8-20-2014 12:00 AM

A Vision System for Automating Municipal Waste Collection

Justin M. Szoke-Sieswerda
The University of Western Ontario

Supervisor
Dr. McIsaac
The University of Western Ontario

Graduate Program in Electrical and Computer Engineering
A thesis submitted in partial fulfillment of the requirements for the degree in Master of Engineering Science
© Justin M. Szoke-Sieswerda 2014

A VISION SYSTEM FOR AUTOMATING MUNICIPAL WASTE COLLECTION

(Thesis format: Monograph)

by

Justin Marcus Szoke-Sieswerda

Graduate Program in Electrical and Computer Engineering

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Engineering Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Abstract

This thesis describes an industry need to make municipal waste collection more efficient. In an attempt to solve this need Waterloo Controls Inc. and a research team at UWO are exploring the idea of combining a vision system and a robotic arm to complete the waste collection process. The system as a whole is described during the introduction section of this report, but the specific goal of this thesis was the development of the vision system component. This component is the main contribution of this thesis and consists of a candidate selection step followed by a verification step.

The LINE2D gradient response map (GRM) method was used to find a candidate location because of its speed in locating texture-less object. However, since this algorithm has a significant false positive rate it was modified in the following three ways: contour filtering was added to the process, linearization of the cosine responses was performed, and applying noise suppressing, via polling, was performed twice instead of once. These additions considerably reduced the false positive rate, however not enough to disregard the addition of a verification step.

Regarding the verification step, the histogram of oriented gradients (HOG) was used. This algorithm produces a highly descriptive vector, based on gradient information. Using the HOG vectors and a simple Euclidean distance metric the verification step successfully dropped the false positive rate to zero. However, these highly descriptive HOG vectors, which successfully rejected areas that did not contain waste receptacles, would at times reject areas that did. This led to an increase in the amount of false negatives. Ideas are presented in the future work section of this thesis that might be able to alleviate the false negatives.

Table of Contents

Table of Figures

Chapter 1

Introduction

This chapter of the report is divided into the following sections. Section 1.1 will provide motivation for the entire project, as well as place the work of this thesis into context to the entire project. Section 1.2 will discuss the goals of the computer vision system. Section 1.3 will enumerate the contributions made by this thesis, and finally section 1.4 will give a brief overview of the layout of the thesis.

1.1 Motivation and Task Identification

The National Renewable Energy Laboratory (NREL) completed a study in the United States on the cost and energy use of municipal waste management and according to their study, which included data collected from six municipalities of various sizes and population, the largest cost in all waste management systems is attributed to the collection process. The collection of municipal waste accounts for, on average, 50% of the total cost to run the waste management system[1, 2]. The remaining 50% is used for general administration, maintaining a landfill, facility costs, and the transfer of garbage. With that being said it is clear that there is an industry need to find more efficient ways of collecting municipal waste.

Over the last few decades progress has been made in making waste collection more efficient. One of the biggest advancements in efficiency can be attributed to the introduction of what the waste management industry calls, semi-automated and fully-automated collection systems. (The nomenclature is confusing because both systems require driver interaction.) These systems use hydraulic arms to lift and dump the waste receptacles. The difference between a semi-automated system and a fully automated system is the type of human interaction required. In a semi-automated system the worker must exit the vehicle and put the waste containers into the lifting mechanism, and in a fully-automated system the work does not have to leave the truck and controls the hydraulic arm via a joystick.

These systems greatly reduced collection costs for municipalities by reducing the number of people per truck, the number of insurance claims per worker, and finally the time required to collect the garbage per house. Regarding this last point of efficiency Waterloo Controls Inc. has employed a team of researchers at the University of Western Ontario (UWO) to develop a system that can automate the collection process even further. The specific aim of this system is to reduce the amount of human interaction involved in the collection process. To accomplish this goal Waterloo Controls Inc. has asked the research team at UWO to design a vision system that can detect a waste receptacle in outdoor conditions and return the receptacle's 3D world coordinates. The coordinates will be used by the control system and robotic arm to pick, empty and return the receptacle to its initial location. With that being said the purpose of this thesis was to develop the vision system algorithm that will be employed in the automated waste collection system. **Error! Reference source not found.** shows a high level abstraction of the system that he research team is employed to create. The author of this thesis was tasked with the algorithm development aspect of this project and the components outlined in red show the focus of this thesis.

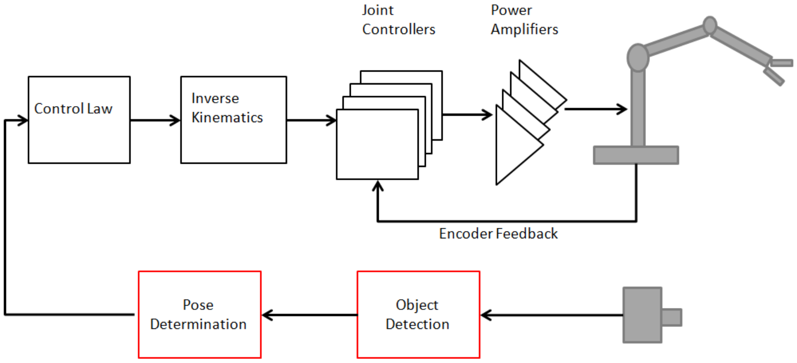


Figure 1.1: High Level Abstraction of System

1.2 Goals of Computer Vision System

The goal of the computer vision system is twofold:

1. Detect the waste receptacle in outdoor conditions
2. Determine pose information so the arm can acquire the waste receptacle

In conjunction to these goals there are a set of constraints that must also be addressed:

1. Only one camera should be used to keep the cost of the system down
2. The algorithm must be computationally efficient in order to reduce collection time
3. The appearance of the waste receptacle cannot be altered

These constraints are externally imposed and as a result reduce the number of feasible solutions. It should also be noted that the goals above were achieved within the constraints outlined. However, it should also be noted that the algorithm has not yet been tested in Canadian winter conditions and so the algorithms success can so far only be applied to three out of four seasons here in Canada.

1.3 Contributions of Thesis

The following list is an enumeration of the contributions made by the author of this thesis.

1. Designed the algorithm pipeline as a whole (**main contribution)**
2. Developed a point selection method during template creation
3. Used image processing techniques to make templates more robust
    a. Image averaging
    b. Edge Sharpening
4. Added contour enhancement to the GRM algorithm
5. Implemented multiple de-noising steps in GRM algorithm
6. Linearized the gradient response maps in order to reduce false positives

1.4 Thesis Overview

The remaining chapters of this thesis are divided as follows. Chapter 2 will provide a literature review of object detection from the perspective of local and global feature vectors, as well as provide a literature review on contour enhancement and suppression algorithms. Chapter 3 will provide detailed background information of the key components used by the proposed algorithm, more specifically it will detail the gradient response map algorithm, the histogram of oriented gradients algorithm, and the frequency filtering algorithm used for contour suppression. Chapter 4 will present the proposed algorithm as well as discuss the contributions made in this thesis. Chapter 5 will present a statistical evaluation of the algorithm. Chapter 6 will provide a conclusion on the success of the algorithm, reiterate the contributions made by this thesis, and outline future work relating to this topic. Finally, this thesis contains an appendix that provides a brief primer/overview of image processing techniques which pertain to this thesis work. The appendix also contains additional test results that support the arguments made in the contribution section of this report.

Chapter 2

Literature Review

This chapter of the report is partitioned into three sections. The first section will provide a literature review on object recognition via local feature vectors. The second section will provide a literature review on object recognition via global feature vectors. Finally, the third section will provide a literature review on processing techniques used for enhancing/suppressing contours with in an image.

2.1 Object Detection Via Local Feature Vectors

Object detection based on local feature vectors is a very popular paradigm in computer vision. Over the past two decades significant advancements in this area have been made, some of the more notable contributions include algorithms such as SIFT and SURF [3, 4, 5]. In order to place past contributions into context this introductory section will give a brief overview of the components involved in creating local feature vectors.

There are two main components to building a local feature vector: keypoint detection and keypoint description, as shown in Figure 2.1. However, before proceeding with a description of these two stages the term keypoint should be addressed. A keypoint is defined as a point of interest within an image, where the main characteristic of this point is that it can be easily distinguished from points with a local neighborhood.

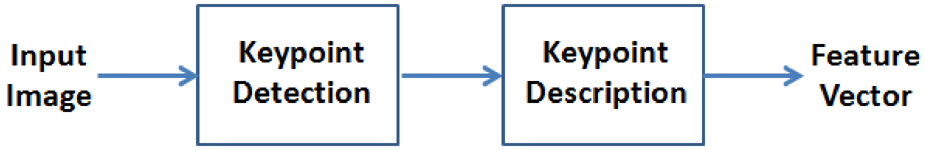


Figure 2.1: Creation of Local Feature Vector

The keypoint detection step is, as the name suggests, concerned with detecting keypoints within an image. There are many methods that can do this [3, 6, 7] but what they all have in common, for the most part, is that they search for points within an image that do not suffer from the aperture problem. Points that do not suffer from the aperture problem are points that can be reliably matched between two images. Figure 2.2 depicts the aperture problem. In this example the points surrounded by the red and black circles suffer from the aperture problem because there would be many corresponding matches between the two images. The red point would match to all interior points and the black point would match to all edge points along the right side. The only points that do not suffer from the aperture problem, in this example, are the corner points. Thus, the goal of all keypoint detectors is to find these types of points within in image.

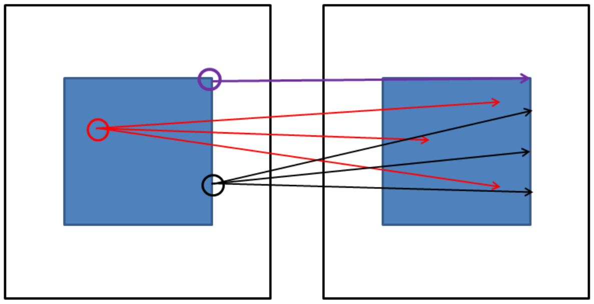


Figure 2.2: Aperture Problem

The keypoint description stage is concerned with building a local descriptor around each keypoint detected in the previous stage. There are many algorithms that can perform this local description [4, 5, 8, 9]. Figure 2.3 illustrates this stage by showing the keypoints detected in the previous stage as green dots, and the local neighborhoods around each keypoint as the red boxes. For simplicity of explanation intensity patches were used to describe the neighborhoods around each keypoint, however, many more robust algorithms exist. After every point has been described the local features can be concatenated to form one feature vector that describes the entire object.

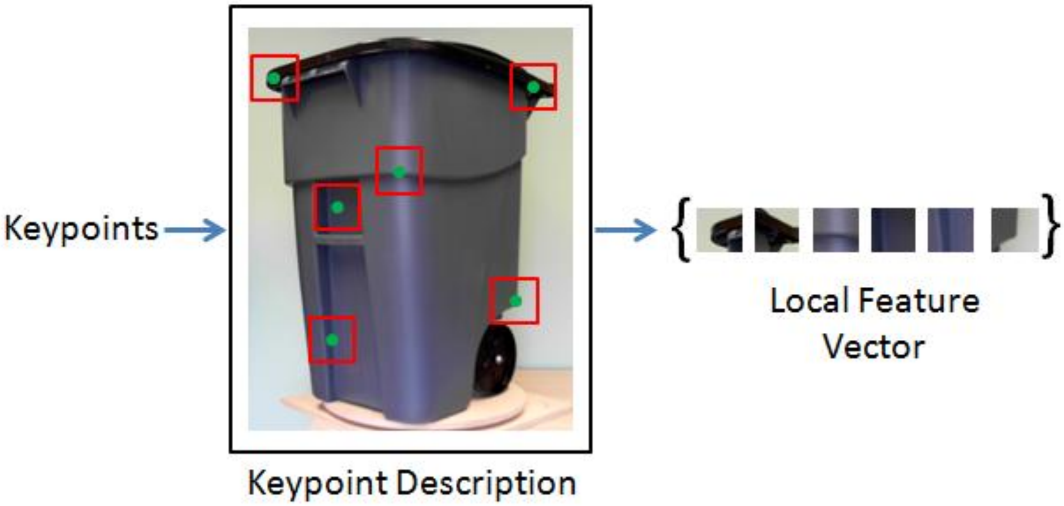


Figure 2.3: Keypoint Description and Vector Creation

After an object has been described, using a local feature vector, the methodology used to match this object to one within a scene can be accomplished in many different ways. For the sake of completeness Figure 2.4 illustrates the matching of a template object and a scene object using local features.

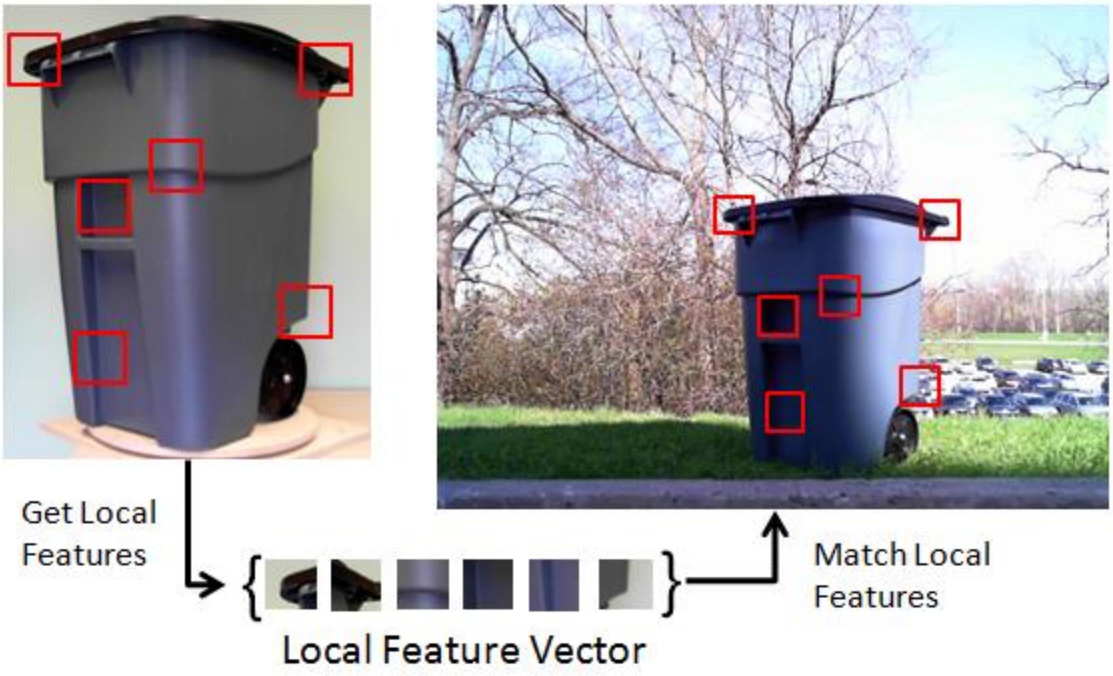


Figure 2.4: Finding an Object using Local Features

2.1.1 A Literature Review of Object Detection based on Local Feature Vectors

Now that some background on local feature vectors has been provided the following section will provide a summary of major contributions in this field. This section was mostly written in a chronological order, but at times it veers off on a tangent during the explanation of contributing work. Also, this field is very vast and not all major contributing works could be included within a reasonable amount of pages, so what is presented is what the author of this thesis considers major milestones.

In 1977 Hans Moravec developed what is known as the "Moravec corner detector" [6]. This corner detector can be considered one of the earlier types of keypoint detectors. It works by employing the sum of squared differences metric to find the maximum similarity score between patches shifted over a four or eight connected region. Moravec mainly used this detector in his work with stereo imaging and mobile vehicle guidance [10].

In 1988 the Moravec corner detector was greatly improved upon by Chris Harris and Mike Stephen [7]. They managed to make the detector isotropic in response to corners through some mathematical transforms. More specifically they used the second order Taylor series expansion and a 2D Hessian matrix to eliminate of the direction dependency response of shifting the intensity patches in a four or eight connected neighborhood. As a result of this expansion the corner detector became rotationally invariant. The computer vision field dubbed this detector the "Harris corner detector" (sorry Stephen). Even today this detector remains a very popular choice as a keypoint detector [11, 8, 12]. In fact, in a survey of keypoint detectors titled "Evaluation of Interest Point Detectors", by Cordelia Schmid [13], the improved Harris corner detector was considered the best [13]. It should be noted that the only difference between the improved Harris detector and the one detailed in the 1988 paper [7] is how the derivatives are calculated.

The two previously described detectors, the Moravec and Harris detectors, employed intensity based metrics to find keypoints. However, this is not the only type of methodology that can be used. In 1990 Radu Horaud et al.[14] employed a contour based method to find keypoints. The method worked by extracting line segments from a contour image and then a keypoint was found by finding where groupings of lines intersected. Compared to the Harris corner detector this keypoint detection algorithm was less successful but it demonstrates that keypoints can be found in other ways than intensity based methods. Two other contour based methods for extracting keypoints include the methods of Shilat et al. and Mokhtarian et al. [15, 16]. These too are not as successful as the Harris corner detector [13].

Prior to the 1995 work of Zhang et al. [12] the Harris corner detector was mainly used in applications regarding stereo imaging and short range motion tracking, however, in Zhang's work [12] the Harris corner detector was used to match keypoints over a large image range. This was accomplished by using a correlation window around each keypoint to select likely matches. The outliers found in this correlation window were removed by solving for a fundamental matrix that described the geometric constraints between the two views.

In 1994 Florack et al. [17] showed that by using set of local derivatives around a keypoint that a vector could be constructed (i.e., a feature descriptor) which is gray level invariant as well as rotationally invariant. In 1997, Florack's work was extended by Schmid and Mohr [11]. They showed that these invariant local features could be employed in the general image recognition problem. This was a major contribution to object recognition because it meant that local features could be extended past the short and wide baseline matching problem and into object matching against a large database of images.

Until David Lowe's ground breaking work, from 1999 to 2004 [3, 4], all keypoint detectors and descriptors developed thus far were not scale invariant. Lowe solved this scale invariant problem with the introduction of his scale invariant feature transform (SIFT) algorithm. It should be noted that SIFT can be considered the most influential algorithm in the field of local feature detection and description. At the time of writing this thesis David Lowe's paper regarding SIFT [3, 4] had a total of 7661 and 24440 citations, respectively. Since SIFT is a two part algorithm, which consists of a keypoint detector algorithm and a keypoint descriptor algorithm, the overview of this algorithm will be split into those two sections.

The SIFT keypoint detector algorithm is a three step process. The first step is to locate extrema in scale space, a concept introduced in 1983 by Witkin [18]. The second step is to filter through all of these extrema points to find ones that do not suffer from the aperture problem and to use Brown's interpolation method to achieve sub-pixel accuracy [19]. The third step of this algorithm is to assign canonical information to each keypoint detected, such as orientation, location, and scale.

The SIFT keypoint descriptor algorithm works by creating a local neighborhood around each keypoint, where the size of this neighborhood is determined by the canonical scale at which the keypoint was found. The scale-space level at which the descriptor is created is also determined by the canonical scale at which the keypoint was detected. Within this local neighborhood the pixels are split into 16 spatial bins and in each spatial bin a gradient orientation histogram is created using 8 orientation bins. The result of this spatial and orientation binning creates a 128 dimensional feature vector around each keypoint. It should be noted that in order to create rotational invariance each orientation within the local neighborhood is referenced to the canonical orientation calculated during the detection step, and not to the global orientation origin.

SIFT is used in many applications outside of detecting objects. It is also used in applications like image stitching, 3D modeling, gesture recognition, and image retrieval [20, 21, 22, 23]. It is a very successful algorithm that can be employed to solve many problems, however, it is optimized for textured images and is not appropriate for this thesis problem. To illustrate this point SIFT, which is an industry standard in this field, will be used to demonstrate the shortcoming of the local feature paradigm when texture-less objects are under consideration.

After Lowe introduced his scale invariant feature detector, in 1999, new algorithms started to be published that used the same scale space concept to achieve scale invariance. In 2000 Baumberg introduced the Harris AffineRegion detector [9]. In 2001 and 2002 Mikolajczyk and Schmid introduced, respectively, the Harris-Laplace and Harris-Affine detectors [24, 8]. As the names of these detectors suggest the Harris corner detector [7] is a key component in each of these algorithms.

Due to the success of SIFT when new algorithms are published in the local feature vector field of computer vision they often uses SIFT as a bench mark for comparison. The aim of these new algorithms is to be more computationally efficient (i.e., faster) than SIFT, while maintaining the same level of repeatability and matching between keypoint points. A few of these newer algorithms include: the PCA-SIFT algorithm developed in 2004 by Ke and Sukthankar [25]; the speeded up robust features (SURF) algorithm developed in 2006 by Bay et al. [5]; and the binary robust independent elementary features (BRIEF) algorithm developed by Calonder et al. [26], which uses the SURF detector to find its keypoints. These algorithms mentioned are in fact computationally faster than SIFT, but this speed comes at the price of repeatability.

2.1.2 Evaluation of Paradigm with Respect to Problem at Hand

Section 2.1 and 2.1.1 provided an overview of the local feature vector paradigm and a quick summary of major contributing works in this field, respectively. This section provides an explanation of why this paradigm is not suited for the problem at hand, which is to detect a waste receptacle and return its three-dimensional coordinates.

In order to illustrate why local features will not work, the SIFT algorithm was used in an attempt to find the waste receptacle. SIFT was chosen because as pointed out earlier it is one of the, if not the most, successful algorithms in this field. Figure 2.5 shows the best matches between keypoints in a template picture (image on the right) and keypoints in a scene with the object present (image on the left). The keypoints found and described in the template picture do not correspond to their expected locations within the scene. Although this is only a single example it illustrates the poor fit of SIFT for this problem.

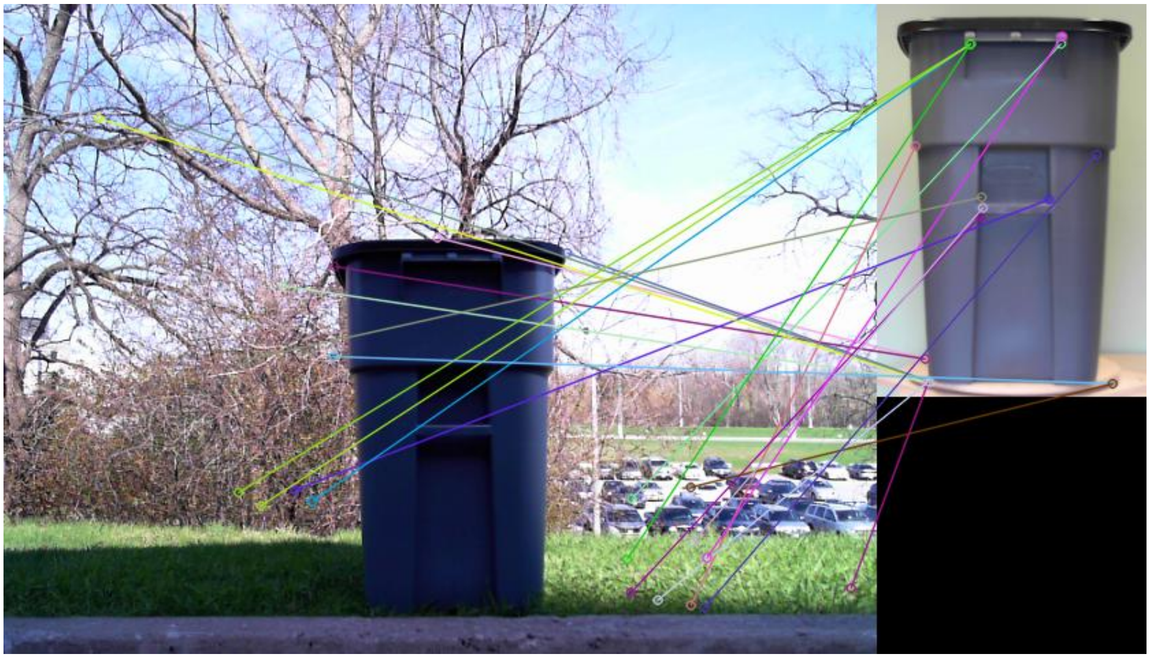


Figure 2.5: Using SIFT to Locate the Waste Receptacle

The reason local features failed to work for this problem is because the waste receptacle is a texture-less object. That is, the waste receptacle does not consist of many interesting points and as a result suffers greatly from the aperture problem. When a keypoint detector is used to try and find interesting points the best it can do is find points located at sharp corners. This is illustrated by the right most image in Figure 2.5 where the keypoints correspond to sharp corners of the object. Since the detector cannot find many interesting points most of the waste receptacle is left un-described and during the matching process an accurate pose estimation cannot be completed and therefore the object cannot be found. For comparison, Figure 2.6 shows the same algorithm being used on a highly textured object. Two images of the object were taken at different depths to create a change of scale. The larger image was used for the template and the smaller image as cropped into the same scene as the waste receptacle to illustrate that the background setting is not an issue. In Figure 2.6, since the object is highly textured the point correspondence between the two images is sufficient to locate the object.

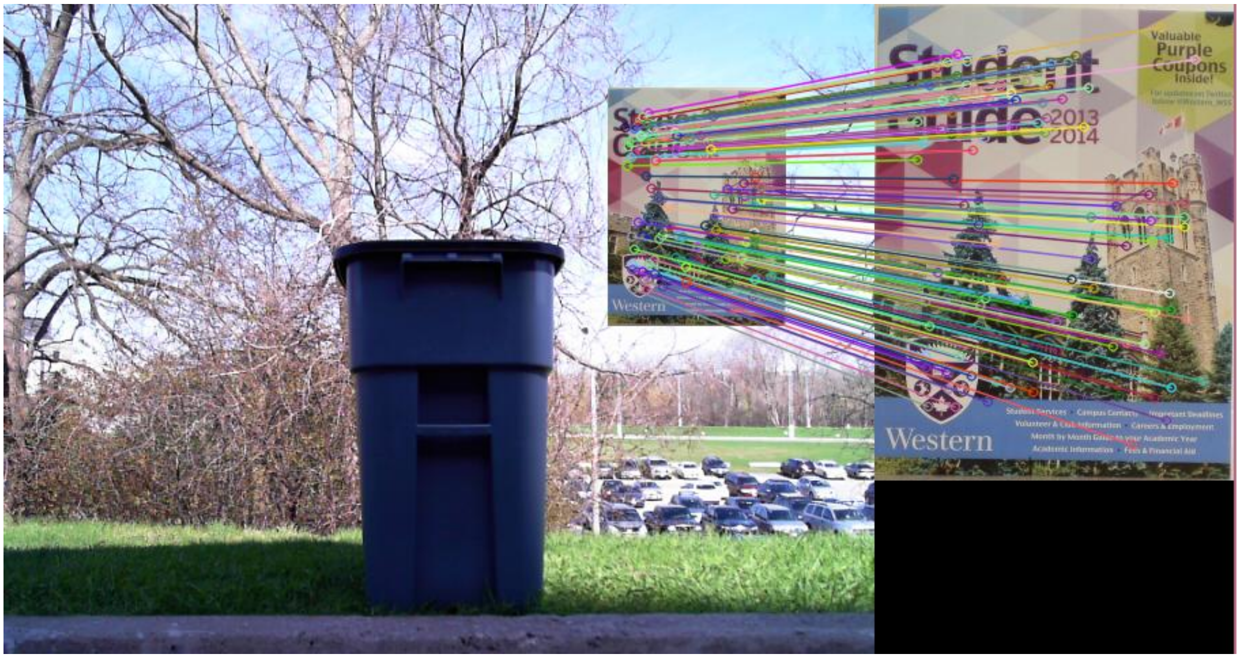


Figure 2.6: Using SIFT to Locate a Textured Object

2.2 Object Detection Via Global Feature Vectors

It was pointed out in the previous section that local feature vectors will not work for solving the problem presented in this thesis. Therefore, in order to find the waste receptacle attention was turned towards the global feature vector paradigm. This section will provide the appropriate background information on global features so that later sections can be placed into context.

When creating a global feature vector the entire template image is used in the description process. This is unlike local features where the first step is to find a set of keypoints and the second step is to only describe these keypoints. Figure 2.7 shows the general flow chart used when creating a global feature vector.

There are many algorithms that can be used to implement the description algorithm block shown in Figure 2.7 but all of these algorithms essentially follow the same procedure when finding a match within a scene. That is, they parse the scene image looking for a match, and a match is determined by either a similarity/distance metric or a machine learning algorithm. In this thesis similarity/distance metrics were used to find matches within a scene, therefore, this type of method will be used to illustrate the process of finding a match in the example below.

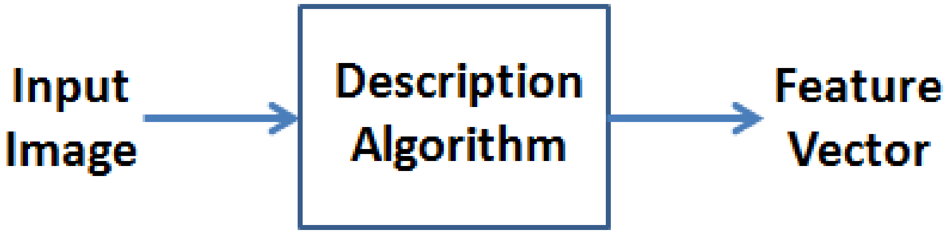


Figure 2.7: Creation of Global Feature Vector

The process of parsing a scene and looking for an object is often called template matching. Algorithms that employ template matching can be differentiated by two identifying features. The first differentiating feature is the method they employ to create the template. One possible categorical division of this field is intensity based methods, gradient based methods, and model based methods. Gradient based methods are currently considered the state of the art [27]. The second differentiating feature is the type of similarity/distance metric, or machine learning algorithm, they employ to compare locations within the scene to a specific template.

Figure 2.8 illustrates template matching using an intensity based template and the sum of squared difference (SSD) similarity metric. The template image is surrounded by a red box and for illustration purposes has been made 50% transparent. Since the template is intensity based the pixel values of the template and the pixel values of the scene location are used in the SSD similarity metric, where $I_1(i,j)$ is the template image intensity and $I_2(x+i,y+j)$ is the image scene intensity with the bin at location (x,y) with offset (i,j). When using SSD the best match corresponds to the lowest score (i.e., this is a minimization problem). For illustration purposes the similarity scores for every location is shown in the bottom image of Figure 2.8. The dark pixel intensities indicate good matches, and bright pixel intensities indicate bad matches.

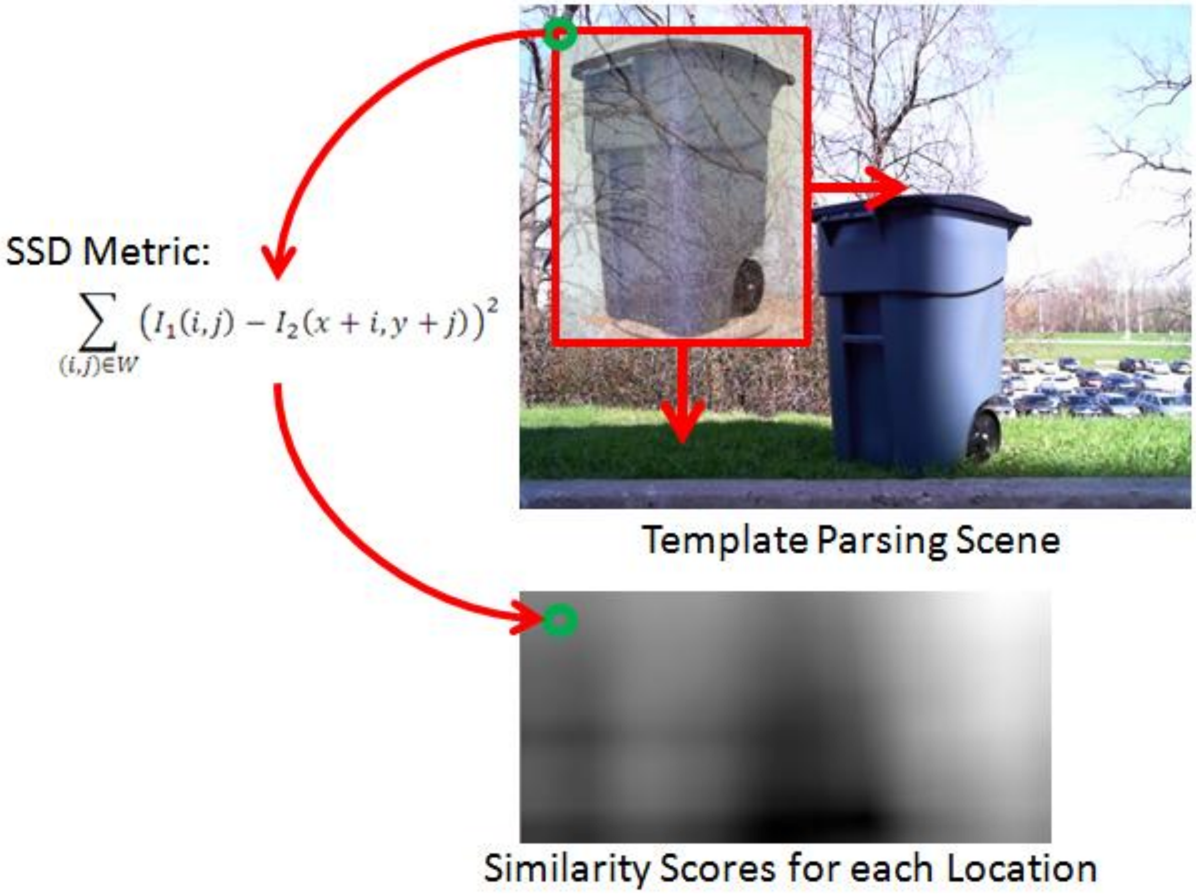

SSD Metric:
$\sum_{(i,j)\in W} \left(I_1(i,j) - I_2(x+i,y+j)\right)^2$
Template Parsing Scene
Similarity Scores for each Location

Figure 2.8: Template Matching with SSD

2.2.1 A Literature Review of Object Detection based on Global Feature Vectors

Now that some background information on object detection using global feature vectors has been provided the following sections will provide a summary of major contribution made in this field. Section 2.2.2 discusses model based methods, section 2.2.3 discusses intensity based methods, and section 2.2.4 discusses gradient based methods. These sections use an approach similar to section 2.1.1 and will only attempt to capture the key developments in the broad field. Since this research uses a gradient based approach this area is reviewed in greater detail in comparison to the other two sections.

2.2.2 Model Based Template Matching

One of the earliest attempts at template matching used a model based approach. In 1977 Barrow et al. [28] introduced chamfer matching for image registration. The idea was that line features in both images can be matched by minimizing the distance between them. In 1988 Gunilla Borgefors used Chamfer matching and contour models of objects to find instances of objects within a scene [29]. Researchers have gone beyond the simple contour models, used by Borgerfors, and have performed object detection using 2D and 3D CAD models of objects [30, 31, 32, 33, 34]. These method have been shown to be successful but they are computationally expensive, and involve significant offline work in model construction. Also, current state-of-art methods tend towards gradient based template matching [27].

2.2.3 Intensity Based Template Matching

In computer vision, template matching using intensity values has a long history because of its intuitive nature, and great success in locating objects in environments that can be controlled, like a factory or laboratory. However, this methodology has recently become less popular in computer vision because of the success of gradient based methods in uncontrolled environments.

The origin of template matching based on intensity values can be traced back to the origins of computer vision and image processing. For example, in the infancy of this area of research, the 1960's, one of the major concerns among researchers was to develop a system that could recognize written and typed characters. (Work in this area was even accelerated by the tensions between the USA and the former USSR because espionage agencies on both sides wanted to develop an OCR system for rapid document scanning.) In 1962 a paper was published that illustrates possible methods to recognize characters and one of the methods pointed out was a device that added the input character to a positive and negative template and if the set of additions produced either a full or null signal then the character was present [35]. This paper and its contemporaries can be considered as the origin of optical character recognition (OCR), which is a branch of computer vision.

One of the major issues researchers face when it comes to template matching, no matter the paradigm, is speed. Speed is an issue because in order to fully represent an object many template images of the object at different angles and scales are needed during the matching process. This means that the computational cost to compare all of these images will add up quickly. This is why some of the most influential papers in template matching are based on methods to increase efficiency. In 1977 Rosenfeld and Vanderbrug proposed the idea of coarse to fine template matching, where the intensity based templates would first be matched against coarse image representations and if a match above a specified threshold was reached then that template would then be matched against the higher resolution image to better locate the matched object [36]. In 1995 Lewis showed that the cross correlation similarity metric, a metric used by intensity based templates, could be efficiently normalized with pre-computed tables [37]. The direct application of this work was for the special effect scenes in the movie *Forest Gump*. In 2002 Tsai and Lin developed another method to speed up the computation of the normalized cross correlation method in order to make this metric better suited for the speed demands of a factory [38]. In 2014 Wu and Toet used integral images and weak classifiers to speed up the computation efficiency of template matching [39].

Intensity based template matching is very successful in controlled environments and will continue to be used in industrial application where the environment can be controlled. For example, this paradigm is successful at inspection of printed-circuit boards, surface mounted devices, wafers, printed characters, fabrics and ceramic tiles [40, 41, 42, 43, 44, 45] and will likely be employed in these types of tasks for a long time to come.

2.2.4 Gradient Based Template Matching

Stepping away from controlled environments and into uncontrolled environments, like outdoor scenes, makes intensity based template matching less attractive because of its inherent inadequacy of coping with variable illumination. There are processing techniques available to help alleviate this problem, but these extra processing steps are usually computationally expensive and result in too much overhead to be practical. Therefore, in an attempt to solve this problem the computer vision industry is moving towards gradient based template matching because of its inherent robustness to illumination changes.

The algorithm used in this thesis relies heavily on gradient based template matching in order to find the waste receptacle. Therefore, this section of the literature review will be more detailed than the previous sections. The layout of this section will consist of a brief overview of gradient based methods and will maintain a chronological order. After the brief overview the focus of the literature review will shift to papers that have cited the following two methods: gradient response maps (GRM) and histogram of oriented gradients (HOG). The reason for this shift in focus is because GRM and HOG are two algorithms that are used in the pipeline of the algorithm created in this thesis.

Before an overview of gradient based methods is given, a short explanation of this method will be given below. Figure 2.9 shows gradient based template matching. It is very similar to intensity methods but instead of pixel values, the information that is compared while parsing the scene is gradient directions, or orientations.

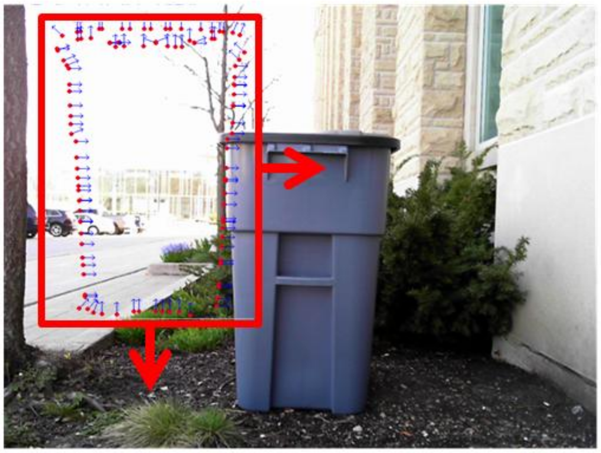


Figure 2.9: Gradient based template matching

In 1997 Olson and Huttenlocher used oriented edge pixels for automatic target recognition [46]. The method worked by comparing two sets of edge points, one set belonging to the template and the other set belonging to a region of interest (ROI) within a scene. These sets are compared using a modified Hausdorff measure, which incorporates orientation information and allows for partial occlusion. The modified Hausdorff measures involves a large number of min and max optimizations between sets of points and as a result is computationally expensive. To alleviate this problem the authors also discussed [46] an efficient searching strategy that is based on hierarchical cell decomposition of the transformation space. By using this strategy large volumes of the pose space can be discarded quickly and only the remaining sections will then need to be searched. Although this method is interesting it suffers from an undesirable number of false positives and is still too computationally slow to be used as a solution to the problem presented in this thesis, as well as most industrial inspection tasks.

In 2002 Carsten Steger developed an object detection metric that is invariant to occlusion, clutter, and illumination changes [47]. The metric works by using the normalized dot product between two global feature vectors. Both vectors consist of location specific direction vectors and one of the vectors belongs to the template image and the other vector belongs to an ROI in the scene being parsed. This metric is modified by the authors of the GRM algorithm [48] and so will be explained in more detail during this review.

A version of the Steger metric is shown in (2.1. The constituents of this equation are: d_i' is the ith direction vector in the transformed template set of points, and $e_{q+p'}$ is the direction vector of location q plus offset p' in the input scene. The numerator of this equation is the absolute value of the dot product between the two vector points, and the denominator is the product of the magnitudes of the vectors. The numerator provides invariance to local contrast changes and the denominator provides invariance to arbitrary illumination changes. This metric is complete when all n points with the template set have been considered, and the sum of these n comparisons is divided by n in order to make templates with variable sizes comparable. Steger reported a 98% recognition rate for the test cases in his paper. (These tests did not contain background clutter). Steger also showed that the similarity score was linearly proportional to the amount of occlusion and so a threshold could be applied to let a desired amount of occlusion be acceptable.

$$s = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|\langle d_i', e_{q+p'}\rangle\right|}{\left\|d_i'\right\| \left\|e_{q+p'}\right\|} \qquad \textbf{(2.1)}$$

In 2005 Dalal and Triggs published a paper describing a novel global feature vector for robust visual object recognition [49]. Since their feature vector is used in this thesis, during the object verification stage, it will be explained in detail in section 3.2. However, for the sake of completeness, in this section a brief overview will be provided. The feature vector developed by Dalal and Triggs is called the histogram of oriented gradients (HOG). The paper [49] centered around human detection as an example to illustrate the effectiveness of this feature vector, but of course it can be used for the detection of any type of object.

The feature vector is created by dividing an image into spatial bins and then for each spatial bin producing a histogram of oriented gradients. The final result is a very large and very descriptive feature vector. This large feature vector is computationally expensive to produce and computationally expensive to compare, but since it is very descriptive it has a high degree of recognition. In fact the accuracy of this feature vector was able to achieve near perfect results on the original MIT pedestrian database, and with that success a new, more challenging, database was created.

In 2008 Hofhauser, Steger and Navab [50] used the Steger metric to match and track objects that have undergone perspective distortion. They accomplish this task under the assumption that spatially coherent structures will stay the same even after undergoing perspective distortion. With this assumption in mind they formed k-clusters of model points around contour sections that are spatially coherent. They then assigned what they call the cluster directions to each cluster, where a cluster direction is a score that indicates the consistency of directions within the cluster. That is if all directions within a cluster are the same then a value of one is produced, and if many of the directions differ, within the cluster, then a value much smaller than one is produced. This method was shown to work fairly well, but the amount of test cases was limited and the computationally efficiency was again too slow to be used for real time applications.

In 2010 Hinterstoisser et al. developed an algorithm for real time object detection that is robust against illumination changes, occlusion, and small deformations [51]. The algorithm worked by creating a template representation called the dominant orientation template, DOT for short. Many of the ideas explored in this algorithm are the same as those in the GRM algorithm and so this algorithm will be explained in a little more detail. This algorithm works by creating a template which is based on dominant gradient orientations. Within a given spatial area a poll is taken and the orientation of the bin with the most votes is used for the dominant orientation of that area. This dominant orientation idea is then used to process the input image, and once the scene has been processed a simple metric that counts the number of matches between template and scene locations is used to calculate a recognition score. One of the highlighted features of this algorithm is that since the scene is split into larger spatial bins the parsing process does not have to look at every pixel location, which means fewer calculations are needed and the process approaches real time. In the end this algorithm was described as being successful in handling small deformations, illumination changes and partial occlusion, but if the object boundaries are cluttered by strong gradients in the background then the dominant orientations would change drastically and the algorithm would fail. This disadvantage was addressed by Hinterstoisser et al. in the paper regarding GRM [48].

In 2012 Thanh et al. combined the generalized distance transform and orientation maps to create a method for object detection that is more robust to strong background clutter [52]. When this method was published it was compared to other contemporary algorithms and was shown to have a better true positive and true negative rates, but this algorithm fails in comparison to GRM which was published later.

In 2012 Hinterstoisser et al. published a paper on a method they developed called gradient response maps (GRM) [48] (In their paper the authors refer to the method as LINE-2D but in this thesis it will be referred to as GRM). Since this algorithm is a key component in the pipeline developed in this thesis it will be explained in detail in section 3.1. However, for the sake of completeness in this section a very brief overview of the algorithm will be provided.

In the original paper three methods for template construction were described that use GRMs for object recognition. They were LINE-2D, LINE-3D and LINE-MOD (which is a combination of the previous two), however, it was pointed out by the authors of [48] that both LINE-3D and LINE-MOD would not work for outdoor scenes because of too much infrared noise, and so with that restriction, and the constraint mentioned earlier, the author of this thesis was left with the LINE-2D method (from here on will be referred to as the GRM method.)

The GRM method is used for real time detection of texture-less objects. At its core, this method is a gradient based template matching algorithm. It works by representing an image scene in terms of GRMs, and by parsing this scene with multiple object templates, each of which is made up of a collection of gradient orientations along the edges of the object. This method is considered state-of-art [27] but the LINE-2D version which is used in this thesis is prone to numerous false positives and additional processing will be introduced to alleviate this drawback.

2.2.5 Recent GRM and HOG Applications

Both the GRM algorithm [48] (50 citations) and the HOG algorithm [49] (8272 citations) have been rapidly adopted by the computer vision community. Yao et al. developed a learning algorithm that could perform fine-grained image categorization [53], where fine-grained image categorization is the act of categorizing objects that are highly similar and only differentiated by subtle features. In order to accomplish this task Yao et al. used feature response maps in their pipeline, which were inspired by gradient response maps [48].

Hsiao and Herbert developed a method for modelling occlusions for object detection under arbitrary viewpoints [54, 55, 56]. Their method is an attempt in unifying texture-less object detection, viewpoint changes, and occlusion handling. One of the key algorithms used in their pipeline is the GRM algorithm [48], which is used for finding an object type and location hypothesis. This type and location hypothesis is then used in conjunction with occlusion hypothesis to determine the likelihood of there being a match.

Chen et al. developed a robust head and hands tracking algorithm intended for human machine interaction that uses GRM in the pipeline [57, 58]. The algorithm works by combing heuristics, colour information, posterior probabilities, and shape modelling. More specifically the GRM algorithm [48] was used for the shape modeling aspect of this algorithm.

Rios-Cabrera and Tuytelaars developed an algorithm that combines GRM, boosting, and cascades to detect 3D objects [59]. They reported an increase in both speed and accuracy when compared to the original GRM algorithm. This methodology used is intriguing and is a source for future work.

Ersen et al. use the LINE-MOD version of the GRM algorithm and HS histograms to locate simple objects within a scene [60]. After the objects are located the spatial relationship among the objects is determined. The purpose of this algorithm is to detect failures during the planning of execution for robotic movement actions.

Karapinar et al. investigated how robots can maintain robustness by gaining experience [61, 62]. The system they developed is called Inductive Logic Programming (ILP) and is intended to be a lifelong experimental learning program. In their pipeline they used the LINE-MOD version of the GRM algorithm to recognize objects within a scene.

Hinterstoisser et al. extended their work with LINE-MOD to included 3D models of objects [63]. They also showed that by using pose estimation and color information the initial hypothesis could be verified. Their new method resulted in a 13% increase in correct detection.

Zhu et al. used HOG to describe salient features of objects/humans and then passed these features to an AdaBoost algorithm, which reduced the set of features to include only the most representative [64]. Using this approach in conjunction with rejection cascades during scene parsing increased the rate of human detection by seventy times when compared to the classic HOG algorithm.

Felzenszwalb et al. use a parts based model to detect objects [65, 66], where each part of this model is described by a variation of HOG. This methodology is very successful at human detection and is a focus of many researchers today. However, one of the big disadvantage to splitting the model into parts is the decrease in speed, which is due to the increase in templates. Recently Dean et al. describe a method that can be used to speed up a parts based model [67] that uses HOG as the means to describe the parts. They reported that 100,000 objects could be simultaneously searched on a single machine in 20 seconds.

2.3 Suppressing Ambient Contours

One of the contributions made by this thesis is the addition of a contour filtering step to the GRM algorithm pipeline. This step is used to suppress spurious edges relating to the background foliage in outdoor scenes where the waste receptacle is located. By suppressing these edges the false positive rate of the GRM algorithm is reduced. This section will serve to highlight the other contemporary algorithms that can suppress background edges.

One possible way to divide the literature is to categorize these edge suppression/enhancement algorithms into one of the three following categories: smoothing based methods, gradient image based methods, or labeling based methods. The distinction between these categories is made by determining where the processing step related to suppressing background edges and enhancing object edges takes place. For instance, smoothing based methods apply algorithms to the input image, gradient image based methods apply algorithms to the contour image, and finally the labeling based methods apply algorithms to the final result. This can be thought of as determining if the processing step takes place at the beginning, middle, or end of the algorithm. The method used in this thesis belongs the gradient image based methods and so a detailed literature review on this category will be given, however, a brief overview of the other two will be provided first.

Smoothing based methods are concerned with trying to reduce the amount of noise and spurious edges by convolving an image with noise suppression filters in the spatial domain. This method can be further divided into methods that try to use a single optimal scale parameter to filter the image [68, 69, 70] or methods that employ multi-scale techniques to filter the image[71, 72, 73, 74]. For the single scale methods, there exists a trade-off between texture removal and edge localization: that is the more you filter, the blurrier the edge gets. The multi-scale techniques are able to achieve high degrees of texture removal while maintaining good edge localization, but this method is computationally unattractive.

Labeling based methods [75, 76, 77, 78] use the final result of previous edge segmentation algorithms to refine the optimal threshold(s) that separate object and background. These methods provide a great improvement in the detection of object edges and the removal spurious edges, but are really only effective when the textured edges surrounding the object are not stronger than the object edges themselves. The reason this is a problem is because no matter how optimal the thresholds are the spurious/textured edges will survive and make it into the final edge map. Therefore, since most natural images produce ambient edges that are stronger, or as strong as, the object edges, this methodology would fail to work in this thesis.

The algorithm used in this thesis [79] to suppress the spurious background edges belongs to the gradient image based methodology and will be explained in detail in the background section of this report. Grigorescu et al. [80] integrated a computational step with the Canny edge detector [81] to suppress spurious edges. Their algorithm was inspired by neuron responses in the primary visual cortex and was shown to be fairly successful, however, a main drawback was that it suffers from object boundary suppression. This problem was addressed by Qu et al. [82]. In their paper they proposed using the SUSAN [83] algorithm to first divide edges into two groups, object contours and texture contours, and then to suppress the texture contours using the method outlined by Grigorescu [80]. Essentially, both of these methods use filtering kernels within the spatial domain. This is a disadvantage because it is well known that filtering in frequency domain can be faster than filtering in the spatial domain. This type of filtering was explored in [79] by ZhiGuo et al. and was shown to be faster and better at suppressing textured edges, which is why it was the method of choice in this thesis. However, it should be noted that in [79] the filtering was used in conjunction with the Canny edge detector [81] and in this thesis the filtering technique was used in conjunction with the gradient magnitude image after thresholding to produce the final contour image.

Chapter 3

Background

This chapter of the thesis serves to provide the reader with much of the background knowledge they would need to understand the key components used in the algorithm pipeline proposed by this thesis. If the reader is unfamiliar with image processing techniques it is recommended that they read appendix section 7.1 before proceeding with this chapter. This chapter is split into the following sections. Section 3.1 will provide background information on the GRM algorithm [48]. Section 3.2 will provide background information on the HOG algorithm [49]. Finally, Section 3.3 will provide background information on the contour filtering algorithm [79] used to modify the GRM algorithm and reduce the presences of spurious edges. The intent of these sections is to describe the algorithms in sufficient detail so that the reader does not have to consult additional papers to understand the pipeline components.

3.1 Gradient Response Maps

Hinterstoisser's method [48] is used for real time detection of texture-less objects. At its core, this method is a gradient based template matching algorithm. It works by representing an image scene in terms of GRMs, and by parsing this scene with multiple object templates, each of which is made up of a collection of gradient orientations along the edges of the object.

3.1.1 Derivation of the Similarity Metric

The similarity metric used in the GRM algorithm is an improved version of the similarity metric introduced by Steger [47]. This improved metric allows templates to be matched against objects within a scene that have undergone small deformations and translations. The improvement is given below in equation 3.1, which shows the cosine version of the unmodified Steger metric, and equation 3.2, which shows the modified version. The constituents of equation 3.1 are as follows: I is the input image scene, T is the template, c is the pixel under consideration in the input scene, r is a pixel location within the template set P, $ori(x, y)$ returns the orientation at point y in image x. The additional constituents of equation 3.2 are as follows: max returns the local maximum within a neighborhood, and t is a point within the neighborhood defined by $R(c + r)$.

$$\varepsilon_{Steger}(I,T,c) = \sum_{r \epsilon P} |\cos\left(ori(T,r) - ori(I,c+r)\right)| \qquad (3.1)$$

$$\varepsilon_{Steger}(I,T,c) = \sum_{r \epsilon P} \left(\max_{t \in R(c+r)} |\cos\left(ori(T,r) - ori(I,t)\right)|\right) \qquad (3.2)$$

The modified equation allows the template points to shift around a small local neighborhood. This change allows the template points to be compared to all the points within a local neighborhood around the expected point location within a scene. Therefore, if an object has undergone small deformations and/or translations during input scene capturing it can still be reliably matched to its correct template. The images below illustrate how this works.

First consider Figure 3.1 which shows the template created from a blue circle object. As you can see the template is a finite set of points located around the border of the object. Each red dot and arrow represent an element within the template set, where the red dots represent the element's location and the arrows represent the elements gradient orientation. The dashed blue line is there to illustrate the border of the object.

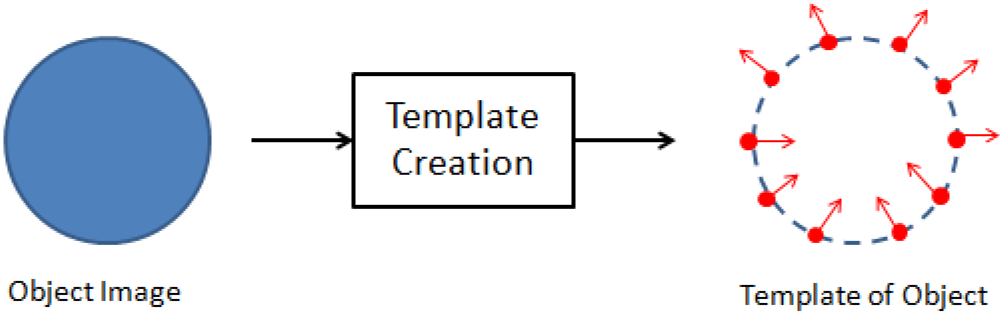


Figure 3.1: Template Creation

Now consider that this object is present within an input scene image, but the object has undergone some kind of deformation. Figure 3.2 shows a simple uniform scaling deformation of the object.

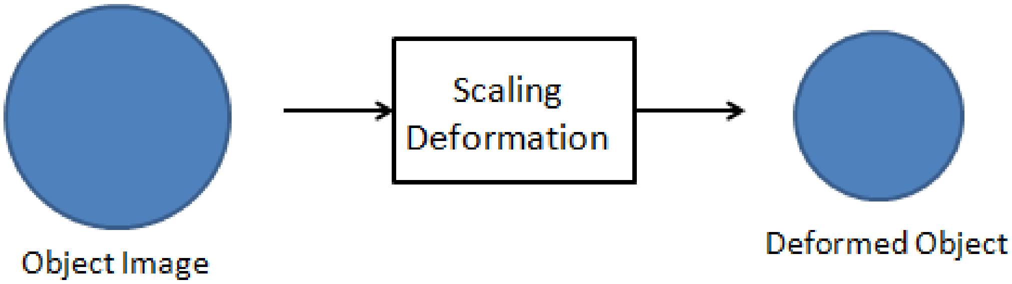


Figure 3.2: Simple Scaling Deformation

Applying the template created in Figure 3.1 to the input scene object in Figure 3.2 and using the unmodified Steger metric yields a low score as a result because the template points do not lie on the object boundary (see Figure 3.3).

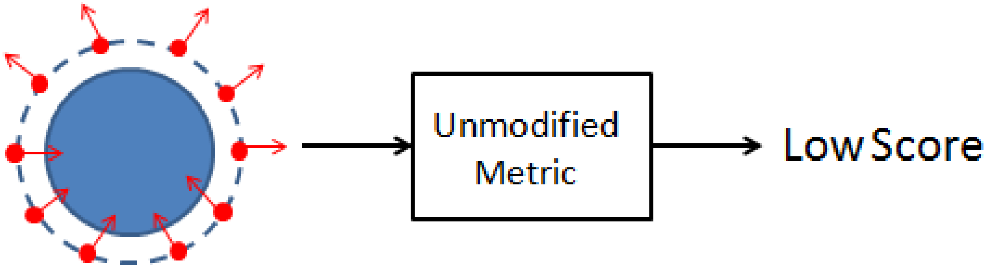


Figure 3.3: Unmodified Steger Metric

The modified version of the Steger method, which is used in the GRM algorithm, easily handles this simple case, and ones that are more complicated. Figure 3.4 and Figure 3.5 show how the modified Steger method works. Before the final similarity score is computed each point within the template set searches a local neighborhood (Figure 3.4) and finds the location with the highest correspondence within the input scene. Figure 3.4 only shows one point being moved but this process is done for all the points within the set.

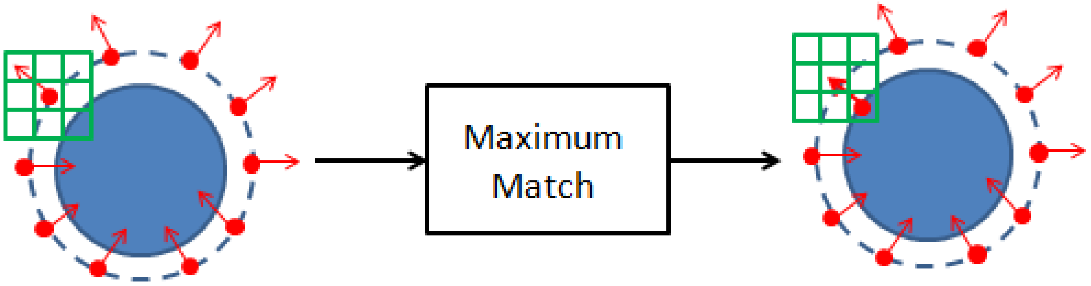


Figure 3.4: Find Maximum with Local Neighborhood

Figure 3.5 shows the final result of searching within all of the local neighborhoods and how this result would produce a higher score when assed with the similarity metric.

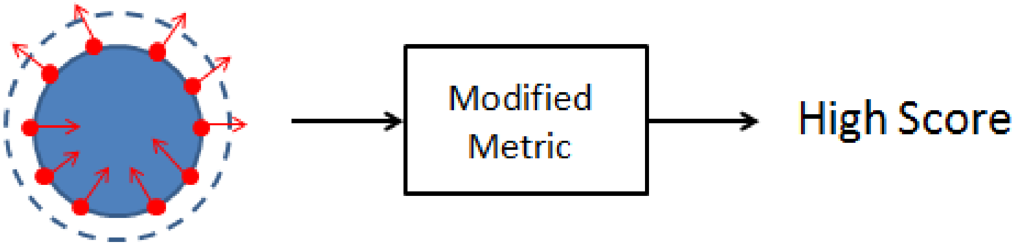


Figure 3.5: Modified Steger Metric

3.1.2 Template Creation

The process used to create the template for the GRM algorithm is one of its unique features because it is able to create a template representation with relatively few feature points. This is desired because similarity calculations can be preformed quickly when fewer feature points are used. Figure 3.1 shows the applicability of the technique to texture-less objects, in which the goal of the template creation algorithm is to capture the boundary information of the object. The following figures will explain this processing in more detail.

The first step of the template creation algorithm is to produce three gradient magnitude maps, one for each channel of the RGB image of the template. For illustrative purposes a waste receptacle image was used. Figure 3.6 shows this step.

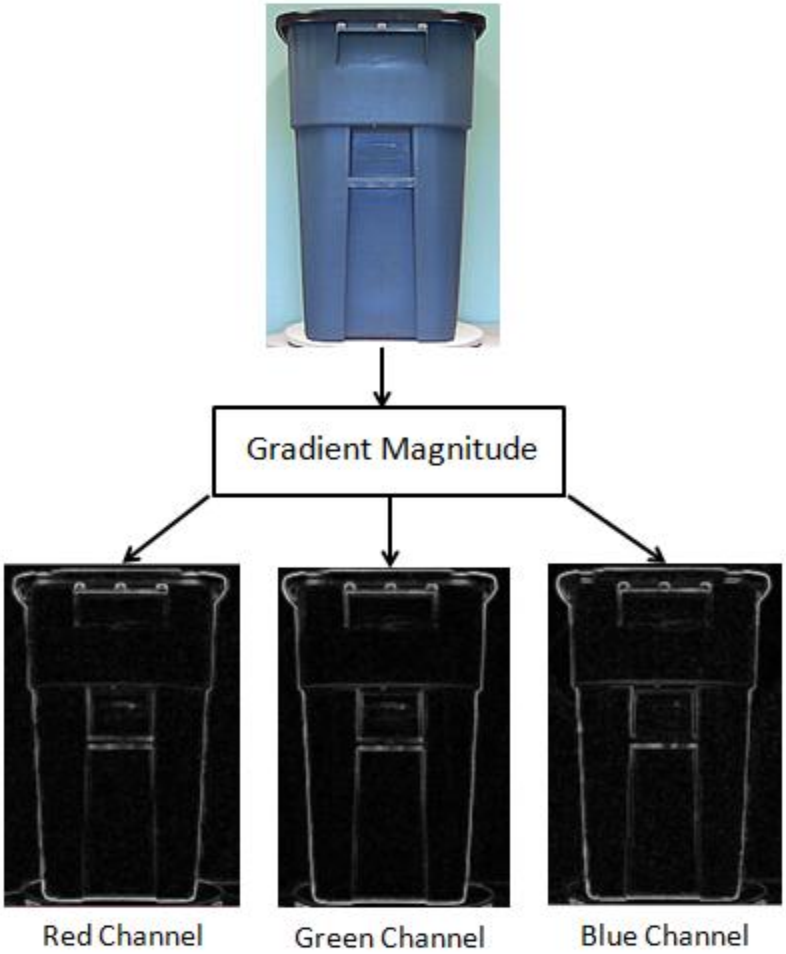


Figure 3.6: RGB Gradient Magnitude Maps

The second step of the template creation algorithm is to produce an orientation map. This step uses each gradient magnitude map from the previous step. The gradient orientation map is populated with the orientation of each color map, if and only if it is the maximum magnitude across all three maps and it is above a designated threshold. (See , where *I* is the orientation map and *C* is the map index for the map which has the maximum magnitude at a specific location.) Figure 3.7 shows the magnitude map produced using the maximum across each colour channel and compares this result to a magnitude map of a gray level image. This figure shows that more information is captured when the maximum channel magnitude is used, as indicated by the additional contour lines, and that the intensity values in the maximum channel approach are actually higher than those in the gray level only version, as indicated by the brighter contours. This is important because it means that the boundary information has a better chance of passing the designated threshold when the orientation map is constructed.

$$I_g(x) = ori\left(\hat{C}(x)\right) \tag{3.3}$$

$$\hat{C}(x) = \underset{C\in\{R,G,B\}}{\arg\max} \left\|\frac{\partial C}{\partial x}\right\|$$

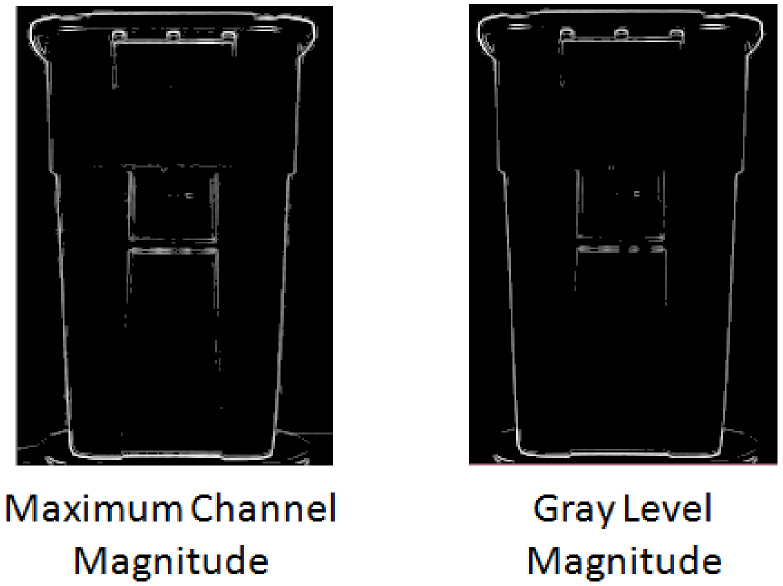


Figure 3.7: Magnitude Maps

The third step in the process is to quantize the orientation map into discrete values. In order to do this GRM omits the gradient direction and only considers the orientation (i.e., two gradient directions separated by 180 degrees share the same orientation) and then splits the orientation space into n_o equal spaces, as shown in Figure 3.8. The advantage to using orientations and not directions is that when used by the similarity metric mentioned above it does not matter if the object intensity values are brighter or darker than their surroundings, which makes the method invariant to object occluding boundaries.

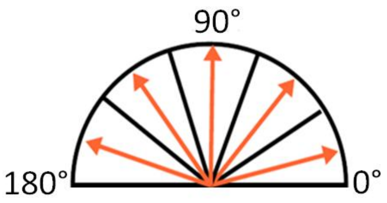


Figure 3.8: Gradient Quantization

The fourth step is to encode the quantized orientations. The encoding scheme used by the GRM algorithm is one-hot encoding. The purpose of this encoding scheme is to accelerate the orientation spreading in the input scene processing stage and to provide a convenient indexing method. This will be explained more clearly in the next section. Figure 3.9 shows the encoding scheme used for the orientation bins. In Figure 3.8 and Figure 3.9 only five bins were used to demonstrate the point, but in the GRM algorithm the number of bins used was eight.

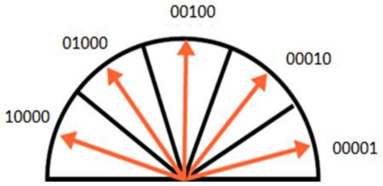


Figure 3.9: One-Hot Encoding

The fifth step of the template creation algorithm is to poll each location in the orientation map and reassign it the orientation that occurs most often within a local neighborhood. This will reduce the amount of noise present within the orientation map. Figure 3.10 shows the orientation map before and after noise reduction. The orientation assignments are more consistent in the map after noise reduction.

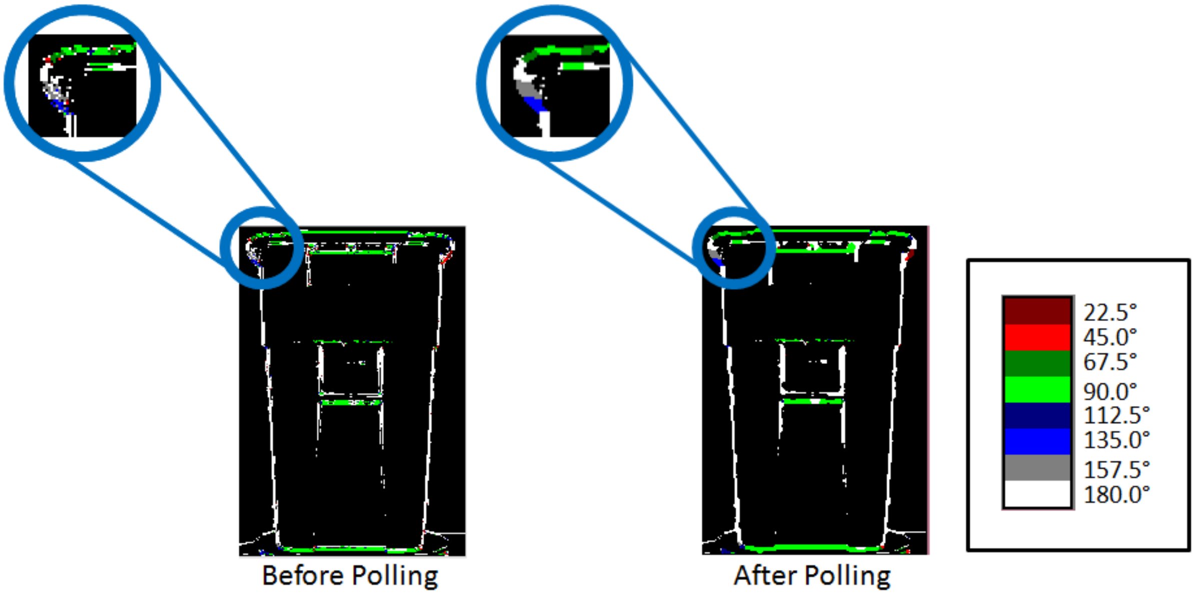


Figure 3.10: Noise Reduction via Polling

The sixth step is to create a set of template points that represent the object. The process is accomplished by selecting the most discriminate gradient orientations based on where the highest gradient magnitudes exist. The author of the GRM algorithm also indicates that in the selection process the location of the points must be taken into consideration in order to avoid points forming a clusters around spots with a high gradient magnitudes and ignoring other relevant boundary areas. The method for how to take location into account is not specified in the paper [48]. In the implementation for this project a grid of local neighborhoods was created and only one point per neighborhood could enter the template set. Figure 3.11 shows the final product of template creation.

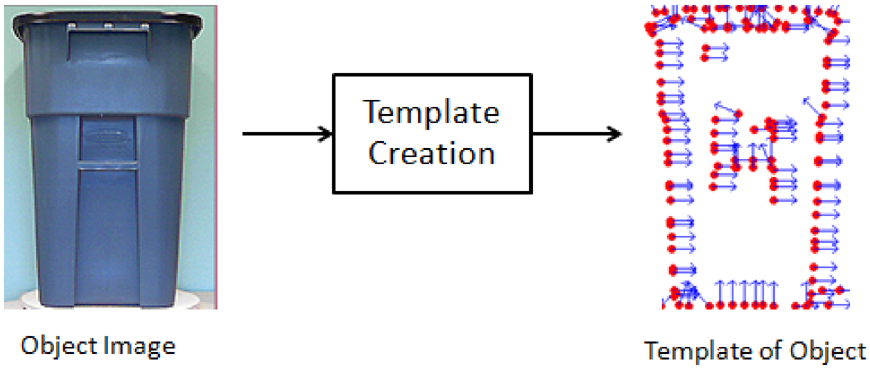


Figure 3.11: Final Set of Template Points

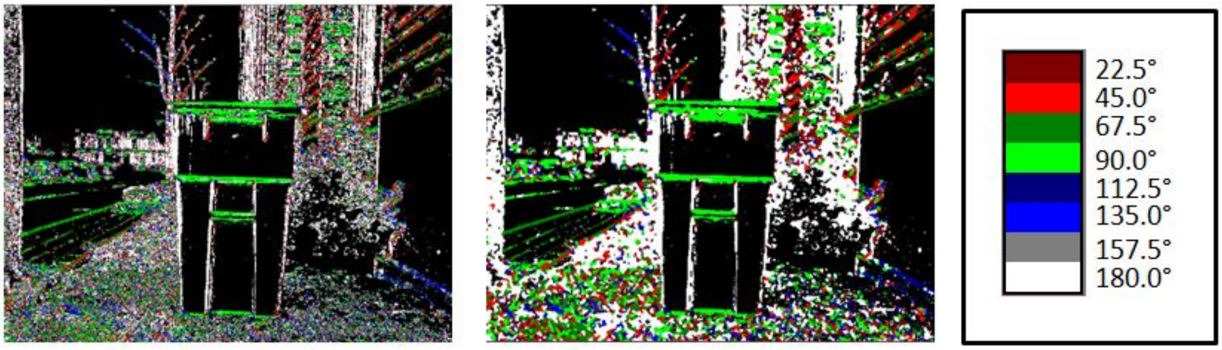
3.1.3 Making the GRMs

The gradient response maps as used in the GRM algorithm, drastically improve the execution speed of the Steger similarity metric by redefining how it is evaluated. The equations below show the modified Steger metric and the redefined version of it that uses the GRMs to facilitate execution. Where $S_{ori(T,r)}$ is the gradient response map referenced for the orientation at point r in template T and the value returned from $S_{ori(T,r)}(c + r)$ is the response value in the gradient response map at pixel location $c + r$. The reason the speed of execution is increased is the entire calculation per pixel location is reduced to a look up table (i.e., no more max operator and absolute cosine calculations per pixel location). The figures below will help clarify the derivation of this new metric notation.

$$\varepsilon_{Steger}(I,T,c) = \sum_{r\epsilon P}\left(\max_{t\in R(c+r)} \left|\cos\left(ori(T,r) - ori(I,t)\right)\right|\right)$$

$$\varepsilon_{Steger}(I,T,c) = \sum_{r\epsilon P} S_{Ori(T,r)}(c+r) \qquad (3.4)$$

Many of the same steps involved in creating the template are repeated in the process of creating the gradient response maps. In fact, all of the steps except the last one are repeated; the set of template points is not created. Figure 3.12 shows an input scene processed all the way up to and including the noise reduction stage.



Before Noise Reduction After Noise Reduction

Figure 3.12: Noise Reduction of Scene

After this point the process of making the gradient response maps differs from that of making the object template. The next step in the process is to spread the orientation of each point in the orientation map around a local neighborhood. The purpose of this step is twofold. First spreading the orientations will eliminate the neighborhood search for the max orientation response in the modified Steger metric, and second it will allow the scene parsing algorithm to skip the size of the neighborhood used in the spreading step when searching for a template match. Both of these advantages increase the speed at which the algorithm can perform template matching. Before an explanation is given about how the spreading of the orientation accomplishes both of those tasks, an explanation should be given about spreading the orientations.

Figure 3.13 shows what is meant by spreading the orientation around a local neighborhood. First consider the orientation map before spreading, as shown by the left most image. The red arrows represent the orientations that were above a given threshold. Second consider the orientation point in the center of the red box in the middle image. Spreading this orientation around a 3x3 neighborhood would result in the blue orientation arrows. Finally if this spreading operation was repeated for all of the original orientations then the final result would be the right most image.

Some of the pixel locations in the final image no longer store just one orientation, they now store multiple orientations. In fact in a real input scene a pixel location can have all orientations present. The purpose of the orientation spread is to avoid the max operator in the modified Steger metric. After the spreading operation each pixel location will carry its own information along with its neighbors' information, which means that each pixel location contains all of the orientations within a local neighborhood. Therefore, there is no need to search in a local neighborhood.

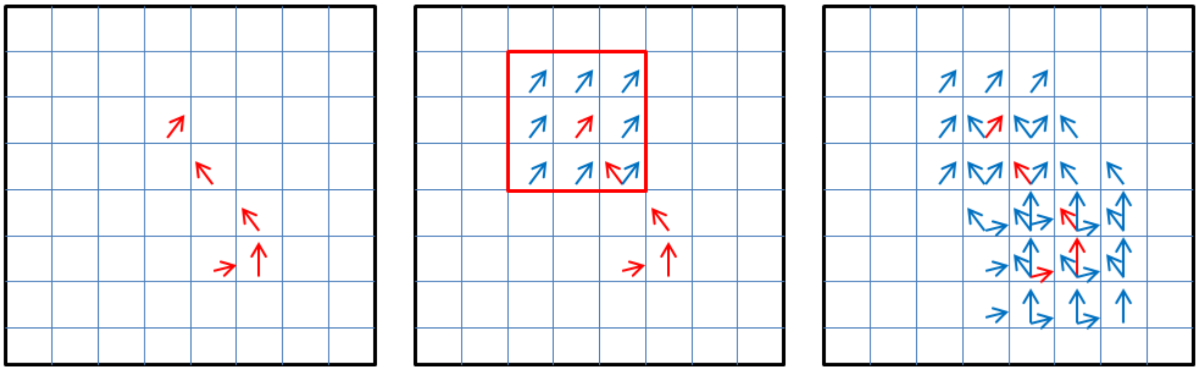
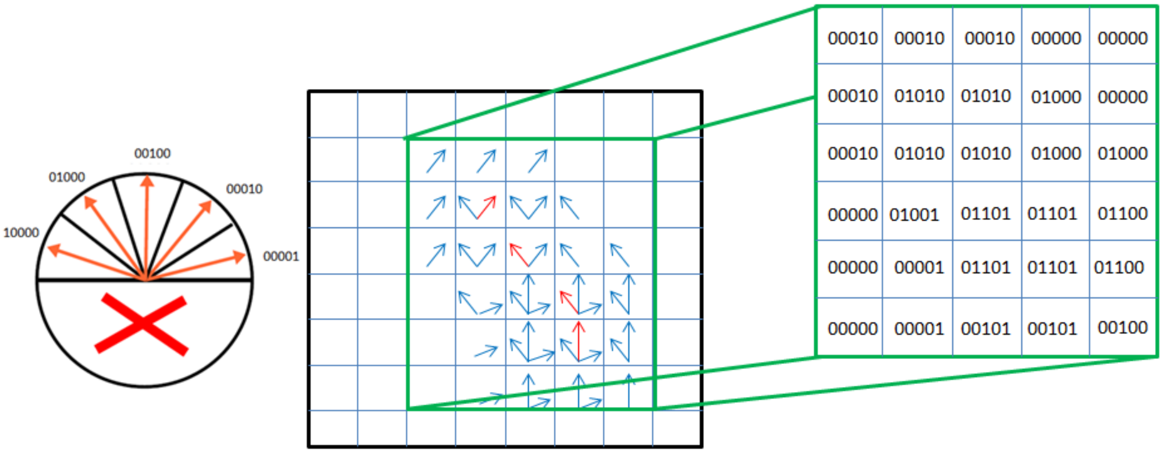


Figure 3.13: Spreading the Orientations

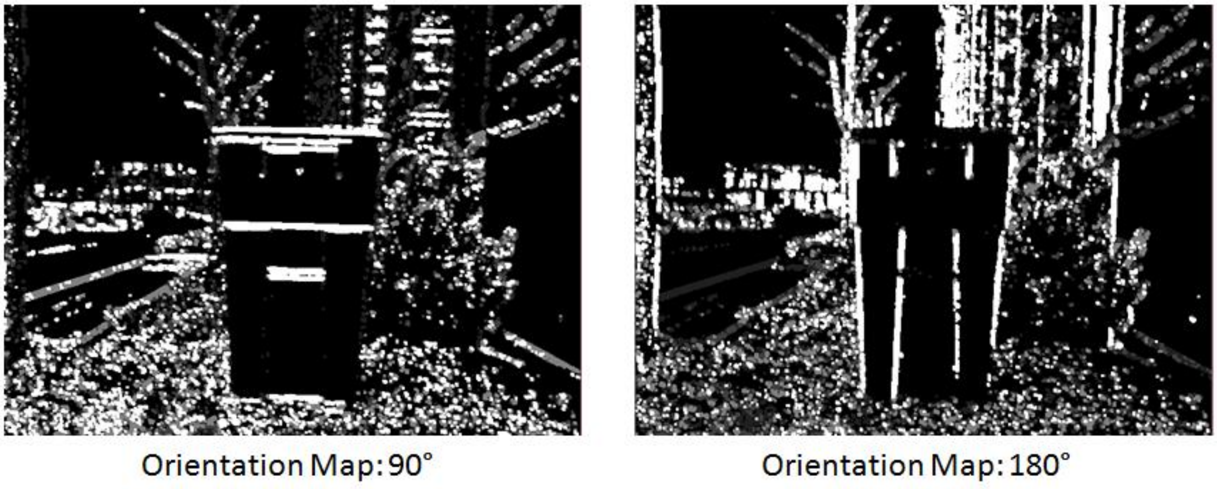
Recall that the orientations have been encoded using the one-hot encoding scheme so a raster patch of the image could be shown using binary strings in the pixel locations. Figure 3.14 shows an image patch using the one-hot encoding scheme. It was mentioned earlier in this report that the one-hot encoding scheme aided in the spreading process. The reason this is true is because since each orientation is represented as a unique bit, in a bit string, the process of spreading can be accomplished by ORing each orientation with its neighbors. Since computers can perform ORing operations very quickly the process of orientation spreading can be accomplished in a time efficient manner.



00010	00010	00010	00000	00000
00010	01010	01010	01000	00000
00010	01010	01010	01000	01000
00000	01001	01101	01101	01100
00000	00001	01101	01101	01100
00000	00001	00101	00101	00100

Figure 3.14: One-Hot Encoding Facilitates Spreading

Recall the definition of a gradient response map: a GRM is a map that stores the similarity scores between the orientations within a scene and one of the quantized orientations. Therefore, if there are eight quantized orientation then there will be eight maps: one for each orientation. Figure 3.15 shows two GRMs: one for 90°, and the other for 180°. The grey level value of the pixels indicate how close the orientations in the scene match the predefined orientation. An exact match between orientations is white(RGB[255,255,255]) , and an orientation match with a 90° separation is black (RGB[0,0,0]). For instance, in Figure 3.15 the 180° orientation map has the brightest pixels along vertical edges, and the darkest pixels along horizontal edges, because the vertical edges match the predefined orientation of 180° and the horizontal edges are separated by 90°. Edges with a separation between 0° and 90° are given intermediate gray level values between RBG[0,0,0] and RGB[255,255,255].


Orientation Map: 90°

Orientation Map: 180°

Figure 3.15: Gradient Response Maps for 90° and 180°

The final orientation map (i.e., the orientation map that has undergone spreading) is used to create GRMs. Following this discussion will be a discussion of how the GRMs can be used to redefine the modified Steger metric to make its calculation more efficient. Figure 3.16 shows how the gradient response maps are calculated. For illustration purposes only two response maps are shown (90° and 180°), but for this example there would be six maps in total since the orientation space has been split into six bins.

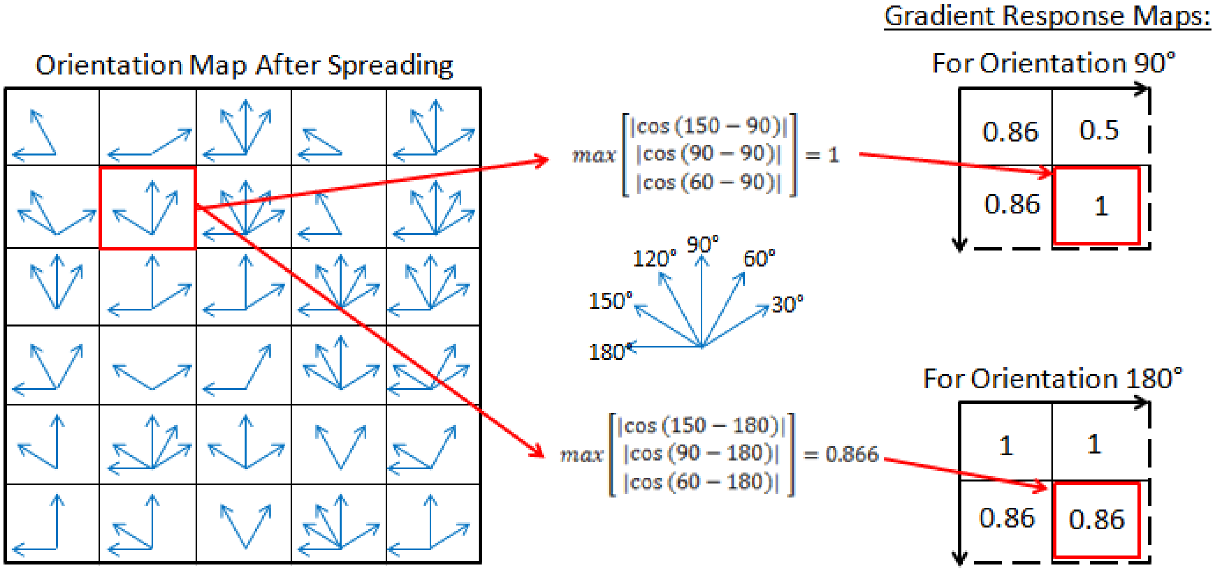
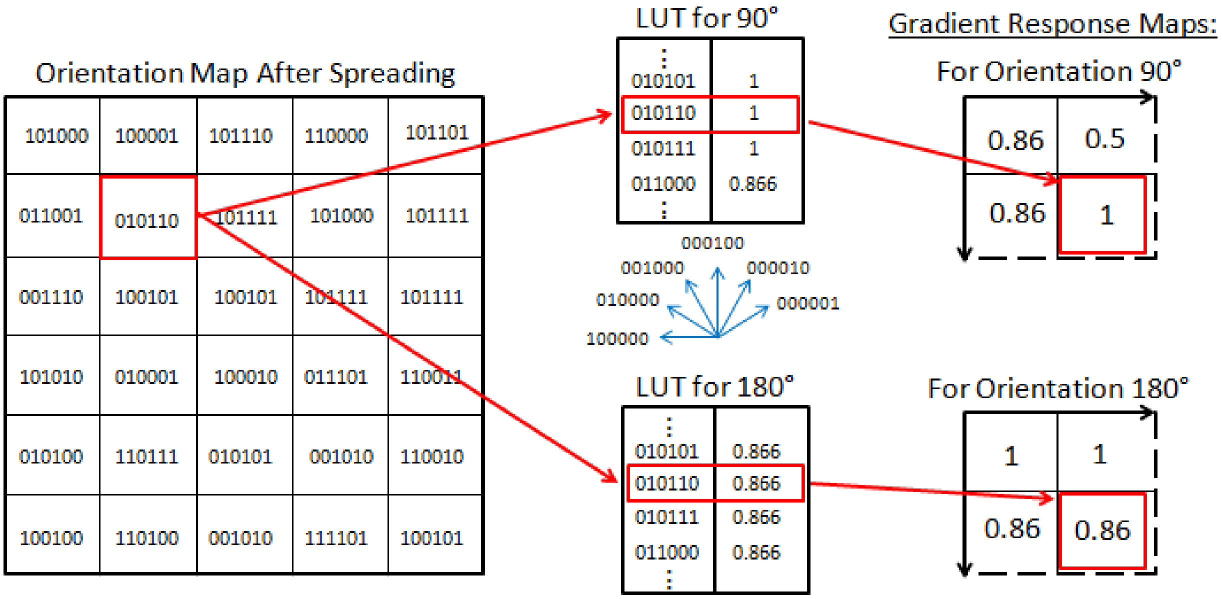


Figure 3.16: Naive Approach to Calculating GRMs

The method used to calculate the gradient response maps in Figure 3.16 can be considered the naive approach. The reason this is the naive approach is because if it were implemented then for every map and every pixel location a set of involved calculations would need to be performed at run time. These calculations include: one maximum calculation, a set of absolute value calculations, and a set of cosine calculations. Also before these calculations could be performed a decoding step would be needed to read which types of orientations were present at each pixel location. These steps and calculations would be very costly to perform a run time, but fortunately they are avoided.

The better approach takes advantage of the encoding scheme and makes use of look up tables (LUTs). Figure 3.17 shows the more sophisticated approach. The orientation map uses the bit strings provided by the one-hot encoding scheme and spreading process to index a set of LUTs, where the value at each indexed location is the maximum response for that set of orientations. Using LUTs considerably increases the speed at which the GRMs can be constructed because now there are no lengthy set of calculations that need to be performed at run time. Of course the calculations still need to be done in order to create the LUTs, but they are done offline and stored in memory.



101000	100001	101110	110000	101101
011001	010110	101111	101000	101111
001110	100101	100101	101111	101111
101010	010001	100010	011101	110011
010100	110111	010101	001010	110010
100100	110100	001010	111101	100101

⋮	
010101	1
010110	1
010111	1
011000	0.866
⋮	

0.86	0.5
0.86	1

⋮	
010101	0.866
010110	0.866
010111	0.866
011000	0.866
⋮	

1	1
0.86	0.86

Figure 3.17: Sophisticated Approach for Calculating GRMs

Once all of the gradient response maps have been computed they can be used to redefine how the modified Steger method is computed. The equations for the modified Steger metric and its redefined version using GRMs have been reproduced below for ease of reference.

$$\varepsilon_{Steger}(I,T,c) = \sum_{r\epsilon P}\left(\max_{t\in R(c+r)}\left|\cos\left(ori(T,r) - ori(I,t)\right|\right)\right.$$

$$\varepsilon_{Steger}(I,T,c) = \sum_{r\epsilon P} S_{ori(T,r)}(c+r)$$

Now, instead of summing all of the maximum responses of the absolute dot products between the template orientations and every orientation within a local neighborhood all that is required is to use the GRMs as a set of LUTs and sum the values returned. Where S_i is a GRM corresponding to orientation bin i and $Ori(T,r)$is the orientation of template point r in the set P and is used to reference the proper GRM. For example, let's assume that template point r in P has an orientation of 90°. Therefore, in order to check how the input scene responds to this orientation we need to reference the 90° GRM. Thus, $S_{90°}$ will reference of the correct GRM and the value stored at location $c + r$ will be returned as an element in the summation of the total similarity score.

3.2 Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients (HOG) is an algorithm that builds a global feature vector of a template image. It was developed to solve the human detection problem and at the time of its publication it was one of most successful algorithms in use. HOG is used in this thesis to verify a candidate waste receptacle location. The algorithm described below is based on the formulation of Dalal and Triggs [49].

3.2.1 Algorithm Overview

The HOG algorithm consists of five steps which transforms an input image into a HOG descriptor. The steps include: Normalization of gamma and colour, computation of gradients, weighted vote into spatial and orientation cells, contrast normalization over overlapping spatial block, and final the production of HOG's over detection window. Figure 3.18 shows the pipeline of the steps involved. The following sections will describe each of these subcomponents.

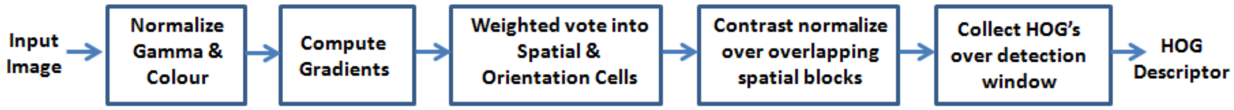

Input Image
Normalize Gamma & Colour
Compute Gradients
Weighted vote into Spatial & Orientation Cells
Contrast normalize over overlapping spatial blocks
Collect HOG's over detection window
HOG Descriptor

Figure 3.18: HOG Algorithm

3.2.2 Normalize Gamma and Colour

There are many types colour normalization and gamma correction techniques, most of which are a combination of point and algebraic operators (refer to appendix section 7.1.2). The specific type of correction techniques implemented by the original authors [49] is not wholly specified. However, it is mentioned that this normalization step had little effect on the overall success of the algorithm. The authors speculate that this normalization step had only a modest effect because of subsequent normalization further down the pipeline. This is probably a contributing factor, but an additional explanation (not mentioned by Dalal and Triggs) is that when collecting gradient orientation information, lighting correction is not needed because gradient orientations are robust to this type of change.

3.2.3 Gradient Computation

Gradient computation is a fundamental area in image processing and computer vision (refer to appendix section 7.1.2). There are many types of derivative masks that can be used to compute gradients. The HOG algorithm uses a simple 1D centered mask with no smoothing. The masks are shown below in Figure 3.19. The final gradient map was created by taking the gradient from each colour channel that had the largest magnitude.

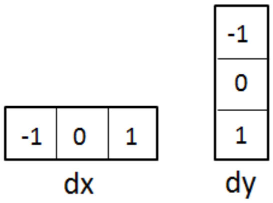


Figure 3.19: 1D Center Derivative Masks

3.2.4 Spatial / Orientation Binning

The operation in this section is a non-linear operation (refer to appendix section 7.1.2.3) that splits the template image into spatial regions called *cells* and then bins the orientation within these *cells* to produce orientation histograms. The original paper [49] describes two types of *cell* structures: circular and rectangular. In this thesis the rectangular 8x8 *cell* structure was used and so will be described. Figure 3.20 shows an image of a waste receptacle split into 8x8 cells.

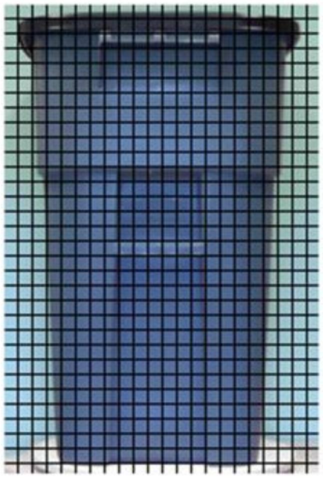


Figure 3.20: HOG Cells (8x8)

For illustrative purposes only a section of the image in Figure 3.20 will be used to describe the rest of the algorithm. More specifically a 400% zoomed in crop of the top left corner will be used. Now that the image has been spatially binned (split into 8x8 cells) the next step is to create orientation histograms for each cell. This process involves the consultation of the gradient maps formed in the step above to create an orientation map for each pixel. Each orientation is calculated as: $ori = tan^{-1}(I_y/I_x)$. After the orientation map has been made the next step is to create an orientation histogram for each spatial bin (i.e., cell). For this thesis each orientation histogram consists of 9 bins and will use bi-linear interpolation during construction. Figure 3.21 shows the creation of a 9 bin orientation histogram for one cell. This process would be done for all cells.

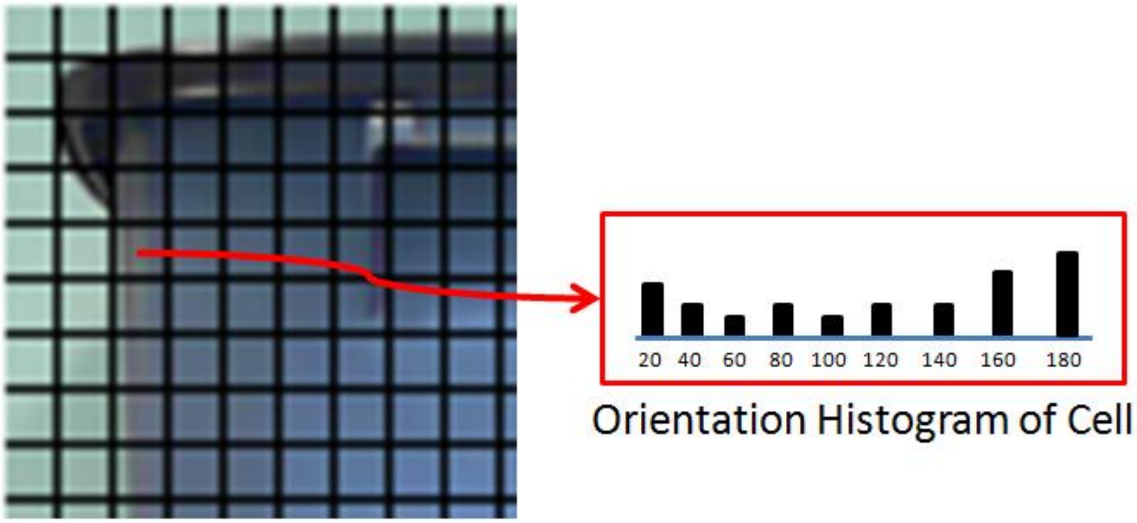


Figure 3.21: Orientation Histogram of a Cell (9 Bins)

3.2.5 Normalization and Descriptor Blocks

The last step of the algorithm is to create *blocks* of *cells* and locally normalize the histograms of each block in order to create the final feature vector. In the original paper [49] the authors experimented with many *block* sizes and overlapping patterns, but for this thesis 2x2 *blocks* of *cells* and a 50% overlapping pattern was used, therefore this section will describe the algorithm with those parameters in mind. Figure 3.22 shows two *blocks* with 50% overlap.

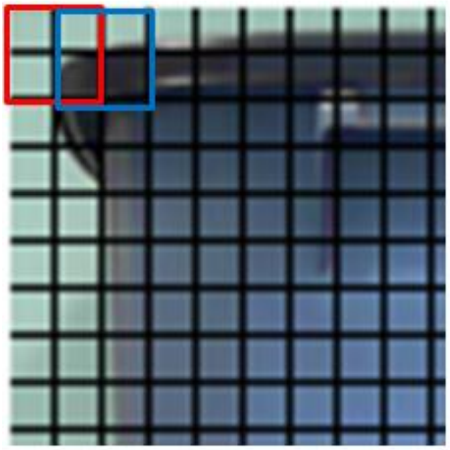


Figure 3.22: Blocks of Cells (2x2 and 50% overlap)

The final feature vector is created by extracting the normalized orientation histogram from each *block*. Since there are four *cells* per *block* each block will produce four normalized orientation histograms. Many types of normalization techniques were tested in the original paper [49], but for this thesis the L2-Hys normalization procedure was used, which is the same procedure used in SIFT [4]. Figure 3.23 shows the creation of the histograms after local *block* normalization. As you can see in Figure 3.23 a *cell* produces multiple orientation histograms and they can be different depending on the result of normalization.

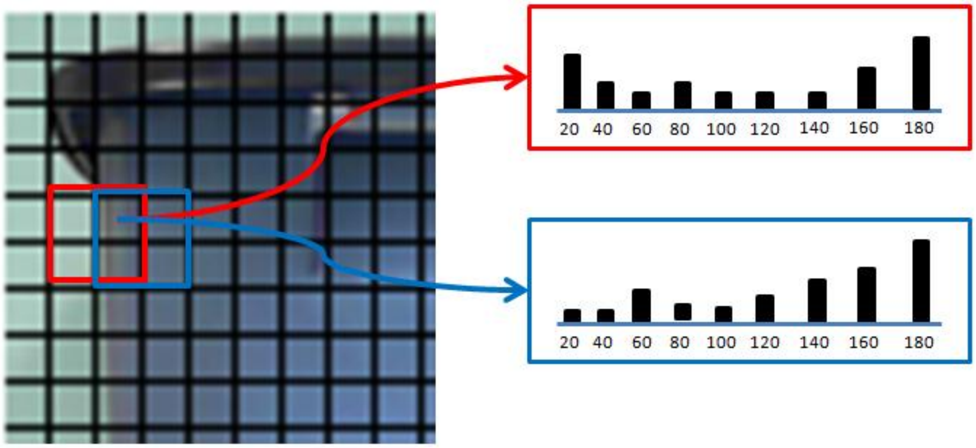


Figure 3.23: Block Normalized Histograms

The final feature vector used to describe the object is a concatenation of all of the orientation histograms. Figure 3.24 shows a visualization of the HOG descriptor, and illustrates a drawback of HOG: the HOG is larger than the original image. This increase in size is due to the fact that *cells* can contribute multiple orientations. The final feature vector size for the image below is approximately 20,000 dimensions. This is a very large feature vector, which means computational comparisons between two of them, whether it be machine learning algorithms or similarity/distance measures, is time consuming. However, since the HOG feature vector is very descriptive it will likely produce good matches, so often times rapid detection is traded in for accurate detection.

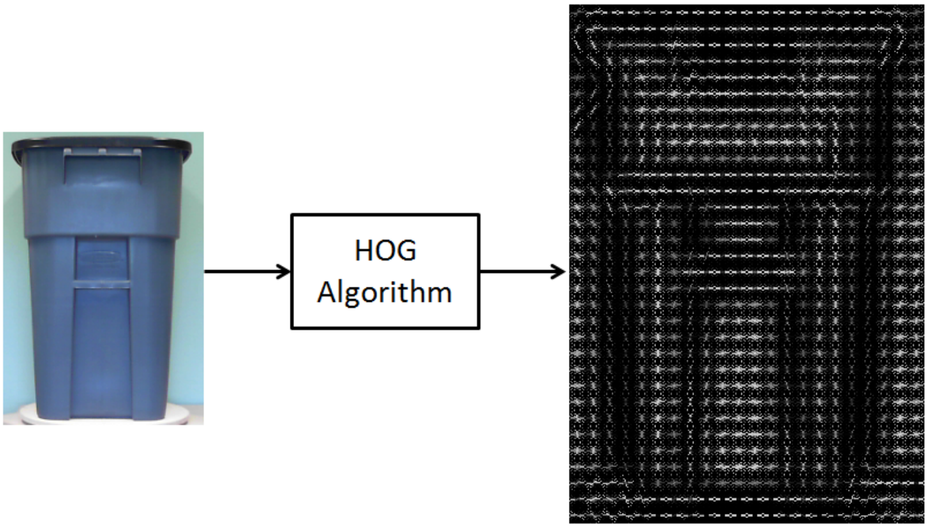


Figure 3.24: HOG Template

3.3 Frequency Domain Filtering to Improve Contour Detection

Edge, or contour, detection is an important and widely used technique in image processing and computer vision. It is usually an early stage processing technique used in many algorithms so the quality of detection at this stage can have an effect on the algorithm as a whole. Edge detection was used in this thesis with enhancements based on the work of Qu et al. [79].

3.3.1 Gradient Based Edge Detection

The most common way to detect edges is via gradients. Using gradients to detect edges means that a threshold was applied to the gradient magnitudes in order to determine the final contour image. This can lead to a problems because intense local changes, whether due to noise or background clutter, can produces edges within an image that are not a part of the object of interest, and cannot be removed via thresholding. Therefore, as a result many contour maps contain spurious edges that make further processing difficult for later stages of the algorithm pipeline. Figure 3.25 shows an edge map (the intensities were inverted for better visualization) containing the object of interest, the waste receptacle, in an outdoor scene. The unwanted edges in this scene would be the edges produced by the grass and foliage surrounding the receptacle.

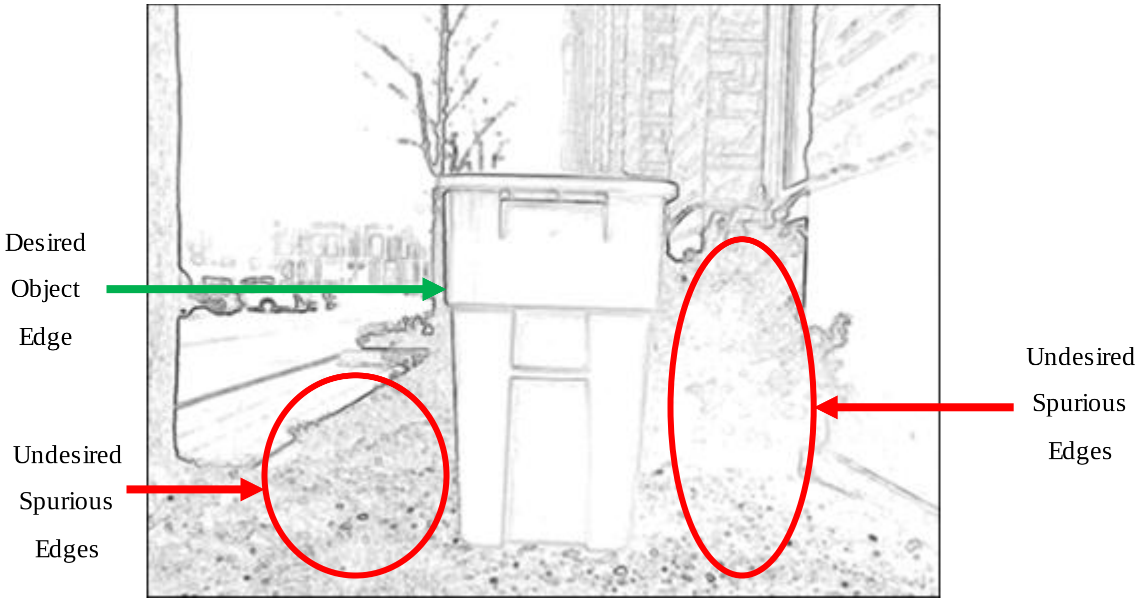


Figure 3.25: Edge Map of Receptacle in Outdoor Scene

3.3.2 Frequency Domain Analysis of Gradient Image

It is well known that spatial and frequency domain filtering are related and that if a kernel (aka a filter) is Fourier transferable then the process of filtering in the frequency domain is faster than its counterpart filtering in the spatial domain, when the kernel is large. With that being known filtering in the frequency domain is advantageous when processing time is a constraint. However, even though frequency domain filtering is advantageous it is sometimes difficult to determine the relationship between specific components within an image and their associated Fourier transform representation (i.e., the frequency spectrum image), which makes the choice of filter difficult.

In order to determine the relationship between the spatial components and the frequency components some simple analysis will be provided. First, it should be pointed out that a well known relationship between the spatial and frequency components is known to all those who study processing in both domains. That relationship is that if there are large and rapid intensity changes in the spatial image then these will correspond to high frequency components in the frequency image. The reverse case is true as well, if there are small and steady intensity changes in the spatial image then these will correspond to low frequency components. The discussion that follows shows some implications of this concept . Figure 3.26 shows the gradient magnitude image of a waste receptacle placed in an outdoor setting with a red profile line superimposed onto it. Figure 3.27 is a plot that shows the intensity values along this red profile line.

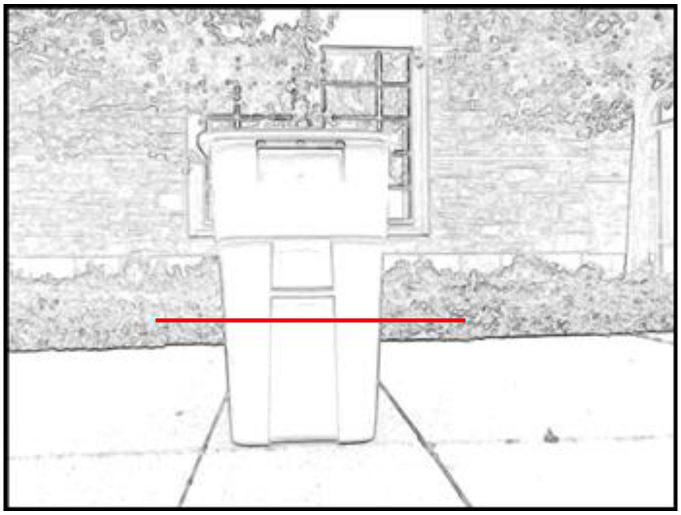


Figure 3.26: Profile Line Imposed on Gradient Magnitude Image

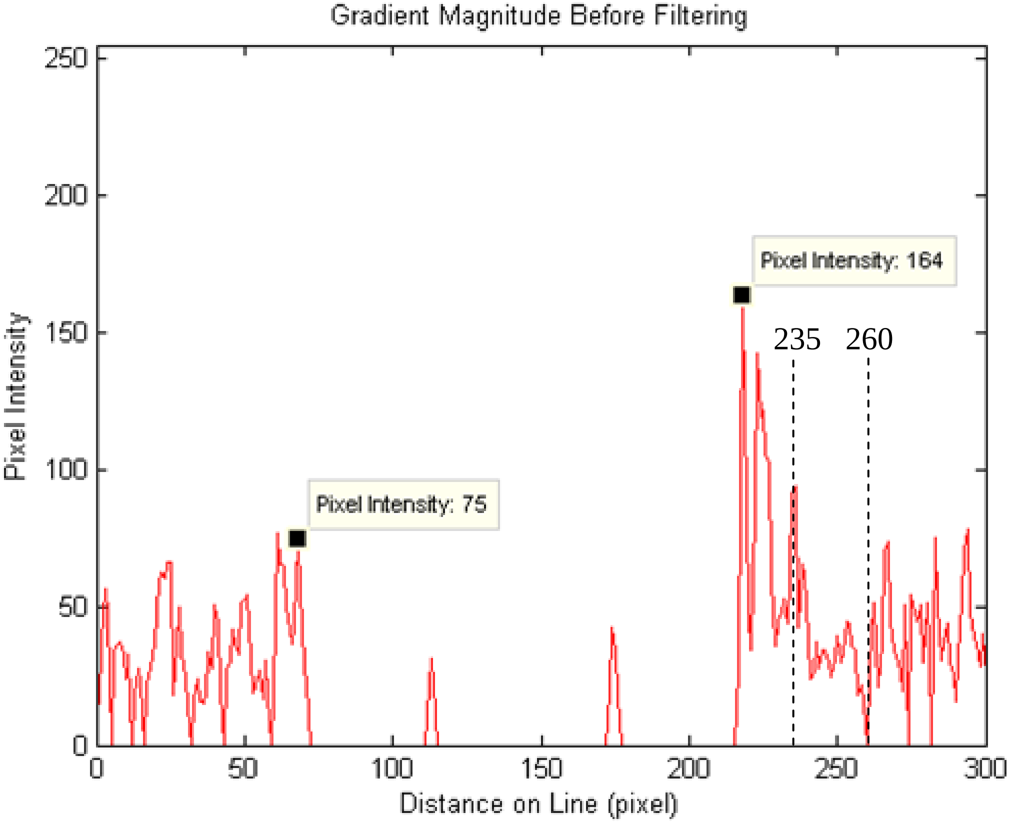


Figure 3.27: Profile Plot of Gradient Magnitude Image Before Frequency Filtering

A brief explanation of what is seen in the profile plot above will be given. There are two main points of reference in the plot above, which are marked by black squares. These points correspond to the edge points of the waste receptacle. The points between these two marked points are the pixels that correspond to the waste receptacle's surface. The surface has zero magnitude everywhere except for the two spikes which corresponds to where the receptacle indents. Every pixel outside of this area belongs to the background foliage in the image and must be attenuated by the choice of filter.

Recall the statement made earlier about how high and rapid intensity changes correspond to high frequencies, and low and stead intensity changes correspond to low frequencies. It is clear that the edges of the waste receptacle correspond to high frequency components because there is a large and rapid change of intensity located at the edge points. Using this logic it can also be concluded that there are some high frequency components present in the foliage. To focus the attention of the reader, and pull out specific examples, consider the area around pixel location 250. To the left of this location, at approximately pixel location 235, and to right of this location, at approximately pixel location 260, two high frequency components would exist due to the apparent rapid and large change in pixel intensity. However, the intensities between these two points can be consider low frequency components because, even though there is a change in intensity, the amount of change is minimal compared to the previously pointed out high frequency components. Therefore, it stands to reason that the best type of filter to eliminate the presence of the background foliage in the gradient magnitude map is a **high-pass filter**.

3.3.3 Applying a High-Pass Filter in the Frequency Domain

The convolution theorem and the concept of applying a filter in the frequency was discussed in the appendix section of this report and so the details will not be repeated here (refer to appendix section 7.1.3). The concluding idea of the previous section was that in order to reduce the presence of the foliage magnitudes a high-pass filter should be applied to the image. Figure 3.28 shows the gradient magnitude image before and after a high-pass filter was applied. In the after image the presences of the foliage magnitudes are greatly reduced. To see the direct effects the filter has on the pixel intensities the profile line is considered again. Figure 3.28 is a plot that shows the intensity values along the profile line in the filtered image.

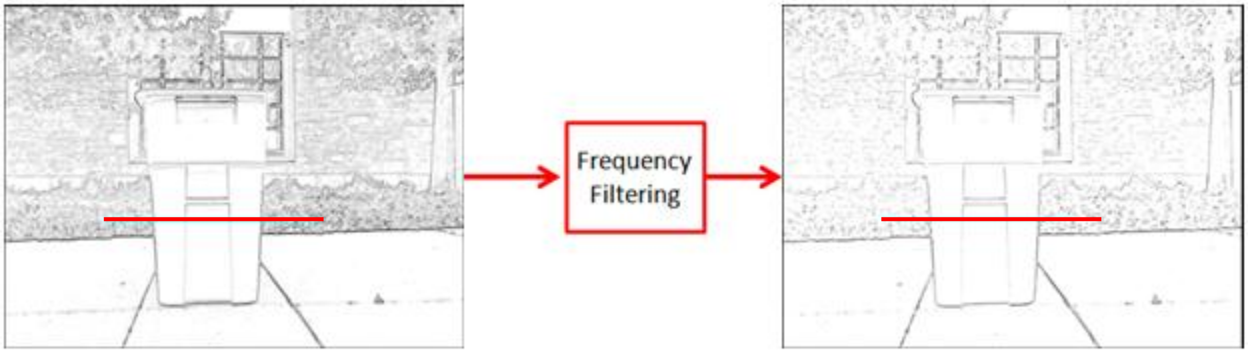


Figure 3.28: Gradient Magnitude Image Before and After Frequency Filtering

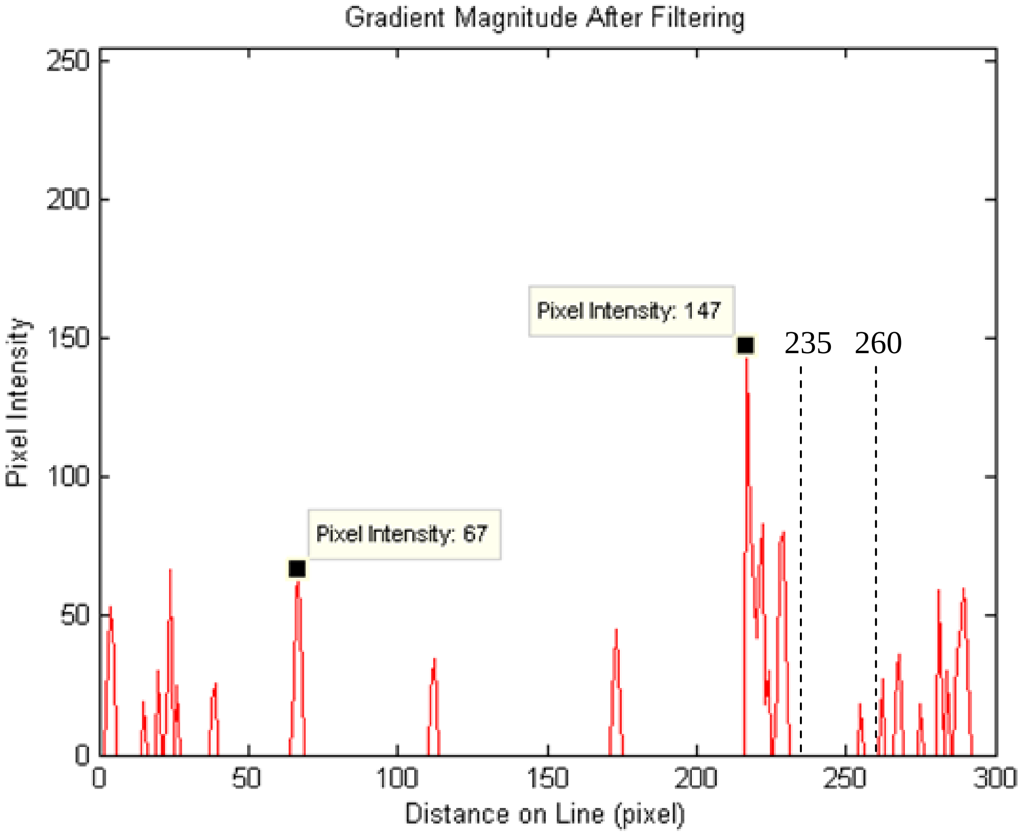


Figure 3.29: Profile of Gradient Magnitude Image After Frequency Filtering

If a comparison in made between the before and after profile plots two things about the filtering process can be concluded. First, it can be concluded that the filtering process was a success because many of the low frequency components were filtered out during the process and all that remains are the high frequency components. The second observation, which is a drawback to the success of the first observation, is that there seems to be edge degradation. This can be seen by comparing the intensity values of the edges points. In the plot before filtering the left and right edge points had pixel intensities 75 and 164, respectively, and after filtering they had pixel intensities 67 and 147, respectively. Therefore, it can be concluded that the process of applying a high-pass filter was a success, but that this success comes with the cost of lowering the intensity strength along desired edge components.

Chapter 4

Contributions

This chapter of the report will outline the contributions made by the author of this thesis. It should be noted that there are two stages to this project: offline database creation, and online object recognition, and that contributions were made to both of these stages. This chapter is organized as follows: a brief overview of the algorithm pipeline, individual summaries of the pipeline components and the contributions made within each, and finally an overview of database creation contributions. The contributions made by the author are listed below for ease of reference.

1. Developed the algorithm pipeline as a whole (**main contribution)**
2. Developed a point selection method during template creation
3. Used image processing techniques to make templates more robust
    a. Image averaging
    b. Edge Sharpening
4. Added contour enhancement to the GRM algorithm
5. Implemented multiple de-noising steps in GRM algorithm
6. Further quantized gradient response maps in order to reduce false positives

In this chapter certain illustrative examples are used as visual aids to show why certain processing steps have been included. The method as a whole is subjected to extensive real world testing in Chapter 5.

4.1 Overview of the Algorithm Pipeline

This section will present the algorithm pipeline as a whole. It should be noted that this pipeline is considered the main contribution of this thesis. Figure 4.1 provides a high level view of the steps involved in the algorithm used to solve the problem of finding a waste receptacle in a natural environment. Figure 4.1 shows the three main components of the algorithm: find pose candidate, verify candidate, and extract pose. In the proceeding sections each step will be explained along with the algorithms in which they employ and the motivation behind including them.

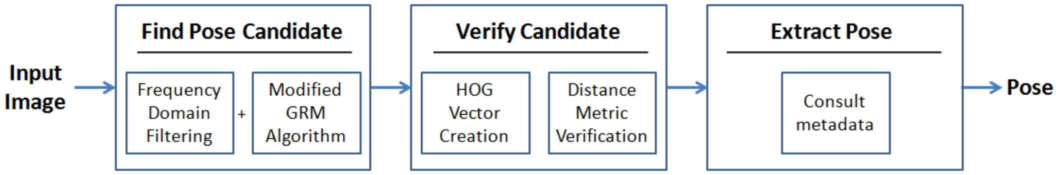


Figure 4.1: High Level View of Algorithm

4.2 Overview of Find Pose Candidate Step

The first step of the pipeline is to find an initial pose candidate within a test image. The purpose of this step is to quickly find where in the image the waste receptacle is most likely to exist and at what given pose. The algorithm employed to find the pose candidate is the GRM algorithm (more specifically the LINE-2D version). This version of the GRM algorithm is highly prone to false positives, so in order to reduce the amount of false positives the original GRM algorithm has been modified in several ways. It should be noted that even after the modifications this step is still prone to false positives and so is proceeded by a verification step, which will be explained in the next section.

The modifications made to the GRM algorithm include:

1. The addition of a contour filtering step
2. Applying noise suppression via polling twice
3. Quantizing the cosine response in the modified Steger metric

For a quick visual representation of the modifications made to the GRM algorithm a high level view of the GRM algorithm is shown below in Figure 4.2. The colour of the component blocks in Figure 4.2 indicate where a modification has occurred within the pipeline.

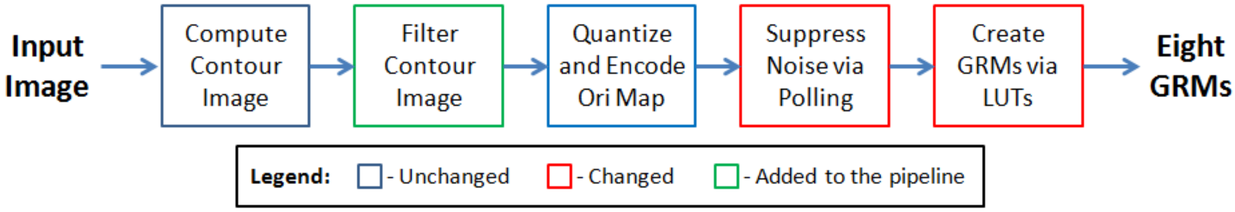
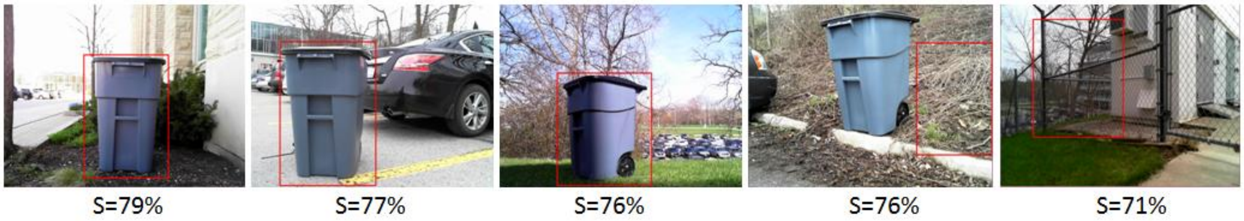


Figure 4.2: High Level View of GRM Modifications

These modifications will be justified and explained below in their own subsections. It should be noted that in order to justify and explain these modifications a few representative test pictures will be examined and the methodology used to evaluate the changes will be introduced as needed.

4.2.1 Rationale for Filtering the Contour Image

The methodology used to filter the contour image [79] was discussed in the both the literature review and background sections of this thesis report (section 2.3 and section 3.3) and so will not be discussed again here. This section will serve to illustrate why this step was added to the pipeline of the GRM algorithm. In order to justify this addition five test images will be used along with the evaluation metric introduced by this author. Consider the five test cases shown below in Figure 4.3. These images were passed through an unmodified GRM algorithm. The red bounding boxes show where the algorithm has indentified a waste receptacle and the score below each image indicates the similarity percentage.



S=79% S=77% S=76% S=76% S=71%

Figure 4.3: Pose Estimate using Unmodified GRM

The most straightforward way to evaluate this result is to plot the scores on a real number line. The points above the line are reserved for instances when the object is present and the points below the line are reserved for instances when the object is not present. A green point indicates that the instance was correctly handled and a red point indicates that the instance was incorrectly handled.

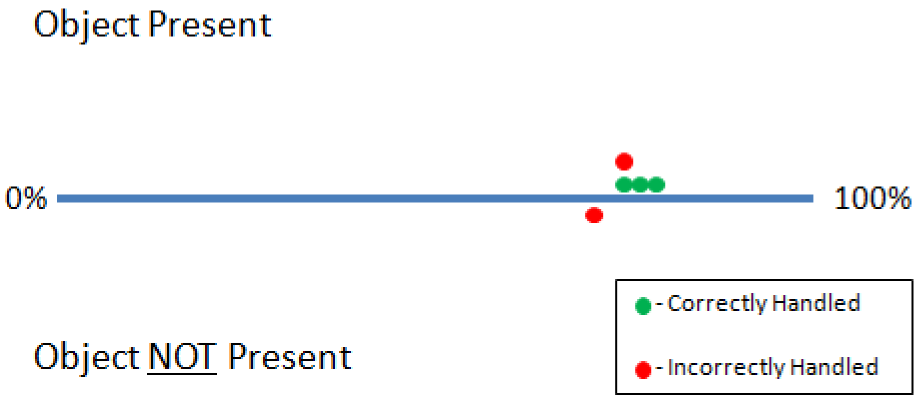


Figure 4.4: Real Number Line Evaluation of Unmodified GRM Scores

In this case two instances were incorrectly handled. One instance corresponds to a false positive when the waste receptacle is not present and the other instance corresponds to a false positive when the waste receptacle is present. In order to rectify the former case a threshold could be applied to the similarity score. That is a score below this threshold would not produce a valid hypothesis for the object. Figure 4.5 shows the real number line again but this time with a threshold being applied. The threshold now correctly processes the case where the waste receptacle in not present, however, the thresholding method contains some weakness.

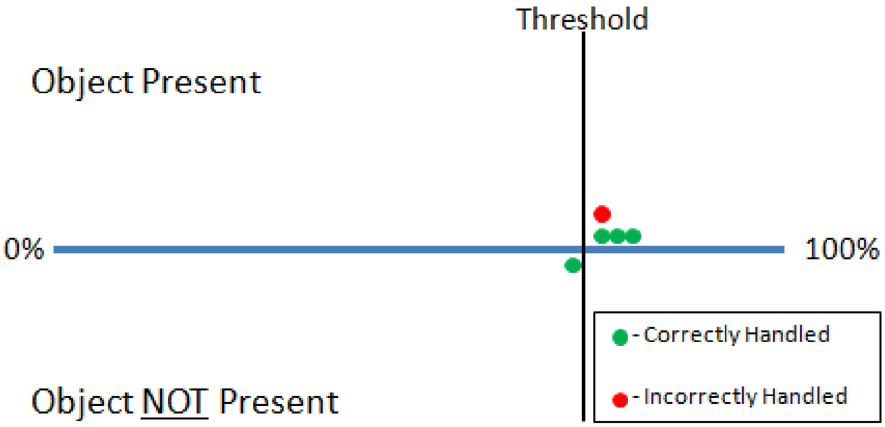


Figure 4.5: Real Number Line Evaluation of Unmodified GRM Scores After Thresholding

The problem with the threshold above is that the points around the threshold are too closely located. This is a problem because with the injection of noise these boundary point scores will change and this change could cause them to cross the threshold and be incorrectly handled Figure 4.6 illustrates this concern by zooming in on the points near the threshold and indicating how the scores of these point can change with the injection of noise.

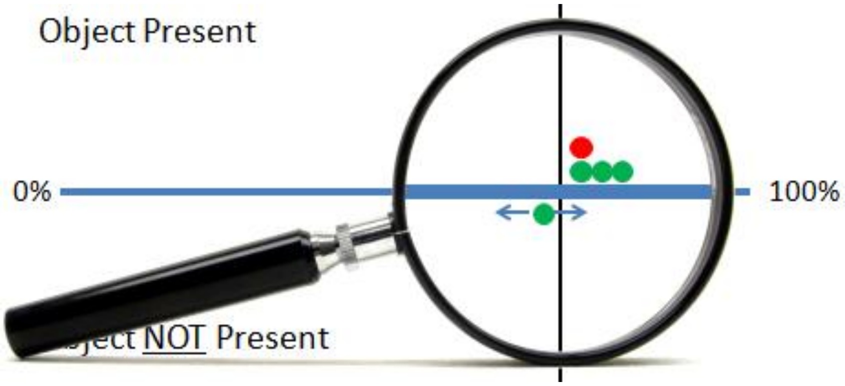


Figure 4.6: Noise Pushing Point Over Threshold

This threshold problem will be solved by the end, but for now it will be ignored and a solution to the latter case, where a false positive is produced even when the waste receptacle is present, will be provided. The issue with this latter case is that the surrounding scenery is producing too many unwanted edges and the algorithm is finding a better match to these edges then that of the waste receptacle. As discussed in the background section of this report, applying a frequency filter to the contour image can suppress the background edges. Figure 4.7 shows the result of applying the modified GRM to the same set of test pictures. That is, the result produced by the GRM algorithm with contour filtering added to the pipeline is shown below.

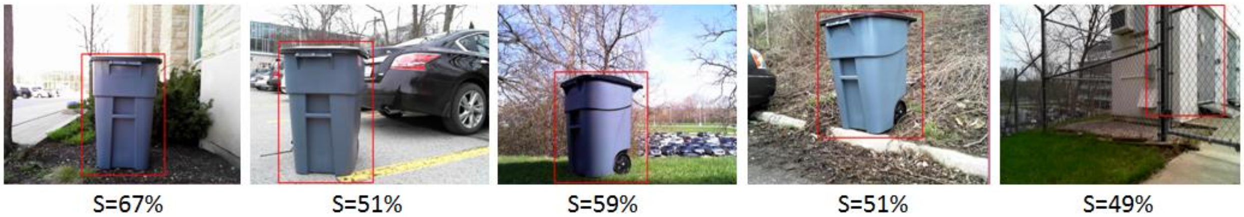


Figure 4.7: Pose Estimate After Adding Contour Filtering

After adding the contour filtering step to the pipeline all instances where the waste receptacle is present are now handled correctly. Again the number line is used in order to evaluate this situation. The scores prior to adding the contour filtering step are represented by the empty circles and the threshold was adjusted to handle the case when the waste receptacle is not present. (Note that the points are still clustered in a non-robust way). The scores have decreased due to edge degradation, as discussed in section 3.3, but all cases are now handled correctly.

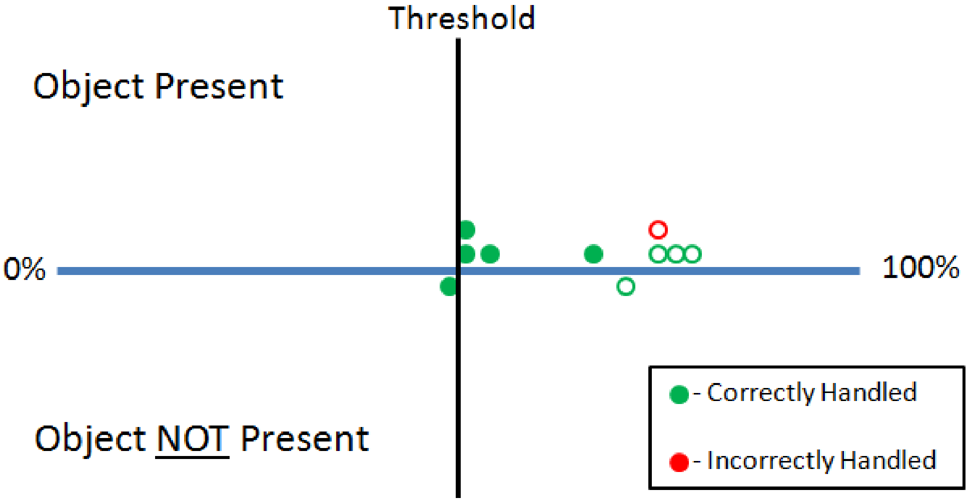


Figure 4.8: Real Number Line Evaluation After Adding Contour Filtering

4.2.2 Rationale for Applying Noise Suppression Twice

Recall from section 3.1 that one of the steps involved in the GRM algorithm is to apply noise suppression via polling. In the original paper [48] this process was applied once to the image, however, in this thesis it was applied twice in order to further suppress noise and reduce the false positive rate.

The following test image, Figure 4.9, will be used to illustrate how applying the polling step twice, instead of once, can reduce the false positive rate. The methods used to evaluate this claim will be based on empirical data. The first evaluation method will compare the algorithm output result for polling once and polling twice using the entire database of templates. The second evaluation method will use 1D and 2D similarity score plots to show that the similarity scores have better separation (i.e., more distinct) when polling is applied twice. For the second evaluation method the database is limited to the correct template, that is only the correct answer is used to produce the 1D and 2D plot scores.

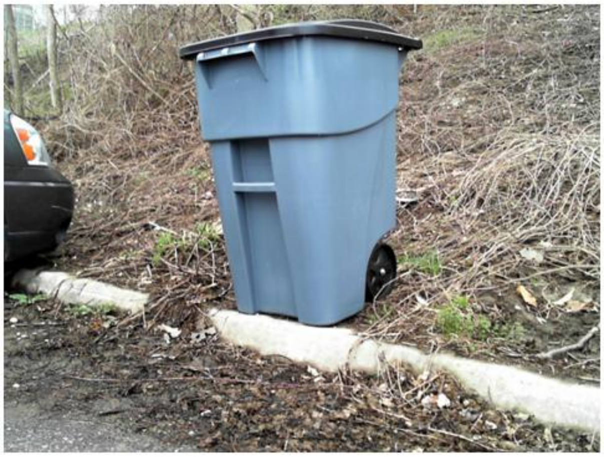


Figure 4.9: Test Image used for Polling Argument

The first evaluation method will compare the outputs produced for polling once and polling twice. It should be noted that at this point in time contour filtering and cosine linearization (discussed later) have been added to the pipeline in both cases, and that the entire database of templates was used in the process of finding a match. The results for polling once and polling twice are shown in Figure 4.10 and Figure 4.11, respectively. As you can see the result is correct when polling is applied twice and incorrect for when polling is applied once. Therefore, just from this test polling twice could be considered advantageous. However, the 1D and 2D plots will better exemplify the inclusion of this second polling iteration.

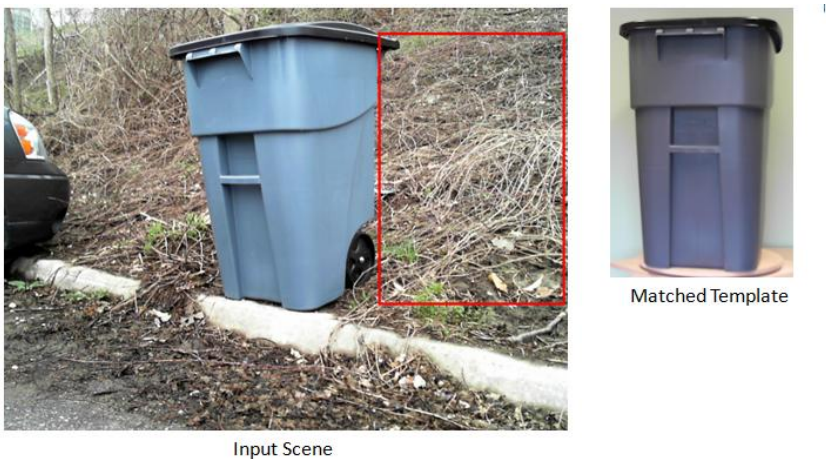


Figure 4.10: Polling Once

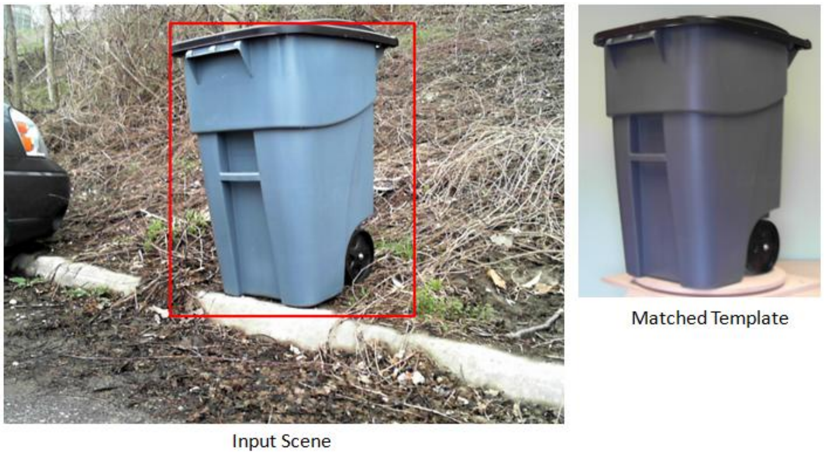


Figure 4.11: Polling Twice

The second evaluation method will use 1D and 2D similarity score plots to show that the similarity scores have better separation (i.e., more distinct) when polling is applied twice. For this evaluation method the database will be limited to the correct template, that is only the correct template will be used to parse the scene and produce scores. The reason this restriction is in place is because when polling is only applied once the wrong template and location are selected. Therefore, this restriction ensures that a direct comparison can be made between the sets of plots.

Figure 4.12 shows polling once being applied and Figure 4.13 shows polling twice being applied. In both plots the maximum score is tagged and these scores corresponds to the correct location. It should be noted that in the 2D plots the colour bar ranges were set automatically with respect to the range of values being plotted. This keeps the data points within each plot relative to each other and allows for better visualization of value separation. It can be seen that when polling twice is applied there is a better separation between the maximum response and all other responses within the 2D plot. That is, in the polling twice plot there are fewer peaks that are comparable to the maximum response. This means the correct answer is more distinct when polling is applied twice.

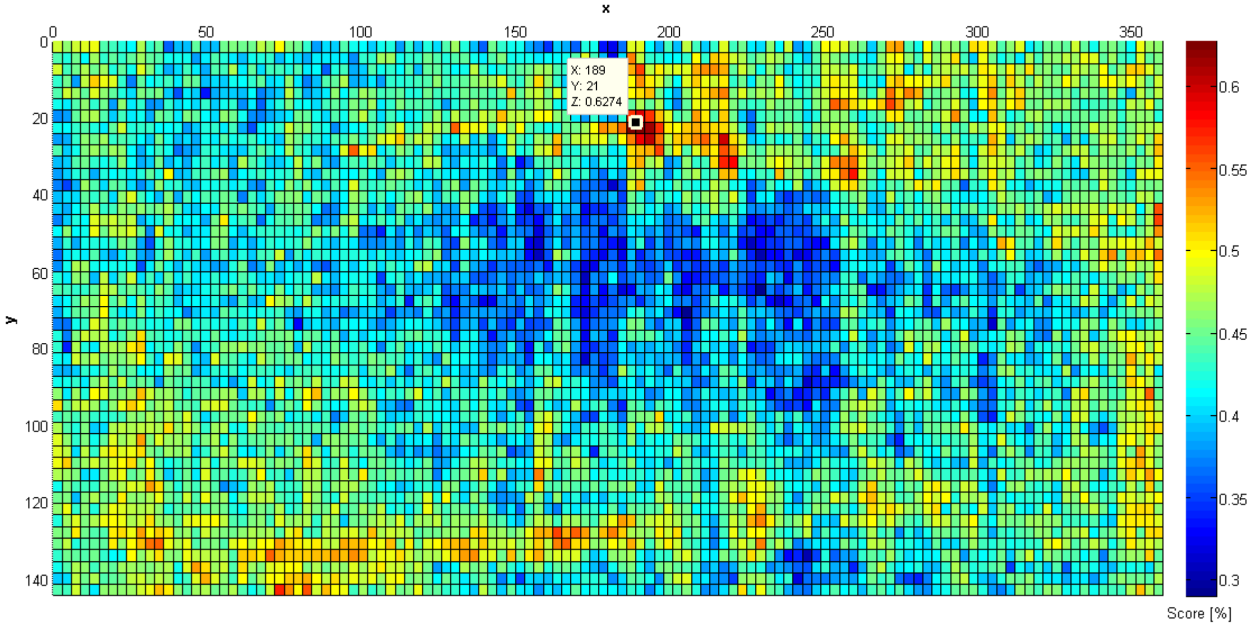


Figure 4.12: 2D Plot of Similarity Scores when Polling is Applied Once

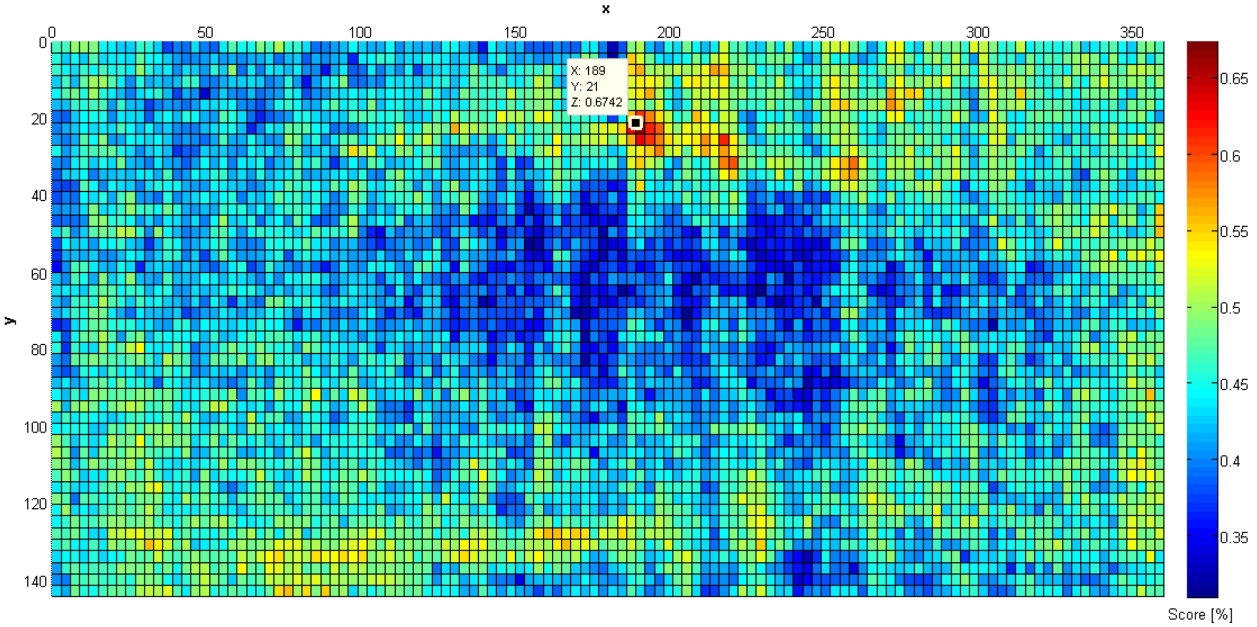


Figure 4.13: 2D Plot of Similarity Scores when Polling is Applied Twice

A similar inference can be drawn from the 1D plots of the similarity scores along the correct rows (i.e., the rows that contains the object), as shown in Figure 4.14 and Figure 4.15. When polling is applied twice there is a 0.1088 point difference between the maximum response and the next closest peak (see Figure 4.15) and when the when polling once is applied there is a 0.0766 point difference between the maximum response and the next closest peak (see Figure 4.14). Therefore an inference from the 1D and 2D plots can be made that polling twice produces scores with better separation, and that better separation should lead to less false positives.

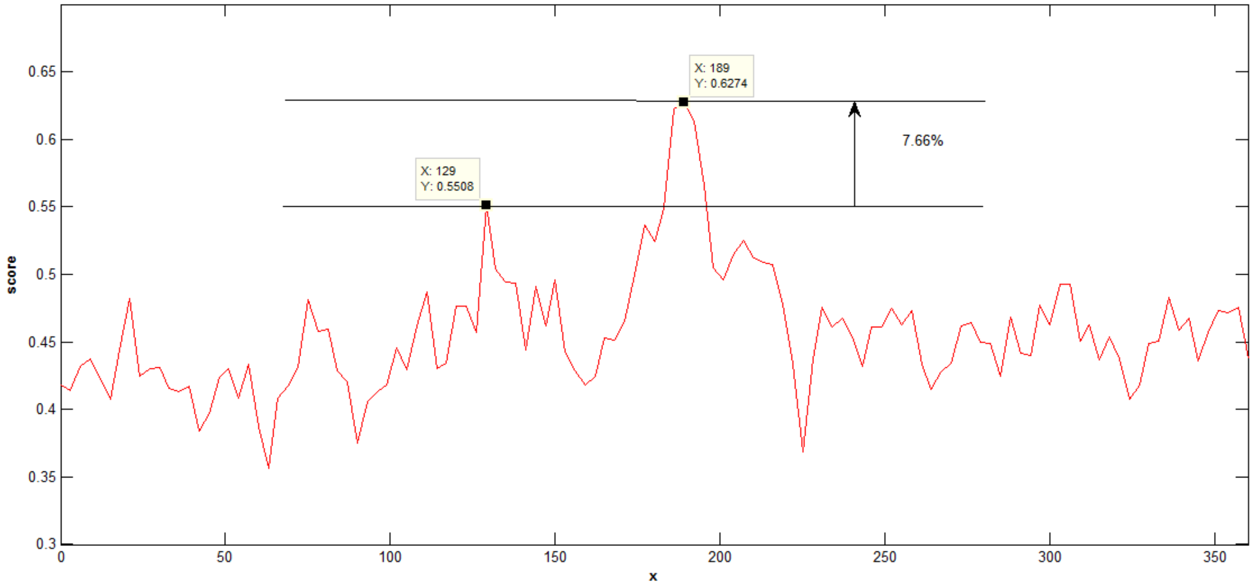


Figure 4.14: 1D Plot of Similarity Scores when Polling is Applied Once

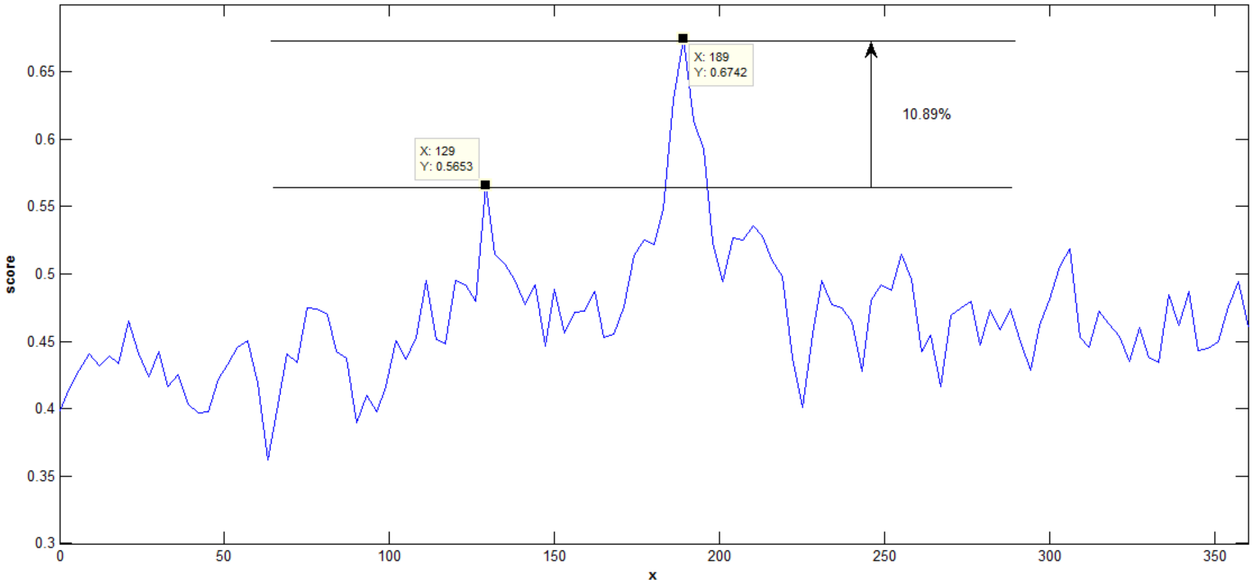
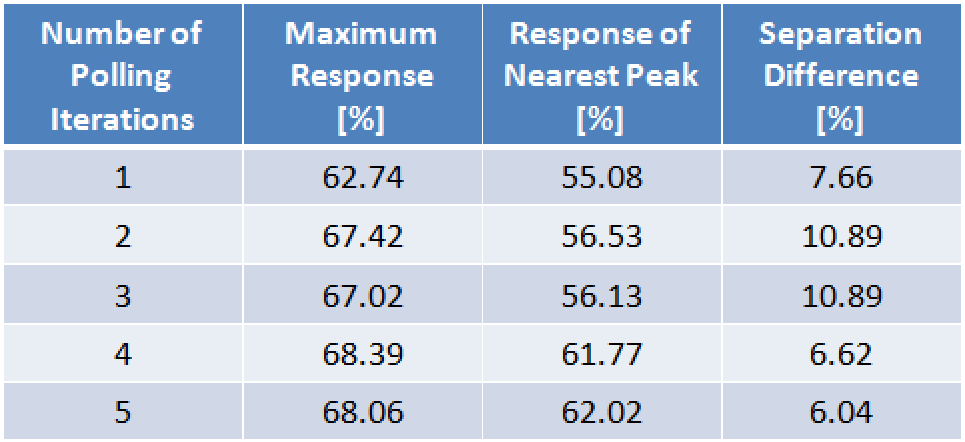


Figure 4.15: 1D Plot of Similarity Scores when Polling is Applied Twice

More than two polling steps is not expected to add to the performance. A table that consists of the separation scores between the maximum response and the nearest peak for polling iterations one through five is shown in Figure 4.16. The third iteration does not change the difference between the max response and the its nearest neighbor, at least for this particular case, and only adds to the computational cost. For iterations four and five the difference between the maximum response the their nearest neighbor begins to shrink. This makes sense because as more and more iterations occur the image becomes more homogenous and eventual converges to stationary state where only one or just a few orientations are left remaining.



Number of Polling Iterations	Maximum Response [%]	Response of Nearest Peak [%]	Separation Difference [%]
1	62.74	55.08	7.66
2	67.42	56.53	10.89
3	67.02	56.13	10.89
4	68.39	61.77	6.62
5	68.06	62.02	6.04

Figure 4.16: Difference Between Responses Based on Number of Polling Iterations

For the sake of brevity only one test image was used to illustrate this point, however, in appendix section 7.2 additional test pictures are used that undergo the same evaluation structure used in this section. The tests shown in the appendix convey the same message as presented here and are included in this report for the sake of completeness and to provide some statistical significance to the arguments made above.

4.2.3 Rationale for Linearizing the Cosine Response in the Modified Steger Metric

Recall from the background section (section 3.1) that the authors of GRM [48] modified the Steger metric [47] so that it would be robust against small deformations and translations by considering a local neighborhood around each point. In addition to this modification they employed orientation spreading and gradient response maps (GRMs) to make the metric more computationally efficient. Also, recall that one of the intermediate steps involved in creating these GRMs was to consult a look up table (LUT). In this thesis the Steger metric was modified again to reduce the number of false positives by creating similarity scores with better separation between the maximum response and those around it. This was accomplished by the linearization of the cosine output before making the LUTs. For the sake of clarity a figure from the background section has been reproduced here so the reader does not have to consult two sections at once to remember the process for GRM construction. Figure 4.17 shows the construction of two GRMs using LUTs.

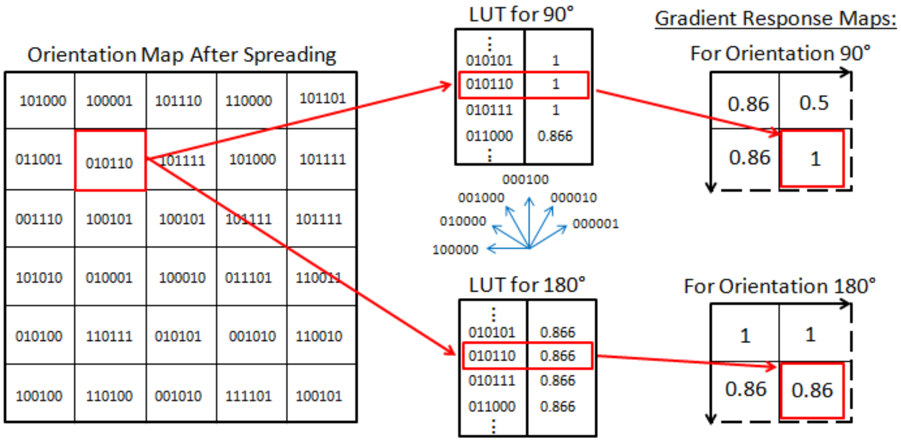
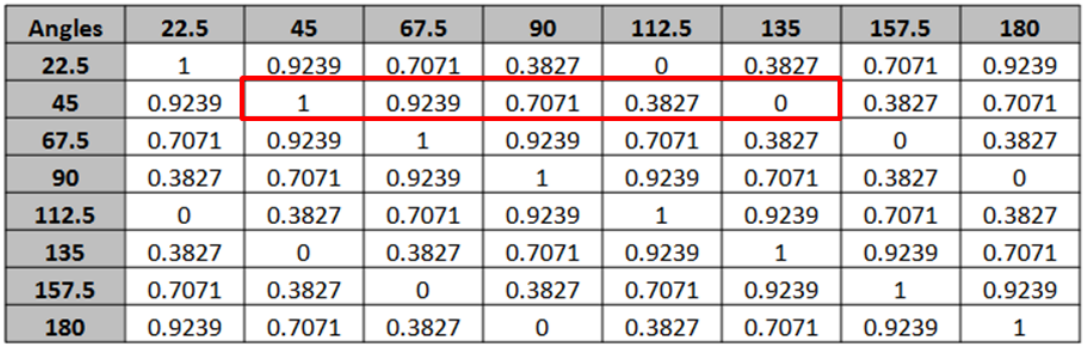


Figure 4.17: GRM Construction using LUTs

In this thesis the values in the LUTs are made to be linear across the range [0,1] instead of using the result of the cosine calculation directly. To explain this clearly consider Figure 4.18 which shows all the possible cosine responses for the orientations present within the orientation map. The red box highlights all five possible responses that could be produced.



Angles	22.5	45	67.5	90	112.5	135	157.5	180
22.5	1	0.9239	0.7071	0.3827	0	0.3827	0.7071	0.9239
45	0.9239	1	0.9239	0.7071	0.3827	0	0.3827	0.7071
67.5	0.7071	0.9239	1	0.9239	0.7071	0.3827	0	0.3827
90	0.3827	0.7071	0.9239	1	0.9239	0.7071	0.3827	0
112.5	0	0.3827	0.7071	0.9239	1	0.9239	0.7071	0.3827
135	0.3827	0	0.3827	0.7071	0.9239	1	0.9239	0.7071
157.5	0.7071	0.3827	0	0.3827	0.7071	0.9239	1	0.9239
180	0.9239	0.7071	0.3827	0	0.3827	0.7071	0.9239	1

Figure 4.18: All Possible Cosine Responses Before Linearization

These responses are a result of the cosine equation used in the modified Steger metric. If these responses are made linear so that they are evenly distributed across the range of [0,1] then the false positive rate is reduced because the entire dynamic range can be used and the score produced have better separation. For the sake of completeness the derivation of this change will be represented by the equations below.

Consider the modified Steger metric:

$$E(I,T,c) = \sum_{r\epsilon p}\left(\max_{t\epsilon R(c+r)} |\cos(ori(T,r) - ori(I,t))|\right)$$

With the linearization of the inner cosine calculation, the equation becomes:

$$E(I,T,c) = \sum_{r\epsilon p}(D(I,T,c))$$

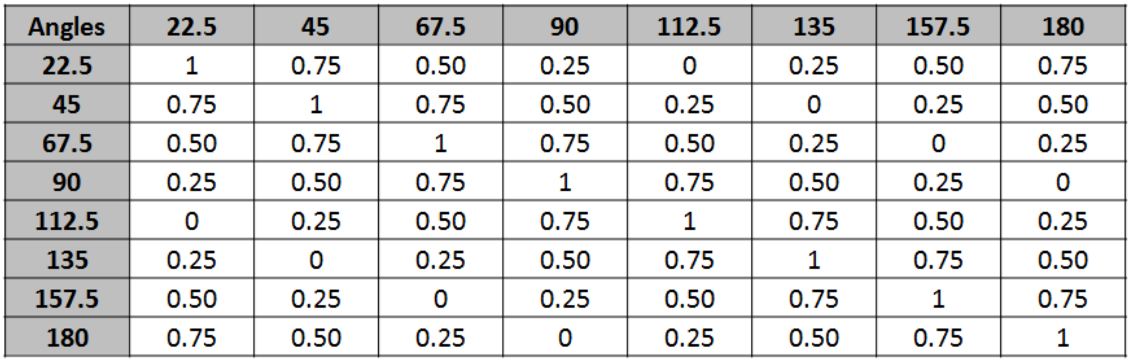
Where,

$$D(I,T,c) = \begin{cases} 1\ , & x\ = 1 \\ 0.75, & x\ = 0.9239 \\ 0.50, & x\ = 0.7070 \\ 0.25, & x\ = 0.3827 \\ 0\ , & x\ = 0 \end{cases}$$

and,

$$x\ =\ \max_{t\epsilon R(c+r)} |\cos(ori(T,r) - ori(I,t))|$$

The result of this modification produces the cosine response table in Figure 4.19. These values are now used in the LUTs to produce the GRMs.



Angles	22.5	45	67.5	90	112.5	135	157.5	180
22.5	1	0.75	0.50	0.25	0	0.25	0.50	0.75
45	0.75	1	0.75	0.50	0.25	0	0.25	0.50
67.5	0.50	0.75	1	0.75	0.50	0.25	0	0.25
90	0.25	0.50	0.75	1	0.75	0.50	0.25	0
112.5	0	0.25	0.50	0.75	1	0.75	0.50	0.25
135	0.25	0	0.25	0.50	0.75	1	0.75	0.50
157.5	0.50	0.25	0	0.25	0.50	0.75	1	0.75
180	0.75	0.50	0.25	0	0.25	0.50	0.75	1

Figure 4.19: All Possible Cosine Responses After Linearization

In order to evaluate the effectiveness of this modification the same types of evaluation methods used in the previous section will be used in this section. That is, using Figure 4.9 as a test image the first evaluation method will compare the algorithm output results when linearization is used and not used. And the second evaluation method will use 1D and 2D similarity score plots to show that the scores have better separation (i.e., more distinct) when linearization is applied.

The first evaluation method will compare the output hypotheses produced when linearization is used and not used. It should be noted that, at this point in time, contour filtering and polling twice have been added to the pipeline in both cases, and that the entire database of templates was used in the process of finding a match. The hypothesis produced when linearization was not applied is shown in Figure 4.20, and the hypothesis produced for when linearization is applied is shown in Figure 4.21. The hypothesis is correct when linearization is applied and incorrect for when linearization is not applied.

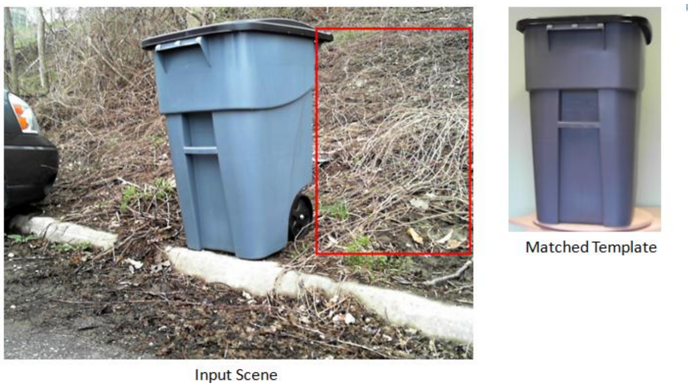


Figure 4.20: Hypothesis without Linearization

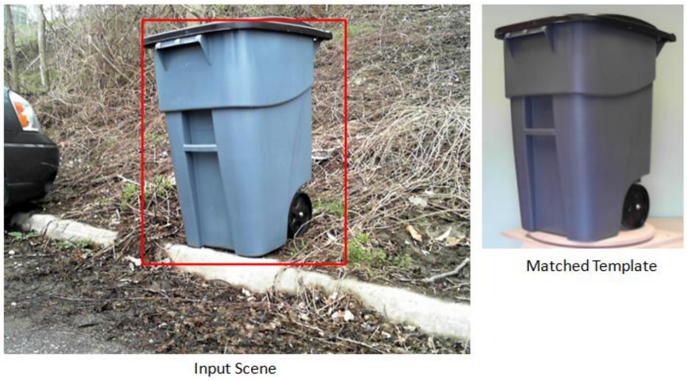


Figure 4.21: Hypothesis with Linearization

The second evaluation method will use 1D and 2D similarity score plots to show that the similarity scores have better separation (i.e., more distinct) when linearization is applied to the cosine responses. For this evaluation method the database will be limited to the correct template, that is only the correct template will be used to parse the scene and produce scores. The reason this restriction is in place is because when linearization is not applied the wrong template and location are selected. Therefore, this restriction ensures that a direct comparison can be made.

Figure 4.22 and Figure 4.23 show the 2D similarity score plots when linearization is not applied and when linearization is applied, respectively. It can be seen that when linearization is applied to the cosine responses there is a better separation between the maximum response and all other responses within the 2D plot. This means that in the linear version a more distinct maximum score is produced. That is with respect to the global maximum there are fewer comparable peaks in the plot with linearization then there are in the plot without linearization.

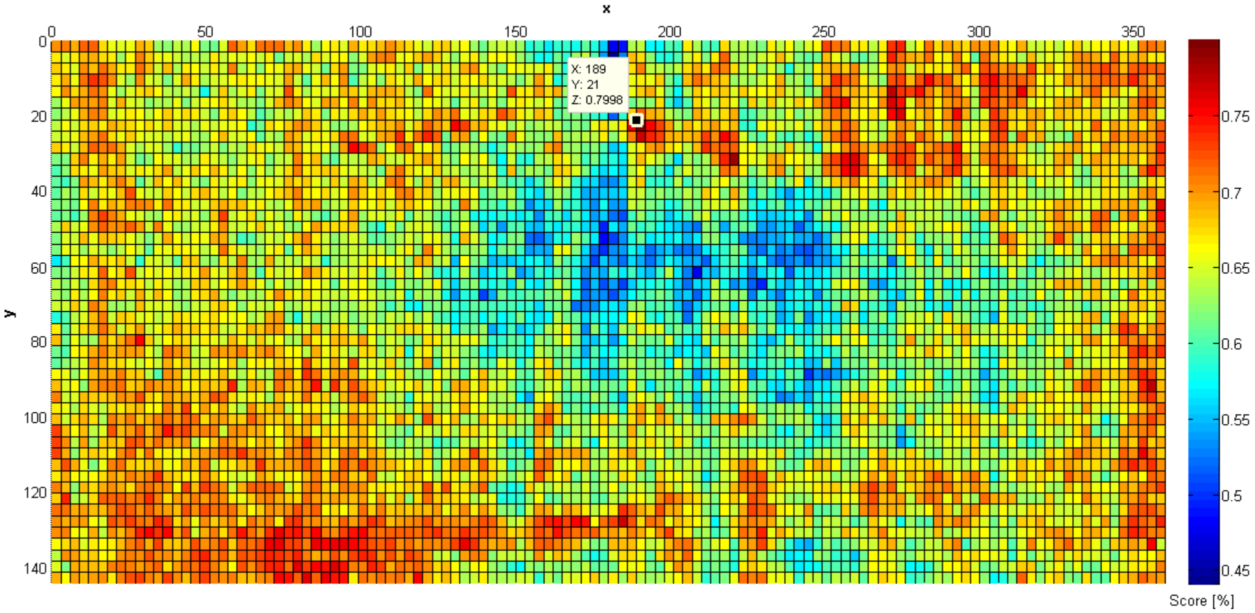


Figure 4.22: 2D Plot without Linearization

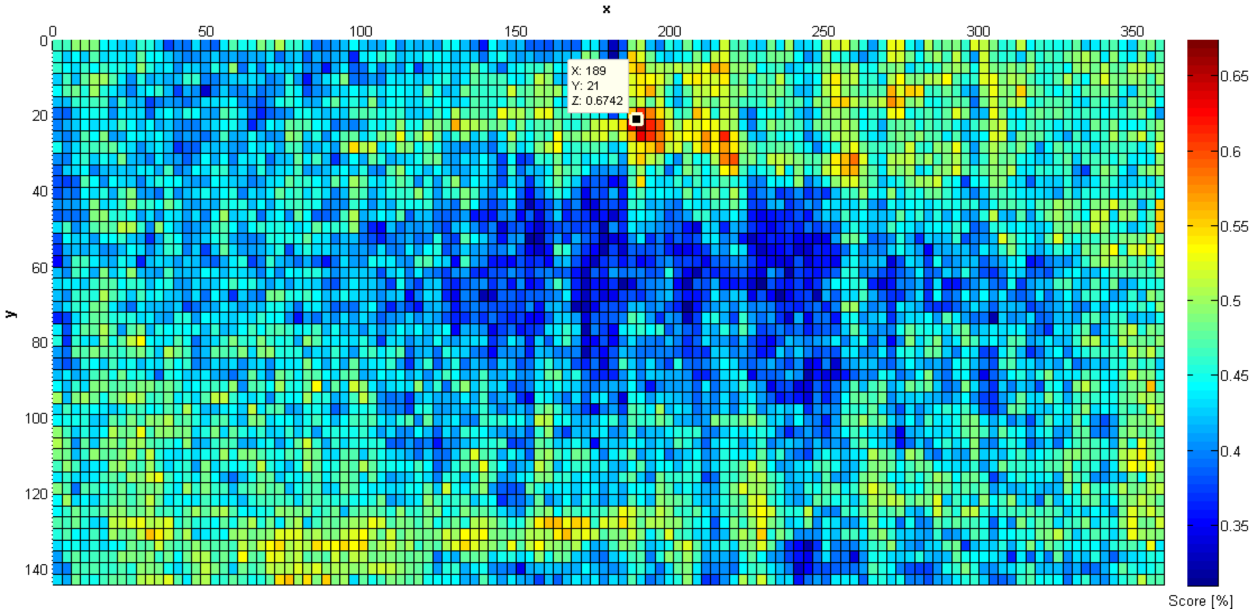


Figure 4.23: 2D Plot with Linearization

Figure 4.24 and Figure 4.25 show the 1D plots of the similarity scores along the correct rows (i.e., the rows that contains the object). When linearization is applied there is a 0.1088 point difference between the maximum response and its nearest neighbor (see Figure 4.25), and when linearization is not applied there is only a 0.0467 point difference between the maximum response and its nearest neighbor (see Figure 4.24). Therefore it can be inferred from the 1D and

2D plots that adding linearization produces scores with better separation, which in turn reduces false positives.

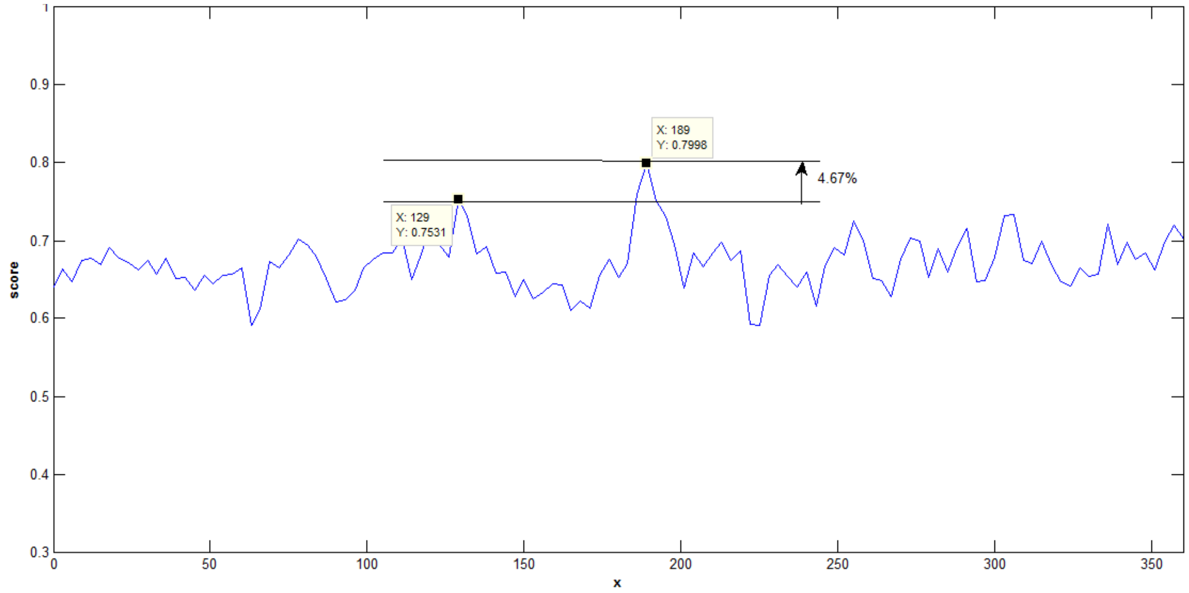


Figure 4.24: 1D Plot without Linearization

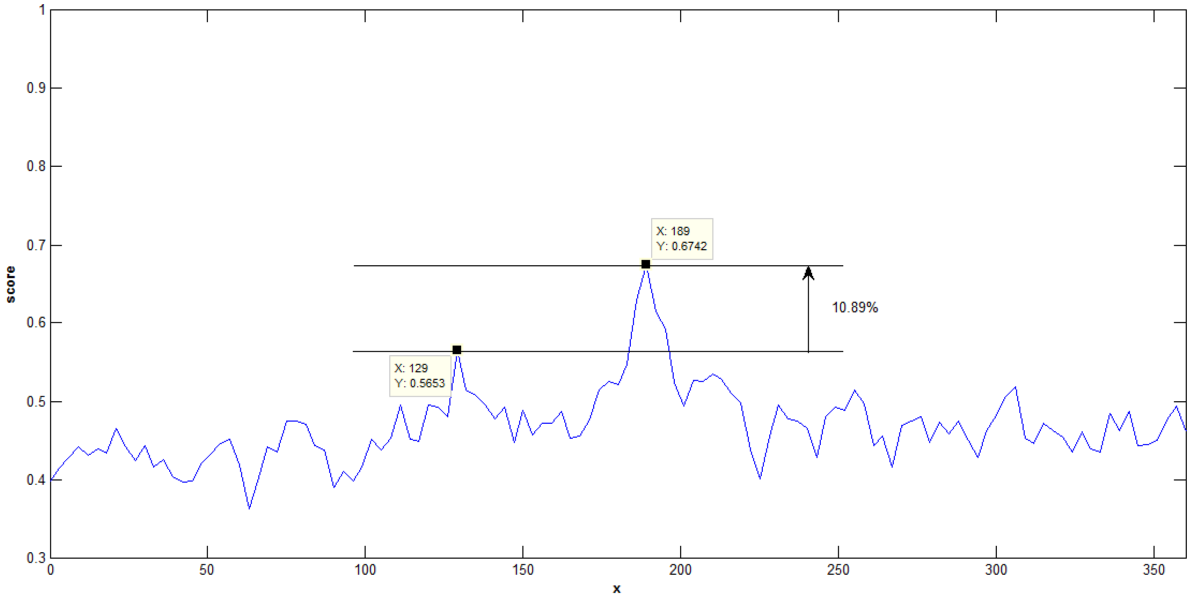


Figure 4.25: 1D Plot with Linearization

For the sake of brevity only one test image was used to illustrate this point, however, in appendix section 7.2 additional test pictures are used that undergo the same evaluation structure used in this section. The tests shown in the appendix convey the same message as presented here but are included in this report for the sake of completeness and to provide some statistical significance to the arguments made above.

4.3 Overview of Verify Pose Candidate Step

The second step in the pipeline is to verify the pose candidate produced in step one. The motivation of this step can be explained using the real number line metric introduced earlier. Figure 4.26 shows the real number line evaluation metric for the modified GRM algorithm. The points closest to the threshold are so close that they are at risk of crossing over with the injection of noise. The goal of step two is to widen this area.

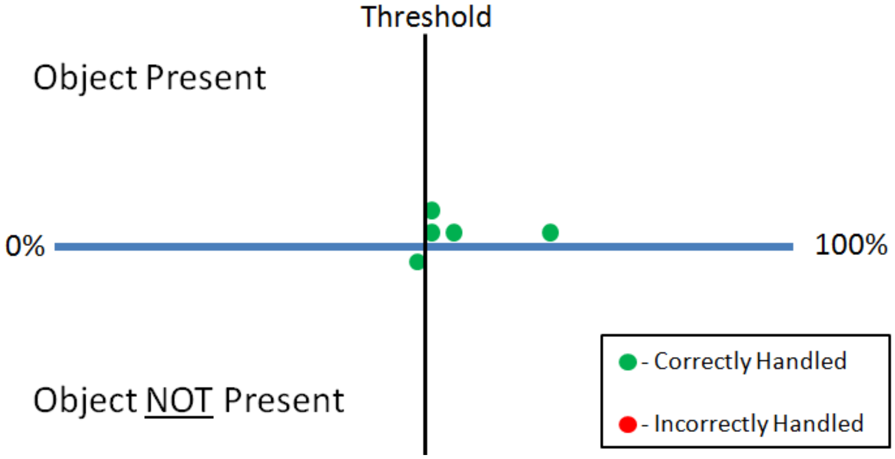
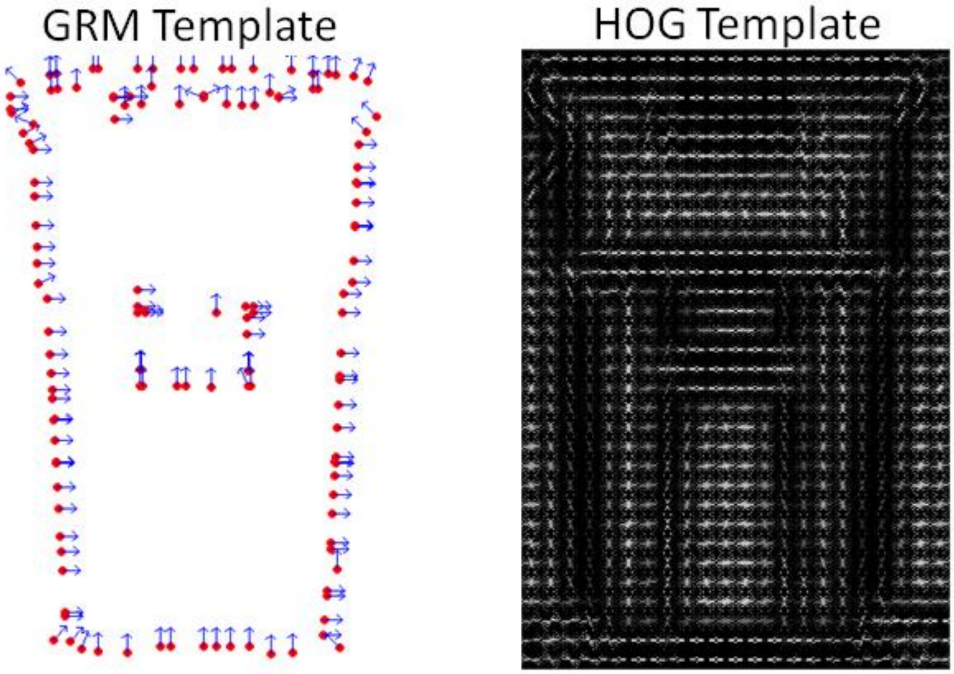


Figure 4.26: Real Number Line Evaluation For Modified GRM Algorithm

The reason the points are so close to one another is because the templates used in the GRM algorithm are not highly descriptive. This is of course one of the reason the algorithm is computationally efficient, but it is also the reason that it suffers from several false positives. Since the problem lies in the fact that the templates are not highly descriptive the answer proposed here to solve the problem would be to verify the pose candidate with a template that is highly descriptive (i.e., the HOG).

This approach strikes a balance between a computationally inefficient, but highly descriptive, method and a coarser, faster, less descriptive, method. First GRM is used to find a candidate pose, then the more complex HOG template is used to verify that pose. HOG was selected to verify the pose candidate because it is highly descriptive and robust to illumination conditions (refer to section 3.2 for a review of HOG). Figure 4.27 compares the relative complexity of the two templates. In this figure the GRM template produces a feature vector with approximately 130 elements and the HOG template produce a feature vector with approximately 20,000 elements.



GRM Template

HOG Template

Figure 4.27: GRM and HOG Templates

Before evaluating the addition of this verification step a brief description of the metric used to compare the HOG of the object and the HOG of the candidate ROI will be given. Normally when HOG is used to parse a scene the type of evaluation metric used is a machine learning algorithm, like SVM or KNN. Machine learning algorithms are normally used because HOG is usually applied to recognize a category of objects like bikes, cars, and people. Therefore, the additional computational overhead of these machine learning algorithms is needed to handle the varying features across a category of objects. However, in this project HOG is not being used to recognize a category but is instead being used to recognize specific objects, making a learning step superfluous. Therefore, instead of the machine learning approach a distance metric was used in substitution. More specifically the L2 norm was used to compare the two HOG vectors.

In order to make the distance between all templates and ROIs comparable the HOG vectors are first normalized before being applied to the L2 norm equation. This ensures that the distance returned by the L2 norm is always between [0,2] and a single threshold can be used for all template and ROI comparisons. However, it should be noted that the implementation of the HOG in this thesis will always constrain the vector elements to be greater than, or equal, to zero.

Therefore, after normalization the L2 norm can only return a distance value between $[0,\sqrt{2}]$. The set of equations below will formally describe the process of comparing the two vectors and determining if the object is present, or not present.

First consider two HOG vectors of size n:

$$T_{hog} = [x_1, x_2, \dots, x_n], \qquad x_i \geq 0 \ \forall \ \{i | i = [1, n]\}$$

$$R_{hog} = [y_1, y_2, \dots, y_n], \qquad y_i \geq 0 \ \forall \ \{i | i = [1, n]\}$$

Where T_{hog} is the template HOG and R_{hog} is the ROI HOG. Now consider the normalized version of these vectors:

$$\overline{T_{hog}} = \left[\frac{x_1}{|T_{hog}|}, \frac{x_2}{|T_{hog}|}, \dots, \frac{x_n}{|T_{hog}|}\right], \qquad |T_{hog}| = \sqrt{\sum_{i=1}^{n} x_i^{\,2}}, \qquad |\overline{T_{hog}}\,| = 1$$

$$\overline{R_{hog}} = \left[\frac{y_1}{|R_{hog}|}, \frac{y_2}{|R_{hog}|}, \dots, \frac{y_n}{|R_{hog}|}\right], \qquad |R_{hog}| = \sqrt{\sum_{i=1}^{n} y_i^{\,2}}, \qquad |\overline{R_{hog}}\,| = 1$$

After the two vectors are normalized they can be applied to the L2 Norm to produce a distance measure between $[0,\sqrt{2}]$.

$$\|d\| = \sqrt{\sum_{i=1}^{n} \left(\frac{x_i}{|T_{hog}|} - \frac{y_i}{|R_{hog}|}\right)^2}$$

Finally, the distance measure $\|d\|$ will be compared to a threshold to check if the object is present or not. A value of one indicates the object is present and zero indicates the object is not present. For this thesis the threshold value is chosen to be half way between the closest points on both sides. In the evaluation section of this report a final threshold value is selected after 1007 test images have been evaluated.

$$present = \begin{cases} 1, & \|d\| < threshold \\ 0, & \|d\| \geq threshold \end{cases}$$

In a less formal manner the object verification step can be represented by Figure 4.28, which shows the vector points of the template HOG and the ROI HOG. In Figure 4.28, the image on the left shows that the distance between the two HOG vectors is within the threshold and so the pose is accepted, and the image on the right shows that the distance is greater than the threshold and so the pose is rejected.

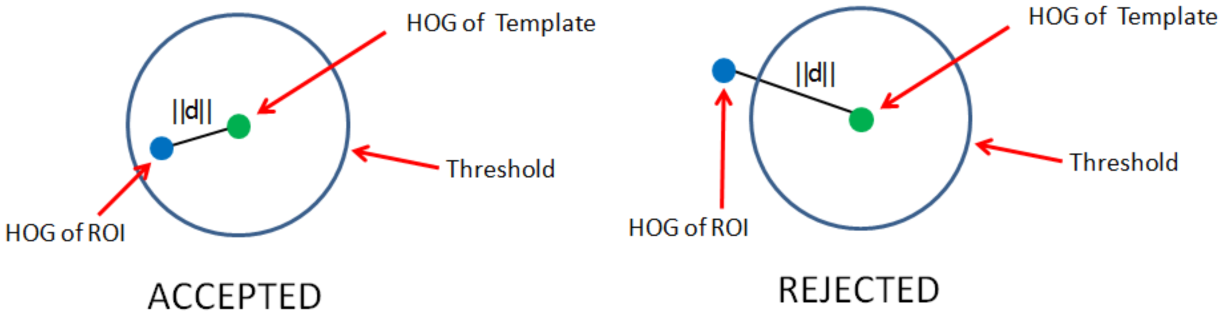


Figure 4.28: Distance Metric Representation of HOG Vectors

In order to evaluate the addition of this step the real number line evaluation method will be used. It should be noted that since the metric being used is a distance metric, and not a similarity metric, a low score is better. That is, if two vectors are identical then their distance would be zero and if two vectors are orthogonal (which is as far apart as they could get for this implementation of HOG) then the distance would be $\sqrt{2}$. The set of test images in this evaluation will include the original five images used in Section 4.2.1 but will also include five new test images. Figure 4.29 shows the set of test images after they have been passed through the modified GRM algorithm and the HOG verification step. These two steps are able to correctly locate the waste receptacle when it is present and correctly reject any hypothesis when the waste receptacle is not present. Figure 4.30 shows the distance values plotted on a real number line. There is now a wide enough spacing between the points and the threshold to accommodate noise injection.

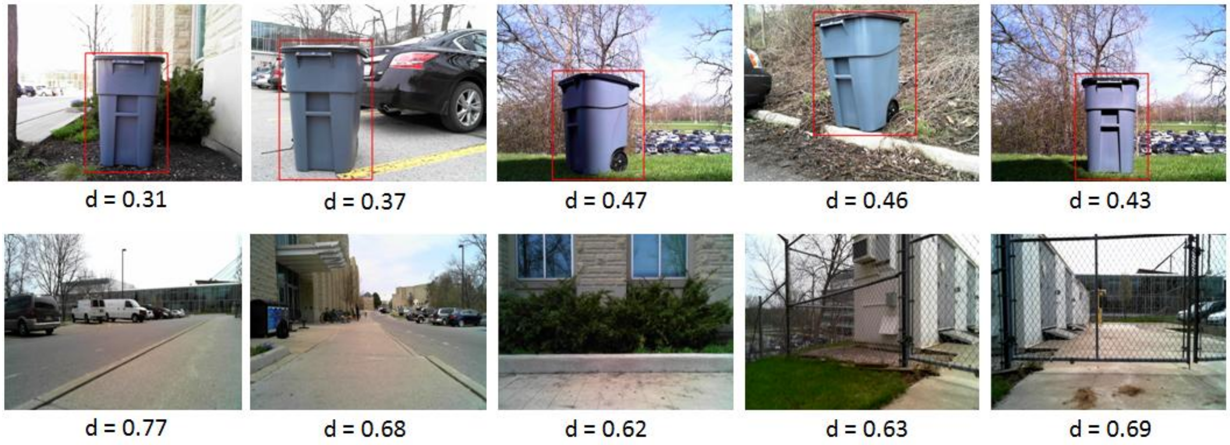


Figure 4.29: Pose Estimate After Step 1 and 2

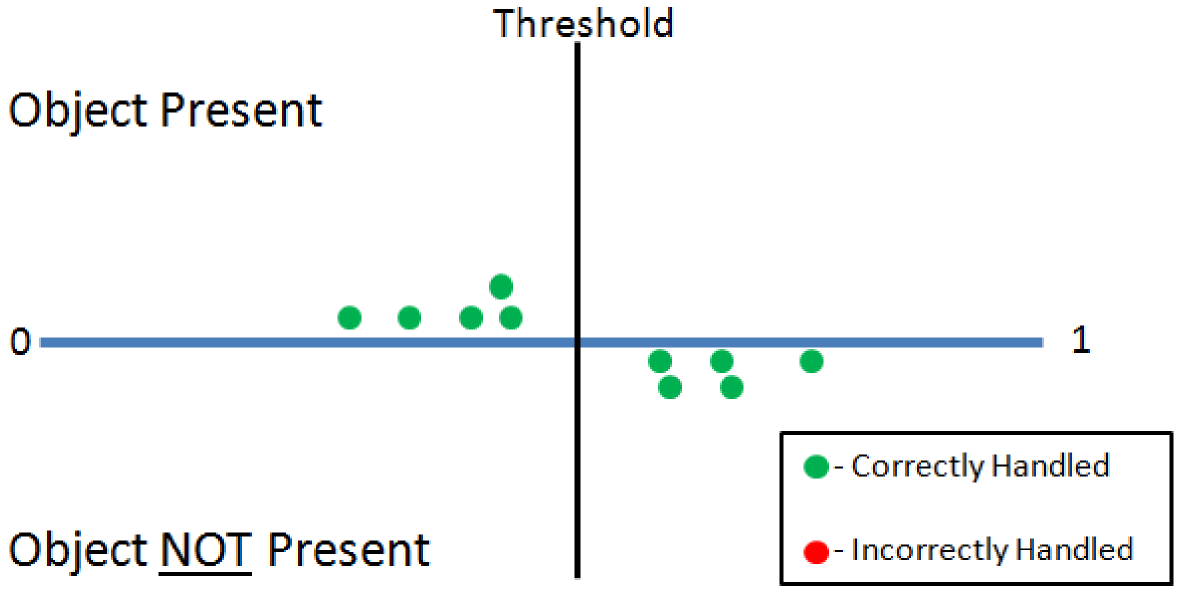


Figure 4.30: Real Number Line Evaluation for Modified GRM Plus HOG Verification

It should be noted that the test set used in this section was kept small in order to keep the illustrations simple and concise. However, in the evaluation section of this report 1007 test images will be used to statistically evaluate this method.

4.4 Overview of Find Pose Extraction Step

The final step in the pipeline is to extract/calculate the pose information for the object hypothesis. This is accomplished by consulting the metadata of the template to acquire the z coordinate, and then using this depth information (the z coordinate value) to calculate the x and y coordinates. During the template creation step data regarding depth and rotation were measured and recorded along with the template. Figure 4.31 shows an illustration of a template and its metadata, where for this particular template the distance between the camera and object during acquisition was 1.5 m with a 0° rotation about all three axes.

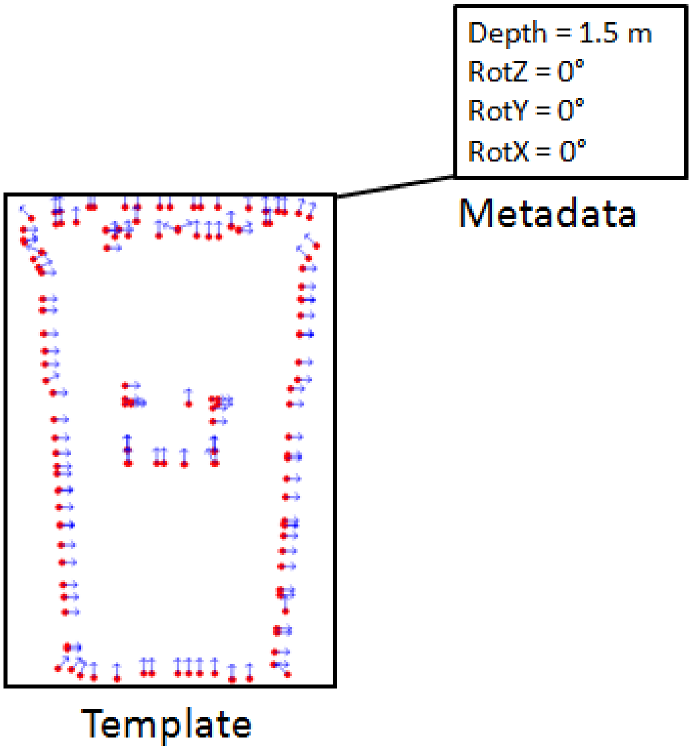


Figure 4.31: Template and its Metadata

After the depth information is extracted from the template's metadata the x and y coordinate are calculated using the pinhole camera model. The reason this works is because during template creation and online testing the intrinsic camera parameters are fixed and therefore the information is directly comparable.

Figure 4.32 shows the pinhole model with the image plane moved in front of the aperture to get rid of the 180° point flip. The associated equations for this model are shown below and will be used to calculate x and y.

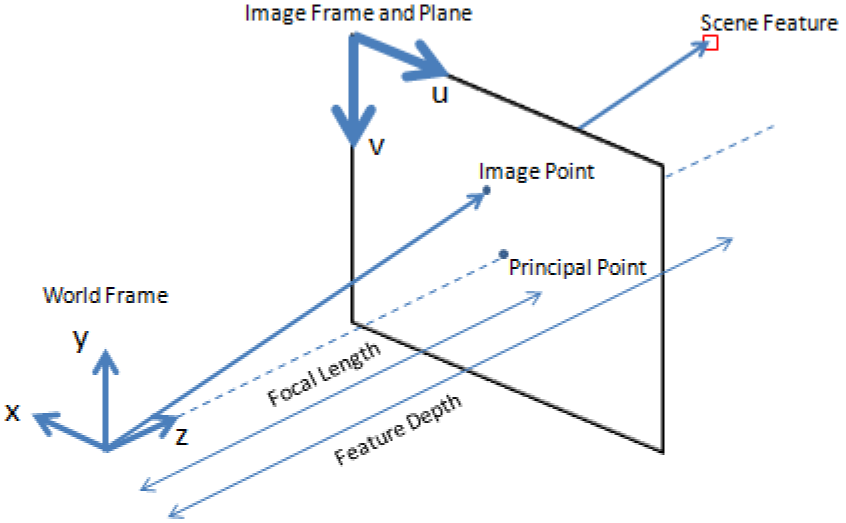


Figure 4.32: Pinhole Camera Model

Using this model the real world coordinates x and y can be calculated using the following relationship.

$$\binom{x}{y} = \frac{z}{f}\binom{u}{v} - \binom{c_x}{c_y}$$

where, z is the extracted depth coordinate from the metadata, f is the focal length, $\binom{u}{v}$ are the pixel coordinates, $\binom{c_x}{c_y}$ is the principal point, and $\binom{x}{y}$ are the calculated real world coordinates with respect to the camera Since f, z, c_x, c_y, u and v are known the calculation is straightforward.

It should be noted that in this thesis the focal length, f, does not change between template image acquisition and scene images acquisition and therefore the camera calibration process used to measure the intrinsic camera parameters, such a f, only needs to be done once. However, if different cameras are used between the two sets of images than calibration would need to be performed on both cameras and both focal lengths would need to be used in the calculation process of the x and y coordinate. This technique of estimating the pose will be evaluated in the evaluation section of the report, where the waste receptacle's coordinates will be measured in each test images and the estimated coordinates will be compared to those in order to determine accuracy.

4.5 Building the Template Database

As mentioned in the introduction of this section there are two stages to this project: online object recognition, and offline database creation. This section will explain the contributions made to the offline database creation stage of this project.

The process of creating a template, as described in the original paper [48], is accomplished by selecting gradient orientations based on where the highest gradient magnitudes exist within the image. It was also indicated in [48] that during the selection process the location of the points must be taken into consideration in order to avoid points forming a cluster around a few spots with high gradient magnitudes. This last point of concern makes sense but the method used to accomplish this task is not fully described by the authors [48] and so the implementation of template point selection and creation can be considered as a contribution made by the author of this thesis. Also, in the original paper [48] the topic of how many points should be used to describe the object is not addressed, and whether or not this number should be a constant across all scales was also not addressed. The latter point is important because as an object decreases in scale the number of boundary points decreases.

In this project a method was employed that produced a variable number of points per template depending on its size. Also, in order to deal with the problem of the points clustering around a few high gradient magnitudes a spatial grid was used to define neighborhoods, where only one point per neighborhood could enter the template set. Figure 4.33 shows three template images taken at different scales (depths). The number of local neighborhood in both images is roughly the same and therefore each template will produce a set a points with roughly the same number of elements. After the image has been split into spatial bins the orientation of the point within the bin with the highest gradient magnitude, above a small threshold, is used as an element within the template set. The fact that only one point per neighborhood is used addresses the problem of point clustering. Figure 4.34 shows the corresponding template point sets overlaid on the images and Figure 4.35 shows just the points with the image taken away. This last figure represents of the fact that in the end the image is not stored in memory and only the set of point locations and orientations are stored.

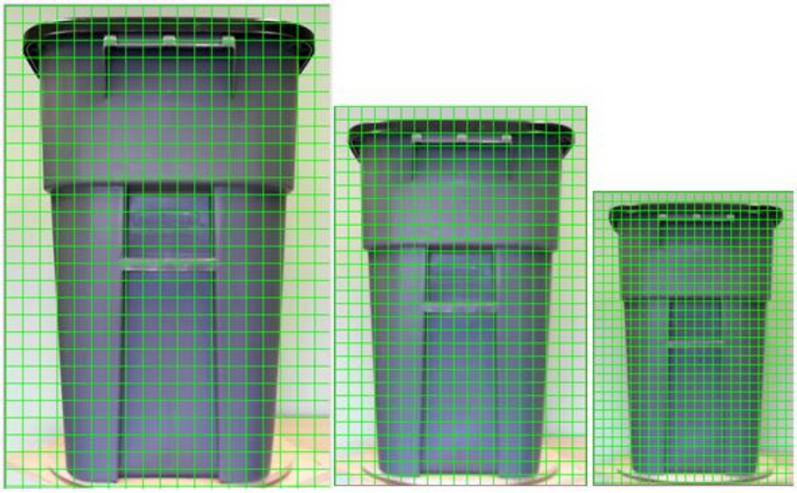


Figure 4.33: Local Neighborhoods for Point Candidates at Different Scales

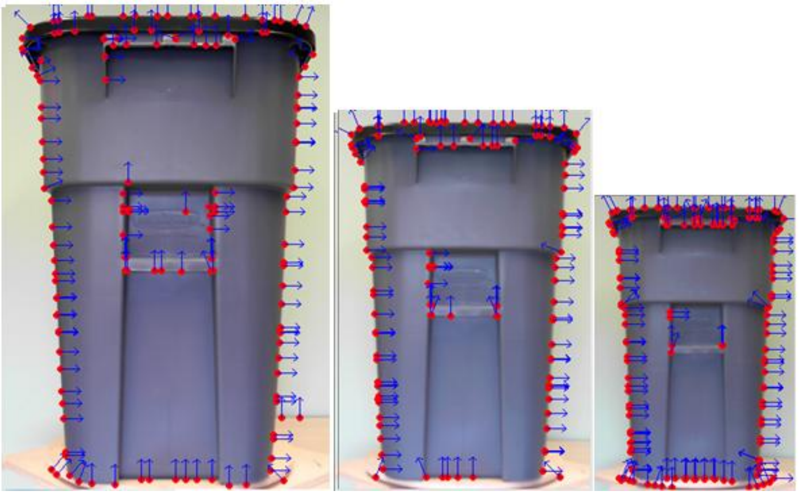


Figure 4.34: Template Points Overlaid on Images at Different Scales

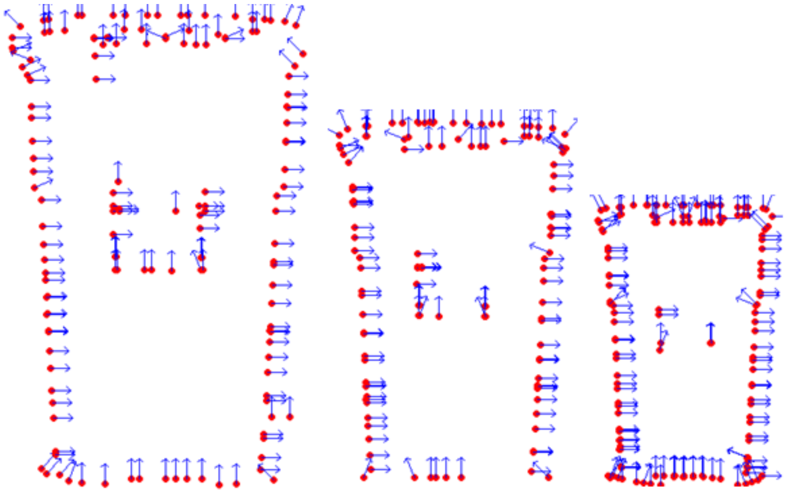


Figure 4.35: Set of Template Points for Each Image

In addition to this method for selecting points, the following image processing steps were performed on the template images before point selection: image averaging, and edge sharpening. Image averaging was done in order to reduce noise and edge sharpening was done in order to produce strong gradient responses along the edges of the waste receptacle. In the case of image averaging ten images of the waste receptacle were taken 20 ms apart from each other and averaged to produce a cleaner image, and in the case of edge sharpening the composite Laplacian kernel was used. For a detailed review of these image processing techniques and how they improve an image and its features please refer to appendix section 7.1.

Chapter 5

Results and Evaluation

This chapter will serve to evaluate the algorithm's performance under various conditions. The evaluation of this method is split into two sections. The first section evaluates the object detection component and the second section evaluates the pose determination component. If the object is found with the right template then the pose can be determined, therefore the evaluation of the pose determination component does not need the 1007 test images acquired to evaluate the detection component, but instead needs a separate set of test images.

5.1 Object Detection

This section uses 1007 test images to evaluate the object detection component of the algorithm. The test images consist of 502 images with the waste receptacle present, and 505 images without the waste receptacle present. The images contain many different lighting conditions, background sceneries, object poses, and weather conditions (unfortunately no snow images were acquired). A sample from each set of tests images is shown below in Figure 5.1and Figure 5.2.

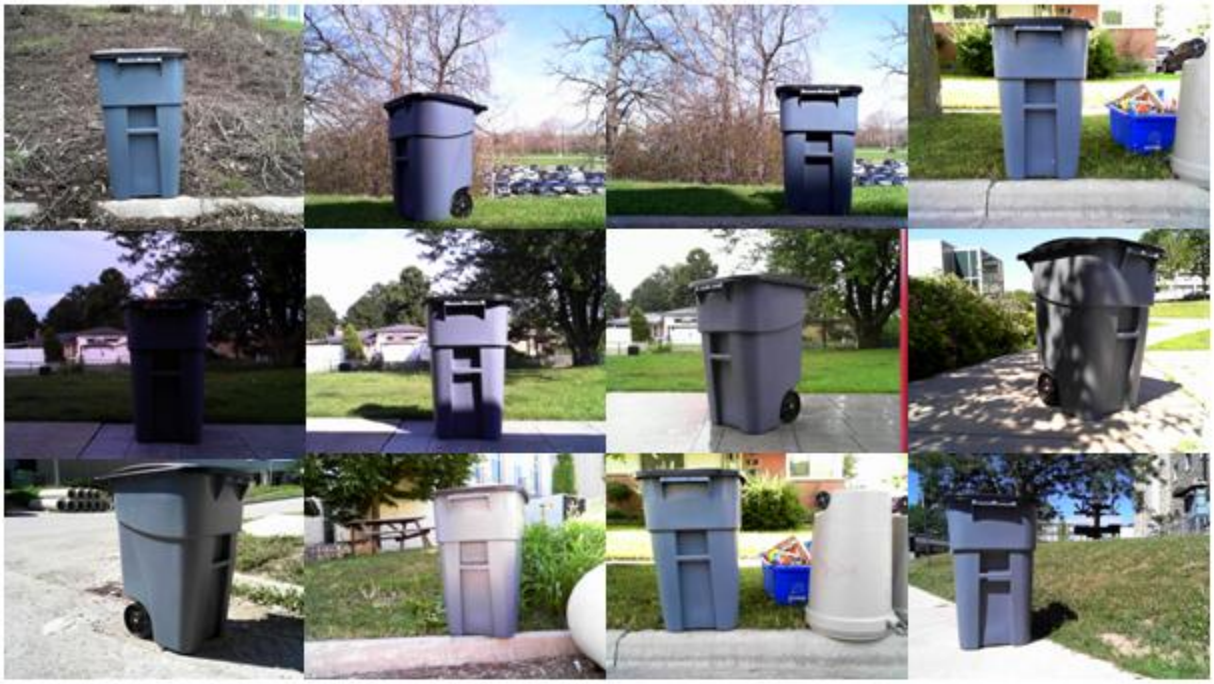


Figure 5.1: Sample of Images with the Waste Receptacle Present

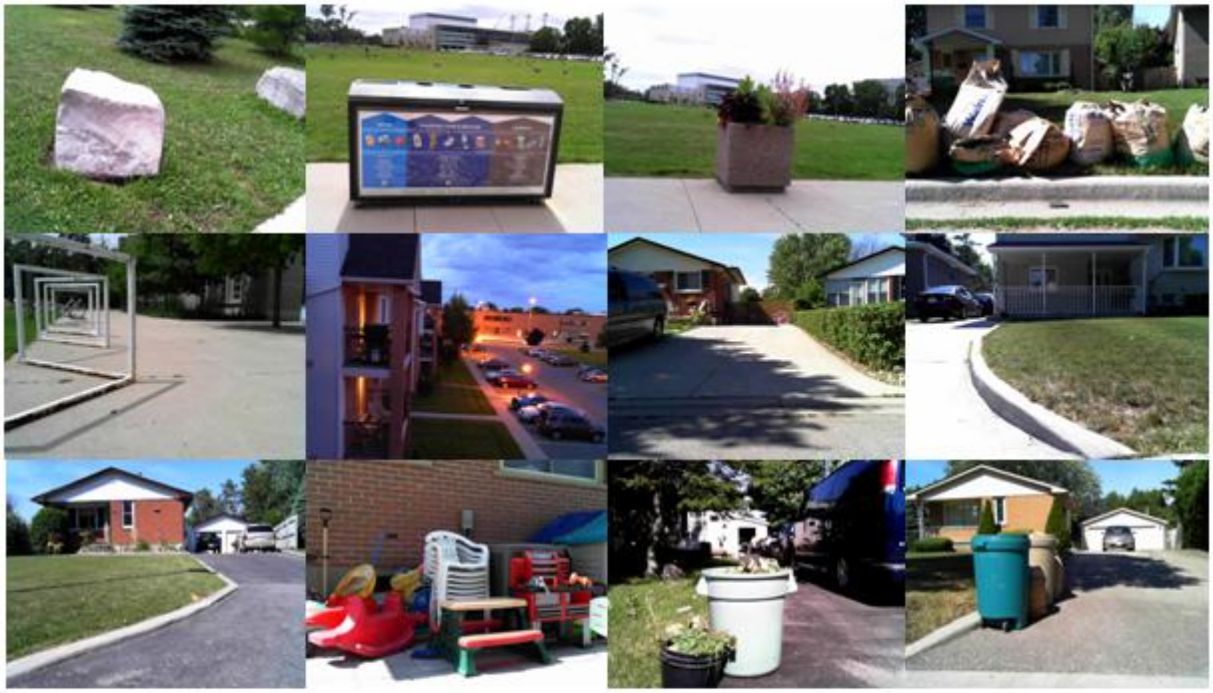


Figure 5.2: Sample of Images Without the Waste Receptacle Present

5.1.1 Real Number Line Evaluation of all Test Images

The first evaluation method used will be the real number line evaluation technique introduced in section Chapter 4. This method will plot the scores along a real number line to show the distribution of the distance/similarity scores. The points above the line correspond to images with the waste receptacle present and the points below the line correspond to image where the waste receptacle is not present. This evaluation method provides a visual representation of the two sets of scores. Ideally the two sets of scores should create two well defined clusters that can be thresholded.

In the plots that follow, the green dots indicate correct handling of the situation: that is a bin was found when it was present, or a bin was not found when it was absent. And the red dots indicate the opposite of the green dots: that is a bin is "found" when the bin is not present, or a bin is not found when the bin is present. Lastly, the vertical placement of the points in the number lines below are random. This was done so that the points do not overlap and create a solid line. Figure 5.3 shows the real number line evaluation for the algorithm described in the contribution section of this thesis. When the red points are not considered, Figure 5.3 shows that two well defined clusters are formed, which can be separated by a threshold. The red dots indicate that the algorithm missed some waste receptacles. These cases will be examined later on.

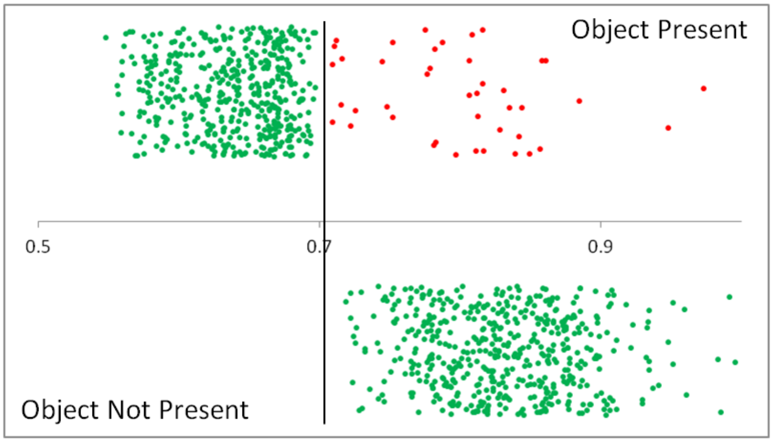


Figure 5.3: Real Number Line Evaluation of Thesis Algorithm

In order to compare the performance of this algorithm a real number line evaluation will be performed on the modified GRM algorithm detailed in Chapter 4 without the HOG verification step, and a real number line evaluation will be performed on the original GRM algorithm without any modifications. In these sets of evaluations, similarity scores are used instead of distance scores, so a higher value is better for when a waste receptacle is present. The results of these evaluations do not indicate if the waste receptacle was correctly handled, and to indicate this the red and green notation was dropped and a blue dot notation was employed.

Figure 5.4 shows the scores for the modified GRM, and Figure 5.5 shows the scores for the unmodified GRM. In both cases the two clusters are not well defined and a threshold cannot be applied to separate the two cases without creating a large amount of false positives and false negatives. There is better separation when the modified GRM algorithm is used which was expected from the results of the illustrative examples in the contribution section.

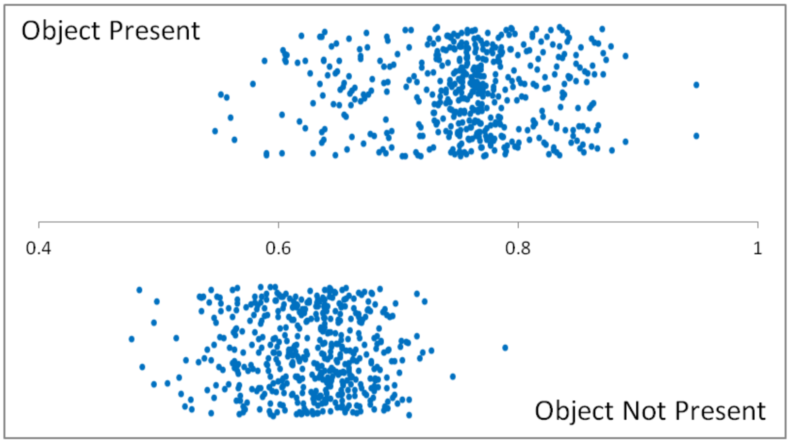


Figure 5.4: Real Number Line Evaluation of GRM with Modifications

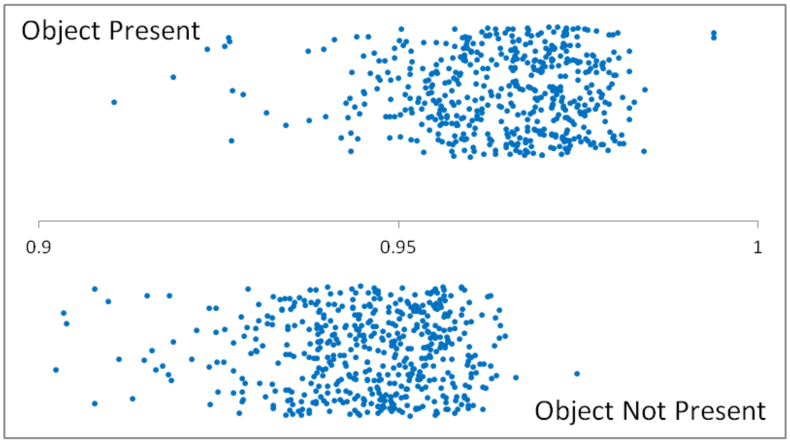


Figure 5.5: Real Number Line Evaluation of GRM without any Modifications

Therefore, these number lines show that the effects of the modifications to GRM and the addition of the HOG verification step observed in the larger data set above match the expectations raised by the illustrative examples presented in Chapter 5 (see Figure 4.4,Figure 4.26, Figure 4.30).

An alternative way of showing this information is to use a frequency vs. score plot. These plots demonstrate the amount of overlap between the two sets of data for each case outline above. It should be noted that in each figure below the red line is associated with the images where the waste receptacle is not present and the green line is associated with the images where the waste receptacle is present.

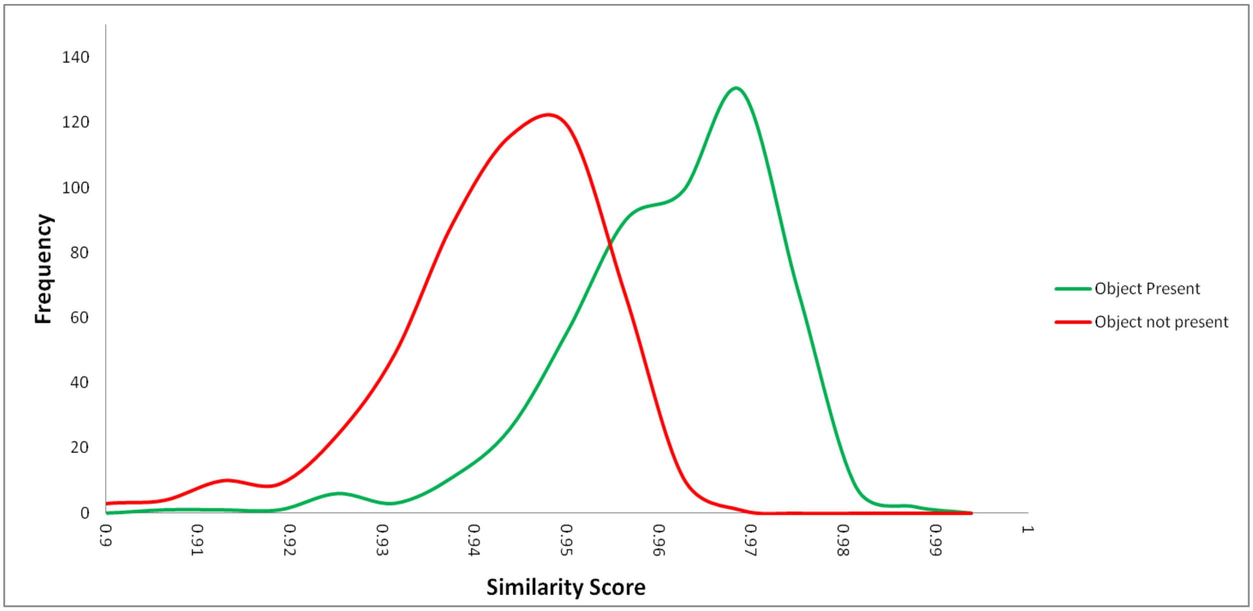


Figure 5.6: Frequency Vs. Similarity for Unmodified GRM

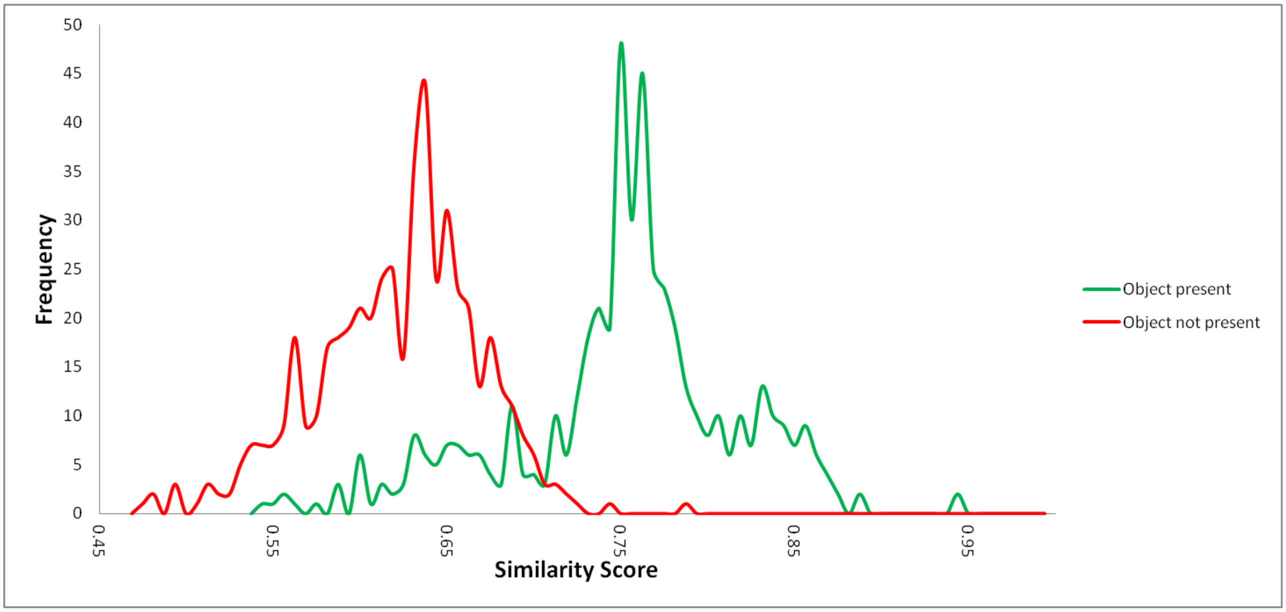


Figure 5.7: Frequency Vs. Similarity for Modified GRM

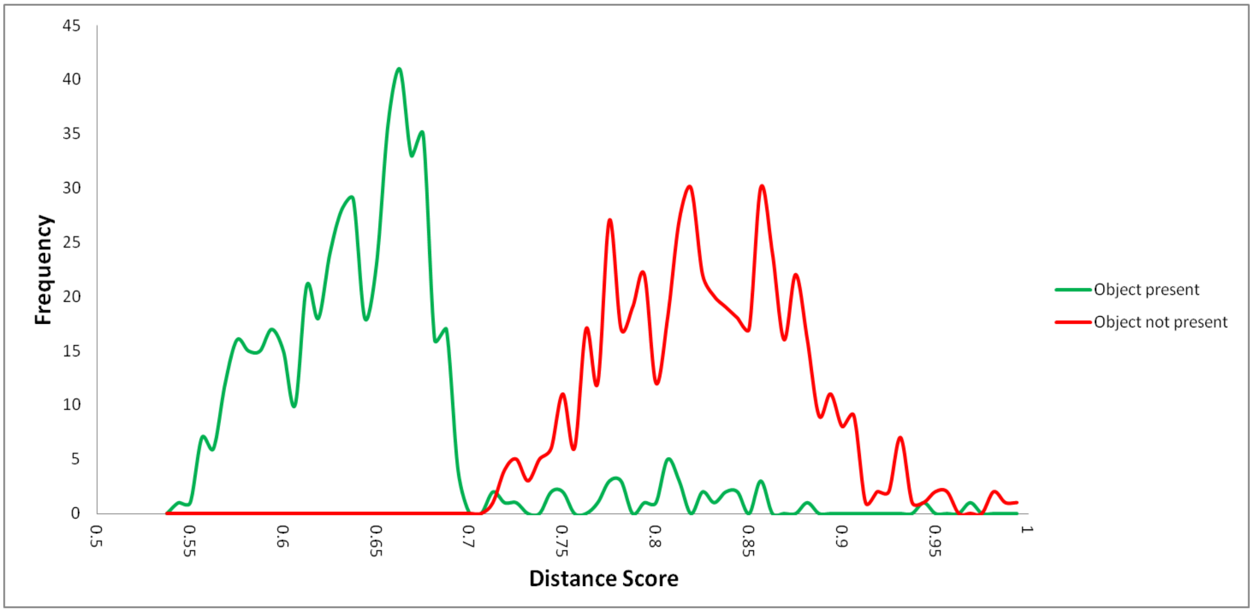


Figure 5.8: Frequency Vs. Similarity for Thesis Algorithm

The methods below evaluate the data set as a whole using statistical measures often used by the machine learning community to describe classifiers. These measures use the numbers in the confusion matrix below (see Figure 5.9).

		Hypothesis	
		+	–
Actual	+	460	42
	–	0	505

Figure 5.9: Confusion Matrix for Thesis Algorithm

There are many names given to these statistical measures. For, instance true positive rate, recall, sensitivity, and detection rate are all aliases of the same statistical measure. Therefore in order to avoid ambiguity Figure 5.10 clearly indicates the measures being used.

Name	Abbreviation	Description/Formulation	Value
True Positive	TP	Object present and found	460
False Positive	FP	Object not present but "found"	0
False Negative	FN	Object present but not found	42
True Negative	TN	Object is not present and not found	505
True Positive Rate	TPR	$\frac{TP}{TP+FN}$	0.916334661
False Negative Rate	FNR	$\frac{FN}{TP+FN}$	0.083665339
True Negative Rate	TNR	$\frac{TN}{TN+FP}$	1
False Positive Rate	FPR	$\frac{FP}{TN+FP}$	0

Figure 5.10: Definitions of Statistical Measures

These statistical measures indicate two things. First, the algorithm never finds an object when it is not present, as indicated by the TNR and the FPR. Second, the algorithm misses 8.37% of all waste receptacles it encounters, as indicated by the TPR and the FNR.

The red dots in Figure 5.3 correspond to the images where the receptacle was missed, and these missed cases can be attributed to one of following two scenarios: the receptacle was found by the GRM algorithm but the HOG vector distance threshold rejected the candidate, or the GRM algorithm failed to find the waste receptacle and the HOG correctly rejected the background candidate. Figure 5.11 shows some sample images from the former scenario, and Figure 5.12 shows some sample images from the latter scenario. In both figures the distance score is located in the bottom left corner and red boxes indicate the rejected GRM candidate. (Some possible methods to correct these shortcomings are mentioned in the future work section of Chapter 7.)

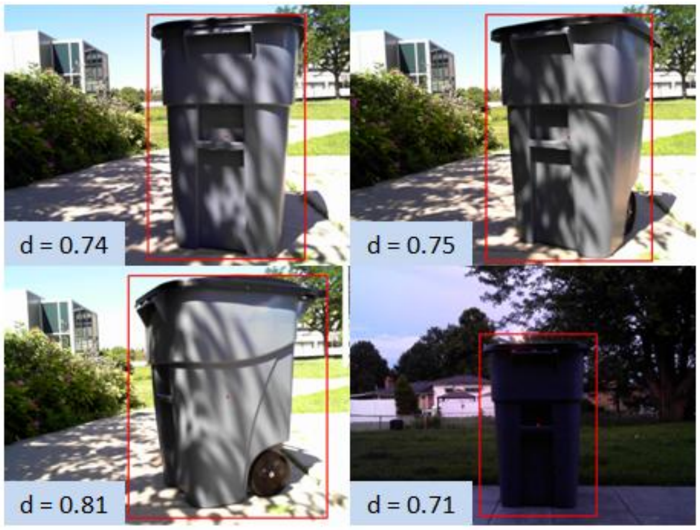


Figure 5.11: GRM Finds the Waste Receptacle but the HOG Rejects it

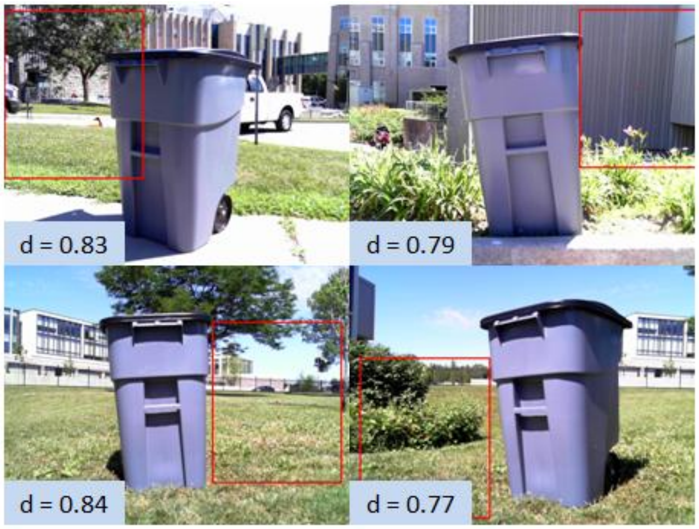
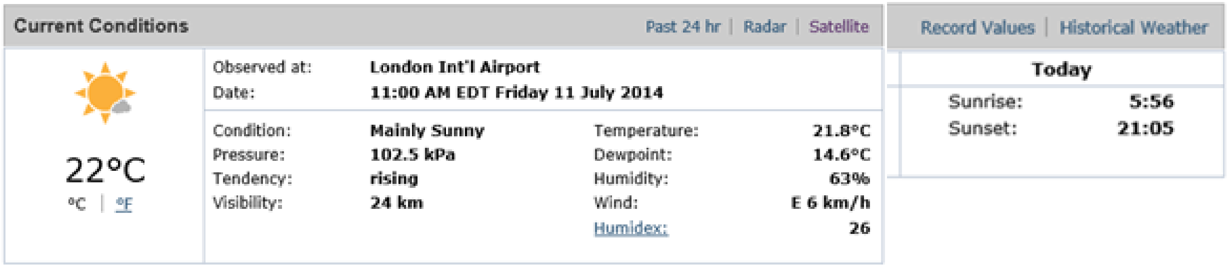


Figure 5.12: GRM does not find the Receptacle and HOG Correctly Rejects the Candidate

5.1.2 Testing Lighting Conditions (Approximately 16 Hour Time Lapse)

This section of Chapter 5 will evaluate the algorithms performance during natural lighting changes throughout a typical summer day. The evaluation takes place from just before sunrise to just after sunset, with pictures taken every ten minutes. Some weather data for this day is shown below (Figure 5.13).



Current Conditions | Past 24 hr | Radar | Satellite

Record Values | Historical Weather

22°C

°C | °F

Observed at:	London Int'l Airport
Date:	11:00 AM EDT Friday 11 July 2014

Condition:	Mainly Sunny	Temperature:	21.8°C
Pressure:	102.5 kPa	Dewpoint:	14.6°C
Tendency:	rising	Humidity:	63%
Visibility:	24 km	Wind:	E 6 km/h
		Humidex:	26

Today

Sunrise:	5:56
Sunset:	21:05

Figure 5.13: Weather Data for Time Lapse Images

In total 95 images were taken from 5:50 am to 9:30 pm. A sample of these images is shown in Figure 5.14 and the distance scores for every image in the entire set is shown in Figure 5.15. The results of this test show that the algorithm is stable during changes in lighting conditions. The only situation where the score crosses the threshold is when the waste receptacle is barely visible after sunset.

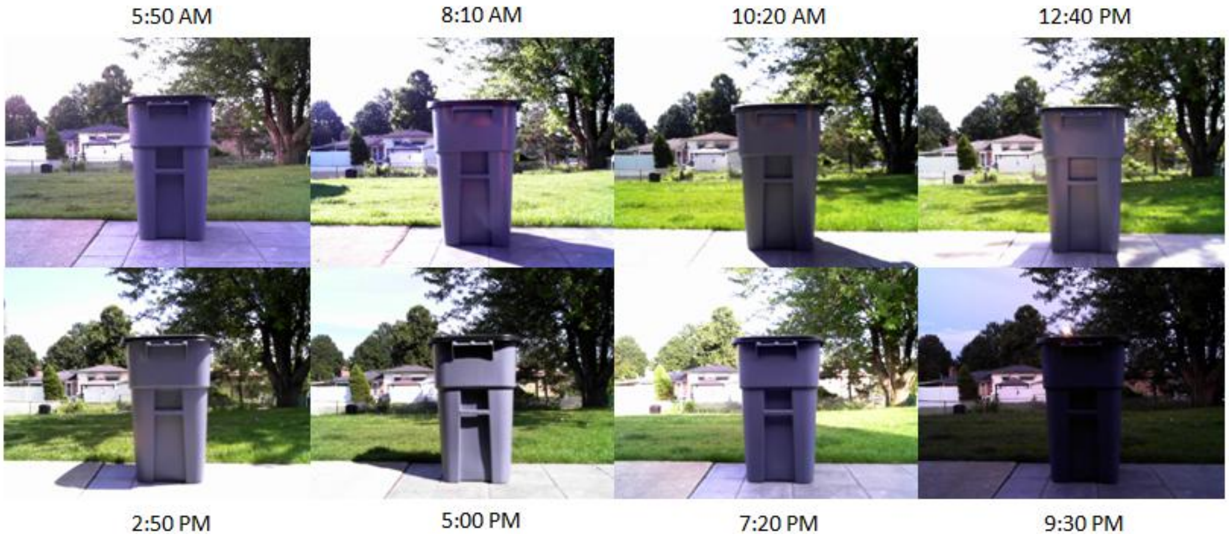


Figure 5.14: Image Approximately Every Two Hours

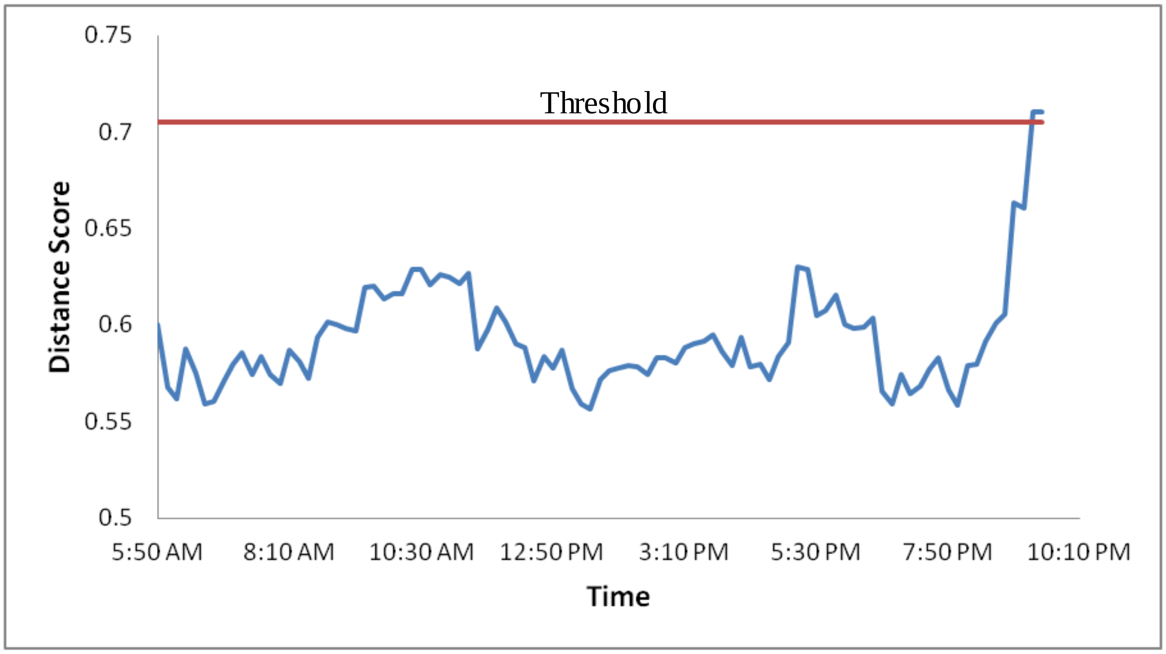


Figure 5.15: Distance Score Vs. Time of Day

5.1.3 Testing Rain Conditions

This section of Chapter 5 will test the algorithms performance during various degrees of rain intensity. A weather radar image showing the variation in rain intensities experienced that day is shown in Figure 5.16: Precipitation Radar Image. For convenience the intensities of the rain have been categorized, where greater than 12 mm/h is heavy, between 4 and 12 mm/h is medium, between 0 and 4 mm/h is light, and 0 mm/h is no rain. Figure 5.17 shows a sample of the rain images that have been passed through the algorithm.

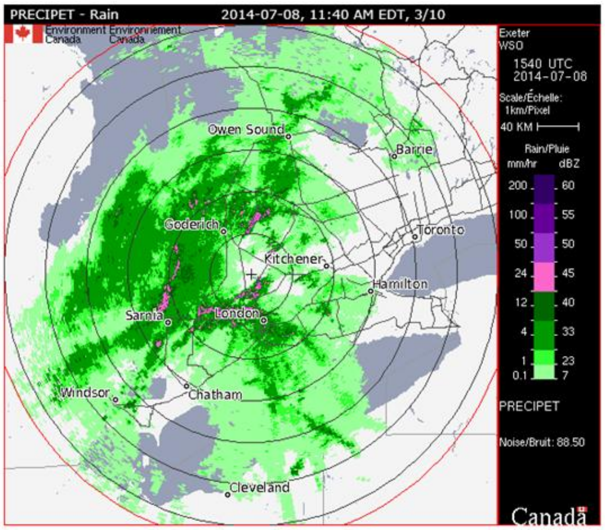


Figure 5.16: Precipitation Radar Image

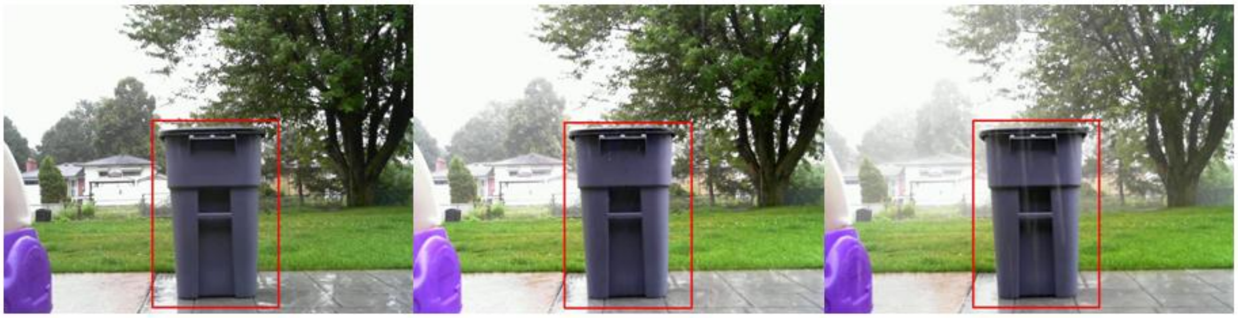


Figure 5.17: Sample of Rain Test Images

In total 224 images were used to evaluate the algorithms performance during various rain conditions. The images include high, medium, light and no rain intensities. Figure 5.18 shows the distance scores for each frame. All frames remained under the threshold and were properly detected. During the medium and heavy rain intensities the distance scores experienced greater fluctuation, likely due to the increased amount of water droplets producing unwanted gradients within the interior of the object during HOG construction. There are still fluctuations in the distance scores during the light and no rain conditions but to a lesser degree. Two things should be noted regarding the no rain situation. First the fluctuations are likely due to the rapid cloud movement, and second the apparent drop in score (which is good) along the transition from light rain to no rain is likely due to the fact that the clouds started to let more light through as the rain stopped.

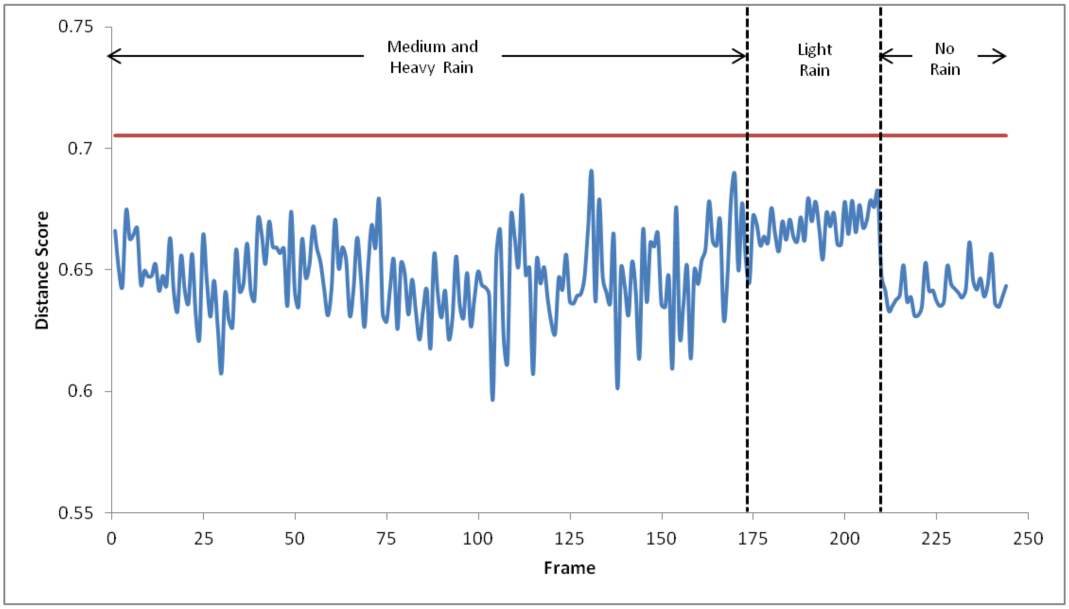


Figure 5.18: Distance Scores for Rain Images

5.1.4 Testing Occlusion

This section of chapter 6 will test the algorithms ability to handle occlusion. To test occlusion a black box was incrementally moved across the test image, which is an approach similar to the one used by Hinterstoisser [48]. As a comparison the GRM algorithm was subjected to the same type of occlusion test. The result of this test showed that, just like the GRM algorithm, the relationship between the algorithm scores and the amount of occlusion is linear.

Figure 5.19 shows the relationship between similarity score and occlusion for the original GRM algorithm and the relationship is indeed linear as reported by the original paper [48]. Figure 5.20 shows the relationship between the distance score and occlusion for the algorithm presented in this paper. The relationship is also linear and when the amount of occlusion exceeds 40% the distance score crosses the threshold and becomes incorrectly segmented. Figure 5.21 shows what the object looks like when it is 40% occluded.

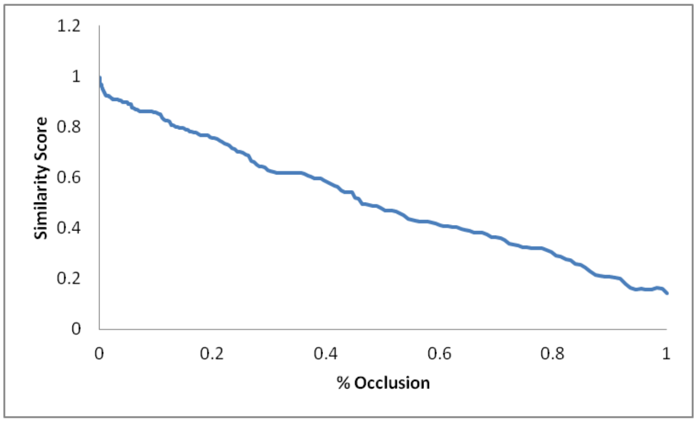


Figure 5.19: Similarity Vs. Occlusion for GRM Algorithm

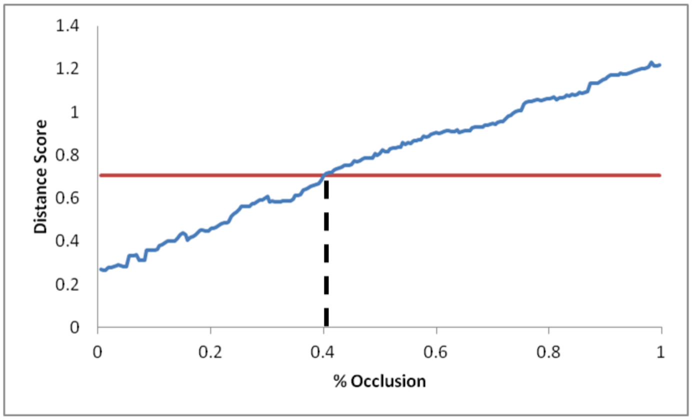


Figure 5.20: Distance Vs. Occlusion for Thesis Algorithm

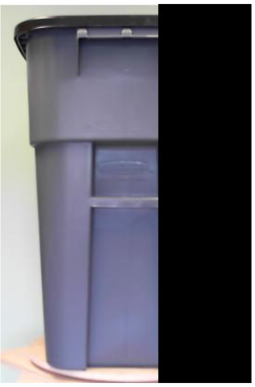


Figure 5.21: Object 40% Occluded

It should be noted that in Figure 5.20 there is an offset in the distance score at zero percent occlusion. This offset is attributed to the GRM returning a candidate location with at most a three pixel translation from the correct location. These small translations do not affect the similarity scores of GRMs because GRMs are robust to small deformation and translations (recall Section 3.1), however, this translation shows up in the HOG vector and causes the distance measure to have an offset when matching the template HOG to the candidate HOG. To better illustrate the effect this translation has on the HOG the process of plotting distance vs. occlusion was repeated, but this time the GRM was restricted to give back a candidate location without any translation. When the GRM filter is not allowed to return a location with a translation the offset at zero percent occlusion is much closer to zero (see Figure 5.22). There is, however, still a small offset but this can likely be attributed to the noise difference between the input image and the template image.

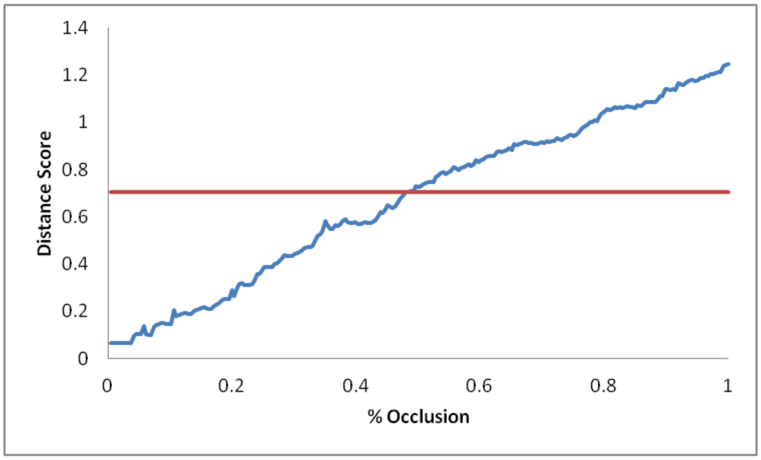


Figure 5.22: Distance Vs. Occlusion when GRM Cannot Return Candidate with Translation

Even though the relationship between distance and occlusion is better when the GRM is not allowed to return a candidate with translation the trade off for this improvement is speed. When the GRM is allowed to return a candidate with a three pixel translation in x and y it is nine times faster than when it cannot because the amount of locations that need to be searched are reduced by a factor of nine. That is for an image with dimension 640x480 no translation would result in 307200 locations being considered and when a three pixel translation is allowed this number is reduced to 34133 locations being considered.

5.2 Pose Determination

This section of Chapter 5 will evaluate the effectiveness of extracting the depth coordinate of the algorithm. Recall from section 4.4 that the pinhole camera model is being used to describe the relationship between the real world coordinates and the image coordinates. Also recall that during both template image acquisition and input scene acquisition the same camera with a fixed focal length is used and therefore camera calibration only needs to be applied once. The pinhole model and its governing equations have been reproduced below for convenience.

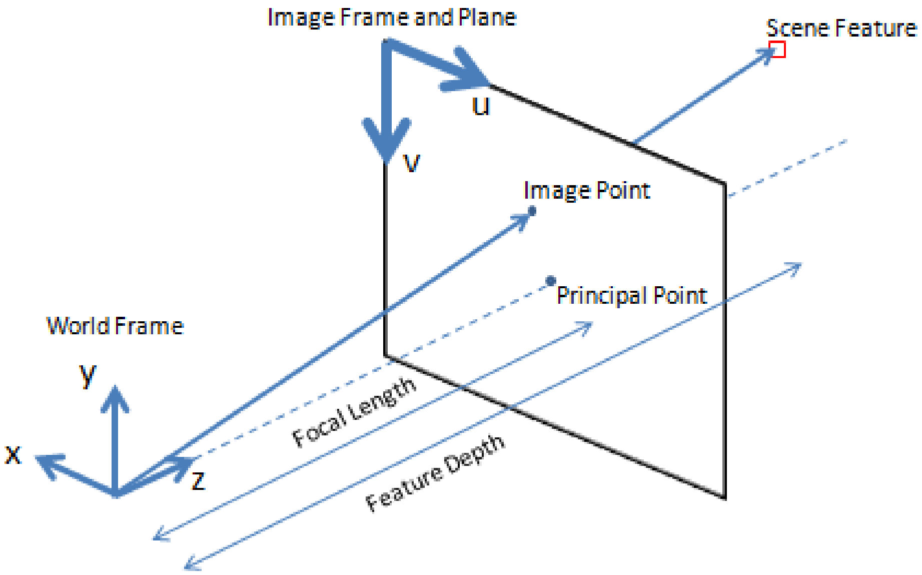


Figure 5.23: Pinhole Camera Model

Using this model the real world coordinates x and y can be calculated using the following relationship.

$$\binom{x}{y} = \frac{z}{f}\binom{u}{v} - \binom{c_x}{c_y}$$

where, z is the extracted depth coordinate from the metadata, f is the focal length, $\binom{u}{v}$ are the pixel coordinates, $\binom{c_x}{c_y}$ is the principal point, and $\binom{x}{y}$ are the calculated real world coordinates with respect to the camera. Since f, z, c_x, c_y, u and v are known the calculation is straightforward. By subscribing to this model the only parameter that needs to be evaluated is the depth estimation parameter, z, because all other parameters are fixed. Therefore, it can be concluded that if there is an accurate depth estimation then real world coordinates can be accurately calculated.

5.2.1 Evaluating the Depth Estimation Stored in the Metadata

In order to evaluate the depth estimation parameter an experiment was conducted that measured the real world length of masking tape placed on the surface of the waste receptacle using the pinhole camera model equations. Three pieces of masking tape twenty centimeters in length were used in this process and a total of 52 images were taken of the waste receptacle at different depths. A sample of the images is shown below in Figure 5.24.

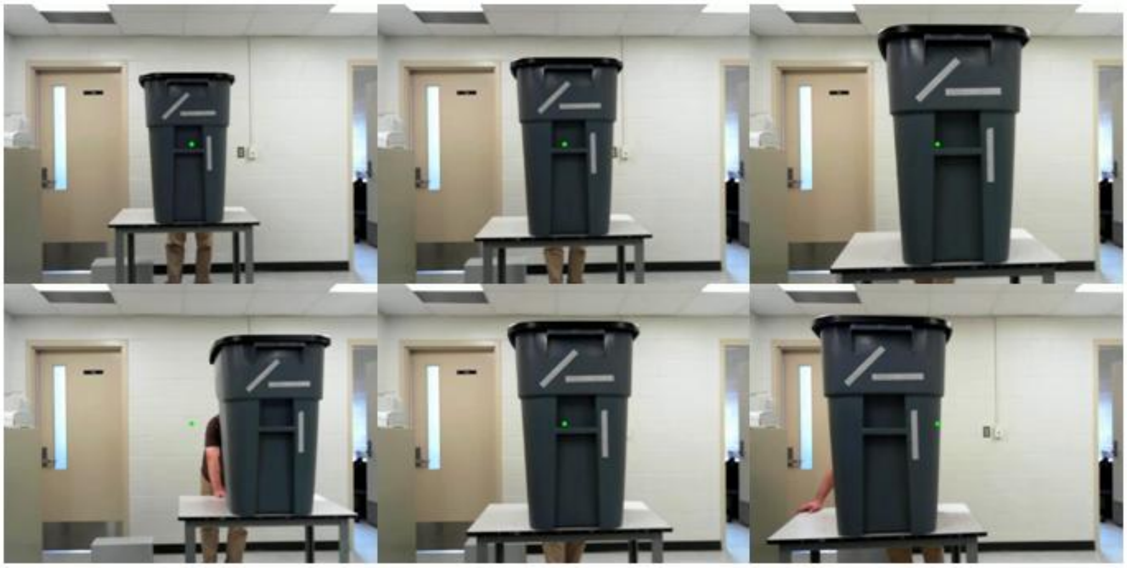


Figure 5.24: Depth Estimation Test Images

The calculated lengths of the tape were averaged to reduce the amount of error that arises in selecting the end points of the tape when measuring the pixel lengths during the calculation process. This averaged length measurement was than compared to the true length of the tape so that the error in the depth estimation could be evaluated. The following set of equations illustrate the process of estimating this error.

In order to evaluate the error in the depth estimation the true length and estimated length of the tape must be defined

$$L_T = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \frac{z_T}{f}\sqrt{(u_2 - u_1)^2 + (v_2 - v_1)^2} = 20\,cm$$

$$L_e = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \frac{z_e}{f}\sqrt{(u_2 - u_1)^2 + (v_2 - v_1)^2}$$

Where z_T and L_T represent the true depth and true length, and let z_e and L_e represent the estimated depth and estimated length. Note the true length is known but the true depth is unknown.

Dividing these two equations yields the following ratio:

$$\frac{L_e}{L_T} = \frac{z_e}{z_T}$$

The error in the depth estimation, measured in percentage, is defined as:

$$error = abs\left(1 - \frac{z_e}{z_T}\right) * 100$$

Therefore, by substituting the left side of the ratio above into the definition for the error calculation it can be shown that the error can be calculated using the lengths of tape, as shown below.

$$error = abs\left(1 - \frac{L_e}{L_T}\right) * 100$$

Using this calculation it was possible to calculate the error in the depth estimation for all 52 images, as shown in Figure 5.25. This figure shows that the error in estimating the depth is generally below 5% and that in the worst case the error was 5.5%. This 5.5% error translates into missing the true depth of the object by 9.2 cm, which is below the quantization distance between each template bin scale (12.7 cm).

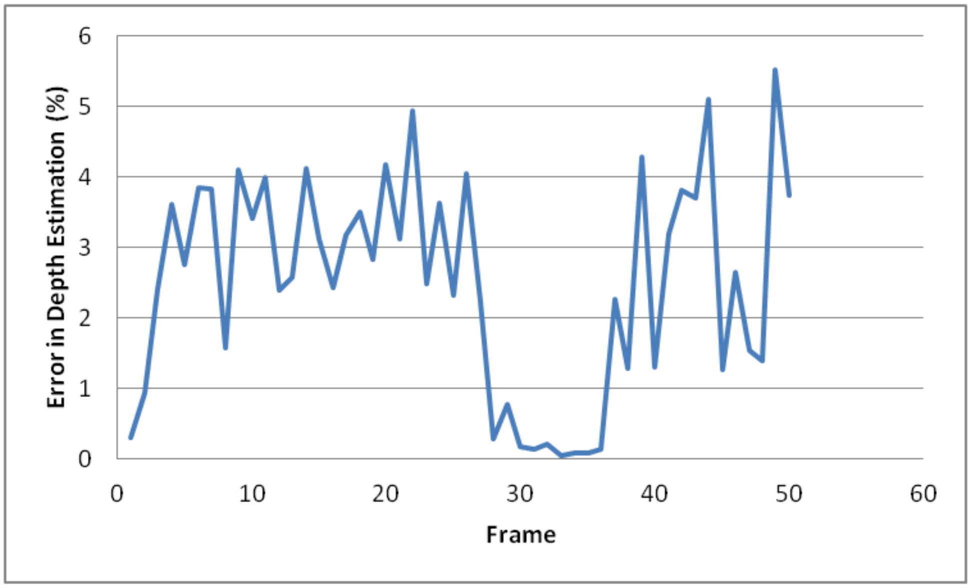


Figure 5.25: Error in Estimating the Depth Coordinate

Chapter 6

Conclusion and Future Work

This chapter is split into three sections. Section 6.1 will provide a summary of the thesis from motivations to results. Section 6.2 will outline the contributions made by the author of this thesis. Finally, section 6.3 will discuss future work.

6.1 Summary

The motivation for this thesis came directly from an industry need to make municipal waste collection more efficient. The goal of Waterloo Controls Inc. (the company sponsoring the project) and the researchers at UWO was to create a vision system that can detect a waste receptacle in real world conditions and then extract its real world coordinates, so that a robotic arm could acquire the waste receptacle. A high level schematic of the system in shown in Figure 6.1: High Level Schematic of Entire System, and the components relevant to this thesis are outlined in red.

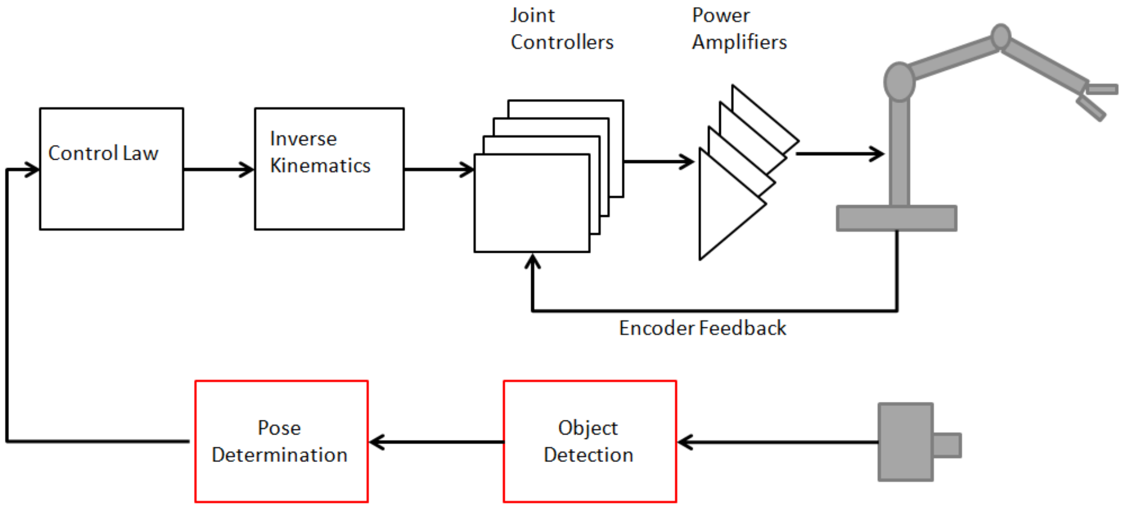


Figure 6.1: High Level Schematic of Entire System

The goals and constraints specific to the vision system component are outlined below. Regarding goal number one it can be concluded from the evaluation section of the report that this goal has largely been achieved. With the exception of winter, the test images capture the seasonal characteristics common to spring, summer and fall. Regarding goal number two, it can be concluded from the evaluation section that the real world coordinates of the waste receptacle can be estimated with at most a 5.5% error in depth, which is close enough for the hydraulic arm acquiring the waste receptacle. All of the design constraints were met. The algorithm used in this thesis is comprised of monocular vision based algorithms. Regarding the performance constraint, even with non-optimized and non-parallelized code the algorithm is able to process two frames a second. The optimization and parallelization of this algorithm is currently being undertaken by another member of the research team. Because of the inherent parallelizability of the designed algorithm, a large increase in efficiency is expected. Finally, the appearance of the waste receptacle was not altered in order to locate the object.

Goals:

1. Detect the waste receptacle in outdoor conditions
2. Determine pose information so the arm can acquire the waste receptacle

Constraints:

1. To keep cost down the system should use only one camera
2. The algorithm must be computationally efficient or collection time will not be reduced
3. The appearance of the waste receptacle cannot be altered

6.2 Contributions

The contributions made by this thesis provide good progress towards the realization of the vision system envisioned by Waterloo Controls Inc and the researchers at UWO. These contributions include:

1. An algorithm pipeline that includes a modified GRM and HOG verification step
2. A point selection method for template creation
3. A set of image processing techniques to make templates more robust
4. A contour enhancement step being added to the GRM algorithm
5. A second polling iteration being applied to the GRM algorithm
6. A linearization of the gradient response maps being applied to the GRM algorithm

6.3 Future Work

This section will present and discuss some opportunities that can be pursued with regards to the algorithm developed in this thesis. The list below presents some future work opportunities, and following the list is a discussion of each point.

1. Optimize and parallelize the algorithm in order to evaluate its efficiency
2. Acquire more test images (including winter scenery) to further validate the algorithm
3. Expand the number of templates to test more angles and objects
4. Evaluate the following changes made to the verification step
    i. Apply dynamic threshold changes based on light intensity measures
    ii. Replace HOG verification step with HSV colour space templates
    iii. Create a multiple layered verification step
5. Allow the GRM algorithm to select multiple candidates instead of just one
6. Allow for user interaction to reduce the instances of false negatives

Optimization and parallelization is the subject of concurrent work by another member of the research team here at UWO. It is expected to result in a marked improvement in the current efficiency of the algorithm, which is currently processing at a rate of 2 fps.

Acquisition of more test and template images is ongoing. This expansion of images will carry into the winter season, which will allow the algorithm to be tested under year round weather conditions.

There are many changes that can be made to the verification step of this algorithm. The rationale for testing changes made to this step would be to find a verification process that can help alleviate the issue of the GRM algorithm proposing a correct candidate and the verification step rejecting it, as shown the by the evaluation results in Section 5.1. One possible direction that could be explored to solve this problem would be to create a dynamic threshold that could be adjusted based on light sensors places near the camera. The theory behind this idea is that as the light intensity decreases the two clusters shift in the positive distance direction, which means the ideal threshold gets shifted as well. Another possible direction would be to replace the HOG verification template with an HSV colour space template and test if the robustness of this colour space can handle these irregular lighting conditions. Finally, another possible direction would be to combine several verifications processes, like HOG and HSV templates, which can be used to create a weighted score.

Regarding the candidate selection process, if the GRM algorithm was allowed to select multiple candidates, instead of just one, then this might alleviate the issue of the GRM algorithm not finding the waste receptacle when it is present. The idea here would be to allow some top percentage of scores be verified by the HOG. This would still reduce the amount of locations considered by the HOG which means the efficiency of the algorithm should not suffer. Some empirical studies would need to be implemented to find out what percentage would be ideal.

When the driver knows the waste receptacle is present but the algorithm fails to locate it (false positive) a method for user interaction could be applied. The idea is that the driver would input a coordinate via a touch screen monitor and the algorithm would process a smaller subsection of the scene around that point with relaxed thresholding. This would need to be tested but it is expected that this would reduce the amount of false negatives to zero. Lastly, since the driver would only have to do this about eight percent of the time (according to the data presented in the evaluation section), a semi-automated solution of this type would still constitute a significant reduction of the cognitive load on the driver.

Chapter 7

Appendix

7.1 A Primer on Image Processing

Image processing has been developed to address three major issues concerning pictures: picture transmission and storage, picture enhancement and restoration, and picture segmentation as an early stage to machine vision. In this thesis image processing techniques were used that center around two of these issues: picture enhancement/restoration, and extracting low level segmentations, such as edge maps. With that being said a brief overview of image processing related to these two issue topics will be discussed.

7.1.1 Algorithm Overview: Inputs, Transforms and Outputs

Image processing is performed on an input image, or images, by subjecting it, or them, to what are known as image transformations, the results of the transformations yields the processed image, or images. Image processing algorithms can vary in complexity depending on the type of process one is performing. Figure 7.1 illustrates this variability by showing how the number of transforms, inputs, and outputs can be variable (i.e., in this figure there are *p* inputs, *n* transforms, and *m* outputs).

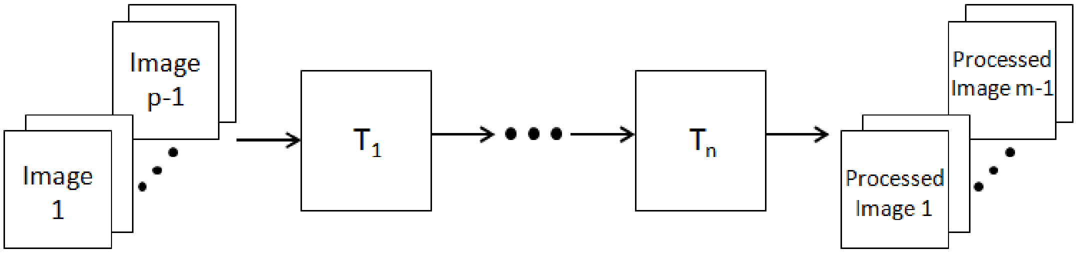


Figure 7.1: Image Processing Overview

For example, the process of averaging a set of images to reduce noise would take in multiple input images, let's say three, and then use one transform to pixel-wise sum a third of all the pixel intensities. This would produce one processed output image, which is composed of the average pixel intensities among all three images for each pixel location. Figure 7.2 illustrates this example.

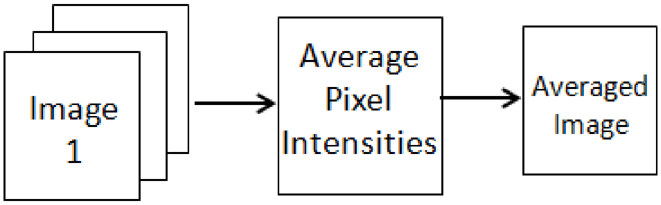


Figure 7.2: Image Processing Example of Averaging

7.1.2 Types of Transforms

Image transformations are performed using operators that take an input image and produce an output image. There are five main types of spatial operators that can be employed to process images: point operators, algebraic operators, geometric operators, convolution/correlation operators, and nonlinear operators. Point operators are functions that assign an intensity value to an output image pixel based solely on the intensity value of an input image pixel, an example application of this type of operation would be contrast enhancement. Algebraic operators are functions that produce output pixel intensities based on the pixel-wise sum, difference, product or quotient of two or more input images, an example application of this type of operation would be noise reduction via averaging (as seen in Figure 7.2). Geometric operators are functions that move pixels in an input image to a new location in an output image, an example of this type of operation would be affine transformation that can rotate, translate, and scale an image. Convolution/correlations operators can be considered functions that assign a pixel intensity in the output image based on a relationship between a neighborhood of pixel intensities in the input image, an example of the convolution operation would be noise reduction using a Gaussian filter. Lastly, a nonlinear operator can be considered a function that assigns a pixel intensity in the output image based on a nonlinear relationship between a neighborhood of pixel intensities in the input image, an example of this type of operator would be polling to reduce noise. In this thesis only three of the five types of operators were used to perform image processing tasks. More specifically this thesis employed the use of algebraic operators, convolution/correlation operators, and nonlinear operators. Therefore, in the following subsections a more in depth explanation of these types of operator will be given.

7.1.2.1 Image Transformations Based on Algebraic Operators

The process of pixel-wise summing, subtracting, multiplying, or dividing a set of input images to produce an output image is known an algebraic operation. Figure 7.3 shows a simple example of subtracting two images and taking the absolute value of the difference to produce a processed image known as the absolute difference image. All four types of algebraic operators are implemented in a similar fashion as shown in Figure 7.3, where the only real difference being the type of algebraic operation they are performing.

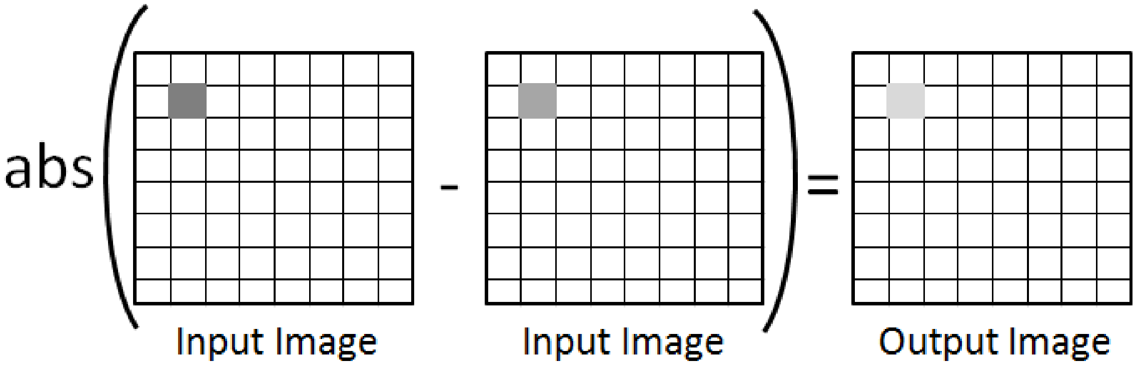


Figure 7.3: Absolute Difference Image

There are many different applications a person could use algebraic operators for so not all of them will be listed here. However, for each type of operation a one application will be pointed out below for completeness, but it should be noted that only two types of algebraic operators were used in this thesis: summation and multiplication.

Using an algebraic operator to sum a set of images can used to reduce noise. By taking the average of the sum the random noise generated during the image acquisition process can be reduced. Figure 7.4 shows the result of averaging images of the same scene taken at different times. The original set of images are very noisy as shown by the left most image, but after averaging sixteen of these types of images the amount of noise is greatly reduced, as seen in the right most image.

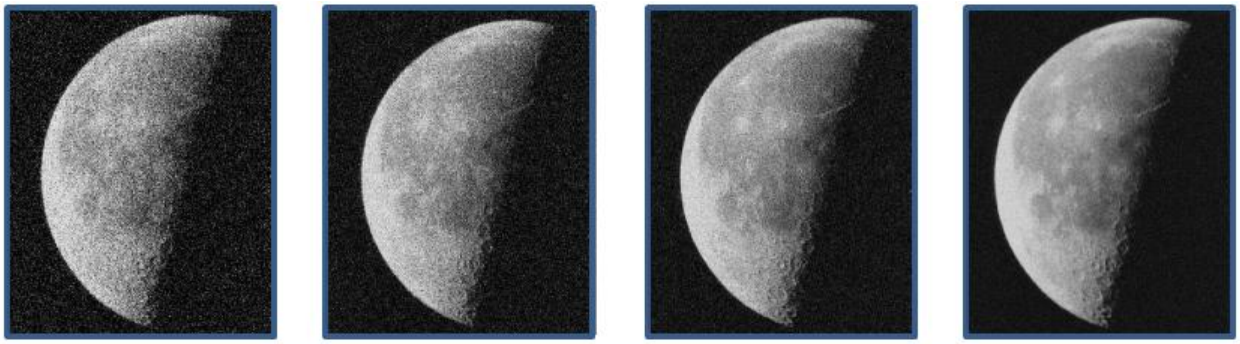


Figure 7.4: Image Averaging to Reduce Noise

Subtraction operators can be used for a multitude of applications, which include: change/motion detection, background removal, and image enhancement. A specific example is a procedure known as digital subtraction angiography [84]. This process involves the subtraction of X-ray images obtained before and after dye has been injected into the arteries of the patient.

Multiplication operators can be used to segment a section of an image. If a logical mask, that is a mask consisting of ones and zeros, is multiplied with an image then all of the spots with the value of one in the mask will remain in the output and all of the spots with zero in the mask will not. Figure 7.5 shows how a region of interest (ROI) segmentation can be obtained using the multiplication operation.

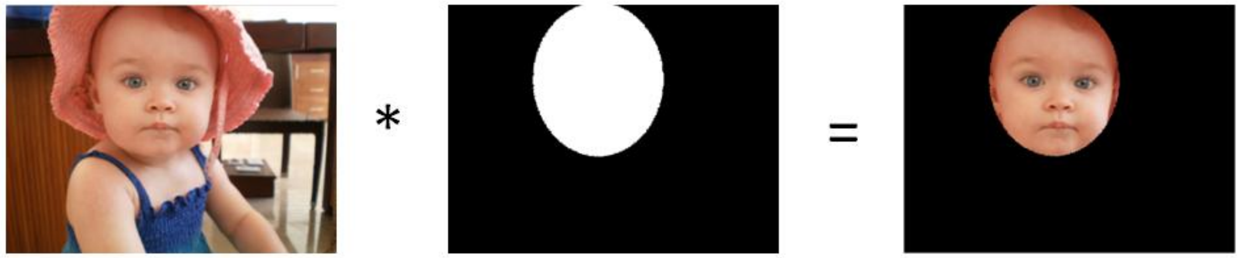


Figure 7.5: Segmenting a ROI using Multiplication

Image acquisition can be modelled as a multiplicative process where the acquired image is the product of object reflection and an illumination component, where the illumination component is considered noise. Therefore, if the illumination component could be measured, or estimated, then dividing the acquired image by this component would result in a noise free image.

7.1.2.2 Image Transformations Based on Convolution/Correlation Operators

Correlation and convolution operators process an input image by passing a kernel (a.k.a. a mask, or a filter) over the image. At each valid location a pixel-wise multiplication of overlapping pixels is computed, the products of these multiplications are summed to produce a pixel intensity in the output image. The only difference between correlation and convolution is how the kernel is passed over the image. In correlation the kernel is not changed and is passed over the image as is, and in convolution the kernel is first rotated 180° before passing over the image. Therefore, if a kernel was symmetrical then correlation and convolution would produce the same output image. Figure 7.6 shows the an example of how convolution/correlations works, it should be noted that the kernel is symmetrical so the result of both types of operations would be the same.

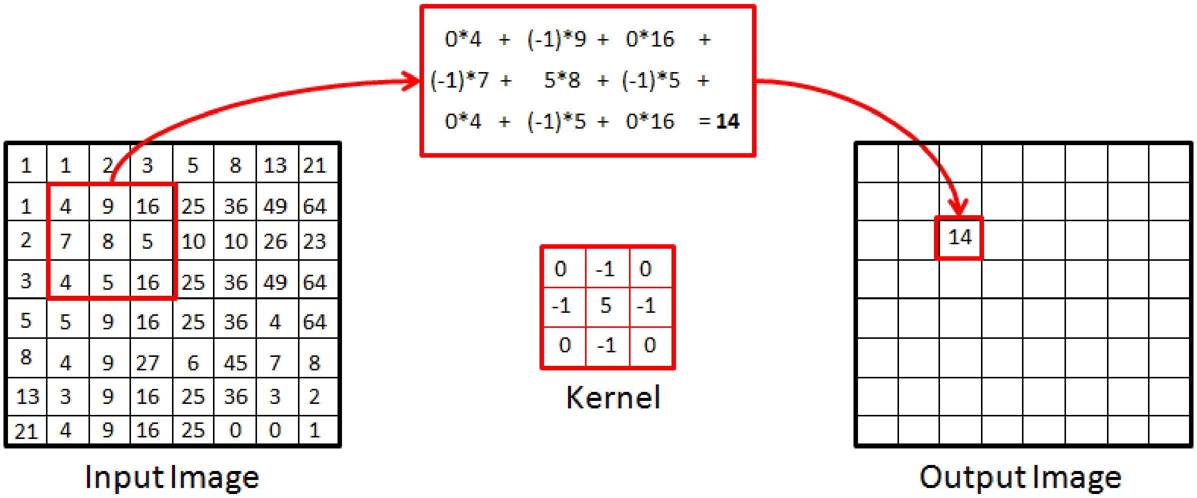
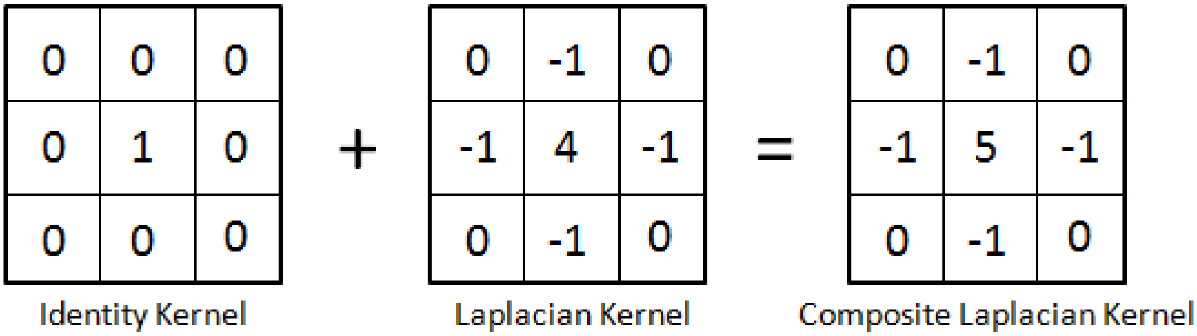


Figure 7.6: Convolution/Correlation Example

There are numerous applications for convolution/correlation operations in the field of image processing. Some of the key applications are: noise reduction, feature enhancement, and feature extraction. There are many more uses of the convolution/correlation operators but for this thesis the only applications mentioned above were used and so they will be briefly explained below.

There are many types of kernels that can provide noise reduction when applied to an image. The basic concept behind all of them is to smooth the image to get rid of noisy pixels, however, the drawback to smoothing an image to reduce noise is that edges within the image will begin to blur and as a result it will become harder to find concise transition between object boundaries within the scene.

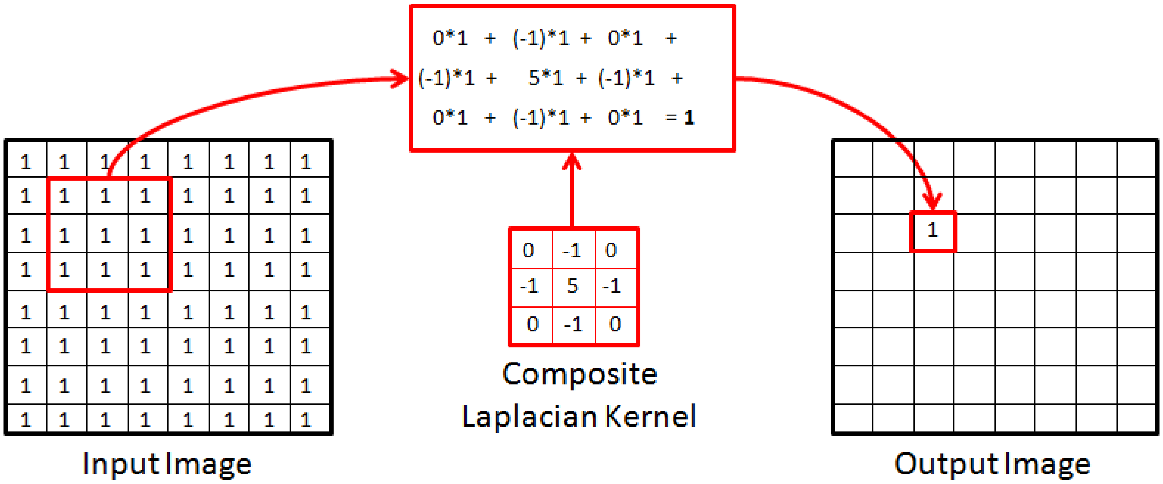
A common use for the convolution/correlation operator is to apply kernels that can enhance the edges of objects within an image. This process of enhancing edges is known as sharpening. The basic idea behind sharpening is to convolve an image with a differential kernel. This thesis uses the composite Laplacian kernel to enhance edges. The composite Laplacian kernel is the combination of the identity kernel and the Laplacian kernel as show below in Figure 7.7.



$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Figure 7.7: Composite Laplacian Kernel

Consider three cases: a pixel located in a homogenous area, a pixel located in on an edge, and a pixel located on a corner. Figure 7.9 - Figure 7.10 show the results of each scenario. The output pixel intensity is enhanced for cases two and three, but for case one the composite Laplacian kernel is reduced to the identity kernel and the output pixel is unchanged with respect to the input pixel.



$$0*1 + (-1)*1 + 0*1 + (-1)*1 + 5*1 + (-1)*1 + 0*1 + (-1)*1 + 0*1 = \mathbf{1}$$

Figure 7.8: Pixel Located in a Homogeneous Area

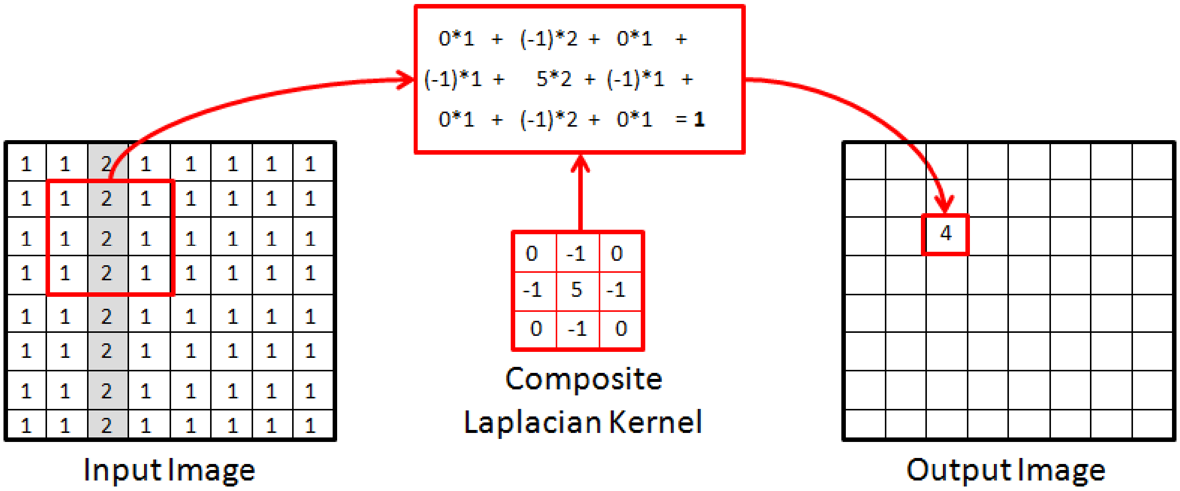


Figure 7.9: Pixel Located on an Edge

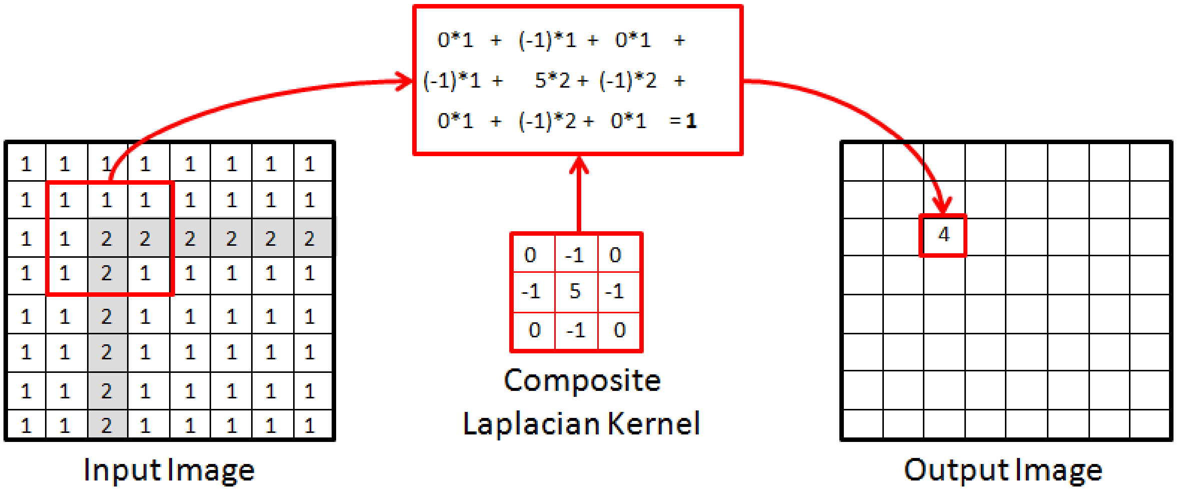
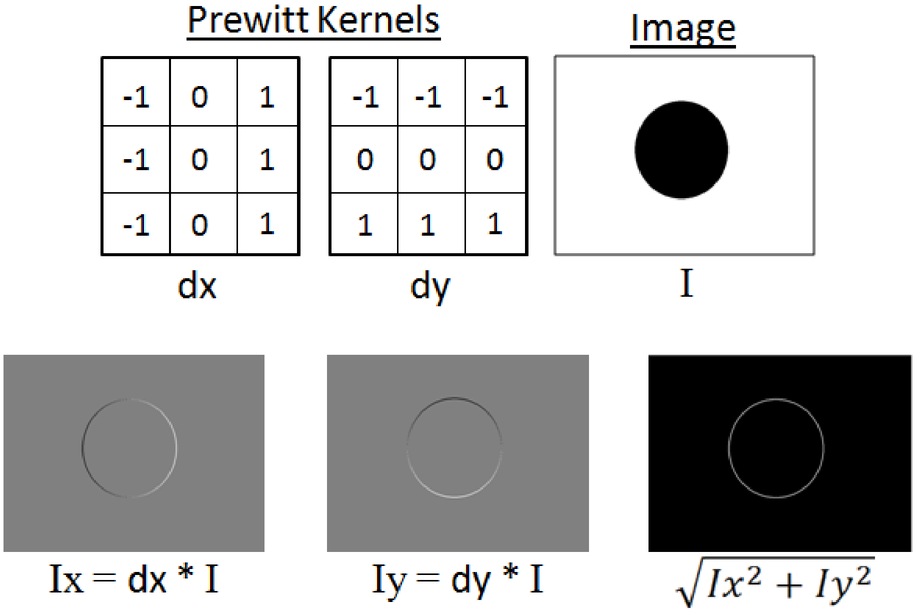


Figure 7.10: Pixel Located on a Corner

A common type of feature extraction is edge detection, which is the process of producing an edge map for an input image. There are many methods that can accomplish this task, but what all of these methods (with the exception of a few heuristic linking methods) have in common are that they use the convolution/correlation operator and derivative kernels. In this thesis all edge maps were produced by thresholding the gradient magnitude map after convolving the image with either the Sobel, or the Prewitt kernels. The specific application of both kernel types will be explained in a later section of this thesis. Figure 7.11 shows how an edge map is created using the Prewitt kernels. First gradient images Ix and Iy are calculated using the convolution/correlation operators and the Prewitt kernels and then the magnitude of these two images are computed to produce the final edge map. It should be noted that if the edge map was noisy then a threshold could be applied to reduce unwanted pixels from entering the edge map.



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

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

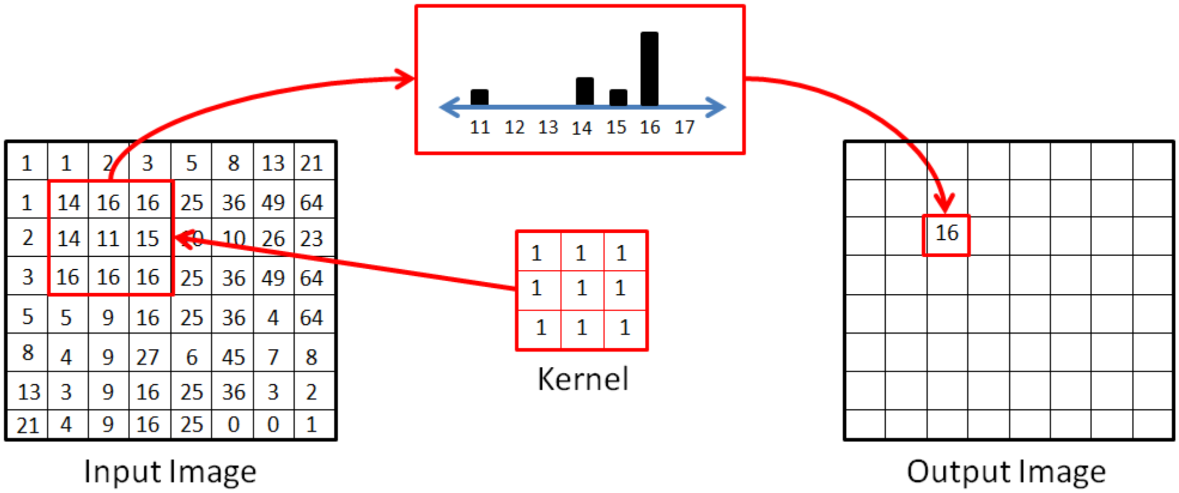
Figure 7.11: Edge Map of Circle

7.1.2.3 Image Transforms Based on Non-Linear Operators

Non-linear operator process an input image in a similar fashion to convolution/correlation, where the process is done by passing a kernel over the image. However, in the case of non-linear operators the kernels elements are not used to perform some kind of mathematical operation they are simple used to indicate which pixels will be active during the non-linear operation. A common non-linear operation is polling (a.k.a. mode filter). This type of kernel was used in this thesis and so will be briefly explained below.

The use of polling to reduce noise can be seen in Figure 7.12. It works by passing a kernel over an image and replacing the center point of the kernel with the pixel value that occurs most often within the active kernel locations (i.e., the modal pixel value is used in the output). In the example illustrated in Figure 7.12 the pixels that are active in the kernel window are indicated by ones. This example used all nine kernel elements as active, but a person could use any combination of zeros and ones to create a structured polling scheme. The reason polling eliminates noise is because normally pixels within a small local neighborhood have relatively similar intensities but if spurious noise, such as salt and pepper noise, is present within the image then a pixel can be drastically different in intensity when compared to its neighbors.

Polling would correct this situation by changing the noisy pixel intensity value to one that is similar to its neighbors. There are of course drawbacks to polling but they depend on the application you are trying to achieve. For instance, if the goal was to get rid of noise in an image but maintain crisp edge boundaries then polling may not apparent because the process will expand the edges and they would no longer be a crisp transition.



1	1	2	3	5	8	13	21
1	14	16	16	25	36	49	64
2	14	11	15	10	10	26	23
3	16	16	16	25	36	49	64
5	5	9	16	25	36	4	64
8	4	9	27	6	45	7	8
13	3	9	16	25	36	3	2
21	4	9	16	25	0	0	1

1	1	1
1	1	1
1	1	1

Figure 7.12: Reducing Noise Via Polling

7.1.3 Frequency Domain

Anyone that is considering image processing as a solution to achieve some kind of picture manipulation should be aware that processing can be done in both the spatial and frequency domains. With that being said, a person who is comfortable with both types of processing domains will likely be more successful at accomplishing their desired image processing task because some operations in the frequency domain are more practical and easier to implement than their counter parts in the spatial domain. Frequency domain filtering was used in this thesis and so a brief overview of how it works will be provided in the following subsections.

7.1.3.1 Producing Frequency Spectrum Images

Before image processing techniques can be applied in the frequency domain the image must first be transformed into the frequency domain. The frequency domain representation of an image consists of a matrix of complex numbers and so in order into view this image the frequency domain matrix must be split into two channels; usually visualized as the amplitude spectrum and the phase spectrum. Figure 7.13 shows an example of an amplitude and phase spectrum of an image.

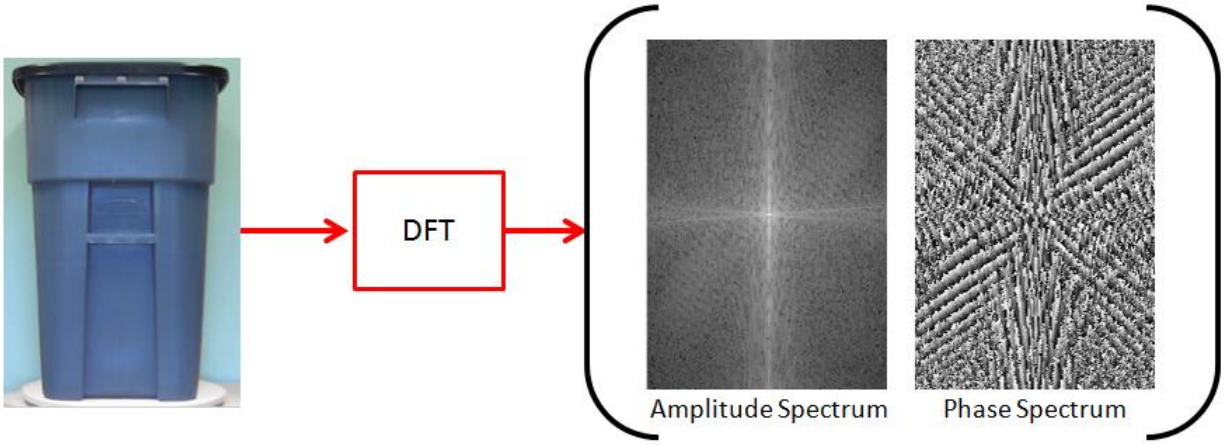


Figure 7.13: Amplitude and Phase Spectrum

The process of converting an image from the spatial domain to the frequency domain is accomplished via the Fourier Transform. More specifically since digital images are discrete by definition the Discrete Fourier Transform (DFT) is used to transform the image into the frequency domain. Conversely, the process of going from the frequency domain to the spatial domain is accomplished by the Inverse Discrete Fourier Transform (IDFT). Respectively, (7.1 and (7.2 show the DFT and IDFT. Where $g(i,k)$ is the spatial image and $G(m,n)$ is the frequency image.

$$G(m,n) = \frac{1}{N}\sum_{i=0}^{N-1}\sum_{k=0}^{N-1} g(i,k)e^{-j2\pi\left(m\frac{i}{N}+n\frac{k}{N}\right)} \tag{7.1}$$

$$g(i,k) = \frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} G(m,n)e^{j2\pi\left(i\frac{m}{N}+k\frac{n}{N}\right)} \tag{7.2}$$

7.1.3.2 Filtering in the Frequency Domain

After an image has been transferred from the spatial domain to the frequency domain it is ready for processing. In the spatial domain a common practice to filter an image is to convolve it with a kernel, however, in the frequency domain the image is filtered via pixel-wise multiplication. This relationship is known as the convolution theorem and is mathematically shown below in (7.3. The equation reads as follows: the Fourier transform of an image convolved with a kernel is equal to the multiplication of the Fourier transform of the image and the Fourier transform of the kernel.

$$\boldsymbol{\mathcal{F}}\big(\boldsymbol{f}(\boldsymbol{i},\boldsymbol{k}) * \boldsymbol{g}(\boldsymbol{i},\boldsymbol{k})\big) = \boldsymbol{F}(\boldsymbol{m},\boldsymbol{n})\boldsymbol{G}(\boldsymbol{m},\boldsymbol{n}) \tag{7.3}$$

In this thesis the convolution theorem was applied to filter out low frequency image components. The type of filter used to accomplish this task was a Gaussian high pass filter (GHPF). Figure 7.14 shows the result of filtering the amplitude spectrum with the GHPF. As you can see the DC component in the center and the low frequency components around it have been attenuated. The specific application of this process in this thesis will be explained in a later section.

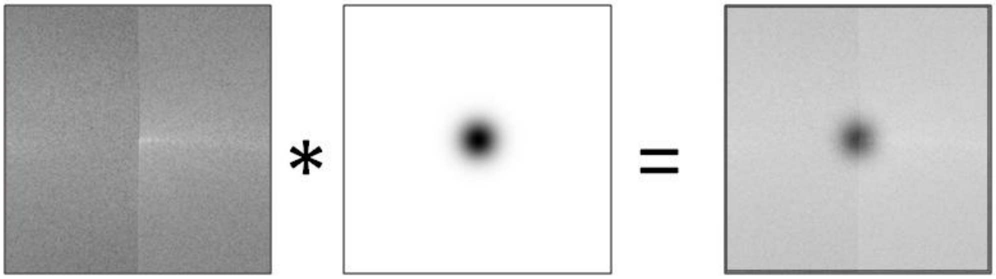


Figure 7.14: High Pass Filtering in Frequency Domain

7.2 Additional Experimental Results for Section 5.2

This section contains four more sets of experimental results regarding the addition of a second polling iteration and the linearization of the cosine response for section 4.2.

7.2.1 1D and 2D Plots for Test Image #1

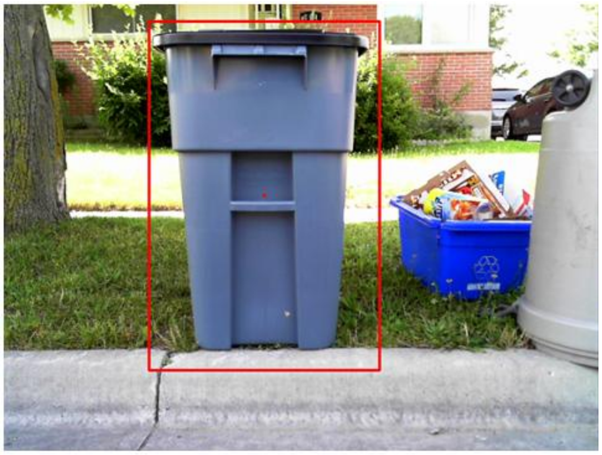


Figure 7.15: Test Image #1

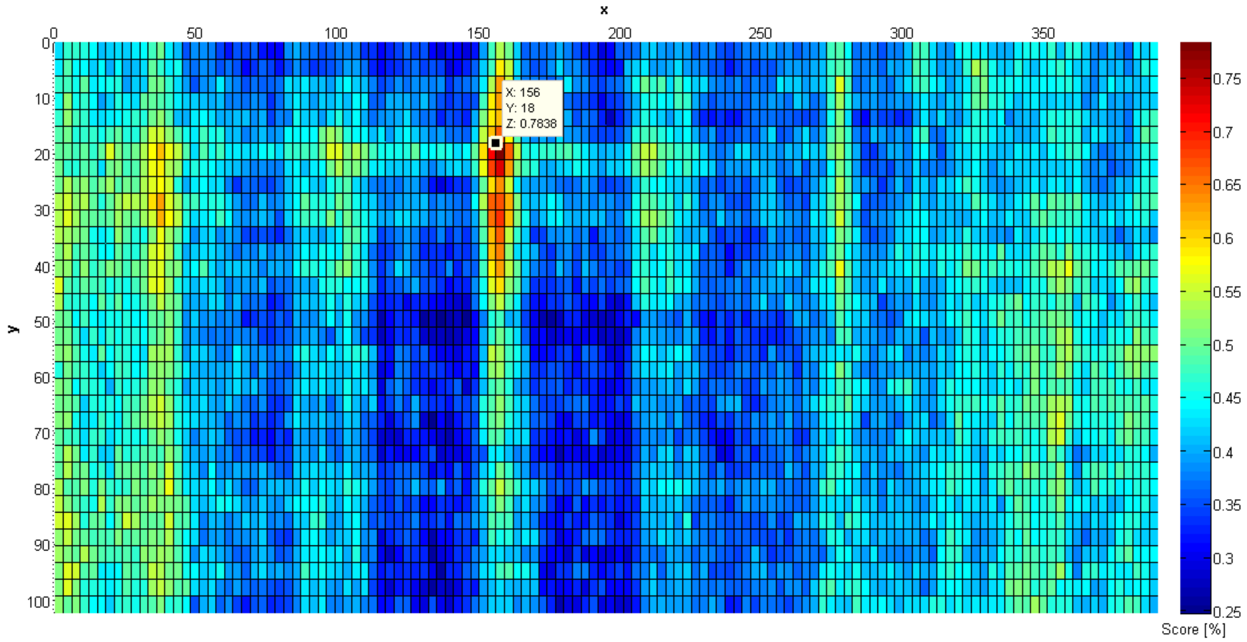


Figure 7.16: 2D Plot with Polling Twice and Linearization for Image Number 1

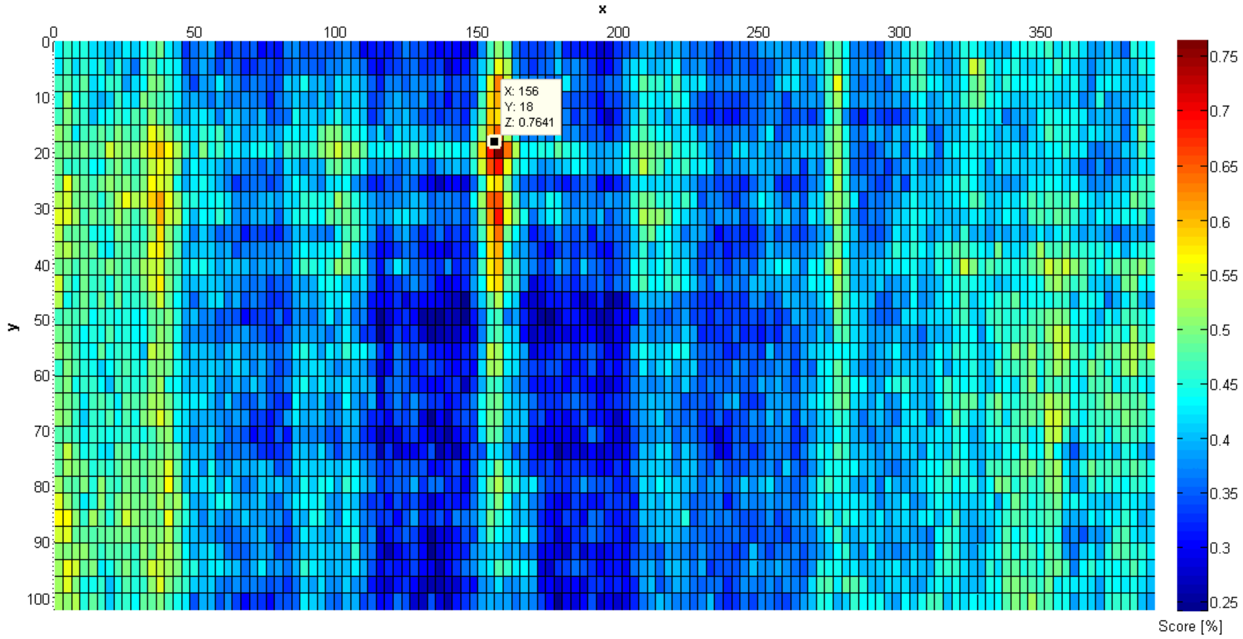


Figure 7.17: 2D Plot with Polling Once and Linearization for Image Number 1

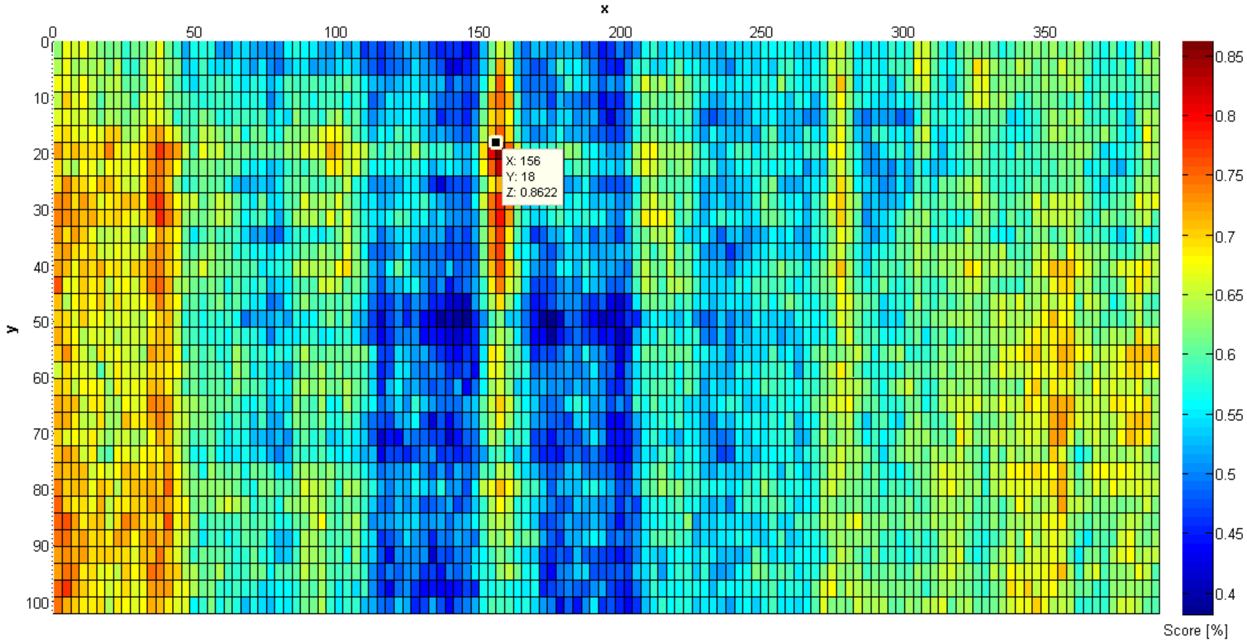


Figure 7.18: 2D plot with Polling Twice and No Linearization for Image Number 1

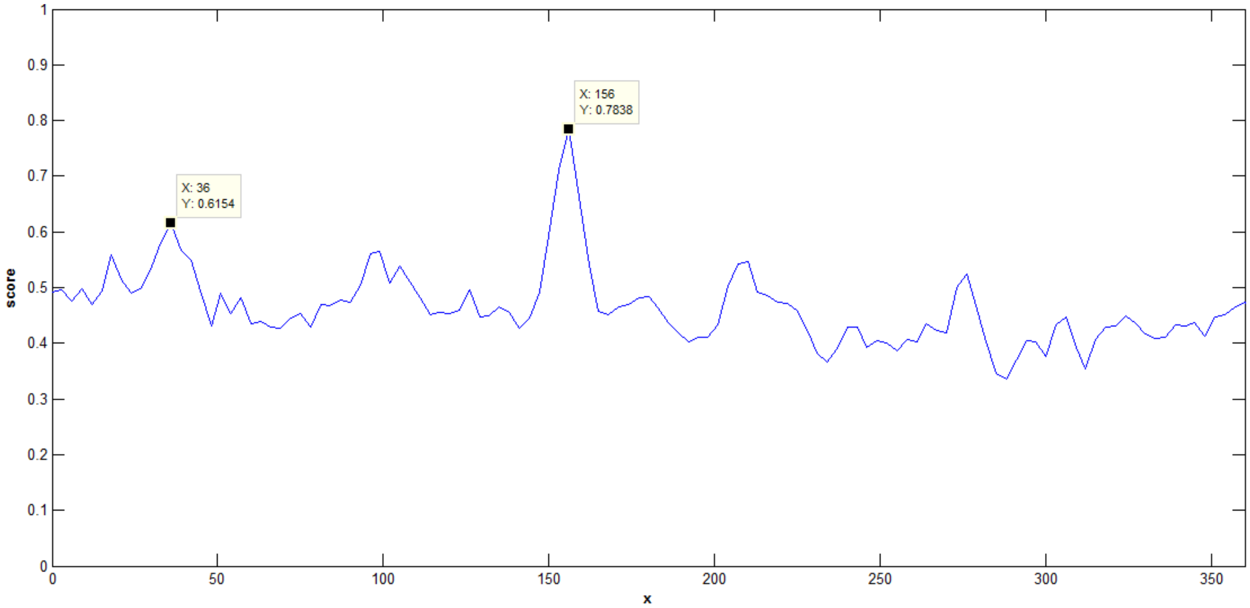


Figure 7.19: 1D Plot with Polling Twice and Linearization for Image Number 1

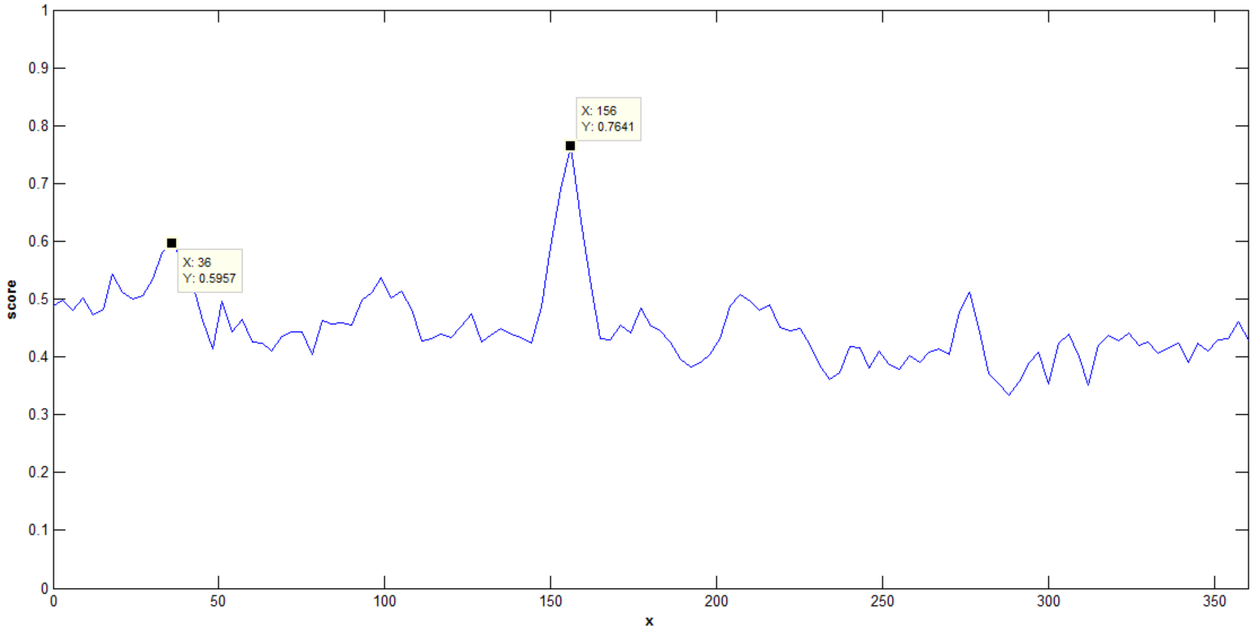


Figure 7.20: 1D Plot with Polling Once and Linearization for Image Number 1

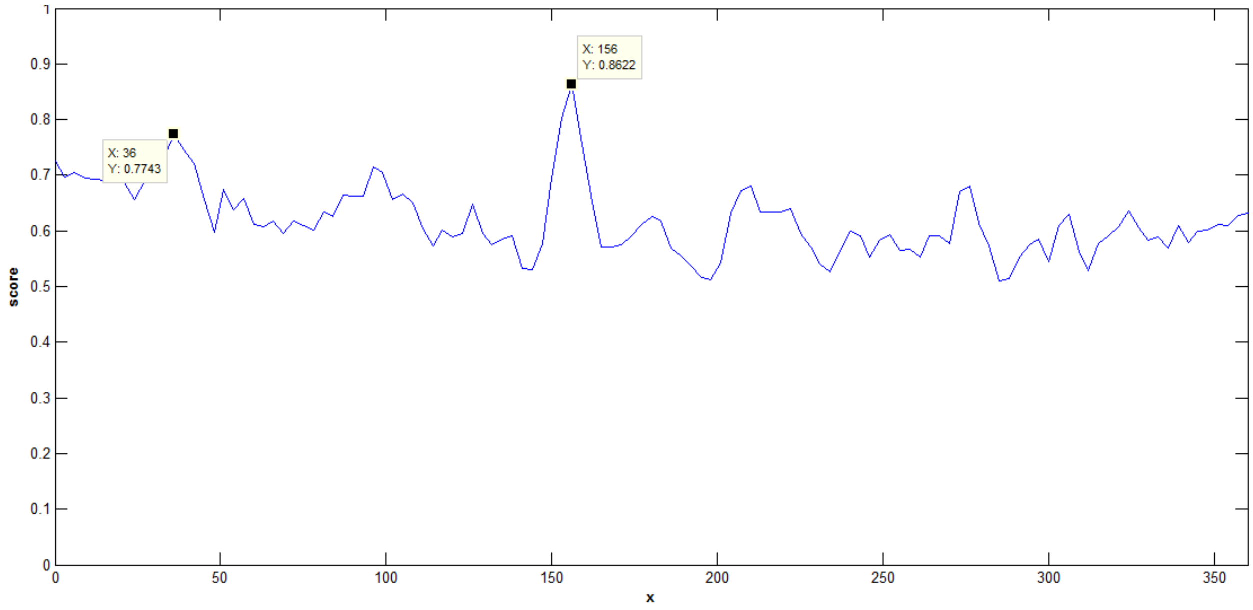


Figure 7.21: 1D plot with Polling Twice and No Linearization for Image Number 1

7.2.2 1D and 2D Plots for Test Image #2

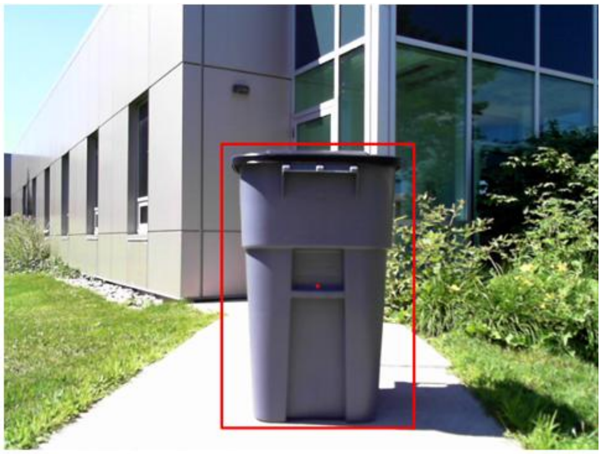


Figure 7.22: Test Image #2

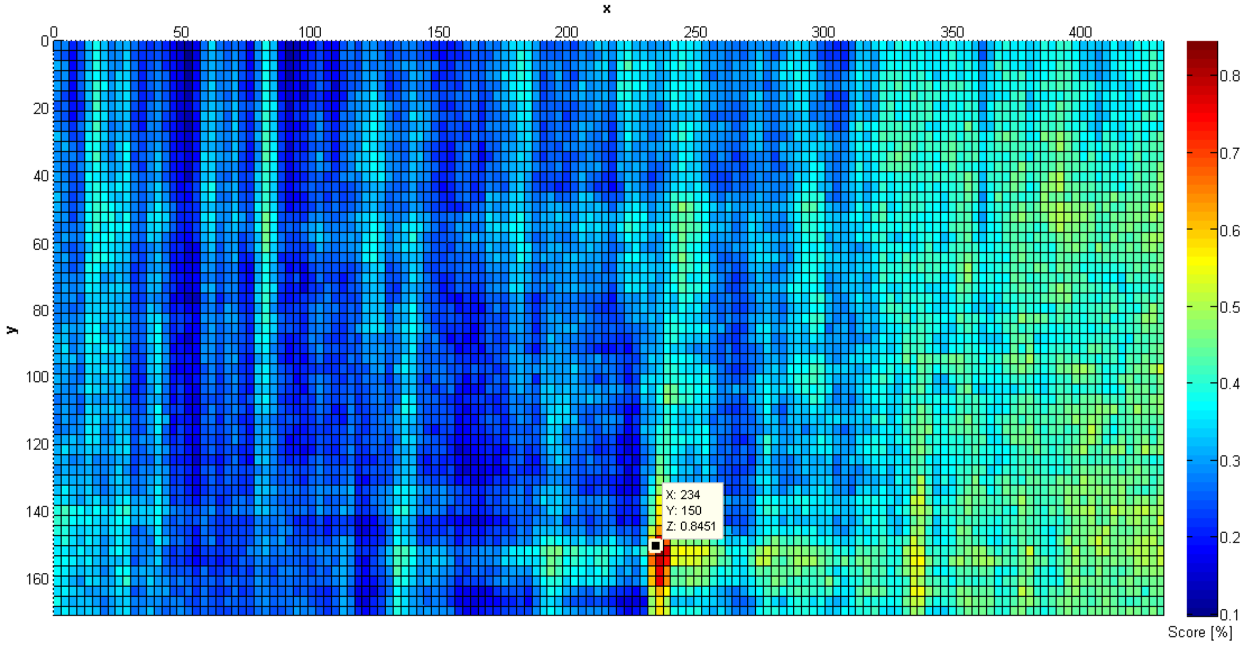


Figure 7.23: 2D Plot with Polling Twice and Linearization for Image Number 2

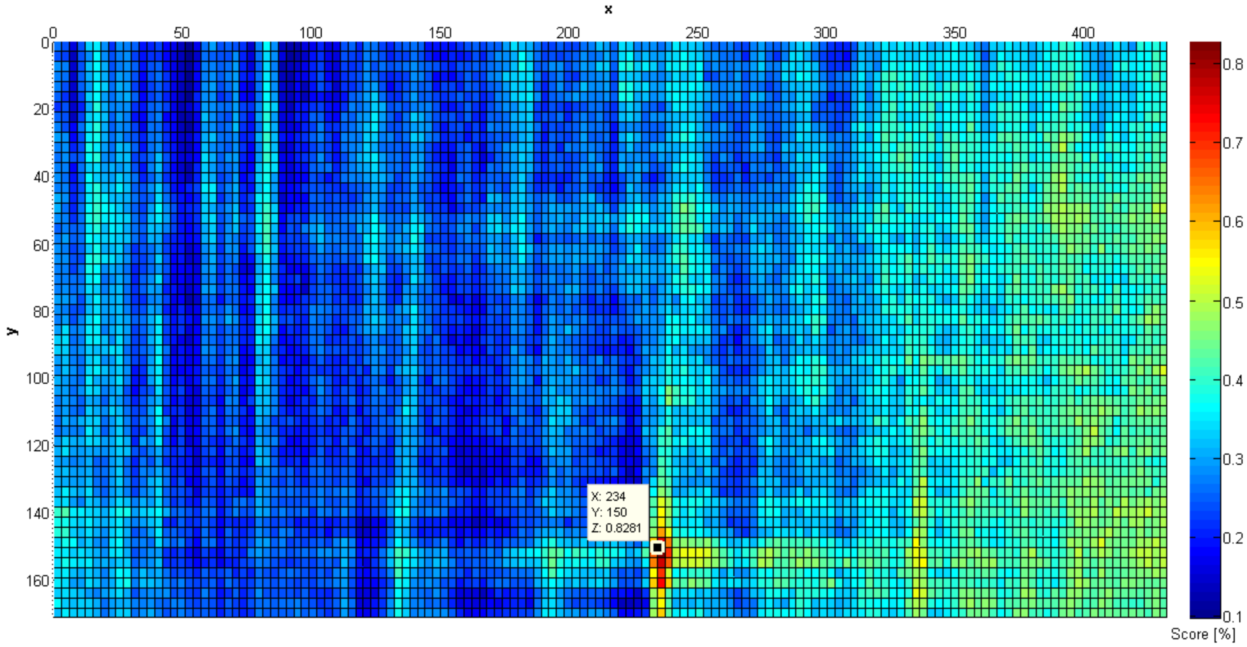


Figure 7.24: 2D Plot with Polling Once and Linearization for Image Number 2

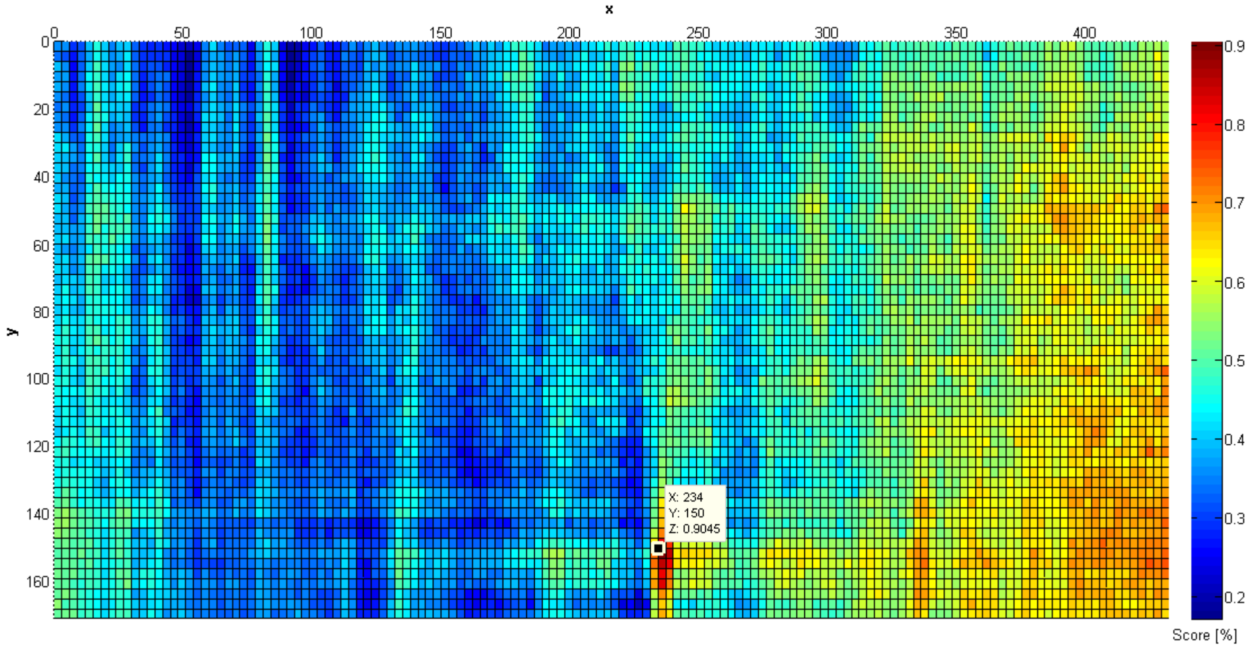


Figure 7.25: 2D Plot with Polling Twice and No Linearization for Image Number 2

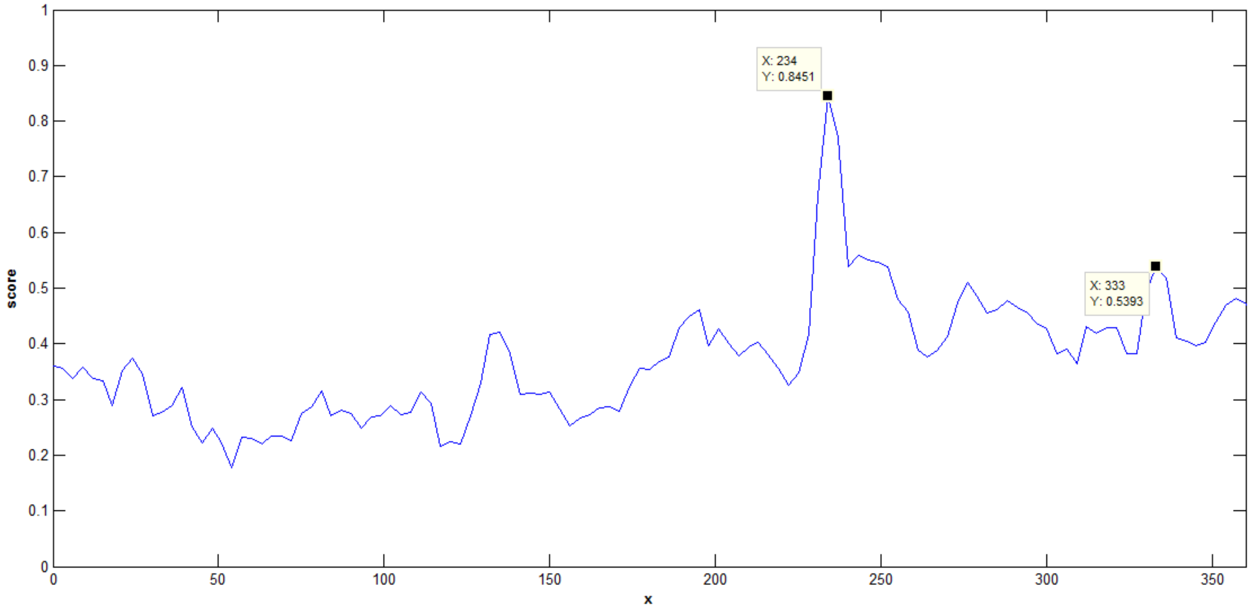


Figure 7.26: 1D Plot with Polling Twice and Linearization for Image Number 2

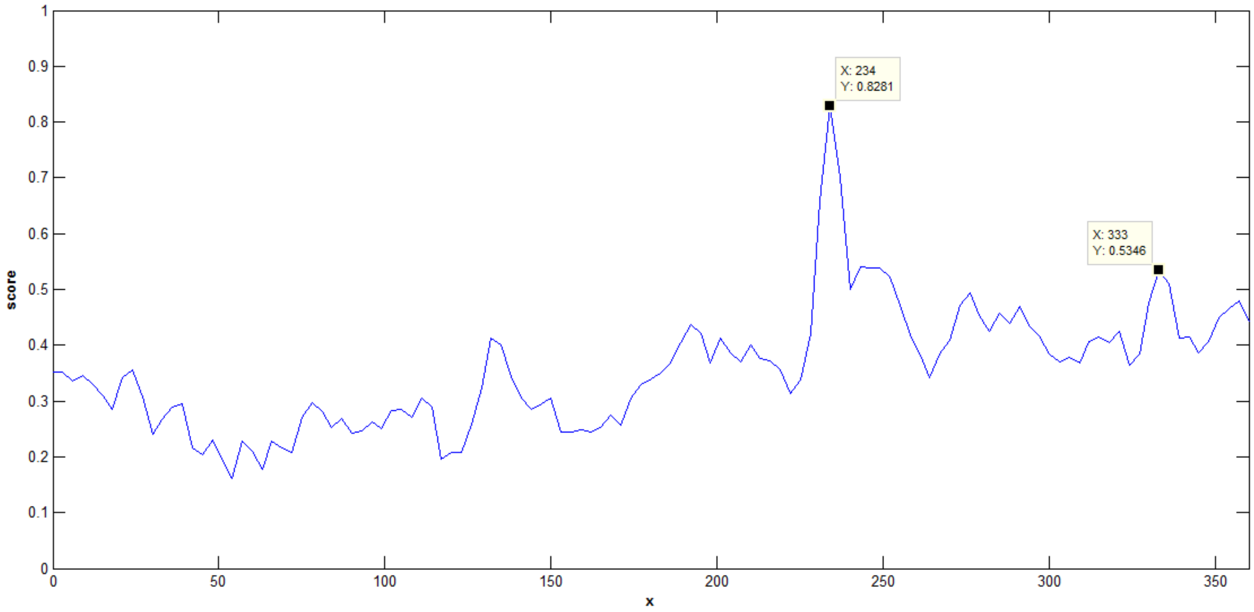


Figure 7.27: 1D Plot with Polling Once and Linearization for Image Number 2

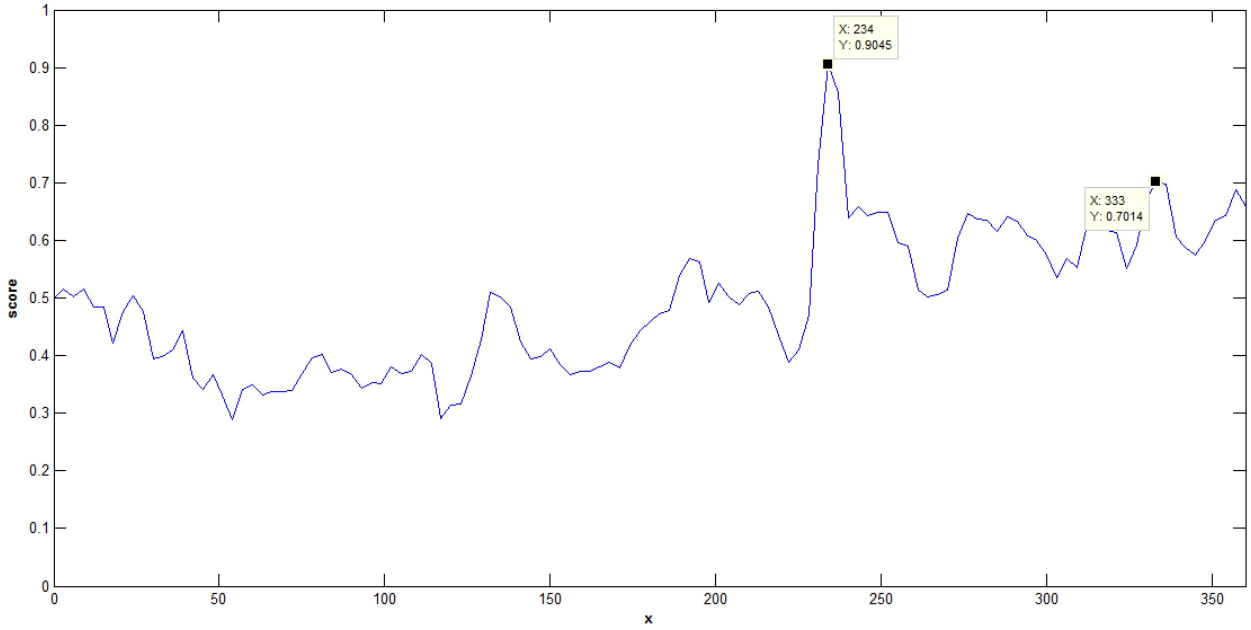


Figure 7.28: 1D Plot with Polling Twice and No Linearization for Image Number 2

7.2.3 1D and 2D Plots for Test Image #3

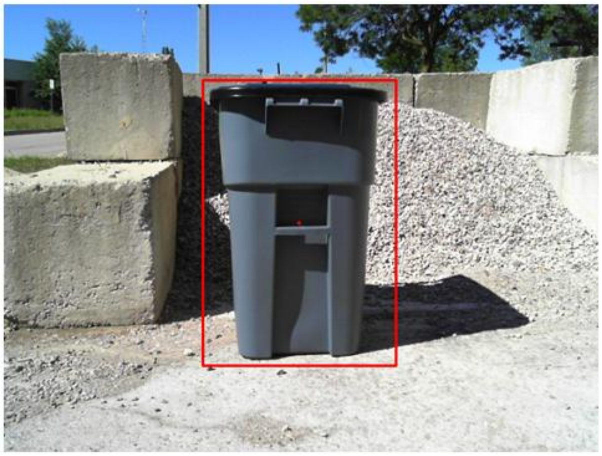


Figure 7.29: Test Image #3

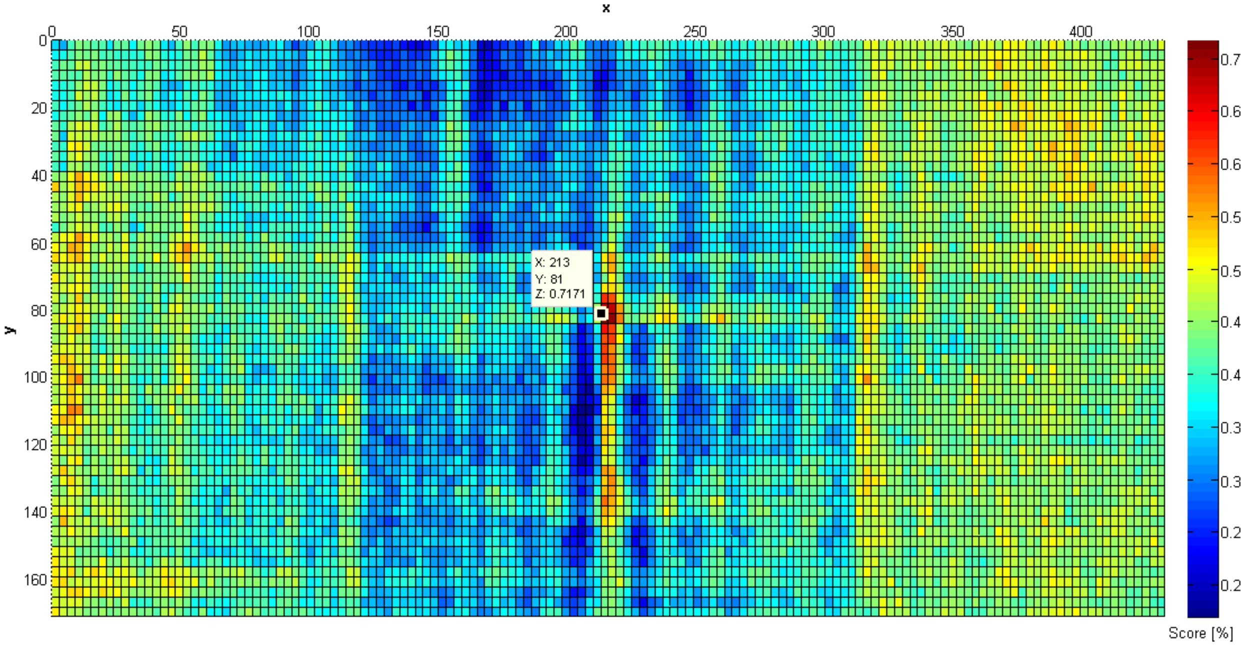


Figure 7.30: 2D Plot with Polling Twice and Linearization for Image Number 3

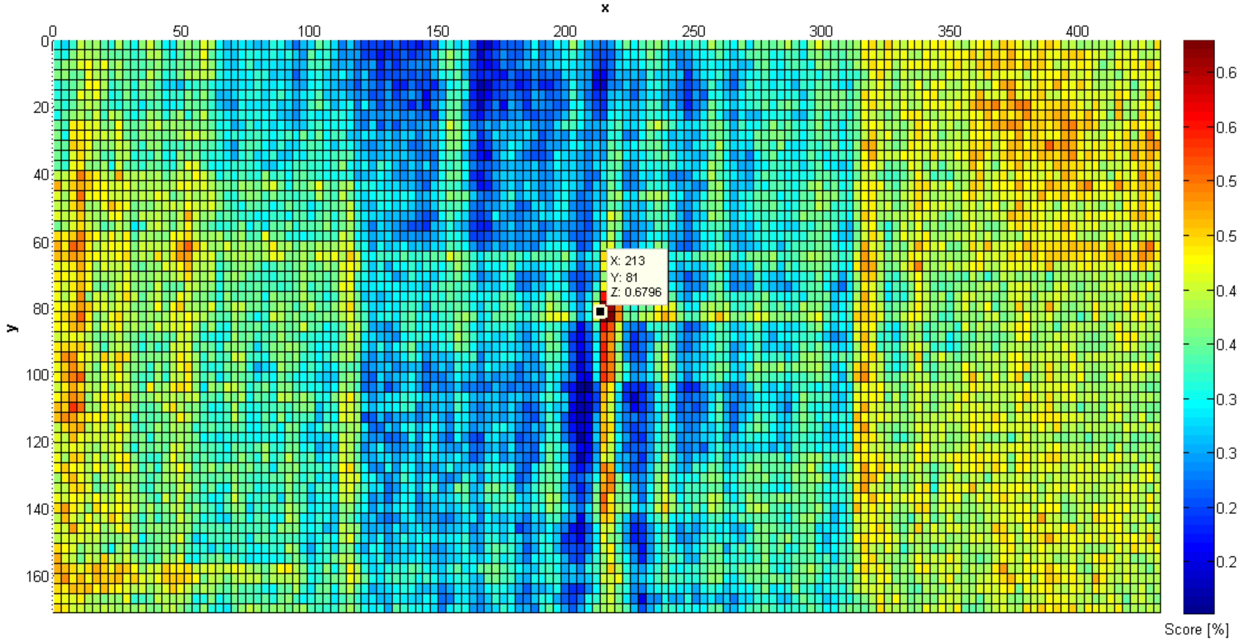


Figure 7.31: 2D Plot with Polling Once and Linearization for Image Number 3

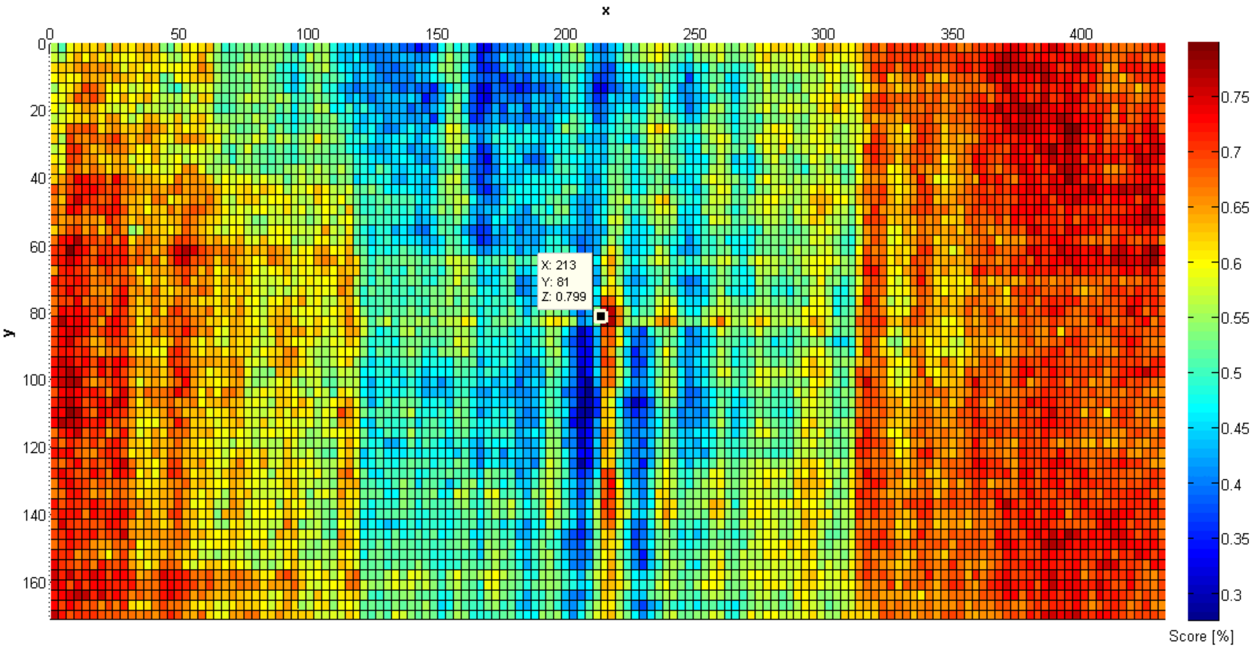


Figure 7.32: 2D Plot with Polling Twice and No Linearization for Image Number 3

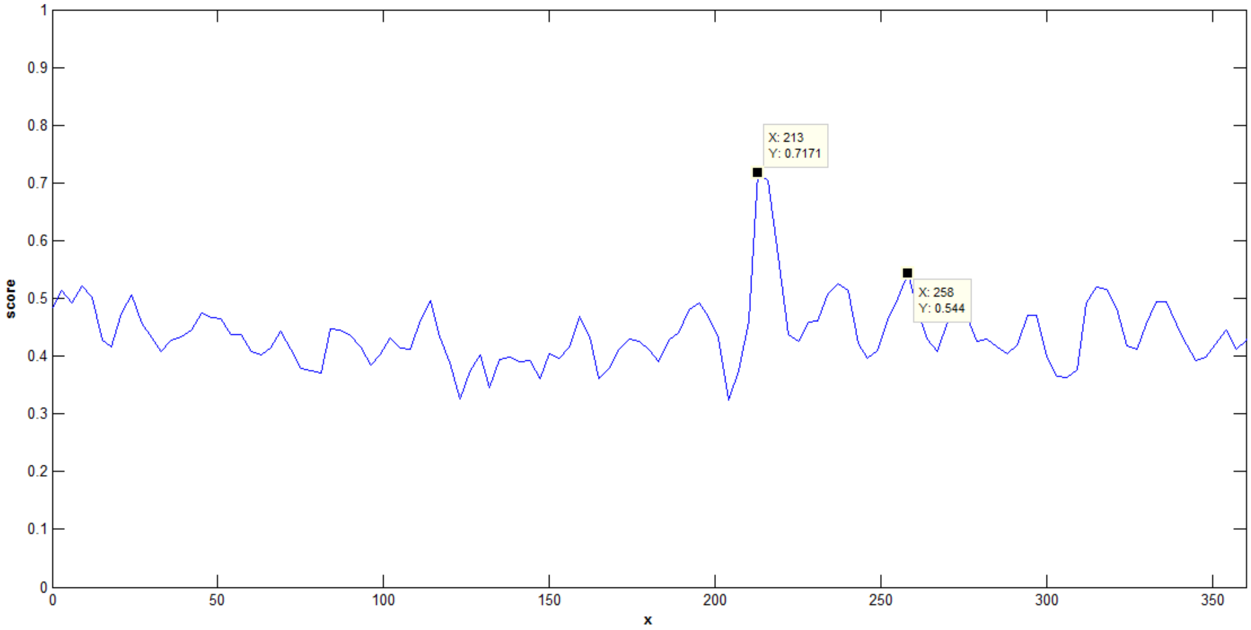


Figure 7.33: 1D Plot with Polling Twice and Linearization for Image Number 3

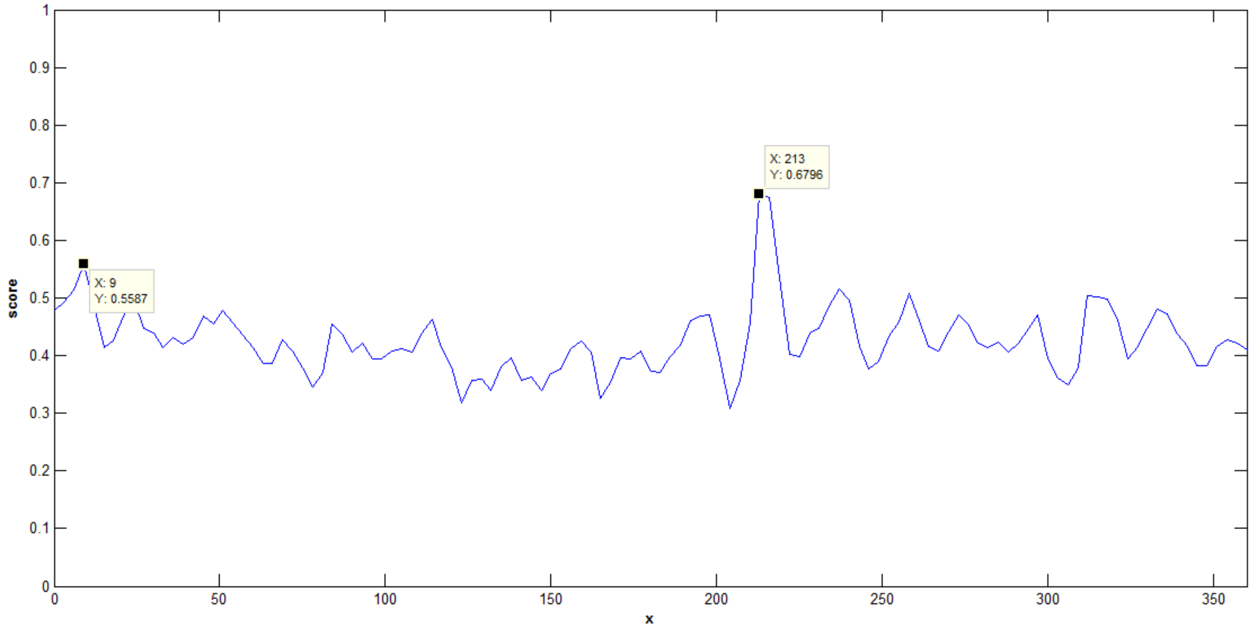


Figure 7.34: 1D Plot with Polling Once and Linearization for Image Number 3

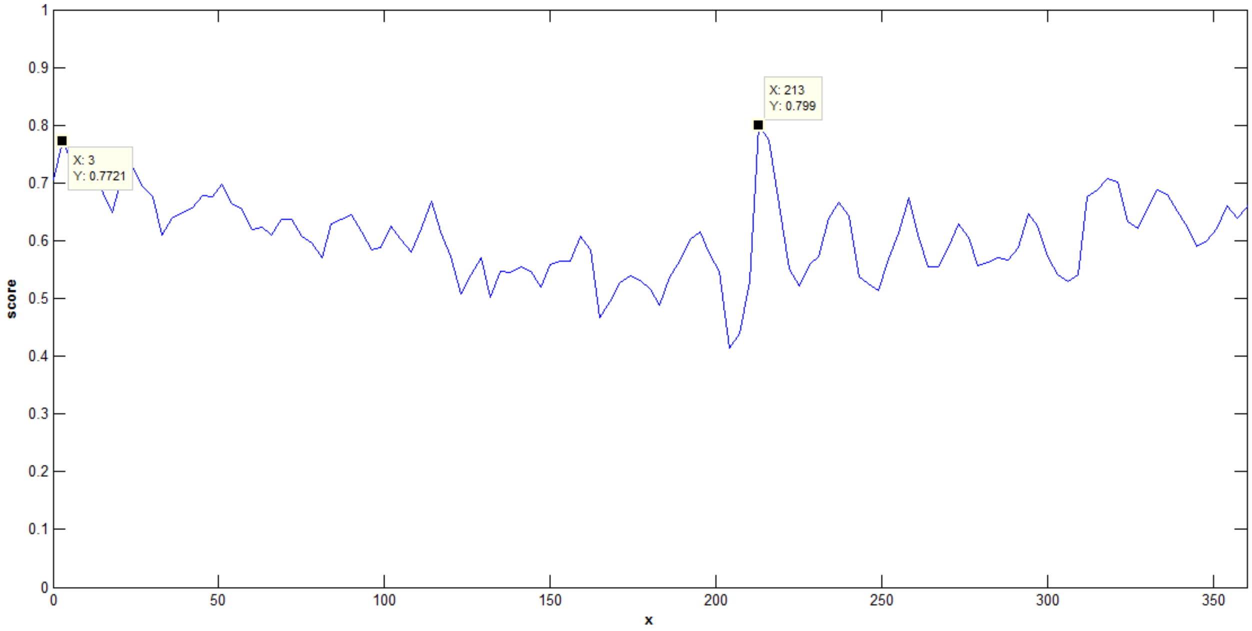


Figure 7.35: 1D Plot with Polling Twice and No Linearization for Image Number 3

7.2.4 1D and 2D Plots for Test Image #4

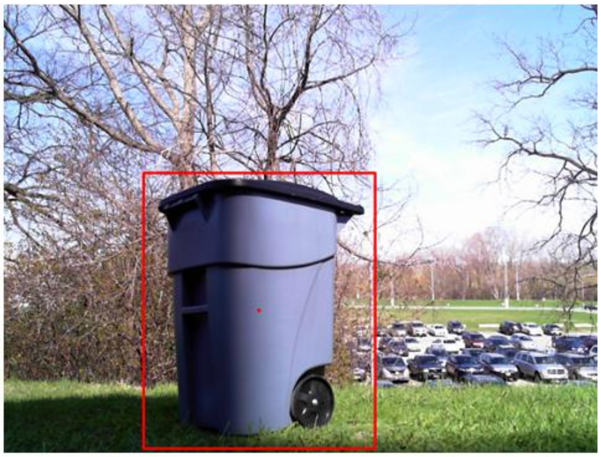


Figure 7.36: Test Image #4

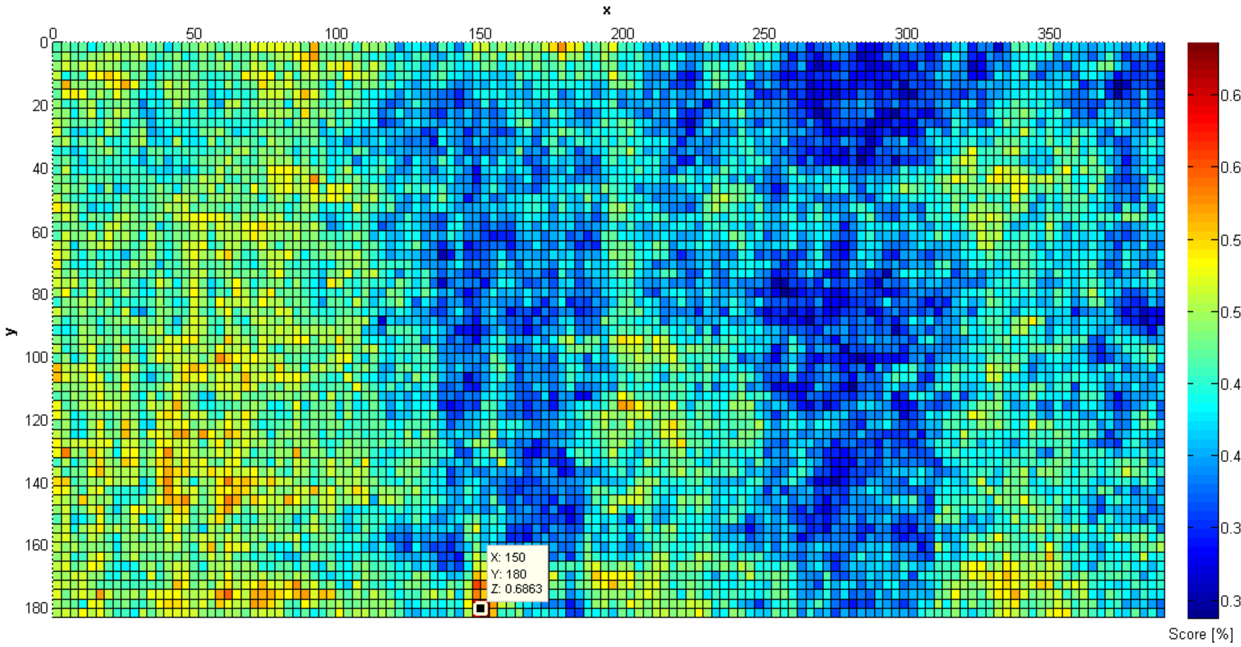


Figure 7.37: 2D Plot with Polling Twice and Linearization for Image Number 4

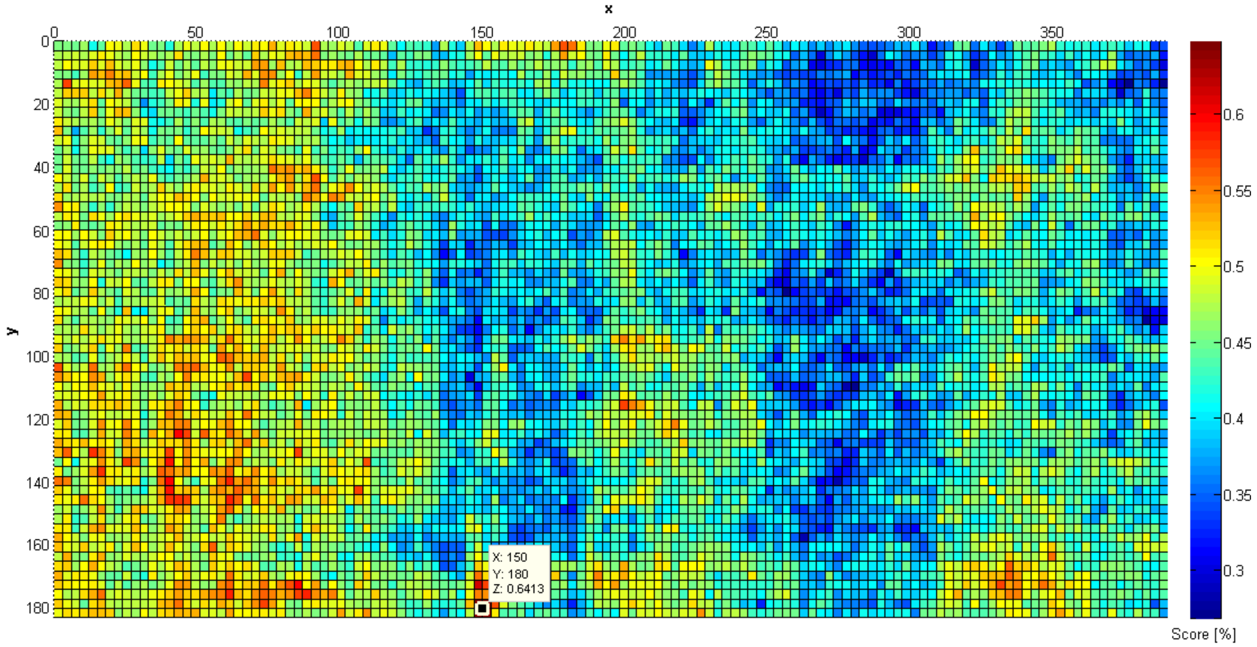


Figure 7.38: 2D Plot with Polling Once and Linearization for Image Number 4

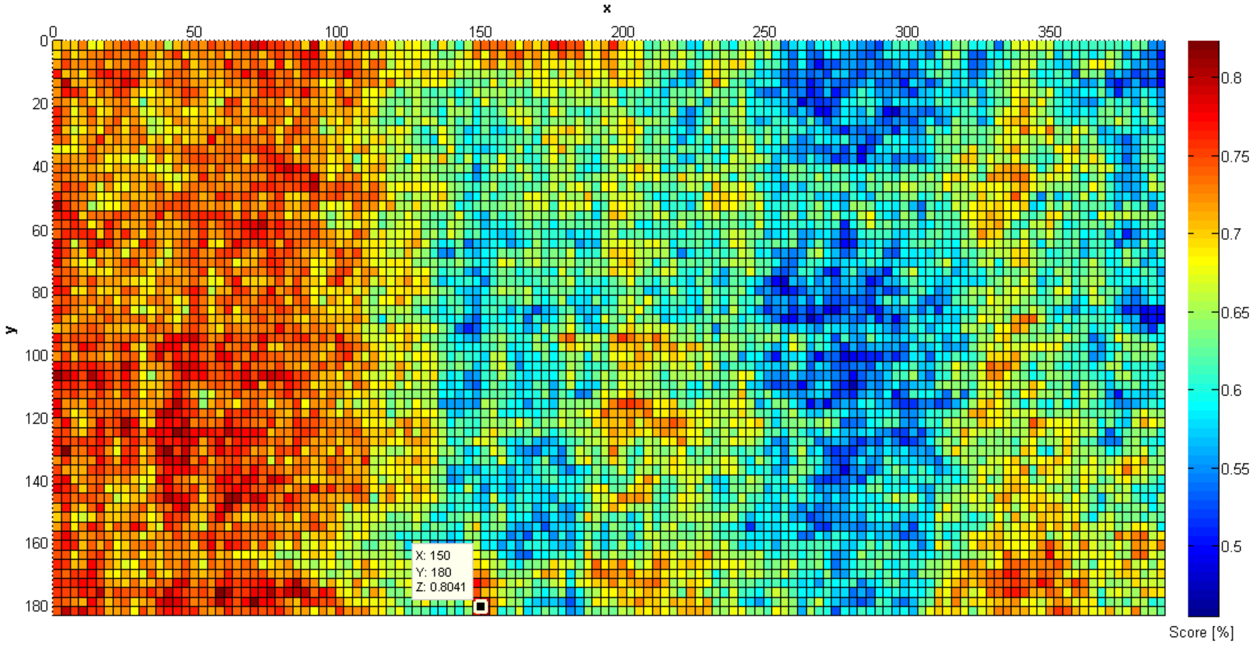


Figure 7.39: 2D Plot with Polling Twice and No Linearization for Image Number 4

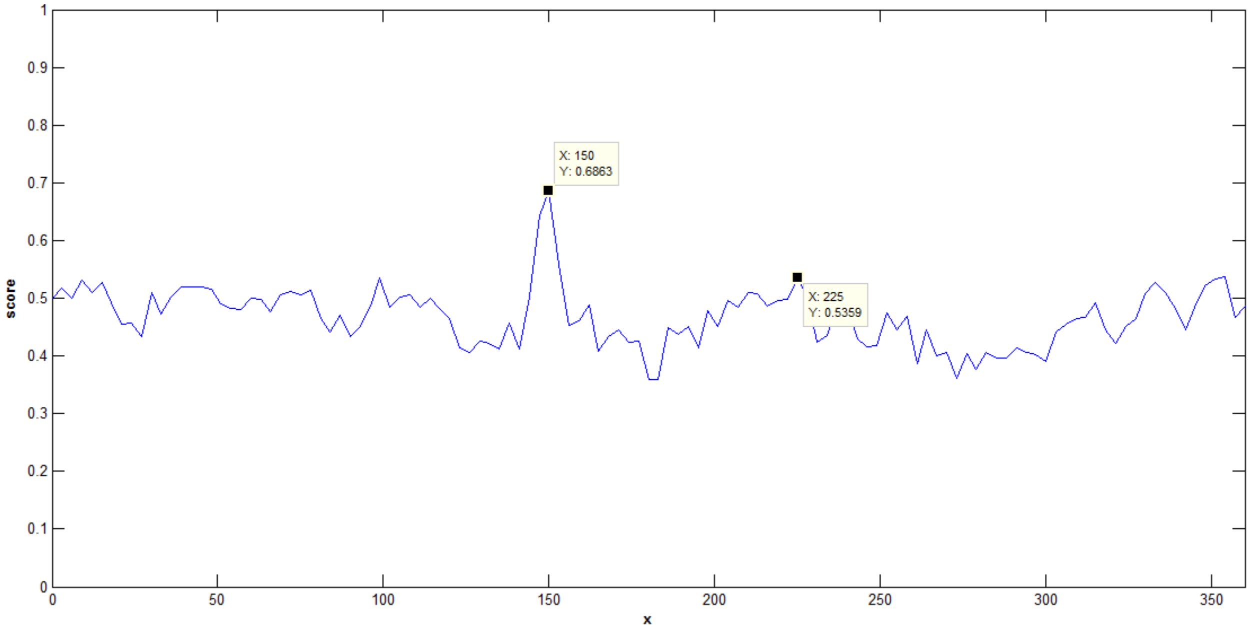


Figure 7.40: 1D Plot with Polling Twice and Linearization for Image Number 4

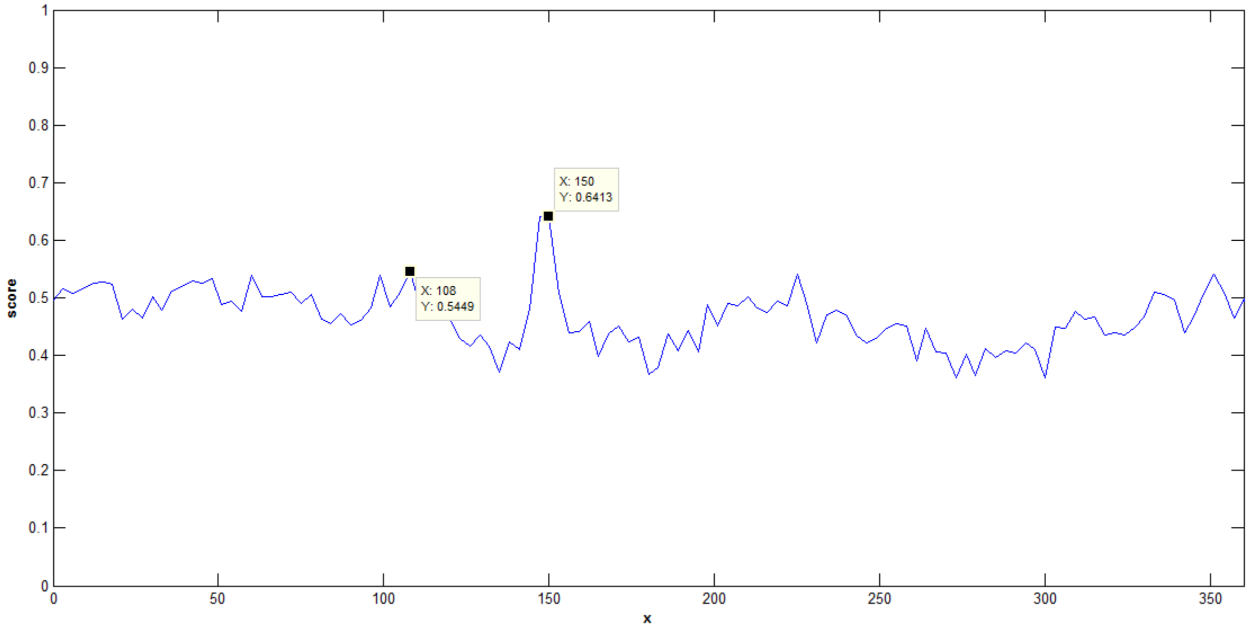


Figure 7.41: 1D Plot with Polling Once and Linearization for Image Number 4

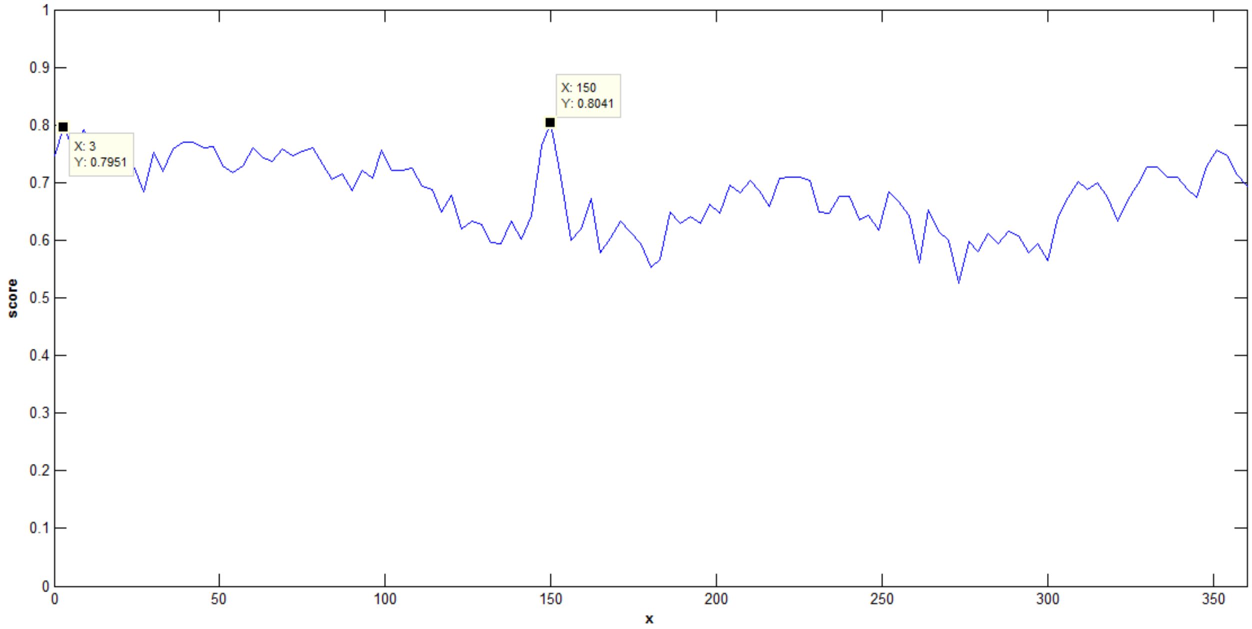


Figure 7.42: 1D Plot with Polling Twice and No Linearization for Image Number 4

References

[1] P. Shepherd, "Integrated Municipal Solid Waste Management: Six Case Studies of System Cost and Energy Use," National Renewable Energy Laboritory, Golden, Colorado, 1995.

[2] "Getting More for Less Improving Collection Efficiency," United States Environmental Protection Agency, 1999.

[3] D. Lowe, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision* , Corfu, Greece, 1999.

[4] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision,* vol. 20, no. 2, pp. 91-110, 2004.

[5] H. Bay, T. Tuytelaars and L. Van Gool, "SURF: Speeded up robust features," in *European Conference on Computer Vision*, Graz, Austria , 2006.

[6] H. Moravec, "Rover visual obstacle avoidance," in *International Joint Conference on Artifical Intelligence*, Vancouver, Canada, 1981.

[7] S. Harris and M. Stephens, "A combined corner and edge detector," in *Fourth Alvey Vision Conference*, Manchester, UK, 1988.

[8] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *European Conference on Computer Vision*, Copenhagen, Denmark, 2002.

[9] A. Baumberg, "Reliable feature matching across widely separated views," in *Conference of Computer Vision and Pattern Recognition*, Hilton Head Island, South Carolina, 2000.

[10] H. Moravec, "Rover visual obstacle aviodance," in *International Joint Conferences on Artificial Intelligence*, 1981.

[11] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 19, no. 5, pp. 530-534, 1997.

[12] Z. Zhang, R. Deriche, O. Faugeras and Q. Luong, "A robust technique for matching two uncalibrated images through recovery of the unknown epipolar geometry," *Artifical Intelligence*, pp. 87-119, 1995.

[13] C. Schmid, R. Mohr and C. Baukhag, "Evaluation of Interest Point Detectors," *International Journal of Computer Vision*, vol. 2, no. 32, pp. 151-172, 2000.

[14] R. Horaud, T. Skordas and F. Veillon, "Finding geometric and relational structures in an image," in *Proceedings of the 1st European Conference on Computer Vision*, Antibes, France, 1990.

[15] E. Shilat, M. Werman and Y. Gdalyahu, "Ridge's corner detection and correspondence," in *Conference on Computer Vision and Pattern Recognition*, Puerto Rico, USA, 2013.

[16] F. Mokhtarian and R. Suomela, "Robust image corner detection through curvature scale space," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 20, pp. 1376-1381, 1998.

[17] L. Florack, B. Ter Haar Romeny, J. Koenderink and M. Viergever, "General Intensity Transformation and Differential Invariants," *Mathematical Imaging and Vision*, vol. 4, no. 2, pp. 171-187, 1994.

[18] A. Witkin, "Scale-space filtering," in *International Joint Conference on Artifical Intelligence* , Karlsruhe, Germany, 1983.

[19] M. Brown and D. Lowe, "Invariant features from interest point groups," in *British Machine Vision Conference*, Cardiff, Wales, 2002.

[20] H. Zhen, L. Yewei and L. Jinjiang, "Image stitch algorithm based on SIFT and MVSC," in *International Conference on Fuzzy Systems and Knowledge Discovery*, 2010.

[21] M. Fiala and C. Shu, "3D model creation using self-identifying markers and SIFT keypoints," in *International Workshop On Haptic Audio Visual Environments and their Applications*, Ottawa, Canada, 2006.

[22] N. Dardas, C. Qing, Georganas, D. Nicolas and E. Petriu, "Hand gesture recognition using bag-of-features and multi-class support vector machine," in *IEEE International Symposium on Haptic Audio-Visual Environments and Games*, Phoenix, AZ, 2010.

[23] Y. Ding, B. Zhoa, Q. You and G. Chai, "Object retrival based on visual word pairs," in *IEEE International Conference on Image Processing*, Orlando, FL, 2012.

[24] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *International Conference on Computer Vision*, Vancouver, Canada, 2001.

[25] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Conference on Computer Vision and Pattern Recognition*, Washington, DC, 2004.

[26] M. Calonder, V. Lepetit, C. Strecha and P. Fua, "Brief: Binary robust independent elementary features," in *European Conference on Computer Vision*, Crete, Greece, 2010.

[27] F. Tombari, A. Franchi and L. Di Stefano, "BOLD Features to Detect Texture-less Objects," in *2013 IEEE International Conference on Computer Vision*, Sydney, NSW, 2013.

[28] H. Barrow, J. Tenenbaum, R. Bolles and H. Wolf, "Parametric Correspondence and chamfer matching: two new techniques for image matching," in *SRI International*, Menlo Park, Califorina, 1977.

[29] G. Borgefors, "Hierarchical chamfer matching: a parametric edge matching algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 10, no. 6, pp. 849-865, 1988.

[30] T. Newman and A. Jain, "Bidirectional template matching for 3D CAD-based inspection," in *1994 International Symposium on Electronic Imaging: Science and Technology*, International Society for Optics and Photonics, 1994, pp. 257-265.

[31] G. Borgerors, "On digital distance transforms in three dimensions," *Computer vision and image understanding,* vol. 64, no. 3, pp. 368-376, 1996.

[32] F. Jurie and M. Dhome, "Real time 3D template matching," in *International Conference on Computer Vision & Pattern Recognition*, Kauai, HI, USA , 2001.

[33] R. Osada, T. Funkhouser, B. Chazelle and D. Dobkin, "Matching 3D models with shape distributions," in *SMI 2001 International Conference on Shape Modeling and Applications*, Genova, 2001.

[34] N. Gupta, R. Gupta and A. Singh, "Object recognition using template matching," Available in: https://tmatch. googlecode. com/svnhistory/r38/trunk/report/report. pdf, 2008.

[35] E. David and O. Selfridge, "Eyes and ears for computers," *Proceedings of the IRE,* vol. 50, no. 5, pp. 1093-1101, 1962.

[36] A. Rosenfeld and G. Vanderbrug, "Coarse-fine template matching," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 7, no. 2, pp. 104-107, 1977.

[37] J. Lewis, "Fast template matching," in *Vision interface*, 1995, pp. 15-19.

[38] F. Jurie and M. Dhome, "Real Time Robust Template Matching," in *BMVC*, 2002, pp. 1-10.

[39] "Speed-up Template Matching through Integral Image based Weak Classifiers," *Journal of Pattern Recognition Research,* vol. 9, no. 1, pp. 1-12, 2014.

[40] J. Kim, H. Cho and S. Kim, "Pattern classification of solder joint images using a correlation neural network," *Engineering Applications of Artificial Intelligence,* vol. 9, no. 6, pp. 655-669, 1996.

[41] J. Gallegos, J. Villalobos, G. Carrillo and S. Cabrera, "Reduced-dimension and wavelet processing of SMD images for real-time inspection," in *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, San Antonio, TX, 1996.

[42] X. Cai, F. Kvasnik and R. Blore, "Wafer fault measurement by coherent optical processor," *Applied optics,* vol. 33, no. 20, pp. 4487-4496, 1994.

[43] H. Penz, I. Bajla, A. Vrabl, W. Krattenthaler and K. Mayer, "Fast real-time recognition and quality inspection of printed characters via point correlation," in *Photonics West 2001-Electronic Imaging*, International Society for Optics and Photonics, 2001, pp. 127-137.

[44] H. Yazdi and T. King, "Application of 'vision in the loop'for inspection of lace fabric," *Real-Time Imaging,* vol. 4, no. 5, pp. 317-332, 1998.

[45] C. Costa and M. Petrou, "Automatic registration of ceramic tiles for the purpose of fault detection," *Machine Vision and Applications,* vol. 11, no. 5, pp. 225-230, 2000.

[46] C. Olson and D. Huttenlocher, "Automatic target recognition by matching oriented edge pixels," *IEEE Transactions on Image Processing,* vol. 6, no. 1, pp. 103-113, 1997.

[47] C. Steger, "Occlusion, clutter, and illumination invariant object recognition," *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences,* vol. 34, no. 3/A, pp. 345-350, 2002.

[48] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua and V. Lepetit, "Gradient Response Maps for Real-Time Detection of Textureless Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 34, no. 5, pp. 876 - 888, 2012.

[49] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 2005.

[50] A. Hofhauser, C. Steger and N. Navab, "Edge-based template matching and tracking for perspectively distorted planar objects," in *Advances in Visual Computing,* Springer, 2008, pp. 35-44.

[51] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua and N. Navab, "Dominant orientation templates for real-time detection of texture-less objects," in *IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, 2010.

[52] N. Thanh, W. Li and P. Ogunbona, "An improved template matching method for object

detection," in *Asian Conference on Computer Vision*, Daejeon, Korea, 2012.

[53] B. Yao, G. Bradski and L. Fei-Fei, "A codebook-free and annotation-free approach for fine-grained image categorization," in *IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, 2012.

[54] E. Hsiao and M. Hebert, "Occlusion Reasoning for Object Detection under Arbitrary Viewpoint," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PP, no. 99, p. 1, 2012.

[55] E. Hsiao and M. Hebert, "Coherent Occlusion Reasoning for Instance Recognition," in *APR Asian Conference on Pattern Recognition*, Naha, 2013.

[56] E. Hsiao and M. Hebert, "Occlusion Reasoning for Object Detection under Arbitrary Viewpoint," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PP, no. 99, p. 1, 2014.

[57] C. Bor-Jeng, H. Cheng-Ming, T. Ting-En and F. Li-Chen, "Robust head and hands tracking with occlusion handling for human machine interaction," in *International Conference on International Conference on*, Vilamoura, 2012.

[58] C. Bor-Jeng, H. Cheng-Ming, T. Ting-En and F. Li-Chen, "Hands tracking with self-occlusion handling in cluttered environment," in *Asian Control Conference*, Istanbul, 2013.

[59] R. Rios-Cabrera and T. Tuytelaars, "Discriminatively Trained Templates for 3D Object Detection: A Real Time Scalable Approach," in *IEEE International Conference on Computer Vision*, Sydney, NSW, 2013.

[60] M. Ersen, S. Talay and H. Yalcin, "Extracting Spatial Relations Among Objects for Failure Detection," in *German Conference on Artificial Intelligence Visual and Spatial Cognition KIK*, Koblenz, Germany, 2013.

[61] S. Karapinar, S. Sariel-Talay, P. Yildiz and M. Ersen, "Learning Guided Planning for Robust Task Execution in Cognitive Robotics," in *AAAI Conference on Artificial*

Intelligence, Bellevue, Washington, 2013.

[62] P. Yildiz, S. Karapinar and S. Sariel-Talay, "Learning Guided Symbolic Planning for Cognitive Robots," in *International Conference on Robotics and Automation* , 2013.

[63] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige and N. Navab, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Asain Conference on Computer Vision*, 2013.

[64] Q. Zhu, M. Yeh, K. Cheng and S. Avidan, "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[65] P. Felzenszwalb, R. Girshick, D. McAllester and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, no. 9, pp. 1627-1645, 2010.

[66] P. Felzenszwalb, R. Girshick, D. McAllester and D. Ramanan, "Visual Object Detection with Deformable Part Models," *Communications of the ACM,* vol. 56, no. 9, pp. 97-105, 2013.

[67] T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan and J. Yagnik, "Fast, Accurate Detection of 100,000 Object Classes on a Single Machine," in *IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, 2013.

[68] T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," *International Journal of Computer Vision,* vol. 30, no. 2, pp. 117-156, 1998.

[69] A. Khashman, "Optimal scale edge detection utilizing noise within images," *J Syst Cybern Inform,* vol. 1, pp. 46-50, 2003.

[70] R. Koren and Y. Yitzhaky, "Automatic selection of edge detector parameters based on spatial and statistical measures," *Computer Vision and Image Understanding,* vol. 102, no. 2, pp. 204-213, 2006.

[71] J. Sivaswamy, "Multi-scale approach to salient contour extraction," in *Proceedings of the International Conference on Cognition and Recognition (ICCR05)*, Mysore, 2005.

[72] K. Liang, T. Tjahjadi and Y. Yang, "Bounded diffusion for multiscale edge detection using regularized cubic B-spline fitting," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 29, no. 2, pp. 291-297, 1999.

[73] B. Tremblais and B. Augereau, "A fast multi-scale edge detection algorithm," *Pattern Recognition Letters,* vol. 25, pp. 603-618, 2004.

[74] B. Jiang and Z. Rahman, "Multi-scale edge detection with local noise estimate," in *SPIE Optical Engineering+ Applications,* San Diego, 2010.

[75] R. Medina-Carnicer, F. Madrid-Cuevas, A. Carmona-Poyato and R. Munoz-Salinas, "On candidates selection for hysteresis thresholds in edge detection," *Pattern Recognition,* vol. 42, no. 7, pp. 1284-1296, 2009.

[76] D. Sen and S. Pal, "Gradient histogram: Thresholding in a region of interest for edge detection," *Image and Vision Computing,* vol. 28, no. 4, pp. 677-695, 2010.

[77] R. Medina-Carnicer and F. Madrid-Cuevas, "Unimodal thresholding for edge detection," *Pattern Recognition,* vol. 41, no. 7, pp. 2337-2346, 2008.

[78] R. Medina-Carnicer, R. Munoz-Salinas, E. Yeguas-Bolivar and L. Diaz-Mas, "A novel method to look for the hysteresis thresholds for the Canny edge detector," *Pattern Recognition,* vol. 44, no. 6, pp. 1201-1211, 2011.

[79] Z. Qu, Y. Gao, P. Wang, P. Wang, X. Tan and Z. Shen, "Contour detection improved by frequency domain filtering of gradient image," *Science China Information Sciences,* vol. 57, no. 1, pp. 1-11, 2014.

[80] C. Grigorescu, N. Petkov and M. Westenberg, "Contour and boundary detection improved by surround suppression of texture edges," *Image and Vision Computing,* vol. 22, no. 8, pp. 609-622, 2004.

[81] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence*, no. 6, pp. 679-698, 1986.

[82] Z. Qu, P. Wang, Y. Gao, P. Wang and Z. Shen, "Contour detection based on SUSAN principle and surround suppression," in *International Conference on Image Processing*, Hong Kong, 2010.

[83] S. Smith and J. Brady, "SUSAN—a new approach to low level image processing," *International journal of computer vision*, vol. 23, no. 1, pp. 45-78, 1997.

[84] D. Harrington, L. Boxt and P. Murray, "Digital subtraction angiography: overview of technical principles," *American Journal of Roentgenology*, vol. 139, no. 4, pp. 781-786, 1982.

[85] "Hands tracking with self-occlusion handling in cluttered environment," in *Asian Control Conference* , Istanbul, 2013.

CV - Justin Szoke-Sieswerda

EDUCATION

- Bachelor of Engineering Science **April 2012**
- Electrical Engineering (Graduated with Distinction)
- *University of Western Ontario, London, ON*

AWARDS

- Western Engineering Summer Research Award **2012**
- NSERC Undergraduate Student Research Award **2011**
- UWO Continuing Admission Scholarship **2008 – 2012**
- Queen Elisabeth Aiming for the Top Scholarship **2008 – 2012**

RESEARCH EXPERIENCE

Graduate Research Student (Masters Degree) **2012 - 2014**
Supervisor: Dr. McIsaac, University of Western Ontario, London, ON

Undergraduate Research Assistant **2011 – 2012**
Supervisor: Dr. McIsaac, University of Western Ontario, London, ON

TEACHING EXPERIENCE

Teaching Assistant

ES1036: Programming Fundamentals for Engineers
Instructor: Dr. Rahman, University of Western Ontario, London, ON **Summer 2014**

MSE 2202: Introduction to Mechatronic Design
Instructor: Dr. Naish, University of Western Ontario, London, ON **Winter 2013**

MSE 2202: Introduction to Mechatronic Design
Instructor: Dr. McIsaac, University of Western Ontario, London, ON **Winter 2012**

ES1036: Programming Fundamentals for Engineers
Instructor: Dr. Rahman, University of Western Ontario, London, ON **Fall 2012**

Course Instructor

Mechatronics Summer Academy Instructor **Summer 2014**
Summer course for high school students at University of Western Ontario

Mechatronics Summer Academy Instructor **Summer 2013**
Summer course for high school students at University of Western Ontario

Guest Lectures

ECE 2277: Digital Logic Systems
Lectured on: Multi-output system optimization and finite state machines **Fall 2013**