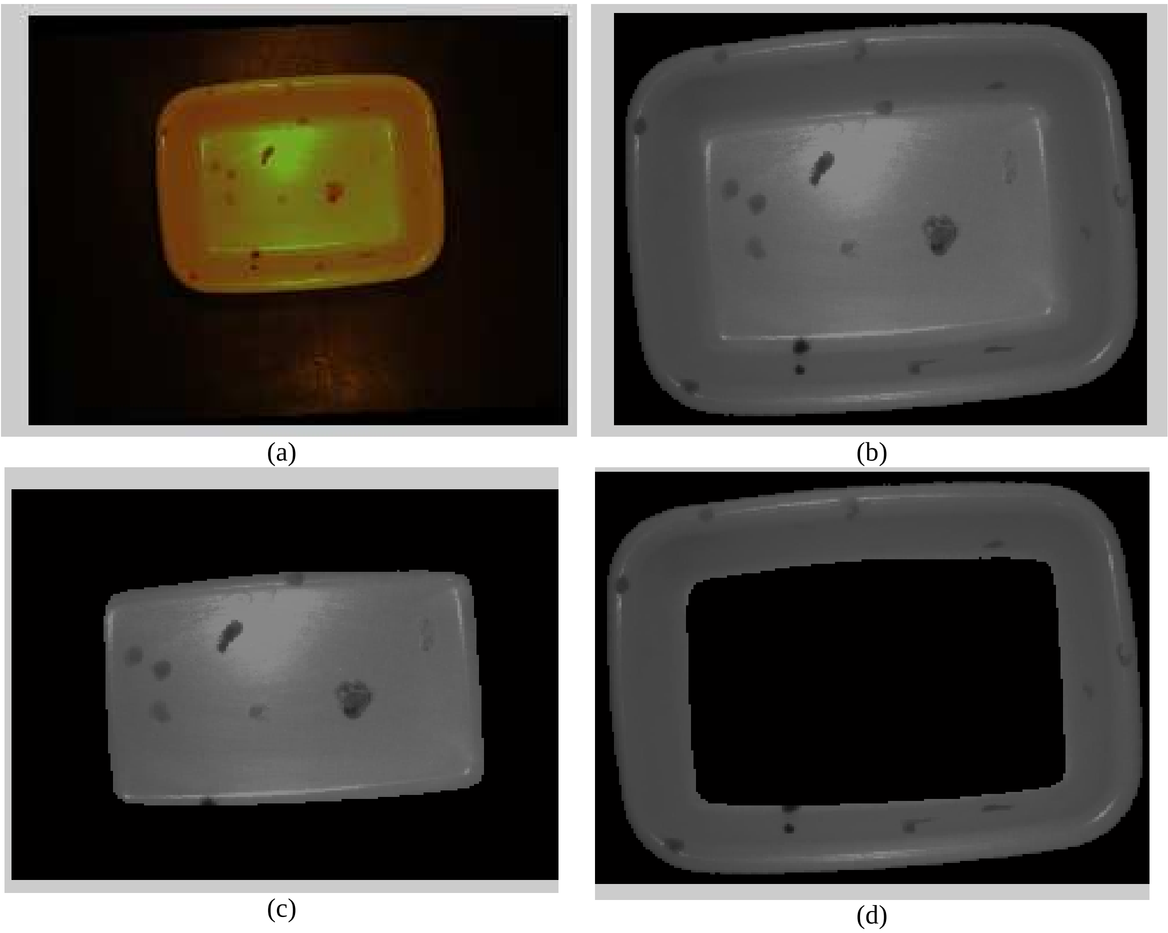


Figure 4.8b: Floor Templates of the Five Dish Type
(a) LC, (b) LP, (c) SC, (d) SP, and (e) SX.

Figures 4.9a and 4.9b illustrate our inspection process. Our experimental results showed that it is unnecessary to use a color image, such that even though we took color images with our camera, we employed only the corresponding gray scale image. Figure 4.9a shows a sample result of the partitioning process based on a dish floor template.



(a)

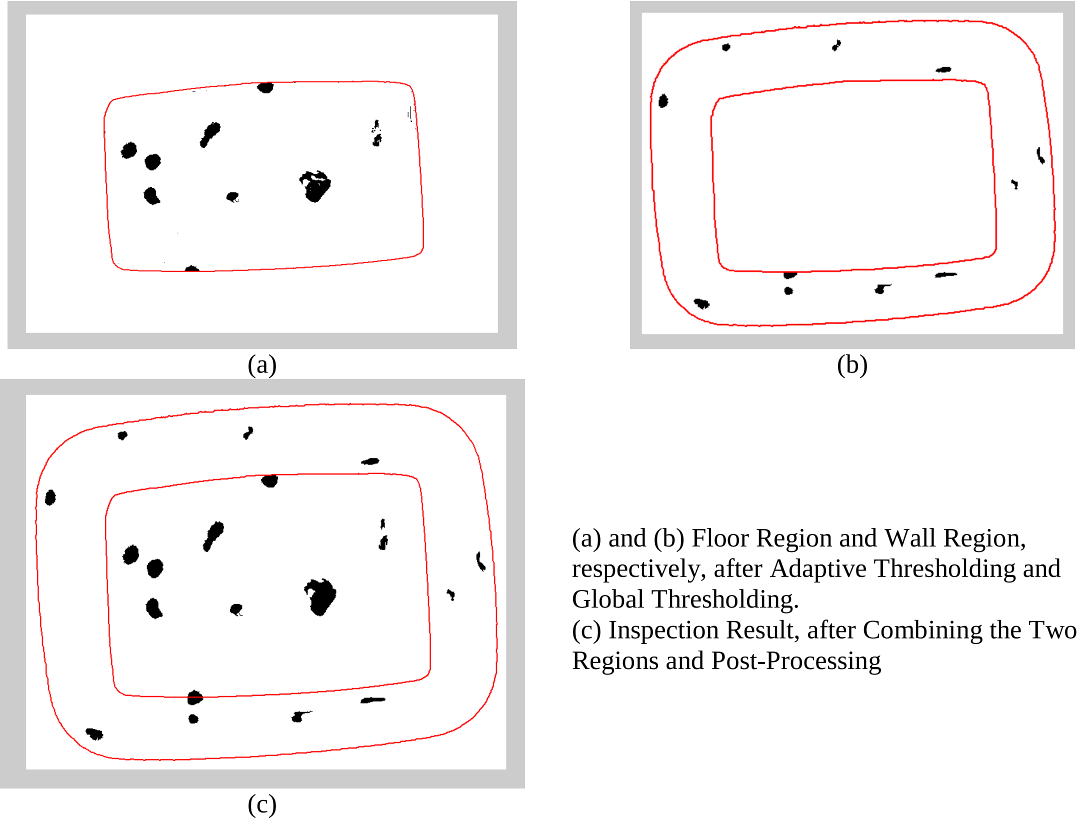
(b)

(c)

(d)

Figure 4.9a: Partitioning Process for a Dirty SC Dish Image
(a) Original camera image; (b) Gray scale image as the result of Pre-processing;
(c) and (d) Dish floor region and dish wall region, respectively, after partitioning.
(Dirt was made of BBQ sauce, coffee stain, and potato sauce,
which were manually placed onto dish surfaces and dried in place)

For the dish image in Figure 4.9a, results of adaptive thresholding combined with global thresholding for the floor region and wall region are shown in Figure 4.9b.



(a)

(b)

(a) and (b) Floor Region and Wall Region, respectively, after Adaptive Thresholding and Global Thresholding.
(c) Inspection Result, after Combining the Two Regions and Post-Processing

(c)

Figure 4.9b: Adaptive and Global Thresholding

For all tested images in the inspection process, the values of parameters for adaptive thresholding and global thresholding, found by trial and error, are given in Tables 4.4 and 4.5 respectively.

Table 4.4a: Adaptive Thresholding Parameters.

Dish Type	Dish floor region		Dish wall region	
	Window size (pixels)	Constant C value	Window size (pixels)	Constant C value
SC	60	0.03	30	0.05
SX	50	0.05	30	0.03
SP	50	0.05	12	0.03
LC	50	0.04	55	0.05
LP	20	0.04	60	0.05

Table 4.4b: Global thresholding parameters.

Dish Type	Dish floor region (intensity level: 0-255)	Dish wall region (intensity level: 0-255)
SC	86	70
SX	99	66
SP	99	54
LC	94	53
LP	80	40

Robustness of Thresholding Parameter Values:

Starting with the values listed in Table 4.4a and 4.4b, by changing (increasing or decreasing) one parameter value by 1%, 2%, 3%,… while keep others unchanged, until those changes affected the inspection results, we examined the robustness of the adaptive thresholding and global thresholding parameter values in Table 4.4a and 4.4b. Robustness result of global and adaptive thresholding parameter values of the SC dish are shown in

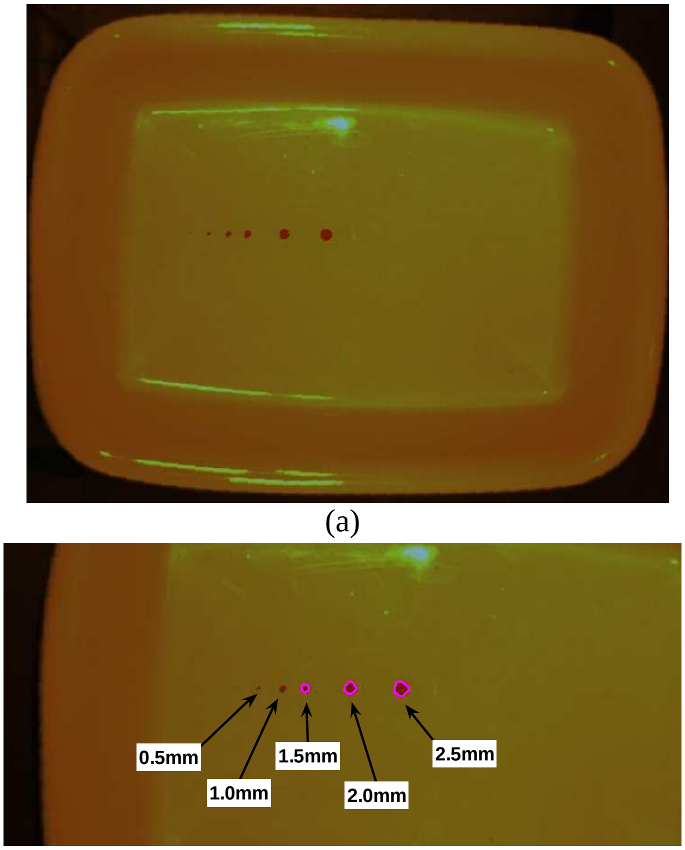
Table 4.5. The table shows the range of variation in each parameter for which inspection results were invariant. For the SC dish floor region, the range of allowable variance in parameter values for both global and adaptive parameters is small. Interestingly, however, for the SC wall region, the results indicate that the adaptive thresholding parameters show good tolerance to variation, while variation in the global thresholding value is much more sensitive in producing good inspection results. Stated another way, tuning values of adaptive thresholding for the wall region is much easier.

Table 4.5: Robustness of Global and Adaptive Thresholding Parameter Values of the SC Dish Type

Dish floor region			Dish wall region		
Window size (pixels)	Constant C value	Global thresholding value	Window size (pixels)	Constant C value	Global thresholding value
60 ± 5%	0.03 ± 4%	86 ± 3%	30 ± 12%	0.05 ± 10%	70 ± 3%

Smallest Detected Dirt Spot Size:

Theoretically, the smallest feature size SFS that our imaging system (camera and lens) has capacity to detect is 0.38mm (Section 2.1). This distance in the field of view corresponds to the distance between two adjacent receptor sensors in the CCD (Charge Couple Device) array. In other word, the distance between the two neighbor-pixels of a captured image under our configuration translates to 0.38 mm at our working distance the camera field of view. A simple test was conducted to verify the actual smallest size of dirt spots could be detected by our system. Artificial dirt spots, having the same color as BBQ sauce placed and dried in the dish surface, were manually drawn on a clean SC dish image using Photoshop® software (see Fig 4.10 (a)).



(a)

(b)

Figure 4.10: Smallest Detected Dirt Spot Size
(a) An SC dish image with artificial dirt spots (same color with the dirt made of BBQ sauce dried in the place) with diameter size from 0.5 mm to 2.5 mm
(b) (Zoom in) All dirt spots with diameter greater than 1.5 mm are correctly detected

Then this image with artificial dirt was inspected using our inspection algorithm. The results (Fig 4.10 (b)) show that all dirt spots with diameter greater than 1.5 mm were correctly detected. We note that the actual smallest detected size will vary depending on the color of dirt spots, or their position on the dish image, which determine the contrast between the dirt and the dish. If the "dirt" was perfectly black and the dish background immediately surrounding the dirt was perfectly white, our calculations in Eq 2.3 show that the smallest detectable dirt size would be 0.38 mm in diameter.

4.4 Inspection Results

To obtain experimental results with our proposed inspection method, we manually applied real food particles to a variety of our dish pieces, varying them in type of food, and size, shape, and location. We used egg yolk, fruit juice, and a variety of sauces, including tomato-based sauces, all of which were allowed to air dry before inspection.

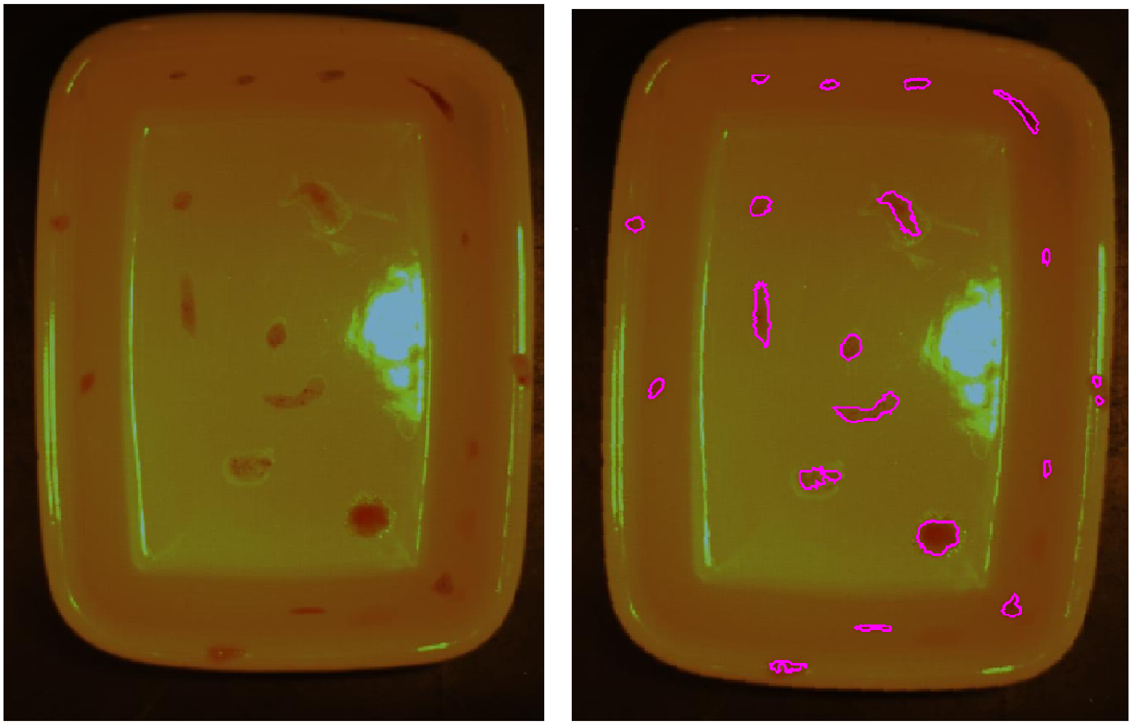


Figure 4.11: Original Dirty SC Dish (left) and All Dirt Spots Detected (right).

Figure 4.11 shows an example of results with the original SC dirty dish image on the left. Detected dirt spot boundaries (on the right) were superimposed on the original dish image. All dirt spots were correctly detected. Notice that the glare (specular reflection) due to the shiny surface of the ceramic dish did not produce spurious results.

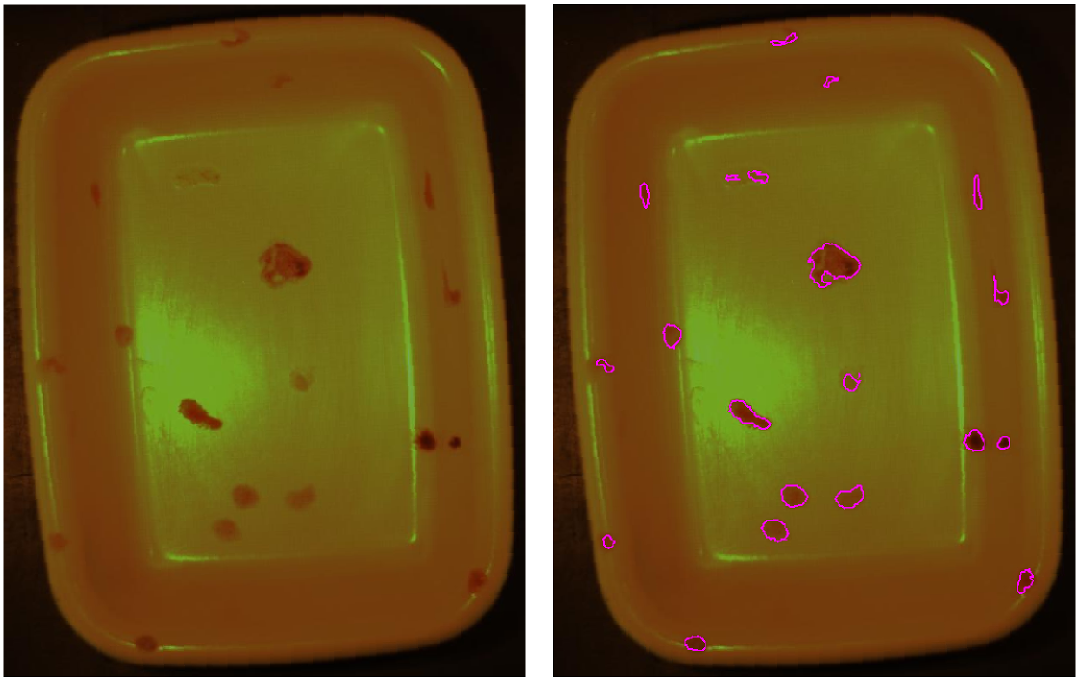


Figure 4.12: Original Dirty SX (left) and All Dirt Spots Detected (right).

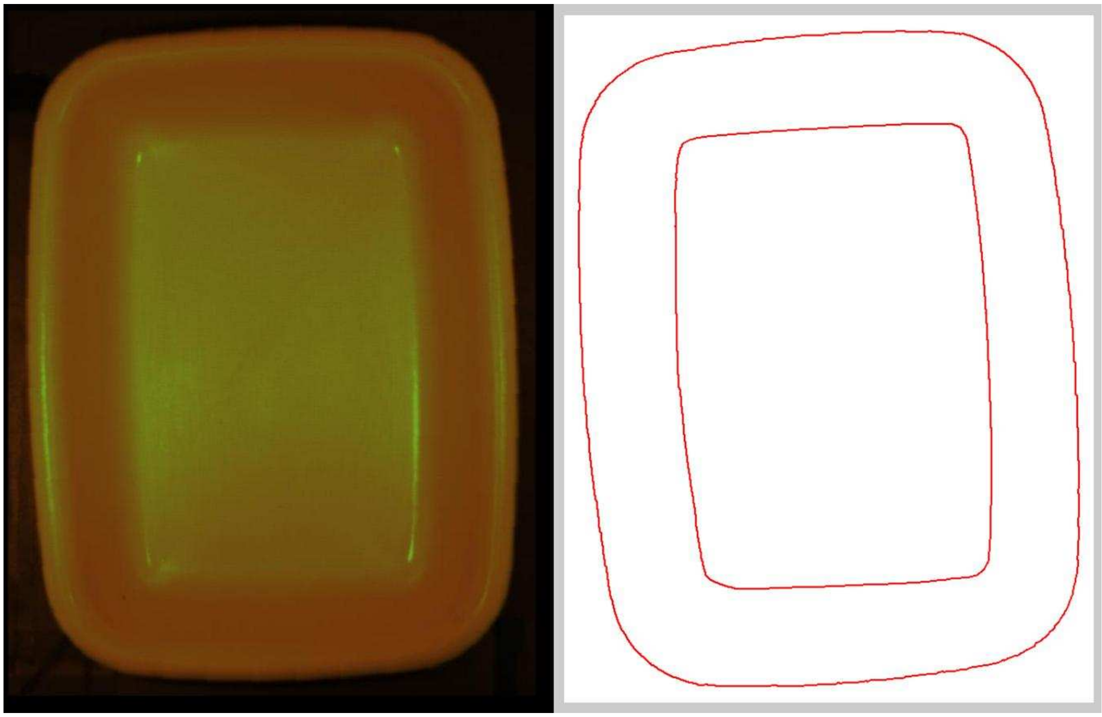


Figure 4.13: Original Clean SX (left) and Correctly Detected Non-dirt (right).

Inspection results of a dirty SX dish and a clean SX dish are presented in Figure 4.12 and 4.13, respectively. Notice in Fig 4.12 the dirt spot on the boundary of the floor region and wall region of the dish image (near the centre of the left image). This location of a dirty spot could be a significant challenge to detect, yet our approach found this spot.

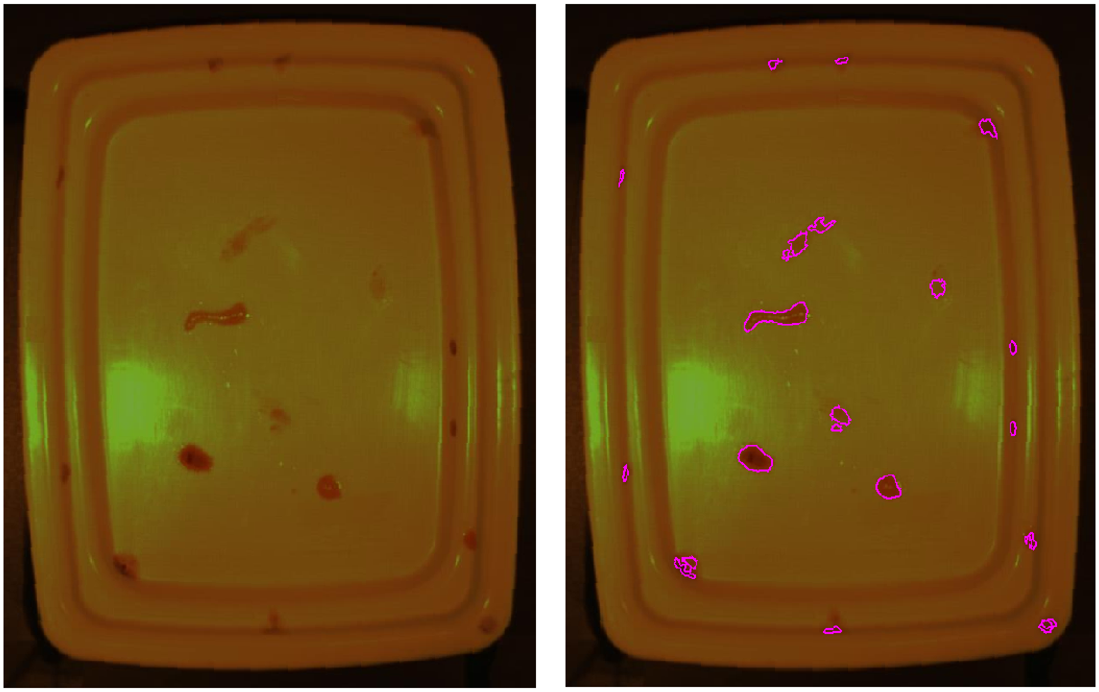


Figure 4.14: Original Dirty SP (left) and All Dirt Spots Detected (right).

One inspection results of a dirty SP dish are shown in Figure 4.14. All dirt spots, even some that are right on the inner edges of multi-layers of the dish, are correctly detected. This type of dish with 3 shallow layers was the most difficult case to tune the parameters of the inspection algorithm.

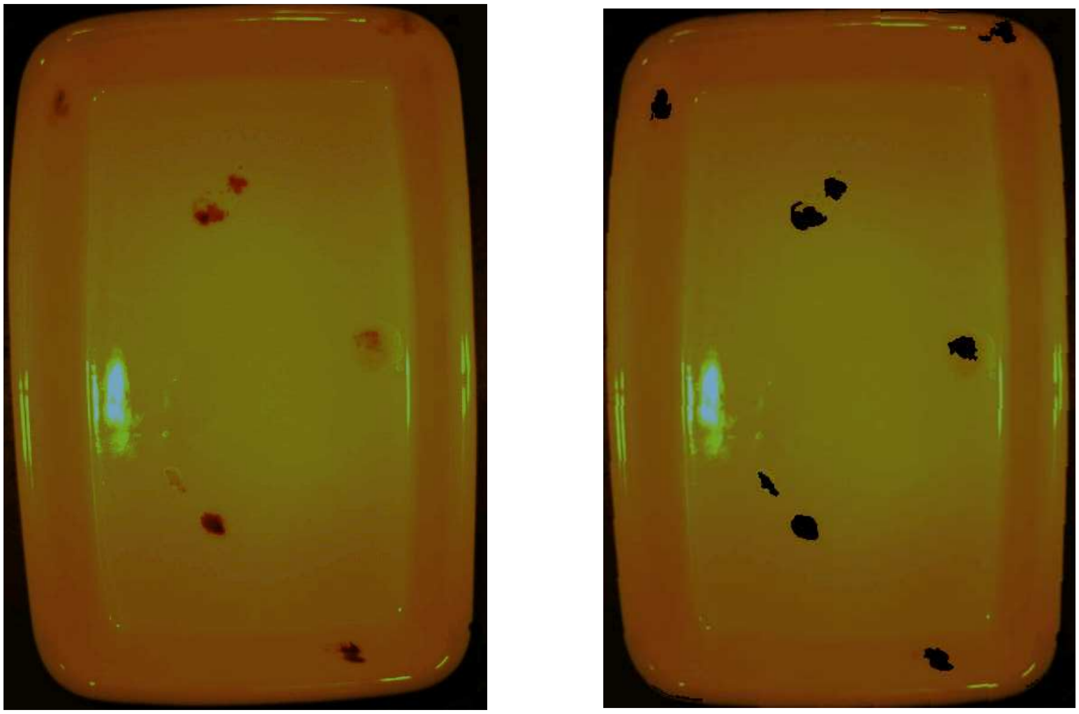


Figure 4.15: Original Dirty LC (left) and All Dirt Spots Detected (right).

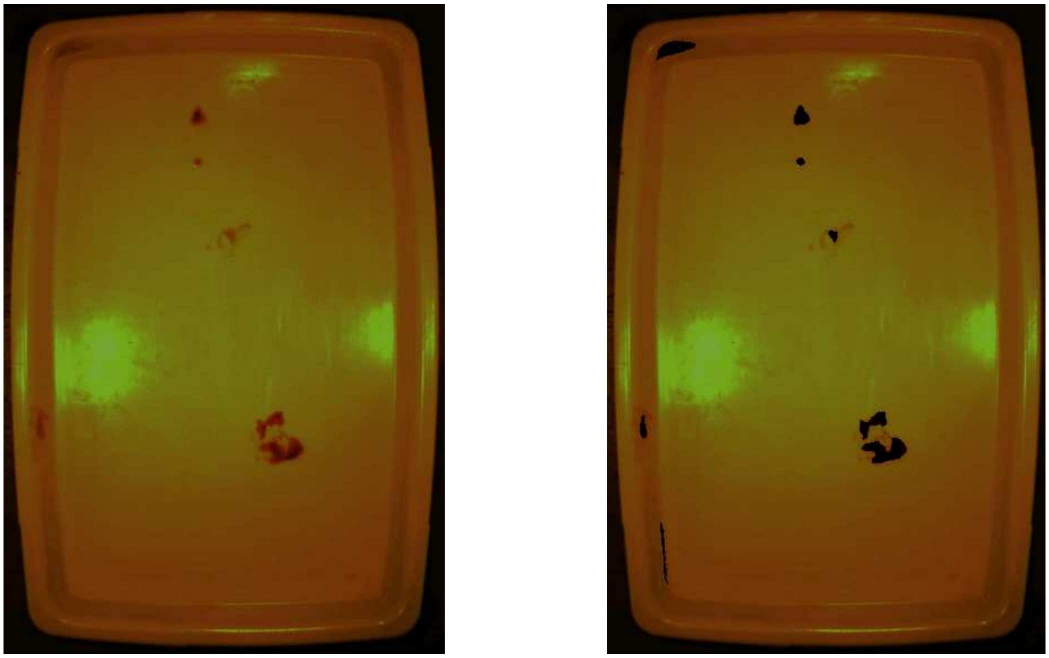


Figure 4.16: Original Dirty LP (left) and All Dirt Spots Detected (right)

Figure 4.15 and 4.16, respectively, show inspection results of a sample LC dish and a sample LP dish. All dirt spots were correctly detected.

A summary of inspection results for all five types of dish pieces, SC, SP, SX, LC, and LP, is given in Table 4.6.

Table 4.6: Summary of Inspection Results

	Number	Correct	Miss	Failed Alarm
Clean SC	5	4	-	1
Dirty SC	5	5	0	-
Clean SP	5	4	-	1
Dirty SP	5	5	0	-
Clean SX	5	5	-	0
Dirty SX	5	5	0	-
Clean LC	5	5	-	0
Dirty LC	5	5	0	-
Clean LP	5	4	-	1
Dirty LP	5	5	0	-
All dishes	50	47 (94%)	0 (0%)	3 (6%)

It is anticipated that the inspection system will produce certain false “clean” and false “dirty” results. A false “clean” result refers to a dirty dish piece that is wrongly classified as clean, and a false “dirty” result refers to a clean dish piece that is wrongly classified as dirty. It is assumed during the design of this system, that a false “clean” result is less preferable than a false “dirty” result, because a false clean is processed through to final clean storage, whereas a false dirty is merely recycled through the washing singulation, identification, and inspection stages of the overall system.

“Failed Alarm” in Table 4.6 means an incorrect result that a clean dish is classified as dirty. Such results would reduce production in a real inspection process line because the clean dish would be sent back to be re-washed when this was not necessary. On the other hand, such a result is deemed superior to a dirty dish that is classified clean, which would be unacceptable for hygienic and other reasons.

The dirt spot sizes (the smallest dimensions) in our experiments varied from approximately 2 mm to 6 mm. Although the results in Table 4.6 show that our algorithm correctly detected dirt spots 100% of the time, we note that defect size is not always a good measure of probability of detection (PoD) using a given inspecting method. According to George (2006), there is usually a large gap between the smallest flaw detected and the largest flaw missed. In our problem, the smallest dirt spot detected depends heavily on the difference between the dirt color and the color of dishes to be inspected, as well as the dirt location on the dish surface. Many more experiments would be needed to adequately address the probability of detection issue.

For the results in Table 4.6, the average computation time for the inspection process is 1.28 second per dish (using Matlab® R14, Image Processing Toolbox V5.0, Window Vista, dual core 1.6GHz, 2GB RAM) as shown in Table 4.7.

Table 4.7: Summary of Inspection Computation Time

	Number	Total Inspection Time* (sec)	Average Time* (sec)
Small dishes SC, SP, SX	30	35.5	1.18
Large dishes LC, LP	20	27.3	1.37
All dishes	50	62.8	1.26

If we add this to the average identification time reported earlier, the total time required for both identification and inspection processes is approximately 1.47 second, which is acceptable for our target of processing 30 dish pieces per minute (2 second per piece).

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In this study we successfully implemented new and novel dishware identification and inspection algorithms. The experimental results show that these algorithms work well under lighting variations and variation of dish position under the camera axis. The algorithms implemented on a standardly available PC (Intel 1.6 GHz dual core, 2GB RAM), with Matlab® R14 and Image Processing Toolbox V5.0, were sufficiently fast for real time processing at a minimum rate 30 dish pieces per minute. In summary:

1) We employed an experimental setup inherited from Zhou (2008) and Lolla (2005), and made modifications in lighting to provide sufficient illumination for identification and inspection tasks.

2) For identification, in order to find the minimum set of descriptors to produce fast, adaptable and efficient automatic dish recognition, we experimented with several shape-based properties, including area, ratio of length to width, and ratio of the object area to the area of minimum bounding box, together with some properties based on the distribution of gray levels of dish images.

Our identification algorithm used the selected set of descriptors consisting of area, ratio of length to width, and ratio of area to area of the oriented bounding box of dish images. Experiments were conducted on 725 images of ceramic and plastic dishes taken under different positions of 84 separate dishes of 5 different styles and shapes. Although the light condition was kept the same (with the exposure time of 20 ms), the varying of dish positions caused significant changes in dish image gray levels because of the non-uniform illumination in the field of view. The results were 100% of correct identification with an average identification time of 0.21 sec per dish.

3) For dish inspection, we propose a new technique using partitioning and adaptive thresholding, combined with global thresholding. Testing was conducted on 50 dish images, of 5 clean and 5 dirty dishes for each type, with variations in colors, size, shape, and position of real food particles, including BBQ sauce, egg yolk, coffee stain, and potato sauce, which were manually placed onto dish surfaces and dried in place. The results were 100% of dirty dishes correctly inspected and 6% of clean dishes miss-classified as dirty ("false alarm"). Overall, then, the accuracy of our inspection algorithm was 94%. Average time for inspection was 1.26 sec per dish. Therefore, the average time for both identification and inspection was 1.47 sec per dish, which is sufficient for a processing rate of 41 dish pieces per minute.

The partitioning and adaptive thresholding, combined with global thresholding, as presented here will not work for dishes that have colored or molded patterns on the dish surface. However, because most dish sets used in large scale dinning operations are mono-colored with a uniform background, our procedure should be widely applicable.

We also experimented with other dish sets, including plain circular and oval shaped plates, and small bowls. With small changes in few parameters, the algorithms work equally well.

5.2 Recommendations

Recommendations for future work include finding means to automatically tune parameters in the inspection algorithm, which must currently be established manually using trial and error. Based on observations that the percent of dirty dish pieces exiting a real dishwashing system is typically small, an iterative-based algorithm might be able to "learn" and adjust the parameters to a number of "known-as-clean" input dish images.

Another improvement could be finding means to automatically partition dish pieces using geometric information. As shown in Section 4.3, using intensity-based partitioning methods, which resulted in zigzag boundaries, did not produce good inspection results. While we were able to overcome this problem by manually defining the floor portion for our dish image template, this manual task reduced the portability of our algorithms. Our suggestion for geometry-based automatic partitioning is motivated by the power of generalized Hough transforms, which was proposed to recognize an arbitrary shape in an input image (Ballard 1981).

REFERENCES

AIA. (2009a). Applied Vision Partners with Stolle Machinery to Produce Machine Vision Systems for 100 Percent Decoration Inspection of Cans. Retrieved 02/12/2009, from http://www.machinevisiononline.org/public/articles/articlesdetails.cfm?id=3831.

AIA. (2009b). ATS Reports Third Quarter Fiscal 2009 Results. Retrieved 02/12/2009, from http://www.machinevisiononline.org/public/articles/articlesdetails.cfm?id=3849.

Ballard D. (1981). Generalizing the Hough Transform to Detect Arbitrary Shapes. Pattern Recognition, Vol.13, No.2, p.111-122, 1981

Ballard D. and Brown C (1982). Computer Vision, Prentice-Hall, 1982, Chap. 4.

Basler Co. (2007). Basler A102k User's Manual. 06. Retrieved 02/20/2009, from www.basler-vc.com.

Bernd G. (2008). Digital Image Processing Lecture Notes. Retrieved 01/30/2009, from www.stanford.edu/class/ee368.

Chow, C. K. and Kaneko K. (1972). Automatic Boundary Detection of the Left Ventricle from Cineangiograms, Comp. Biomed. Res.(5), 1972, pp. 388-410

Department of Defense Handbook (1999). Nondestructive evaluation system reliability assessment.

EIO. (2009a). Fundamental Parameters of an imaging system. Electronic Imaging Resource Guide Retrieved 01/24/2009, from http://www.edmundoptics.com/techsupport/DisplayArticle.cfm?articleid=287

EIO. (2009b). Illumination. Electronic Imaging Resource Guide. Electronic Imaging Resource Guide Retrieved 02/19/2009, from http://www.edmundoptics.com/techSupport/DisplayArticle.cfm?articleid=291

George A. G. and Jacobi Consulting Limited(2006). Probability of Detection (PoD) curves; derivation, applications, and limitations.

Gonzalez, R. C. and Woods, R. E. (2002). Thresholding. In Digital Image Processing, pp. 595–611. Pearson Education.

Gonzalez, R. C. and Woods R. E.(2004). Digital Image Processing. Upper Saddle River, NJ, Pearson/Prentice Hall.

Guoliang F. (2009). ECEN5793 Digital Image Processing Lecture Notes. Oklahoma State University, Spring 09.

Hashimoto, S. (1995). Separation of Silverware for Machine Vision Sorting and Inspection. School of Mechanical and Aerospace Engineering, Oklahoma State University. M.S. Thesis.

IMVE (2008a). Vision 2008 continues to expand. Imaging & Machine Vision Europe(26): 6-6.

IMVE (2008b). Vision companies report steady growth. Imaging & Machine Vision

Europe(29): 4.

Jain A.(1989). Fundamentals of Digital Image Processing, Prentice-Hall, 1989, Chap. 9.

Johnson, A. K. (1993). Machine Vision Sorting and Inspection in Commercial Automatic Dishwashing. School of Mechanical and Aerospace Engineering, Oklahoma State University. M.S. Thesis.

Lolla, S. V. G. (2005). Identification and inspection of silverware using machine vision. School of Mechanical and Aerospace Engineering, Oklahoma State University. M.S. Thesis.

MVA/ SME (2004). Basic Guide: Lighting and Optics, Machine Vision Association of Society of Manufacturing Engineers.

Nagaraj, S. (2003). Silverware Sorting Machine. School of Mechanical and Aerospace Engineering, Oklahoma State University. M.S. Thesis.

Peddi, R. V. (2005). Silverware Sorting and Orienting System. School of Mechanical and Aerospace Engineering, Oklahoma State University. M.S. Thesis.

Russ, J. C. (2007). The image processing handbook. Boca Raton, CRC/Taylor and Francis.

Shapiro, L. G. and Stockman, G. C. (2002). Computer Vision. Prentice Hall.

Smith, M. L. (2001). Surface Inspection Techniques - Using the Integration of Innovative Machine Vision and Graphical Modelling Techniques, John Wiley & Sons**:** 209.

Vernon D. (1991). Machine Vision. Prentice-Hall, 1991, Chap. 6.

Webb, A. R. (2002). Statistical pattern recognition. New Jersey, Wiley.

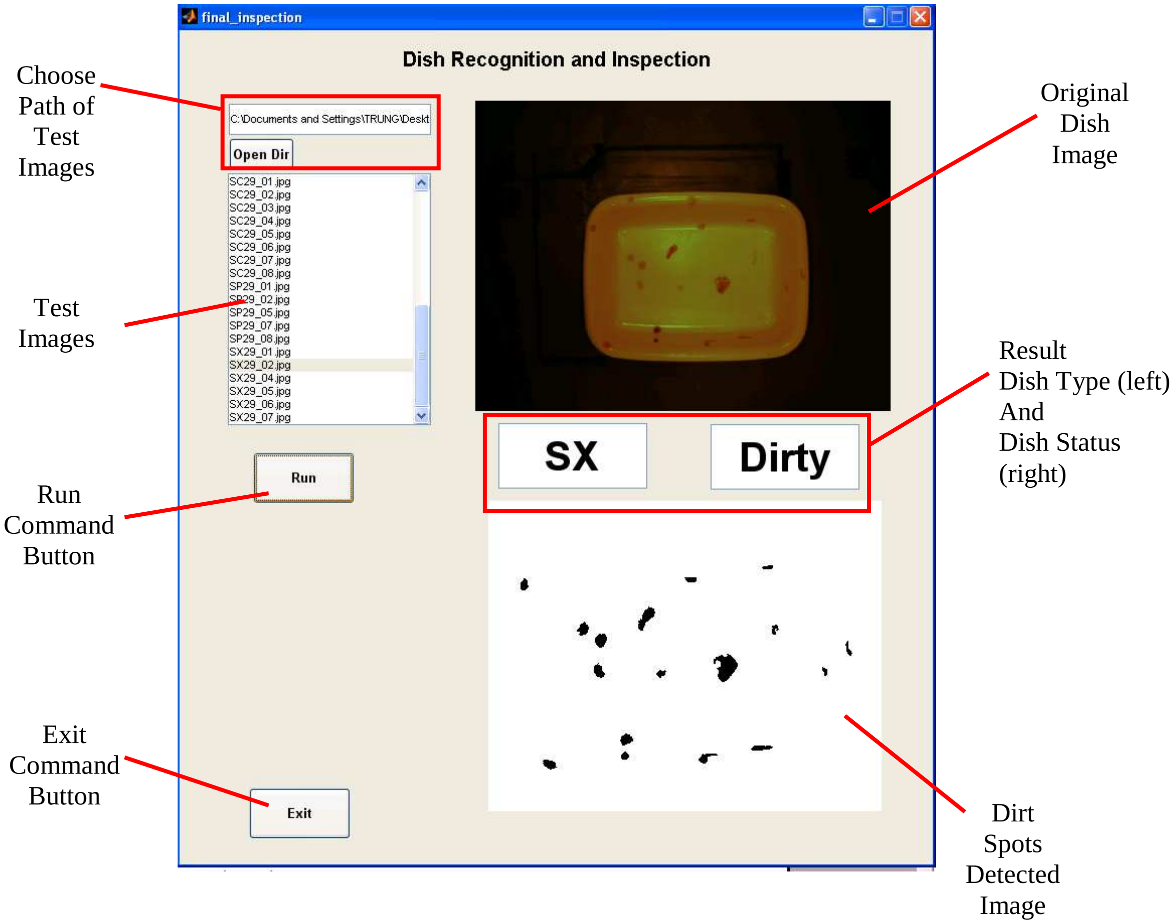
Yeri, S. (2003). Classification of Silverware Pieces Using Machine Vision. School of Mechanical and Aerospace Engineering, Oklahoma State University. M.S. Thesis.

Zhou, Y. (2007). On-line Flaw Detection on Metal Contoured Surfaces Using Image Processing Techniques. School of Mechanical and Aerospace Engineering, Oklahoma State University. Ph.D. Dissertation.

Zhou, Y. and Hoberock, L. L. (2008). Flaw Detection Using Image Registration and Fusion Techniques. Proceedings of the ASME 2008 International Design Engineering Technical Conference & Computers and Information in Engineering Conference.

APPENDICES

A. Graphic User Interface



final_inspection

Dish Recognition and Inspection

C:\Documents and Settings\TRUNG\Deskt

Open Dir

SC29_01.jpg
SC29_02.jpg
SC29_03.jpg
SC29_04.jpg
SC29_05.jpg
SC29_06.jpg
SC29_07.jpg
SC29_08.jpg
SP29_01.jpg
SP29_02.jpg
SP29_05.jpg
SP29_07.jpg
SP29_08.jpg
SX29_01.jpg
SX29_02.jpg
SX29_04.jpg
SX29_05.jpg
SX29_06.jpg
SX29_07.jpg

Run

SX

Dirty

Exit

Choose Path of Test Images
Test Images
Run Command Button
Exit Command Button
Original Dish Image
Result Dish Type (left) And Dish Status (right)
Dirt Spots Detected Image

B. Code listing:

```
%*********************************************%
%*********************************************%
% Main Function.m
% Including .fig Figure
%*********************************************%

function varargout = final_inspection(varargin)
% Last Modified by GUIDE v2.5 01-Apr-2009 14:09:37

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @final_inspection_OpeningFcn, ...
                   'gui_OutputFcn',  @final_inspection_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before final_inspection is made visible.
function final_inspection_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to final_inspection (see VARARGIN)

% Choose default command line output for final_inspection
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

initial_dir = 'C:\Documents and
Settings\TRUNG\Desktop\work\dishes\Sep29\';
set(handles.text1,'String',initial_dir);

% Populate the listbox
load_listbox(initial_dir,handles);
% Show preview
list = get(handles.listbox1,'String');
item_selected = list{get(handles.listbox1,'Value')};
```

```
axes(handles.axes1);
imshow(item_selected);
axes(handles.axes2);
bw = ones(519,696,'uint8') > 0;
imshow(bw);


% UIWAIT makes final_inspection wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = final_inspection_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;



function text1_Callback(hObject, eventdata, handles)

function text1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function ed_type_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function ed_type_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function ed_clean_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function ed_clean_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pb_open.
function pb_open_Callback(hObject, eventdata, handles)
```

```
% --- Executes on button press in pb_exit.
function pb_exit_Callback(hObject, eventdata, handles)

close(gcf);

% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
index_selected = get(hObject,'Value');
list = get(hObject,'String');
item_selected = list{index_selected}; % Convert from cell array
%disp(item_selected)
axes(handles.axes1);
imshow(item_selected);
axes(handles.axes2);
bw = ones(519,696,'uint8') > 0;
imshow(bw);
set(handles.ed_type,'String','');
set(handles.axes2,'Visible','Off');
set(handles.ed_clean,'String','');



% --- Executes on button press in pb_run.
function pb_run_Callback(hObject, eventdata, handles)
%====================================
%set(handles.text_wait,'Visible','On');

%===== Code at home
index_selected = get(handles.listbox1,'Value');
list = get(handles.listbox1,'String');
item_selected = list{index_selected}; % Convert from cell array

I = imread(item_selected);
%Pre-processing and Identification
dish_type = Which_dish_final(I);
set(handles.ed_type,'String',dish_type);

%Inspection process:
isclean = dish_inpection(item_selected,dish_type,handles)
set(handles.ed_clean,'String',isclean);




%============================================
% load listbox: from MATLAB-help
function load_listbox(dir_path, handles)
cd(dir_path)
%dir_struct = dir(dir_path);
dir_struct = dir([dir_path '*.jpg']);
[sorted_names,sorted_index] = sortrows({dir_struct.name}');
handles.file_names = sorted_names;
handles.is_dir = [dir_struct.isdir];
handles.sorted_index = sorted_index;
guidata(handles.figure1,handles)
set(handles.listbox1,'String',handles.file_names,'Value',1)
```

```
%**********************************************%
%**********************************************%
% Dish_inspection.m
%**********************************************%
function isclean = dish_inpection(fname,dtype,handles)

switch dtype
    case ‘SC’
        M = imread('bmaskSC.jpg');
        th1 = [50,105,60,.03,20,86];
        th2 = [15,90,30,0.05,20,70];
        dd = 1;
    case ‘SP’
        M = imread('bmaskSP.jpg');
        th1 = [50,99,50,.05,20,99];
        th2 = [40,75,12,0.03,20,54];
        dd = 2;
    case ‘SX’
        M = imread('bmaskSX.jpg');
        th1 = [50,99,50,.05,20,99];
        th2 = [0,255,30,0.03,20,66];
        dd = 3;
    case  ‘LC’
        M = imread('bmaskLC.jpg');
        th1 = [0,95,50,.04,20,94];
        th2 = [30,53,50,0.05,25,53];
        dd = 4;
    case ‘LP’
        M = imread('bmaskLP.jpg');
        th1 = [0,70,20,.04,6,80];
        th2 = [20,53,60,0.05,6,40];
        dd = 5;
end

M = im2bw(M);
L = bwlabel(M);
STATS = regionprops(L, 'Orientation','Centroid');
orien = STATS.Orientation;
mc = STATS.Centroid;
ground_level = 10;
I = imread(fname);

if dd>3
    I = imresize(I,1/2);
    ground_level = 30;
end

[Ig,Stats] = maxObj(I,.18);
Ig = imrotate(Ig,-Stats.theta + orien);
%I = imrotate(I,-Stats.theta + orien);

bw = Ig > ground_level;
% Center of dish:
```

```
[xx,yy] = find(bw);
Cx = round(mean(xx));
Cy = round(mean(yy));

%Full and bigger:
haftx = 336;
hafty = 426;
if dd>3
    haftx = 197;
    hafty = 291;

end
%J = I(Cx-haftx:Cx+haftx,Cy-hafty:Cy+hafty,:);
II = Ig(Cx-haftx:Cx+haftx,Cy-hafty:Cy+hafty);

bw = II > 10;
% Center of dish:
[xx,yy] = find(bw);
Cx = round(mean(xx));
Cy = round(mean(yy));

dxy = [Cx-mc(2), Cy-mc(1)];
%make the transform matrix
T = [1 0 0;0 1 0;dxy(1) dxy(2) 1];
tform = maketform('affine',T);
%Transform BB
[afx afy] = size(II);
M = imtransform(M,tform,'XData',[1 afy],'YData',[1 afx]);

% Floor detection

tmp = II;
%tmp(tmp==0) = th1(1);
tmp(tmp>105) = th1(2);

bw3 = adaptivethreshold(tmp,th1(3),th1(4),0);

%post process: using global thresholding
I3 = II;
I3(M==0) = 0;
hl = (I3<th1(5))|(I3>th1(6));

bw = bw3 | hl;

% Treat wall and outer dish the same

tmp = II;
%tmp(tmp<15) = 15;
tmp(tmp>90) = 90;
bwA = adaptivethreshold(tmp,th2(3),th2(4),0);

%post process: remove highlight
A = II-I3;
hl = (A<th2(5))|(A>th2(6));
```

```
bwAA = bwA | hl;
%figure, imshow((bw==1)&(bwAA==1))
% Remove small spot
bwall = ~((bw==1)&(bwAA==1));
bwall = bwareaopen(bwall,60);
bwall = imclearborder(bwall, 4);
bwall = ~imfill(bwall,'holes');

axes(handles.axes2)
imshow(bwall)
% hold on
% B = bwboundaries(II>40,'noholes');
% bd2 = B{1};
% plot(bd2(:,2),bd2(:,1),'r','LineWidth', 2);
if sum(find(bwall==0))>50
    isclean = 'Dirty';
else isclean = 'Clean';
end


%*********************************************%
%*********************************************%
% Dish Identification.m
%*********************************************%
function
[dish_code,descriptors]=Which_dish_final(I,areaRanges,SPXline,...

LCPline,threshold,down_scale)
%Recognition 5 type of dishes
%Base on shape descriptors:
%   1. Area
%   2. oREC = Length/Width
%   3. oEXT = Area/Area of aligned bounding box
%INPUTs:
%   I: matrix of image of dish
%   areaRanges: Ranges of area of each type of dish
%        = [LCmax LCmin LPmax LPmin SCmax SCmin SPmax SPsafe SXsafe
SXmin]
%        = 1e4*[5.51,5.25,...  %LCmax, LCsafe
%               5.14,4.96,...  %LPsafe, LPmin
%               2.83,2.65,...  %SCmax, SCmin
%               2.642,2.555,...%SPmax, SPsafe
%               2.52,2.45];    %SXsafe, SXmin
%   SPXline: The line to saperate SP,SX on oREC-oEXT properties plane
%        default = [1.25  0.925, ...% point_1
%                   1.41  0.96]     % point_2
%   LCPline: The line to saperate LC,LP on oREC-oEXT properties plane
%        default = [1.5  0.935, ...% point_1
%                   1.505   0.955]   % point_2

%   threshold: gray threshold, default = 0.1
%   down_scale: down size the image, default = 1/4
%
%
%
```

```
%OUTPUT:
%   dish_code = LC: Large Ceramic   ,= SC: Small Ceramic
%             = LP: Large Platic    ,= SP: Small Platic
%                                   ,= SX: Small
%   descriptors: Area, oREC, oEXT
%---------------------------------------------------------
%Feb 09,08

if nargin < 6, down_scale = 1/4; end
if nargin < 5, threshold = .1; end
if nargin < 4,
    LCPline = [1.500,  0.935, ...% point_1
               1.505,  0.955];  % point_2
end
if nargin < 3,
    SPXline = [1.295,  0.915, ...% point_1
               1.342,  0.950];    % point_2
end
if nargin < 2,
    areaRanges = 1e4*[5.51,5.25,... %LCmax, LCsafe
                     5.14,4.96,...  %LPsafe, LPmin
                     2.83,2.65,...  %SCmax, SCmin
                     2.642,2.555,...%SPmax, SPsafe
                     2.52,2.45];    %SXsafe, SXmin
end
%Adjust areaRanges if down_scale ~= 4:
areaRanges = areaRanges*4*down_scale;

dishtypes = ['LC';'LP';'SC';'SP';'SX';'UN'];


I = imresize(I,down_scale);
I = rgb2gray(I);
bw = im2bw(I,threshold);
bw = bwareaopen(bw,50);
bw = imfill(bw,'holes');

L = bwlabel(bw);
STATS = regionprops(L, 'Area');

%Find the max obj
[maxAreaObj,kmax] = max([STATS.Area]);

descriptors = [maxAreaObj; 0; 0];
pos = find(areaRanges < maxAreaObj);

if isempty(pos),
    dish_code = dishtypes(6,:);
    return;
else pos = pos(1);
end

if (mod(pos,2)==1)&(pos ~= 9)&(pos ~= 3) ,
    dish_code = dishtypes(6,:);
    return
```

```
elseif (pos ~= 9)& (pos ~= 3)
    dish_code = dishtypes(pos/2,:);
    return
else    %now pos == 9 -> SP-SX overlap
        % or pos == 3 -> LC_LP overlap

%Need other descriptors: oREC & oEXT
    STATS = regionprops(L,'Orientation','Image','Centroid');
    OBJ = STATS(kmax).Image;

    centroid = STATS(kmax).Centroid;
    dx = -round(centroid(1));
    dy = -round(centroid(2));
    orien = pi/180*(STATS(kmax).Orientation);

    T = [1 0 0;0 1 0;dx dy 1];
    Tin =[1 0 0;0 1 0;-dx -dy 1];
    R = [cos(orien) sin(orien) 0;-sin(orien) cos(orien) 0;0 0 1];
    tform = maketform('affine',T*R*Tin);
    OBJ = imtransform(OBJ,tform);

    L = bwlabel(OBJ);
    ST = regionprops(L,'BoundingBox');
    box = ST.BoundingBox;
    oREC = box(3)/box(4);
    oEXT = maxAreaObj/(box(3)*box(4));

    descriptors(2:3) = [oREC, oEXT];
    if pos == 9
        %CASE area in the overlap of SP-SX
        % if o_REC,o_EXT fall on the SP region:
        x1 = SPXline(1);         y1 = SPXline(2);
        nx = SPXline(3) - x1;    ny = SPXline(4) - y1;
        if ((oEXT-y1)*nx - (oREC-x1)*ny  > 0)
            dish_code = dishtypes(4,:); %'SP';
        else
            dish_code = dishtypes(5,:); %'SX';
        end %of SP region
    end %of case SP-SX

    if pos == 3
        %CASE area in the overlap of LC-LP
        % if o_REC,o_EXT fall on the LP region:
        x1 = LCPline(1);         y1 = LCPline(2);
        nx = LCPline(3) - x1;    ny = LCPline(4) - y1;
        if ((oEXT-y1)*nx - (oREC-x1)*ny  > 0)
            dish_code = dishtypes(2,:); %'LP';
        else
            dish_code = dishtypes(1,:); %'LC';
        end %of SP region
    end %of case SP-SX

end
```

```
%*********************************************%
%*********************************************%
% Pre-processing
%*********************************************%
function [F,Stats] = maxObj(f_name,threshold)
% MAXOBJ: Retrieve the max Object of the image
%INPUT: file name, or matrix representation of image
if ischar(f_name),
    I = imread(f_name);
else I = f_name;
end

%I = imresize(I,1/4);
if size(I,3)>1,
    I = rgb2gray(I);
end
%threshold = .18;
bw = im2bw(I,threshold);
bw = bwareaopen(bw,30);

%   se = strel('disk',2);
%    bw = imclose(bw,se);
%figure, imshow(bw)
bw = imfill(bw,'holes');
%figure, imshow(bw)

L = bwlabel(bw);

%Find the max obj
STATS = regionprops(L, 'Area');
%Find the max obj
[maxArea,kmax] = max([STATS.Area]);
% Diride the max obj
F = uint8(I);
F(L~=kmax) = 0;

Stats.Area = maxArea;
%Find Center Gravity and major-Orientation
L(L~=kmax) = 0;
L(L==kmax) = 1;
STATS = regionprops(L, 'Centroid','Orientation');
%save to output
Stats.xy = STATS.Centroid;
Stats.theta = STATS.Orientation;



%*********************************************%
%*********************************************%
% Adaptive thresholding
%*********************************************%
function bw=adaptivethreshold(IM,ws,C,tm)
%ADAPTIVETHRESHOLD
%  Based on information from
%  http://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm
```

```
if (nargin==3)
    tm=0; % default: using mean
    % tm =1 , using median
elseif (tm~=0 && tm~=1)
    error('tm must be 0 or 1.');
end

IM=mat2gray(IM);

if tm==0
    mIM=imfilter(IM,fspecial('average',ws),'replicate');
else
    mIM=medfilt2(IM,[ws ws]);
end
sIM=mIM-IM-C;
bw=im2bw(sIM,0);
bw=imcomplement(bw);

%*********************************************%
%*********************************************%
% Color Segmentation
%*********************************************%
function I = colorseg(varargin)
% S = COLORSEG('Euclidean',F,T,M), F: input image,
%       T is threshold, M 1x3 is average color
% S = COLORSEG('Mahalanobis',F,T,M,C), C is 3x3 covariant matrix
% F: input RGB image

f = varargin{2};
M = size(f,1); N = size(f,2);
%convert f to stack vector
[f,L] = imstack2vectors(f);
f = double(f);
%Initial I as vector, reshape later
I = zeros(M*N,1);
T = varargin{3};
m = varargin{4};
m = m(:)'; % make sure row vector

if length(varargin) == 4
    method = 'E'; %Euclidean
elseif length(varargin) == 5
    method = 'M'; % Mahalanobis
else
    error('Wrong number of inputs');
end
switch method
    case 'E'
        p = length(f);
        D = sqrt(sum(abs(f - repmat(m,p,1)).^2,2));
    case 'M'
        C = varargin{5};
        D = M_distance(f,m,C);
    otherwise
        error('Unknown method');
```

```
end

I(find(D <= T)) = 1;
I = reshape(I,M,N);

%====================================
% Sub: Mahalanobis distance:
function d = M_distance(Y,mx,Cx)
%   mx: mean, row vector
%   Cx: covariance matrix
ny = size(Y,1);
mx = mx(:)';
Yc = Y - mx(ones(ny,1),:);
d = real(sum(Yc/Cx.*conj(Yc), 2));
%*********************************************%
%*********************************************%
% Color_segmentation based on Cosine angular
%*********************************************%
function I = cos_seg(f,cmean,d)
%page 121 Color Image Processing
%Method and Application, Rastislav Lukac
% cos(theta) = dot(c1,cmean)/(|c1||cmean|)
% c = {R,G,B}
%Criteria: 255*sqrt(1 - cos(theta)^2) <= d
% cosT^2 >= 1 - d/255
if (ndims(f) ~= 3) | (size(f,3) ~= 3)
   error('Input image must be RGB');
end
M = size(f,1); N = size(f,2);
f = double(reshape(f,M*N,3));
%Initial I as vector, reshape later
I = zeros(M*N,1);
cmean = cmean(:)'; %make sure a row

cosT_2 = dot(f,repmat(cmean,M*N,1),2).^2./(sum(f.*f,2)+eps)...
                                         /sum(cmean.*cmean);
I(cosT_2 >= 1-d/255) = 1;
I = reshape(I,M,N);


%*********************************************%
%*********************************************%
% Edge detecting based on Color Gradient
%*********************************************%
function [VG, A, PPG] = colorgrad(f,T)
% F: color image,
% T: threshold, 0.0->1.0
%OUTPUT:
% VG: vector gradient
% A : angle (radian)
% PPG: composite gradient

sh = fspecial('sobel');
sv = sh';
Rx = imfilter(double(f(:,:,1)),sh,'replicate');
Ry = imfilter(double(f(:,:,1)),sv,'replicate');
```

```
Gx = imfilter(double(f(:,:,2)),sh,'replicate');
Gy = imfilter(double(f(:,:,2)),sv,'replicate');
Bx = imfilter(double(f(:,:,3)),sh,'replicate');
By = imfilter(double(f(:,:,3)),sv,'replicate');
% vector gradient
gxx = Rx.^2 + Gx.^2 + Bx.^2;
gyy = Ry.^2 + Gy.^2 + By.^2;
gxy = Rx.*Ry + Gx.*Gy + Bx.*By;
A  = 0.5*atan(2*gxy./(gxx - gyy + eps));
G1 = 0.5*(gxx + gyy + (gxx-gyy).*cos(2*A) + 2*gxy.*sin(2*A));
%Repeat for angle+pi/2
A = A + pi/2;
G2 = 0.5*(gxx + gyy + (gxx-gyy).*cos(2*A) + 2*gxy.*sin(2*A));
G1 = G1.^0.5;
G2 = G2.^0.5;
%Form VG, pick maximum at each (x,y), scale to [0,1]
VG = mat2gray(max(G1,G2));
% gradient per plane
RG = sqrt(Rx.^2 + Ry.^2);
GG = sqrt(Gx.^2 + Gy.^2);
BG = sqrt(Bx.^2 + By.^2);
%composite, scale to [0,1]
PPG = mat2gray(RG + GG + BG);
if nargin==2
    VG  = (VG  > T);%.*VG;
    PPG = (PPG > T).*PPG;
end



%*********************************************%
%*********************************************%
% Find Mean and Covariance of a matrix
%          (group of sample points)
%*********************************************%
function [C,m] = covmatrix(X)
%Input: X Kxn
%OUTPUT:
%   m : mean vector, nx1
%   C: covariance matrix, nxn
[K,n] = size(X);
X = double(X);
if n == 1
    C = 0;
    m = X;
else
    m = sum(X,1)/K;
    X = X - m(ones(K,1),:);
    C = (X'*X)/(K-1);
    m = m'; %to colummn vector
end
```

VITA

Trung Huy Duong

Candidate for the Degree of

Master of Science

Thesis: DISHWARE IDENTIFICATION AND INSPECTION FOR AUTOMATIC DISHWASHING OPERATIONS

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Amthuong, Phutho Province, Vietnam, on December 10th, 1981, the second son of Duong Huy Lung and Vu Thi Luc. Married to Doan Huong Ly on November 2005.

Education: Graduated from Vinhphuc Talented High School, Vinhphuc, Vietnam, in 1999. Received a Bachelor degree in Mechatronics, Mechanical Engineering from Talented Engineers Training Program, Center for Talent Training, Hanoi University of Technology, Vietnam, in Jun 2004. Completed the requirements for the Master of Science in Mechanical Engineering at Oklahoma State University, Stillwater, Oklahoma in May, 2009.

Experience: Piping Design Engineer (in trial), CIMAS Co., Vietnam, summer 2004. Lecture Assistant, Department of Mechanical Engineering, University of Communication and Transports, Vietnam, from 2004 to 2007. Graduate Research Assistant, Department of Mechanical and Aerospace Engineering, Oklahoma State University from January 2008 to May 2009.

Name: Trung Huy Duong   Date of Degree: May, 2009

Institution: Oklahoma State University   Location: OKC or Stillwater, Oklahoma

Title of Study: DISHWARE IDENTIFICATION AND INSPECTION FOR
AUTOMATIC DISHWASHING OPERATIONS

Pages in Study: 95   Candidate for the Degree of Master of Science
Major Field: Mechanical Engineering

Scope and Method of Study: Commercial dishwashing systems currently involve human loading, sorting, inspecting, and unloading dishes and silverware pieces before and after washing, in hot and humid environments. Automation is desirable, especially in large scale kitchens, to improve safety and efficiency. We propose automatically identifying dishes in mixed batches by using statistics of shape descriptors of dish pieces. Experiments were conducted on 1225 images of ceramic and plastic dishes taken in different lighting conditions using different positions of 84 separate dishes of 5 different styles and shapes. In order to find the minimum set of descriptors to produce fast, adaptable and efficient automatic dish recognition, we employed several shape-based properties, including area, perimeter, ratio of length to width, extension, and minimum bounding box, together with some properties based on gray level and color of dish images. Selected set of descriptors were area, ratio of length to width, and ratio of area to area of the oriented bounding box of dish images. For dish inspection, we propose a new technique using partitioning and adaptive thresholding, combined with global thresholding. Matlab® R14 and Image Processing Toolbox V5.0 were used.

Findings and Conclusions: The machine vision algorithms, developed in this study, are fast, simple, and produce results invariant with lighting conditions and dish rotation about the camera-dish axis.

ADVISER'S APPROVAL: Dr. Lawrence L. Hoberock