OBJECT RECOGNITION VIA MULTI-VIEW INSPECTION USING SATURATION-WEIGHTED DISTRIBUTIVE HUE HISTOGRAMS AND DEPTH INFORMATION

by

Jamie Westell
B.A.Sc., Simon Fraser University, 2008

A Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Applied Science
in the School
of
Engineering Science

SIMON FRASER UNIVERSITY
Summer, 2010

APPROVAL

Name:	**Jamie Westell**
Degree:	**Master of Applied Science**
Title of Thesis:	**Object Recognition via Multi-view Inspection Using Saturation-Weighted Distributive Hue Histograms and Depth Information**

Examining Committee:

Chair:

Dr. Rick Hobson P. Eng
Professor, School of Engineering Science

Dr. Parvaneh Saeedi,
Senior Supervisor
Assistant Professor, School of Engineering Science

Dr. Ivan Bajic,
Supervisor
Assistant Professor, School of Engineering Science

Dr. Kamal Gupta,
Examiner
Professor, School of Engineering Science

Date Approved: July 30, 2010

SFU SIMON FRASER UNIVERSITY
LIBRARY

Declaration of Partial Copyright Licence

Abstract

A computer vision algorithm is presented for the detection and localization of objects within unknown environments. To search for a specific object within an image, information about the object's appearance is first extracted from database images of the object. To effectively represent the appearance of the object, the Saturation-Weighted Distributive Hue Histogram is presented which encapsulates the intrinsic color information of the object as well as the spatial arrangement of colors within the object's boundaries. These histograms are then searched for in the scene image of the environment. The use of depth information allows searching for objects at depth variant planes in order to achieve scale invariance and to obtain the 3D coordinates of match candidates. A mobile robot platform is employed to move to various locations within the environment and to inspect matching regions by capturing and processing further images.

Acknowledgments

I would like to thank my supervisor, Dr. Parvaneh Saeedi, for providing me the opportunity to explore so many new ideas and technologies throughout my graduate studies. Your dedication to providing me with feedback, guidance, and support have made working on this project an amazing experience. Also, thank you to my examining committee, Dr. Rick Hobson, Dr. Ivan Bajic, and Dr. Kamal Gupta, for taking time out of their busy schedules in order to review my work. I would also like to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada for providing financial support for my research in the form of an Alexander Graham Bell Canadian Graduate Scholarship. Lastly, I would like to thank my family for all of their encouragement and support throughout my academic career. I could not have done this without you.

Contents

List of Figures

Chapter 1

Introduction

The goal of the work presented in this thesis is to automatically recognize and localize a specific object within an unknown environment using image processing techniques on images taken of the environment. For humans to accomplish this same task, information about the appearance of the specific object is searched for in the surrounding environment by comparing this information to appearance information of the surrounding objects. Similarly, this work presents a way for appearance information to be automatically extracted from a digital image of a specific object and used to locate instances of the object in additional images. Color information is used to represent the appearance of the object being searched for and depth information within the environment is used to effectively search the environment.

Returning to the analogy of a human searching for an object within an environment, if the human believes that a certain object within the environment is in fact the target object, further confidence is gained in this hypothesis by inspecting the object from different viewpoints. Similarly, this work presents a technique which may be used to integrate image matching results from images taken from multiple viewpoints within an environment.

The problem of automatically recognizing and locating objects within an unknown environment is a valid one as many applications arise from a viable solution. By equipping such a system on a navigational mobile robot, the complete system may respond to commands by autonomously searching an environment for a given object and either retrieving the object or simply recording the acquired coordinates of the object's location. Such complete systems are beneficial for many purposes including military missions requiring the dangerous exploration of inaccessible regions and also assisting disabled or elderly people with mobility issues.

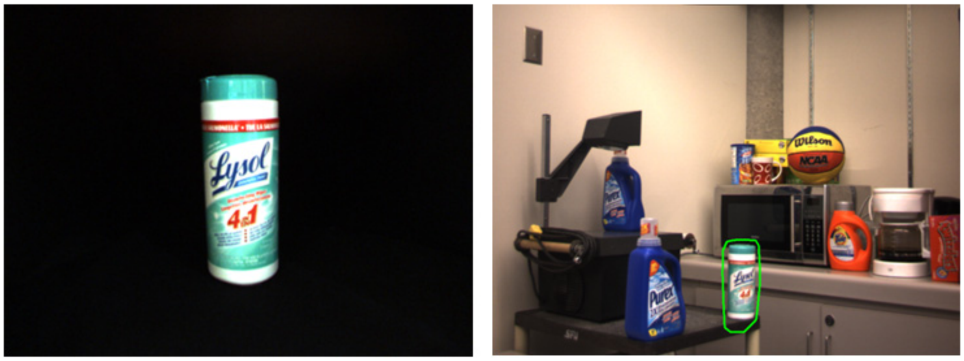

Figure 1.1: Recognizing and Locating an Object. Appearance information extracted from the model image of the object (*left*) is used to recognize and locate an instance of the object in a scene image (*right*). The located region is outlined in green.

This thesis provides a complete solution to the object recognition problem that is capable of locating known objects within images of unknown environments and also integrating matching results from images taken from multiple viewpoints within an environment. Depth information used in this process allows for the exact 3D coordinates of the matching result to be obtained. Figure 1.1 shows an example in which the system recognizes and locates an instance of a model object within a scene image. Appearance information extracted from the model image of the object on the left is used to recognize and locate an instance of the object in a scene image on the right. The located region is automatically outlined in green.

1.1 Contributions

In this work, three significant contributions towards the goal of object recognition will be presented.

1. The development of a novel object recognition algorithm capable of locating known 3D objects despite common challenges such as occlusion, scale change, illumination changes, and shading.
2. The development of the Saturation-Weighted Distributive Hue Histogram local feature descriptor which is robust to variations such as shading, blurring, specularities, and illumination changes.
3. The development of a technique used to integrate object recognition results from

multiple viewpoints using depth information from a stereo imaging system.

1.2 Thesis Organization

The remainder of this thesis is divided into five chapters. Chapter 2 presents the state of the art in object recognition research and presents several past works in detail. Chapter 3 defines the object recognition problem while Chapter 4 presents a detailed explanation of the solution proposed in this work. Experimental results are presented in Chapter 5 followed by a conclusions from this work as well as possible future research directions in Chapter 6.

Chapter 2

The State of the Art in Object Recognition

The task of recognizing a 3D object in a 2D image has a long history in the field of computer vision. Early work on this task focused on the use of known geometric models of objects and the recognition of the 2D projections of such models. Due to inherent challenges with the use of geometric models, recent work in this area has focused on the matching of local features extracted from the model and scene images. The incorporation of color information has also been used to improve the performance of the object recognition algorithms; however, the problem of color constancy under varying illumination provides an additional challenge. More recently, object recognition algorithms have been implemented on mobile robot platforms so that a target object may be actively searched for in an unknown environment. Using images captured from multiple viewpoints within an environment, the object recognition algorithm may check for match consistency across multiple views.

The following sections will explore the previous developments in the field of object recognition. The use of geometric models is first presented followed by the move towards matching local features. The incorporation of color and its associated challenges is also presented followed by recent work on the use of mobile robot platforms to capture images from multiple views and locate known objects within unknown environments.

2.1 Object Recognition using Geometric Models

Geometric characteristics extracted from object models can be used to predict the appearance of an object at any viewpoint. By calculating the projection of these models onto a specified plane, the expected appearance of the object is known. The assumption used in this approach is that such geometric models are obtainable and accurate. With this assumption, several techniques were developed to recognize 3D objects in 2D images.

One of the first techniques to incorporate geometric models into an object recognition system was that of Roberts [34] in 1965. In this work, known as Blocks World, objects were represented as composite structures of polyhedral components, or blocks. A generic library of polyhedra was established and the projections of each polyhedral into perspective images was analyzed. Edge detector and line fitting algorithms were developed to extract lines and vertices from the object images. By matching these features to the projection images of the library polyhedra, recognition of the composite polyhedral components was achieved. Inherently, the algorithm could not accommodate curved surfaces or complex shapes. Regardless, the approach was pursued by several research groups [19, 8, 29].

The limitations of the Blocks World approach were addressed in 1977 by Nevatia and Binford [32]. Here, objects were represented by a composite structure of cylinders. This method, known as Generalized Cylinders, exploited the fact that curved surfaces may be expressed as a sweep of a cross section along a curved axis [5, 2]. A 3D object was segmented into simpler sub-parts, each described by a generalized cylinder description. Recognition was then achieved by matching such descriptions to those of known geometric models.

The problem of matching a 3D model with a 2D image of an object has also been approached using the method of Viewpoint Consistency [21, 41]. In this approach, features from the image, such as edges or corners, are matched to features located on the 3D model. For each match, a transformation is derived to align a projection of the 3D model with the image and the expected locations of the model features are examined. Projections that show a large number of consistent features indicate a positive match between the scene object and the geometric model.

Huttenlocher and Ullman [21] implemented a recognition algorithm using the Viewpoint Consistency approach in an exhaustive manner. Each triplet of features in the scene were matched to the model features and the resulting projection of the model were compared to the object in the scene. Projections with a significant number of consistent features indicate a positive match.

Thompson and Mundy [41] took the same approach to matching images to 3D models; however, a generalized Hough transform was used to vote for the correct affine transformation that projects the model to the 2D image [3, 39]. As well, features composed of vertices of the intersections of two line segments were used to match between the scene and the model in order to reduce complexity.

The task of matching complete geometric models of objects to 2D images faces three significant challenges; occlusion of the object, background clutter, and low contrast regions at edges. Occlusion prevents the matching of portions of the model, background clutter leads to matching features not belonging to the correct object, and low contrast edges prevent the detection of features in the scene image. The use of geometric models in object recognition relies on the extraction and matching of lines, contours, and surfaces, each of which are extremely sensitive to the problems of occlusion, background clutter, and low contrast regions. In order to overcome these challenges, new directions using local feature matching were developed.

2.2 Object Recognition using Local Invariant Features

In the last decade, the use of local invariant features has been utilized to match local regions of objects between model and scene images. Three steps are typically carried out in this approach: determining where in the image to extract features from, describing the features and surrounding regions with a unique descriptor, and matching the local features detected in the image to features extracted from an object database. A final match verification stage is also commonly implemented to verify that the geometric relations between the image locations of the extracted features are consistent with the relative locations of the features extracted from the database images.

The use of local invariant features was influenced greatly by Schmid and Mohr [36] but perhaps the most celebrated technique using this approach is the SIFT technique by Lowe [27, 28]. SIFT, or Scale Invariant Feature Transform, is a technique in which feature points are detected in scale space using a Difference-of-Gaussian operator. The use of scale space allows features extracted at different scales to be matched successfully, thus providing scale invariance. Once the feature points are detected, a SIFT descriptor is used to describe the local region. The SIFT descriptor uses gradient information in the local region to form a unique description which is invariant to viewpoint and scale variations. The discriminative nature of the SIFT descriptor along with the invariance to scale changes, rotations, and changes in object pose allowed objects in novel images to be accurately matched to database images of objects.

The use of local invariant features in object recognition often leads to points which are falsely matched to the database along with many points that are accurately matched. To successfully recognize an object in an image, the false matches must be eliminated so that only true matches are found. A common technique in verifying local feature matches is to analyze the geometric relations of features detected in the image and ensure that they are consistent with the geometric relations of the matching points found in the database images of the objects. Within the SIFT object recognition system, Lowe analyzes semi-local regions in the scene (clusters of detected features) to ensure that the object location, orientation, and scale agree with the matched features in the database image. This step is shown to correctly filter the correct matches from the full set of matches.

With the success of the SIFT technique, a framework was established for object recognition. Local features in images must first be located, unique descriptions of these features must then be generated, and finally, these feature descriptors must be matched to a database of descriptors extracted from model images. In the years after the development of the SIFT technique, several research groups focused their attention on developing local feature descriptors to accurately match regions within images [4, 35, 25, 23].

In [4], Belongie et al. present a feature descriptor known as the Shape Context. Once a series of feature points are located, each point is described by the distribution of the relative positions of all other points. A log-polar histogram is used to place more significance on the points closer to the feature point. This technique is used primarily for character and shape recognition; however, extensions to object recognition are also presented.

Schaffalitzky and Zisserman developed a local feature descriptor using complex filters [35]. In this approach, a bank of linear filters are applied to the local region and the filter response is used as the feature descriptor. A total of sixteen linear filters, similar to Gaussian derivative filters, are applied to generate each descriptor. The presentation of this approach demonstrated the ability to locate correspondences between multiple images of the same object from different viewpoints in order establish the geometrical relationship between the viewpoints.

A feature descriptor known as Spin Images was developed by Lazebnik et al. [25] in order to recognize and classify different textures. Here, a 2D histogram was generated encoding the distribution of image brightness values in the neighborhood of a central feature point. The two dimensions of the histogram were intensity and distance from the feature point. With positive results in matching the texture of local regions within images, a natural extension to object recognition was also explored.

A more direct augmentation to the SIFT technique was presented by Ke and Sukthankar [23] with the PCA-SIFT technique. Principle component analysis has been used in many fields to reduce the dimensionality of a problem while retaining key information. With PCA-SIFT, the dimensionality of the SIFT descriptor was reduced by projection on to the eigenvectors of the descriptor. The authors showed that the identity-related variations of the local feature were retained in the new description while unwanted distortions within the description were discarded.

In order to evaluate the abundance of local descriptors being developed, Mikolajczyk and Schmid carried out a performance comparison of several techniques [31]. The SIFT descriptor was compared against the Shape Context, Complex Filters, Spin Images, as well as PCA-SIFT and several other techniques such as cross-correlation. The object recognition framework of detecting feature points, describing the feature points using each of the listed techniques, and matching the feature descriptors between scene and model images was implemented. The authors also presented a new descriptor, the Gradient Location and Orientation (GLOH) descriptor, which is a direct extension of the SIFT descriptor. The GLOH descriptor is formed by computing the SIFT descriptor on a log-polar location grid and then reducing the dimensionality of the resulting vector by use of principle component analysis. The results of the evaluation of local descriptors showed that the GLOH descriptor performed best in most test cases closely followed by the SIFT descriptor. The Shape Context descriptor also performed well but failed on texture images or where edge detection could not be reproduced across images. This work showed the distinctiveness of the SIFT based descriptor as applied to image matching.

While SIFT based local features have enjoyed much success in the field of object recognition, they do not exploit the discriminative color information within images. All of the techniques discussed in this section operate in the grayscale domain with many using the gradient information as the key descriptive feature. The avoidance of color information in recognition techniques stems from the problem of color constancy: colored regions under different types of illumination (i.e. sunlight, incandescent lighting, fluorescent lighting, etc.)

generate different responses within image acquisition devices. While the human brain can automatically compensate for changes in illumination, achieving color constancy in computer vision is a challenging problem.

2.3 Object Recognition using Color

Color provides a distinctive cue in recognizing objects within scene images. In scenes with controlled illumination, simple algorithms have proven to be effective. With uncontrolled lighting conditions, the camera sensors capture drastically different color values from the same object. Because of this challenge, color constancy algorithms needed to be incorporated into object recognition algorithms. In the last decade, the use of color constancy combined with local features has incorporated the discriminative power of color information into the local feature matching algorithms in order to recognize 3D objects in 2D images.

2.3.1 Color Object Recognition under Controlled Illumination

The use of color in object recognition began with the influential work of Swain and Ballard in 1991 [40]. Using the RGB color space, the authors constructed a 3-dimensional histogram with the red, green, and blue channels as axes. The histogram was composed of the frequency counts of each discretized color in the pixels of an image. Such a histogram was created from a single image of an object and used to represent the object. To compare histograms, the authors presented a technique known as histogram intersection. This technique computes the element-wise minimum of two histograms and sums the resulting values. Provided that each histogram is normalized so that the sum of all elements equals one, the histogram intersection value provides a score between 0 and 1 representing the similarity between histograms. While Swain and Ballard showed that the color histogram was useful in recognizing object among database images (controlled illumination) of many objects, the authors admitted that the effects of variations in illumination was not addressed.

Two further approaches incorporating color information into the task of object recognition were presented by Huang et al. [20] and Chang and Krumm [7]. These approaches, using Color Correlograms and Color Coocurrence Histograms respectively, incorporated spatial information to the color histogram.

The Color Correlogram of [20] encapsulates the distribution of separation distances between each pair of colors in an image. The colors are first discretized and for each pair of pixels in an image, the separation distance and the colors of each pair of pixels is recorded. This structure retains all information within the color histogram but also adds discriminative spatial information that describes the relative positions of colors in the image. Experimental results showed superior performance to the color histogram.

The Color Coocurrence Histogram (CCH) of [7] uses extracted edges to describe the distribution of colors on either side of edges in images. For each edge point in an image, the colors of the pixels on either side of the edge are noted as cooccurring. The frequency distribution of each pair of colors co-occurring across edges in an image is encapsulated in the CCH. This spatial information increases the discriminative power of the color histogram allowing objects to be recognized in scenes with moderate clutter and occlusion; however, the problem of illumination variation was still not addressed.

2.3.2 Illumination Invariant Color Object Recognition

The human visual system is capable to adapt quite well to changes in illumination. Objects illuminated by fluorescent light, for instance, are quite easily recognizable under incandescent lighting. To achieve this in computer vision, a system is required to emulate this inherent ability of humans.

In 1971, Land and McCann [24] presented research that would prove to be instrumental in the development of a field of research known as color constancy. The authors proposed a theory of color vision known as the Retinex Theory. This theory claimed that perceived color depends not only on colors of specific points in images but also colors of surrounding regions. The computation of color-constant color descriptors was presented using information available from sensor responses. Over the next several decades, many researchers presented new and improved techniques to extract the reflectance colors of an object that were invariant to the color of the illumination present [30, 14, 11].

The idea of illumination invariance was applied to the field of object recognition by Funt and Finlayson [15]. Here, the color indexing technique of Swain and Ballard is implemented by indexing illumination invariant surface descriptors formed based on the Retinex theory of Land and McCann. The algorithm was shown to reduce the sensitivity to lighting conditions that hampered the original technique by Swain and Ballard.

Gevers and Smeulders [17] investigated several other color spaces that are invariant to illumination changes under certain photometric conditions and assumptions. Normalized *rgb*, *hue* and *saturation*, as well as three novel color spaces proposed in the paper were used in the construction of color histograms for the use in object recognition. Under varying illumination, only one color model, based on color ratios of neighboring pixels, did not show substantial degradation in recognition accuracy.

While significant progress had been made in the application of color constancy research to the field of 3D object recognition, the use of a single color histogram to represent a model object in order to recognize instances of the model object in scene images was commonly used up until the early 2000's. Single color histograms showed little distortion under slight variations in geometry and viewpoint direction; however, occlusions and background clutter were a serious challenge in the recognition of objects in scene images. The use of local features, which has been greatly pursued in the last decade, overcame these challenges quite readily. A natural progression to such approaches was to incorporate the discriminative power of color into the local features.

2.3.3 Color Object Recognition using Invariant Local Features

With the success of the SIFT descriptor presented in [28] and [31], many research groups looked to improve upon the object recognition technique to achieve even better results. One such improvement was the addition of color information to the descriptor. Developments within the field of color constancy provided robustness to photometric variations such as illumination and shading.

One such development was the *colored*-SIFT (CSIFT) descriptor presented by Abdel-Hakim and Farag [1] in 2006. Here, the authors extracted the SIFT descriptor in the same fashion as Lowe only it was extracted from an illumination invariant color space developed by Geusebroek et al. [16]. This illumination invariant color space was constructed based on the Kubelka-Munk theory of image formation. Using assumptions of illumination distribution and uniformity, the *rgb* color space was transformed to three new channels of which changes in illumination had no effect. The SIFT algorithm was carried out using these color channels by finding gradient responses within the image at different scales. The experimental results show better repeatability of feature extraction under varying illumination conditions.

The SIFT approach has also been applied to the hue-saturation-value (HSV) color space in [6]. Bosch et al. apply the feature detector and descriptor to each channel of the color space and concatenate the resulting descriptors to one color feature descriptor.

In recent years, the shift in object recognition research has been towards class recognition, also known as object category recognition. The task is to classify a novel image of an object with a specific category label (i.e. car, dog, book, etc.). To do this, many techniques apply machine learning algorithms to recognize features within a database of images. For instance, to correctly classify an image of a car with the category label 'car', features are first extracted from many car images in a supervised learning stage. The features are then matched to the novel image and a classifier determines whether the label should be applied.

The use of color information in the task of object category recognition has shown mixed results [43]. It is clear that some categories (such as footballs, roses, or stop signs) show a high correlation in colors within the category, while others (such as cars, flowers, or dogs) show very low correlation. Because of this variability, descriptors encapsulating the color information show similar results to those generated from grayscale images.

2.4 Object Recognition using Multiple Views

When recognizing objects within scene images, there is always some uncertainty in the result. If a match is found, it may be incorrect (false-positive); if a match is not found, the object may have in fact been present in the scene but not correctly matched (false-negative). To improve recognition rate, multiple views of the scene may be used. Images captured at different viewpoints within a scene allow for the object recognition algorithm to be applied to each image and a consistency check to be applied to ensure that the object was recognized from each viewpoint.

The advancement of mobile robot technology has allowed imaging devices to be placed on mobile robots equipped with navigation, localization, path-planning, and obstacle avoidance algorithms. With the ability to navigate through an environment, the mobile robot may capture images from different viewpoints and receive input from the object recognition algorithm to determine where to move next in order to further inspect certain locations. Whereas single image object recognition algorithms only required a single processing stage, algorithms incorporating multiple views must process each image captured.

Early work incorporating mobile robots with the field of object recognition was presented by Frintrop et al. [12] in 2004. In this work, a 3D laser scanner is used to find an illumination invariant depth map describing the relative distances between the mobile robot and objects within the environment. The authors use the principles of visual attention inspired by Itti et al. [22] to detect salient regions in the depth map as well as a captured scene image. Subsequently, the detected regions are processed by a trained classifier based on the popular boosting technique used by Viola and Jones [45]. Frintrop et al. did not present the incorporation of multiple views but the potential for mobile robots to navigate through environments and recognize objects from different viewpoints was demonstrated.

With an initial salient region detection stage, an exhaustive search of an image for a specific object is not necessary. By only searching the salient regions, the algorithm is sped up considerably. Orabona et al. [33] incorporated object-specific information in to the initial visual attention stage to locate salient regions that had similar appearance properties to a target object. The authors use a stationary robot and use the object-based visual attention system to locate potential matching regions. Once the regions are detected, the head of the robot turns to center the region within an image and the object recognition process is carried out.

Tsotsos and Shubina outline a more complete system design that incorporates object recognition with mobile robot navigation in [42]. In this work, the authors use a stereo vision system mounted on a pan-tilt apparatus located on a mobile robot. The depth information from the stereo vision system is associated with each image region so that 3D coordinates are obtained. The authors use a grid of cubes to describe regions in the 3D world and associate a match probability with each cube based on object recognition results. A navigational algorithm is used to decide where to move next based on the locations of matching regions. At the new location, the recognition process is repeated and the match probabilities for each cube are updated in a Bayesian fashion. A probability threshold is used to conclude whether or not a match exists and what the coordinates of the match are.

The advancement of object recognition algorithms using mobile robot platforms has spurred the development of two robot competitions: The Semantic Robot Vision Challenge (SRVC) and RoboCup@Home. These competitions require robot platforms to explore an unknown environment and locate specific objects within. Training data is collected based on an Internet image search automatically performed by each system.

The winners of the 2007 SRVC competition, Forssen et al. [10], presented a comprehensive robotic system that incorporated many components from several different fields. In this system, a high resolution camera was used as well as a stereo vision camera. The vision system was mounted on a pan-tilt apparatus atop a mobile robot as in [42]. Due to time constraints in the SRVC competition, the system would navigate through the environment capturing images from as many locations as possible before applying the object recognition algorithm. Localization, navigation, obstacle avoidance, and mapping algorithms were incorporated with the SIFT object recognition system. A post-processing stage was utilized to register the images captured from all viewpoints to determine the coordinates of the best matching region within the environment.

A similar approach was implemented by Sjo et al. [38]. Also inspired by the SRVC competition, the authors developed a system which utilized a laser range finder to generate the relative depths of objects in the room. An initial mapping stage is implemented using SLAM to generate a map of navigable coordinates within the environment. A secondary stage employs a visual attention algorithm with the SIFT object recognition system in order to locate the best matching regions.

If an object is completely occluded from one viewpoint, the single image recognition system has no chance in locating it. However, using mobile robot navigation and multiple viewpoints, it is possible to obtain a viewpoint in which the target object is not completely occluded. This process more closely mimics the behavior of a human searching for a particular object within the surrounding environment.

The recent successful approaches by Sjo et al. [38] as well as Forssen et al. [10] provide a complex framework in which many technologies are incorporated. In both systems, the object recognition system relies on grayscale matching using SIFT features. A natural improvement to such approaches is to use the distinctive power of color to locate specific objects within unknown environments.

Chapter 3

Thesis Objective

The objective of the work presented in this thesis is to develop an object recognition system, based on the matching of illumination invariant local color features, that may be incorporated with a mobile robot platform for the use of searching for specific objects within unknown environments. The evolution of object recognition algorithms, presented in the previous chapter, shows the natural progression towards this goal and also shows several works that approach similar problems.

The early works in object recognition required the use of geometric models. While these works may be credited for their influence on the development of the field, it is clear that this approach is not feasible for applications on mobile robots. Occlusion and background clutter are two very common and challenging problems in object recognition. With an object occluded or placed in front of a cluttered background, it is likely that the projection of the geometric model will not be matched to the edge map of the scene image.

To overcome these challenges, researchers worked on matching only portions of objects. By using local features, local regions of an object could be matched in a scene image. Many of the original works using local features used local descriptors extracted from grayscale images. This avoided the problem of color constancy under varying illumination but it meant that the techniques could not use the descriptiveness of the color information inherent in objects. The original works using local features also concentrated only on single image object recognition and did not incorporate results from multiple viewpoints.

With the development of several color constancy algorithms, local feature techniques for object recognition eventually incorporated color information; however, these techniques were applied to the problem of object category recognition. As noted in the previous chapter, intra-category differences in color information posed a serious problem for local color features. The local feature techniques using color information were not applied to the task of recognizing a specific object in an unknown environment.

The recent developments in the use of object recognition in conjunction with mobile robot navigation have been spurred on by the competitions of SRVC and RoboCup@Home. One key requirement of these competitions is that the input to the system must be a description of the object to be searched for. The systems must then connect to the Internet and search for images using the object description and learn about the appearance of the object from the search results. In many cases, only general information about the object's appearance may be extracted from Internet search results. By using a learning stage that employs information specific to the appearance of the target object, a more accurate match may be found in the environment.

The work presented in this thesis describes a system that is able to utilize the color information for the use of searching for specific objects within unknown environments. Images captured from multiple viewpoints within the environment and the corresponding object recognition results are combined to produce a more certain result. To generate accurate matching results, an object-specific learning stage is carried out to extract unique information about the appearance of the target object.

Searching for and locating objects within unknown environments has several useful applications. One such application would be to assist humans in fetching and retrieving household objects. In particular, disabled or elderly people could employ a mobile robot with a particular command such as, “Bring me my slippers,” to increase the accessibility of everyday tasks. The mobile robot could then apply the object recognition system to search the household setting for the requested object. Another useful application would be searching for objects in dangerous environments. Currently robots are employed by the military in a remotely controlled fashion to locate explosive devices. With an autonomous mobile robot able to locate explosive devices without human interaction, multiple systems may be used to search the environment more thoroughly and effectively without placing humans in harms way.

Chapter 4

A Novel Object Recognition System

In this chapter, a solution is presented to the problem of recognizing objects within an unknown environment. A stereo imaging system is used in conjunction with a mobile robot platform to capture images and corresponding depth information within the environment from multiple viewpoints. An object recognition process is then carried out and the results from the multiple view images are combined to determine if and where the target object is located in the environment.

A local feature matching approach is used so that cluttered backgrounds and partial occlusions do not hinder the performance of the system. In a training stage, information is extracted from database images of specific objects. The extracted information is in the form of local color features constructed in such a way that it allows for illumination and viewpoint changes while retaining the important color information of the region.

Once a scene image is captured, the illumination of the image is compensated to match the illumination of the database images. With the depth information obtained from the stereo imaging system, the image is then segmented based on relative distance from the imaging system. The target object is then searched for in each segment. Local feature matching using histogram intersection is used followed by a verification stage using triangular constraints. The result is a confidence map showing potential matches within the scene. Figure 4.1 presents the flowchart for the complete object recognition process. Both

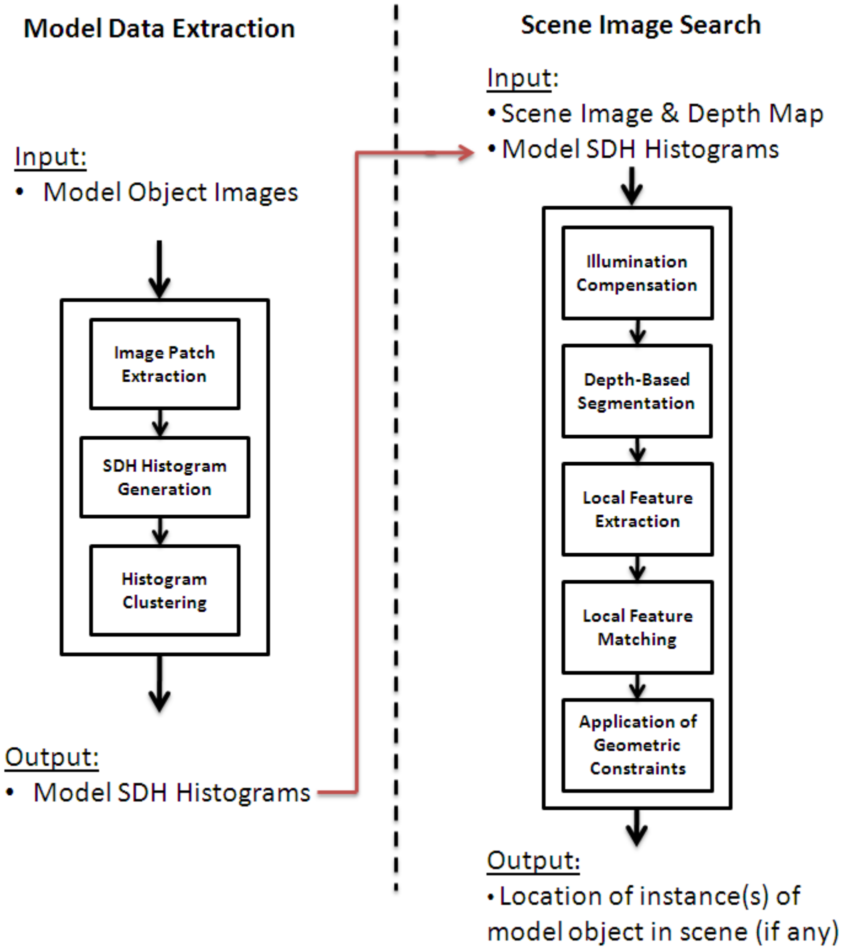

Figure 4.1: System Flowchart. Both the model data extraction stage and the scene image search stage are presented.

the model data extraction stage and the scene image search stage are shown. As an example of the implementation of this system, Figure 4.2 shows the model images of an object (*basketball*) along with the matching results in the scene.

After locating image regions that match the appearance of the target object, the 3D coordinates of the location is computed using the depth information from the stereo imaging system. This 3D location may then be used to plan a path to another location within the environment at which a secondary view of the matching region may be captured. By performing the object recognition process on a second view of the same region, the confidence in the matching results from the first image is then increased or decreased based on the

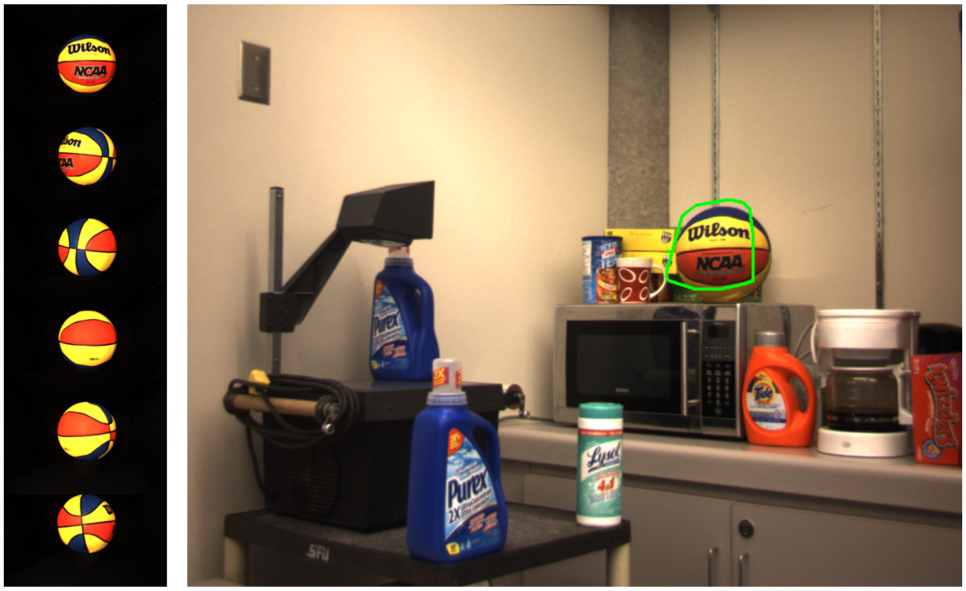

Figure 4.2: Object Recognition Results. Database of target object images are shown on the left. The location of the target object is successfully detected in the scene image.

subsequent results.

The robot platform may be used to capture images from many viewpoints within an environment. After detecting a potential matching region in one image, images from multiple viewpoints are captured and processed. After comparing results from at least three viewpoints, a potential match is either rejected or confirmed based on the combined results of all views.

The following sections in this chapter describe each module of the proposed solution in detail. The hardware used is first described followed by the formulation and extraction of the local feature used in the object recognition process. The training stage is also described followed by the scene search stage in which potential matching regions are found. Lastly, the robotic inspection stage describes the incorporation of images captured from multiple viewpoints within the environment.

4.1 Hardware

The implementation of this work used two important hardware components: an imaging system and a robotic platform. The imaging system used was a BumbleBeeXB3 stereo vision system from Point Grey Research Inc. This device is composed of three Sony ICX445 1/3 progressive scan CCDs. The maximum resolution achievable by each camera is 1280x960. Having three cameras such as this allows for two possible baseline distances for capturing depth information. Both 12cm and 24cm baselines are achievable with the BumbleBeeXB3 model. For the application presented here, the 12cm baseline was used to generate better stereo processing results for objects closer to the camera.

The robotic platform used was a PeopleBot mobile robot from MobileRobots Inc. This model is equipped with several sensor modalities such as sonar, infrared, and pressure. The robot is controlled via an on-board computer and accepts instructions sent wirelessly over a network.

The stereo vision system was mounted atop the mobile robot at a height of 120cm from ground. At this height, items located on shelves and tabletops are easily captured and able to be inspected from multiple depths and angles. The complete system is shown in Figure 4.3.

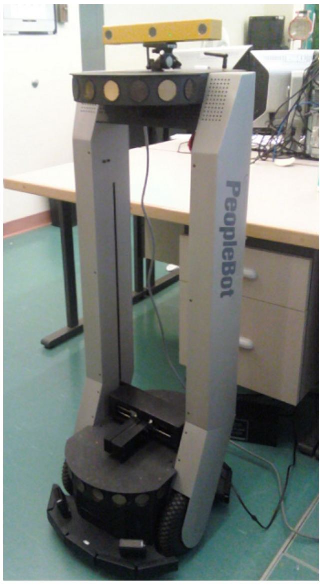

Figure 4.3: BumbleBeeXB3 from Point Grey Research mounted atop a PeopleBot mobile Robot from MobileRobots Inc.

4.2 The Saturation-Weighted Distributive Hue Histogram

The method of using local features to recognize objects in images relies on the ability to match local regions within a model image of an object to local regions within a scene image. This means that the descriptor used to represent local regions of the model object must be both descriptive and discriminative.

The goal of this work is to utilize the color information of an object to provide the discriminative power and descriptiveness required. To do this, a local feature known as the Saturation-Weighted Distributive Hue Histogram (SDH Histogram) has been developed. The SDH histogram exploits the invariance properties inherent in the hue channel of the *hue-saturation-value* (HSV) color space by forming a distributed histogram that can be used to compare the color content of extracted image patches.

From [44], van de Weijer and Schmid show that the hue channel in the *hue-saturation-value* color space is invariant to lighting geometry, specularities, and illumination variations. The challenge in using the hue channel for describing colors is the instability near the chromatic axis (i.e. when saturation is low). To this point, van de Sande et al. [43] show that the certainty of the hue value is proportional to the corresponding saturation value. That is, colors with low saturation (closer to the chromatic axis) have higher margins of error in the hue component.

To take advantage of the invariance properties of the hue component, the certainty of the hue value must be taken in to account. Using the robust histogram construction technique presented by Gevers and Stokman [18], the saturation value is used as an error probability for the discretization. This robust histogram construction technique is used to construct local saturation-weighted distributive hue histograms.

4.2.1 SDH Histogram Construction

In general, the use of histograms is associated with discretization of measurements and counting the frequency of occurrence of such discretized units. This is often looked at as a drawback to the use of histograms because the number of 'bins' used to discretize the measurements is often arbitrary and can lead to errors in the representation of the data.

In 2004, Gevers and Stokman [18] presented a more robust technique to constructing histograms. The technique was not sensitive to the placement of the bin 'edges'. Instead of discretizing a measurement and then incrementing the frequency count of the corresponding bin, an incrementation distribution is generated in Gaussian form based on the certainty of each measurement. Each bin is then incremented by an amount corresponding to the distribution.

The Gaussian incrementation distribution is centered at the measured value and the function values that landed on each of the bin centers are used as the incrementation amount for each bin. The variance of each incrementation distribution is determined based on the certainty of the measurement. If a measurement has low certainty, the distribution variance is high and each bin is incremented by a similar amount. If a measurement has high certainty, the distribution variance would be low and bins centered near the measured value are incremented by a greater amount than bins further away. Since the hue channel is a periodic channel, incrementation distributions are generated in a periodic manner.

Using the robust histogram construction technique, a color histogram of an image patch can be generated through the hue and saturation measurements of the pixels within the patch. The value component of the HSV color space is ignored in order to gain invariance to shading in images. Alone, the hue component is unstable near the chromatic axis. That is, with low saturation, the error within hue measurements is high. As saturation is a radial dimension, it has a linear relationship with the certainty in the hue measurement. This value can then be used to determine the variance of the incrementation distributions while constructing the histograms in the robust manner described previously. It has also been shown in [44] that the certainty of the hue is in fact directly proportional to the saturation measurement. Since the increments are distributed and weighted by the saturation measurements, the name Saturation-Weighted Distributive Hue (SDH) Histogram will be used to describe this local feature.

To construct the SDH histogram for a given image patch, the hue and saturation components of each pixel are first obtained. For each pixel, an incrementation distribution is generated based on these values to determine the amount to add to each bin. This distribution is given by

$$f(b;h,s) = \left(\frac{s}{\sqrt{2\pi}}\right)\exp\left(\frac{-\left(h-h_b\right)^2}{2\left(\frac{1}{sN}\right)^2}\right), \tag{4.1}$$

where b is the bin number, h and s are the hue and saturation components respectively, h_b is the hue value centered at the current bin, and N is the number of bins in the histogram.

For this work, 64-bin histograms were used. After the incrementation distributions have been added to the histogram for each pixel in the image patch, the histogram is normalized so that the sum of all values in the histogram is equal to 1. This allows histograms constructed from image patches of different dimensions to be compared via histogram intersection.

Further motivation for the use of a distributive construction technique for the SDH histogram is to provide invariance to illumination variations. With a change in lighting or photometric conditions, the hue values of pixels may change. By incrementing more than one bin in a histogram by an amount corresponding to the certainty of each measurement, invariance to such changes is achieved.

The construction of a SDH histogram is demonstrated in Figure 4.4. The top image in the figure shows the image patch to be described. As mentioned, each pixel in the image patch generates a histogram incrementation distribution all of which are summed together. Two pixels are highlighted in the image patch in this figure and the corresponding incrementation distributions are presented in the lower-left and lower-right images.

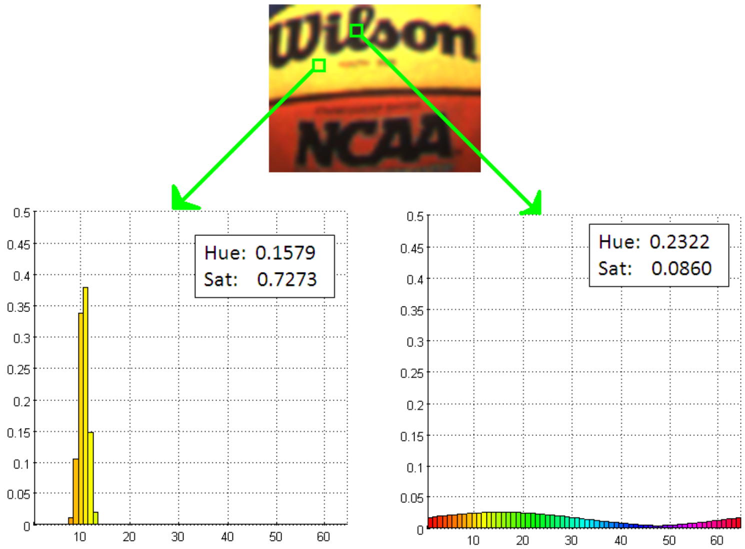

Figure 4.4: Saturation-weighted Histogram Incrementation Distributions. *Top:* An example image patch is shown with two pixels highlighted. *Bottom:* The corresponding histogram incrementation distributions for the two highlighted pixels are shown. Note that the pixel on the left has high saturation and thus high certainty, while the pixel on the right has low saturation and thus low certainty.

The highlighted pixel on the left of the image patch in Figure 4.4 has a hue value of 0.1579 and a saturation value of 0.7273. This saturation value shows a high certainty in the hue component and so the corresponding incrementation distribution has low variance and is centered at the hue value. The second pixel highlighted in the image patch has a hue value of 0.2322 and a saturation value of 0.0860. This shows a low certainty in the hue component and thus the corresponding incrementation distribution has high variance across the bins. After summing the incrementation distributions from all pixels in the image patch and normalization, the SDH histogram is obtained. Figure 4.5 shows the corresponding SDH histogram to the image patch shown in Figure 4.4.

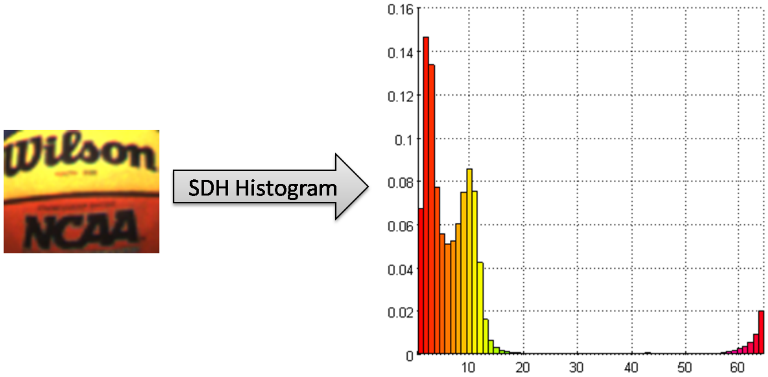

Figure 4.5: SDH Histogram generated from image patch

The SDH histogram in Figure 4.5 is shown with each bin colored with the corresponding hue. This particular SDH histogram shows two modes of red and yellow hues. This descriptor encapsulates the information that both yellow and red regions are present in the image patch while remaining invariant to shading, specularities, and illumination variations. The SDH histograms extracted from image patches of three other objects are also shown in Figure 4.6. It is clear that the modes in the SDH histograms correspond to the hues present in the image patch.

4.2.2 SDH Histogram Performance under Image Variations

The SDH histogram has been constructed in such a way to encapsulate the distinctive color information in an image patch. At the same time, the SDH histogram is robust to such variations as shading, photometric effects, specularities, and illumination changes.

To demonstrate this robustness, such effects were applied to the image patch of Figure 4.4. The SDH histogram was then generated from the resulting image and then compared with the original via histogram intersection in which the percentage of overlap between two histograms is calculated. The results are seen in Figure 4.7. Note that a histogram

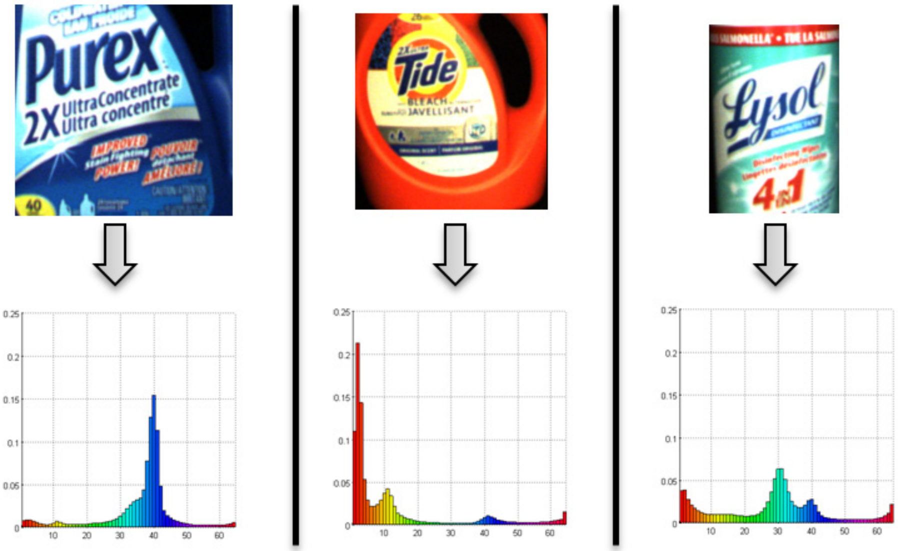

Figure 4.6: SDH Histograms from Multiple Objects

intersection score of 100% indicates a perfect match while 0% indicates a complete mismatch.

Four effects were simulated in this experimentation:

- *Shading.* The shading effect was implemented by reducing each pixel value by 50%. This simulates the object being in a poorly lit region of the environment.
- *Blurring.* The blurring effect was implemented by applying a Gaussian low-pass filter to the image patch. The blur simulates a photometric effect such as an out of focus area or camera movement.
- *Specularity.* The specularity was simulated by replacing a top-left region of the image patch by a white region. A specularity occurs when a region of the object reflects all light back to the camera; this usually occurs on highly reflective surfaces such as metal.
- *Illumination Change.* The illumination change was simulated by increasing the pixel values in the *red* channel of the *rgb* color space by 150%. This shift simulates a warmer illuminant being used to provide light to the objects.

Figure 4.7 shows that the resulting image patches produce high histogram intersection matching scores with the original image patch despite the applied effects. The shading of

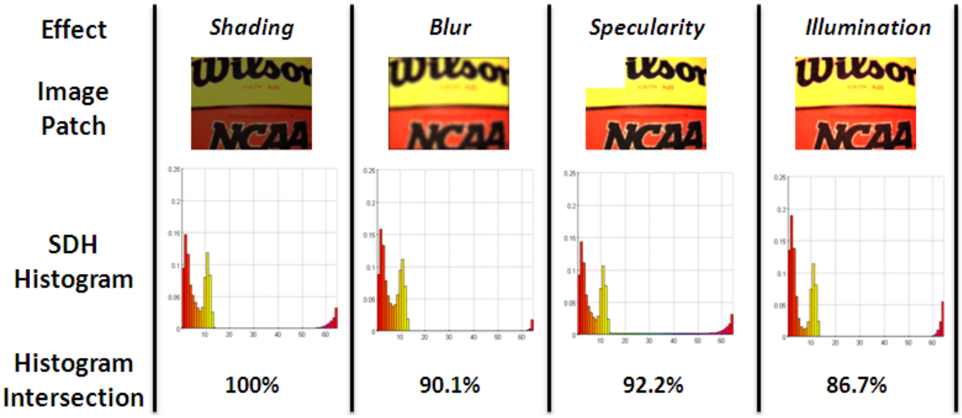

Figure 4.7: SDH Histogram Performance under shading, blurring, specularities, and illumination changes.

the image patch had no effect on the matching score. This is a result of neglecting the *value* component of the *hsv* colorspace. The blurring of the image patch had some effect but the matching score was still high. This result demonstrates the advantage in not being dependent on detecting corner points or edges within the image. The specularity effect also showed a high matching score despite a significant portion of the image patch pixels being replaced by white. Since the white pixels are on or close to the chromatic axis, the distributive construction technique places less weight on their contribution to the SDH histogram. Lastly, the illumination change showed the lowest matching score; however, it is still within 15% of a perfect match. This may be most surprising because the image patch appears unchanged. This demonstrates the ability of the human eye to accurately compensate for changes in illumination. The increase in the *red* component of the image provides a shift in the histogram towards the hue bins closer to red. With the distributive construction technique, a margin of error is provided by the variance of the incrementation distributions. This makes the SDH histogram robust to shifts such as this.

Three image patches extracted from other model objects were also tested in the same manner. The results of these tests are shown in Figure 4.8. Similar results to that of Figure 4.7 were obtained for each object.

With no information known about two image patches, it is equally likely that the two image patches match or do not match. Beacause of this, the *a priori* probability that a macth exists between two image patches is 0.5. After performing a SDH histogram intersection, the resulting score provides new information regarding the matching of the two image patches.

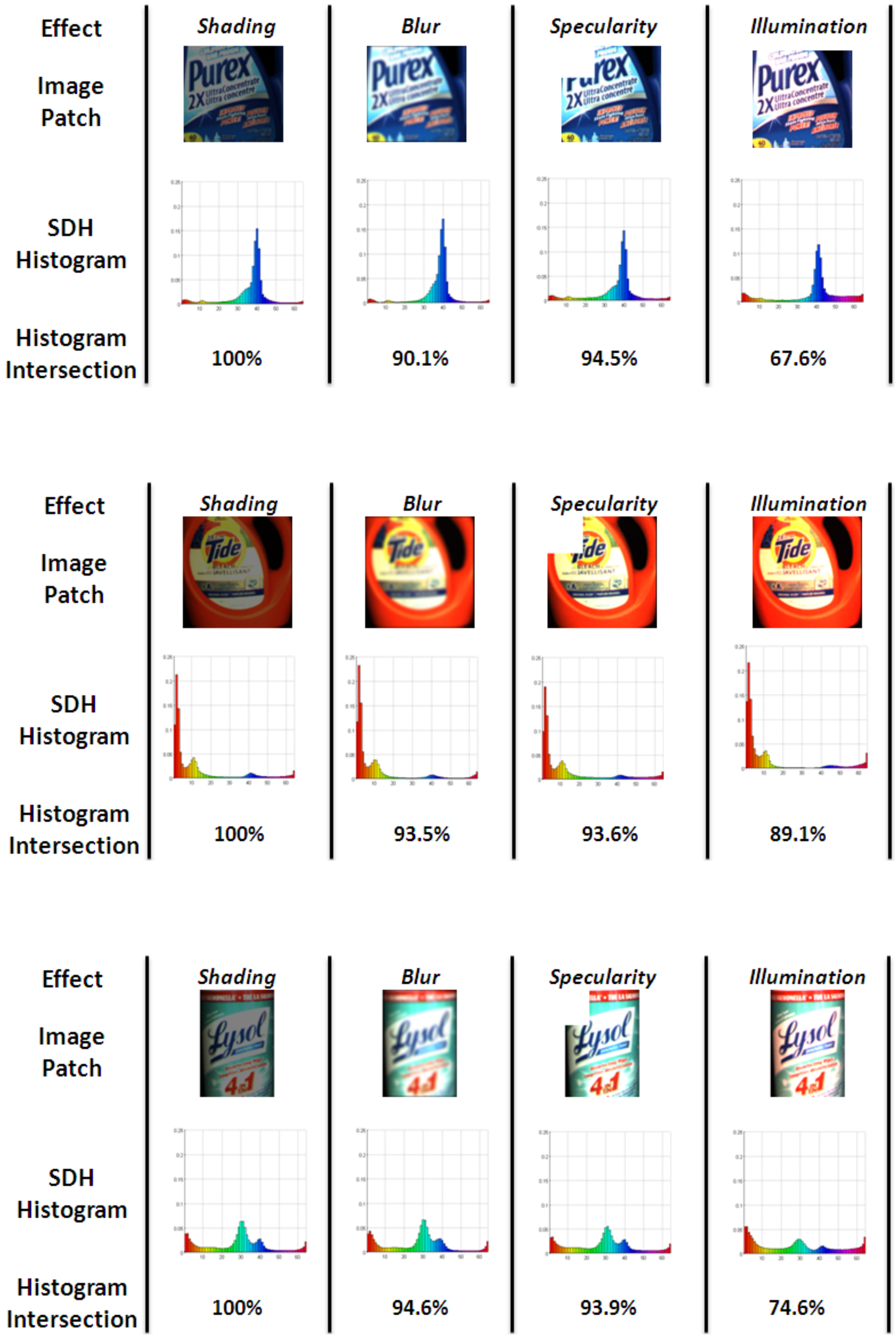

Figure 4.8: SDH Histogram performance under shading, blurring, specularities, and illumination changes for multiple objects.

By modeling the histogram intersection score as a match probability, each test provides a probability that a match exists. Therefore, histogram intersection scores greater than 50% provide evidence that a good match is present. In each effect applied to each object image patch, a histogram intersection score was obtained above 67%. This demonstrates the robustness of the SDH histogram to the effects of shading, blurring, specularities, and changes in illumination.

4.3 Extraction of Training Data from Model Images

In order to locate a specific object in a scene image, information about the object's appearance must first be known. It is this information that is searched for in the scene image. For the system presented here, the appearance information is of the form of SDH histograms generated from image patches extracted from database images of an object.

For this system, an object database was constructed by capturing images of model objects in a controlled setting. Six images of the object were captured at $60\,^\circ$ increments along the horizontal axis surrounding the object. An example of these six views are shown on the left of Figure 4.2. The objects are placed in front of a black velvet background so that all light not reflecting off the object is absorbed by the background material. This simplifies the isolation of the object so that information is extracted within the object boundaries. Additionally, the distance between the camera lens and the object is recorded for future use in the scene image search stage. This distance is measured manually and is a constant parameter in the construction of the model image database. The use of this distance value is presented later in Section 4.4.4.

For each view of each model object, the object is first segmented from the black background. Subsequently, SDH histograms are generated from image patches extracted at a range of locations within the objects boundaries. The composition of all SDH histograms extracted from all views of the object completely describe the color composition of the object. As a representation, the composition of all SDH histograms extracted is too large to be used to search for in a scene image. Typically between 100 and 200 SDH Histograms are obtained. To reduce the dimensionality of this representation, the extracted SDH histograms are clustered. The object is then described by the set of SDH histograms located closest to each cluster center. The extraction and clustering of SDH histograms is explained in further detail in the following subsections.

4.3.1 SDH Histogram Extraction from Model Images

The purpose of extracting color information from the object images is to use this information as an appearance description of the object. The purpose of using local image regions is to generate local region descriptions so that background clutter and occlusions do not cause false matches. That is, even if some of the object is occluded or if the background is somewhat confused with the object, local regions within the visible portion of the object will still be successfully matched since there is no dependency on surrounding regions.

A common problem with the use of local features is the selection of locations at which to extract them from. Ideally, a local feature is extracted from a certain location on the object in the database image and extracted from the same exact position on the object in the scene image. To do this, common approaches extract local features from corner or edge points found using simple filters. However, under changes in viewpoint and illumination or under photometric inconsistencies such as motion blur, these feature extraction techniques may not be repeatable.

To overcome this problem, this system uses an exhaustive technique in which image patches are extracted at a sliding grid of locations. An image patch is first extracted from the upper-left corner of the object and then slid to a position to the right with 2/3 overlap. A second image patch is extracted from this position and the process repeats until the object boundary is reached. Once reached, the window is slid downwards with 2/3 overlap and the sliding/extracting process repeats in the opposite direction. This process is continued until all object regions in the image have been extracted.

A final consideration in the extraction of SDH histograms is the size of the image patches to be extracted. Due to the nature of the SDH histogram, the complete representation of the object is somewhat sensitive to the sizes of the extracted image patches. Ideally, the extraction of the patches captures information regarding the color composition of the object as well as information of which colors occur within the same vicinity as other colors. For instance, the SDH histogram shown in Figure 4.5 shows two modes in the red and yellow regions. This shows that red and yellow are found in the came vicinity within this object.

On the selection of image patch sizes, if the patch sizes are too small, most patches will contain only one color or hue. The resultant set of SDH histograms will hold no more information than the color histograms first presented by Swain and Ballard [40]. At the other extreme, if the patch sizes are too large, the advantages of the use of local features are

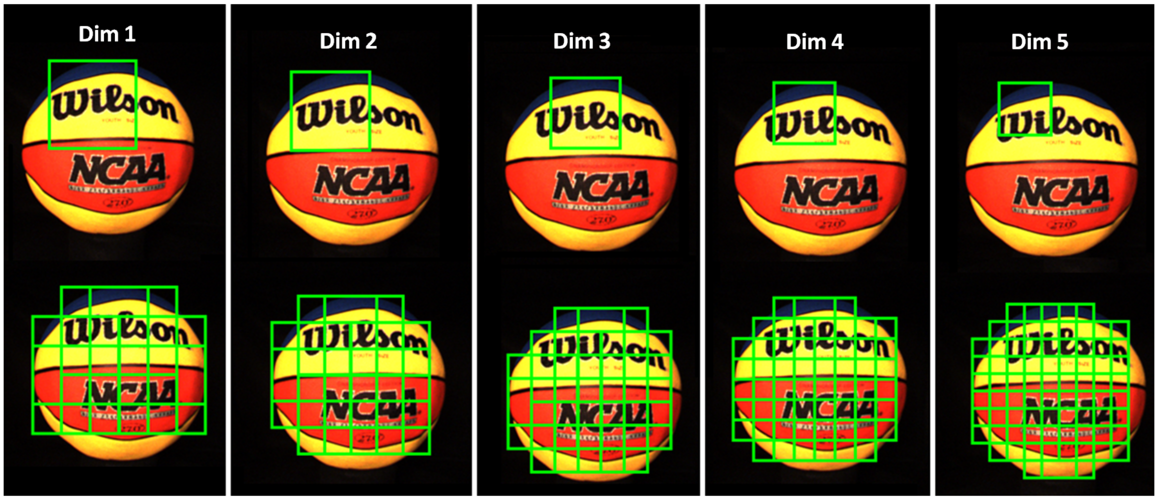

Figure 4.9: Image Patch Extraction. *Top:* Example window extracted from the object at each dimension. *Bottom:* Overlay of all extracted windows within the object boundaries.

not exploited and the algorithm becomes sensitive to occlusions and background clutter. To mitigate the problem of image patch size selection, a range of sizes is used in this work. Five sets of dimensions ranging from 30% to 50% of the complete object's dimensions are used and the sliding grid extraction technique is used for each dimension. This technique extracts redundant information while also avoiding the problem of image patch size selection.

Figure 4.9 shows the extraction process for a single view of a database object. Five dimensions are used to extract image patches (30%, 35%, 40%, 45%, and 50% of the complete object's dimentions). For each dimension, the figure shows an example window extracted from the object as well as the overlay of all extracted windows.

Clearly, this exhaustive extraction approach results in a large number of SDH histograms extracted. While the set fully describes the color content of the object, much of the information is redundant. To reduce the number of histograms used to represent the object's appearance, a dynamic clustering technique is implemented.

4.3.2 Dynamic Clustering of SDH Histograms for Object Representation

The information contained in the set of extracted SDH histograms represents the color content of local regions within the boundaries of the object. Due to the exhaustive sliding grid extraction technique, many of these SDH histograms hold similar information. It is for this reason that clustering of the SDH histograms is needed to remove redundancy while retaining the descriptive information.

With clustering comes the challenge of determining the correct number of clusters to use. In the case of this application, a constant number of clusters would be arbitrary. For objects that contain few color regions, fewer clusters would be sufficient while objects with complex color designs would be better represented by a greater number of clusters. Because of this, a dynamic clustering technique has been developed in order to determine the appropriate number of clusters to use.

When matching a SDH histogram from a scene image to the set of SDH histograms representing the model object, it is important that representative SDH histograms are distinctive. If too many histograms are used to represent the object, there would be a high similarity between the cluster centers. In turn, this makes it difficult to determine a match between a histogram extracted from the scene and one of the histograms used to represent the model. If too few SDH histograms are used to represent the model, the model may not be fully represented leading to no matches being found. With the correct number of clusters used, the color information from the model object is distinctively and effectively represented.

The dynamic clustering technique presented here clusters the data using several different numbers of clusters. Using the K-means method, the data is first clustered into two clusters and the distance between the cluster centers is measured. Since the data being clustered are histograms, an effective way to compare histograms is the histogram intersection technique. A distance measure which corresponds to the histogram intersection metric is the L1 distance [46]. The L1 distance, also known as city block distance or taxicab distance, is defined as the sum of absolute differences between vector elements. As this metric is analogous to the histogram intersection metric, the clustering technique presented here uses the L1 distance metric to measure distances between data points.

If the cluster centers show a histogram intersection of greater than 90%, it is determined that the data has been over clustered and that the correct number of clusters is one less than the current number used. For example, if five clusters were used, and two of the resultant cluster centers were very similar (greater than 90% histogram intersection), it would be determined that the appropriate number of clusters to use is four.

If the closest cluster centers show a histogram intersection of less than 90%, the number of clusters is incremented and the process continues to repeat until the termination case is found. In the rare case that all SDH histograms extracted from the model image are very distinct, the clustering terminates when the number of clusters equals the number of SDH histograms extracted. A flowchart outlining the dynamic clustering process is shown in Figure 4.10.

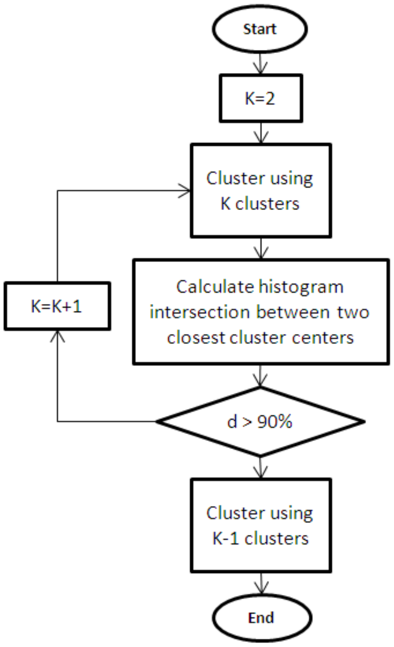
Start
K=2
Cluster using K clusters
Calculate histogram intersection between two closest cluster centers
K=K+1
d > 90%
Cluster using K-1 clusters
End

Figure 4.10: Dynamic Clustering Technique Flowchart.

This method of determining the a representative number of clusters avoids any arbitrary clustering while effectively reducing the number of data points used to represent the model object. Once an final clustering is found, the SDH histograms located closest to the centers of each cluster (using L1-distance) are used to represent the model image. Therefore, the number of clusters used equals the number of SDH histograms used to represent an object. Typically, in the model objects used in this work, five to ten clusters are used. The pixel locations of the image patches used to generated these SDH histograms are also recorded and used in a final geometric verification stage presented later. This process is repeated for each of the six viewpoint images of the database object.

4.4 Locating Objects within Scene Images

The goal of this object recognition system is to locate a specific object within a surrounding environment. Once information has been extracted from database images of a target object, images captured of the environment are then searched. By comparing the information extracted from the database images to the information extracted from the scene images, regions within the scene image that show strong similarity are marked as potential matching regions.

This work presents a new and unique method to accomplish the goal of object recognition in a scene image. Both stereo (depth) information as well as the color information encapsulated in the SDH histograms are used to locate regions within the environment where potential matches exist. Once the scene image and depth maps have been acquired from the imaging device, five processing stages have been implemented to locate matches in the scene as seen in Figure 4.1: Illumination Compensation, Depth-Based Segmentation, Local Feature Extraction, Local Feature Matching, and Application of Triangular Constraints. Each stage is explained in further detail in the following subsections.

4.4.1 Image and Stereo Data Acquisition

At any given location within the environment, images may be captured with the BumbleBeeXB3 stereo vision system. At one time, two images are captured simultaneously from two of the cameras which are separated by a baseline distance of 12cm. The BumblebeeXB3 uses the Triclops Stereo Vision Software Development Kit also provided by Point Grey Research Inc. The SDK performs stereo processing on the two images by establishing correspondences between them. The system uses the Sum of Absolute Differences (SAD) correlation method.

The SAD technique is a simple stereo processing technique that locates correspondences between the two images obtained based on image correlation. Based on the change in location of corresponding points in each image, a depth value is calculated. As a result of using the SAD technique, correspondences are found only for distinctive regions (non-uniform) in each of the images. For regions that are void of distinctive features (uniform regions), often no correspondences are found and thus no depth information is provided. Of the two images captured by the stereo imaging system, one reference image is used. Each pixel in the reference images is assigned a depth value based on the depth calculations.

More complete stereo algorithms introduce an interpolation step in order to fill in these

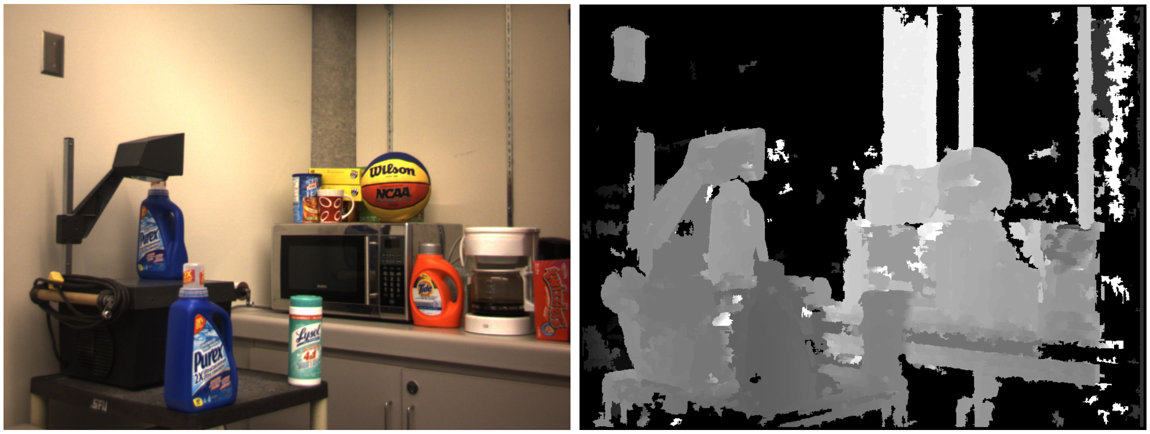

Figure 4.11: Image and Stereo Data Acquisition. *Left:* Image acquired by the BumbleBeeXB3 imaging system. *Right:* Stereo depth map generated from the acquired stereo data. Pixel values are proportional to the distance between each object and the camera lens.

regions of no depth information. However, for the recognition system presented here, the lack of depth information is used as a visual attention cue. Regions that have no depth information have no distinctive features and thus the probability of the target object being located in these regions is substantially low. The assumption being made here is that the target objects appearance is composed of distinctive features such as lines, corners, and textures.

Figure 4.11 shows the Image and Stereo data obtained in the acquisition process. The stereo data is represented by a depth map in which the grayscale pixel value is proportional to the distance to the camera. Regions that have no stereo data (regions with no distinctive features) are represented by black pixels.

4.4.2 Illumination Compensation

Once the reference image of the scene is captured, it is important to adjust the illumination of the image to match that of the database images. If the database images are captured under incandescent lighting and the scene is illuminated by fluorescent lighting, the pixel values within the images may vary drastically for the same object. From Section 4.2.2, it is clear that a slight change in illumination causes a strong effect in matching results.

In this system, the database images were captured with an incandescent spotlight used to illuminate the face of the object. As the search environment is unknown, and thus

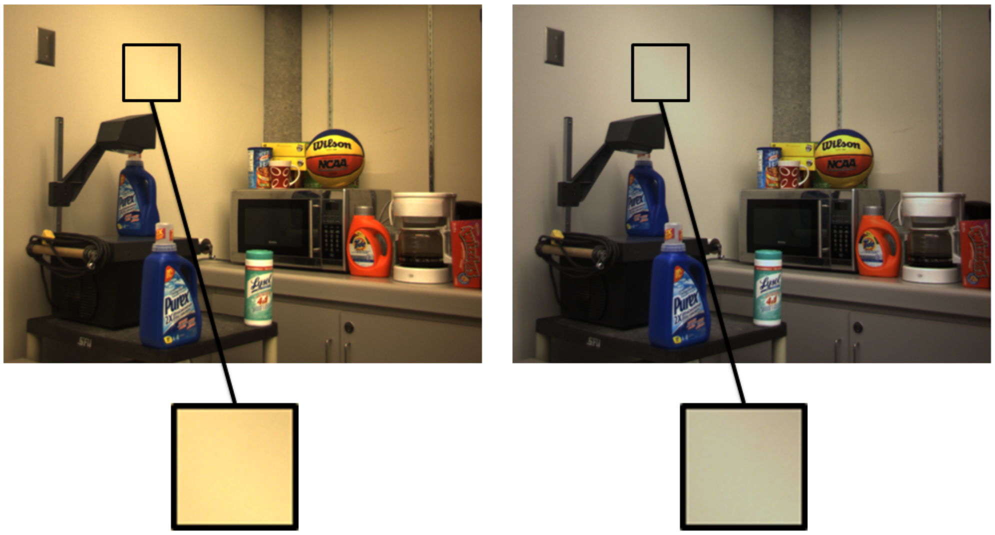

Figure 4.12: Illumination Compensation. *Left:* Original scene image with a sample patch isolated for demonstration. *Right:* Scene image after applying the illumination compensation stage again with sample patch isolated. The difference in color is clearly seen when comparing the isolated patches.

the illumination is uncontrolled, an illumination compensation stage is used to adjust the temperature of the scene illumination to match that of the database images.

The White Patch Retinex algorithm is implemented to compare white values in both the model and scene images [13]. By selecting the top 5% of pixels (in luminance) in both the database image and scene image, the color of *white* under the respective illuminants is compared. The average *rgb* pixel values of the top 5% luminous pixels in the database image is divided by that in the scene image to obtain a diagonal tranform. This transform is then applied to all pixels in the scene image by multiplication as in [9]. The resultant scene images is compensated for any variations in illumination.

Figure 4.12 shows a scene image before and after the illumination compensation stage is applied. It is clear that the diagonal transform has a large effect on the colors present in the image. By mitigating the variation in illuminations in unknown environments, the matching performance of the SDH histogram is improved.

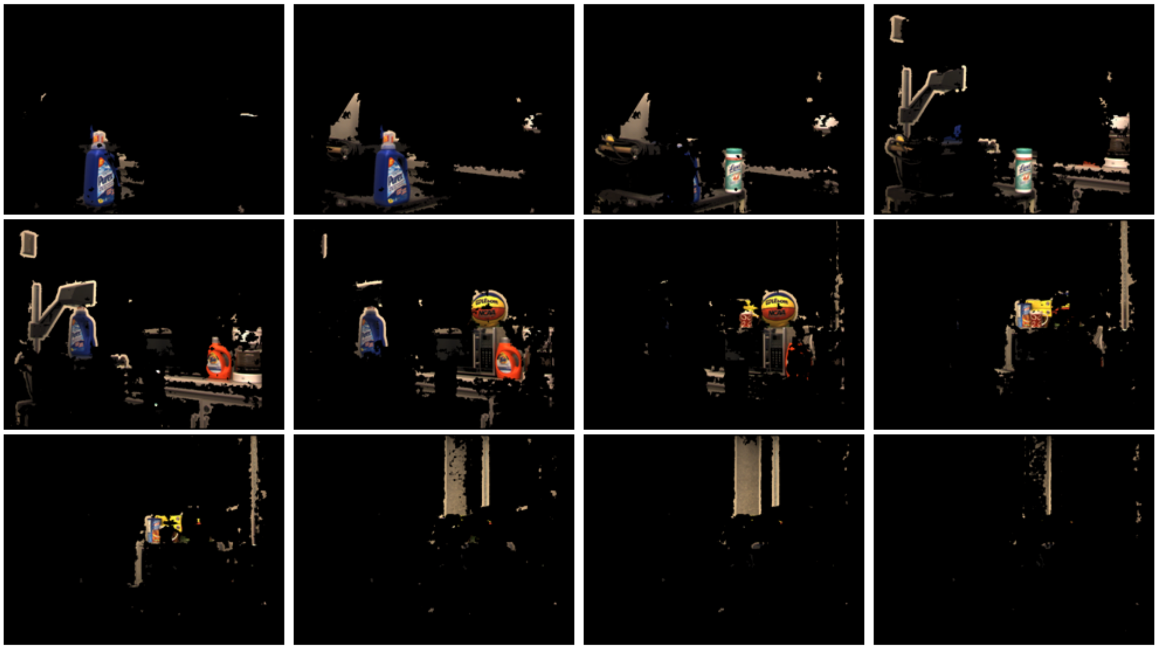

Figure 4.13: Depth-Based Scene Segmentation. Depth slices spanning 30cm in depth with 15cm overlap obtained using image and stereo data.

4.4.3 Depth-Based Scene Segmentation

Depth information obtained from the stereo vision system provides valuable cues for searching the scene image. As described previously, the local features extracted from the database images of the target object were extracted at a certain range of dimensions. To search regions in the scene with corresponding window sizes, the distance between the camera to each region must be known. Regions that are further away from the camera need to be searched with smaller window dimensions while regions closer to the camera need to be searched with larger window sizes. In this manner, an exhaustive search using many window sizes for all regions is avoided.

To separate the regions that are to be searched with different window sizes, a depth-based segmentation algorithm has been developed. The minimum and maximum depth values are first obtained from the stereo data. Overlapping depth intervals spanning 30cm (with 50% overlap) are then generated and the corresponding image segments are extracted. Each set of image segments corresponding to a certain depth interval is referred to as a depth slice. Figure 4.13 shows the depth slices obtained from the acquired scene image of Figure 4.11.

The overlapping of the depth intervals is used to ensure that all objects in the scene image are captured in one of the intervals and not split between intervals. This overlap also causes some objects to be located in multiple depth slices; however, this does not affect the final result. A system parameter, d_s, is used as an estimate of the size of the target object and used as an amount of overlap for each slice. This estimate ensures that the target object is completely captured in one of the depth slices.

With the regions in the scene images divided into depth slices, the object recognition process is carried out on each slice. The sliding grid technique is again used to extract local features from each slice and with the depth of each slice known, appropriate window sizes are chosen.

4.4.4 Local Feature Extraction

To search for the target object within each of the depth slices, a sliding grid of window locations is used similar to the training data extraction stage. For each depth slice, only the image regions found in the corresponding depth interval are searched. The appropriate window sizes to use for this search stage depend on the depth corresponding to each depth slice, the size of the windows used in the training data extraction stage, and the distance between the object and the camera in the training data extraction process.

The appropriate size of the windows used to search a specific depth slice are calculated by,

$$n_{slice} = n_{train} * \frac{d_{train}}{d_{slice}}, \tag{4.2}$$

where n_{train} is the window dimension used in the training data extraction process, d_{train} is the distance between the object and camera in the training images, and d_{slice} is the median depth of the specific depth interval.

Equation 4.2 assumes that the same camera is used for both the scene and database images at the same resolution. With this assumption removed, the equation used to calculate the correct dimensions to search a specific depth slice is,

$$n_{slice} = n_{train} * \frac{d_{train}}{d_{slice}} * \frac{P_{scene}}{P_{train}} * \frac{f_{scene}}{f_{train}} * \frac{\theta_{train}}{\theta_{scene}}, \tag{4.3}$$

where P_{scene} and P_{train} are the pixel densities (pixels per linear centimeter of camera sensor width) of the scene and database image respectively, f_{scene} and f_{train} are the focal lengths of the cameras used to capture the scene and database images respectively, and θ_{train} and θ_{scene} are the angular fields of view of the database and scene images respectively.

In the training data extraction stage, five different dimensions were used ranging from 30% to 50% of the total size of the object. This range of sizes is again used in the search stage by calculating five new dimensions based on these dimensions used in the training stage. These five values of n_{train} generate five values of n_{slice} to use as the search window dimensions for each slice.

This process of obtaining the appropriate window dimensions to search for in each scene region depending on its depth provides robustness to scale changes for the object recognition algorithm. This means that the object can be very close (large) or very far (small) in the scene and the algorithm will still be able to search for it effectively.

4.4.5 Local Feature Matching

For every window extracted from each depth slice in the scene image a SDH histogram is generated. These histograms are then compared to each of the SDH histograms used to represent the model object by a histogram intersection. As outlined in Section 4.2.2, image patches that generate a histogram intersection above 50% with any of the model SDH histograms provide evidence of a potential match and thus are marked as potential matching regions. Figure 4.14 shows two image patches and their corresponding SDH histograms. The histogram intersection score is also shown.

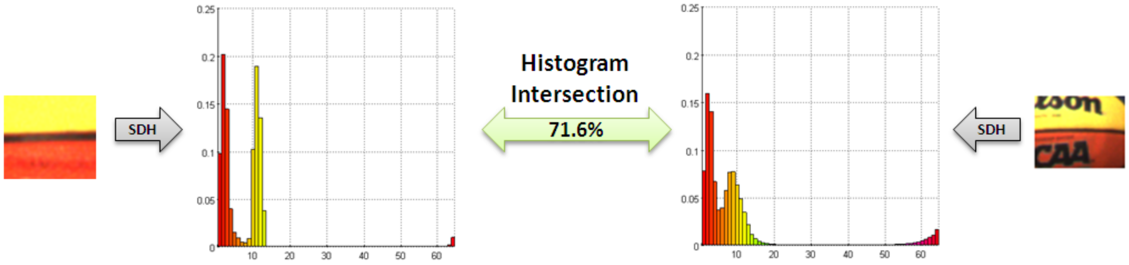

Figure 4.14: Histogram Intersection. Two SDH histograms are generated from two distinct image patches. The percentage of intersection, or overlap, of the two histograms is used as the matching score.

To increase the discriminative power of the matching algorithm, the concept of spatial pyramid matching is utilized [26]. If an image patch generates a SDH histogram intersection of greater than 50% with a model histogram, the spatial pyramid matching technique is applied. To implement the spatial pyramid technique, the scene image patch as well as the matching image patch from the model image are divided into four quadrants and the SDH histogram is generated for each quadrant. A match between image patches is only considered positive if all four quadrants in the scene image patch generate a SDH histogram intersection greater than 50% with their corresponding quadrants in the model image patch.

An example of how the spatial pyramid matching benefits the object recognition process is shown in Figures 4.15 and 4.16. In Figure 4.15, two image patches composed of similar colors are extracted from different objects. A matching score is obtained above 50% indicating a match however since the patches are from two different objects, this is a false positive result. Figure 4.16 demonstrates the application of the spatial pyramid technique. Each image patch is divided in to four quadrants, the SDH histograms are generated, and a histogram intersection score is obtained for each quadrant. Only two of the four quadrants yield positive matches and thus the two image patches are concluded to be non-matches. This example shows how the spatial pyramid technique reduces the amount of false positive matches in the search stage.

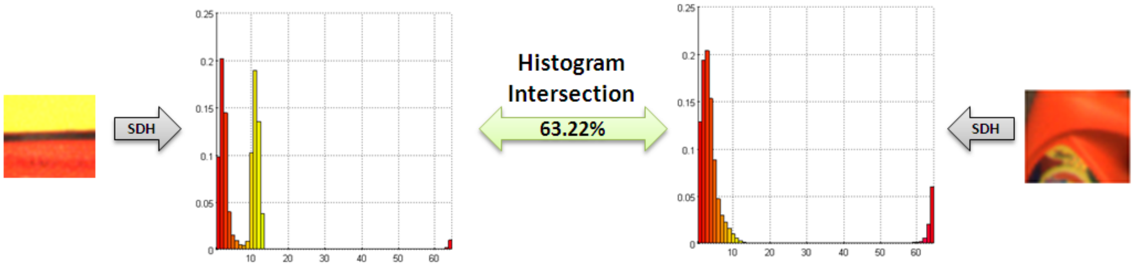

Figure 4.15: Histogram Intersection False Positive. Two SDH histograms are generated from two image patches extracted from two different objects. A score above 50% is obtained and a false positive match is

Once a positive match is found, the histogram intersection score of the entire patch is used as a confidence score to generate a confidence map. Also, the locations of the matching image patches are recorded along with the locations of the matching patches in the image of the database object. Each region where a positive match was found is assigned a corresponding histogram intersection score. For regions where multiple windows overlap, the average confidence score at each location is used.

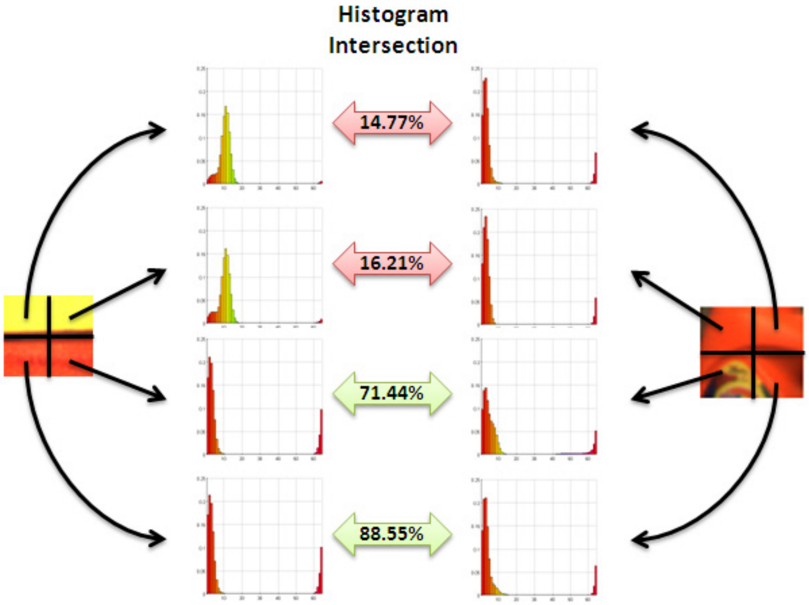

Figure 4.16: Spatial Pyramid Matching. The four quadrants of each image patch are extracted and SDH histograms are generated for each. Histogram intersection results show that only two of the four quadrants show matches yielding a negative matching result.

4.4.6 Application of Triangular Constraints

After detecting regions within each depth slice as potential matches, a final stage incorporates contextual information from the target object to validate each region. The concept of semi-local constraints is used to check the consistency of the relative locations of matches between the scene and model images.

Semi-local constraints are used to check pairs of matching points in the scene to ensure that their separation distance is consistent with the separation distance of the corresponding patches matched in the model image [37]. For each pair of matches in the scene, the pixel distance between them is measured and compared to the corresponding pixel distance of the matched points in the database image. If the distances are consistent based on the scale change and within a margin of error, the points are considered valid matches. Otherwise, the matching points are disregarded.

To increase the discriminative power of semi-local constraints, the work presented here uses a third point to form a triangular constraint. The addition of a third constraining point adds complexity to the algorithm but also reduces the number of false positives found in the matching results. By using three points and ensuring that the distances between each of the points are consistent with the distances between each of the corresponding matching points, the triangular constraint ensures that each triplet of points is geometrically consistent between the model image and the scene image. By searching each depth slice for matching image patches which pass triangular constraints, the expected separation distance of the image patches is known.

Since the distance between the camera and the model object is known in each of the model images, and the depth information is available for all regions in the scene image, the separation distance of the matching image patches in the model image may be scaled to the expected separation distance in the scene image for each depth slice. This ensures that only matching image patches found in the scene which have a spatial arrangement consistent with image patches in the model image are found.

The triangular constraint also ensures that at least three regions of the model object are located in the scene. Scene regions to which less than three image patches from the model are matched will not pass triangular constraints and thus are eliminated from the matching results.

For a specific matching point in the scene image, if there are two other matching points in the scene which pass distance constraints with each other as well as with the specific point being examined, a triangular constraint is passed. Points which do not pass a triangular constraint are rejected while those that pass are marked as matching regions. Figure 4.17 shows three image patches matched between a scene image and a model image of the *basketball* object. The triangular constraint is demonstrated as the Euclidean distances between the image patches in the scene are consistent with the distances between the matching image patches in the model. This figure also shows the invariance to the large scale difference between the *basketball* object in the model and scene images.

Figure 4.18 shows matching results before and after the triangular constraints stage is applied when searching for the ball of Figure 4.2. Green squares indicate positively matched image patches. Note that after the triangular constraints are applied, positive matches are only found in the vicinity of the ball in the scene. The confidence map generated from the histogram intersection scores of the matching image patches is shown in Figure 4.19.

The application of triangular constraints is the last processing stage in the object recognition algorithm. With the use of local features, regions within the scene image are matched to the database images of the model object. Local feature matching also provides robustness

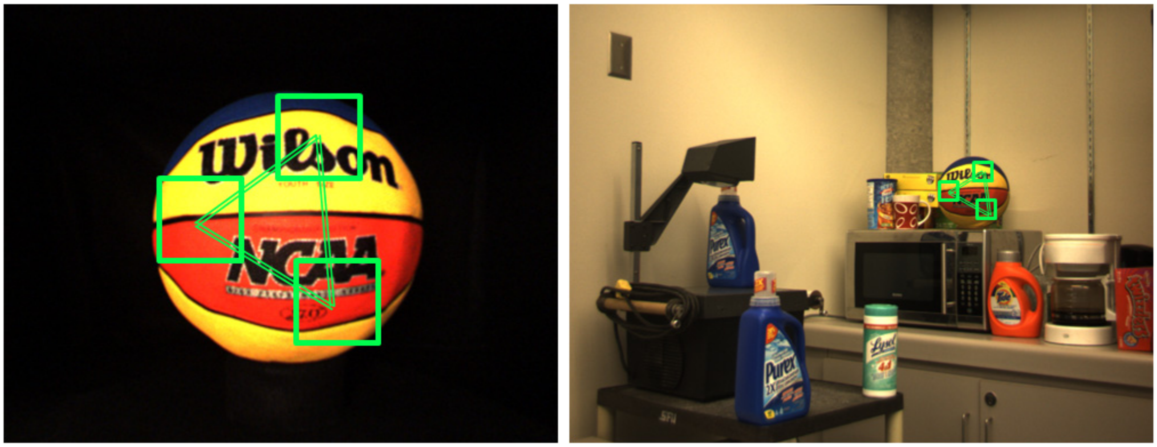

Figure 4.17: Triangular Constraint Example. Three image patches extracted from the scene image (right) are matched to the model image (left). The distances between the image patches in the scene are consistent with the image patches in the model, thus passing the triangular constraint.

to partial object occlusion as only part of the object is needed for a successful match to be found. As long as at least three image patches are matched between the scene image and the model object image, and a triangular constraint is passed by these three image patches, a successful match in the scene image may be located. Figure 4.20 shows the ability for the model object to be located despite partial object occlusion. In each row of the figure, an object is searched for in a database image of itself with varying occlusion. The results show that of the four objects tested, all were successfully identified under 20% and 40% occlusion, two of the four were successfully identified under 60% occlusion, while no matches were found under 80% occlusion.

4.5 Robotic Inspection

After capturing an image from a certain location within the environment and obtaining a confidence map showing potential matches to a target object, the algorithm is enhanced by moving the imaging system to a secondary location to capture a second image from a different viewpoint for further inspection. If once again matches are found at the same locations, the confidence in the resulting target object locations is increased. Otherwise, if no match is found from a different viewpoint, the confidence in the original results will be decreased.

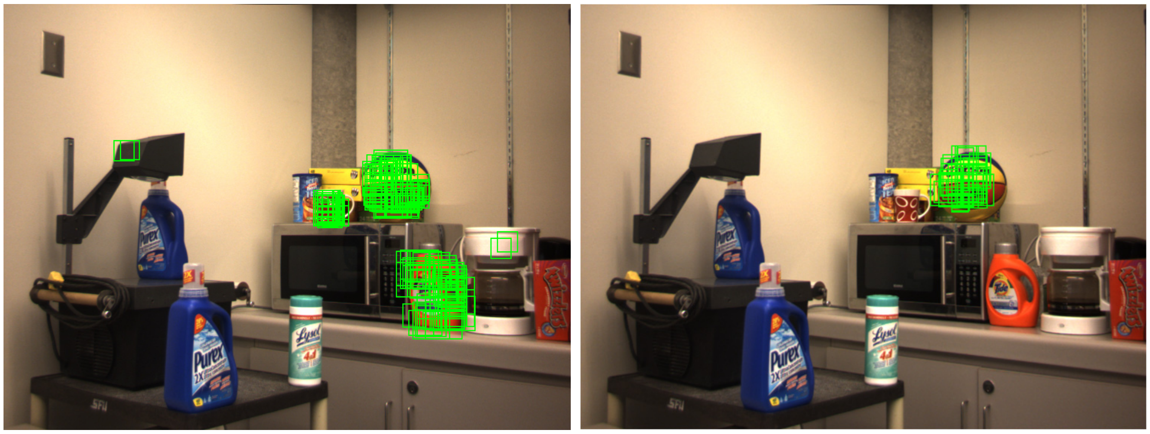

Figure 4.18: Before and After Triangular Constraints. Matching results are displayed before and after the triangular constraints are applied. Note that after the triangular constraints are applied, positive matches are only found in the vicinity of the ball in the scene.

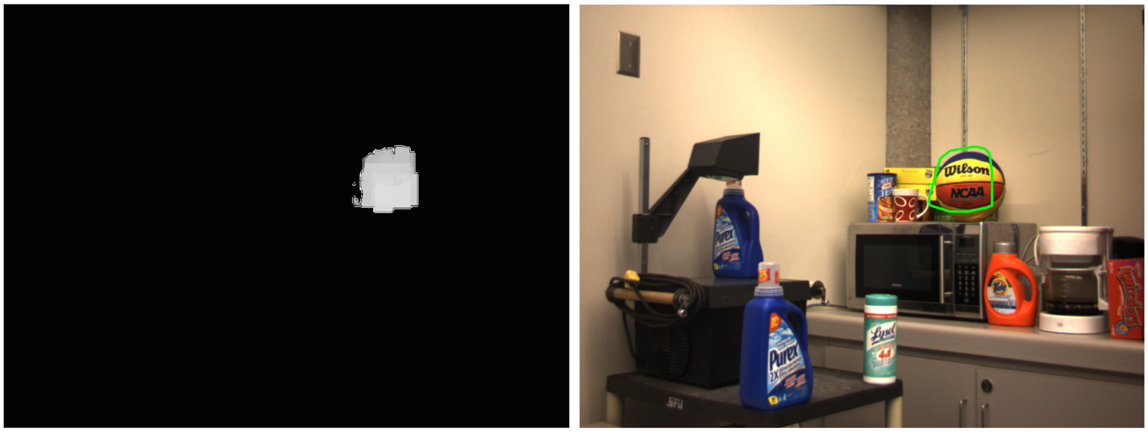

Figure 4.19: Confidence Map. *Left:* The confidence map generated from the histogram intersection scores of the matching patches from Figure 4.18. *Right:* The outline of the matching confidence region overlaid on the scene image. The target object has been successfully located.

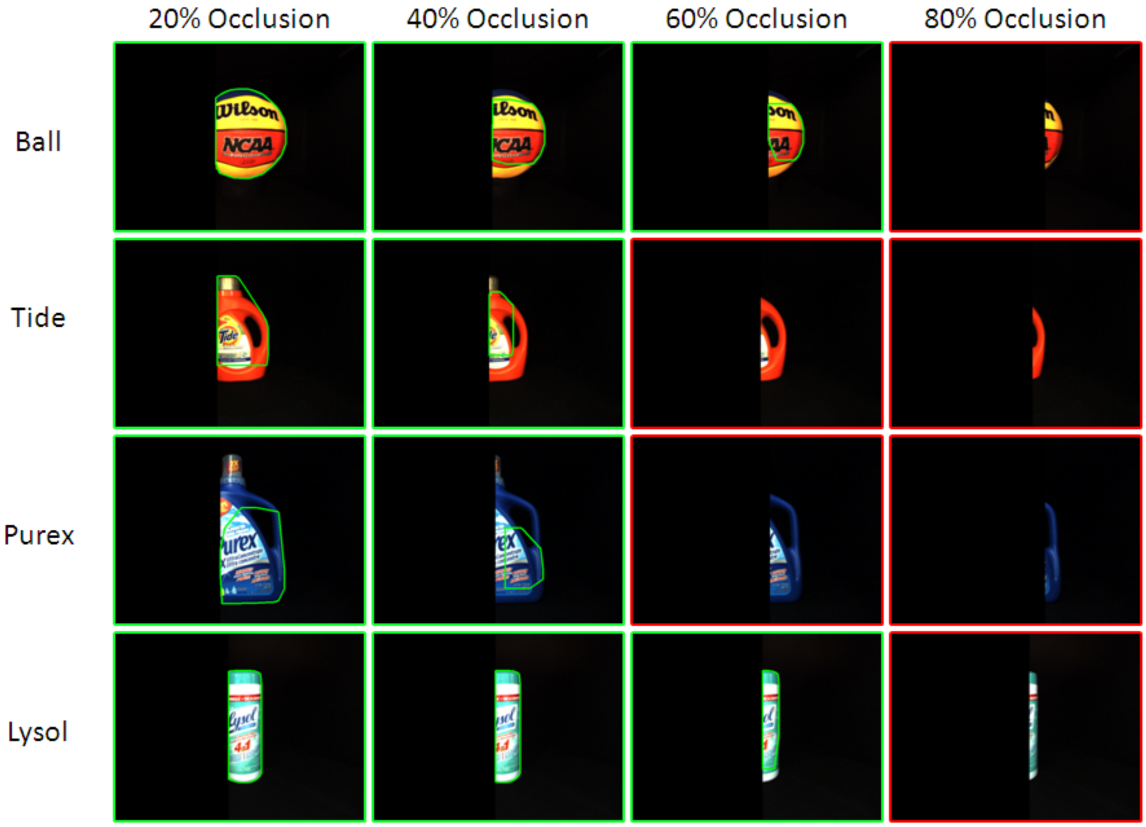

Figure 4.20: Occlusion Test. In each row of the figure, an object is searched for in a database image of itself with varying occlusion. Each object is successfully identified under 20% and 40% occlusion, two of the four objects are successfully identified under 60% occlusion, while no matches are found under 80% occlusion.

By using the stereo depth information captured from the BumbleBeeXB3 imaging system, the locations of the potential match regions generated in the object recognition process are obtained in three dimensions. Two dimensions are provided by the image coordinates while the third is provided by the stereo depth information of the distance between the lens and the region.

With the 3D locations of the potential matches known, it is possible to generate a secondary location for the robot platform to move and capture an image from a different viewpoint. The depth of all other regions in the room along with the sonar and infrared sensor data available on the PeopleBot platform may be used for navigational and collision avoidance purposes in reaching the secondary location.

The following subsections describe the process in transforming object recognition results and stereo depth information to 3D coordinates of potential matching regions. Also, the integration of results from object recognition processes from different viewpoints is explained and a rule for determining whether or not the object is present at the candidate locations is shown.

4.5.1 Spatial Representation

The pixel locations of matching regions in an image together with depth information from the stereo vision system provide 3D spherical coordinates. The three dimensions of the spherical coordinate system are inclination angle (ϕ), azimuth angle (θ), and radial distance (r). The distance from each object in the environment to the camera is obtained by the stereo processing stage and provides the radial distance coordinate. The camera used in the BumbleBeeXB3 imaging system has a $50\,^\circ$ horizontal field of view and a $30\,^\circ$ vertical field of view. With the center pixel of the image being the reference point, an angle of inclination and azimuth angle can be interpolated for each pixel in the image based on the vertical and horizontal distance from the reference point.

With the spherical coordinates obtained for each pixel within the confidence map, the following spherical-to-Cartesian transformation is applied to obtain the Cartesian coordinates of the matching regions.

$$x = r \sin\theta \cos\phi \tag{4.4}$$

$$y = r \sin\theta \sin\phi \tag{4.5}$$

$$z = rcos\theta \tag{4.6}$$

A confidence map is then regenerated in a birds-eye-view manner by using two of the three Cartesian coordinates. This map shows the relative positions of the matches within the environment. Figure 4.21 shows a birds-eye view confidence map generated from stereo and matching results. The horizontal field of view boundaries are also indicated on the map. For this work, confidence maps were generated with a resolution of $1pixel : 1cm$.

After obtaining the 3D coordinates of the potential matches within the environment, it is possible to determine a secondary location within the environment at which a second image may be captured of the potentially matching region. Depth information from the stereo imaging system may be used in conjunction with sensor data from the robot to plan a path to the secondary location and avoid obstacles within the environment. Robot navigation and obstacle avoidance are large and complex fields of research with many successful approaches. The development and integration of such an algorithm is outside the scope of the research presented in this work. In place, a secondary location is chosen by the user and a second scene image is captured from the specified location. This process can be repeated multiple times from several different viewpoints. The integration of the matching results from all scene images captured from different viewpoints is used to produce robust matching results.

4.5.2 Multiple Viewpoint Integration

Once a second image of the environment is captured from a different viewpoint, the object recognition process is repeated to locate potential matches within the new image. If matching regions are found, the locations of the potential matches in the second image are found in Cartesian coordinate space relative to the new location of the robot. A reference frame transformation is then applied to align the coordinates of the two views.

Once the two views are aligned to the same reference frame, the confidence information in the intersecting regions are merged by taking the average confidence rating at each viewable coordinate. It is critical that only viewable regions are merged so that negative matching results for objects which are occluded in one view are not merged with positive matching results for objects which are visible in another view. This merging of confidence regions causes regions found in only one of the views to be decreased in confidence while regions found in both views to remain. Regions which fall below a 0.5 confidence rating after the merging operation are eliminated from the confidence map. Figure 4.22 demonstrates the

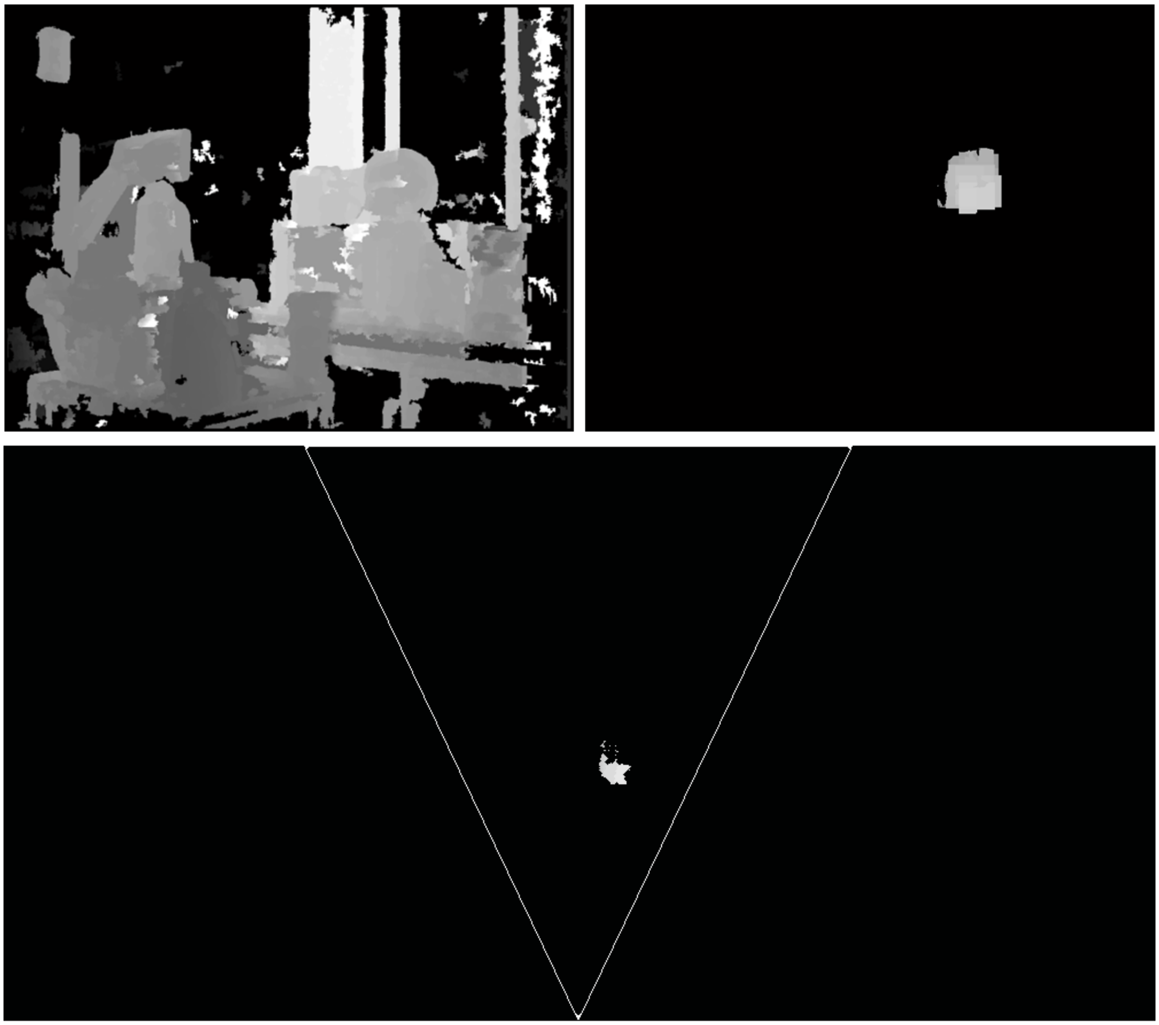

Figure 4.21: Spatial Representation. Through a spherical-to-Cartesian coordinate transformation, a birds-eye-view map is generated. *Top-Left:* Stereo depth map of scene. *Top-Right:* Corresponding confidence map generated from histogram intersection matching scores of the matched patches (Target Object: Basketball). *Bottom:* Birds-eye View map generated from the image coordinates and stereo depth map. The horizontal field of view limits are also outlined.

integration of object recognition results from two viewpoints. In this example, the basketball object is successfully located in both viewpoints and the combined results show a verified match in the birds-eye view confidence map.

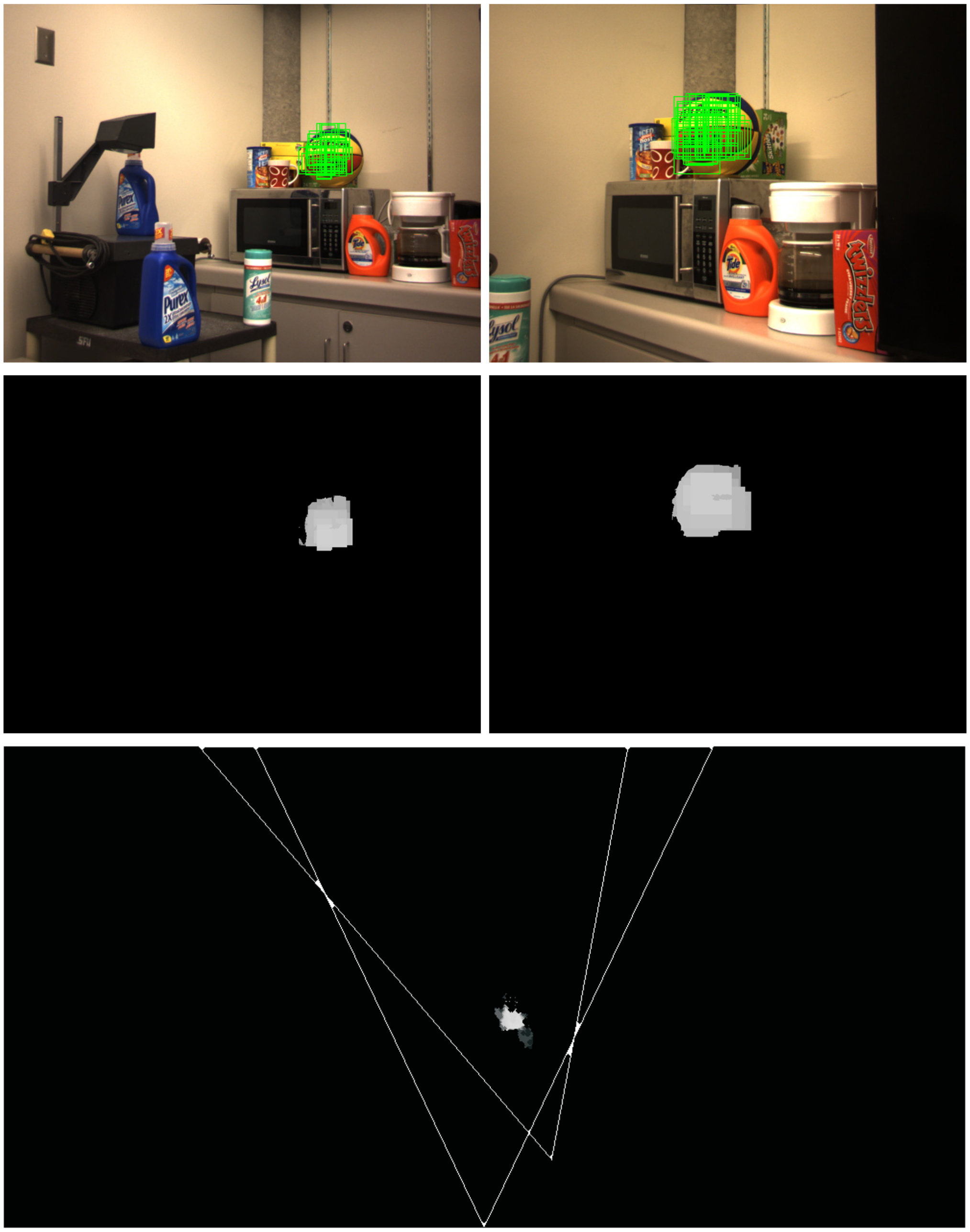

Figure 4.22: Multiple Viewpoint Integration. Object recognition results from two scene images captured from different viewpoints are combined to verify results. *Top:* Two scene images captured from different viewpoints. Matching image patches are outlined (Target object: Basketball). *Middle:* Corresponding confidence maps generated from two views. *Bottom:* The combined birds-eye view confidence map of both viewpoints. The matching regions in the intersection of the two views are averaged and the matching results are verified.

Chapter 5

Experimental Results

To evaluate the performance of the object recognition algorithm presented in this work, several experiments were carried out. First, the algorithm was tested in a controlled environment without any image variations such as shading, blurring, specularities, illumination changes, or background clutter. Next, real world scenes were captured with objects placed at various locations within the scenes. In these tests the scenes contained significant clutter and the illumination was uncontrolled. Finally, the multi-view object recognition algorithm was implemented on the mobile robot. This evaluated the integration of object recognition results from multiple viewpoints and the ability to locate objects within an unknown environment. These three experimental scenarios are explained in further detail with the corresponding experimental results in the following sections.

5.1 Cross-Database validation

In order to evaluate the effectiveness of the complete object recognition algorithm presented in Chapter 4, tests were first carried out in a controlled environment. By eliminating background clutter and any photometric variations, ideal conditions are achieved.

To achieve such a controlled environment, the database images of the objects were used as scene images. This means that the information extracted from the database images of a certain object was searched for in the same database images in order to produce the object recognition results. In total, 112 objects were tested, each being searched for in the database images of all 112 objects. Two experiments using this framework were carried out. First, only a single view of each object was used as a scene image. Second, three views of each object were used as scene images taken from distinct viewpoints.

Two key factors of the object recognition algorithm are examined in the cross-database validation experiments: descriptiveness and distinctiveness. For each target object, the information extracted from the database images should be descriptive enough so that the target object is successfully identified in the database image of itself. Also, the extracted information should also be discriminative enough so that target objects are not incorrectly matched to database images of other objects.

5.1.1 Single-View Cross-Database Validation

For each of the 112 target objects used, the first cross validation experiment used the 0 ° database image of each of the 112 objects as a scene image. Essentially, these images are of environments containing nothing but the database objects located at a certain distance from the camera. After applying the object recognition algorithm, a final confidence score is generated for each test ranging between 0 and 1. Scores below 0.5 are set to zero as a non-match is concluded while scores above 0.5 indicate a match.

To test the descriptiveness of the SDH histograms and the corresponding extraction and clustering techniques used to represent each database object, each object was first searched for in its own database image. Naturally, each object should be successfully identified; this is a prerequisite for any object recognition algorithm. If the information extracted from a database image of an object is not successfully matched to that same database image, the information extracted does not sufficiently represent the appearance of the object.

Figure 5.1 presents the results of 6 of the database objects when searching for each object in its own database image. In total, all of the 112 objects were successfully located with an average matching score of 0.95. The scores of all objects are plotted in Figure 5.2.

To test the distinctiveness of the SDH histograms and the corresponding extraction and clustering technique used, each database object was also searched for in the database images of all other objects. Ideally, while searching for a specific object in the database images of all other objects, no successful matches are found (zero false-positives). This ideal result would show that the SDH histogram representations of the objects are extremely distinctive as a match is found only in its own database image while all other objects are rejected. However, if a database object is incorrectly matched to a large number of other database objects, the representation would have little distinctiveness as many false-positives were found.

For each of the 112 database objects, a search was performed on the images of each of the

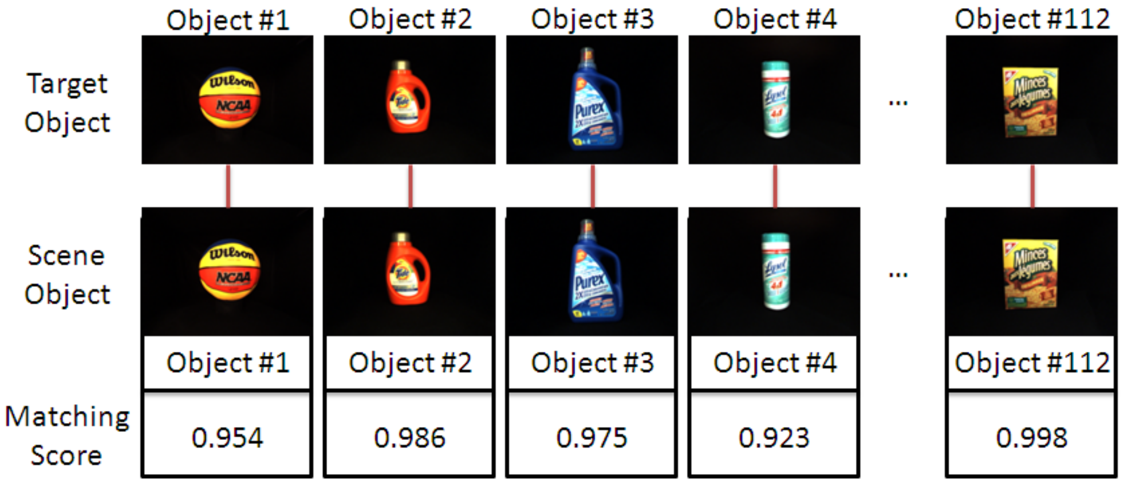

Figure 5.1: Single-View Database Object Testing. Each object is searched for in a scene image of itself. Under these ideal conditions, a successful match is found for each object. 112 objects are tested in total.

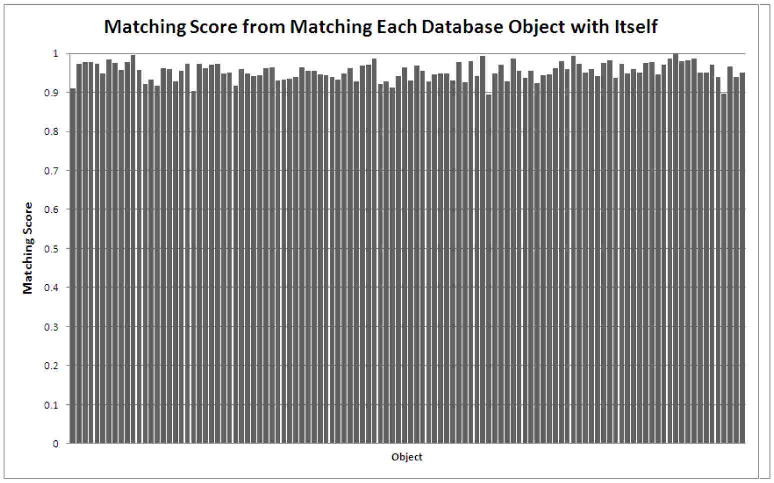

Figure 5.2: Single-View Database Object Test Results. Each object was searched for in a single database image of itself and the resultant matching score (between 0 and 1) is plotted for each of the 112 objects. A high matching score is obtained for each object.

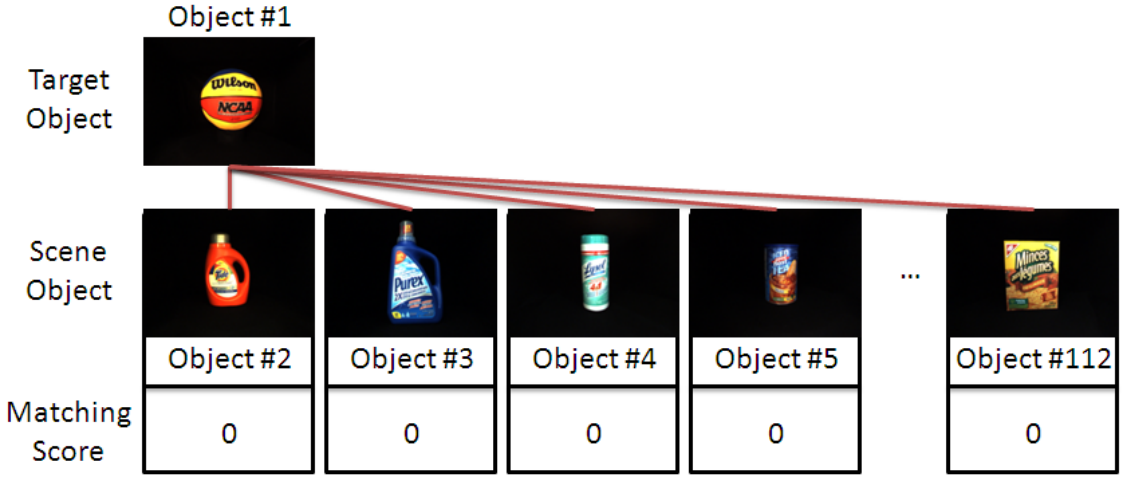

Figure 5.3: Single-View Cross Database Testing. Each object was searched for in a single database image of all of the 111 remaining database objects. Ideally, no successful matches are found. Here five examples are shown of which no successful matches are found.

111 remaining database objects. In total, 12432 tests were carried out. Figure 5.3 shows an example of the cross-database tests performed for one of the 112 database objects. Again, for each test, a confidence score between 0 and 1 was generated and any score greater than 0.5 was indicated as a match.

For each target object, the average number of objects that were incorrectly matched to the target object was 7.9 out of 111 (7.1% false-positive rate). To further investigate the cause of the false-positive matching results, the 10 objects with the most false positive matches were analyzed. Figure 5.4 shows the database images of these 10 objects. The common trait in these objects is that they are all composed of only one dominant color.

The results presented here demonstrate an important limitation in the object recognition algorithm presented in this work. As mentioned in Section 4.3.1, the extraction of the SDH histograms captures information regarding the color composition of the object as well as information of which colors occur within the same vicinity as other colors. If an object containes more than one color, the proportions of each color as well as the spatial layout of the colors in the object are retained in the SDH histogram extraction process. However, if an object contains only one color, only the information that that color exists in the objects appearance is retained. This limitation causes target objects of only one color to be incorrectly matched to other database objects composed of the same or similar color.

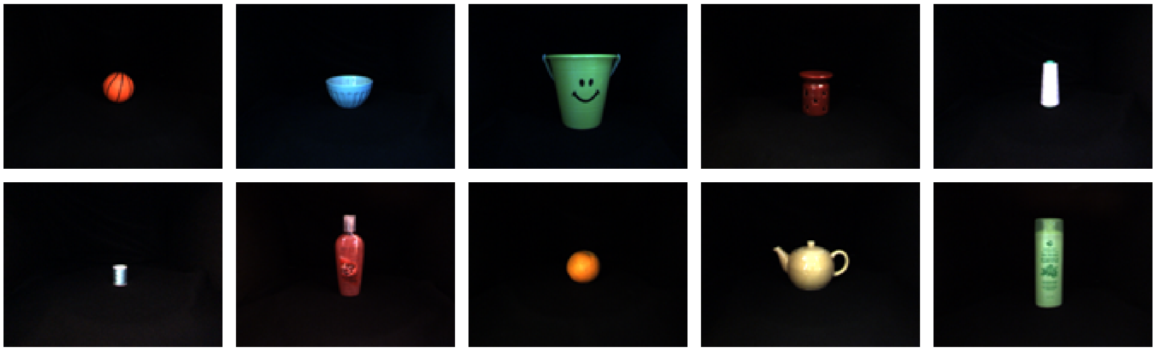

Figure 5.4: Objects Generating False-Positive Matches. The 10 objects with the most false positive matches within the cross-database testing are shown. The common trait among these objects is that they are composed of only one dominant color. This demonstrates a an important limitation in the object recognition algorithm.

In some cases, only one view of an object contains only a single color. By looking at the object from multiple viewpoints, a more distinctive representation of the object is obtained and the algorithm can distinguish between objects of the same color. For instance, if the front of the target object is red while the back is blue, it would most likely be incorrectly matched to an image of an scene object that is completely blue. However, by looking at the scene object from multiple viewpoints, no match would be made to the red region at the back and thus the scene object would be rejected. The next section incorporates multiple viewpoints for the database images to improve the cross-database validation results.

5.1.2 Multi-View Cross-Database Validation

To provide increased distinctiveness, the use of the multi-view object recognition is applied to the cross-database validation experiments. To test the multi-view object recognition algorithm presented in this work, three views of each object from the database were used as scene images. Since the position at which each database image was taken is known, the relative position each scene image was captured from is also known and the combination of views is easily achieved. As described in Chapter 4, the object recognition results of all three scene images are combined by taking the average confidence score at each location in the 3D environment. Figure 5.5 shows the birds-eye view setup of the multi-view cross-database validation test. The relative locations of the three views are shown as well as the three scene images for the *basketball* object.

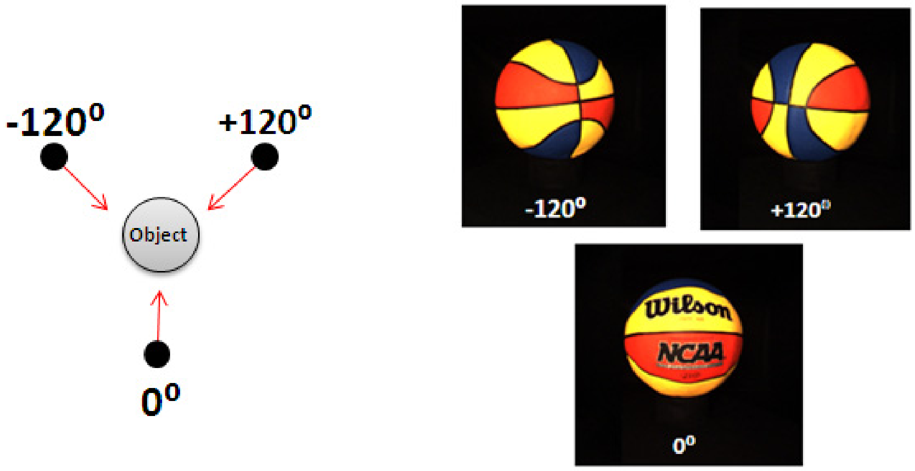

Figure 5.5: Multi-view Cross-Database Validation Setup. *Top:* The birds-eye view layout of the three viewpoints of the scene object used to search for the target object. *Bottom:* The corresponding images of the three viewpoints for the *basketball* object.

Figure 5.6 shows some examples of the cross-database validation tests carried out using multi-view inspection. Again, 112 objects were tested against each of the 111 remaining objects in the database. Using the multi-view integration algorithm, the average number of objects that were incorrectly matched to the target object was 3.2 out of 111 (2.9% false-positive rate). Comparing these results to those obtained using only a single view of the scene object, the false positive rate was greatly reduced from 7.1% to 2.9% as seen in Figure 5.7.

In these controlled experiments, background clutter and photometric variations were eliminated. To test the true efficacy of the object recognition algorithm, scene images of a real-world environment must be used.

5.2 Object Recognition in a 3D Environment

The common challenges that object recognition algorithms must overcome are those of illumination variations within the scene, background clutter in the scene, occlusion of objects within the scene, variations in object pose, and variations in the distance between the camera and the objects in the scene (scale). To test the robustness of the object recognition algorithm presented in this work, objects were placed in three cluttered scenes with uncontrolled

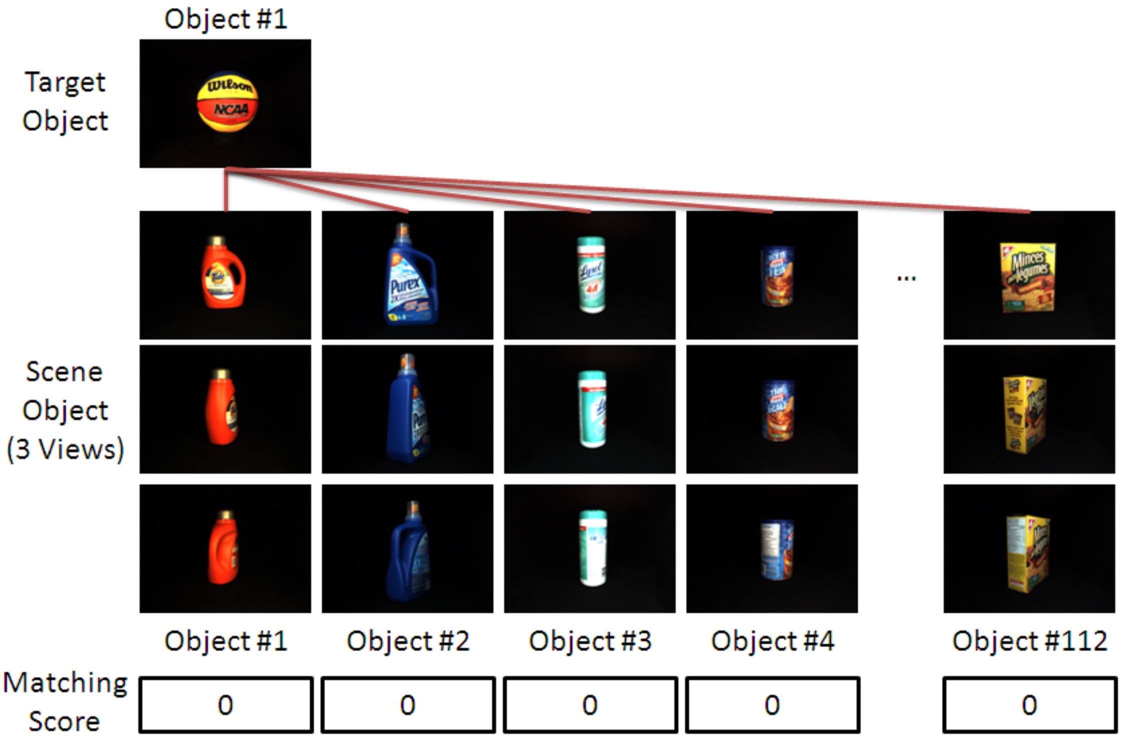

Figure 5.6: Multi-View Cross-Database Testing. Each object was searched for in three database images of each of the 111 remaining database objects. Ideally, no successful matches are found. Here five examples are shown of which no successful matches are found.

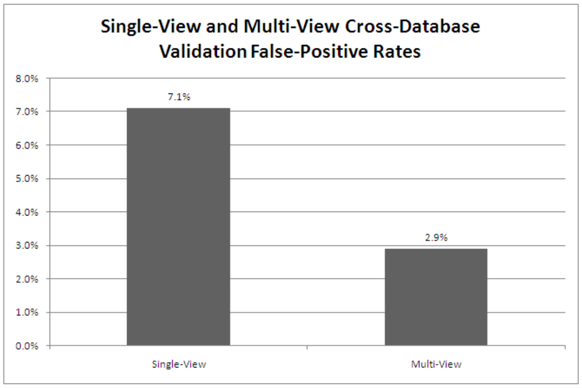

Figure 5.7: Comparison Between Single-View and Multi-View Matching Results. The false positive rate was greatly reduced from 7.1% to 2.9% when using multiple viewpoints.

illumination at different locations and poses.

Figures 5.8, 5.9, and 5.10 show the results for the object recognition algorithm applied to 3D environments. For each scene, four objects were searched for in independent tests. The top row of each figure displays the target object of each test. The second row highlights the matching image patches for each search. The third row highlight the matching image patches that persist after the application of the triangular constraints. The corresponding confidence map is displayed in the fourth row and the matching region is outlined in the fifth row.

In Figure 5.8, each target object was successfully located. In the case of the *basketball*, *Tide bottle*, and *Lysol container* objects, a single instance was placed in the scene and each was successfully identified. In the case of the *Purex bottle*, two instances were placed in the scene and both were successfully identified. This shows the ability of the algorithm to locate multiple instances of objects within the scene.

In Figure 5.9, again, each target object was successfully located. While many image patches extracted from the scene were highlighted as potential matching regions, only the regions located on the actual object remained after the application of the triangular constraints. In the case of the third column, the object located in the scene is greatly occluded by another object. In this case, the algorithm matches the visible regions of the object and is not dependent on all regions of the object being visible. In the case of the fourth column, the target object is very dark and contains little color information. Regardless, the object is still successfully located in the scene.

The third and final scene, shown in Figure 5.10, demonstrates the performance of the algorithm on objects with low saturation in the contained color information. The first column shows a target object that is successfully located in the scene image. The second and third columns show objects which have similar color content to each other as well as the surrounding bookcase background. In these cases, the target object was identified but along with other regions in the scene that are not correct. These false positive matches arise from the similarity in color content between the target objects and other regions in the scene. While most of the matching image patches were rejected after the triangular constraints were applied, several regions in the image persisted. The fourth column in this figure shows a search for an object that is not located in the scene. Here, no match is found in the scene despite the low saturation and lack of color information within the target object.

The results of these real-world scene experiments show the robustness of the algorithm

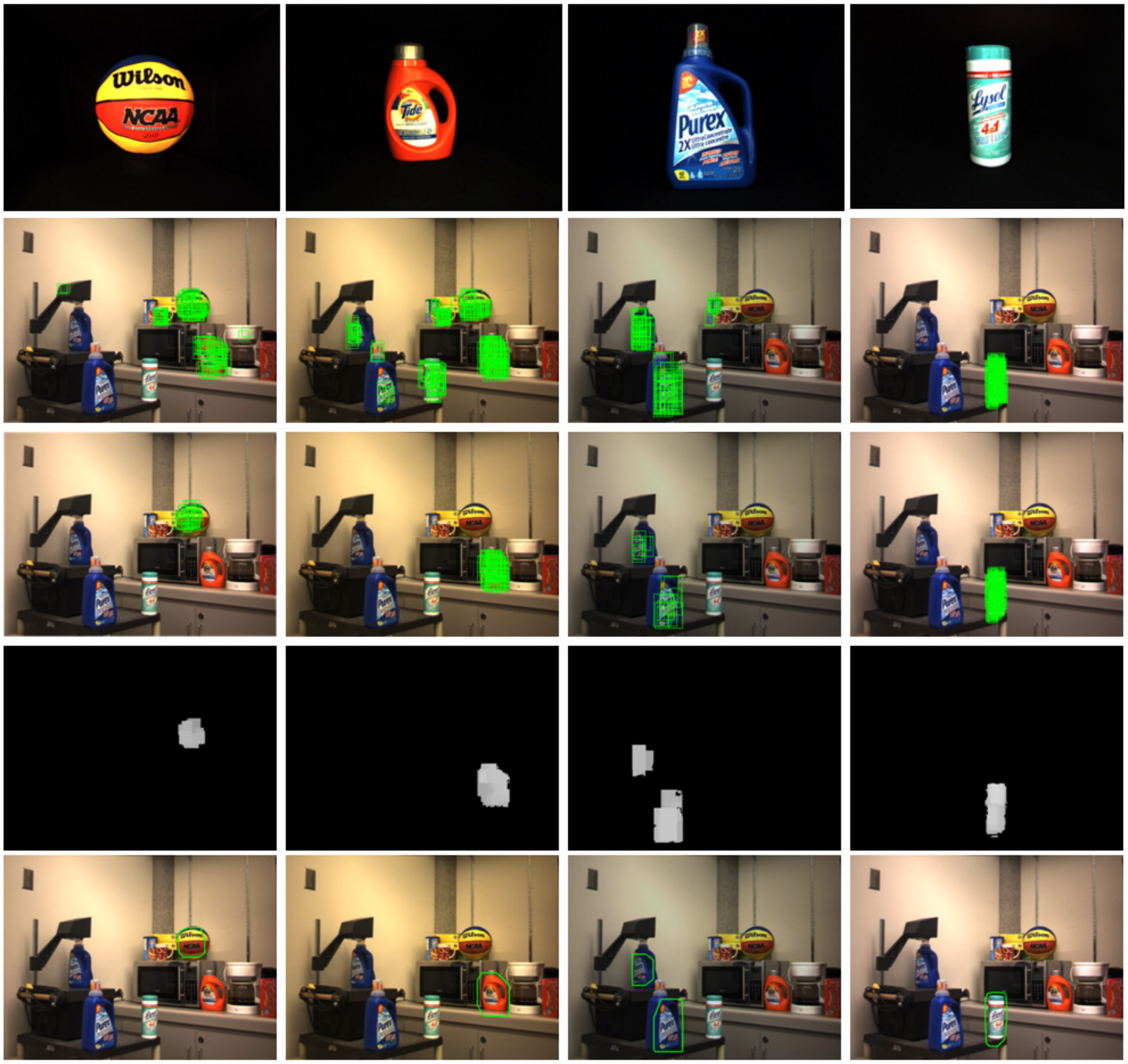

Figure 5.8: Object Recognition Results (Scene 1). *Row 1:* Target Object. *Row 2:* Matched scene image patches. *Row 3:* Matched scene image patches passing triangular constraints. *Row 4:* Resultant confidence map. *Row 5:* Matching regions outlined.

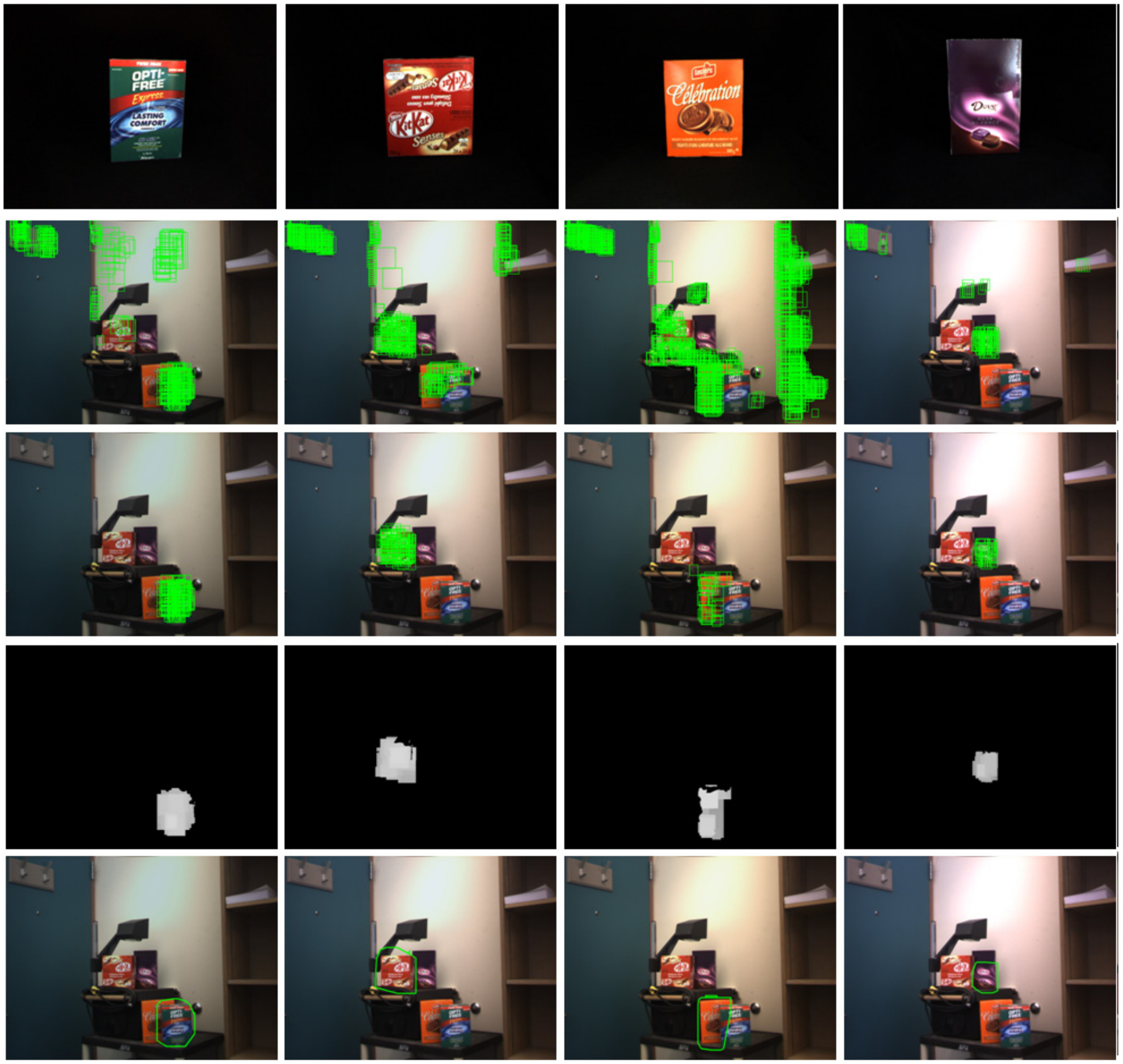

Figure 5.9: Object Recognition Results (Scene 2). *Row 1:* Target Object. *Row 2:* Matched scene image patches. *Row 3:* Matched scene image patches passing triangular constraints. *Row 4:* Resultant confidence map. *Row 5:* Matching regions outlined.

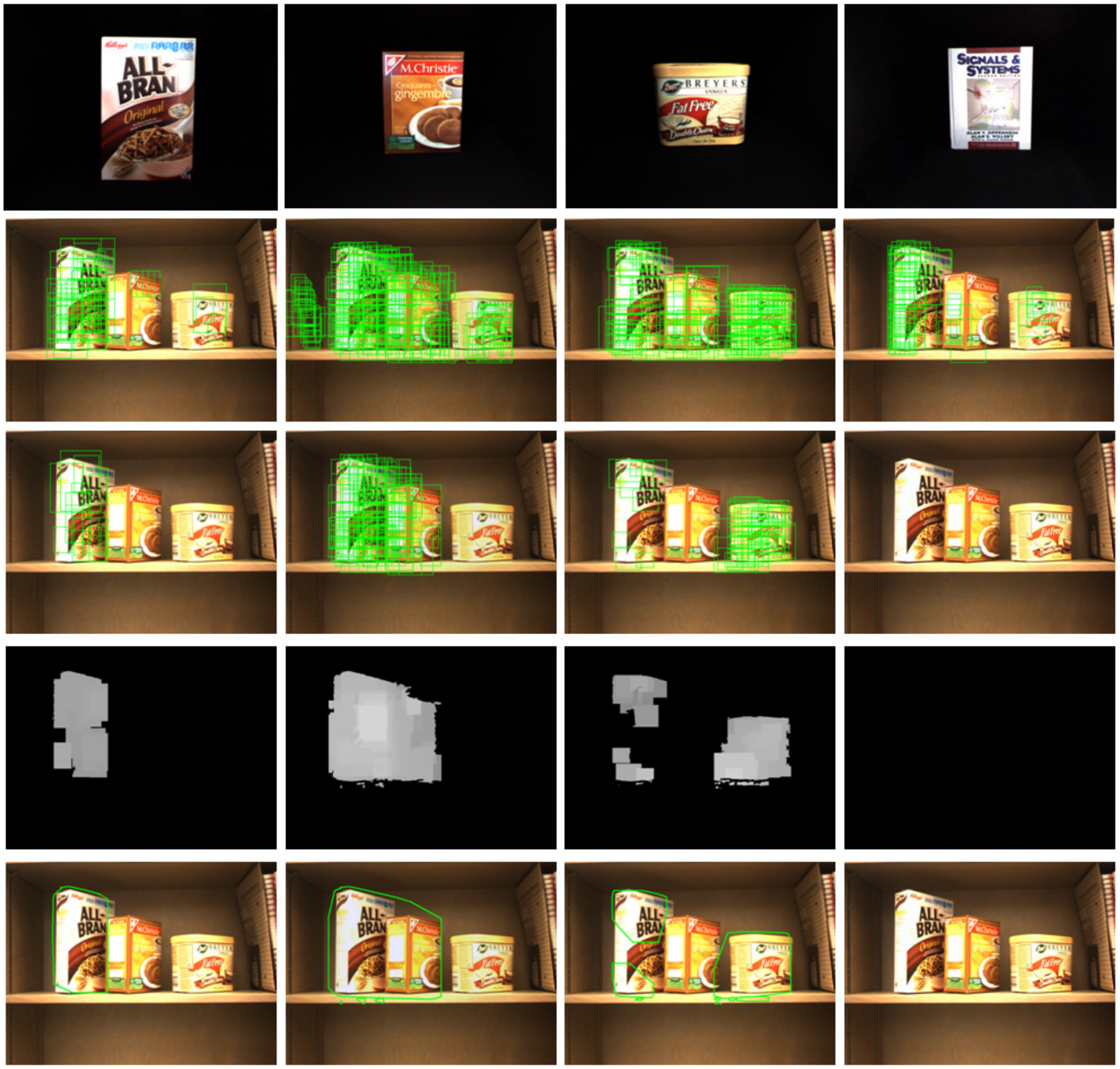

Figure 5.10: Object Recognition Results (Scene 3). *Row 1:* Target Object. *Row 2:* Matched scene image patches. *Row 3:* Matched scene image patches passing triangular constraints. *Row 4:* Resultant confidence map. *Row 5:* Matching regions outlined.

to several variations. While the database images of the objects were captured under incandescent illumination, the scene was illuminated by a non-uniform combination of sunlight from the windows and fluorescent lights within the room. Also, several regions in the images show significant amounts of shading. The large scale variation between the database object and the instances of the objects in the scene is also overcome with the use of depth information from the stereo vision system. The ability of the algorithm to locate objects with different poses is also shown. Objects were placed at differing angles within the scene to emulate a more typical scenario. The successful identification of the objects despite these variations in a scene with significant background clutter demonstrates the robustness of the object recognition algorithm.

While a single scene image may generate a set of matching regions, each has a certain confidence value. In the case of false-positive or false-negative matching regions, further inspection of the scene is needed to increase the confidence of the results and to eliminate such false matches or non-matches.

5.3 Object Recognition in a 3D Environment using Multiple Viewpoints

The incorporation of a mobile robot platform allows the environment to be inspected from multiple viewpoints. In Chapter 4, the experimental setup of the BumblebeeXB3 stereo imaging system mounted atop the PeopleBot mobile robot was shown in Figure 4.3. Two types of experiments were carried out to demonstrate the use of multiple viewpoints in the object complete recognition system. First, multiple views of the scenes from the previous section are incorporated to improve the recognition results. Second, a complete search algorithm for locating hypothesis regions within an unknown environment and further inspecting these regions from multiple viewpoints is demonstrated.

5.3.1 Multi-View Inspection

The process of incorporating matching results from images captured from different viewpoints was explained in detail in Section 4.5.2. To demonstrate the effectiveness of this approach, the scenes from the previous section were inspected from two additional viewpoints and the resultant matching regions identified while searching for a specific object were incorporated.

The scene from Figure 5.9 was first inspected and the four objects placed in the scene were searched for independently. Figures 5.11 to 5.14 show the matching results for each of the objects within the three views of the scene as well as the resultant birds-eye view confidence map for each case.

Figures 5.11 and 5.12 show that each object was successfully located in each of the three views of the scene. The resultant confidence map thus shows the coordinates of object within the environment based on the depth information obtained from the stereo camera.

Figures 5.13 and 5.14 both show that in one of the three views inspected, the target object was not successfully identified. In each case, the object was severely occluded and thus no matching regions could be found. Due to the fact that the objects were still found in the remaining two views of the two scenes, the correct location of the object was successfully identified in the birds-eye view confidence map. These two cases demonstrate the advantage of multi-view inspection of an unknown environment.

The scene from Figure 5.10 was also inspected using multiple viewpoints as seen in Figures 5.15 to 5.17. Again, images captured from two additional viewpoints were searched for using the object recognition algorithm and the multi-view inspection algorithm was used to incorporate the matching results from all views.

The object being searched for in Figure 5.15 was successfully located in the initial view from Figure 5.10. However, in one of the two additional views, the object was not successfully located. As the side of the object has very little color information, the algorithm failed to match any SDH histograms to the image. Regardless, using the multi-view inspection algorithm, the coordinates of the object were correctly determined in the birds-eye view confidence map.

Figure 5.16 shows the results after searching for the box located in the middle of the bookshelf. In the initial view, as seen in Figure 5.10, the object to the left was also identified as part of the matching region and both were outlined as matching regions. With the incorporation of multi-view inspection, the object was successfully located in the two additional views. Therefore, after merging the matching results of the three images, the intersection of the matching regions identify the correct 3D coordinates of the object in the birds-eye view confidence map.

Lastly, the object on the right of the bookshelf was searched for in Figure 5.17. The initial view from Figure 5.10 shows that the correct location was identified as a matching

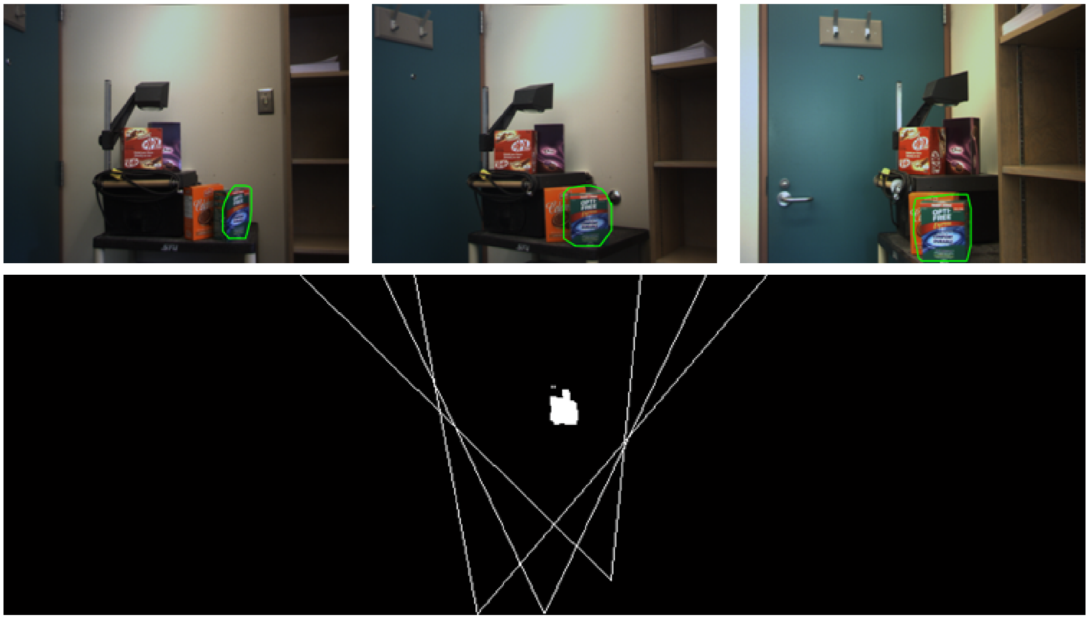

Figure 5.11: Multi-View Inspection Results - Opti-Free. The mathching results are shown along with the resultant birds-eye view confidence map.

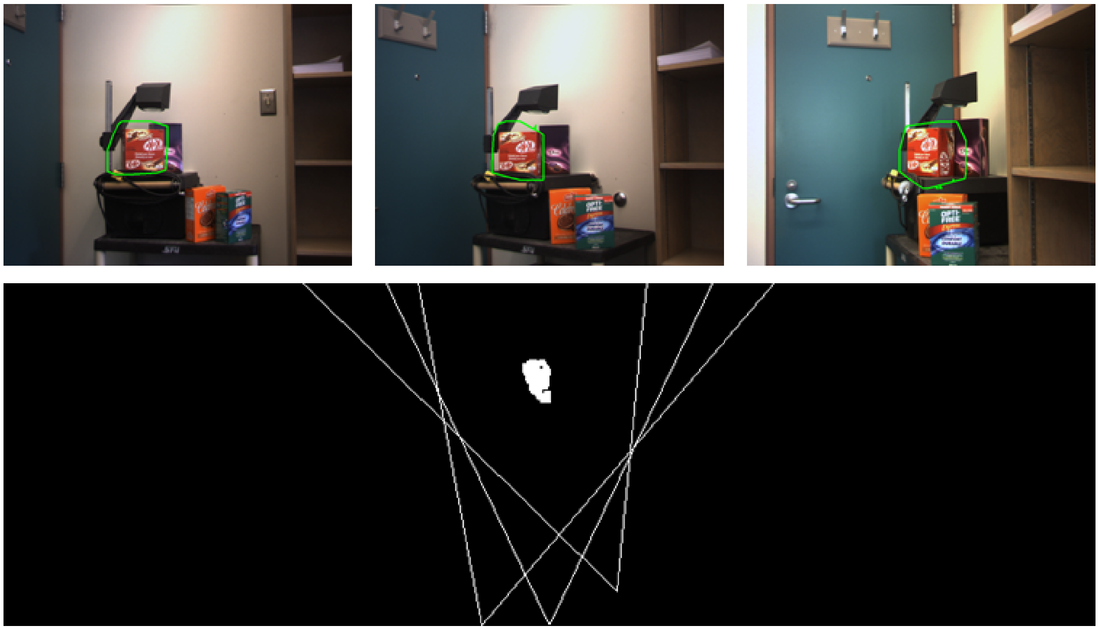

Figure 5.12: Multi-View Inspection Results - Kit-Kat. The mathching results are shown along with the resultant birds-eye view confidence map.

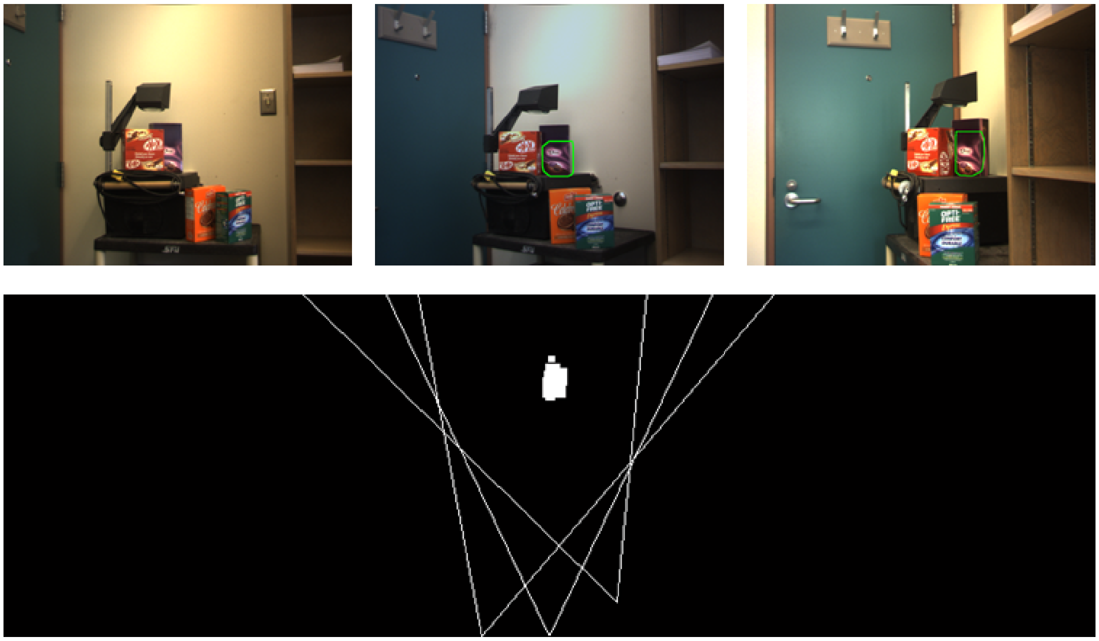

Figure 5.13: Multi-View Inspection Results - Dove. The mathching results are shown along with the resultant birds-eye view confidence map.

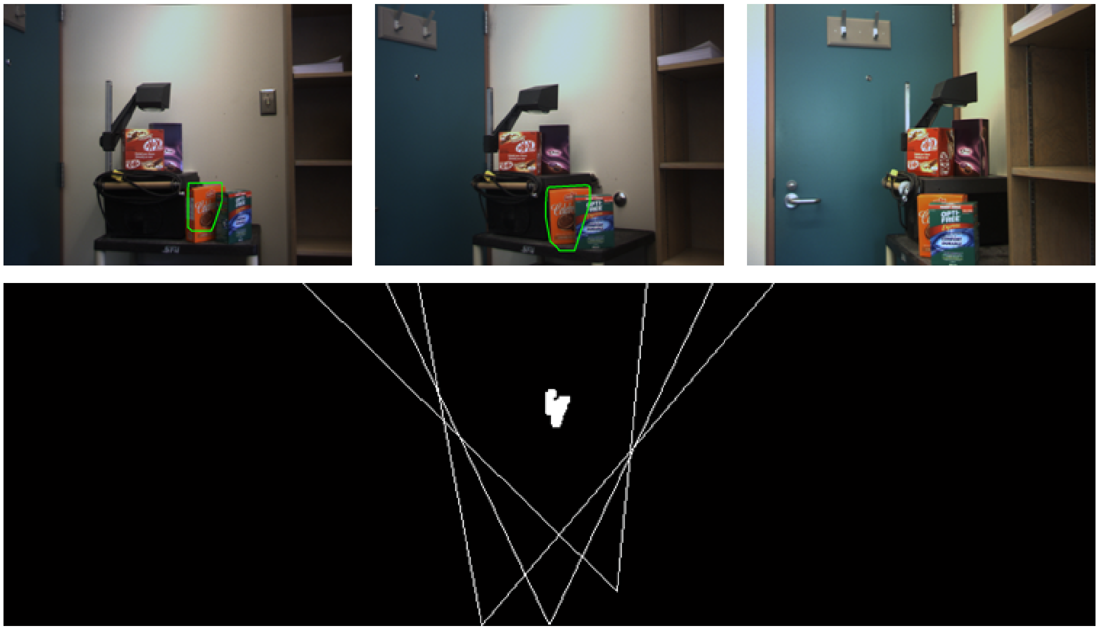

Figure 5.14: Multi-View Inspection Results - Cookies. The mathching results are shown along with the resultant birds-eye view confidence map.

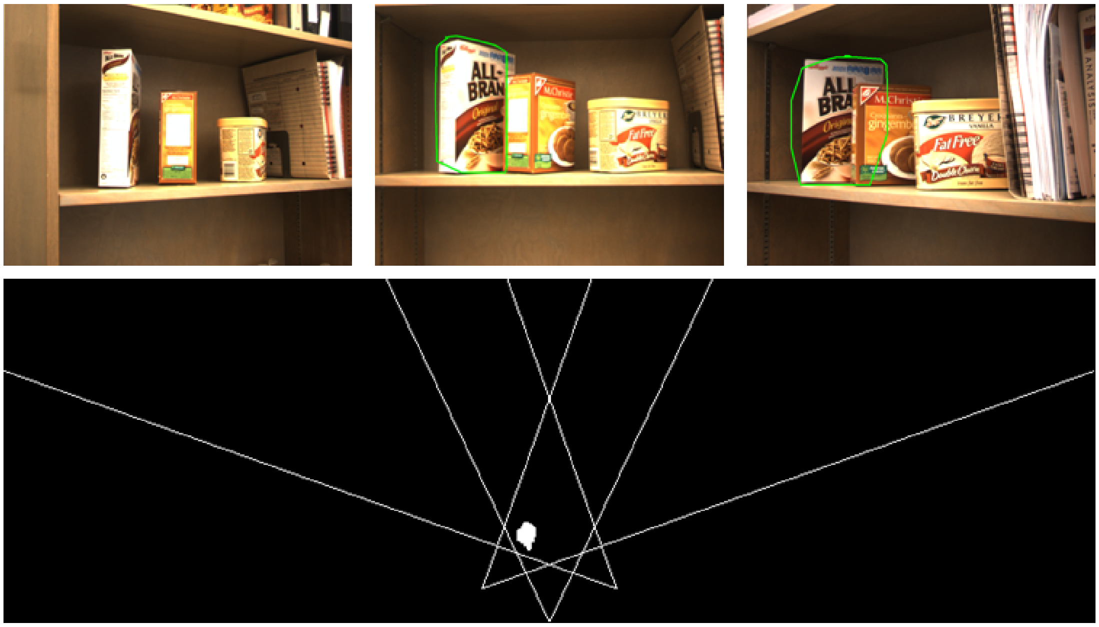

Figure 5.15: Multi-View Inspection Results - All Bran. The mathching results are shown along with the resultant birds-eye view confidence map.

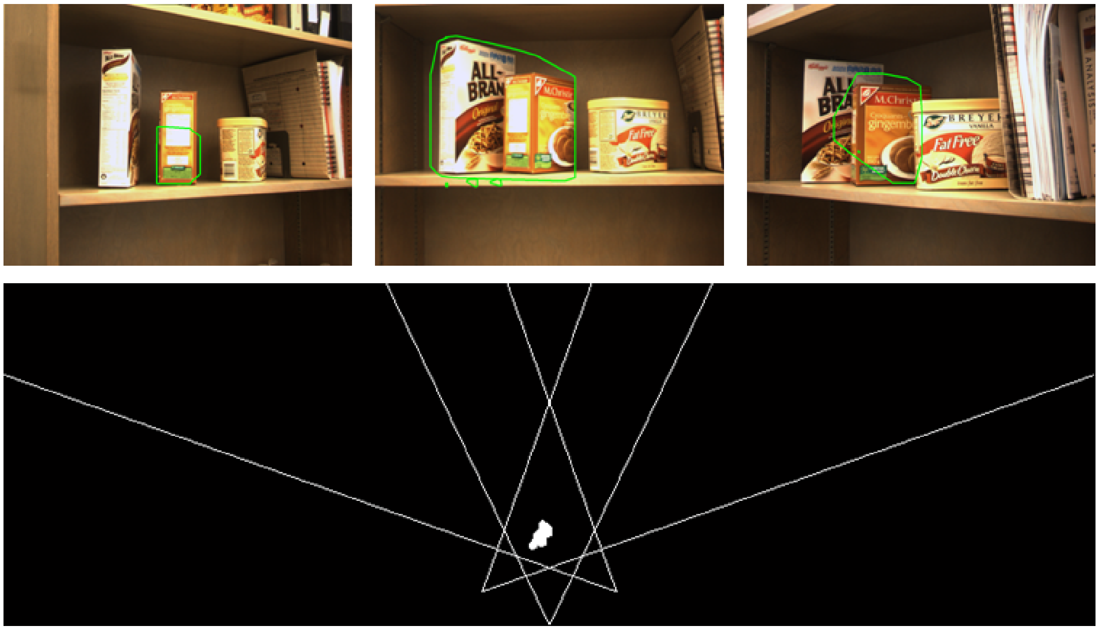

Figure 5.16: Multi-View Inspection Results - Gingersnaps. The mathching results are shown along with the resultant birds-eye view confidence map.

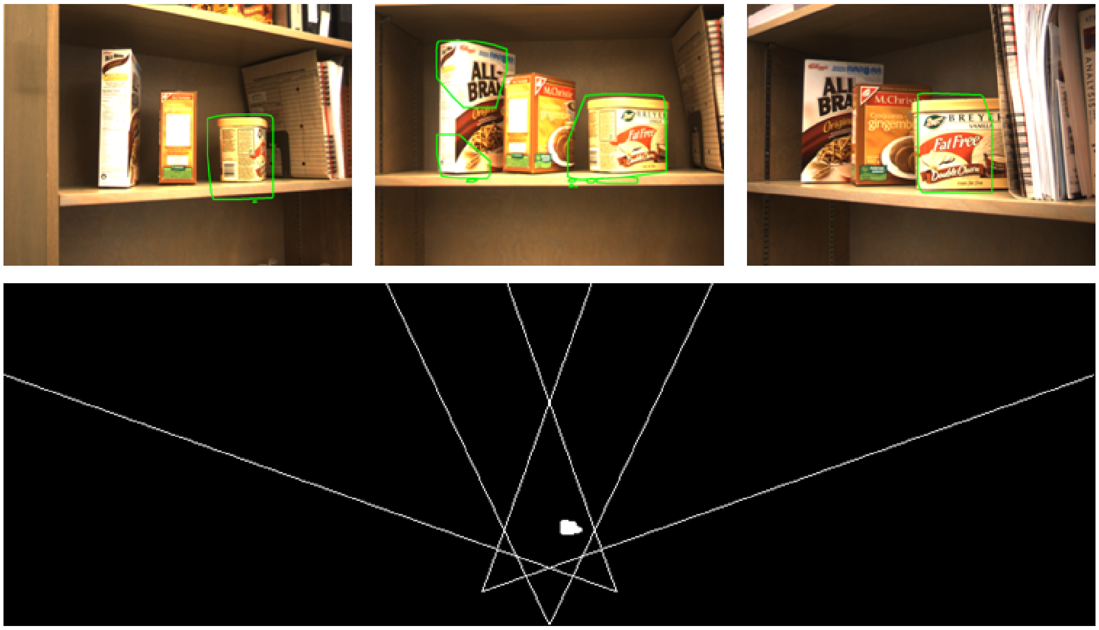

Figure 5.17: Multi-View Inspection Results - Ice Cream. The mathching results are shown along with the resultant birds-eye view confidence map.

region within the scene image along with two false positive regions within the leftmost object. Again, by using multiple views, Figure 5.17 shows that the false positive regions are not found in the two additional images whereas the correct object is. The resultant birds-eye view confidence map indicates the correct coordinates of the target object.

5.3.2 Multi-View Inspection using a Mobile Robot Platform

To demonstrate the ability of the system to be implemented on an autonomous mobile robot, the following algorithm was developed to move the robot about the environment searching for hypothesis regions and further inspecting these regions using the multi-view inspection algorithm.

From the initial location of the robot upon powering on, a scan of the surroundings is carried out first. The robot captures three images: straight forward, rotated 45 ° in a counter clockwise direction, and rotated 45 ° in a clockwise direction. As the camera used on the BumbleBeeXB3 imaging system has a 50 ° horizontal field of view, there is some overlap in the images. The object recognition algorithm is performed on each of the captured images and a birds-eye view confidence map is generated based on the matching results.

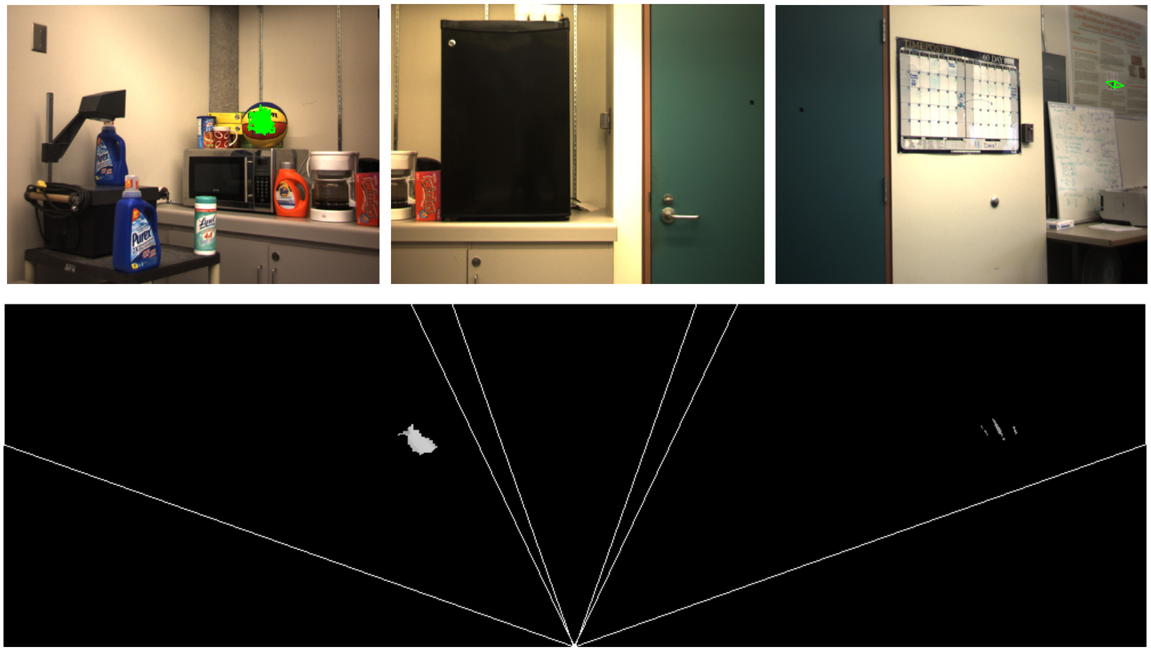

Figure 5.18: Multi-view Room Scan. *Top:* three images taken spanning a 135 ° field of view, *Bottom:* the corresponding birds-eye-view map generated from the confidence regions in all three images.

In searching for the *basketball* object, Figure 5.18 shows an initial scan of a surrounding environment as well as the corresponding confidence map of the environment. Note that the results of each image is combined into a single confidence map representing the entire environment. From these initial images, there are two matching regions. One matching region is a true positive region identifying the true location of the *basketball* while the second region corresponds to a false-positive match located on the wall within the scene.

To further inspect each matching region to verify the existence of the ball, the robot is moved to a secondary location where further images are captured. As mentioned previously, the calculation of an optimal secondary location to move to, as well as the incorporation of navigational and obstacle avoidance algorithms, is beyond the scope of the research presented here. In place, the robot moves to a somewhat arbitrary secondary location in which the initial matching regions remain viewable. The exact location of the secondary position is not of high importance as long as the matching regions are viewable from an angle distinguishable from the original room scan viewpoint.

Once at the secondary position, the robot rotates to face the coordinates of each matching region. An image is captured and the object recognition algorithm is applied once again. The resulting confidence maps are transformed to birds-eye view format and merged with the existing map of the environment. The confidence score of matching regions which are located within the intersection of two or more views are averaged and the resulting confidence score is utilized. After averaging the results of multiple views, regions with confidence scores below 0.5 are discarded while regions with scores above 0.5 persist.

This process of moving to new locations and capturing images of matching region from different viewpoints is repeated until a specific terminating condition. The confidence scores of each matching region are continually combined (using the mean function) after application of the recognition algorithm from each viewpoint.

Figure 5.19 shows the secondary and tertiary viewpoints along with the confidence maps generated from each captured scene image. The first two images in this figure are taken from the same secondary location (at different orientations) capturing each of the initial confidence regions. The last two images are captured from a third location, again of the two initial matching regions. Note that the secondary and tertiary images captured of the matching region identifying the true location of the *basketball* again successfully locate the basketball without any false positive regions. The secondary and tertiary images of the false-positive region show no matching regions.

After combining the results from each stage, the matching region identifying the true location of the *basketball* persists while the false-positive region is discarded. As a terminating condition, if a single matching region persists after being processed from three distinct viewpoints, the region is concluded to be a positive match. Figure 5.20 shows the total confidence map of the environment. All views are merged and the viewpoint lines are shown. It is clear that the only persisting matching region indicates the true location of the target object.

This demonstration of the multi-view object recognition algorithm as implemented on a mobile robot platform shows the ability of the system to search an unknown environment for target objects from the object database. The uncontrolled conditions of the scene show the robustness of the algorithm against photometric variations as well as a significantly cluttered background.

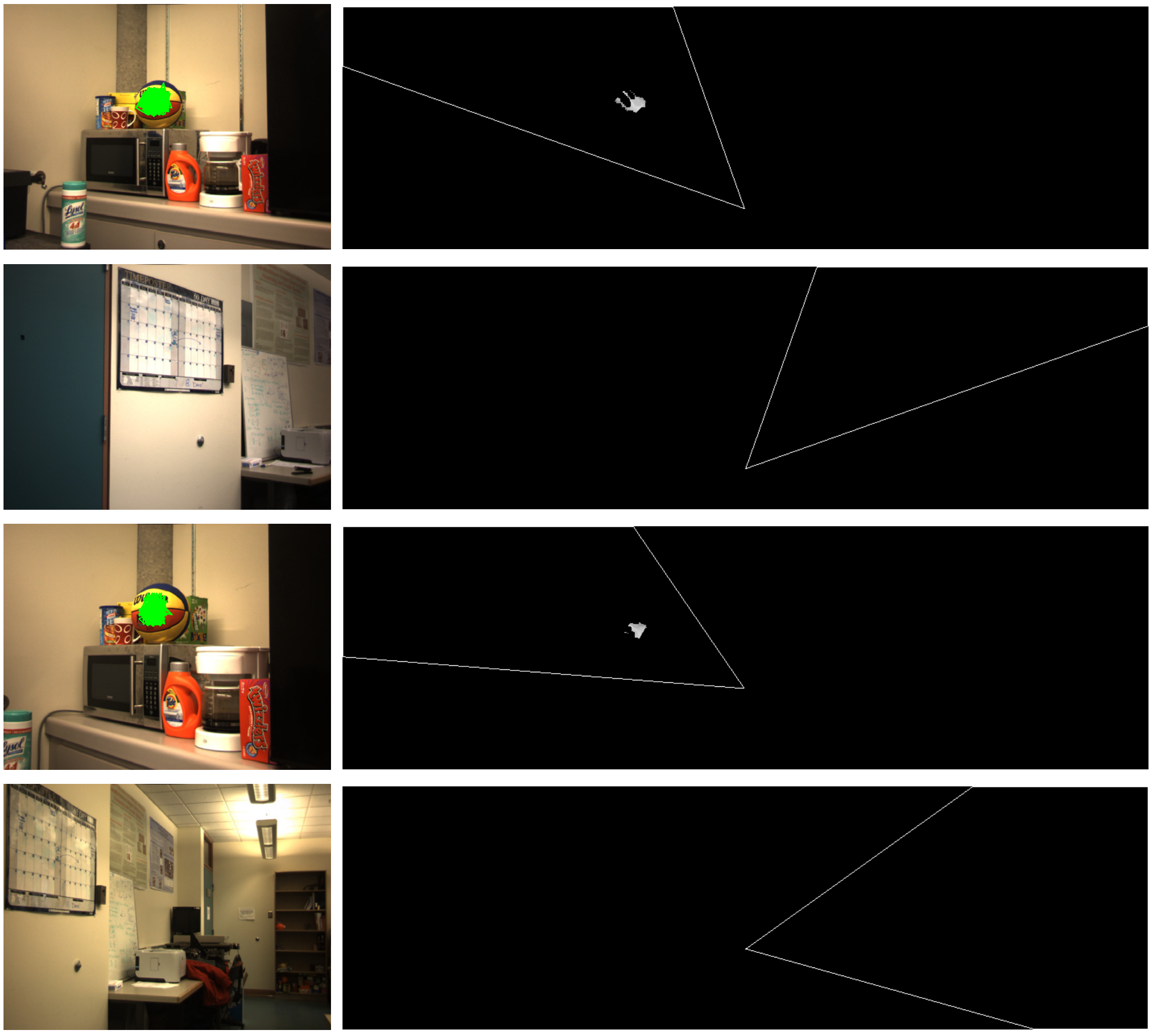

Figure 5.19: Secondary Views. *Left:* images taken at 3 distinct viewpoints in the environment, *Right:* the corresponding birds-eye-view confidence maps for each view.

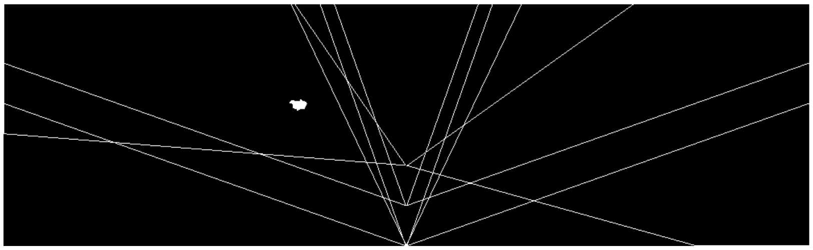

Figure 5.20: Final Confidence Map. After merging the results from all viewpoints, regions which have an average confidence value of above 0.5 are determined to be true matches.

5.4 Run-Time Considerations

The complete object recognition and multi-view inspection algorithm demonstrated in this chapter was implemented in the MATLAB programming environment. As MATLAB uses an interpreted programming language, the execution of code is generally slower than languages which are compiled such as C/C++. For the generation of the SDH histograms, which is required several thousands of times during the processing of a single image, the code was written in C language and compiled to a MATLAB compatible MEX file for use within the MATLAB programming environment. The use of the MEX file greatly reduces the processing time of each image.

The real-world experimental results presented in sections 5.2 and 5.3 are comprised of 40 independent scene image searches. For each experiment, the scene image search and the model data extraction stages were profiled. The minimum, maximum, and average search times were calculated and plotted in Figure 5.21. These results give a general idea of the time required to process a scene image to find matching regions. The platform used to process these images uses a Intel Dual Core processor, each operating at 1.86GHz.

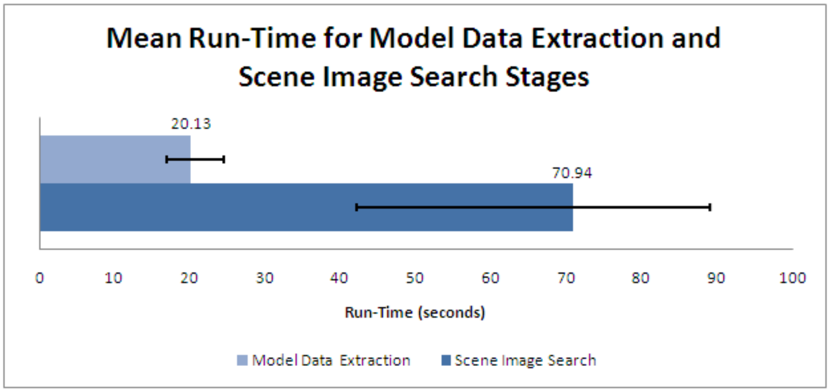

Figure 5.21: Run-Time for Model Data Extraction and Scene Image Search Stages.

It is important to note that the model data extraction stage may be carried out once for each object and then loaded in to memory for use for each subsequent scene image search. While this algorithm does not run in real-time, future considerations would be to accelerate the algorithm to achieve this. With a real-time implementation, views may be continuously captured as a robot navigates an environment and a conclusive result may be achieved more quickly. The current implementation demonstrates the potential for such improvements.

Chapter 6

Conclusions

In this work, a specific research problem was addressed: the development of an object recognition system, based on the matching of illumination invariant local color features, that may be incorporated with a mobile robot platform for the use of identification of specific objects within unknown environments. A complete and concise solution was presented and experimental results were provided.

The solution to the object recognition problem required information from an object to be extracted and searched for in a scene image. Matching regions in the scene could then be identified based on the matches between the scene and database information. The SDH histogram, and the corresponding extraction and clustering techniques, were first presented as a method of extracting information from the database images of objects. The object recognition algorithm used to search a specific scene was next presented. Here, both image and depth information were used to search different regions in the scene effectively. Lastly, a technique was presented which incorporated the object recognition results processed from multiple viewpoints within a 3D environment. This technique was designed for the integration of a mobile robot platform to be used to automatically navigate and search an unknown environment in order to locate a specific object.

Experimental results showed the capability of the algorithm to locate instances of database objects within a scene. Based on these results, four important conclusions can be made:

1. The object recognition algorithm presented in Chapter 4 is capable of recognizing and locating objects in cluttered 3D environments with uncontrolled illumination. Several experiments were carried out in order to test the ability of the system to

locate objects in scene image. In some cases objects were not recognized; however, using the multi-view inspection algorithm this was compensated by matching results from other viewpoints.

2. The Saturation-Weighted Distributive Hue (SDH) histogram local feature descriptor, presented in this work, effectively encapsulates the descriptive color information from an objects appearance. While objects without any color information are not effectively represented with SDH histograms, objects composed of one or more colors are effectively represented using this structure.

3. The incorporation of multiple views into the object recognition algorithm increases the discriminative power of the algorithm by providing further validation to initial match hypotheses. From some viewpoints, objects may be severely occluded making it difficult to obtain matching results. Using multi-view inspection, the target object may still be successfully located by inspecting the environment from multiple viewpoints.

4. The depth information generated from the stereo imaging system is very useful in determining the appropriate scales to search for in a scene image as well as integrating the object recognition results from multiple viewpoints within an environment. A single image contains only 2D information about the environment. By incorporating the depth information obtained from the stereo camera, 3D coordinates may be obtained for hypothesis matching regions and these regions can be further inspected from different positions within the environment.

The following two sections will summarize both the contributions made in this work as well as the possible future research directions stemming from these contributions.

6.1 Summary of Contributions

In this work, three significant contributions were made towards the goal of object recognition.

1. The development of a novel object recognition algorithm capable of locating known 3D objects despite common challenges such as occlusion, scale change, illumination changes, and shading.

2. The development of the Saturation-Weighted Distributive Hue Histogram local feature descriptor, based on previous histogram construction and color constancy techniques, which is robust to such variations as shading, blurring, specularities, and illumination changes.

3. The development of a technique used to integrate object recognition results from multiple viewpoints using depth information from a stereo imaging system.

6.2 Future Research

The work presented here provides an object recognition algorithm that can be used to integrate the results from several images captured at multiple viewpoints within an environment. A natural direction for the research initiated here is towards the development and incorporation of navigational and obstacle avoidance algorithms for the mobile robot platform. By incorporating such systems, the object recognition algorithm presented here could be carried out on an autonomous vehicle. The search for a specific object could then be carried out without human interaction.

While the object recognition system presented in this work exploits the color information inherent in the appearance of many objects, its drawback is recognizing objects void of color. A possible future research direction would be to incorporate this object recognition system with one that uses only grayscale and gradient information. In this direction, both color information and gradient information could be used to recognize known objects in unknown environments.

Bibliography

[1] A. E. Abdel-Hakim and A. A. Farag. Csift: A sift descriptor with color invariant characteristics. In *Proc. CVPR'06*, volume 2, 2006.

[2] G. J. Agin. *Representation and description of curved objects*. PhD thesis, Stanford University, 1972.

[3] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 509–522, 2002.

[5] T. O. Binford. Visual perception by computer. In *Proc. IEEE Conference on Systems and Control*, volume 261, 1971.

[6] A. Bosch, A. Zisserman, and X. X. Muñoz. Scene classification using a hybrid generative/discriminative approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):712, 2008.

[7] P. Chang and J. Krumm. Object recognition with color cooccurrence histograms. In *Proc CVPR'99*, page 2498, 1999.

[8] M. B. Clowes. On seeing things. *Artificial Intelligence*, 2(1):79–116, 1971.

[9] G. D. Finlayson, M. S. Drew, and B. V. Funt. Diagonal transforms suffice for color constancy. In *Proc. ICCV'93*, pages 164–171, 1993.

[10] P. E. Forssén, D. Meger, K. Lai, S. Helmer, J. J. Little, and D. G. Lowe. Informed visual search: Combining attention and object recognition. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)'08*, pages 935–942, 2008.

[11] D. A. Forsyth. A novel algorithm for color constancy. *Computer Vision*, 1990.

[12] S. Frintrop, A. Nuchter, and H. Surmann. Visual attention for object recognition in spatial 3d data. *Attention and Performance in Computational Vision*, pages 168–182, 2005.

[13] B. Funt, K. Barnard, and L. Martin. Is machine colour constancy good enough? In *Proc. ECCV'98*, pages 445–459, 1998.

[14] B. V. Funt and M. S. Drew. Color constancy computation in near-mondrian scenes using a finitedimensional linear model. In *Proc. CVPR'88*, pages 544–549, 1988.

[15] B. V. Funt and G. D. Finlayson. Color constant color indexing. *IEEE Transactions on Pattern analysis and Machine Intelligence*, 17(5):522–529, 1995.

[16] J. M. Geusebroek, R. Van den Boomgaard, A. W. M. Smeulders, and H. Geerts. Color invariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1338–1350, 2001.

[17] T. Gevers and A. W. M. Smeulders. Color based object recognition. *Pattern Recognition*, 32:453–464, 1997.

[18] T. Gevers and H. Stokman. Robust histogram construction from color invariants for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):113–118, 2004.

[19] A. Guzman. Decomposition of a visual scene into three-dimensional bodies. In *Proc. 1968 Fall Joint Computer ConferenceI*, pages 291–304, 1968.

[20] J. Huang, S. R. Kumar, M. Mitra, W. J. Zhu, and R. Zabih. Image indexing using color correlograms. In *Proc. CVPR'97*, page 762, 1997.

[21] D. P. Huttenlocher and S. Ullman. Object recognition using alignment. In *Proc. ICCV'87*, pages 102–111, 1987.

[22] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[23] Y. Ke and R. Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. In *Proc. CVPR'04*, volume 2, 2004.

[24] E.H. Land and J. J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61(1):1–11, 1971.

[25] S. Lazebnik, C. Schmid, and J. Ponce. Sparse texture representation using affine-invariant neighborhoods. In *Proc. CVPR'03*, 2003.

[26] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR'06*, volume 2, 2006.

[27] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV'99*, page 1150, 1999.

[28] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[29] A. K. Macworth. Interpreting pictures of polyhedral scenes. *Artificial Intelligence*, 4(2):121–137, 1973.

[30] L. T. Maloney and B. A. Wandell. Color constancy: a method for recovering surface spectral reflectance. *Readings in computer vision: issues, problems, principles, and paradigms*, page 293, 1987.

[31] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[32] R. Nevatia and T. O. Binford. Description and recognition of curved objects. *Artificial Intelligence Journal*, 8:77–98, 1977.

[33] F. Orabona, G. Metta, and G. Sandini. Object-based visual attention: a model for a behaving robot. In *Proc. CVPR'05*, pages 89–89, 2005.

[34] L. G. Roberts. Machine perception of 3-d solids. *Optical and Electro-optical Information Processing*, 1965.

[35] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or how do i organize my holiday snaps?. In *Proc. ECCV'02*, pages 414–431, 2002.

[36] C. Schmid and R. Mohr. Combining greyvalue invariants with local constraints for object recognition. In *Proc. CVPR'96*, pages 872–877, 1996.

[37] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[38] K. Sjo, D. G. Lopez, C. Paul, P. Jensfelt, and D. Kragic. Object search and localization for an indoor mobile robot. *Computing and Information Technology*, 2008.

[39] G. Stockman. Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing*, 40(3):361–387, 1987.

[40] M. J. Swain and D. H. Ballard. Color indexing. *Computer Vision*, 1991.

[41] D. W. Thompson and J. L. Mundy. Three-dimensional model matching from an unconstrained viewpoint. In *Proc. ICRA'87*, pages 208–220, 1987.

[42] J. Tsotsos and K. Shubina. Attention and visual search: Active robotic vision systems that search. In *Proc. ICVS'07*, 2007.

[43] K. van de Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[44] J. van de Weijer and C. Schmid. Coloring local feature extraction. In *Proc. ECCV'06*, 2006.

[45] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR'01*, volume 1, page 511, 2001.

[46] J. Wu and J. M. Rehg. Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. In *Proc. ICCV'09*, 2009.