ACTIVE OBJECT RECOGNITION FOR 2D AND 3D APPLICATIONS

By

Natasha Govender

Submitted in fulfilment of the requirements for the degree

Doctor of Philosophy (Electrical Engineering)

in the

Department of Electrical Engineering

at the

UNIVERSITY OF CAPE TOWN

Advisor: Dr F. Nicolls

August 2015

ACTIVE OBJECT RECOGNITION FOR 2D AND 3D APPLICATIONS

NATASHA GOVENDER

Abstract

Active object recognition provides a mechanism for selecting informative viewpoints to complete recognition tasks as quickly and accurately as possible. One can manipulate the position of the camera or the object of interest to obtain more useful information. This approach can improve the computational efficiency of the recognition task by only processing viewpoints selected based on the amount of relevant information they contain. Active object recognition methods are based around how to select the next best viewpoint and the integration of the extracted information. Most active recognition methods do not use local interest points which have been shown to work well in other recognition tasks and are tested on images containing a single object with no occlusions or clutter.

In this thesis we investigate using local interest points (SIFT) in probabilistic and non-probabilistic settings for active single and multiple object and viewpoint/pose recognition. Test images used contain objects that are occluded and occur in significant clutter. Visually similar objects are also included in our dataset. Initially we introduce a non-probabilistic 3D active object recognition system which consists of a mechanism for selecting the next best viewpoint and an integration strategy to provide feedback to the system. A novel approach to weighting the uniqueness of features extracted is presented, using a vocabulary tree data structure. This process is then used to determine the next best viewpoint by selecting the one with the highest number of unique features. A Bayesian framework uses the modified statistics from the vocabulary structure to update the system's confidence in the identity of the object. New test images are only captured when the belief hypothesis is below a predefined threshold. This vocabulary tree method is tested against randomly selecting the next viewpoint and a state-of-the-art active object recognition method by Kootstra et al. [1]. Our approach outperforms both methods by correctly recognizing more objects with less computational expense.

This vocabulary tree method is extended for use in a probabilistic setting to improve the object recognition accuracy. We introduce Bayesian approaches for object recognition and object and pose recognition. Three likelihood models are introduced which incorporate various parameters and levels of complexity. The occlusion model, which includes geometric information and variables that cater for the background distribution and occlusion, correctly recognizes all objects on our challenging database. This probabilistic approach is further extended for recognizing multiple objects and poses in a test images. We show through experiments that this model can recognize multiple objects which occur in close proximity to distractor objects. Our viewpoint selection strategy is also extended to the multiple object application and performs well when compared to randomly selecting the next viewpoint, the activation model [1] and mutual information. We also study the impact of using active vision for shape recognition. Fourier descriptors are used as input to our shape recognition system with mutual information as the active vision component. We build multinomial and Gaussian distributions using this information, which correctly recognizes a sequence of objects.

We demonstrate the effectiveness of active vision in object recognition systems. We show that even in different recognition applications using different low level inputs, incorporating active vision improves the overall accuracy and decreases the computational expense of object recognition systems.

TABLE OF CONTENTS

CHAPTER ONE

INTRODUCTION

Object recognition is essential for a large number of computer vision applications which include automated surveillance, video retrieval, and content-based image retrieval, and is important for mobile platforms/robots to interact in human environments. Recognizing objects allows simultaneous localization and mapping (SLAM) applications for robots to build maps of the environment which enables them to localize, avoid obstacles and navigate. Knowing the identity of an object enables a mobile platform or manipulator to interact with the object, for example picking it up and moving it to a different location.

Object recognition is simple for humans as once an object has been learnt, they can still identify the object fairly easily even if the shape, color or size changes or if it has been rotated or partially occluded. This, however, is a challenging problem in computer vision. A number of 2 dimensional (2D) and 3D object recognition systems have been developed using information gathered from sensors such as RGB and infra-red cameras, lasers and new additions like the Microsoft Kinect which provides 3D point clouds of the scene. Various factors affect the strategy used for object recognition, such as the type of sensor, the viewing transformations, the type of object, and the object representation scheme.

Object recognition systems usually consist of a database of objects which are required to be recognized at a later stage in a test environment. Images are captured of these objects and then various types of information are extracted. This information is used to model or describe the object and is then stored in the database. Numerous methods have been developed to model objects. Most of these systems consider the problem of recognizing objects based on information gathered from a single image [2, 3], with varying results. In real-world situations a single viewpoint may be of poor quality or may simply not contain sufficient information to reliably recognize or verify the object's identity

unambiguously. This is especially true if it is occluded, appears in cluttered environments, or if there are a number of objects in the database that have viewpoints in common. The recognition of objects that appear in cluttered environments with significant occlusions is a complicated and challenging problem. If it is not possible to recognize an object from one viewpoint due to ambiguous situations, a more promising viewpoint needs to be selected [4, 5]. In 3D object recognition, gathering more evidence improves recognition [6] as some viewpoints will be more informative than others.

The question that arises in multiple view object recognition is where to capture the next viewpoint. It is not computationally efficient to process images captured from every viewpoint in a test scene. Ideally only images providing useful information about the object should be captured and processed. This idea of looking for informative viewpoints is referred to as active vision and the term was first used by Bajcsy [7]. Human vision is an active process as humans can change their position or focus, among others things, to get a better understanding of a scene. This is the motivation behind active vision. Active vision is based on the premise that an observer (human or computer) may be able to understand an environment more effectively if sensors interact with the environment, selectively looking for relevant information to complete a specific task. This is in contrast to the more conventional, passive approach to computer vision where the camera is supposed to capture an image of the entire scene and complete the vision task based on the single image.

Active vision has applications in vehicle tracking, interactive MRI segmentation and robotic applications such as object recognition, surveillance and scene reconstruction and understanding. Active vision is important as it provides mechanisms to deal with problems such as occlusions and limited field of view and resolution of a camera. In robotic applications, active vision allows a mobile platform to decide where to move to capture the next viewpoint. In the context of object recognition, active vision refers to the ability to manipulate a camera or the object of interest to obtain more useful information to complete the object recognition task as quickly and accurately as possible. Using a digital camera as a sensor for example, the strategies may include the ability to zoom into and out of the object of interest when required. It may also include moving either the camera or the object of interest itself to capture more information. By looking for more relevant information and not processing unnecessary viewpoints, active vision improves the computational efficiency of the task and its overall performance as it is used to gather more evidence for recognition [6]. Active object recognition methods are based around how to select the next best viewpoint and the integration of the extracted information. Methods for next best viewpoint selection often include selecting viewpoints that reduce the entropy or ambiguity of a system, and fusion of data methods include Bayes and Dempster-Shafer. The movement of the sensor can also be incorporated into an active vision systems. Depending on the sensor used, a weighting can be included based on the time taken and energy required to move the sensor to a specific viewpoint. This can be combined with the weighting/potential information of each viewpoint to provide a new combined cost function. The change in the results will depend considerably on the weighting scheme implemented for the movement of the

sensor. Most systems developed for active object recognition do not use local interest points, which in recent years have proven effective for 3D object recognition tasks [8–11], and these are generally tested on scenes containing a single object with no occlusion or clutter.

In this thesis we investigate using local interest points in probabilistic settings for active object and viewpoint/pose recognition. We examine how well these work in cluttered environments for both active single and multiple object recognition. We also study the impact of using active vision for shape recognition. Our aim is to research the benefits, if any, of using active vision algorithms with different inputs, which in our experiments are local interest points and Fourier descriptors, on different object recognition systems. The extraction, modeling and active vision techniques implemented for the different recognition applications are presented in the the form of systems.

System 1

For the first instance, an object recognition system proposed by [12] is implemented. Objects are represented by a set of local interest points. Description in terms of local interest points has the advantage that the representation is robust to occlusions, noise and changes in viewpoint [13, 14]. Local interest points or features are interesting visual characteristics in an image which can include points, edges or even objects. Feature detectors are used to locate these areas in an image. Feature detectors may be combined with feature descriptors, which are vectors that uniquely identify each feature. Examples of such feature detectors and descriptors include the scale invariant feature transform (SIFT) descriptors and the speeded up robust feature (SURF) [9] descriptors. These descriptors are then matched to information extracted from test images using predefined metrics. The ideal situation will be to detect a large number of meaningful features in a typical image and match them reliably across different views of the same scene or object. Critical issues in detection, description and matching are robustness with respect to viewpoint and lighting changes, the number of features detected in a typical image, the frequency of mismatches, and the computational cost of each step.

We use SIFT in our experiments [13]. SIFT has been used successfully in many computer vision tasks including object recognition [3, 10], localization and navigation [15] and gesture recognition [16]. The SIFT descriptor is invariant to translations, rotations and scaling transformations and is robust to changes in illumination and affine transformation. The object recognition system creates a pseudo-3D model for each object by clustering similar viewpoints together. Feature matching is accomplished by using SIFT matching, which calculates the Euclidean distance between descriptors. These matches are then refined using the Hough transform [17], which imposes geometric constraints on the objects and assists in removing spurious or ambiguous matches. This system demonstrates that SIFT performs well in recognizing objects that may be occluded in a cluttered environment. An issue arises if the object is significantly occluded, if there are objects that are visually similar or if the image is blurry. In these cases there is not sufficient evidence to uniquely identify the object from a single image and another viewpoint of the test scene is required. The question arises as to how to select the next best viewpoint.

System 2

In System 2, we introduce a novel 3D active object recognition system which consists of a mechanism for selecting the next best viewpoint and an integration strategy to provide feedback to the system. It is made up of the following components:

- object representation,
- the next best viewpoint selection strategy and stopping conditions, and
- the integration task, which accumulates evidence gathered from a sequence of viewpoints.

This system is designed using a selector-observer framework, where the selector is responsible for finding the next best viewpoint based on the expected information of each viewpoint and a Bayesian 'observer' component updates the belief hypothesis and provides feedback. The selector and observer components are independent and thus the algorithms in each could be altered without affecting the other. We use SIFT features and descriptors to model the objects. We create our own database of 20 everyday objects for use in these experiments. Objects that are visually similar are also included in the database. The SIFT features and descriptors extracted from the training images are used as input into a vocabulary tree data structure which is used to determine the next best viewpoint. The vocabulary tree data structure is used to calculate a weighting for each feature based on its perceived uniqueness, allowing the system to select the viewpoint with the greatest number of 'unique' features. Ideally a viewpoint is selected that will provide useful information to uniquely identify an object. The independent Bayesian observer framework uses the modified statistics from the vocabulary tree structure to update the system's confidence in the identity of the object. Bayesian approaches to active object recognition have proved effective in a number of cases, allowing information across views to be integrated, and permitting a principled approach to data acquisition. While most multiple view object recognition systems have no tangible method of determining the accuracy of the recognition method, our system provides a certainty/belief as to the current object's identity and pose.

The process is sped up as new images are only captured at the 'next best viewpoint' and processed when the belief hypothesis of an object is below a predefined threshold. Experiments are carried out for object verification and recognition for the 20 objects. Objects from the training database appear in a cluttered environment with significant occlusion in the test images. Both randomly selecting the next best viewpoint and System 2 correctly verifies all objects in the database but our approach requires fewer viewpoints. Our active object recognition approach is tested against randomly selecting the next best viewpoint and a state-of-the-art active object recognition method by Kootstra et al. [1]. Both systems use SIFT features for object recognition, but use contrasting model, update and viewpoint selection strategies. Our vocabulary tree system outperforms randomly selecting a viewpoint and the method by Kootstra et al. as it correctly recognizes more objects in the database with less computational effort.

System 3

Given our database, there are two objects which are not correctly recognized by any of the methods. To improve the recognition accuracy we introduce System 3, which makes use of the extracted features in probabilistic models for object and pose/viewpoint recognition. Existing approaches rely on probabilistic models which make simplifying assumptions such as that features may be treated independently and that objects will appear without clutter at test time. Two Bayesian probabilistic methods for object recognition and object and pose recognition are introduced. Three likelihood models are developed for use in these Bayesian approaches, each with increasing levels of complexity. These models are explicitly designed to cope with situations in which such assumptions fail, and show them to perform well in a Bayesian active recognition setting using test data in which objects appear in cluttered environments with significant occlusion. Through these experiments we show that incorporating geometric information as well as information about the background and possible occlusions are important, with this model correctly recognizing all objects in our database.

System 4

Having presented the probabilistic object and viewpoint models that are able to recognize single objects present in cluttered test images, we extend this probabilistic framework to recognize multiple objects and their poses in a scene (System 4). The test images may contain any number of objects from the database which are required to be recognized as well as distractor objects which do not appear in the database. The system is designed to recognize multiple objects and their poses, if any are present, in the test images. We illustrate a single probabilistic model using the Bayesian framework which is extended to the multiple object recognition scenario. The next viewpoint selection is modified for multiple object recognition. This selection mechanism is compared with previous methods in both Bayesian and non-Bayesian contexts and performs well in terms of efficiency and accuracy in the multiple object setting compared to mutual information, random and the activation model presented by Kootstra et al. [1]. Mutual information measures the mutual dependence of two random variables. System 4 is an active vision object recognition system designed to handle the realistic situation of simultaneously recognizing multiple objects in close proximity, which may be subject to extensive occlusions and clutter from distracter objects. A Bayesian model for data fusion, which maintains a distribution over multiple object and viewpoint hypotheses, is developed and shown to work well in the multiple object recognition scenario.

System 5

In all the systems discussed thus far we used SIFT features and descriptors for active object recognition, but this may not always be the most appropriate manner to describe an object. Situations may arise where objects have no visual texture or color thus making local features inappropriate for recognition. We look at an alternative method for modeling objects which contain no visual texture using Fourier descriptors. Fourier descriptors are widely used for shape description and recognition.

System 5 initially uses Fourier descriptors for recognizing shapes in a non-active setting. The Euclidean distance is calculated between the descriptors to determine a match. This method achieves a recognition accuracy of 80%. System 5 is then extended to an active shape recognition system using mutual information. The aim of this system is to determine the correct sequence of the objects/shapes. We use mutual information as the active vision component to look for additional information about the object/shape in the sequence that it is most uncertain about. We show that actively looking for information in this type of recognition task improves the overall accuracy of the shape recognition system.

Most active object recognition systems that exist in the literature are tested in environments which contain a single object with no occlusion or background clutter, although exceptions include [1]. Our systems are tested on a challenging dataset with objects to be recognized occurring with occlusion in a cluttered environment. Systems 2 and 3 provide excellent recognition accuracies given the difficult setting. These systems also provide a certainty/belief about the identity of the object and new viewpoints are only processed when the belief is below a threshold (set to 80% in our experiments), which not all active vision systems provide [1, 18]. Using the vocabulary data structure to weight unique features is new and proves to provide excellent results. The active vision systems that exist are focused on recognizing a single object in a test image. System 4 is novel as it recognizes multiple objects and their poses in cluttered test scenes with good results. There are a number of unique components in System 5 which include using active vision, in particular mutual information, in a shape recognition system with Fourier descriptors as input. Systems 2, 3, 4 and 5 produce excellent results on the challenging datasets in an active setting. These can easily be used on mobile platforms in future for reliable and efficient object recognition.

Chapter 2 introduces the local interest point detector and descriptor used, the Hough transform and its application to object recognition applications. Our novel 3D active object recognition system, along with the next viewpoint selector and observer components are also detailed in this chapter. Experiments and results are presented for object verification and recognition. Our system is compared to randomly selecting the next viewpoint and the method by Kootstra et al. [1]. The SIFT features extracted are then used as input to probabilistic models for object and object and viewpoint recognition, which are discussed in chapter 3. Three different likelihood models are also presented with various levels of complexity and these are shown to perform well given our challenging dataset. Chapter 4 describes our probabilistic approach to recognizing multiple objects and poses which is tested on our database. Our viewpoint selection is extended for the multiple object scenario and is compared to randomly selecting the new viewpoint and mutual information and is also discussed in chapter 4. Having implemented several systems which use local interest points and the vocabulary tree for viewpoint selection for active object recognition, chapter 5 describes a different approach for recognizing objects in an active setting. This chapter describes the process of using Fourier descriptors as input to a shape recognition system with mutual information as the active vision component. Chapter 6

summarizes the results and importance of active vision in various object recognition applications.

Novel contributions

The original contributions made in this thesis are:

- A next best viewpoint selector using a vocabulary tree data structure and local interest points.
- A Bayesian framework which uses statistics generated from the vocabulary tree to update the system's belief in the identity of an object.
- A framework which incorporates the next best viewpoint selector and Bayesian observer components where new images are only captured when necessary, which reduces the computational expense of the system. This system has also been shown to outperform randomly selecting the next best viewpoint and a state-of-the-art system presented by Kootstra et. al. [1].
- Incorporating local interest points and geometric constraints using the Hough transform for probabilistic models for single object and pose recognition.
- A probabilistic framework for recognizing multiple objects and their poses in a test image. The next best viewpoint selector and update components are extended for the multiple object scenario.
- An active shape recognition system using Fourier descriptors and mutual information.
- Training images for 20 objects (including visually similar objects).
- Testing images for single and multiple objects occurring in cluttered environments with occlusion.

CHAPTER TWO

ACTIVE 3D OBJECT IDENTIFICATION USING VOCABULARY TREES[1]

2.1 INTRODUCTION

Object recognition is an essential component in a number of computer vision tasks. In the context of mobile platforms and manipulators, sensors (cameras or lasers) on the platform are used to capture information about the scene to enable the platform to interact with the environment. This interaction may include tasks such as navigation, obstacle avoidance or grasping an object. The following steps are involved in most object recognition systems:

- Creating a database of objects which are required to be recognized at a later stage. This process is accomplished by using various techniques to model the objects which are then stored in a database.
- Images are captured of the test scene or environment and the goal is to recognize if any of the objects in the database are present in the test image. Object recognition is achieved by matching the extracted data from the test image to those in the database using some predefined metric.

[1] Related publications:

- Natasha Govender, Jonathan Claassens, "Grasping Objects from a Users Hand using Time-of-Flight Data", Pattern Recognition Association of South Africa (PRASA), November 2010.
- Natasha Govender, Jonathan Claassens, Phillip Torr and Jonathan Warrell, "Active Object Recognition using Vocabulary Trees", IEEE Workshop on Robot Vision, 16-18 January 2013, Florida, United States of America.

Techniques for modeling objects can include using the appearance or geometry of the objects or extracting local interest points or features. Features refer to structures of interest such as points, edges or objects in an image. Features have been shown to work well in single and multiple view object recognition provided the objects contain sufficient discriminatory information such as visual texture.

Two object recognition systems are introduced in this chapter. The first is an implementation of the system presented in [12]. This is a 3D object recognition system which extracts local features using the SIFT detector and descriptor. SIFT has successfully been used in object recognition applications [3,10]. The SIFT descriptor is invariant to image translations, rotations and scaling and partially invariant to changes in illumination and local image deformations. Initially for object recognition tasks, recognition was accomplished by matching to individual training images. This process is computationally inefficient and does not allow for ways to integrate information across images, which decreases the robustness of the system. A number of state-of-the-art object recognition systems now combine multiple training images to produce a single model representation of an object [19–23]. The object recognition system introduced by [12] also incorporates a view clustering algorithm which allows multiple training images under a range of imaging conditions to be combined to produce a pseudo-3D model representation of each object in the database. This model has the advantage of allowing the system to robustly recognize a 3D object from any viewpoint. Training images are clustered together based on their similarity, which is calculated using SIFT matching. An additional filtering step is implemented using the Hough transform which enforces geometric constraints and helps to remove spurious and ambiguous matches. The Hough transform uses the translation, scale and orientation transformation parameters to vote on the transformation of an object and only matches that agree on a specific transformation are kept. Each SIFT feature contains its location, scale and orientation relative to the object model. This system is implemented as background work to illustrate that SIFT works in a realistic environment and to introduce certain algorithms, such as SIFT and the Hough transform, which are used later in the more complex active object recognition systems.

SIFT matching and the Hough transform are also used to match the information extracted from the test images to those present in the database for the recognition process. A drawback of the view clustering method is that the model does not provide an estimate of the pose (position and orientation) of the object and no mechanism is available for selecting a new viewpoint if a single view does provide enough information to accurately recognize an object. In many instances single images of objects appearing in cluttered scenes are insufficient to accurately identify objects and thus multiple images are required [24].

The second object recognition system introduced in this chapter is a novel framework for active 3D multiple view object verification and recognition. Object verification refers to deciding if a known object is present in a scene while object recognition aims to identify which objects, if any, are present in a scene. This system makes use of the SIFT and Hough algorithms used in System 1. We aim to use

active vision in this context to actively search an object to obtain more informative views to increase the accuracy and decrease the computational expense [6] of the object identification process. The computational expense can be decreased by reducing the number of images processed to accurately identify an object. Active object recognition systems seek to dynamically and intelligently select more informative viewpoints to reduce the number of images processed to accurately recognize an object. These active object recognition systems also provides a framework for collecting evidence until we obtain a sufficient level of confidence in one object hypothesis. The system has to provide tentative object hypotheses for each single view. Combining observations over a sequence of active steps moves the burden of object recognition slightly away from the process used to recognize using a single view to the processes responsible for integrating the classification results of multiple views and for planning the next action.

As mentioned previously, the two main focus areas of 3D active object verification and recognition are selecting the next best viewpoint to be processed, and integrating of the extracted information in a meaningful manner. For selecting the next best viewpoint many systems simply use active vision to select the sequence in which a set of pre-captured images should be processed for recognition, but optimization of appraisal time by reducing the number of images is not considered [18]. We introduce a unique framework for feature-based active object verification and recognition. Our system uses SIFT [13] detector and descriptor to extract relevant object features. The structure of the system is, however, not SIFT dependent and thus any detector and descriptor can be used for feature extraction.

The framework comprises an automatic viewpoint selector, to select the most informative viewpoints given the current task, and an independent observer component to integrate the extracted information. The automatic viewpoint selector uses a vocabulary tree data structure [25] to weight the uniqueness of every feature extracted from a viewpoint. Every viewpoint for all objects in the database is then given a value which is obtained by summing the uniqueness measure of all its features. The higher the value, the more unique the viewpoint. This quantity is then used to select the next best viewpoint. The vocabulary tree provides a method to discretize the feature space to reduce feature dimensionality when considered in the observer component. The observer component updates the system's belief in the identity of an object with current view information in a recursive Bayesian manner using a prior determined from previous views. These two components are designed to be independent of each other. The advantage of this framework is that the algorithm for the next viewpoint selection can be altered or completely rewritten and it would not affect the integration component and vice-versa. We show that this method performs better than randomly selecting the next viewpoint and another state-of-the-art active object recognition system proposed by [1].

In the next section, we discuss the related work on object recognition, active object recognition systems and its various components. The object recognition system introduced by Lowe [12, 13] is presented in section 2.3 which also details the SIFT and Hough transform concepts. Our novel active 3D object recognition framework, which comprises a next viewpoint selector and observer component

is discussed in section 2.4. Experiments are conducted for object verification and object recognition and this process as well as the results are also described in section 2.4. We compare our active object recognition system to the method presented by Kootstra et al. [1] which is detailed section 2.5 along with the results. Our conclusions are presented in section 2.6.

2.2 RELATED WORK

Object recognition has many computer vision applications which include quality control and assembly in industrial plants [26], robot localization and navigation [27], monitoring and surveillance [28] and content based retrieval [29]. An object recognition system consists of an algorithm to model an object and a metric to match to an object captured from a test environment. Recognizing objects under varying conditions, lighting, distance, shape, size or occlusion is a simple task for humans as it is easy for us to generalize. This is a much more difficult and challenging problem for computer vision application as images are captured in 2D from a 3D environment, which leads to an inherent loss of information. Views of the same object captured from different viewing angles can give rise to different images. Techniques for object representation thus need to be invariant to viewpoint changes and object transformation and robust to noise and occlusions.

Many algorithms have been developed over recent years for object representation, primarily using model-based (based on the object's shape or appearance), context-based (based on the context in which objects may be found) or function-based approaches (based on the function for which objects may serve), with varying results. The most popular of these are model-based approaches. Model based representation schemes vary considerably from edge detection [30, 31], shape detection [32], aspect graphs [19, 20, 24, 33], histogram of gradients [18], and neural networks [34] to feature extraction [9, 13]. Features or interest points may refer to corners, linear edges or objects themselves present in the image. Feature extraction methods have proven to produce accurate results for object recognition applications [10, 18, 35, 36]. Local interest points or features have the advantage that the representation is robust to occlusions, noise and changes in viewpoint [13, 14].

An evaluation was conducted between the feature detector methods of SIFT, PCA-SIFT and SURF in [37]. These methods were evaluated on repeatability, processing time, scale changes, image rotation and blur and illumination changes. It was found that SIFT produced the most accurate results although it was slower than the other two methods. Various descriptors were evaluated by [38] to determine the distinctiveness and robustness to changes in viewing conditions as well as to errors of the detector. They concluded that the gradient location and orientation histogram (GLOH) performed best with the SIFT descriptor very close in performance. Moreels and Perona [39] compared the most popular feature detectors and descriptors to determine their performance in recognizing 3D objects. These detectors and descriptors were tested for robustness to change in viewpoint, lighting and scale. In their experiments they found that the best overall choice was an affine-rectified detector [40] followed

by SIFT [13]. In [41], comparisons were conducted between SIFT and its variants which include PCA-SIFT, GSIFT, CSIFT, SURF and ASIFT. Performance was evaluated on scale change, rotation change, blur change, illumination change, and affine change. SIFT performed the best under scale and rotation. SURF performs the worst in the different situations, but runs the fastest which is marginally faster than SIFT (1.8 milliseconds as opposed to 2.1 milliseconds). ASIFT is the slowest at 6.7 milliseconds.

Although SURF is computationally less expensive since it makes use of a 64 bit descriptor as opposed to SIFTs 128 bit descriptor, as can be noted from the above studies SIFT outperforms SURF in accuracy and robustness. There is a trade-off between the robustness of the feature detector and descriptor and the speed. Although speed is a consideration, the main focus of these recognition systems is the robustness and accuracy. ASIFT also uses a 128 descriptor and thus there would be no improvement in computational cost. Based on the results of these evaluations, we chose to use the SIFT detector and descriptor in our experiments. SIFT has successfully been used in many computer vision tasks including object recognition [3, 10], localization and navigation [15] and gesture recognition [16].

Objects may look different under varying conditions such as light, color, size, shape or viewing direction and therefore a single image may not contain sufficient information to recognize an object unambiguously. This is especially true if the test environment is cluttered, if the object in question is occluded or if two or more objects have a view in common with respect to a feature set. Such objects may be distinguished only through a sequence of views. The question then arises as to how to select the next best viewpoint as it is not computationally efficient to process all viewpoints.

Active vision provides a mechanism for actively looking for relevant information in an environment. The first general frameworks for an active vision system were introduced by [7] for optimal sensor placement and [42] for improving the perceptual quality of tracking results approximately two decades ago. Significant progress has been made in various areas including active vision techniques for industrial inspection, object recognition, security and surveillance, site modeling and exploration, multi-sensor coordination, mapping, navigation and tracking [43]. In the last 7–10 years this field has become very active because of the wide range of applications in the field of robotics. Active vision provides mobile platforms and manipulators the ability to look for and process relevant information making these systems more robust and efficient. In surveillance applications, active vision can be used to select the next best viewpoint to control the pan-tilt-zoom parameters of the cameras to keep the object or person of interest in view [44, 45]. Selecting the 'next best viewpoint' can increase the performance and accuracy of applications [6]. This idea of sensor planning can be widely found in most autonomous robotic systems [46].

We investigate specifically the field of active object recognition which can be used to determine which viewpoints are more informative to complete the task as quickly and reliably as possible. A number of different approaches have been proposed for active object recognition [1, 18, 47–50]. Apart from the object representation schemes used, the major differentiating factors are the next best viewpoint

selection algorithm and the data fusion methods. Object representation schemes include parametric eigenspace data [47], entropy maps [50] and SIFT features [1, 51].

The next best viewpoint refers to the viewpoint that will provide the most amount of information to complete the task as quickly and as accurately as possible. This prevents the system from processing unnecessary viewpoints and decreases the computational expense of the system. Most active object recognition systems are based on selecting viewpoints that will minimize ambiguity using Shannon entropy [47, 52] or Dempster-Shafer theory [19], minimize a weighted error [18] or maximize a defined activation function [1]. In [1] a robot is used to actively explore objects. The system learns objects from different viewpoints and is then used to actively select viewpoints for optimal recognition. Their system is SIFT based and describes an activation function calculated using SIFT descriptors. This calculation is performed between each new test image and all the training images in the database to select the next best viewpoint. Our system is compared to this method as both are based on extracting local interest points using SIFT.

For next best viewpoint selection, we use an efficient bag-of-words approach to organize the training feature database using a vocabulary tree data structure [25]. Vocabulary trees have been used in many computer vision applications such as object detection [53, 54], object recognition [55], scene recognition [56], image retrieval [57], human action recognition [58] and simultaneous localization and mapping (SLAM) approaches for matching similar images and for loop closure [59]. We use this data structure to weight the uniqueness of each feature extracted, which in turn is used to calculate a weighting for each viewpoint. This approach builds on techniques of indexing descriptors extracted from local regions which was traditionally used in text retrieval and was first introduced for object matching in videos [60]. In text retrieval, a document is deconstructed into its component words and these are assigned unique identifiers. Each document is then defined by a vector containing the number of occurrences of words appearing in the document. The set of vectors describing all the documents in the corpus are organized as an inverted file [61]. An inverted file contains an entry for each word in the corpus and then a list of all the documents in which it occurs. A text is retrieved by computing its vector of word frequencies and returning the documents with the closest vectors, which is measured by angles.

In [60] two types of viewpoint covariant regions are computed for each frame in the video, namely shape adapted (SA) and maximally stable regions. Each region is then represented by a SIFT descriptor. The descriptors, which now represent the 'visual words', are vector quantized into clusters using k-means clustering. Weightings are applied to the descriptors so that descriptors appearing less often are weighted higher. This standard weighting is known as the term frequency-inverse document frequency (tf-idf). We use the idea of the tf-idf calculation to weight the uniqueness of the features extracted but use the implementation described in [25].

In [25], a vocabulary tree is used to hierarchically quantize the local region descriptors extracted whereas in [60] a flat structure is used. This hierarchical structure allows a larger vocabulary to be

used and has the potential for dynamic insertions of new objects into the existing database. The vocabulary tree also directly defines the quantization of the descriptors as it is built using k-means clustering. We also modify the vocabulary tree data structure to include a discrete density function at each leaf node, which is used to calculate the system's belief in an object's identity i.e. the probability that it is a specific object.

With the exception of [62], many of the active object recognition systems use a predetermined number of images and merely use active vision to select the sequence in which they should be used [18]. Our system is different: it only captures a new image when required and thus optimizes the number of views needed for reliable recognition or verification. Our system also provides a confidence or certainty measure for an object's identity. The vocabulary tree is also used to generate statistics to update the object belief. Following [34] this system relies on a Bayesian framework for updating a belief function. Other methods used for fusion include discriminative approaches [1, 18], Dempster-Shafer theory [19, 63] and particle filters [48].

We use SIFT matching for object recognition. False matches can arise due to noise and ambiguity within objects. We use the Hough transform as an additional filtering step after SIFT matching to reduce the number of false and spurious matches. The classical Hough transform was first used to identify lines in the image [17], but was later extended to identify shapes such as circles or ellipses [64–66]. It identifies shapes using a voting procedure in parameter space. The Hough transform is very versatile and has been used in a number of computer vision applications which include 3D object classification [67], action recognition [68], tracking of non-rigid objects [69] and analysis of textual images [70]. It is also a very robust detector with a low sensitivity to noise and outliers. The Hough transform works best for a low number of parameters (we use 4). Random sample consensus (RANSAC) can also be used for this functionality. However, it can only handle a moderate percentage of outliers without becoming computationally very expense and only provides a probabilistic census on the model parameters. Given these reasons and our experimental set-up, the Hough transform is a better choice in terms of accuracy and cost. Our implementation of the Hough transform is similar to the one presented in [12].

When classifying objects, all of the above systems, except for [1], consider synthetic images or scenes with a single object [18, 47, 48]. In [1] the target object is placed in the centre of the image with no occlusions, although there is some degree of clutter in the background. Our system recognizes and verifies objects which not only occur in cluttered environments but are also occluded. Few systems in the literature consider datasets with objects that share many visual similarities. Exceptions include [34, 62]. The database used for our experiments contains a number of visually similar objects which can only be differentiated by appraising specific viewpoints. The sizes of the databases used are also smaller than ours. We create a database of 20 objects whereas [1] uses 7 objects, [18] has 9 objects and both [35, 47] have a database of 15 objects.

The majority of active vision systems compare their results to randomly selecting the next best view-

point [1,18,34,47,52,62,63]. Our system is tested against randomly selecting the next viewpoint and against the system proposed by [1].

2.3 BACKGROUND WORK: SYSTEM 1

Here we describe the system introduced by Lowe [12]. We use our own database and show results for recognizing an object from a single view.

2.3.1 DATASET

To create the dataset for use in our experiments, approximately 30 images were captured for six objects using a Prosilica GE1900C camera and a white background. Images captured for the six objects are displayed in Figures 2.1 and 2.2. Objects selected for use in the dataset were of varying sizes and shapes and contained variations in texture.

Figure 2.1: Images of objects in our dataset.

2.3.2 VIEW CLUSTERING

The captured images from the dataset are used to create a model for each object, which enables the system to recognize the object at some later stage. The SIFT detector and descriptor was used to extract relevant features from all the training images captured. A detailed description of the SIFT detector and descriptor and SIFT matching is given in section 2.3.3.

We use the view clustering algorithm presented in [12] to combine multiple training views to create pseudo-3D models of each object. The idea is that similar image views of an object are clustered into a single model view. The first training image is used to build an initial model. This consists of all the SIFT features extracted from the training view. Subsequent images are matched using SIFT matching, which calculates the Euclidean distance between the descriptors. Two features are considered a match if the ratio of the closest to the second closest match is < 0.8.

Figure 2.2: Images of objects in our dataset.

Each SIFT feature contains a record of its location, orientation, and scale within the model view. These parameters are used by the Hough transform [17, 65] to vote for an approximate transformation. Only those features with a consistent interpretation of the object transformation are kept. When clusters of features are found to vote for the same object transformation the probability of the interpretation being correct is much higher than for a single feature. The Hough transform enforces geometric constraints and assists in removing spurious and ambiguous matches.

Each SIFT match votes for an approximate translation, scaling and rotating of the object. For the scale and rotation values, the difference of the matched pair of descriptors are calculated. These values then vote for the appropriate bin size. To generate the translation votes, we solve for the similarity transform that will map the locations, the (x, y) coordinates of the matched features, using the known scaling and rotation above with an unknown translation.

The similarity transform is solved as follows. The similarity transform gives the mapping of a model point $[xy]$ to an image point $[uv]$ in terms of image scaling s, image rotation θ, and image translation $[t_x, t_y]$:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta \\ s\sin\theta & s\cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \tag{2.1}$$

Defining $m = s\cos\theta$ and $n = s\sin\theta$ gives

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m & -n \\ n & m \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \tag{2.2}$$

Equation 2.2 can then be written in a linear form by collecting the unknown similarity transform

parameters into a vector [12]:

$$\begin{bmatrix} x & -y & 1 & 0 \\ y & x & 0 & 1 \\ \cdots & & & \end{bmatrix} \begin{bmatrix} m \\ n \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix}. \tag{2.3}$$

Equation 2.3 describes a single feature match, but any number of further matches can be added, with each contributing two more rows to the first and last matrix.

We can write this linear system as $Ax = b$. The least-squares solution for the parameters x can be determined by solving the corresponding normal equations, $x = [A^T A]^{-1} A^T b$, which minimizes the sum of squared distances from the projected model locations to the corresponding image locations. We can then use this solution to calculate the error e between the projected model feature and the image feature:

$$e = \sqrt{\frac{2 \parallel Ax - b \parallel^2}{r - 4}}, \tag{2.4}$$

where r is the number of rows in matrix A. The factor 2 in the numerator allows the squared errors in two rows to be summed to measure a squared image distance. Features are then clustered based on this value.

When a new training image is used as input into the system, it is matched to previous model views and depending on the value of *e* one of three cases can occur:

1. A new cluster model is created if the new training images matches an existing model and $e > \mu$, where μ is a significant threshold.
2. The training image is clustered with an existing model if it matches an existing model and $e \leq \mu$. All features from the training image are transformed into the coordinates of the model view using the similarity transform solution and are linked to any matching features.
3. A new cluster model is created if the training does not match any of the existing clusters.

2.3.3 SCALE INVARIANT FEATURE TRANSFORM (SIFT)

SIFT is a feature detector and descriptor introduced by Lowe [13] for matching different viewpoints of an object or scene. It is now used for many different computer vision tasks such as motion tracking, reconstructing 3D structure from multiple images, and stereo correspondence. The SIFT features that are extracted are generally fairly distinctive and allow features to be correctly matched using their descriptors in a large dataset which is essential for object recognition systems.

The four main stages of SIFT are:

- Scale-space extrema detection

The Difference of Gaussians (DoG) function is applied in scale space to a series of smoothed and resampled images. For example, one pixel in an image is compared with its 8 neighbors as well as 9 pixels in the next scale and 9 pixels in the previous scale. If pixel is a local extremum, it is a potential keypoint. Keypoints are defined as the maxima and minims of the result, which indicates that the keypoint is best represented in that scale.

- Keypoint localization

  Once potential keypoints locations are found, they are refined to select the strongest keypoints. A Taylor series expansion of scale space is used to get more accurate location of extrema, and if the intensity at this extremum is less than a threshold value (0.03 as per [13]) it is rejected. Low contrast candidate points and edges are also discarded using the Hessian matrix.

- Orientation assignment

  An orientation based on local image properties is assigned to each keypoint to achieve invariance to image rotation. A neighborhood is taken around the keypoint location depending on the scale, and the gradient magnitude and direction is calculated in that region. An orientation histogram with 36 bins covering 360° is created. The highest peak in the histogram is taken and any peak above 80% of the maximum is also considered to calculate the orientation. The keypoints are created with the same location and scale, but different directions. These steps ensure that the keypoints are stable for matching and recognition.

- Keypoint descriptor

  A 128-element vector descriptor is determined for each keypoint selected. The descriptor is created using the gradient magnitude and orientation from the region around the keypoint location.

SIFT features detected for an object are displayed in figure 2.3.

Features extracted from the test image are matched to features from the training images by calculating the Euclidean distance between the descriptors. In [13] a modification of the kd-tree algorithm called the best-bin-first search method is used that can identify the nearest neighbors with high probability using only a limited amount of computation. To further improve the efficiency of the best-bin-first algorithm the search was cut off after checking the first 200 nearest neighbor candidates. For a dataset of 100 000 keypoints this provides a speedup over exact nearest neighbor search by about 2 orders of magnitude, yet results in less than a 5% loss in the number of correct matches. To eliminate spurious or incorrect matches, the distance of the closest neighbor is compared to the second closest neighbor. If this ratio is greater than 0.8 then it is considered a false match which eliminates 90% of false matches while discarding less than 5% of the correct matches.

Figure 2.3: SIFT features detected.

2.3.4 HOUGH TRANSFORM

The distance ratio test used for SIFT matching allows false matches arising from background clutter to be removed while possibly keeping those that arise from other valid objects. The Hough transform allows the extracted SIFT features to vote for an approximate location, scale and orientation of an object as described in [12]. This identifies clusters of features with a consistent interpretation of the object transformation, which helps to remove spurious matches.

The classical Hough transform was developed to identify lines in an image [17], but was later extended to identify shapes such as circles or ellipses [64, 65]. It identifies shapes using a voting procedure in parameter space. To illustrate the process, we describe the case of identifying a straight line in an image.

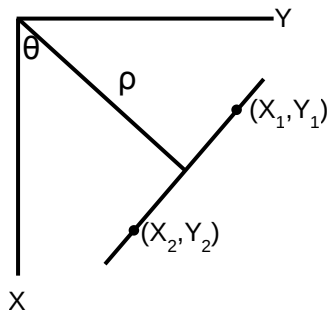


Figure 2.4: Straight line in an image.

Every point on a straight line satisfies;

$$\rho = x\cos\theta + y\sin\theta, \tag{2.5}$$

where ρ is the distance from the center of the coordinate system to the line. Each point, (x, y), that

has been detected by an edge detector algorithm is substituted into the equation for varying values of θ. Every point has an infinite number of lines going through it but for ease of computation a discrete number of θ values is used. For each θ and resulting ρ value a vote is cast. The position with the highest number of votes indicates the strongest line present in the image. The Hough transform is conceptually simple and can be adapted to many geometric shapes, not just lines. However, it can also be computationally complex for objects with many parameters.

Each SIFT feature extracted from a test image contains its 2D location, scale and orientation relative to its model coordinate system. Each match is then allowed to vote for an approximate translation, scaling and rotation of the object. For the scale and rotation values, the difference of the matched pair of descriptors is calculated. These values then vote for the appropriate bin size. We discretize the transformations into 32 bins each for the x and y translations, 5 bins for scale and 12 bins for rotation.

Using the (x, y) coordinates of the matched features and the known scaling and rotation values, we can calculate the translation by solving the similarity transform as shown in equation 2.1. A similarity transformation is a special case of an affine transformation where the shear is zero. The similarity transform implied by these 4 parameters is only an approximation to the full 6 degree-of-freedom pose space for a 3D object and also does not account for any non-rigid deformations.

The bin containing the highest number of votes for each parameter is then determined. Only those features which voted for all the bins with the highest values are kept, as these clusters of features have a consistent interpretation of the object pose. SIFT matching and the Hough transform also assists in removing outliers. Only features matched via SIFT matching is passed to the Hough transform. SIFT matching removes initial outliers. Since the Hough transform enforces geometric constraints between two views, only features agreeing on the translation, orientation and scale are kept which should removing any remaining outliers.

Figure 2.5 displays the initially SIFT matches for the elephant object with Figure 2.6 showing the SIFT matches after the filtering step using the Hough transform. It can be seen that after the Hough transform has been applied, a number of spurious and ambiguous matches have been eliminated.

2.3.5 RESULTS

Our aim was to show that SIFT works well for the object recognition task in a realistic environment, where objects in the test image are partially occluded and appear in clutter with varying illumination. In our system we used the view clustering approach to model the objects in the dataset and SIFT matching and the Hough transform for recognition. Given our test images the system was able to correctly recognize all objects. In [12], when recognizing an object a probability model was used for verification. It was not necessary for us to implement this probability model as the system performed accurately without it. We used the additional filter of the Hough transform in the recognition stage,

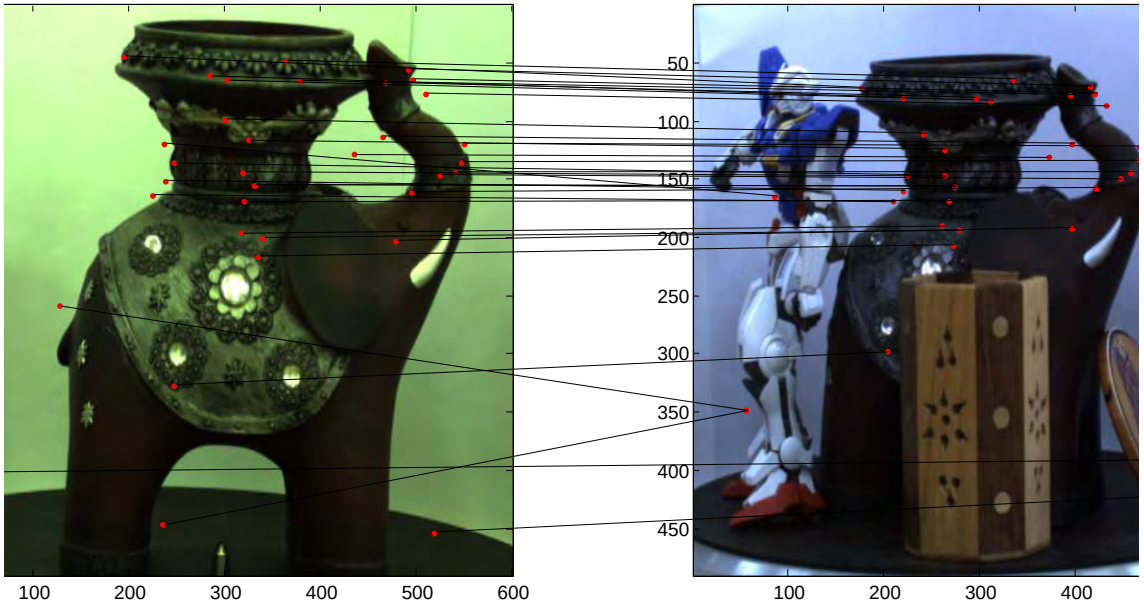


Figure 2.5: SIFT matches between a training and testing image.

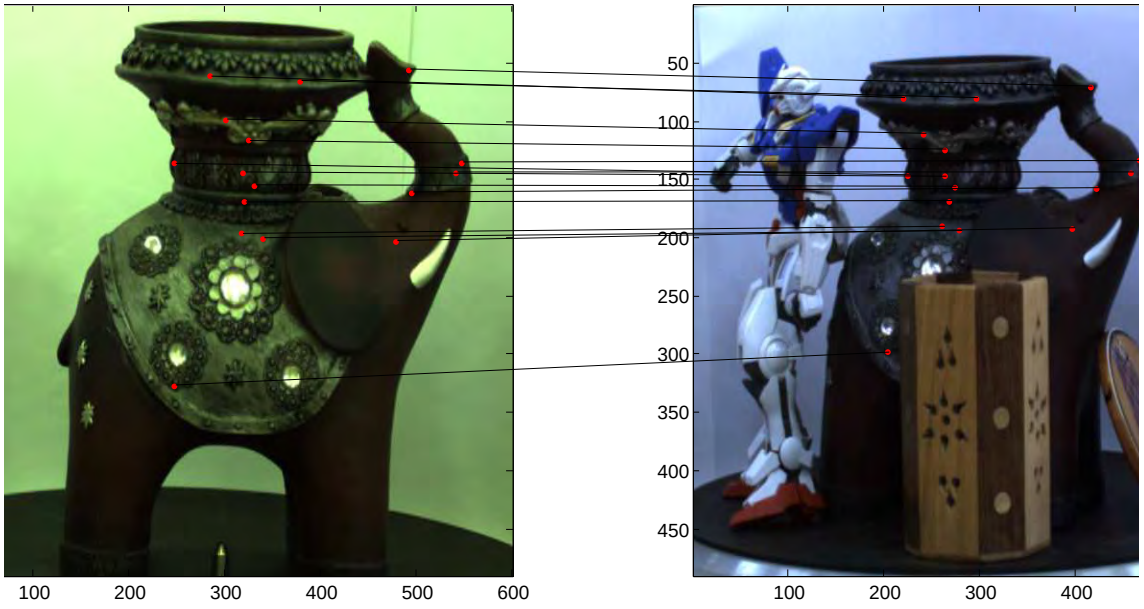


Figure 2.6: SIFT matches after implementing the Hough transform.

not used by [12], which indicates that using geometric constraints is an important step in robust 3D object recognition. Figure 2.7 displays the SIFT matches detected in a test image.

The objects to be recognized in our test images were similarly occluded to those used in the experiments in [12], as can be seen in Figure 2.8.

Figure 2.7: SIFT features detected for the recognized object in a cluttered test image.

Figure 2.8: An example of a test image in [12] and in our dataset.

2.3.6 CONCLUSIONS

In System 1 the object recognition algorithm proposed by [12] was implemented using a local feature detector and descriptor. The view clustering algorithm used multiple training images to create a single pseudo-3D representation of an object. Recognition was performed using SIFT matching which calculates the Euclidean distance between descriptors and the Hough transform which assists in eliminating spurious and ambiguous matches. The test images captured the objects in a cluttered environment with some occlusion. This system however does not provide an estimate of the pose of the object, which may be important for mobile platforms interacting with the object or environment. If the system was unable to recognize the object from a single viewpoint, additional viewpoints would be required. The question arises as to how to select the next best viewpoint as it is not feasible to capture and process images from every viewpoint, which is computationally expensive.

2.4 ACTIVE OBJECT RECOGNITION: SYSTEM 2

In this section we present a novel feature-based 3D active object recognition system which incorporates a next best viewpoint selector and a Bayesian component to integrate the gathered information. This model outputs the confidence the system has in the identity of the object.

2.4.1 DATASET

The training dataset used consists of 20 everyday objects such a cereal box, a salad dressing bottle, a handbag and spray cans. It is much larger than other currently available active vision datasets.

To assemble the training dataset, images were captured for every object at 20° intervals against a plain background on a turntable using a static Prosilica GE1900C camera. There is virtually no vertical deviation and thus is not taken into consideration in our experiments. Each object consists of 18 training images. Figures 2.9 and 2.10 display training images captured for a spray can and a salad dressing bottle respectively.

Figure 2.9: Different viewpoints captured for a spray can taken at $0^\circ/360^\circ, 60^\circ, 120^\circ, 180^\circ$ and 220°.

This dataset contains objects with varying visual textures, heights and sizes. We also included objects that are visually similar in the dataset, as shown in Figure 2.11. Visually similar objects can only be differentiated by looking at certain viewpoints. These were included to test the robustness of the system.

SIFT features are extracted for all training images. For each object, the SIFT features extracted from a training image, say at 20°, is matched to the SIFT features extracted from its neighboring viewpoint, in this case the viewpoint at 40°. Only matched features are then stored in the dataset. This additional matching step helps to eliminate spurious features that have been detected, and selects those features that are stable across viewpoints.

For the test set, the objects used in the training data were also captured at every 20° in a cluttered

Figure 2.10: Different viewpoints captured for a salad dressing bottle taken at $0°/360°, 60°, 120°, 180°$ and $220°$.

Figure 2.11: Objects that are visually similar that appear in our dataset.

environment with significant occlusion. Figures 2.12 and 2.13 display the testing images captured for the spray can and the salad dressing bottle.

2.4.2 ACTIVE VIEWPOINT SELECTION

The aim of the automatic view selection algorithm is to choose the next best viewpoint for object verification and recognition, namely the viewpoint which will provide the most amount of useful information to optimally complete the process. Our system extracts SIFT features and descriptors from all the training data and inputs them into a vocabulary tree data structure, which is then used to weight all the viewpoints. Features are generally used to denote some information or distinctive attribute about the image that can be used to solve a computer vision task. These can refer to points or edges in an image or to more complex structures such as objects. Descriptors are usually defined

Figure 2.12: Testing images captured for the spray can.

Figure 2.13: Testing images captured for the salad dressing bottle.

as vectors used to describe extracted features from a training image. These descriptors are then used to match corresponding features in test images to recognize, in this case, objects. Features detected in the training images need be detectable even under changes in image scale, noise and illumination. Features extracted on the occlusion boundary are generally discarded through the SIFT matching and the Hough transform processes. Thus, there is little to no computational cost added by including these features.

Matching is accomplished by calculating the Euclidean distances between SIFT descriptors (as in [13]), rather than by identifying features associated with the same visual word from the vocabulary tree. This alleviates quantization effects that may be introduced by using the matched visual words.

When considering multiple view object recognition it is essential to chose viewpoints that will provide the most information to reduce the number of images that need to be processed. This in turn reduces the computational expense and processing time of the task and can improve the overall accuracy of the algorithm. We propose a system to weight each viewpoint of an object, and during the recognition process the viewpoints with the highest weightings are selected. Our system also provides a confidence value for the object's identity and thus stops processing new viewpoints once a certainty of 80% is reached.

Vocabulary tree data structure

Features extracted are used as input into a vocabulary tree, which is constructed using hierarchical k-means clustering where similar features are clustered together. In our system, clustering is determined using the SIFT descriptors of each feature. The number of children of each node of the tree is defined by k. Initially k-means clustering is run on all the training data, which defines k cluster centers. The training data is then used to construct k groups, where each group consists of SIFT descriptors closest to a particular cluster center. This process is recursively applied to each group up to some depth D. This process is illustrated in Figure 2.14.

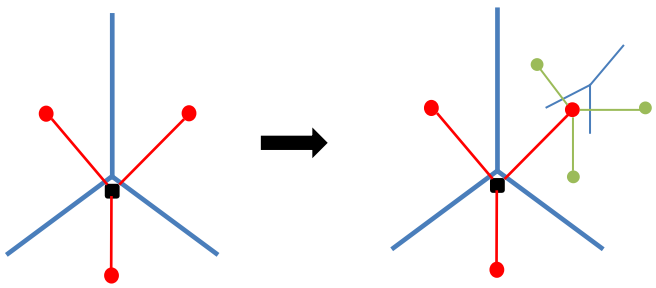


Figure 2.14: An illustration of the process of building a vocabulary tree with branch factor 3 ($k = 3$) and two levels. At each level the descriptors are quantized using the vocabulary tree structure, where in the first layer the descriptor is assigned to the closest of the three red centers. In the second layer it is assigned to the closest of the three green descendants.

Each node of the vocabulary tree has an associated inverted file, which lists references to all the

images which contain instances of that node. Therefore for each node a metric similar to the tf-idf (term frequency-inverse document frequency) is calculated to determine a node's uniqueness. The tf-idf formula presented in [60] is defined in the following way:

$$t_i = \frac{n_{id}}{n_d} \ln \frac{N}{N_i}, \tag{2.6}$$

where n_{id} is the number of times the word, in our case descriptor i, appears in the image/document d divided by n_d, the total number of word descriptors in the document image. This gives us the word frequency in an image. The total number of images in the dataset is given by N and N_i is the number of occurrences of i in the whole dataset, which gives us the *inverse document frequency*. The word frequency describes words/descriptors occurring often in a dataset, and describes it well, while the inverse document frequency downweights words/descriptors that appear often in the dataset. In our calculations we only use the inverse document frequency as we are interested in the frequency of descriptors in the entire dataset and not just in a single image. It gives us a global weighting for each feature given the current dataset. Thus, the equation we use is

$$w_i = \ln \frac{N}{N_i}, \tag{2.7}$$

where *N* is the total number of images in the dataset and N_i is the number images in the dataset with at least one feature that passes through node i in the vocabulary tree.

Using this quantity, a feature's uniqueness can be calculated. This is done in the following way. The feature's path through the vocabulary tree is determined by evaluating the closest cluster centers at each level. A measure of uniqueness is given by the sum of all the weights of the nodes it passes through. The higher the weighting, the more unique the feature. The uniqueness of the viewpoint may then be given by summing these totals for all the SIFT features extracted from that viewpoint. We term this metric the viewpoint weighting. This calculation is performed for every image captured in the training dataset.

It is important to note that SIFT features detected on the background will not negatively affect the weighting since all images were captured using the same background, and their uniqueness weighting will be extremely low. Figure 2.15 displays the viewpoint weightings for a spice bottle object in the dataset. The plot indicates that the most distinctive viewpoint (highest weightings) is at 180°, and the most indistinguishable (lowest weighting) is at 0°. The corresponding images are displayed in Figure 2.16.

It can be seen that the viewpoints with the highest weightings correspond to the images with more distinguishable visual information. The weightings for each viewpoint are used as the criterion for selecting the next best viewpoint.

There a few steps that have been implemented to limit the number of spurious or inconsequential features. Features extracted from the training images are matched to neighboring training images. These matches are then subjected to the Hough transform which enforces geometric constraints. Only these

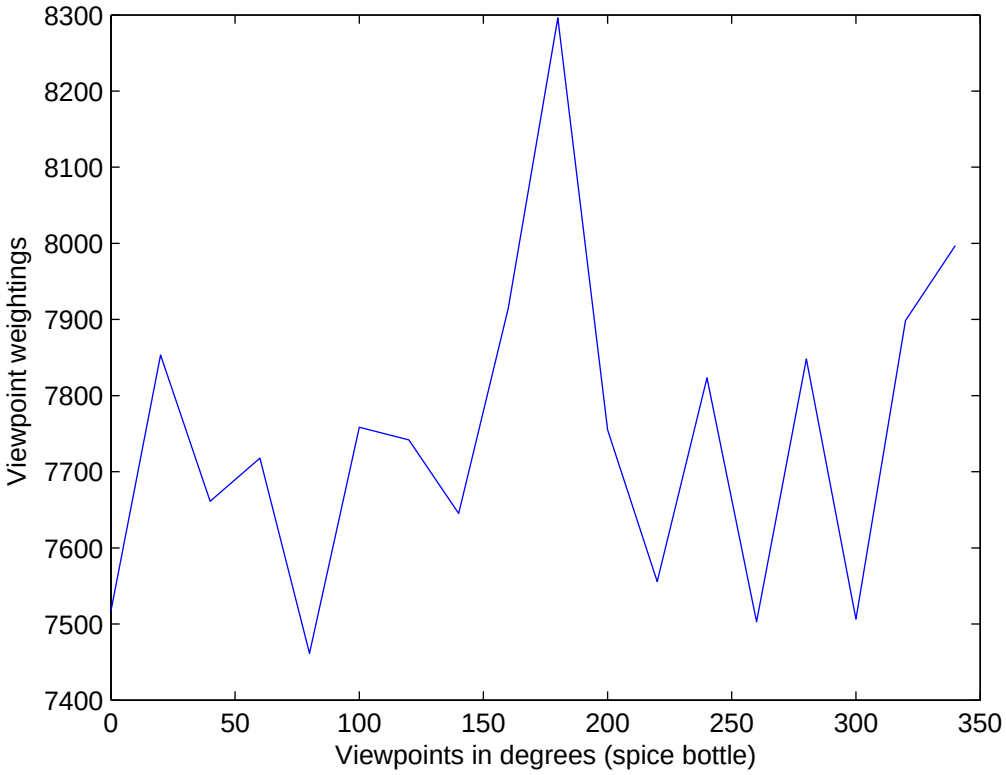


Figure 2.15: Viewpoint weightings for a spice bottle object in the dataset.

Figure 2.16: Viewpoints of the spice bottle corresponding to $180°$ and $0°$ respectively.

features are then used as input to the vocabulary tree. Stop lists are also implemented which discards weights below a certain threshold thus removing irrelevant or abundant features which provide no value but add to the value of the viewpoint.

For object verification, a test image is provided to the system with the necessary object hypothesis. SIFT features are detected and extracted along with their descriptors. These features are matched to the hypothesized object's training images using standard SIFT matching followed by the Hough transform. As mentioned before, we use SIFT matching instead of the vocabulary tree as this alleviates quantization effects that maybe introduced using matching visual words. This matching process is performed to determine the closest training image, which provides the initial pose estimate of the object. Relative to this pose estimate the next best viewpoint selection component selects a view that has the highest uniqueness weighting for that object and which has not been previously visited.

For object recognition, no object hypothesis is given to the system. As in the object verification case, we extract the SIFT features and descriptors from the test image. This is in turn matched to the training images in the dataset, also using SIFT matching and the Hough Transform. We use the number of matches returned from the Hough geometric matching method to select the best matching pose for each object at a given time step. We then align the weightings for each object based on these best pose estimates. The criteria for selecting the next best viewpoint is based on the viewpoint which has the highest combined weighting across all objects in the dataset and has not been previously visited. In the experiments section we show that both these selection methods significantly outperform randomly selecting the next viewpoint.

2.4.3 INDEPENDENT OBSERVER COMPONENT

Our framework is also designed to provide the current belief in the identity of an object. This is provided as a percentage for all objects after each viewpoint has been processed. The system stops selecting and processing viewpoints when the belief in an object's identity is 80% or greater. The vocabulary tree structure used in the active view selection component is altered to store the statistics used for these calculations.

The vocabulary tree built by the active view selection component is modified to include a discrete density function at each leaf node. This represents the likelihood of the feature appearing at least once given a certain object and is used to update the system's belief. These densities are represented as $P(N|O)$ where O is an integer and represents the object and N denotes the node. A representation of the modified tree in shown in Figure 2.17.

The discrete density elements are determined as follows. If any feature from an object's training set, when passed through the vocabulary tree, reaches a leaf node say N_5, then the corresponding element of $P(N_5|O)$ is assigned p_o. Elements that are not reached by this object's training set are assigned p_{no}. Constants p_o and p_{no} are assigned in a 'soft' manner, i.e. no elements are assigned zero. This avoids overcommitted densities. In these experiments, $p_o = 2$ and $p_{no} = 1$ were selected empirically. These parameters are not sensitive to the value selection as long as $p_o > p_{no}$. Once the leaf node densities are populated they are normalized so that all elements sum to one.

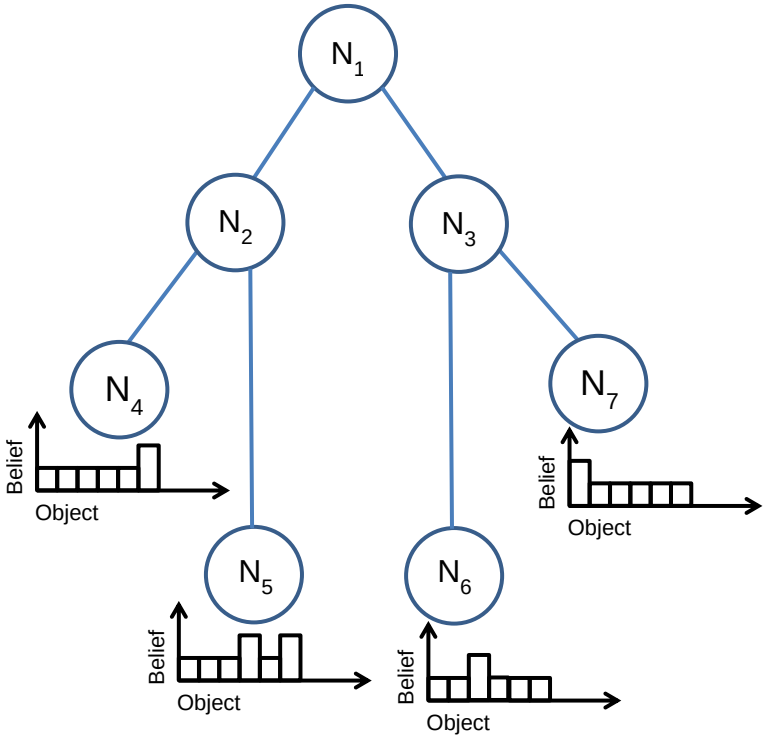


Figure 2.17: An illustration of the modified vocabulary tree.

2.4.4 BAYESIAN PROBABILITIES

A Bayesian framework is adopted due to its flexibility in incorporating diverse modeling choices in a principled manner. Further, in [63] a comparison was conducted between probabilistic (Bayesian), possibilitic and Dempster-Shafer theory approaches to data fusion. They concluded that the probabilistic approach worked best for 3D active object recognition, although all these methods use test images with a single object in an uncluttered environment with no occlusions.

Pipeline

With the tree constructed, the observer component will proceed in the following manner to update its belief:

1. **Initialization**: A uniform prior is assumed over all object hypotheses:

$$P(O) = 1/N, \tag{2.8}$$

   where N is the number of objects. (This initialization is used for both verification and recognition.)

2. **Image processing**: A test image is input to the system. SIFT features are extracted from the test image.

   In the case of object verification, these features are then matched using Lowe's method [13]

and the Hough transform to the training images of the hypothesized object. The image with the most matches to the test image is used as the initial pose.

For object recognition, the features extracted from the test image are matched against all the training images in the dataset. For each object, the best matched image is selected and taken as the initial pose. The next viewpoint selected is taken relative to this initial pose.

3. **Fusion**: Each feature provided by the previous step is cascaded through the vocabulary tree by selecting the children with the closest centroids. The leaf node associated with each feature contains a density as described. Every feature's density is fused recursively with the prior using

$$P(O|N) = \frac{P(N|O)P(O)}{P(N)}, \tag{2.9}$$

where $P(N|O)$ is the density at the leaf node and $P(N)$ is a normalizing coefficient. All nodes are considered independently.

4. **Stopping criteria**: If the posterior belief has a probability of greater than some threshold, μ, for an object the process terminates. We may also stop if a maximum number of viewpoints has been reached, otherwise we take the resulting posterior belief, request a new view from the selector component and return to step 2. If the maximum number of viewpoints has been reached for object recognition, implies that no object accumulated a confidence of $\geq 80\%$.

The system processes for object verification and object recognition are shown in Figure 2.18 and Figure 2.19 respectively.

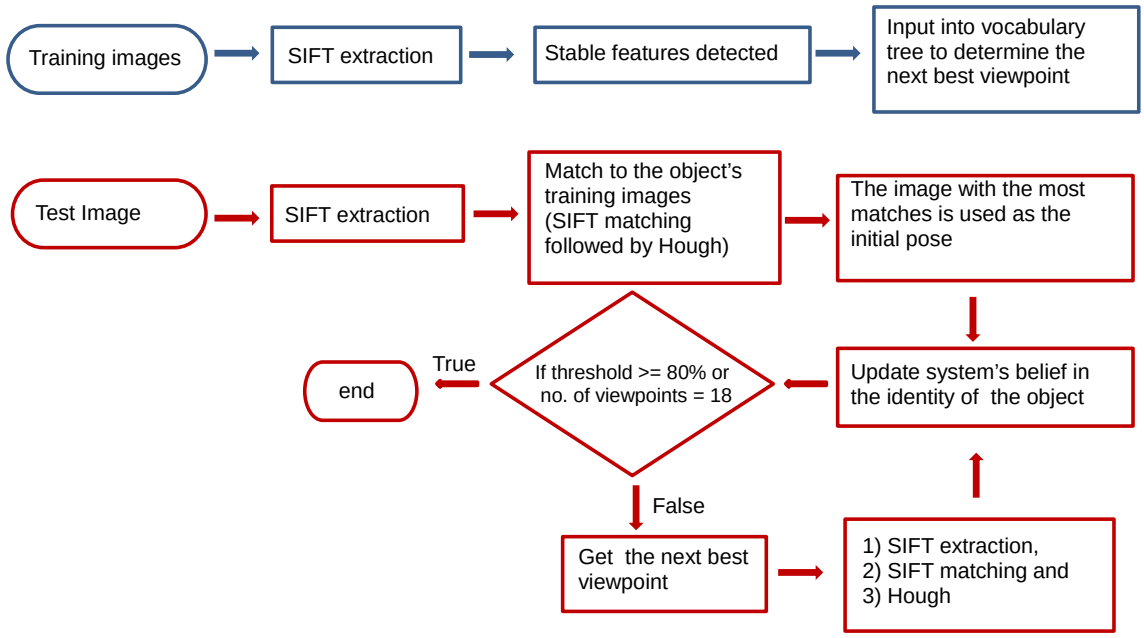


Figure 2.18: Object verification process.

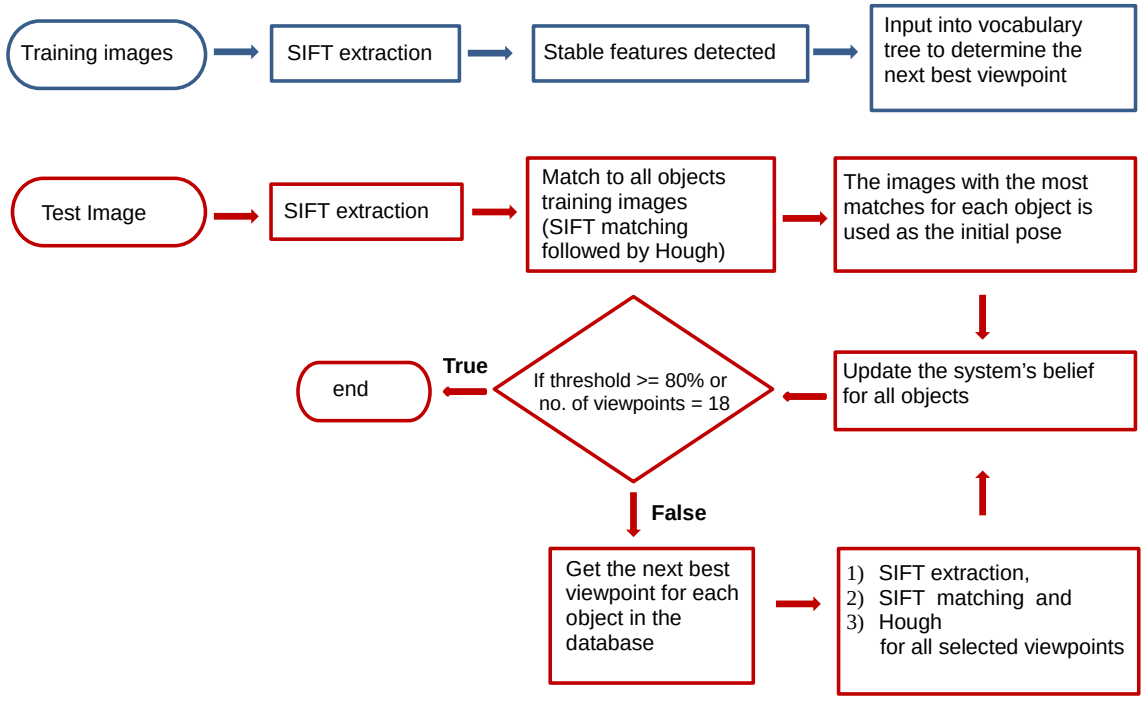


Figure 2.19: Object recognition process.

2.4.5 EXPERIMENTS

The main purpose of an active object verification or recognition system is to reduce the computational expense and improve the accuracy required to determine an object's identity. In addition, our system also provides a measurement for how certain the system is of an object's identity. Test images were captured with the relevant objects in occluded locations in cluttered environments, as shown in Figure 2.20.

An initial test image is presented to the system at an arbitrary pose. The belief probability is then updated as each subsequent view is processed. The system retrieves the next best viewpoint until a confidence or belief probability of $\geq 80\%$ is reached. In accordance with previous state-of-the-art active object recognition systems [1,34,47,52,62,63], the results are compared to randomly selecting the next best viewpoint. When randomly selecting the next best viewpoint, the experiments for both verification and recognition were conducted ten times and the average number of views for each object was recorded.

2.4.6 VERIFICATION

Object verification refers to verifying if a specific object is present in a cluttered image. The system is given the identity of the object that needs to be verified in the test image.

Both our method and randomly selecting the next viewpoint correctly verifies all objects. We are,

Figure 2.20: Examples of test images of occluded objects appearing in a cluttered environment.

however, more interested in the number of viewpoints required to correctly verify an object as this greatly influences the computational expense of the system. Table 2.1 displays the number of views required by each method to verify an object with a belief of 80% or higher.

For each of the 20 objects our method requires fewer viewpoints, in some cases significantly so, to reach this confidence level. This indicates that our method is selecting more informative viewpoints, which can significantly decrease the processing time of an object verification system.

When verifying objects in images, a system may gather evidence to incorrectly verify an object that does not exist in the scene. We tested our system against this pitfall and the results for a sample of objects are displayed as a confusion matrix in Table 2.2. No object in the dataset was incorrectly verified.

2.4.7 RECOGNITION

The system was then tasked to recognize objects that may be occluded in cluttered scenes. This differs from verification in that the object's identity is not known to the system. It has to determine

Table 2.1: Number of views: object verification.

Object	Vocabulary Tree	Random
Cereal	1	1
Battery	1	1
Can 1	3	6.8
Can 2	4	7.5
Curry 1	2	4.4
Curry 2	3	7.5
Elephant	1	1
Handbag	2	3.3
Jewelry box 1	14	16
Jewelry box 2	13	16.5
Lemon bottle	9	14.4
Mr Min	1	2
Salad bottle	12	15
Sauce 1	3	5.8
Sauce 2	3	7.1
Spice 1	6	6.2
Spice 2	15	17
Spray can 1	5	7.8
Spray can 2	5	7.6
Spray can 3	11	16.3
Average	5.7	8.21

Table 2.2: Confusion matrix: verification

	Cereal	Battery	Can 1	Curry 1	Elephant
Obscured Cereal	**1**	0.0015	0.581	0.1063	0.0199
Obscured Battery	0.0802	**1**	0.0145	0.0138	0.0209
Obscured Can 1	0.0374	0.0071	**0.992**	0.116	0.0396
Obscured Curry 1	0.1065	0.1963	0.0491	**0.9996**	0.0834
Obscured Elephant	0.0082	0.0214	0.111	0.0178	**0.9978**

the identity based on which object has accumulated the greatest belief given the current dataset. The system retrieves the next best viewpoint until a confidence or belief probability of 80% is reached for any of the objects in the dataset. The next best viewpoint is selected based on which viewpoint has the highest combined weighting over all objects. The results are displayed in Table 2.3.

Table 2.3: Number of views: object recognition.

Object	Vocabulary Tree	Random
Cereal	1	1
Battery	1	2
Can 1	5	15
Can 2	10	18
Curry 1	4	5.8
Curry 2	7	8.8
Elephant	2	8.3
Handbag	3	5.1
Jewelry box 1	16	18
Jewelry box 2	15	18
Lemon bottle	14	16
Mr Min	2	3.1
Salad bottle	15	18
Sauce 1	4	10
Sauce 2	4	9.2
Spice 1	16	18
Spice 2	18	18
Spray can 1	9	17.5
Spray can 2	11	13
Spray can 3	18	18
Average	8.75	12.04

The vocabulary tree method correctly recognizes 18 out of the 20 objects, while randomly selecting the next viewpoint only recognizes 14 out of the 20 objects. The two objects not recognized by the vocabulary tree method are displayed in Figure 2.21 and Figure 2.22. As can be seen from these test images, the objects are significantly occluded and thus the system does not gain enough confidence ($\geq 80\%$) to correctly recognize these objects.

An additional measure of interest here is the number of viewpoints required to correctly recognize an object. Table 2.3 displays the number of views required for each object in the dataset to reach a confidence level of 80% for object recognition. If all 18 viewpoints have been processed and the confidence of the correct object is still below 80% then the object has not been recognized. The vocabulary tree method clearly outperforms randomly selecting the next viewpoint. It is more accurate and requires fewer views for all objects to attain a confidence of $\geq 80\%$. This leads to a significant decrease in computational expense and processing time for recognizing objects which are occluded

Figure 2.21: The spice bottle object that was not recognized.

Figure 2.22: The spray can object that was not recognized.

in cluttered environments. The same methods are used to extract data from the query image and to integrate the information, so processing fewer viewpoints implies less computational time for our method than randomly selecting the next viewpoint.

The difference in information provided by the varying choice of viewpoints can also be shown. Figures 2.23 and 2.24 display the increase in belief after each view for the 'Curry 1' object for verification and recognition respectively. We can see that even after the second view our method has a much higher belief than randomly selecting a viewpoint for both verification and recognition. After four views in the case of verification and five views for recognition, our method reaches a confidence level of 1.

To remove any bias that the initial image pose may have introduced into the system, all 18 views of the battery object were used as input, in turn, as the initial pose for object recognition. The results are

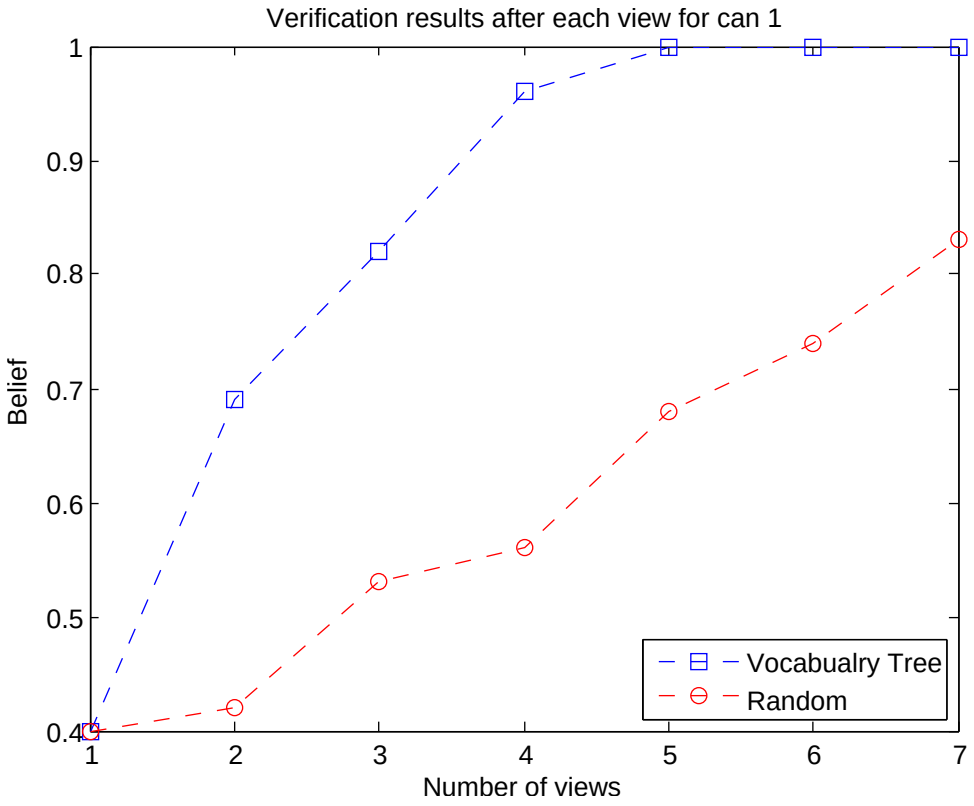


Figure 2.23: Confidence values after each view for verification.

shown in Figure 2.25.

Irrespective of the initial image given, our system still outperforms randomly selecting the next view in all but one instance, which is at view 15 (and correlates to the view at 300°). This could be because the next best viewpoint selected by our algorithm happened to be a viewpoint where the object was severely occluded in the test image.

A multiple view object recognition system without an active viewpoint selector generally selects the next viewpoint randomly. We have shown that our 3D active object recognition system outperforms randomly selecting the next viewpoint. Performance is based on the number of viewpoints required to accurately recognize an object and the overall recognition accuracy of the system.

2.5 COMPARISON OF ACTIVE OBJECT RECOGNITION SYSTEMS

A number of methods have explored active object recognition [1, 18, 19, 52, 62], but use experimental setups not directly comparable to the system described here. Active 3D object recognition systems usually include different methods for 3D object modeling, selecting the next best viewpoint and fusion of the extracted data. The state-of-art method described in Kootstra et al. [1], which uses an activation model, was adapted to run on our dataset and compared to our vocabulary tree method.

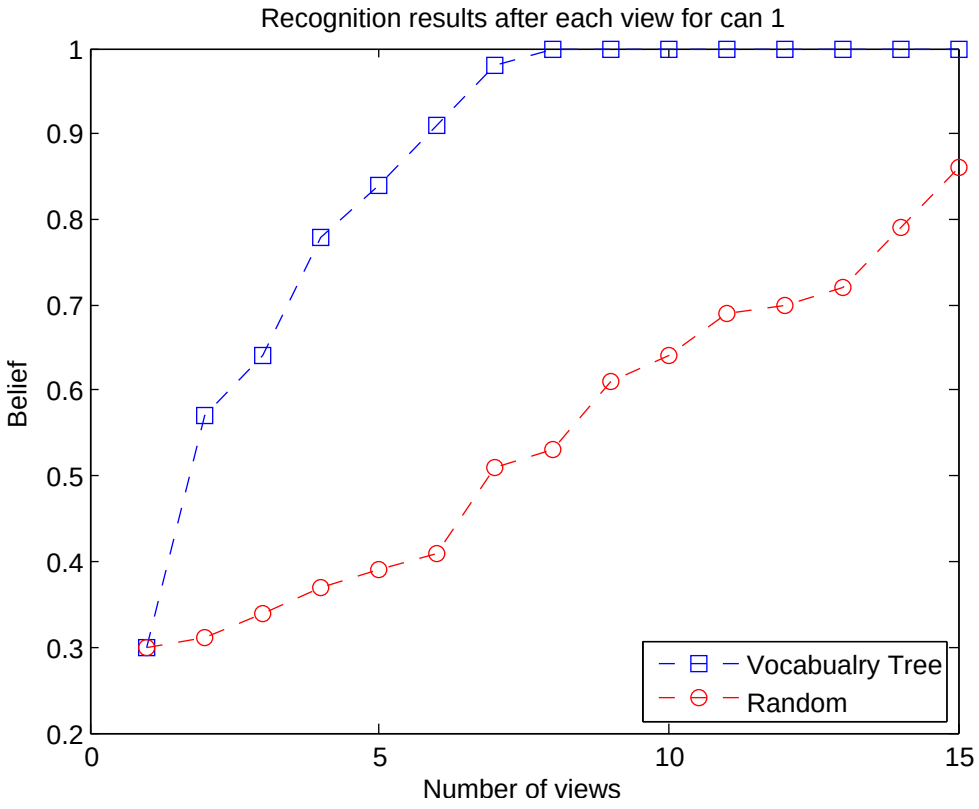


Figure 2.24: Confidence values after each view for recognition.

These two methods have different approaches to the next best viewpoint selection and update strategies, but both methods make use of SIFT features and descriptors. Very few of the available active object recognition methods use local features that are robust to occlusion. The vocabulary tree method outperforms the activation model based on the time taken to process viewpoints and the overall recognition. In section 2.5.1 we give a brief description of the activation model presented in Kootstra et al. [1]. Experiments comparing our method and the activation model are detailed in section 2.5.2.

2.5.1 ACTIVATION MODEL

In [1], to capture images for their dataset a camera mounted on a mobile platform. It followed a circular trajectory around an object of interest which was placed in the center. The platform stops every 10° to capture an image of the object for the training data. SIFT is then used to detect features in the images. Features that are visible from two sequential images, which in this case are taken 10 degrees apart, are considered stable features and only these are retained. The use of stable features removes features that are very sensitive to rotation, translation and other affine transformations. These are used to segment the object from the background. Object segmentation is achieved by noting that as the robot rotates about the object, the background features are displaced substantially more than the object features. Moreover, the assumption is that the robot moves on a flat surface and as such there is

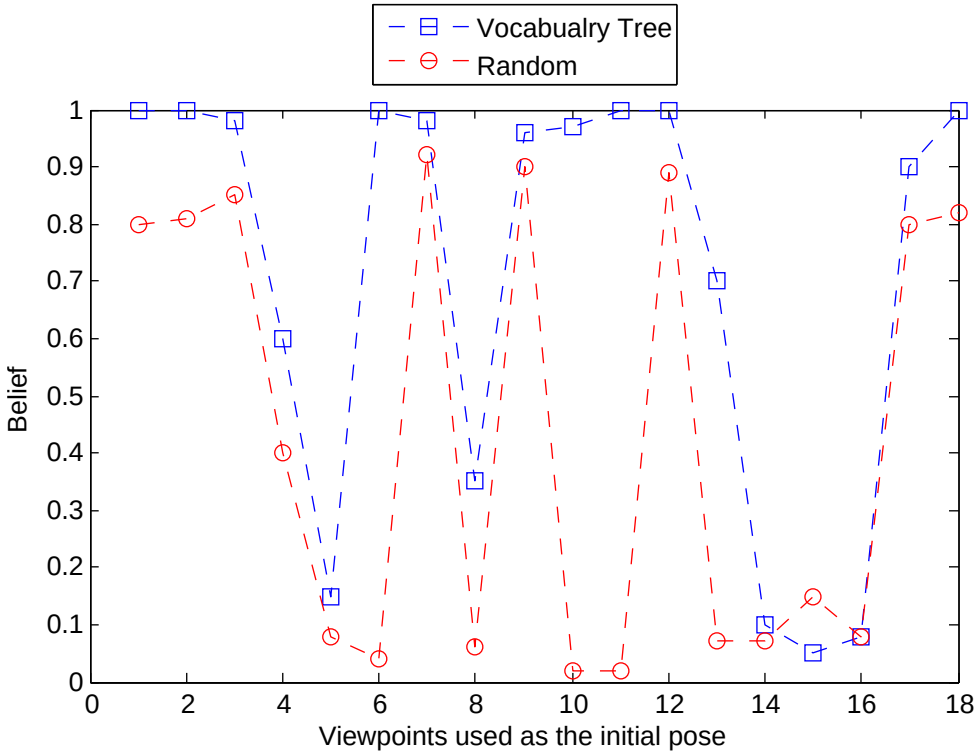


Figure 2.25: Confidence values after each view for recognition.

little change in the vertical components of the positions of the features. Thus the task of segmenting the object is to find stable features that satisfy the following condition:

$$(|x_i - x_j| \leq x_\mu) \wedge (|y_i - y_j| \leq y_\mu),$$

where (x_i, y_i) is the location of the feature in the current image and (x_j, y_j) is the position of the same feature as seen in the previous image. A feature is a match to its nearest neighbor in the previous image if the Euclidean distance between their descriptors is less than 0.6. The original criterion for matching as defined by [13] was in checking if the ratio of the closest to the second closest match was ≤ 0.8. The authors used this alternative method as it performed best on their data. The parameters x_μ and y_μ are given in units of pixels. Each feature is assigned the identity of the object (ID), and pose θ. Models of different objects are kept separate.

In [1], the next best viewpoint is selected by calculating an activation value. An activation value is calculated between the query image and each training image in the dataset. The activation value is determined by summing all the differences in the distances of SIFT descriptors of the matched features, which is then divided by the number of features detected in that particular image. The closer the query object image is to one of the viewpoints of a model the higher the activation value of that viewpoint. The viewpoint with the highest activation is selected as the next best viewpoint.

For every new viewpoint that needs to be selected, the query image has to be matched to every image in the dataset and the activation value recalculated each time.

2.5.2 EXPERIMENTS

In this experiment, the two active vision algorithms were compared using their original format. In subsequent chapters other experiments are conducted, such as substituting the next best viewpoint selection algorithm described in this chapter into the activation model method to determine if there is improvement or decline in performance. This also helps to isolate the contribution of the model and the viewpoint selection strategy on the results.

The stopping condition for our system is set such that an object is considered recognized when the belief or confidence of the system in the identity of an object is $\geq 80\%$. A stopping condition was not specified for the activation model in the original paper. However, a stopping condition was communicated by the authors. The condition was to place a lower bound threshold on the ratio of the largest to the second largest activation values. If this ratio exceeds the threshold then the object with the largest activation value corresponds to the query object. The threshold was set to 1.25 in the experiments (determined empirically), and the model with the largest activation value corresponds to the object for which we are searching.

Both methods were presented with the same initial test image, and each method was halted when its stopping criteria were met. The number of objects recognized, the computational time and the number of views taken to recognize the objects are used to measure the performance of these two systems.

2.5.3 ADAPTATION OF ACTIVATION MODEL

Changes were made to the activation model method to ensure that there was no bias towards either method in the experiments. The first change was implemented because the setup we used to capture the training and testing images differed from that presented in [1].

As mentioned, the activation models segmentation process assumes that a robot rotates about a fixed object of interest. This means that the features that belong to the object are close to the center of rotation, and as a result move substantially less as compared to the features that belong to the background. As a result, a stable feature belongs to the object of interest if its position between two consecutive images satisfies the conditions $(|x_i - x_j| \leq x_\mu = 12)$ and $(|y_i - y_j| \leq y_\mu = 4)$.

In our dataset, however, images were captured with the camera fixed and the object of interest placed on a rotating turntable. As a result, a stable feature belongs to the background if its location does not change between two consecutive images:$(|x_i - x_j| \leq x_\mu = 4)$ and $(|y_i - y_j| \leq y_\mu = 4)$. Here x_μ and y_μ are also defined in units of pixels.

The second change was the implementation of the stopping criterion.

2.5.4 RESULTS

Tables 2.4 and 2.5 indicate which objects were recognized and the time taken for recognition using the activation model method by [1] and our vocabulary tree method respectively.

Table 2.4: Activation model method.

Object	Recognized	Time taken (s)
Cereal	Yes	283.99
Battery	Yes	351.97
Can1	No	
Can2	No	
Curry 1	Yes	321.03
Curry 2	Yes	1163.06
Elephant	Yes	349.34
Handbag	No	
Jewelry box 1	No	
Jewelry box 2	No	
Lemon bottle	No	
Mr Min	Yes	253.59
Salad bottle	Yes	569.11
Sauce1	No	
Sauce2	No	
Spice 1	No	
Spice 2	No	
Spray can 1	No	
Spray can 2	Yes	269.5
Spray can 3	No	
Number of Objects Recognized	8 / 20	

The vocabulary tree method correctly recognizes 18 out of the 20 objects (90%) from the dataset while the activation model which only correctly recognizes 8 objects (40%). A factor that could negatively influence the recognition rate for the activation model method is that when matches are found between the query and dataset images, no additional filters are implemented to remove false or ambiguous matches. This could lead to incorrect activation values being calculated.

The vocabulary tree model is also on average 37 times faster per object than the activation model in terms of the computational cost involved. This is due to the fact that every feature from the test image must be matched against every feature in the dataset to calculate the activation value for each image, and this process is carried out after each new viewpoint is selected. We match the test image to a

Table 2.5: Vocabulary tree method.

Object	Recognized	Time taken(s)
Cereal	Yes	4.6
Battery	Yes	5.9
Can 1	Yes	22
Can 2	Yes	41
Curry 1	Yes	13
Curry 2	Yes	34
Elephant	Yes	14.2
Handbag	Yes	10.38
Jewelry box 1	Yes	55.8
Jewelry box 2	Yes	54.1
Lemon bottle	Yes	50
Mr Min	Yes	6.4
Salad bottle	Yes	40.3
Sauce 1	Yes	12.6
Sauce 2	Yes	14.4
Spice 1	Yes	32
Spice 2	No	68
Spray can 1	Yes	31.1
Spray can 2	Yes	41.5
Spray can 3	No	74.8
Number of Objects Recognized	18 / 20	

predetermined next best viewpoint for each object. The number of images used in our method thus is determined by the number of objects in the dataset.

Table 2.6 details the number of viewpoints required by each method to correctly recognize an object. The results show that when the activation model correctly recognizes an object it generally requires fewer viewpoints than the vocabulary tree method. This could be due the fact that in the activation mode when a test image is presented to the recognition system it is matched to every image in the training dataset to find the one with the highest activation value. It chooses the next viewpoint based on the test image, whereas in the vocabulary tree method the next best viewpoint is determined based on the object characteristics in the training dataset. Although this method requires fewer viewpoints, it correctly recognizes significantly fewer objects with greater computational expense than the vocabulary tree method.

Table 2.6: Number of views: object recognition.

Object	Vocabulary Tree	Kootstra et al.
Cereal	1	1
Battery	1	1
Can 1	5	18
Can 2	10	18
Curry 1	4	4
Curry 2	7	11
Elephant	2	1
Handbag	3	18
Jewelry box 1	16	18
Jewelry box 2	15	18
Lemon bottle	14	18
Mr Min	2	1
Salad bottle	15	5
Sauce 1	4	18
Sauce 2	4	18
Spice 1	16	18
Spice 2	18	18
Spray can 1	9	18
Spray can 2	11	6
Spray can 3	18	18
Average	8.75	12.3

2.6 CONCLUSIONS

Active object recognition is important because it provides object recognition systems with a mechanism to select more informative viewpoints and thus reduce computational costs and improve accuracy. In this chapter a new framework for active object verification and recognition was introduced, consisting of an selector and an observer component. The selector determines the next best viewpoint and the observer updates the belief hypothesis and provides feedback to the system. The observer component works independently of the selector and thus any exploration or manipulation of an object can occur for selecting the next best viewpoint without interfering with the observer component. This framework, which has proven to work efficiently, can be applied to any active vision task.

A new database was also introduced in this chapter which is much larger (containing 20 objects) than other currently available active vision databases. It contains objects with varying visual tex-

tures, heights, sizes and visually similar objects. In terms of the experiments,the test images used contained these objects appearing in cluttered environments with significant occlusions. Other active vision databases available use test images with a single object with little or no clutter or occlusions. Therefore, the databases used in these experiments are significantly more complex and difficult than currently available active vision databases.

To select the next best viewpoint, features appearing in every viewpoint in the training dataset were weighted based on their uniqueness. This was determined using an inverted file derived from the vocabulary tree data structure. The vocabulary tree data structure used to generate the feature statistics can easily incorporate more objects with little or no additional computational complexity. For verification, the viewpoint with the highest weighting for the object to be verified was selected as the next best viewpoint. In the case of object recognition, the viewpoint with the highest weighting over all objects was selected as the next best viewpoint. Both these methods proved to be significantly better than randomly selecting the next viewpoint. Bayes' rule was used to update the belief hypothesis and provide feedback to the system. The path of each matched feature in the test image was traced through the vocabulary tree and the statistics contained in the leaf node were used to update the belief hypothesis. New images were only captured when the belief was below a predefined threshold. This reduces the computational and processing time as only the minimal number of images will be processed to complete the object recognition task. Our system also provides a measure of certainty for the object's identity. This system uses test images where the object to be verified or recognized is occluded and appears in a cluttered environment. Even with these difficulties, the system presented correctly verifies all objects and correctly recognizes 90% of the objects. It also requires fewer viewpoints than randomly selecting the next viewpoint for both tasks, and in some cases significantly so.

The vocabulary tree active object recognition system was then compared to another state-of-the-art active object recognition system [1]. Our method outperforms the method presented in [1] primarily because it recognizes a larger set of objects and is computationally more efficient. These experiments show the successful use of our active object recognition system for 3D object verification and recognition for significantly occluded objects in cluttered environments.

CHAPTER THREE

PROBABILISTIC OBJECT AND VIEWPOINT MODELS[1]

3.1 INTRODUCTION

In the previous chapter we described an active 3D object recognition system for recognizing occluded objects appearing in a cluttered environment. The system used SIFT features and a vocabulary tree data structure to weight the uniqueness of viewpoints and to update the system's confidence in the identity of an object. The system correctly recognized 18 out of the 20 objects in the database. The two objects that were not recognized, a spice bottle and a spray can, were significantly occluded in the test images and the system could not gather enough evidence to reach a threshold of $\geq 80\%$.

To improve the recognition accuracy of the system, in this chapter we conduct several experiments with different feature integration probability models to address the issue of recognizing significantly occluded objects. We continue to use a Bayesian framework for integrating the information in a principled manner across multiple views. Bayesian methods have proved effective in many active vision scenarios, and general frameworks for active sensing have been proposed [71], along with specific models for scene exploration and tracking from surveillance videos [44] and object recognition from a mobile platform [1]. In many of these cases, however, attention is paid to the general problems of finding optimal methods for fusing data and planning sensing strategies while assuming that a probabilistic model for the phenomenon of interest (object/environment) is given. By assuming simple

[1]Related publication:

- Natasha Govender, Jonathan Warrell, Philip Torr and Fred Nicolls, "Probabilistic Object and Viewpoint Models for Active Object Recognition", Africon, September 2013.

probabilistic models and using highly controlled datasets, general methods for fusion and planning are easily demonstrated. However, it is unclear how well such models cope with unconstrained settings.

The primary question is how to represent an object probabilistically in order to perform effective active object recognition in the challenging scenario of highly cluttered test scenes. While adopting a standard Bayesian framework for data fusion, two probability methods are presented. The first method is defined for object recognition and the second for object and pose recognition. For each of these methods, three different likelihoods models are investigated to update the Bayesian framework and these are shown to perform well on our challenging dataset. The likelihood models include using independent features, a binary model and an occlusion model. These will be explained in detail in section 3.3.

As described in chapter 2, the method presented here is also based on SIFT features. Drawing on the techniques of [12, 13] for non-active recognition, it incorporates geometric structure by filtering the features processed at a given view using the Hough transform. This transform is used to identify the most likely transformation from a training example. However, the method presented in Chapter 2 does not explicitly include the transformation as part of the probabilistic model, and does not model the background or occlusion process. We deal with these issues in this chapter by introducing a background distribution and latent occlusion and transformation variables, and incorporate this distribution into both object and pose recognition. The probabilistic models considered here are not dependent on the particular low-level representation choices.

In section 3.2 we discuss the various methods that have been used for data fusion in active and non-active recognition. Section 3.3 defines the problem statement and the Bayesian framework for object recognition and object and pose recognition. The three likelihood models that are used in the Bayesian frameworks are described in section 3.4. Experiments conducted using these approaches and results are discussed in section 3.5 with conclusions in section 3.6.

3.2 RELATED WORK

A wide range of general frameworks for active vision and active sensing have been explored, including Bayesian approaches [34, 44, 51], discriminative approaches [1, 18], and approaches based on other theoretical models such as possibilistic and Dempster-Shafer theory [19, 63]. As previously discussed a Bayesian framework is adopted due to its flexibility in incorporating diverse modeling choices in a principled manner. Further, in Borotschnig et al. [63] a comparison was conducted between probabilistic (Bayesian), possibilitic and Dempster-Shafer theory approaches to data fusion. Experiments were conducted using parametric eigenspaces on eight objects, two of which were visually similar. They concluded that Bayesian reasoning was the most consistent scheme for fusion of additional information and worked best for 3D active object recognition, although all these methods

use test images with a single object in an uncluttered environment with no occlusions. The probabilistic approach was also the fastest in their experiments. A limitation of this model is that, because a global eigenspace representation is used, the model copes poorly with recognizing highly occluded objects and requires uncluttered test sequences.

Bayesian approaches also allow flexibility in the type of input that is used to update the system. These include local features [51], entropy values [71] and neural networks [34]. The method by Kootstra et al. [1] uses SIFT features but since their update model is non-probabilistic and there is no natural way to build additional assumptions into the framework.

3.3 BAYESIAN ACTIVE OBJECT AND POSE RECOGNITION

We redefine the active object recognition task presented in Chapter 2 in a manner which can be used in this probabilistic setting.

Problem Statement: At training time, for each object $o = 1, ..., O$ we capture a set of images, one at each of a series of P regularly spaced training views around the object, indexed by their viewing angle. For example, $\theta \in \{0°, 20°, 40°, ..., 340°\} = \Theta$ and $P = |\Theta|$. For simplicity we consider only varying the viewing angle around one axis (e.g. vertical), although minimal changes are necessary to incorporate poses from across a viewing sphere. We thus have a training image $I^{\text{train}}_{o,\theta}$ for each object and view pair.

At test time we are presented with one of the training objects, and must identify

- the object present $o^\star$, and possibly
- the orientation of the object,

which may be specified by the training pose $\theta^\star$ corresponding to a reference test view. We are allowed to capture images of the test object at a sequence of test views, $\delta_1, \delta_2, ... \in \{0°, 20°, 40°, ...340°\}$, where the angles δ_t can be in any order. We label the image corresponding to the test view t as $I^{\text{test}}_{\delta_t}$, and $\delta_1 = 0°$ is treated as a reference view (i.e. $I^{\text{train}}_{o^\star,\theta^\star}$ will denote the training view we believe corresponds to $I^{\text{test}}_{\delta_1}$). The active object recognition algorithm presented in Chapter 2 selects both the sequence of test views $\delta_1, \delta_2, ...$ and when to stop capturing further poses, and generates an output.

Bayesian probability provides a framework within which to build algorithms, and we give the general outline of two methods. The first outlines the approach for object recognition, and the second the approach for object and viewpoint/pose recognition. Each of these methods requires us to specify a likelihood model for objects or poses, and we look at three specific options for these in section 3.4 to update the Bayesian framework. The viewpoint selection strategy explained in Chapter 2 is used for each of the methods.

Bayesian algorithm for object recognition: In this case we require a probability model for our image feature representation of image I, $\mathbf{f}_I$ given object o: $P(\mathbf{f}_I|o)$. At a given time step t during test time we are interested in estimating $P_t(o) = P(o|\mathbf{f}^{\text{test}}_{\delta_1}, ..., \mathbf{f}^{\text{test}}_{\delta_t})$, that is, the probability of each object given the images we have seen so far (writing $\mathbf{f}^{\text{test}}_{\delta}$ for $\mathbf{f}_{I^{\text{test}}_{\delta}}$). Assuming the images seen to be independent samples from the object's probability model, we can estimate $P_t(o)$ recursively using Bayes theorem:

$$P_t(o) = \frac{P(\mathbf{f}^{\text{test}}_{\delta_t}|o)P_{t-1}(o)}{\sum_o P(\mathbf{f}^{\text{test}}_{\delta_t}|o)P_{t-1}(o)}. \tag{3.1}$$

If we have no information prior to testing, setting $P_0(o) = 1/O$ is an appropriate initial distribution. This update mechanism is combined with the next viewpoint selection strategy. The stopping criteria are to cease capturing further views when $\max(P_t(o)) > \mu$ with μ a threshold parameter, and output $o^\star = \text{argmax}(P_t(o))$, or when all 18 poses have been processed which indicates the object was not recognized.

Bayesian algorithm for object and pose recognition: For object recognition we assume that the images we view at test time are generated independently given the test object o. In general this will not be the case, since we expect there to be high correlations between the images we see at particular poses. We can build this information into our approach by using separate probability models for each object/pose combination: $P(\mathbf{f}|o, \theta)$. Now we are interested in estimating at each time step t a distribution $P_t(o, \theta) = P(o, \theta|\mathbf{f}^{\text{test}}_{\delta_1}, ...\mathbf{f}^{\text{test}}_{\delta_t})$, where we denote by $P_t(o, \theta)$ the probability at time t that the test object is o and the pose at the reference test angle $\delta_1 = 0^\circ$ corresponds to the training view θ. Again, we can estimate this distribution recursively:

$$P_t(o, \theta) = \frac{P(\mathbf{f}^{\text{test}}_{\delta_t}|o, \theta + \delta_t)P_{t-1}(o, \theta)}{\sum_o P(\mathbf{f}^{\text{test}}_{\delta_t}|o, \theta + \delta_t)P_{t-1}(o, \theta)}, \tag{3.2}$$

where we note that the offsets δ_t are required to select the correct likelihood models to combine at time step t. As in the approach for object recognition, a uniform prior can be selected for $P_0(o, \theta)$. If we are primarily interested in identifying the correct test object, we can further calculate

$$P_t(o) = \sum_\theta P_t(o, \theta), \tag{3.3}$$

at each time, and again stop when $\max(P_t(o)) > \mu$, outputting $o^\star = \text{argmax}(P_t(o))$ and $\theta^\star = \text{argmax}_\theta(P_t(o^\star, \theta))$.

3.4 PROBABILISTIC MODELS FOR OBJECT AND POSE RECOGNITION

Having described the Bayesian approaches for object (equation 3.1) and object and pose recognition (equation 3.2), we outline below a number of possible models that can be used for the likelihoods. We discuss three models, which incorporate increasing levels of structure. We are particularly interested

in identifying objects that may be occluded at test time, as will be explored in the experimentation, and the final two options below explicitly build this into the generative model. Having described the Bayesian approaches for object (equation 3.1) and object and pose recognition (equation 3.2), we outline below a number of possible models that can be used for the likelihoods. We discuss three models, which incorporate increasing levels of structure. We are particularly interested in identifying objects that may be occluded at test time, as will be explored in the experimentation, and the final two options below explicitly build this into the generative model.

3.4.1 INDEPENDENT FEATURES

For our first likelihood model we assume that we have access to a preprocessing method to extract a sparse set of visual words from each training/test image (as noted in the experimentation, we will use a vocabulary tree for this purpose). Letting $\mathcal{N} = \{1, ..., N\}$ be the set of all visual words (the dictionary), and assuming initially for convenience that all images contain the same number of words, M, we can represent training image $I^{\text{train}}_{o,\theta}$ by the vector $\mathbf{f}^{\text{ind}}_{o,\theta} \in \mathcal{N}^M$, where the ordering of entries in $\mathbf{f}^{\text{ind}}_{o,\theta}$ is generated by assuming a fixed ordering strategy, such as top-left to bottom-right.

For object recognition we can estimate the per-object distribution for individual features based on whether we observe an individual feature associated with an object during training:

$$P(n|o) \quad \propto \quad p_a[(\sum_{\theta,m} f^{\text{ind}}_{o,\theta}(m) = n) = 0] + p_b[(\sum_{\theta,m} f^{\text{ind}}_{o,\theta}(m) = n) > 0], \tag{3.4}$$

where $n \in \mathcal{N}$ is a particular visual word and p_a and p_b are parameters of the distribution controlling the probabilities when node n is not seen and is seen respectively (which relate to the p_{no} and p_o parameters in Chapter 2). The likelihood for a test image with features $\mathbf{f}^{\text{ind}}$ is then formed simply by treating all observed visual words as independent draws from equation 3.4:

$$P(\mathbf{f}^{\text{ind}}|o) = \prod_{m=1,...,M} P(f^{\text{ind}}(m)|o). \tag{3.5}$$

For object and pose recognition, an object and pose model can be formed similarly by storing $P(n|o, \theta)$ for each combination (removing the summations across θ in equation 3.4), and using these to form $P(\mathbf{f}^{\text{ind}}|o, \theta)$ similarly to equation 3.5. Finally we note that, although we assume each image to contain M features, we can simply build a dependence on M into equation 3.5:

$$P(\mathbf{f}^{\text{ind}}|o) = P(M_{\mathbf{f}^{\text{ind}}}|o) \prod_{m=1,...,M_{\mathbf{f}^{\text{ind}}}} P(f^{\text{ind}}(m)|o), \tag{3.6}$$

where $M_{\mathbf{f}^{\text{ind}}}$ is the length of $\mathbf{f}^{\text{ind}}$. If we assume $P(M_{\mathbf{f}^{\text{ind}}}|o)$ to be uniform within certain bounds (e.g. always between 10–1000 features) these factors will cancel in the Bayesian updates.

3.4.2 BINARY MODEL

The independent features model above does not represent geometric structure in any way, and as such is susceptible to noise. This will especially be a problem in our experimental setup, in which we are interested in recognizing objects amongst clutter. We thus present here a simple likelihood model which embeds a notion of geometric structure using the Hough transform.

For object recognition, we set $\mathbf{f}^{\text{round}}$ to be a binary indicator vector, $\mathbf{f}^{\text{round}} \in \mathbb{B}^O$, where $\mathbb{B} = \{0, 1\}$ and we have $|\mathbf{f}^{\text{round}}| = 1$ (i.e. there is a single 1, and [(number of objects $- 1$)zeros]. The position of the 1 in $\mathbf{f}^{\text{round}}$ indicates the object model with the highest number of matching words after applying the Hough transform. Explicitly, we denote by $H_{o,\theta}(I)$ the maximum number of matches for a transformation between image I and training image $I^{\text{train}}_{o,\theta}$, which in the case of the Hough method is $H_{o,\theta}(I) = \max_t H_t(I, I^{\text{train}}_{o,\theta})$. Thus we can write

$$\mathbf{f}^{\text{round}}_I(o) = \begin{cases} 1 & \text{if } o = \text{argmax}_{o'} \text{argmax}_\theta H_{o',\theta}(I) \\ 0 & \text{otherwise,} \end{cases} \tag{3.7}$$

where ties are broken arbitrarily. Our likelihood model then assumes a simple form depending on a single parameter, p_c, according to

$$P(\mathbf{f}^{\text{round}}|o) = p_c[\mathbf{f}(o) = 1] + ((1 - p_c)/(O - 1))[\mathbf{f}(o) = 0]. \tag{3.8}$$

That is, we assume an object o generates a binary vector with $\mathbf{f}^{\text{round}}(o) = 1$ with probability p_c, and a vector with a 0 positioned elsewhere with probability $1 - p_c$. This probability being is then evenly divided between the [(number of objects) $- 1$] other cases.

A similar likelihood model can be formed for object and pose recognition. Here, let $\mathbf{f}^{\text{round}} = (\mathbf{f}^{\text{round,obj}}, \mathbf{f}^{\text{round,pose}})$ with $\mathbf{f}^{\text{round,obj}} \in \mathbb{B}^O$ and $\mathbf{f}^{\text{round,pose}} \in \mathbb{B}^P$. Then we set $\mathbf{f}^{\text{round,obj}}(o)$ similarly to equation 3.7 and

$$\mathbf{f}^{\text{round,pose}}_I(\theta) = \begin{cases} 1 & \text{if } \theta = \text{argmax}_{\theta'} \max_o H_{o,\theta'}(I) \\ 0 & \text{otherwise.} \end{cases} \tag{3.9}$$

The likelihood model is then

$$\begin{aligned} P(\mathbf{f}^{\text{round}}|o, \theta) \quad = \quad & p_c[\mathbf{f}^{\text{round,obj}}(o) = 1 \wedge \mathbf{f}^{\text{round,pose}}(\theta) = 1] + \\ & ((1 - p_c)/(O \cdot P - 1))[\mathbf{f}^{\text{round,obj}}(o) = 0 \vee \mathbf{f}^{\text{round,pose}}(\theta) = 0]. \end{aligned} \tag{3.10}$$

3.4.3 OCCLUSION MODEL

The binary model incorporates geometric structure by projecting onto a binary feature vector but it loses a large amount of information such as the number of matched and unmatched points as well as

the extent of the background/clutter. As a final model we propose a likelihood function which more explicitly models the generative process of occluded test images. Here, we take $\mathbf{f}^{\text{occ}}$ to include the visual word indices as in the independent model, along with the discretized position. For example, considering translation only, if we represent by X the set of all possible image positions, we let $f^{\text{occ}} \in (N \times X)^M$, where N is the dictionary of all visual words and M is the number of visual words detected per image. The ordering of vector entries is arbitrary. We begin by outlining the object and pose recognition model, before mentioning how it is adapted for object recognition.

Here we will explicitly include the transformation τ into the generative process. That is, for a given test image, τ will be a latent variable. Further, we introduce a second set of latent variables $\alpha_{o,\theta} \in \mathbb{B}^{M_{o,\theta}}$, which represent occlusion maps for each of the training object/pose images, where $M_{o,\theta}$ is the number of visual words detected in the training image for o and θ. Here $\alpha_{o,\theta}(m) = 1$ implies that m is visible in the test image, and 0 implies it is not (where m is a detected visual word). We then propose the likelihood model

$$P(\mathbf{f}^{\text{occ}}|o,\theta) = \sum_{\tau,\alpha_{o,\theta}} P(\tau)P(\alpha_{o,\theta})P(\mathbf{f}^{\text{occ}}|o,\theta,\tau,\alpha_{o,\theta}). \tag{3.11}$$

We may take a uniform distribution for $P(t)$, and simply characterize $P(\alpha_{o,\theta})$ as

$$P(\alpha_{o,\theta}) = \prod_{m=1,\ldots,M_{o,\theta}} (\alpha_{o,\theta}p_d + (1-\alpha_{o,\theta})(1-p_d)), \tag{3.12}$$

where p_d is a general probability that a word is visible (which can be set from the rate of occlusion). Given test image representation $\mathbf{f}^{\text{occ}}$, object/pose hypothesis o,θ and transformation τ, we can construct a subset of matching visual words in the test image that are potential matches of training words, $\mathcal{M}(\mathbf{f}^{\text{occ}}|o,\theta,\tau) \in \{1,\ldots,M\}$, which match in terms of visual word, and are transformed consistently according to τ. For instance, for the Hough matching procedure described earlier, $\mathcal{M}(\mathbf{f}^{\text{occ}}|o,\theta,\tau)$ contains all visual word pairs in $\mathbf{f}^{\text{occ}}$ which voted for transformation τ when matched with training image $I^{\text{train}}_{o,\theta}$. Only these visual words can be unoccluded. The remaining words must be generated by the background distribution, which we take to be uniform $p_e = 1/(N|X|)$:

$$\begin{aligned} P(\mathbf{f}^{\text{occ}}|o,\theta,\tau,\alpha_{o,\theta}) = \prod_{m=1,\ldots,M} & [\alpha_{o,\theta}(m)=1][m \in \mathcal{M}(\mathbf{f}^{\text{occ}}|o,\theta,\tau)] + \\ & [\alpha_{o,\theta}(m)=0]p_e. \end{aligned} \tag{3.13}$$

To avoid summing across all possible transformations in equation 3.11, we instead make the following approximation:

$$\begin{aligned} P(\mathbf{f}^{\text{occ}}|o,\theta) &\approx \tilde{P}(\mathbf{f}^{\text{occ}}|o,\theta) \\ &= \kappa_{o,\theta} \max_t \sum_{\alpha_{o,\theta}} P(\alpha_{o,\theta})P(\mathbf{f}^{\text{occ}}|o,\theta,\tau,\alpha_{o,\theta}), \end{aligned} \tag{3.14}$$

where $\kappa_{o,\theta}$ is a normalizing constant. This approximation implicitly assumes the likelihood is always highly peaked around the $t^\star$ achieving the maximum in equation 3.14[2]. If this is the case then $\kappa_{o,\theta} \approx 1$ for all (o, θ) and can be ignored. Also, assuming $p_d > 0.5$ and $p_d > p_e$, the maximization over t can be achieved by using the Hough transform [12]. Collecting terms, we can therefore further simplify the likelihood model to

$$P(\mathbf{f}^{\text{occ}}|o,\theta) \approx (p_d + (1-p_d)p_e)^{H_{o,\theta}(\mathbf{f}^{\text{occ}})}(1-p_d)^{M_{o,\theta}-H_{o,\theta}(\mathbf{f}^{\text{occ}})}p_e^{M-H_{o,\theta}(\mathbf{f}^{\text{occ}})}, \quad (3.15)$$

writing $H_{o,\theta}(\mathbf{f}^{\text{occ}})$ for $H_{o,\theta}(I)$ as introduced in Section 3.4.2, where $\mathbf{f}^{\text{occ}} = \mathbf{f}_I$. As in the independent features model, we can explicitly alter equation 3.15 to allow for a variable number of test features by letting $\tilde{P}(\mathbf{f}^{\text{occ}}|o,\theta) = P(M_{\mathbf{f}^{\text{occ}}}|o)\tilde{P}(\mathbf{f}^{\text{occ}}|o,\theta,M_{\mathbf{f}^{\text{occ}}})$, where $\tilde{P}(\mathbf{f}^{\text{occ}}|o,\theta,M_{\mathbf{f}^{\text{occ}}})$ is as in equation 3.15, but with $M_{\mathbf{f}^{\text{occ}}}$ substituted for M. Again, for a uniform $P(M_{\mathbf{f}^{\text{occ}}}|o)$ this does not affect the updates in Section 3.3.

Finally, we can define a likelihood model for object recognition in section 3.3 by incorporating a further maximization across θ,

$$P(\mathbf{f}^{\text{occ}}|o) \propto \max_{\theta} \tilde{P}(\mathbf{f}^{\text{occ}}|o,\theta), \quad (3.16)$$

which can be evaluated as in equation 3.15 where θ is replaced by $\theta^\star = \text{argmax}_{\theta'} H_{o,\theta'}(\mathbf{f}^{\text{occ}})$.

3.5 EXPERIMENTS

We test each of the three likelihood models described (independent features, binary and occlusion models) on our database to test the accuracy for object recognition and for object and pose recognition. Our aim is to determine if the likelihood models presented would recognize the two objects which the vocabulary tree method failed to recognize. We use a subset of objects from the database (ten objects) and included the two objects not recognized by the vocabulary tree method.

We first tested the independent feature model for object recognition, which produced a recognition accuracy of 20%. This low performance was expected given that this model does not distinguish between the foreground and the background or occlusions. For object and pose recognition only one object and no poses/viewpoints were identified using this method.

We then tested the following probability models:

- Binary model for object recognition,
- Occlusion model for object recognition,
- Binary model for object and pose recognition, and

[2]This assumption can be empirically tested. For our experimental setup we tested it by plotting $P(\mathbf{f}^{\text{occ}}|o,\theta)$ for a large number of test images and (o,θ) combinations. The distributions were sharply unimodal in approximately 90% of the cases.

- Occlusion model for object and pose recognition.

The results from these models are compared to the vocabulary tree method presented in Chapter 2.

3.5.1 PARAMETER SETTING

Our threshold for recognition μ, presented in section 3.3, is set to 0.8. We set p_a and p_b in section 3.4.1 to 1 and 2 respectively, as described in Chapter 2. The parameter p_c in section 3.4.2 is set to 0.7 so that we see at least 2 poses before reaching μ and making a decision. The parameter p_d in section 3.4.3 is set to 0.9, which corresponds roughly to the inverse of the proportion of occluded pixels in the test images, and $p_e = 1/(N|X|)$ as discussed.

3.5.2 RESULTS: OBJECT RECOGNITION

A confusion matrix was generated for each model. We show the results of the confusion matrix generated for the binary model for object recognition in Table 3.1. The output probability for each test object is placed in the respective rows with the diagonal representing the agreement between the true and estimated objects. The binary model recognizes eight out of ten objects as shown in Figure 3.1. The spice bottle and spray can 2 objects are not recognized as their accumulated belief/probability is $\leq 80\%$. These are the same two objects not recognized by the vocabulary tree model as they are significantly occluded in the test images.

Table 3.1: Confusion matrix for object recognition with the binary model.

	Obscured Cereal	**Obscured Battery**	**Obscured Curry 1**	**Obscured Elephant**	**Obscured Handbag**
Cereal	**0.9800**	0.0022	0.0022	0.0022	0.0022
Battery	0.0022	**0.9800**	0.0022	0.0022	0.0022
Curry 1	0.0021	0.0447	**0.9383**	0.0021	0.0021
Elephant	0.0022	0.0022	0.0022	**0.9800**	0.0022
Handbag	0.0022	0.0022	0.0022	0.0022	**0.9800**

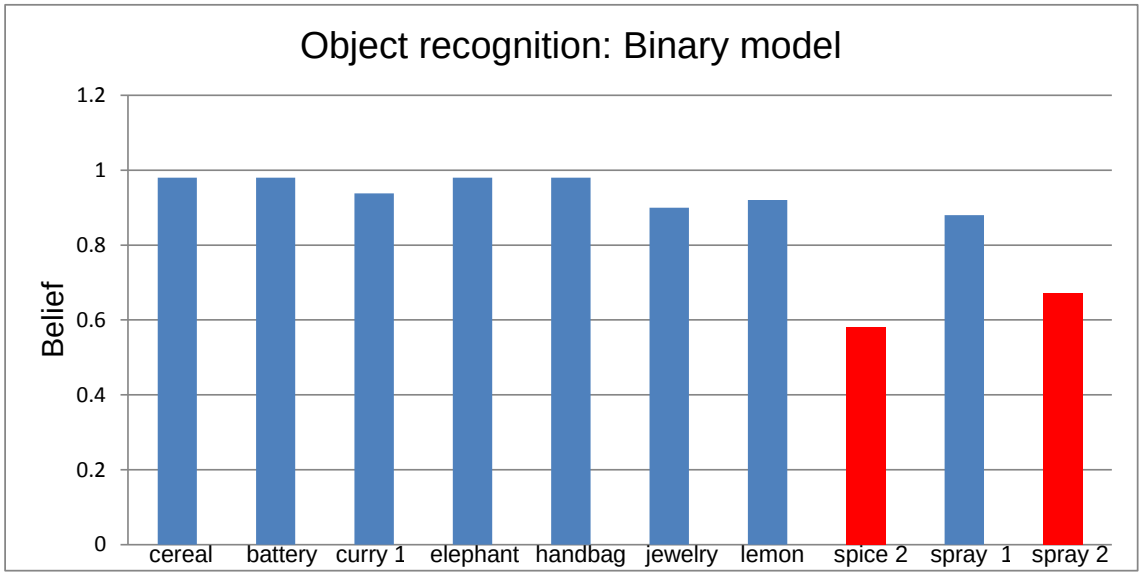


Figure 3.1: Object recognition results using the binary model.

The object recognition results for all the methods are presented in Table 3.2. The occlusion model correctly recognizes all objects in this challenging dataset (including the spice bottle and the spray can). It also requires less time to complete the recognition task for all objects.

Table 3.2: Object recognition results for all methods.

	Recognition rate	**Sum of Diagonal**	**Time(s)**
Vocabulary tree method	80%	7.56	404.5
Binary model	80%	7.88	448.7
Occlusion model	100%	9.99	240.6

Figure 3.2 displays the number of viewpoints required by the different methods, namely the vocabulary tree, binary model and occlusion model to correctly recognize an object in our dataset. For all 10 objects, the occlusion model requires fewer viewpoints to correctly recognize an object. The binary model and vocabulary tree method are comparable in the number of viewpoints required to recognize the various objects, although the vocabulary tree method is faster.

3.5.3 RESULTS: OBJECT AND POSE RECOGNITION

We then conducted experiments using the binary and occlusion models for object and pose recognition. Pose estimation accuracy refers to the system accurately predicting the correct pose of the test objects to within 20°. The results are displayed in Table 3.3. Using the object and pose recognition framework, the binary model is able to correctly identify 7 poses and the occlusion model 9 poses to within 20°. The occlusion model fails to correctly classify the pose of the spice 2 object as shown in

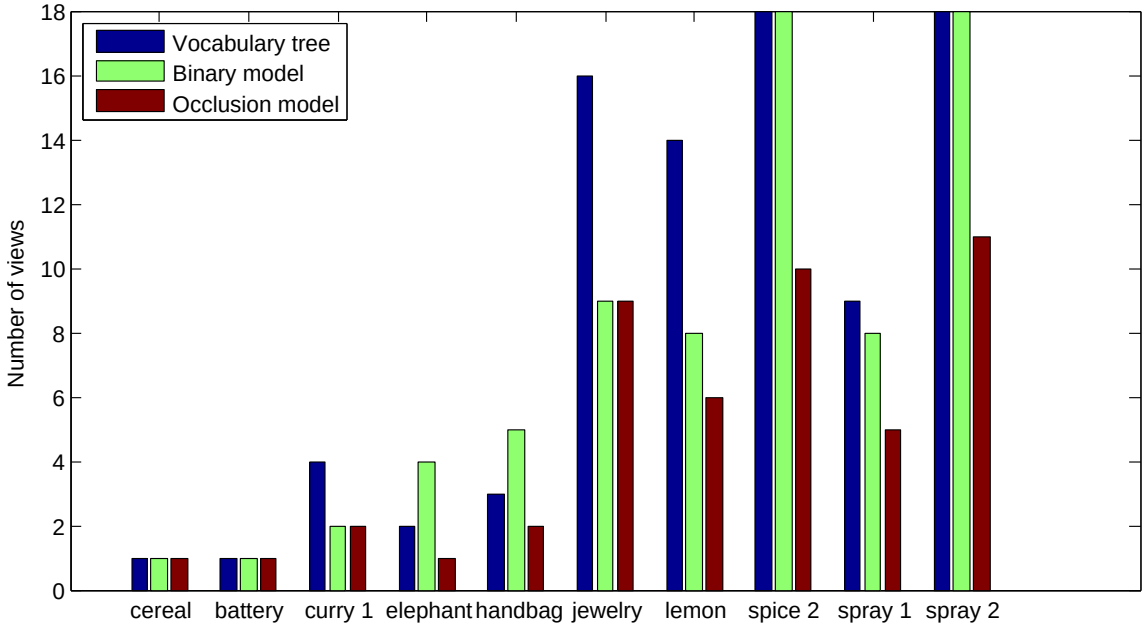


Figure 3.2: The number of viewpoints required by the vocabulary tree method, binary model and occlusion model to recognize the objects in the dataset.

Figure 3.3.

Table 3.3: Object and pose recognition results.

	Recognition rate	**Time(s)**	**Pose**
Binary model	80%	1673	70%
Occlusion model	100%	238.7	90%

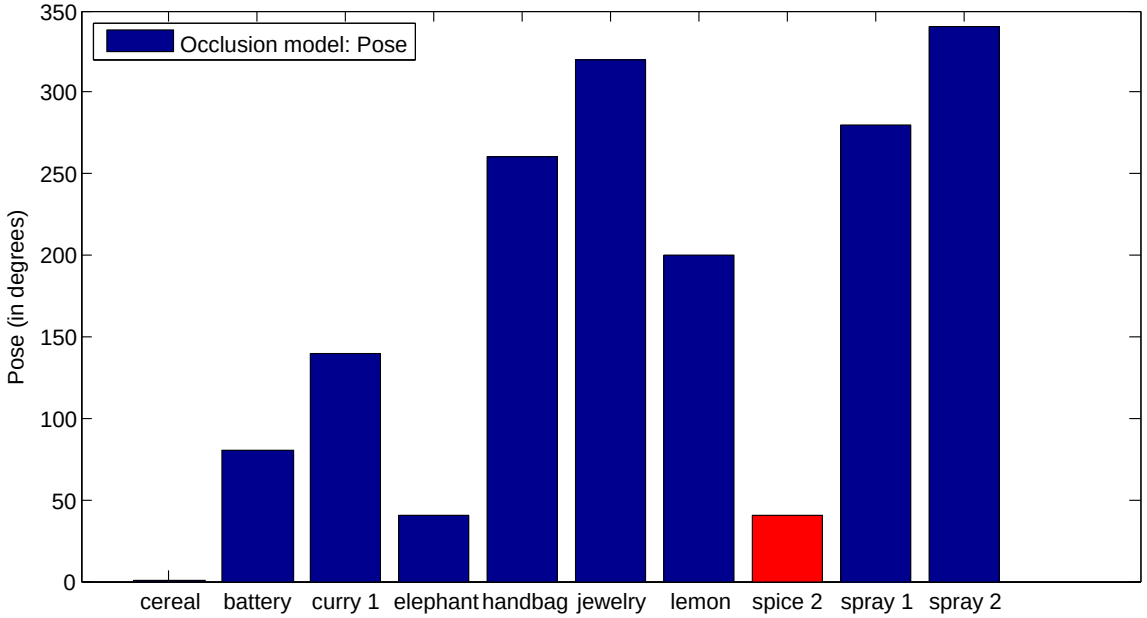


Figure 3.3: Pose identification using the occlusion model.

3.6 CONCLUSIONS

We presented Bayesian approaches for active object recognition and active object and pose recognition. The Bayesian framework allows information across several poses to be integrated and provides a quantitative value as to the system's confidence in the object's identity and pose. Our test set consisted of ten objects appearing in cluttered environments with occlusion. The probabilistic object and pose models were explicitly designed to cope with such a difficult environment. The independent feature likelihood model did not perform well on our dataset, recognizing only two objects. This could be attributed to the fact that it did not take into account any geometric structure, foreground, background or possible occlusions. The binary model took into account geometric structure using the Hough transform and recognized eight out the ten objects in the dataset. It failed to recognize the same two objects as the vocabulary tree method. The occlusion model took into account the geometric structure as well as incorporating object and background distributions and occlusions. The occlusion model produces the best results. It correctly recognizes all objects in the database and is also the fastest method. Given that the occlusion model recognizes all the objects, we can conclude that it is important to take into account the background features, as well as to explicitly model the geometric transformation and occlusion of features.

CHAPTER FOUR

MULTIPLE OBJECTS RECOGNITION[1]

4.1 INTRODUCTION

In previous chapters we extracted SIFT features from both training and testing images, and used SIFT matching and the Hough transform for object recognition of a single object in the test images. The Hough transform imposed geometric constraints on the initial SIFT matches allowing ambiguous and spurious matches to be removed. These matches were then tested in various probability models to determine their ability to correctly recognize a single object in cluttered environments. In chapter 3 the final probability model, which took into consideration geometric matching as well as modeling the occlusion in an image, was able to correctly recognize all objects presented. In these experiments the aim was to recognize a single object from the test image containing multiple objects. In this chapter we aim to recognize multiple objects contained in the database that appear in the test image. We continue to use SIFT features, SIFT matching and the Hough transform. We introduce a probability model for single and multiple object recognition.

We show how representing a test image by a feature vector containing the best matching counts from all training views contains sufficient information to build effective probabilistic models for active recognition in the multiple object scenario. We are able to outperform existing active recognition methods that are similarly based on SIFT features, but which do not incorporate geometric matching [1].

[1]Related publication:

- Natasha Govender, Jonathan Warrell, Mogomotsi Keaikitse, Philip Torr, Fred Nicolls, "Probabilistic Active Recognition of Multiple Objects using Hough-based Geometric Matching Features", Yu Sun, Aman Behal and Chi-Kit Ronald Chung (eds), in Introduction to Robot Vision, Springer, pp. 89-108.

We adopt a Bayesian framework for data fusion and explore a number of probabilistic models based on the Hough-matching feature representation specifically designed to represent hypotheses about the presence of multiple objects, as well as the poses of all objects that are present. This allows us to cope effectively with more complex test data.

In the case of matching multiple objects, we show how effective probabilistic models can be built by using empirical distributions of matching counts for each object/viewpoint. We demonstrate empirically the gains that can be achieved through using such multiple object models over simpler single-object models (both ours and previous methods) in our test scenario. A challenge for multiple object settings is determining a means of combining data across viewpoints, while maintaining information about uncertainty for multiple objects. We show that the multiple object model introduced performs well on our complex dataset.

Finally, we provide an extensive evaluation of a viewpoint selection mechanism introduced in chapter 2 for the case of active recognition of single objects, extended here to the multiple object case. This algorithm uses a vocabulary tree data structure [25] to cluster all SIFT vectors from our training set and builds a uniqueness map for each object, which summarizes the uniqueness of each object viewpoint by summing a quasi term frequency-inverse document frequency (TFIDF) metric across the counts of the leaf-node clusters appearing at that viewpoint. The next viewpoint is selected using the TFIDF metric, based on the current belief about which object/objects are present. This approach is particularly efficient compared to mutual information, as commonly used in Bayesian models [44,71–73], since it does not require the averaging of entropy scores for every possible outcome. Further, the vocabulary tree viewpoint selection strategy can also be used in non-Bayesian active contexts, such as the model of [1], where it is more efficient to evaluate than the expected activation. We compare this selection mechanism with previous methods in both Bayesian and non-Bayesian contexts, showing it to perform well in terms of efficiency and accuracy in the multiple object setting compared to mutual information and expected activation.

In summary, we develop an active recognition pipeline specifically to handle the realistic situation of simultaneously recognizing multiple objects in close proximity, which may be subject to extensive occlusions and clutter from distractor objects. Within our approach we highlight the main contributions:

- We develop a Bayesian model for data fusion that maintains a distribution over multiple object and viewpoint hypotheses. This enables us to recognize multiple objects from the dataset that is present in a test image.
- We extend the viewpoint selection mechanism presented in chapter 2 to multiple objects and provide an extensive evaluation, comparing it to alternative mechanisms.

Section 4.2 discusses the related work with sections 4.3 and 4.4 defining our approaches to single and multiple object and pose recognition, respectively. The mutual information algorithm and its relation-

ship to the vocabulary tree data structure are described in section 4.5. The experiments conducted, including comparisons of the single and multiple object recognition algorithms, viewpoint selection algorithms and the accuracy of the multiple object recognition approach are detailed in section 4.6 along with the corresponding results. The conclusions are presented in section 4.7.

4.2 RELATED WORK

A number of methods have considered feature representations similar to ours outside of the active vision setting. Our Hough-based matching feature representation is inspired by [12], who adopt a similar matching process for locating 3D object views in single images. However, [12] uses matching counts along with a variety of other features to build a probabilistic model to accept or reject the presence of an object in a image, while we construct a variety of distributions for active recognition scenarios using the matching counts alone as features. Our simpler representation however proves sufficient in the active setting for multiple object recognition.

Our framework follows that of [34, 50, 71] in terms of the general Bayesian form of our updates. Systems presented by [18, 62, 71] consider only recognizing single objects in uncluttered environments. Further, these methods consider only the case of a single object/viewpoint hypothesis, and do not consider the updates that are required in the multiple object/viewpoint hypothesis case we consider. In addition, [71] proposes the mutual information criterion as a viewpoint selection mechanism, which has subsequently been used by [44, 72, 73] in an active setting. Mutual information is expensive to calculate and requires the collection of extensive statistics at training time although, as [71] discusses, it provides the optimal strategy provided the underlying models are correct.

We show empirically that our proposed viewpoint selection method outperforms this strategy, thus showing the utility of our mechanism in a non-Bayesian context.

4.3 ACTIVE RECOGNITION OF A SINGLE OBJECT

In chapter 3 we detailed a Bayesian approach to object and viewpoint recognition, which is used in the experiments in this chapter. Here we describe how this Bayesian approach is extended to multiple object and viewpoint recognition. Separate likelihood models are presented for single and multiple object and viewpoint recognition. Although we give specific forms for the image representation, likelihood model and viewpoint selection rules, the framework is general and different choices can be substituted for these.

Problem statement: The experimental setup for the active recognition task for a single object is as defined in chapter 3, where at training time for each object $o = 1, ..., O$ we capture a set of images,

one at each of a series of P regularly spaced training views around the object, indexed by their viewing angle. For example, $\theta \in \{0^\circ, 20^\circ, 40^\circ, ..., 340^\circ\} = \Theta$ and $P = |\Theta|$.

At test time we are presented again with one of the training objects, and must identify the object present and its orientation. We are allowed to capture images of the test object at a sequence of test views $\delta_1, \delta_2, ... \in \{0^\circ, 20^\circ, 40^\circ, ..., 340^\circ\}$, where the angles δ_t can be in any order. Typically an active object recognition algorithm will include the following components: an *update strategy* for incorporating new information from each viewpoint as it is seen and using it to update a belief/score for the correct $o^\star$ and $\theta^\star$; a *viewpoint selection strategy* for choosing the sequence of test views $\delta_1, \delta_2, ...$; and a *stopping criterion* to decide when to stop capturing further views and generate the output. We outline a Bayesian algorithm below for single object active recognition which incorporates these elements. These are listed in turn, following a description of our image representation. This information feeds into the likelihood model that is used to update the Bayesian framework.

Image representation: For a given test image, I_δ^{test}, we apply the method of [13] to generate a sparse set of SIFT descriptors to represent the image. We index these by $\mathcal{J}_\delta^{\text{test}} = \{1, ..., N_\delta^{\text{test}}\}$, where N_δ^{test} is the number of descriptors found for test image I_δ^{test}. Each descriptor index is associated with a 128-dimensional SIFT descriptor, a location, scale and orientation. We can form a sparse representation for a given training image, $I_{o,\theta}^{\text{train}}$, by introducing the functions $d_{o,\theta}^{\text{train}}$, $x_{o,\theta}^{\text{train}}$, $y_{o,\theta}^{\text{train}}$, $s_{o,\theta}^{\text{train}}$ and $\phi_{o,\theta}^{\text{train}}$ over index set $\mathcal{J}_{o,\theta}^{\text{train}}$.

We now consider the set of *matched pairs* of descriptors between training image $I_{o,\theta}^{\text{train}}$ and test image I_δ^{test}, which we write $\mathcal{M}_\delta^{o,\theta}$. This consists of all pairs of descriptors whose distance falls below a certain threshold, μ^{match}:

$$\mathcal{M}_\delta^{o,\theta} = \{(n_1, n_2) \in \mathcal{J}_{o,\theta}^{\text{train}} \times \mathcal{J}_\delta^{\text{test}} || d_{o,\theta}^{\text{train}}(n_1) - d_\delta^{\text{test}}(n_2)|_2 < \mu^{\text{match}}\}. \quad (4.1)$$

These matches are then used as input to a Hough transform voting procedure to assist in removing any randomly occurring SIFT matches. This is achieved by allowing each match to vote for an approximate translation, scaling and rotation of the object. Given $\mathcal{B}_1$ (x-translation), $\mathcal{B}_2$ (y-translation), $\mathcal{B}_3$ (rotation) and $\mathcal{B}_4$ (scale) for the sets of bins used in the Hough transform, for each $m \in \mathcal{M}_\delta^{o,\theta}$ we generate votes $v_1(m) : \mathcal{M}_\delta^{o,\theta} \rightarrow \mathcal{B}_1$,..., $v_4(m) : \mathcal{M}_\delta^{o,\theta} \rightarrow \mathcal{B}_4$ as follows. For the scale and rotation votes, we simply quantize the scale/orientation ratios/differences of the matched pair of descriptors to the nearest bin: $v_3(m) = v_3(n_1, n_2) = \text{round}_{\mathcal{B}_3}(\phi_\delta^{\text{test}}(n_2) - \phi_{o,\theta}^{\text{train}}(n_1))$ and $v_4(m) = v_4(n_1, n_2) = \text{round}_{\mathcal{B}_4}(s_\delta^{\text{test}}(n_2)/s_{o,\theta}^{\text{train}}(n_1))$ (writing $\text{round}_{\mathcal{B}_3}$ and $\text{round}_{\mathcal{B}_4}$ for functions which return the corresponding bin for a given scale/orientation ratio/difference). To generate the translation votes we solve for the similarity transform that will map $(x_{o,\theta}^{\text{train}}(n_1), y_{o,\theta}^{\text{train}}(n_1))$ to $(x_\delta^{\text{test}}(n_2), y_\delta^{\text{test}}(n_2))$ using the known scaling and rotation above with an unknown translation (tx_m, ty_m) (details of this calculation are given in chapter 2). We then set $v_1(m) = \text{round}_{\mathcal{B}_1}(tx_m)$ and $v_2(m) = \text{round}_{\mathcal{B}_2}(ty_m)$, with $\text{round}_{\mathcal{B}_1}$ and $\text{round}_{\mathcal{B}_2}$ defined similarly to above.

Having generated the required votes we find:

$$b_i^* = \text{argmax}_{b\in\mathcal{B}_i}\sum_m [v_i(m)=b], \tag{4.2}$$

for $i = 1, ..., 4$ and argmax returns the bins which separately accumulated the most votes. We define the Hough-matching score for $I_{o,\theta}^{\text{train}}$ and I_δ^{test} to be the number of matched descriptors voting simultaneously for these bins:

$$H_\delta^{o,\theta} = |\{m \in \mathcal{M}_{\delta_t}^{o,\theta}| \bigwedge_{n=1,...,4} v_n(m) = b_n^*\}. \tag{4.3}$$

We then form a feature vector for a given test image $\mathbf{f}_\delta^{\text{test}}$ by concatenating these scores across all training images:

$$\mathbf{f}_\delta^{\text{test}} = [[H_\delta^{o_1,\theta_1}; H_\delta^{o_1,\theta_2}; ...]; [H_\delta^{o_2,\theta_1}; H_\delta^{o_2,\theta_2}; ...]; ...], \tag{4.4}$$

where $[.;.]$ denotes vertical concatenation.

Update strategy: We outline here a Bayesian update strategy which maintains a distribution over object and viewpoint random variables, O and α_0, given the images observed up to a given time step t. This can be expressed as $P_t(o, \theta_0) = P(o, \theta_0|\mathbf{f}_{\delta_1}^{\text{test}}, ...\mathbf{f}_{\delta_t}^{\text{test}})$, where $\mathbf{f}_{\delta_t}^{\text{test}}$ is as above, and we denote by $P_t(o, \theta_0)$ the probability at time step t that the test object is o and the test view at the reference test viewpoint $\delta_1 = 0^\circ$ corresponds to training view $\theta_0 \in \Theta$. For simplicity we assume that the images we see at different viewpoints are generated independently given o and θ_0. In general this will not be the case, since we expect there to be high correlations between the images we see for instance at neighboring viewpoints. However, making this assumption allows us to build a separate probability model for each object/viewpoint combination $P(\mathbf{f}_\delta^{\text{test}}|o, \alpha_\delta = \theta_\delta)$, where the random variable α_δ corresponds to the training view seen at a particular δ, which stands in the deterministic relation to α_0, $\alpha_\delta = \alpha_0 + \delta$ modulo 360°. We can recursively estimate $P_t(o, \theta_0)$ as

$$P_t(o, \theta_0) = \frac{P(\mathbf{f}_{\delta_t}^{\text{test}}|o, \theta_0 + \delta_t)P_{t-1}(o, \theta_0)}{\sum_{o,\theta_0} P(\mathbf{f}_{\delta_t}^{\text{test}}|o, \theta_0 + \delta_t)P_{t-1}(o, \theta_0)}. \tag{4.5}$$

By default, a uniform prior can be selected for $P_0(o, \theta_0)$. If we are primarily interested in identifying the correct test object, we can further calculate

$$P_t(o) = \sum_{\theta_0} P_t(o, \theta_0). \tag{4.6}$$

It remains to specify fully the likelihood model. Our model depends on two parameters, $p_a,\ p_b \in \mathbb{R}$ ($p_a > p_b$):

$$P(\mathbf{f}_\delta^{\text{test}}|o, \theta_\delta) \propto \begin{cases} 0 & \text{if } \max_{o',\theta'}(f_\delta^{\text{test}}(o', \theta')) \geq M \\ p_a & \text{if } \max_{o',\theta'}(f_\delta^{\text{test}}(o', \theta')) < M \text{ and} \\ & (o, \theta_\delta) \in \text{argmax}_{o',\theta'}(f_\delta^{\text{test}}(o', \theta')) \\ p_b & \text{otherwise,} \end{cases} \tag{4.7}$$

where argmax returns the subset of arguments attaining the maximum value. The parameter M may be set arbitrarily high, and simply allows the model to be normalized (specifying a number of matches that cannot be exceeded). By specifying $p_a > p_b$ we ensure that when we are looking at object o and viewpoint θ, we expect to see feature vectors generated where $f_\delta^{\mathrm{test}}(o, \theta)$ takes the maximum value in the vector.

Viewpoint selection strategy:

The viewpoint selection strategy presented in chapter 2, which uses the vocabulary tree data structure, is used in these experiments.

4.4 ACTIVE RECOGNITION OF MULTIPLE OBJECTS

The algorithm outlined in Section 4.3 assumes that we are viewing a single object at test time from a variety of angles. However, in natural scenes we rarely encounter single objects isolated from each other, and more typically see collections of objects which occlude each other and may contain cluttering objects that we are not trained to recognize. The Bayesian framework presented above is readily adapted to recognize collections of objects (and their orientations) in place of single objects.

Problem statement:

We assume that we have access to the same training data as described in chapter 2. As before, we write $I_{o,\theta}^{\mathrm{train}}$ for the training image of object o at viewing angle θ. Instead of being presented with a single object at test time we now assume we are viewing a collection of objects, which may include objects from our training set as well as unknown objects. Our task is to identify:

- which out of the set of known objects are present, and
- for every object present, its orientation with respect to a reference viewpoint.

This output may be expressed by an $N_O \times N_\theta$ matrix, $S*$, with entries in $\mathbb{S}^+$, whose entry $S^*(o, \theta)$ denotes the number of occurrences of object o at orientation θ in the test collection. As in the single object case, we are allowed to capture a sequence of images of the test collection at viewing angles $\delta_1, \delta_2, ..., \in \{0°, 20°, 40°, ..., 340°\}$ (with respect to rotation about the center of the collection) and we treat $\delta_1 = 0°$ as the reference viewpoint to label the orientation of the objects present.

We make two simplifying assumptions in the model presented (which are respected in our experimental data). First, that the camera positions when viewing the test collection are such that all object centers project close to the center of the image (i.e. the collection is not too dispersed), and thus that we do not need to compensate for projection effects when identifying the orientations of objects at different positions in the collection (implying that we have approximately an orthogonal projection

over the collection). Second, we assume the same object does not occur more than once at the same orientation, and thus that the matrix S to be estimated is binary[2].

Image representation: We use the same image representation as in the single object recognition case. Hence, given a new test image, I_δ^{test}, we calculate the matching descriptor sets $\mathcal{M}_\delta^{o,\theta}$ for each (o, θ) pair, and the corresponding Hough matching scores $H_\delta^{o,\theta}$ (equation 4.3). The feature representation $\mathbf{f}_\delta^{\text{test}}$ is again formed by concatenating the Hough scores as in equation 4.4.

Update strategy: Since we are interested in estimating a collection of objects and associated poses under the assumption that no object appears multiple times at the same viewpoint, we introduce a binary random variable $B(o, \theta)$ for each (o, θ) pair, which will take the value 1 if object o is present in the test collection at orientation θ, and 0 otherwise. Making the simplifying assumption that appearances of object/orientation pairs in the test collection are independent, we maintain a separate distribution for each of these binary variables. Our belief that object o is present at orientation θ in the test collection, given the images observed up to time step t, can be expressed as

$$P_t(B(o,\theta)) = P_t(B(o,\theta)|\mathbf{f}_{\delta_1}^{\text{test}}, ..., \mathbf{f}_{\delta_t}^{\text{test}}), \tag{4.8}$$

where $B(o, \theta) \in \{0, 1\}$ is the value taken by the matrix $S(o, \theta)$. We update in parallel each of these distributions using a Bayesian update strategy. As in section 4.3, we assume that images at different test viewpoints are generated independently given the test collection. We thus require a likelihood model for the generation of a feature vector given a collection of objects at specific offsets. We can express this as $P(\mathbf{f}_\delta^{\text{test}}|\{S_\delta(o, \theta_\delta) = B\delta(o, \theta_\delta), o = 1, ..., N_O, \theta \in N_\theta\})$, where we have the deterministic relation $S_\delta(o, \theta_\delta) = S_0(o, \theta_0 + \delta)$. We assume that this likelihood factorizes as follows:

$$P(\mathbf{f}_\delta^{\text{test}}|\{B_\delta(o,\theta_\delta), o = 1, ..., N_O, \theta \in N_\theta\}) = \prod_{o,\theta} P(f_\delta^{\text{test}}(o,\theta_\delta)|B_\delta(o,\theta_\delta)). \tag{4.9}$$

This allows us to express the required Bayesian updates as

$$P_t(B_0(o,\theta_0)) = \frac{P(f_{\delta_t}^{\text{test}}(o,\theta_{\delta_t})|B_{\delta_t}(o,\theta_{\delta_t}))P_{t-1}(B_0(o,\theta_0))}{\sum_{B_0(o,\theta_0)=\{0\ 1\}} P(f_{\delta_t}^{\text{test}}(o,\theta_{\delta_t})|B_{\delta_t}(o,\theta_{\delta_t}))P_{t-1}(B_0(o,\theta_0)).} \tag{4.10}$$

To estimate the probability that a particular object is present at any orientation, we can evaluate

$$P_t(B(o)) = 1 - \prod_{\theta_0} P_t(B_0(o,\theta_0) = 0). \tag{4.11}$$

By equation 4.9, we can express our likelihood model directly in terms of $P(f_\delta^{\text{test}}(o, \theta_\delta)|B_\delta(o, \theta_\delta))$. We also introduce a smoothing constant β, and a constant $M' < M$ (which is used to pool together

[2]Our experimental data in fact allows the stronger assumption that the same object does not occur more than once. This can easily be incorporated, although for simplicity we outline the model here without this assumption, which we did not find to provide significant gains in practice.

less frequently occurring larger values of n). Our general likelihood model can then be expressed as

$$P(f_\delta^{\text{test}}(o,\theta_\delta)|z_\delta(o,\theta_\delta)) \propto \begin{cases} 0 & \text{if } f_\delta^{\text{test}}(o,\theta_\delta) \geq M \\ \lambda_{o,\theta_\delta,f_\delta^{\text{test}}(o,\theta_\delta)}^{z_\delta(o,\theta_\delta)} + \beta & \text{if } f_\delta^{\text{test}}(o,\theta_\delta) < M' \\ \frac{(\sum_{n=M'\ldots M} \lambda_{o,\theta_\delta,n}^{z_\delta(o,\theta_\delta)}) + \beta}{M-M'} & \text{otherwise.} \end{cases} \tag{4.12}$$

For this purpose, we assume we have access to counts (from a sample of validation images with statistics similar to our test data) $\kappa_{o,n}^0$, $\kappa_{o,n}^1$, $\kappa_{o,\theta,n}^0$, $\kappa_{o,\theta,n}^1$, $n = 0, ..., M$, where M is a maximum value used for normalization as in Section 4.3:

- $\kappa_{o,\theta,n}^1$ indicates how many times we observe a Hough score of n between a validation image containing viewpoint θ of object o and training image $\mathcal{I}_{o,\theta}^{\text{train}}$.
- $\kappa_{o,\theta,n}^0$ indicates how many times we observe a Hough score of n between such a validation image and all other training images.
- $\kappa_{o,n}^1$ indicates how many times we observe a Hough score of n between a validation image containing o and any training image containing o regardless of orientation.
- $\kappa_{o,n}^1$ indicates how many times we observe a Hough score of n between a validation image containing o and any training image containing o regardless of orientation.

We consider two cases. For the first (which we call *likelihood model 1*) we let $\lambda_{o,\theta,n}^b = \kappa_{o,n}^b$, where $b \in \{0, 1\}$. This allows a different distribution of Hough matching scores for each object, but does not distinguish between different viewpoints. For the second (*likelihood model 2*), we let $\lambda_{o,\theta,n}^b = \kappa_{o,\theta,n}^b$, hence giving a different distribution of Hough matching scores for each viewpoint and orientation.

Viewpoint selection strategy: The same viewpoint selection strategy may be applied as in Section 4.3, since it depends only on the feature vector representation and not on the probability model. Implicitly this selection strategy respects an assumption that the same object is not present in the collection at multiple orientations. This is appropriate in our experimental setting, but may not be appropriate in general.

Stopping criterion: We adapt the stopping criterion of section 4.3 to cope with the multiple object hypothesis. Here we fix a value N_{obj}, which represents the number of objects in the test image the system should search for and stop when at least N_{obj} objects reach a belief of μ^{stop} of being present: for the first t at which $\sum_o [P_t(B(o)) > \mu^{\text{stop}}] \geq N_{\text{obj}}$ we halt and output S^*, where $S^*(o, \theta_0) = 1$ if $P_t(S_0(o, \theta_0) = 1) > 0.5$ and $S^*(o, \theta_0) = 0$ otherwise. This stopping criterion implicitly assumes at least N_{obj} objects will be present in a collection, which is appropriate in our experimental setting, but may not be true in general.

4.5 MUTUAL INFORMATION

Mutual information (MI) of two random variables, which is also known as transinformation, is a measure of the variables' mutual dependence. Mutual information measures the reduction in uncertainty about one random variable given knowledge of another. High mutual information indicates a large reduction in uncertainty; low mutual information indicates a small reduction; and zero mutual information between two random variables means the variables are independent.

For two discrete variable X and Y whose joint probability distribution is $P_{XY}(x, y)$, the mutual information between them is denoted by $I(X;Y)$ [74]:

$$I(X;Y) = \sum_{x,y} P_{XY}(x,y) \log \frac{P_{XY}(x,y)}{P_X(x)P_Y(y)} = E_{P_{XY}}\{\log \frac{P_{XY}}{P_X P_Y}\}. \tag{4.13}$$

Here $P_X(x)$ and $P_Y(y)$ are the marginals

$$P_X(x) = \sum_y P_{XY}(x,y), \tag{4.14}$$

and

$$P_y(y) = \sum_x P_{XY}(x,y), \tag{4.15}$$

and E_p is the expected value over the distribution P. To understand what $I(X;Y)$ means, we need to explain the concepts of entropy and conditional entropy.

Entropy is a measure of uncertainty; the higher the entropy the more uncertain one is about a random variable. Entropy can be defined as

$$H(X) = -\sum_x P_X(x) \log P_X(x) = E_{P_X}\{\log P_X\}. \tag{4.16}$$

The conditional entropy is the average uncertainty about X after observing a second random variable Y, and is given by

$$H(X|Y) = \sum_y P_Y(y)[-\sum_x P_{X|Y}(x|y) \log(P_{X|Y}(x|Y))] = E_{P_y}\{-E_{P_{X|Y}} \log P_{X|Y}\}, \tag{4.17}$$

where $P_{X|Y}(x,y) = \frac{P_{XY}(x,y)}{P_Y(y)}$ is the conditional probability of x given y.

With the definitions of $H(X)$ and $H(X|Y)$, we can rewrite the equation for mutual information as

$$I(X,Y) = H(X) - H(X|Y). \tag{4.18}$$

Mutual information is therefore the reduction in uncertainty about the variable X after observing Y.

4.5.1 RELATIONSHIP TO VOCABULARY TREE DATA STRUCTURE

We briefly discuss here the relationship between our viewpoint selection algorithm using the vocabulary tree data structure, introduced in section 4.3, and an approach to viewpoint selection based on mutual information. As we note, our approach is typically more efficient in terms of complexity.

The use of mutual information for data selection in active sensing tasks has been proposed by [71], and again more recently in [44, 72]. In terms of our multiple object problem, this approach would direct us to select the next test viewpoint on the basis of which has the highest mutual information with the random variables $Z_0(o, \theta_0)$ we are interested in. Hence, we search for

$$\delta_{t+1} = \text{argmax}_{\delta' \in \times \setminus \{\delta_1 \ldots \delta_t\}} I(\mathbf{f}_{\delta'}^{\text{test}}; \mathbf{z}_0) \tag{4.19}$$

where the mutual information is defined as

$$I(\mathbf{f}_{\delta'}^{\text{test}}; \mathbf{z}_0) = H(\mathbf{z}_0) - H(\mathbf{z}_0 | \mathbf{f}_{\delta'}^{\text{test}}) \tag{4.20}$$

with $H(.)$ the Shannon entropy, and $H(.|.)$ the conditional entropy. Given that $H(\mathbf{z}_0)$ is independent of δ', maximizing equation 4.19 is equivalent to minimizing the conditional entropy

$$\delta_{t+1} = \underset{\delta' \in \times \setminus \{\delta_1, \ldots, \delta_t\}}{\text{argmin}} H(\mathbf{z}_0 | \mathbf{f}_{\delta'}^{\text{test}}.) \tag{4.21}$$

Given the factorization of the likelihood function in equations 4.9 and 4.12, we can evaluate the conditional entropy as

$$\begin{aligned} H(\mathbf{z}_0 | \mathbf{f}_{\delta}^{\text{test}}) &= -\sum_{\mathbf{f}_{\delta}^{\text{test}}, \mathbf{z}_0} P(\mathbf{z}_0) P(\mathbf{f}_{\delta}^{\text{test}} | \mathbf{z}_{\delta}) \log(P(\mathbf{z}_{\delta} | \mathbf{f}_{\delta}^{\text{test}})) \\ &= -\sum_{o, \theta_0} \sum_{f_{\delta}^{\text{test}}(o, \theta_{\delta}), z_0(o, \theta_0)} P(z_0(o, \theta_0)) P(f_{\delta}^{\text{test}}(o, \theta_{\delta}) | z_{\delta}(o, \theta_{\delta})) \\ &\qquad \log(P(z_{\delta}(o, \theta_{\delta}) | f_{\delta}^{\text{test}}(o, \theta_{\delta}))) \end{aligned} \tag{4.22}$$

For equation 4.12 with likelihood model 1 ($\lambda_{o,\theta,n}^{b} = \kappa_{o,n}^{b}$), we have that $P(f_{\delta'}^{\text{test}}(o, \theta_{\delta'}) | z_{\delta'}(o, \theta_{\delta'})) = P(f_{\delta''}^{\text{test}}(o, \theta_{\delta''}) | z_{\delta''}(o, \theta_{\delta''}))$ when $f_{\delta'}^{\text{test}}(o, \theta_{\delta'}) = f_{\delta''}^{\text{test}}(o, \theta_{\delta''})$ and $z_{\delta'}(o, \theta_{\delta'}) = z_{\delta''}(o, \theta_{\delta''})$, and thus MI cannot be used with this model as all viewpoints will give rise to the same conditional entropy. A similar problem affects the single object model in section 4.3, since equation 4.7 is symmetric across o and θ. However, since equation 4.12 with likelihood model 2 ($\lambda_{o,\theta,n}^{b} = \kappa_{o,\theta,n}^{b}$) results in distinct matching score distributions for each object/orientation combination, the MI selection strategy above can be applied with this model.

The fact that we can only apply a MI selection strategy when we have distinct distributions for each viewpoint highlights the reliance of the MI strategy on extensive training/validation statistics. Our vocabulary tree method can be used in cases where we do not estimate these. Further, the evaluation of the required conditional entropies in equation 4.21 is $O(N_O N_\theta K)$, where K is the complexity of

evaluating the expectation across the feature space (in general $K = |\mathcal{F}|$, where $\mathbf{f}_\delta^{\text{test}} \in \mathcal{F}$, and for our likelihood model 2 we have $K = M'$ due to the form of equation 4.12). In contrast, our vocabulary tree method requires only $O(N_O N_\theta)$ using the selection rule described. We note though that, if accurate probability models are available for each viewpoint, the MI selection rule is optimal in the sense of achieving the lowest expected misclassification loss for a given number of viewpoints [71].

4.6 EXPERIMENTATION

In this chapter we have presented approaches to both single and multiple object and pose recognition. We first test the performance of the multiple object and pose recognition algorithm on a difficult dataset. Comparisons are then conducted between the single and multiple object models as well as the viewpoint selection strategy which was extended to the multiple object scenario. Our viewpoint selection strategy is compared with random and mutual information for viewpoint selection in a multiple object scenario. Further experiments are performed using the active object recognition method by Kootsta et al. [1], initially in its original format and then substituting our viewpoint selection strategy to determine if there are any changes to the accuracy.

Dataset

For our experiments we use the active recognition training dataset introduced in Chapter 2. For the test set, a 'primary' object was placed in the centre of the turntable with 'secondary' objects, which or may not belong to the training set, surrounding it. The distractor objects include everyday objects such as a towel, pencil box, thyme bottle and bracelet. Example images from the test set are shown in Figures 4.1 and 4.2. These images are used in the experiments for testing either the performance against a single object hypothesis, where we desire recognition of the central object, or multiple objects hypotheses, where we desire recognition of all objects appearing in the training set. For both training and test data, images are captured around the y-axis, which represents 1 degree of freedom (DoF).

Experiment 1: Comparing single and multiple object recognition algorithms

Our first experiment compares the performance of the two algorithms outlined, using the single and multiple object hypotheses. We consider two tasks: recognizing the primary objects in the test sequences, and recognizing all objects in the test sequences that are present in the database. For the primary object task we take the object with highest probability for both single and multiple object hypotheses ($\text{argmax}_o P_t(o)$ and $\text{argmax}_o P_t(z(o))$ respectively).

For the all object task, we generate precision-recall curves by thresholding both the single object and multiple object posteriors ($P_t(o)$ and $P_t(z(o))$). We note that these operations are valid probabilistically only for the primary-task/single-object and all-object-task/multiple object combinations respectively. For the multiple object algorithm we use likelihood 1 model. We select 3 evenly-spaced

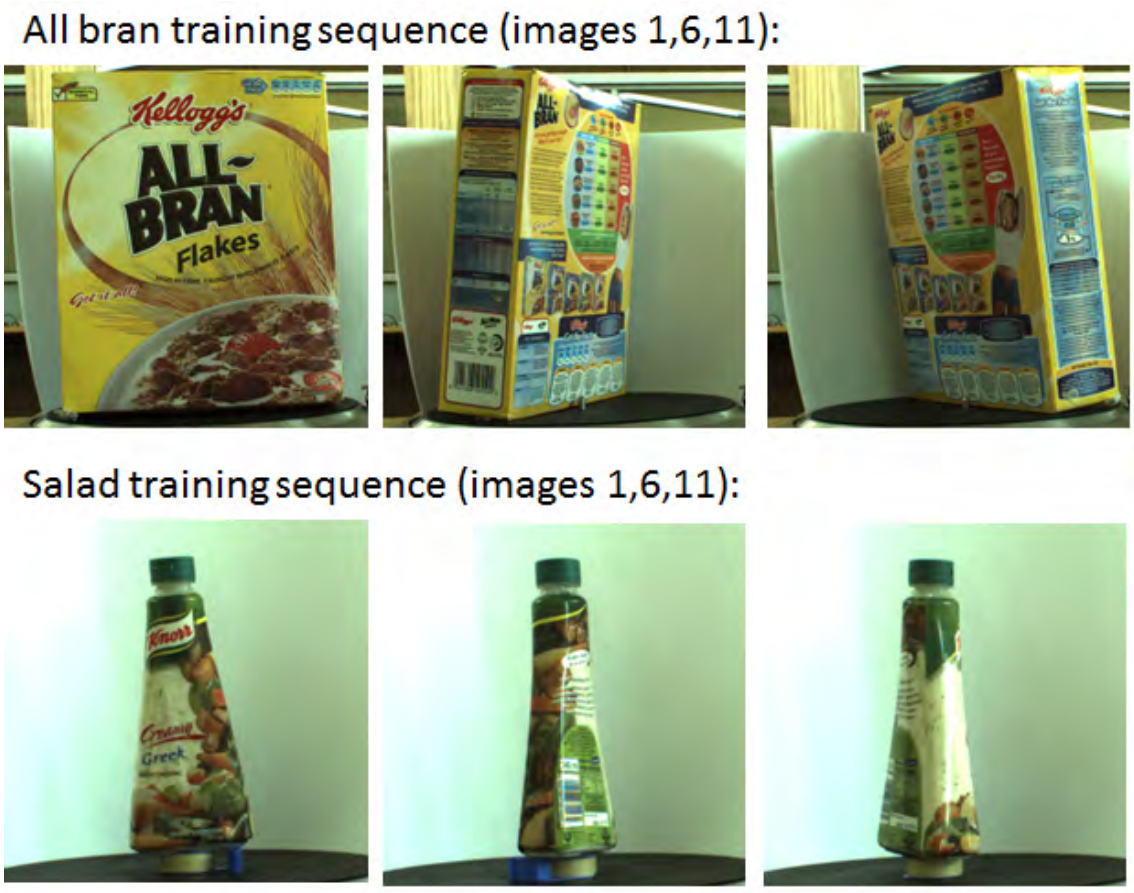


Figure 4.1: Example images from our training set.

viewpoints (5, 11 and 17) from all test sequences to form the validation set from which to record the counts $\kappa_{o,n}^{\{0,1\}}$ used in this model. We restrict all models to the 15 remaining viewpoints at test time for a fair comparison. We record the performance of the algorithms on both tasks after $1, ..., 15$ viewpoints, and average across all 15 starting viewpoints for all performance measures to generate a robust comparison. Results are shown in Figures 4.3 and 4.4.

The results show that the two models have different performance characteristics when we consider recognition of all objects present. The single object hypothesis approach generally has higher precision when the recall is low i.e. < 0.5. The multiple object hypothesis approach though has higher recall when the precision is low. This appears to indicate that we can only take advantage of the more accurate probabilistic model in the lower precision range on our test data. The single object model loses out in this range, as it will tend to suppress hypotheses if they are significantly weaker than the most dominant hypothesis. In contrast, this behavior seems to help in the high precision range, since it is more likely to suppress spurious hypotheses. As shown by comparing these graphs, the latter effect becomes less pronounced as more data are gathered and the range in which the multiple object model dominates expands.

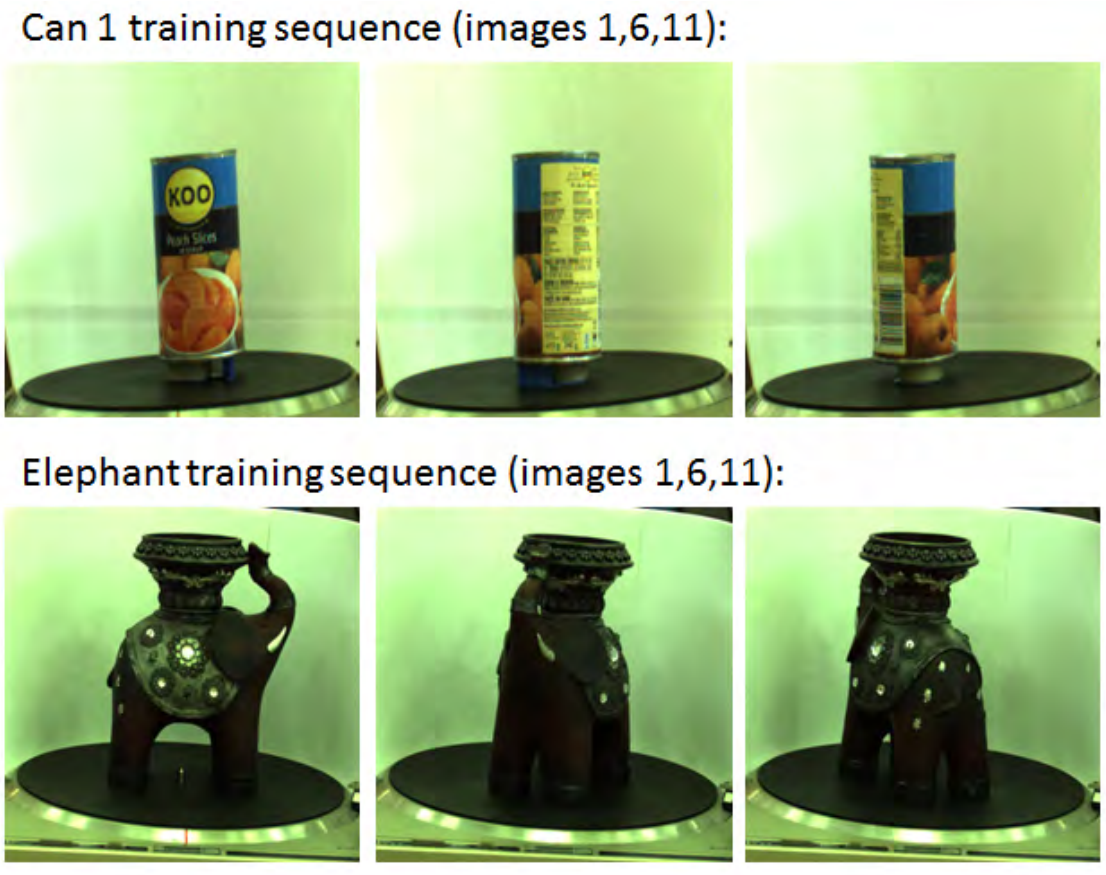


Figure 4.2: Example images from our training set.

Experiment 2: Comparing viewpoint selection algorithms

Our second experiment runs several tests to compare our proposed viewpoint selection strategy based on the vocabulary tree method with alternative selection strategies. We used 15 objects from the dataset for these experiments.

First, we test this strategy against a random selection mechanism (simply choosing at random one of the remaining viewpoints at each time step). We use the same setup and compare the primary object recognition rate for the multiple object recognition algorithm with the vocabulary tree method and random viewpoint selection rules in all combinations. As shown in Figure 4.5, the multiple object hypothesis approach using our viewpoint selection algorithm outperform those using random selection across a large range of viewpoints.

Second, we test our vocabulary tree viewpoint selection strategy against random selection and mutual information using the multiple object algorithm with likelihood model 2. We select all even-numbered viewpoints as our validation set to gather the statistics $\kappa_{o,\theta,n}^{\{0,1\}}$, and treat the remaining 9 odd-numbered viewpoints as our test set (we let $\kappa_{o,\theta,n}^{\{0,1\}} = 0.5 \sum_{m=\{n-1\ n+1 \bmod N_\theta\}} \kappa_{o,\theta,m}^{\{0,1\}}$) for odd n. As noted, these more extensive statistics (compared to likelihood model 1) are necessary in order to evaluate

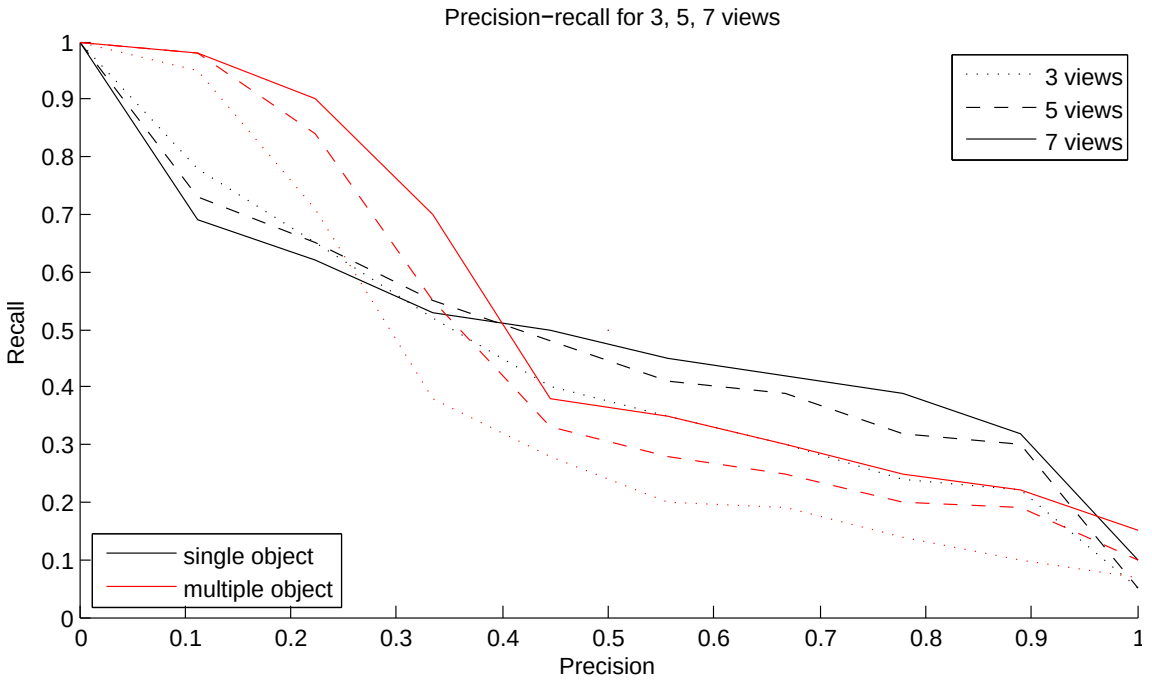


Figure 4.3: The average precision-recall curves for both primary and secondary objects for our single and multiple object models after 3, 5 and 7 viewpoints.

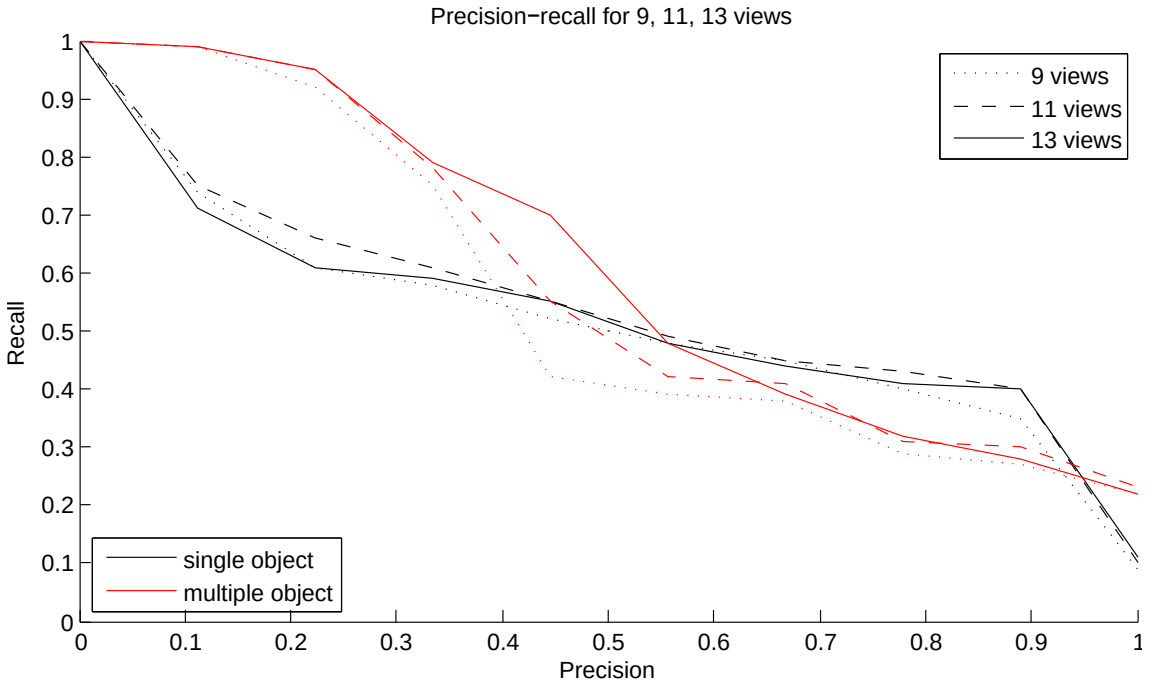


Figure 4.4: The average precision-recall curves for both primary and secondary objects for our single and multiple object models after 9, 11 and 13 viewpoints.

the mutual information per viewpoint. Figure 4.6 compares the performance of likelihood model 2 with 3 viewpoint selection rules: random, mutual information and our vocabulary tree method. As shown, both the vocabulary tree method and mutual information outperform random selection across

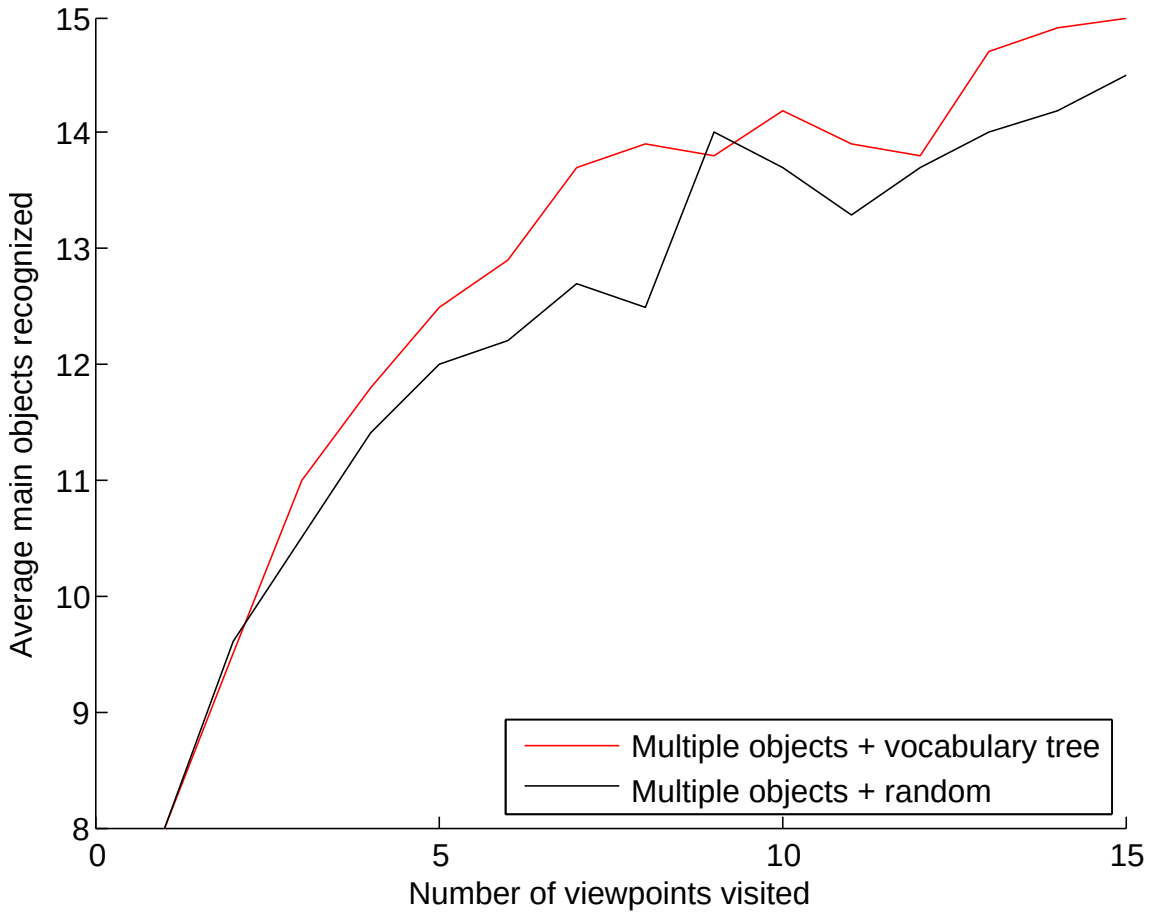


Figure 4.5: Comparison of the performance of the multiple object algorithm using the vocabulary tree method for viewpoint selection and randomly selecting the next viewpoint.

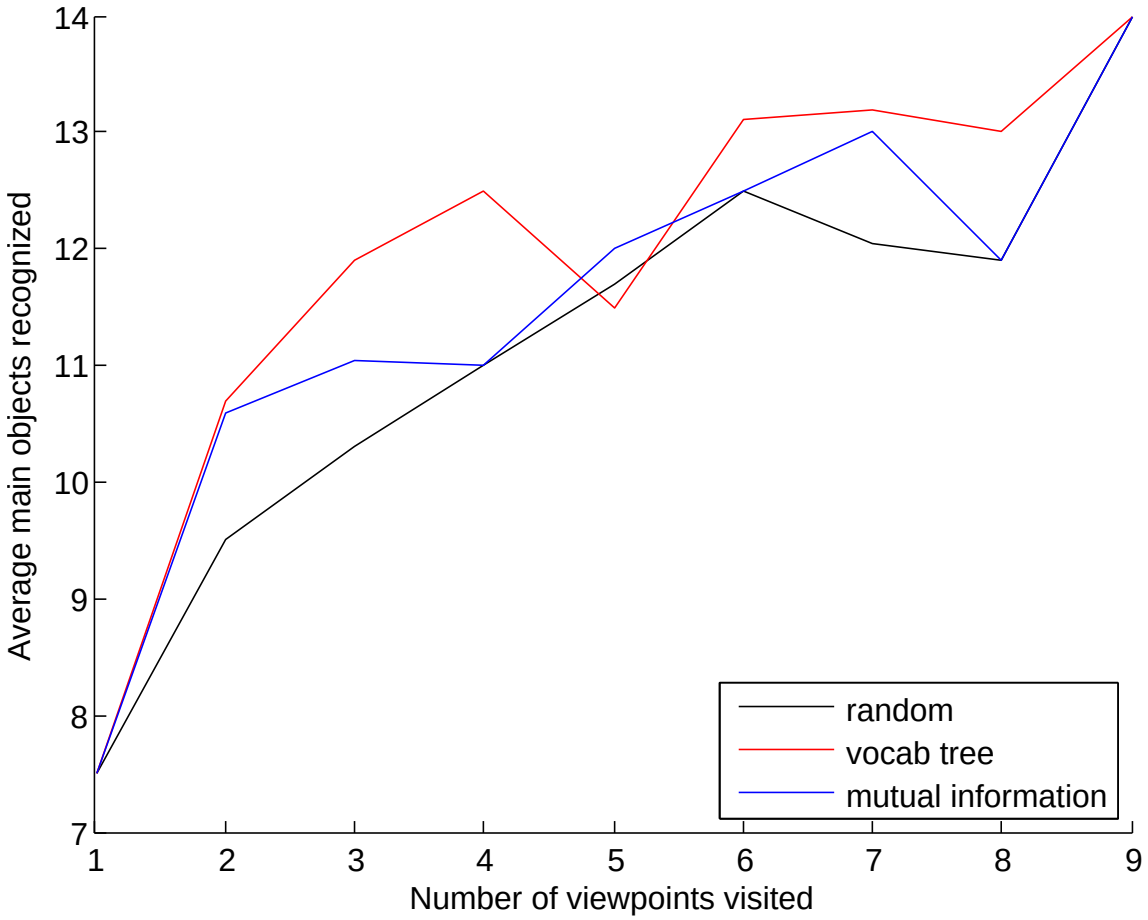


Figure 4.6: Comparison of the performance of our multiple object algorithm with likelihood model 2 using the vocabulary tree method, random, and a mutual information-based viewpoint selection strategy.

most viewpoints. The graph also shows that our vocabulary tree method performs competitively with mutual information, and in fact outperforms it on a range of viewpoints. This is despite the fact that the mutual information can be shown to be the optimal strategy if the probabilistic model is accurate [71]. The low performance of mutual information may thus be taken to indicate that insufficient data was available to estimate accurate statistics for likelihood model 2 in this setting. The results however generally support our claim that the vocabulary tree method provides an effective alternative to previous viewpoint selection mechanisms, and is robust to inaccuracies in the underlying model.

Finally, we test our vocabulary tree viewpoint selection strategy in a non-Bayesian context against the activation model used in Kootstra et al. [1]. Figure 4.7 compares the performance of [1] with the original expected activation selection mechanism, and with our vocabulary tree mechanism substituted for multiple object recognition. The graph shows the precision-recall curves for all objects present at a range of viewpoints. The curves are generated from 4 points, found by successively selecting the 1–4 objects with the highest ranked activations summed across viewpoints. The graph illustrates

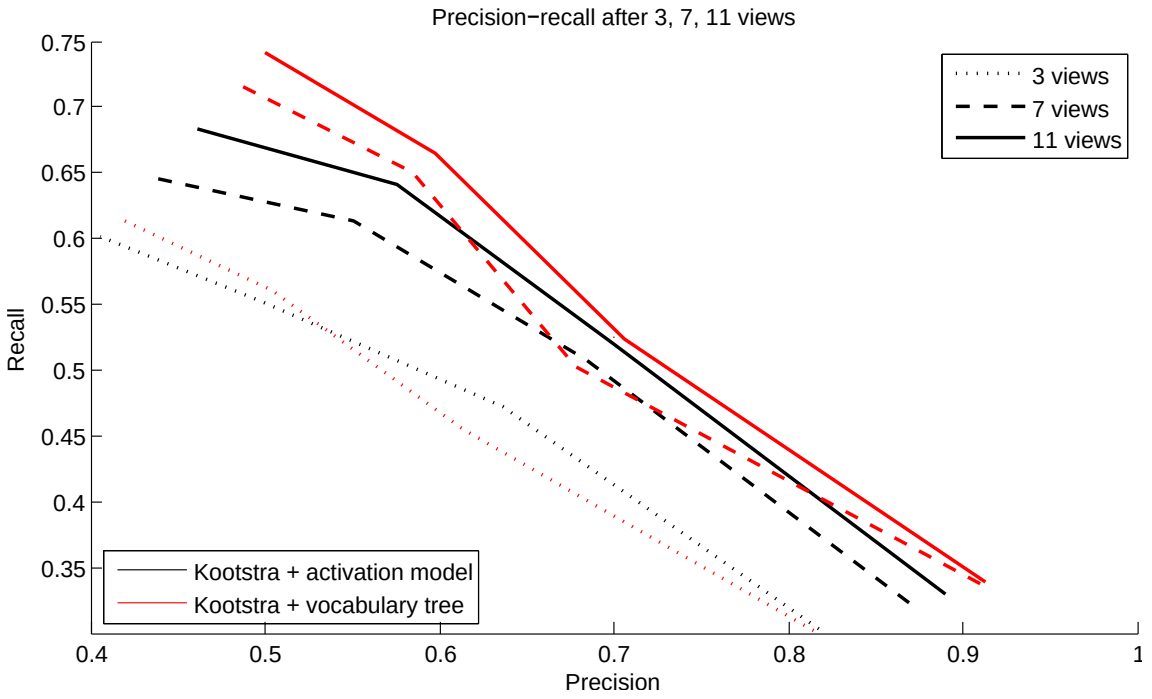


Figure 4.7: This graph compares the average precision-recall curves across all primary and secondary objects for Kootstra et al. method (black) and when highest ranking 1–4 objects are selected.

that when the recall is higher than approximately 0.5, Koostra et al. [1] with our viewpoint selection algorithm has higher precision. When the recall falls below 0.5, the precision values for both methods are comparable. The results thus validate our method's effectiveness in a non-Bayesian context.

Experiment 3: Multiple object recognition

In our final experiment, we investigate the performance of the multiple object recognition algorithm under various settings of the stopping conditions. In general we can use the stopping criteria to manipulate the trade-off between the overall accuracy of each algorithm and its expected overall timing. Table 4.1 gives a number of performance measures for the algorithm under different settings, where we manipulate μ^{stop} for both methods and N_{obj} for the multiple object method. The performance measurements are averaged across all 15 starting viewpoints, with 'score' being the average number of objects recognized in the primary object setting, and precision-recall values given when the top 1 and 4 ranked objects are selected.

Our single object method has slightly better performance than the multiple object method in the high-precision/low-recall range. The multiple object model is better in the low precision/high-recall range (as seen by comparing the results across methods in the prec(4) and rec(4) columns). We note that these results are only suggestive, since they may be affected by our particular implementations, and also in a realistic active vision situation factors other than processing time may be involved (such as time to move to a new viewpoint).

Table 4.1: Comparing timing and stopping criteria: The table compares average performance of the single and multiple recognition models. Shown are the average number of viewpoints, average time taken (s), average primary object score, average precision-recall across primary and secondary objects when taking the top-ranked object and the top 4 ranked objects and the stopping criteria applied.

	Ours (single)	**Ours(single)**	**Ours (multiple)**	**Ours (multiple)**
Views	8.9	11.6	9.7	11
Time	439.7	573.0	479.2	543.4
Score	14.1	14.1	13.5	13.6
Precision(1)	0.919	0.916	0.835	0.844
Recall(1)	0.341	0.340	0.310	0.313
Precision(4)	0.354	0.376	0.509	0.530
Recall(4)	0.525	0.558	0.756	0.786
Stopping criteria	$\mu \geq 0.5$	$\mu \geq 0.8$	$\mu \geq 0.5$, 2 objects	$\mu \geq 0.6$, 2 objects

Figures 4.8 and 4.9 displays the multiple object recognition results for our single and multiple object models and the activation model by Kootstra et al. For the test image in figure 4.8, two objects (robocop and spice 2) are required to be recognized. The activation model recognizes the robocop object and our single model approach recognizes the spice 2 object. Our multiple object model correctly recognizes both objects in the scene.

The ground truth sequence for the test images in figure 4.9 shows that four objects are to be recognized (can 1, can 2, sauce 1 and teddy bear). The activation model recognizes three of the four objects although the score for the teddy bear object is lower than two incorrect guesses. Our single object model correctly recognizes two objects with the multiple object approach correct recognizing all objects. In terms of timings, the multiple object approach requires slightly more time than the single object model but recognizes more objects. The activation model is significantly more computationally expensive.

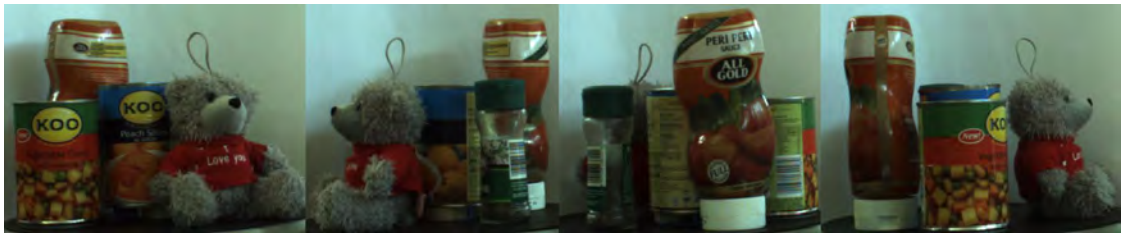
Spice 2 test sequence (images 1, 6, 11, and 16)

Ground truth	Kootstra et al.	Ours (single object)	Ours (multiple object)
robocop	elephant : 5.8869 -	spice 2 : 0.45797 +	spice 2 : 0.95604 +
spice 2	robocop : 5.6327 +	spray can 1 : 0.0495 -	robocop : 0.8573 +
	battery : 5.4467 -	salad : 0.04652 -	can 1 : 0.73769 -
	toy : 5.2443 -	elephant : 0.03516 -	salad : 0.58172 -

	Kootstra et al.	Ours (single object)	Ours (multiple object)
Timings (s)	1425	442.309	470.2307

Figure 4.8: Example images from our testing set. The list of ground truth objects in the sequence is displayed in column one. The highest ranking objects predicted by our single object and multiple object algorithms, along with the results of Kootstra *et al.* [1], are also shown. The final probabilities/scores of the predicted objects are shown, along with +/- to indicate if the object is present or not, and the time taken to process the sequence.

Can 1 test sequence (images 1, 6, 11 and 16)



Ground truth	Kootstra et al.	Ours (single object)	Ours (multiple object)
can 1	sauce 1 :1.1781 +	can 2 : 0.51366 +	can 2 : 0.98922 +
can 2	can 2 : 0.9814 +	sauce 1 : 0.032653 +	can 1 : 0.93986 +
sauce 1	spice 3 :0.74646 -	robocop : 0.028842 -	sauce 1 : 0.71543 +
teddy bear	handbag 2 : 0.57768	spray can 1 : 0.028831 -	sauce 2 : 0.59597 -
	teddy bear : 0.53704 +	salad : 0.024364 -	teddy bear : 0.50077 +
	robocop : 0.53436 -	toy : 0.02325 -	jewelry box 2: 0.23187 -

	Kootstra et al.	Ours (single object)	Ours (multiple object)
Timings (s)	1291.3333	335.469	371.537

Figure 4.9: Example images from our testing set. The list of ground truth objects in the sequence is displayed in column one. The highest ranking objects predicted by our single object and multiple object algorithms, along with the results of Kootstra *et al.* [1], are also shown. The final probabilities/scores of the predicted objects are shown, along with +/- to indicate if the object is present or not, and the time taken to process the sequence.

4.7 CONCLUSIONS

We have investigated active object recognition in the context of identifying groups of objects in cluttered scenes. We have shown that features formed from Hough-based geometric matching counts provide sufficient information to perform object recognition in this context, and that using geometric information allows us to achieve better performance against methods such as [1]. Further, we have developed a Bayesian framework which accurately models the assumptions of this testing context, and have shown that providing such an accurate probabilistic model can provide enhanced performance in certain circumstances. Finally, we have provided an extensive empirical evaluation of a multiple object version of the vocabulary tree viewpoint selection strategy introduced in Chapter 2, and have shown this to provide an efficient and accurate alternative to strategies such as mutual information, and non-Bayesian approaches.

In general, our results suggest that techniques for coping with object recognition in clutter can be used effectively in an active vision context. Indeed, our results show that the added robustness of the active vision setting allows simpler overall representations to be used, as seen by comparing our count-based features with the approach of [12].

CHAPTER FIVE

2D ACTIVE OBJECT RECOGNITION USING FOURIER DESCRIPTORS AND MUTUAL INFORMATION[1]

5.1 INTRODUCTION

2D and 3D object recognition is essential for robotic platforms to navigate and interact in both static and dynamic human environments. The use of these robotic platforms in industrial and household environments are steadily on the increase. They are finding applications in a wide range of fields from medical science to manufacturing to space exploration. In manufacturing, robots are used to weld, bend, sort or even perform quality inspection. In many production lines these robots are required to recognize different objects of varying shapes and sizes. Textural and feature-based approaches are often not appropriate for these types of applications because parts may contain little or no distinctive features other than boundary shape. Environments may also not have consistent lighting conditions which can also adversely affect these approaches.

In this chapter we investigate an alternate method to active object recognition using the shape in-

[1]Related publications:

- N Govender, J Claassens, "Recognition of Arbitrary 2D Shapes for Pick and Place solutions using Robot Manipulators", Pattern Recognition Association of South Africa (PRASA), November 2011.
- N Govender, J Warrell, P Torr, F Nicolls, "Probabilistic Models for 2D Active Shape Recognition using Fourier Descriptors and Mutual Information", Advances in Computer Science, Istanbul, Turkey, 22-23 August 2014, pp. 69-74.

formation from the object instead of local interest points. Active vision is accomplished in these experiments using mutual information. This shape recognition system using mutual information was designed and implemented as a proof of concept. Experiments are conducted to determine if the addition of an active vision component to a different type of recognition application will improve the overall accuracy.

We initially introduce a shape recognition system using Fourier descriptors to model the shape information. Fourier descriptors are widely used in the field of computer vision for recognition applications [75–78]. Our database for these experiments consists of ten animal shapes which are required to be placed in the correct position on a puzzle board. Certain animal shapes are similar and these were selected to determine the robustness of the system. The system is tasked with recognizing the correct position on the board for each shape. Here, recognition is achieved by calculating the Euclidean distances between the Fourier descriptors extracted from the training and test objects. The minimum distance between a training and test object indicates a match. Experiments were conducted using both complex and polar coordinates with the polar coordinate shape signature producing better results. This system however, which contains no active vision component, does not recognize all the shapes correctly.

We then extend the Fourier descriptor shape recognition system to actively look for information to determine if this improves the recognition accuracy of the system. Many algorithms have been developed for object recognition using shape information with varying results. However, few of the proposed methods actively look for additional information to improve the initial recognition results. We investigate using mutual information as proposed by Denzler et al. [71] to actively look for information. In this context, the object recognition system is presented with a sequence of the ten animal shapes and is required to determine the correct sequence of the shapes. We build multinomial and Gaussian probabilistic models using the extracted Fourier descriptors and show how actively looking for cues using mutual information can improve the overall results. When the system is determining the correct object sequence, mutual information provides the system with the mechanism to select the position in the sequence that it is most uncertain about. These probabilistic models achieves excellent results, and improve on the initial system. Our experiments show that using the probabilistic models with mutual information outperforms the use of Fourier descriptors as well as the probabilistic models without mutual information. The Gaussian model with mutual information correctly recognizes all the objects.

Section 5.2 provides background information on Fourier descriptors, probabilistic-based models for shape recognition and mutual information. The dataset used in our experiments is introduced in section 5.3. Section 5.4 discusses the process of calculating Fourier descriptors with the initial experiments and results are presented in section 5.5. The multinomial and Gaussian probability models are described in section 5.6. The experiments conducted and the results using the probability models and mutual information are shown in section 5.7 with the conclusions in section 5.8.

5.2 RELATED WORK

Numerous methods exist for extracting information from objects for object recognition tasks. These range from parametric eigenspace data [47, 48] and entropy maps [50] to extracting local features [1, 51]. Depending on the application context, different methods are selected. In our experimental setup the objects have no visual texture and are therefore not conducive to the extraction of local interest points. Instead we use the shape to discriminate between various objects.

Various shape representation methods, or shape descriptors, exist in the literature. These can be classified into two categories: region-based and contour-based methods. In region-based techniques, all the pixels within a shape are taken into account to obtain the shape representation [79,80]. Contour-based shape representation exploits shape boundary information. Fourier descriptors are contour-based and capture global shape features in the first few low frequency terms, while higher frequency terms capture finer features of the shape. Wavelet descriptors can also be used to model shape and have an advantage over Fourier descriptors in that they maintain the ability to localize a specific artifact in the frequency and spatial domains [81]. However, wavelet descriptors are impractical for higher-dimensional feature matching [82].

A comparison between various shape descriptors was conducted by [83] which included Fourier Descriptors (FD), Curvature scale space (CSS), Curvature scale space descriptors (CCSD), Zernike Moment Descriptors (ZMD) and grid descriptors. Results show that in terms of affine invariance, robustness, compactness, low computation complexity, ZMD and FD outperformed the other methods.

We use Fourier descriptors in our experiments due to their versatility. They have been used in a variety of fields over the years, including in the commerce, medical, space exploration and technical sectors. In the field of computer vision, Fourier descriptors have been used for human silhouette recognition for surveillance systems [78], content-based image retrieval [84,85], shape analysis [77,86], character recognition [75, 87] and shape classification [76]. In these methods, different shape signatures have been exploited to obtain the Fourier descriptors. These include using the central distance, complex coordinates, polar coordinates, curvature functions and cumulative angles [88]. We use both complex and polar coordinates in our experiments.

There have been a number of probabilistic-based models for shape recognition proposed, such as using Procrustean models [89], probability density functions [90], geometric features [91] and generative models [92]. None of these methods use Fourier descriptors as their input to the shape recognition system. In addition none of these methods use active vision by incorporating mutual information to improve their initial results. Mutual information was introduced as a viewpoint selection mechanism for active vision [71], which has been subsequently in various active vision settings [44, 72, 73]. Mutual information can be expensive to calculate and requires the collection of extensive statistics at training time, although discussed in [71], it provides the optimal strategy provided the underlying models are correct. Using mutual information also makes it easy to incorporate probabilistic assump-

tions to assist with active information selection. Our framework follows that of [62,71] in terms of the general Bayesian form of our updates and we use a sampling scheme to make the mutual information calculations tractable.

5.3 DATASET

A board containing cut out shapes of different animals was used in the experiments. The shapes were removed from the board and placed on a table. Figures 5.1 and 5.2 display the board and the shapes used in the experiments, respectively. Mutual information requires extensive statistics for training. To enable us to use mutual information, 20 close-up images for each shape were also captured. Information from these images is used as input in the probabilistic models. Examples of the close-up images are displayed in Figure 5.3.

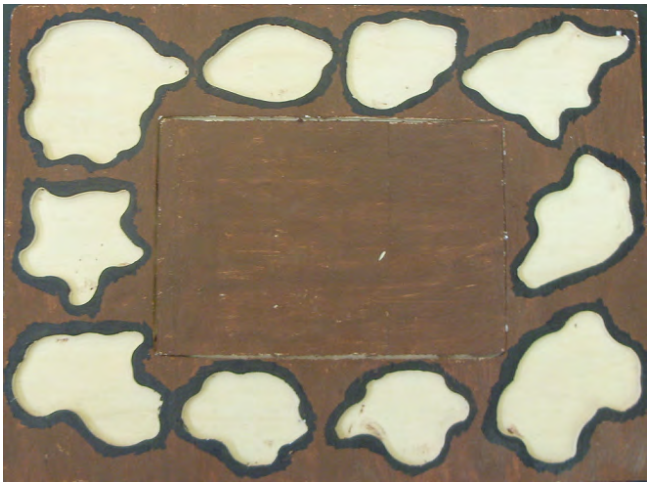


Figure 5.1: The board with the shapes removed.

Figure 5.2: The animal shapes to be recognized.

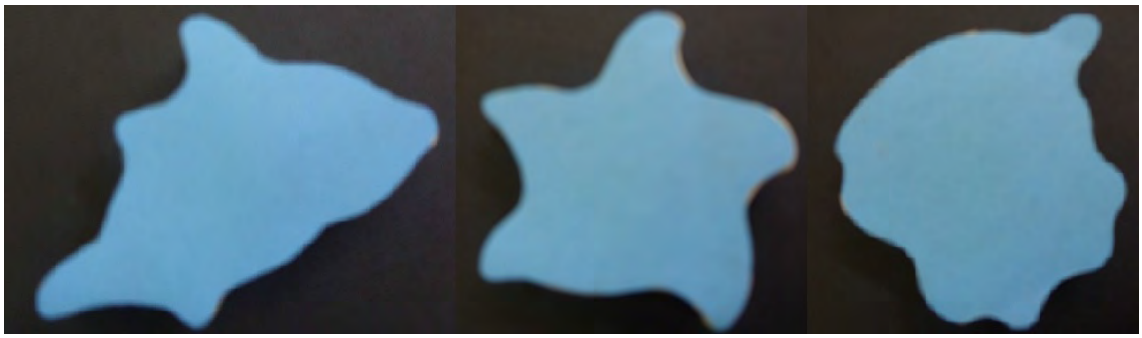


Figure 5.3: Close-up images of the shapes.

5.4 FOURIER DESCRIPTORS

Fourier transforms are used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the frequency domain. The term frequency here refers to the variation in brightness or color across the image, i.e. it is a function of spatial coordinates, rather than time. Transforming an image from the spatial domain to the frequency domain allows large filtering operations to be completed faster and facilitates the removal of noise from images, amongst other operations and measurements which would not be possible in the spatial domain.

In our experiments we use Fourier descriptors to describe the boundary of a shape in 2D space. To accomplish this, we first extract the boundary coordinates of the shape starting at an arbitrary point on the boundary. The boundary is then traversed extracting the (x, y) coordinates. It is not necessary to extract every point on the boundary. A sampling method can be introduced, for instance extracting every second or third boundary coordinate. Once the set of boundary coordinates have been extracted, we represent each coordinate by (x_k, y_k) where $0 < k \leq N - 1$ and N represents the number of boundary coordinates. As the Fourier transform works with complex numbers, each coordinate pair can be treated as a complex number such that $s(k) = x(k) + iy(k)$ for $k = 0, 1, 2, ..., N - 1$, where the x-axis is treated as the "real" and the y axis as the imaginary axis. This changes the interpretation of the sequence but the nature of the boundary itself has not been changed. The advantage of this representation is that it reduces a 2D problem into a 1D problem. The discrete Fourier transform (DFT) of $s(k)$ is

$$a(u) = \frac{1}{N} \sum_{k=0}^{n-1} s(k) e^{\frac{-i2\pi uk}{n}}, \tag{5.1}$$

for $u = 0, 1, 2, ..., N - 1$. The complex coefficients $a(u)$ are called the Fourier descriptors of the boundary.

The (x, y) boundary coordinates can also be represented as polar coordinates, that is in terms of its distance from the origin (magnitude), and the angle that it makes with the positive real axis (angle):

$$x = r \cos \theta y = r \sin \theta \tag{5.2}$$

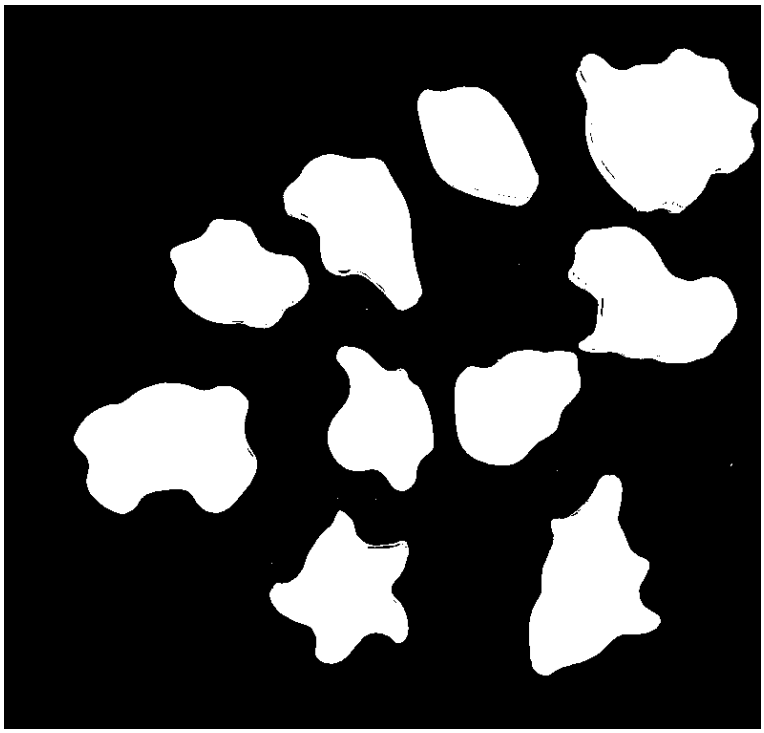


Figure 5.4: Binary images of the test data.

where $r = \sqrt{(x^2 + y^2)}$ and $\theta = \tan^{-1}(y/x)$.

The Fourier descriptors are a frequency-based description of the boundary of an image. Comparing the descriptors of different objects gives a measurement of their similarity, which is usually accomplished by calculating the Euclidean distance.

5.4.1 EXTRACTION

The training images captured are converted into binary images, as shown in Figure 5.4.

We found that converting images into binary before performing edge detection produced better results. Edge detection is performed on these images and each boundary is then segmented and the Fourier descriptors extracted. Figure 5.5 displays the result after edge detection.

The extracted boundary coordinates were converted to both complex and polar coordinates. We found the polar coordinates to provide gains in storage and computational cost as compared to using complex coordinates. Rotation, translation, scaling and the starting point on the boundary influences the descriptor that is calculated. The descriptors, calculated from two different images, of the same object will differ if their rotation, translation, scaling or starting points are different. This however can be easily removed. Translation has no effect on the descriptors, except were $k = 0$. Ignoring the first descriptor removes the effect of translation. Scaling affects the boundary and the descriptors by the same amount. This constant can easily be calculated and scaling corrections can be made. Variances

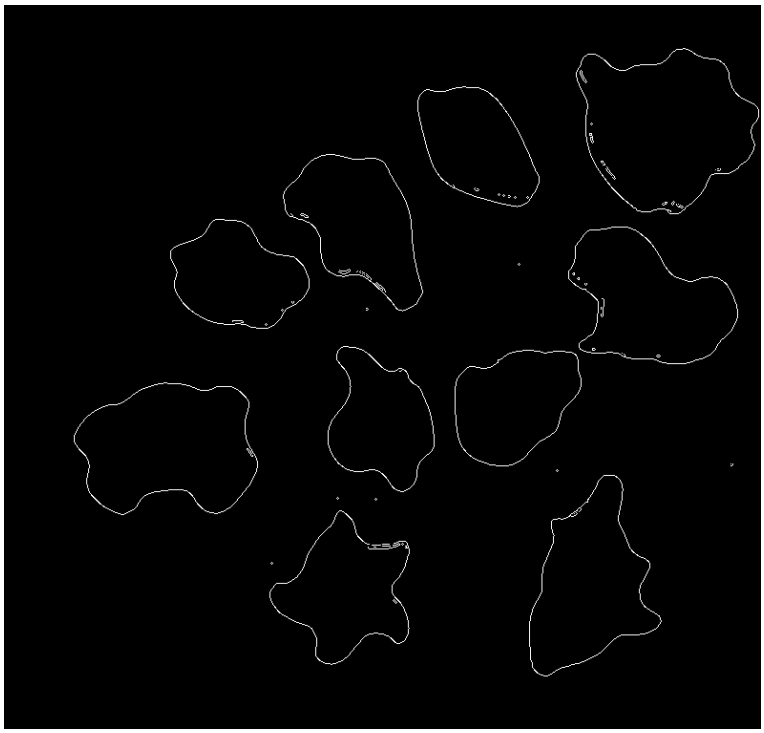


Figure 5.5: Edge detection on the images.

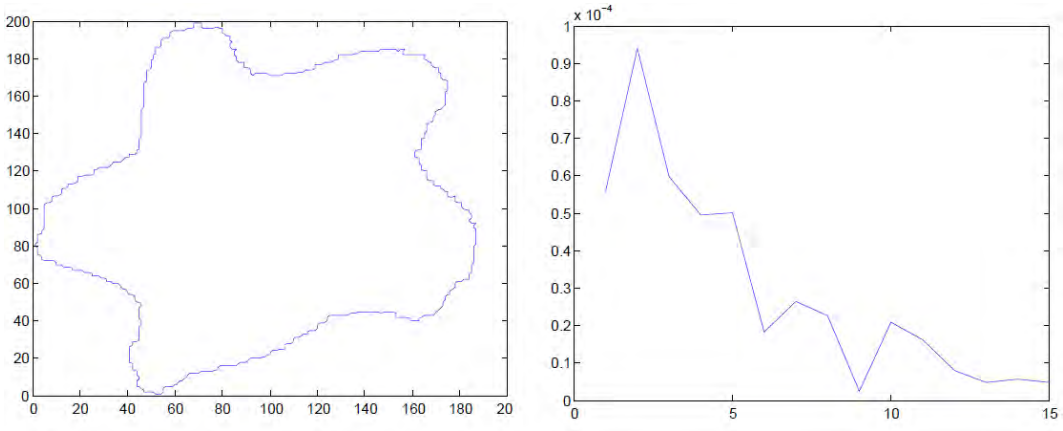


Figure 5.6: Fourier descriptor representation of the star shape.

due to rotation and differences in starting points are removed at the same time by taking the absolute value of each descriptor.

Figure 5.6 displays the magnitude of the first 15 Fourier descriptors extracted for the star shape.

5.5 SHAPE RECOGNITION

Once the boundary coordinates have been extracted and the Fourier descriptors calculated for both complex and polar coordinates, recognition is accomplished by matching the Fourier descriptors from the test images to those extracted from the training set by calculating their using Euclidean distance.

It has been shown that the low-frequency components of the Fourier transform are sufficient to capture the primary shape of a boundary [78, 84, 88] and thus the entire transform does not need to be used. The lower frequency components of the boundary form the basis for differentiating between distinct boundary shapes. By using fewer terms to approximate the boundary some of the higher frequency components of the boundary are lost, leading to a loss in fine detail.

We found that using the first 15 Fourier coefficients (excluding the very first component $F(0)$) provided sufficient discriminatory information to model a shape. The lowest frequency term $F(0)$ is the only component in the Fourier descriptor that is dependent on the actual location of the shape. By ignoring the first component, it becomes translation invariant. The $F(0)$ component tells us nothing about the shape; only the mean position. The Fourier descriptor is then normalized to remove any scaling effects. Since the energy in the Fourier components decreases sequentially, we artificially boost the contribution of each component. For our experiments components 0–5 were multiplied by a factor of 5, components 6–10 by a factor of 10 and the remaining components by a factor of 15. These values were determined empirically. The shape on the board with the smallest Euclidean distance to a shape in the test image is considered to be the match.

5.5.1 RESULTS

The shapes used in the experiments are in the form of animals, which include a tortoise, whale, seal, dolphin, fish, crab and so on as seen in Figure 5.2. There are ten shapes in total. Table 5.1 shows the results obtained using both complex and polar coordinate representations of the Fourier descriptors for shape recognition.

Using the first 15 Fourier descriptors, the Euclidean distances were calculated between each object in the training and testing set. The minimum distance indicated a match. The complex coordinate shape signature correctly recognized three shapes while using polar coordinates correctly recognized eight shapes. The latter incorrectly identifies the fish and the tortoise shapes. Experiments were also conducted using the first 20, 25 and 30 Fourier descriptors with no improvement in the recognition accuracy in either method.

Table 5.1: Recognition results.

Shape	Complex coordinate method	Polar coordinate method
Whale	yes	yes
Seal	no	yes
Fish	no	no
Crab	no	yes
Dolphin	yes	yes
Mussel	no	yes
Snail	no	yes
Octopus	no	yes
Star Fish	yes	yes
Tortoise	no	no

5.6 PROBABILITY MODELS

Using polar coordinates to calculate the Fourier descriptors for shape recognition achieved an accuracy of 80%. We now investigate including an active vision component to improve the recognition accuracy. In these experiments we place the ten shapes in a specific sequence and the system is required to determine the correct order of the shapes in the sequence. We use mutual information to actively look for information about which shape in the sequence it is most uncertain about and additional information is then gathered for that shape. We build multinomial and Gaussian probabilistic models using the extracted Fourier descriptors and conduct experiments to determine if actively looking for cues using mutual information improves the overall results.

5.6.1 MULTINOMIAL DISTRIBUTION

For the multinomial distribution we extracted the Fourier descriptors from the dataset containing the close-up images of the shapes. The 20 close-up images for each shape were split into two sets containing 14 images for training and 6 images for testing. The training set was further split into two sets containing 7 images each. One was used for training and the other as a validation set. This was done to determine a quasi-ground truth histogram distribution which can be used for testing. The Euclidean distance was calculated between every image in the training and validation set. This process was carried out 10 times. The minimum distance value was then calculated, which identified the object class for each image. A distribution histogram for each image class was then calculated. A bias was placed at the correct class to provide the system with a reliable ground truth distribution.

Let N be the number of shapes and D the dimension of the Fourier transforms (in this case we used 15 descriptors). Let x represent a possible permutation $x \in \mathcal{P} \subset \{1, ..., N\}^N$. Here $\mathcal{P}$ denotes all

permutations of N objects, hence is a subset of $\{1, ..., N\}^N$ which contains no repetitions. Observation O for the close-up shapes takes the form $O = [O_1, O_2, ..., O_N]$, where $O_n \in \mathcal{V} \subset (\mathbb{Z}^+)^N$ are the counts in the histogram for test images in class n derived above. Let $\theta = [\theta_1, \theta_2, ..., \theta_N]$ represent the parameters of the distributions for each of the object classes. For the multinomial model we use $\theta_n = \alpha_n$, where α_n is the multinomial mean vector set using the counts from the training images. For initialization a noisy prior is selected for the board. This is done to incorporate the variability that may occur due to illumination changes and the camera or lens used. The prior for the board can be represented by $\pi(x)$. The probability of a permutation given all observations is described as

$$P(x|O) = \frac{P(O|x) \cdot \pi(x)}{P(O)} \propto \pi(x) \prod_n P'(O_n|\theta_{x_n}), \tag{5.3}$$

where $P'(O_n|\theta_{x_n}) = \text{Mult}(O_n|\alpha_{x_n}) = (M!/\prod_m O_{nm}!) \prod_n \alpha_{x_n}^{O_{nm}}$ for the multinomial likelihood, m ranges across the histogram bins, and M is the number of test images per class.

Bayes' theorem can be used to update the probability after each new individual observation. This is given by

$$\begin{aligned} P_0(x) &= \pi(x) \\ P_t(x|O_1..O_t) &\propto P'(O_{n(t)}|\theta_{x_{n(t)}}) P_{t-1}(x|O_1, ..., O_{t-1}), \end{aligned} \tag{5.4}$$

where $n(t)$ is the index of the observation seen at time t.

Mutual information (MI) assists in the selection of the position to look at since there can be no repetitions. Once the system is fairly certain of the position of a class in the permutation, mutual information can assist in deciding which position to look at next, namely the position that is the most uncertain. Randomly selecting the next position to look at does not not take this information into account. The MI selection rule is as follows:

$$n(t+1) = \text{argmax}_{n \neq n(1),...,n(t)} \text{MI}(O_n; x). \tag{5.5}$$

Mutual information values increase with uncertainty. In this equation we want to select the position in the permutation with the most uncertainty for a given observation.

We can rewrite the above equation in terms of the conditional entropy as follows:

$$MI(O_n; x) = H(x) - H(x|O_n), \tag{5.6}$$

where $H(.)$ represents the Shannon entropy and $H(.|.)$ represents the conditional entropy. We need to minimize the conditional entropy. This is described as

$$n(t+1) = \text{argmin}_{n \neq n(1),...,n(t)} H(x|O_n). \tag{5.7}$$

The conditional entropy can be written as

$$\begin{aligned} H(x|O_n) = -\sum_{O_n \in \mathcal{V}} P_t(O_n) [\sum_{x' \in \mathcal{P}} & P(x'|O_n, O_{n(1)}, ..., O_{n(t)}) \\ & \log(P(x'|O_n, O_{n(1)}, ..., O_{n(t)}))]. \end{aligned} \tag{5.8}$$

To evaluate $P_t(O_n)$ we introduce mixing coefficients β:

$$\beta_m = \sum_{(x|x_n=m)} P_t(x) \tag{5.9}$$

for $m = 1, ..., N$, which weight the likelihoods for each class. This gives us

$$P_t(O_n) = \sum_m \beta_m P'(O_n|\theta_m). \tag{5.10}$$

To avoid exhaustively summing across $\mathcal{V}$ in equation 5.8, we can consider the conditional entropy as the expectation across $P_t(O_n)$ and approximately evaluate the sum by sampling from this distribution:

$$\begin{aligned} H(x|O_n) &= E_{o\sim P_t(O_n)}[H(x|o)] \\ &\approx \frac{1}{n}\sum_{o_i} H(x|o_i) \\ &= -\frac{1}{n}\sum_{o_i}[\sum_{x'\in\mathcal{P}} P(x'|o_i, O_{n(1)}, \ldots, O_{n(t)}) \cdot \\ &\qquad \log(P(x'|o_i, O_{n(1)}, ..., O_{n(t)}))], \end{aligned} \tag{5.11}$$

where E denotes expectation. In equation 5.11, o_i represents the samples drawn from the mixture distribution.

5.6.2 GAUSSIAN DISTRIBUTION

The image set was treated in the same manner as used in the multinomial distribution. The training images were used to learn a Gaussian distribution for each class, $\theta_n = (\mu_n, \sigma_n)$. For σ_n we used a diagonal covariance matrix. For each observation O_n we included all test images $O_n = [O_{n1}, ..., O_{nM}]$, where M is the number of test images per class. The feature space is $\mathcal{V} = (\mathbb{R}^+)^{DM}$ since we have one D-dimensional Fourier descriptor for each image. For the likelihood in equation 5.10 we used the joint likelihood of these observations:

$$P'(O_n|\theta) = P'(O_n|\mu, \sigma) = \prod_{i=1,...,M} \mathcal{N}(O_{ni}|\mu, \sigma), \tag{5.12}$$

where $\mathcal{N}$ represents the Gaussian distribution and O_{ni} is the ith descriptor of observation n.

Since the feature space is now continuous the summation in equation 5.8 changes to an integral. We can use the sampling technique to approximate this integral:

$$\begin{aligned} H(x|O_n) &= \int_{\mathcal{V}} H(x|o)P_t(o)do \\ &= E_{o\sim P_t(O_n)}[H(x|o)] \\ &\approx -\frac{1}{n}\sum_{o_i}[\sum_{x'\in\mathcal{P}} P(x'|o_i, O_{n(1)}, ..., O_{n(t)}) \cdot \\ &\qquad \log(P(x'|o_i, O_{n(1)}, ..., O_{n(t)}))]. \end{aligned} \tag{5.13}$$

The sampling distribution used here is the same as described in equation 5.11, with $P'(O_n|\theta)$ as in equation 5.12.

5.7 EXPERIMENTS

The sequence on the board was used as the ground truth. Noise was added to the initial models to take into account possible illumination changes and noise introduced by the camera. Each simulation was run 100 times with a different split of the training and the testing images each time. For both models we want to identify the correct sequence. Once the system is fairly certain about the object at a specific position, mutual information allows us to select the next position to look at as the one that the system is most unsure about. In the random case this position is randomly selected. We introduced an artificial flipping method where two object positions would be flipped at random for 20% of the objects. The reason for doing this was to demonstrate the effectiveness of using mutual information when the initial guesses are not very accurate. We ran simulations with the restricted number of objects in the sequence ranging from four to ten, and display the results when only seven and ten objects are used. Since we found it was computationally expensive to go through all the possible combinations when using nine or ten objects, we introduced the sampling method as discussed in equation 5.11 to reduce this complexity.

5.7.1 MULTINOMIAL DISTRIBUTION

The multinomial distribution is initialized using the class histogram calculated at the start. The probabilities of correctly predicting the sequences containing seven and ten objects are 92% and 94% respectively, as shown in Figure 5.7. These are the results after conducting the simulation 100 times. Using the multinomial distribution provides better results than just using the Euclidean distance of the Fourier descriptors as shown in Table 5.1. In addition, using MI to select the next position in the sequence to look at outperforms randomly selecting this position. For the seven object sequence, using MI achieves an accuracy of on average 92% after investigating three to four positions in the sequence while random selection looks at all seven viewpoints. For the ten object sequence, six viewpoints are processed on average by MI and nine by the random method to achieve an accuracy of 94%.

5.7.2 GAUSSIAN DISTRIBUTION

In the Gaussian case the class histogram is not used. Instead we use a covariance matrix to calculate the likelihoods as explained in section 5.6.2. The probabilities after looking at seven and ten views are 99% and 100% respectively, as shown in Figure 5.8. Using the Gaussian distribution outperforms both the Euclidean distance and the multinomial distribution methods. In these experiments MI also achieves better results then just randomly selecting the next viewpoint to process. On average MI

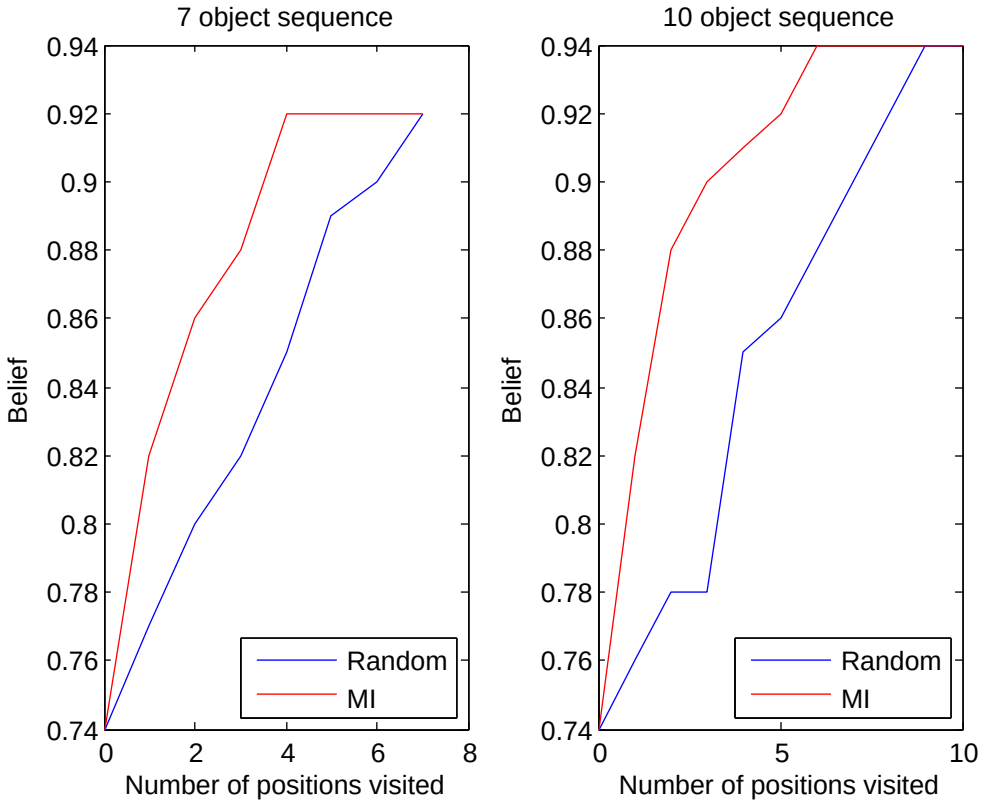


Figure 5.7: The system's belief in the correct sequence after 7 and 10 views for the multinomial distribution.

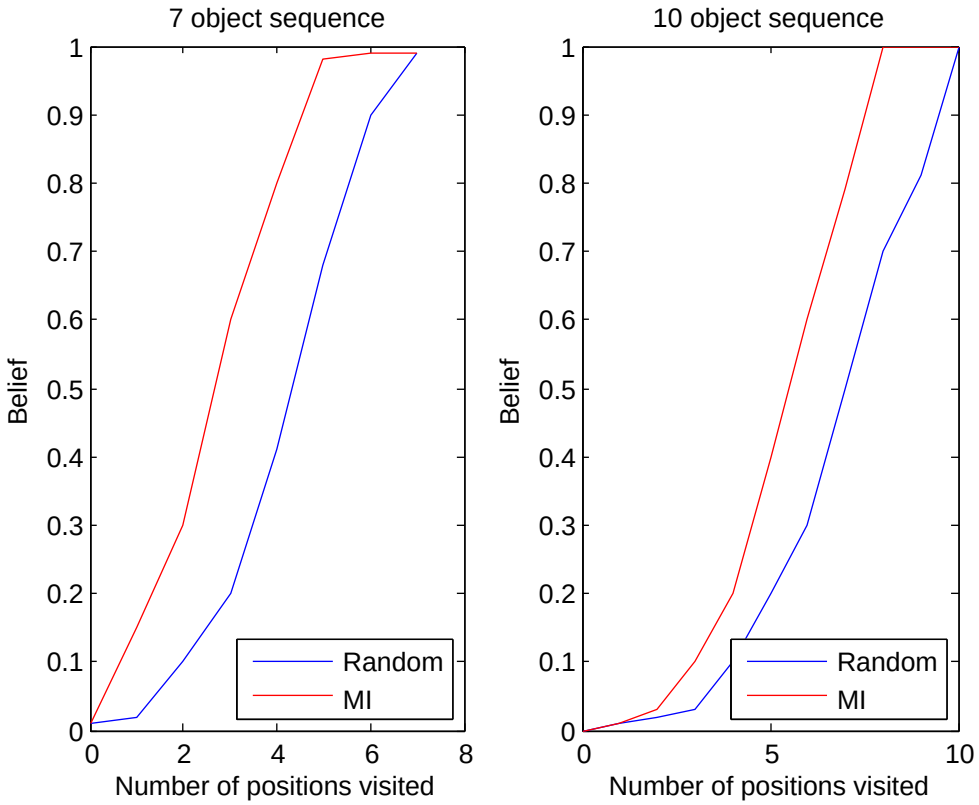


Figure 5.8: The system's belief in the correct sequence after 7 and 10 views for the Gaussian distribution.

reaches an accuracy of 99% after processing five viewpoints and 100% after processing eight viewpoints on average for the seven object and ten object sequences, respectively. The random method requires all the seven viewpoints for the seven object sequence and nine viewpoints for the ten object sequence, on average.

In Table 5.2 the average timings are given for choosing the position to view next and updating the current distribution after making an observation. For the mutual information, we used 20 samples per position. The timings are similar for the methods usings both the multinomial and Guassian distributions. As shown, the mutual information increases the time taken over random selection, although this could be reduced by using fewer samples and trading off accuracy.

Table 5.2: Timings for single position update.

Method	Number of Objects	Random (s)	MI (s)
Multinomial	7	0.003	0.074
	10	0.155	33.916
Guassian	7	0.013	0.090
	10	0.173	33.651

5.8 CONCLUSIONS

We have presented a system which extracts Fourier descriptors from ten different animal shapes to be used for shape recognition. Initially recognition was performed using the Euclidean distance between the shapes. This resulted is an accuracy of 80% with the system confusing the fish and the tortoise shapes. We then set about using the Fourier descriptors from the shapes in multinomial and Gaussian probability models. The results show that probabilistic models with mutual information outperform using Fourier descriptors as well as the probabilistic models without mutual information. Using mutual information to actively select the next most uncertain position in the sequence provides better results than randomly selecting the next position. Both the multinomial and Gaussian models correctly identify all the objects. We have shown that using probability models for shape recognition, incorporating information about the current state of the system (MI) and actively selecting which uncertain position to look at, produces excellent results and improves on the overall accuracy of a system without an active vision component. This 2D shape recognition system was implemented to demonstrate the effectiveness of active vision in different recognition scenarios. Future work could include using these probabilistic models on larger and different types of datasets to determine their robustness.

CHAPTER SIX

CONCLUSION

Many methods have been developed for active object recognition with different techniques for object representation, selection of the next best viewpoint and fusion of extracted data in both probabilistic and non-probabilistic settings. These systems have achieved varying results but most were tested on scenes containing a single object with no occlusion or clutter. In this thesis, we presented novel approaches to 3D and 2D active object recognition which are implemented in the form of systems. Systems 2, 3 and 4 perform 3D active recognition using the SIFT local interest point detector and descriptor and a vocabulary tree data structure. System 5 uses the shape of an object and mutual information for actively recognizing a 2D object.

System 2, which forms the basis of the 3D active object recognition system, comprises automatic viewpoint selector and observer components. The automatic viewpoint selector uses the SIFT features extracted from the test images as input to the vocabulary tree data structure. The vocabulary tree is used to calculate a uniqueness weighting for each feature. The next best viewpoint is determined by summing up the weights for all the features that appear in a specific viewpoint. The higher the weighting, the more unique the viewpoint. A Bayesian framework is used in the observer component to update the system's belief in the identity of an object. Bayesian probability theory provides a principled manner for data to be integrated across multiple views. New images are only captured if the belief in the identity of any object in the database, in the case of recognition, is below a predefined threshold. Reducing the number of images processed to unambiguously identify an object reduces the computational time required by the system. The test images used consisted of 20 everyday objects appearing in cluttered environments with occlusion. Our system outputs the identity of the object and the belief value.

Experiments were conducted for object verification and recognition. Object verification just deter-

mines if a specific object is in a scene while recognition finds the identity of the object, if any, in the scene. For object verification, the system correctly verifies all objects and requires fewer viewpoints to do so than randomly selecting the next viewpoint, in some cases significantly so. For object recognition, our approach correctly identifies 18 out of the 20 objects in the database. The two objects that were not recognized were a spray can and a spice bottle that were significantly occluded in the test images. This method, however, outperforms randomly selecting the next viewpoint. It was then tested against a state-of-the-art active object recognition system [1] that is based on local interest features. Our system correctly recognizes more objects and is less computationally expensive. When the system designed by Kootstra et al. [1] does recognize an object it requires fewer viewpoints, in certain cases. This could be attributed to the fact that they select the next viewpoint based on the test images and not on a predefined sequence, as in our case. The activation model used by [1] does not incorporate an additional filtering step or enforce any geometric constraints after matching, which could lead to substantial noise being added to the recognition process. Hence it recognizes fewer objects.

System 3 was designed and implemented to improve upon the recognition accuracy of System 2. The extracted SIFT features were used as input into Bayesian probability models for object recognition and object and pose recognition. These models were explicitly designed to cope with the challenging dataset. Three likelihood models were presented, each with increasing levels of complexity. The first model uses the extracted features directly with no additional filtering on the matches. This does not perform well, only recognizing 20% of objects. The second likelihood model adds an additional geometric filtering step using the Hough transform. This model recognizes 80% of objects in the database. The final likelihood model includes parameters for modeling occlusion, the background distribution and geometric filtering. This occlusion model correctly recognizes all objects in the database. It also recognizes 90% of the viewpoints/poses correctly to within 20°. The occlusion model is also computationally faster than the binary model and the vocabulary tree model presented in System 2. Given these results we can conclude that it is important to add additional geometric filtering steps, and include some mechanism to model the occlusion and/or background for object recognition in cluttered scenes.

Having presented the probabilistic object and viewpoint models, which are able to recognize single objects present in cluttered test images, we extended this probabilistic framework to recognize multiple objects and their poses in a scene (System 4). The test images for these experiments contained any number of objects from the database that were required to be recognized, as well as distractor objects which do not appear in the dataset. This approach was designed to recognize multiple objects and their poses, if any were present from the dataset, in the test images. The Bayesian approach to single object and pose recognition introduced in System 3 was extended to multiple object and pose recognition. The next best viewpoint selection algorithm based on the vocabulary tree data structure was also extended for multiple object recognition. This viewpoint selection algorithm was compared to random selection, using mutual information and the activation method presented in [1] and was

shown to perform well in terms of efficiency and accuracy. Our model was able to recognize multiple objects and their poses in test images with clutter and objects appearing in close proximity. The Bayesian approach for data fusion, which maintains a distribution over multiple object and viewpoint hypotheses, was developed and shown to work well in this multiple object recognition scenario.

Systems $2, 3$ and 4 used SIFT, a local interest detector and descriptor, as input to the active vision methods for single and multiple object recognition. This may not always be the optimal object recognition scheme, especially if the objects in question contain little or no visual texture or differentiating features. We then investigated an alternative object representation scheme using Fourier descriptors for shape recognition (System 5). Fourier descriptors have been widely used for shape description and recognition. The Euclidean distances were calculated between the descriptors extracted for each shape. The minimum distance between descriptors indicated a match. These experiments were initially conducted in a non-active setting and achieved a recognition accuracy of 80%. System 5 was then extended to active shape recognition using mutual information. The aim was to determine the correct sequence of the objects/shapes present. We used mutual information as the active vision component to look for additional information about the object/shape in the sequence that it was most uncertain about. We showed that actively looking for information, even for this type of recognition setup, improved the overall recognition accuracy.

Through the novel active recognition systems implemented in this thesis, we have demonstrated through our experiments that including an active component to multiple view object recognition improves the efficiency and robustness of the system. Object recognition systems in a non-active setting have no tangible method for selecting the next best viewpoint and thus may process viewpoints which do not provide any relevant information. This leads to greater computational expense and possibly less accurate results if noise is present. The vocabulary tree data structure used in our active systems to weight the uniqueness of each feature is an original approach to selecting the next best viewpoint. The process can be completed offline and new objects can be added on the fly without recomputing the vocabulary tree. We have shown through various experiments that this is an effective technique and outperforms the use of random view selection. It achieves comparable results and in some cases better accuracy than the method proposed in [1] and methods based on mutual information.

The Bayesian framework allows us to integrate the extracted information in a principled manner when using features or Fourier descriptors. The framework of the systems presented here use the observer/Bayesian component to provide feedback to the system and thus new viewpoints are only processed when the system's belief in the identity of an object is below a predefined threshold. This allows fewer more relevant viewpoints to be processed. We not only presented an approach to single object recognition but also one for recognizing multiple objects in an active setting with excellent results. No active vision systems currently exist that recognize multiple objects in a test image. Most test images used in experiments contain a single object with no occlusions or clutter. In certain cases background information is introduced, but the object to be recognized still occurs in the centre

of the image with no occlusions. Our test images contain objects to be recognized occurring in substantial clutter with occlusions. Our method was still able to achieve high recognition accuracies despite the challenging dataset. We also introduced an active object recognition system using Fourier descriptors and mutual information to determine the correct sequence of a set of shapes. In this experimental setting we show that adding an active vision component improves the initial recognition results. Actively looking for information can improve the results of computer vision tasks and enables systems, especially mobile platforms, to process fewer viewpoints while investigating an environment to complete tasks closer to real time.

Novel contributions The novel contributions that were presented in this thesis are:

- An 3D active vision object recognition framework consisting of an automatic viewpoint selector and observer component. Methods in either component can be altered or completely changed without affecting the other component.
- An automatic viewpoint selector which uses local interest points extracted using SIFT as input to a vocabulary tree data structure. The vocabulary tree assigns a weighting to each feature based on its uniqueness given the current dataset. For each viewpoint, a score calculated by summing the weights of all the features that appears in a specific viewpoint. The higher the score, the more unique the viewpoint.
- The Bayesian framework which update the system's belief in the identity of an object. Statistics generated from the vocabulary tree are used as input to this framework.
- In this framework new images are only captured when the system's belief in the identity of an object is below a pre-defined threshold. This reduces the computational expense of the system as unnecessary viewpoints are not captured or processed. This framework has been shown to outperform randomly selecting the next best viewpoint and a state-of-the-art system presented by [1].
- The probabilistic models designed and implemented for single object recognition and single object and pose recognition. The binary and occlusion models incorporate the Hough transform, which enforces geometric constraints and produces excellent recognition results. The occlusion models, which also takes into account object and background distributions and occlusions, is able to correctly recognize all objects in this challenging dataset.
- A probabilistic framework for actively recognizing multiple objects as well as their poses in a test image. The next best viewpoint selector and observer components are extended for the multiple object scenario. This framework is able to keep track of multiple objects and their poses in this active setting and produces outstanding results.
- An active 2D shape recognition system was implemented using Fourier descriptors and mutual information. The aim was to recognize the correct sequence of 10 visually similar shapes.

The Fourier descriptors extracted were used as input to multinomial and Gaussian distributions. Mutual information was used by the system to actively look for information about the position in the sequence it was most uncertain about. The Gaussian distribution using mutual information is able to correctly the sequence.

- A dataset of 20 objects which includes visually similar objects. Training images were captured at every 20° intervals against a plain background on a turntable using a static Prosilica GE1900C camera. Each object consists 18 training images. Testing images were captured with objects appearing in cluttered environments with occlusion. These images were used for both single object and multiple object recognition.

6.1 FUTURE WORK

Adding an active vision component to a mobile platform and manipulator provides a mechanism to look for information in an environment. Future work may include estimating the time taken for a platform to move to a required viewpoint and using this as an additional criterion for selecting the next viewpoint. In terms of the features used in our statistical models, we would like to investigate using additional statistics as well as more complex object models such as texture and shape together. Further, our results suggest that refining our overall probabilistic models to accurately reflect the assumptions of the test data can provide performance gains in a Bayesian active vision setting. A slight caveat here is that we must have access to sufficient training/validation data in order to accurately estimate such models to see significant effects. Possibilities for future work along these lines include investigation of probabilistic active vision models that incorporate knowledge of correlations between objects (which objects are seen often/seldom together) and enhanced occlusion reasoning.

REFERENCES

[1] G. Kootstra, Y. Jelmer, and B. De Boer, "Active exploration and keypoint clustering for object recognition," in *IEEE International Conference on Robotics and Automation*, 2008.

[2] A. Zisserman, D. Forsyth, J. Mundy, C. Rothwell, J. Liu, and N. Pillow, "3D object recognition using invariance," in *Artificial Intelligence*, 1995, pp. 239–288.

[3] D. G. Lowe, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision*, 1999, pp. 1150–1157.

[4] A. Collet, D. Berenson, S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," in *International Conference on Robotics and Automation*, 2009.

[5] A. Selinger and R. C. Nelson, "Appearance-based object recognition using multiple views," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2001, vol. 1, p. 905.

[6] S. Roy, S. Chaudhury, and S. Banerjee, "Active recognition through next view planning: a survey," in *Pattern Recognition*, 2004, vol. 37, pp. 426–446.

[7] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, pp. 996–1005, 1988.

[8] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Simultaneous object recognition and segmentation from single or multiple views," in *International Journal of Computer Vision*, 2006, vol. 67, pp. 159–188.

[9] H. Bay, T. Tuytelaars, and L.Van Gool, "SURF: Speeded up robust features," *European Conference on Computer Vision*, pp. 404–417, 2006.

[10] I. Gordon and D. G. Lowe, "What and where: 3D object recognition with accurate pose," in *Toward Category-Level Object Recognition*, 2006, vol. 4170, pp. 67–82.

[11] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce., "3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *International Journal of Computer Vision*, pp. 231–259, 2006.

[12] D. G. Lowe, "Local feature view clustering for 3D object recognition," in *International Conference on Computer Vision and Pattern Recognition*, 2001, pp. 682–688.

[13] D. G. Lowe, "Distinctive image features from scale invariant keypoints," in *International Journal of Computer Vision*, 2004, vol. 60 of *91-110*.

[14] C. Schmid and R. Mohr, "Local greyvalue invariants for image retrieval," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997, pp. 530–534.

[15] S. Se, D. Lowe, and J. Little, "Local and global localization for mobile robots using visual landmarks," in *International Conference on Intelligent Robots and Systems*, 2000.

[16] C. Wang and K. Wang, "Hand posture recognition using Adaboost with SIFT for human robot interaction," in *Recent progress in robotics: viable robotic service to human*, pp. 317–329. Springer Berlin Heidelberg, 2008.

[17] P. V. C. Hough, "Method and means for recognizing complex patterns," Google Patents, December 1962, US Patent 3069654.

[18] Z. Jia, "Active view selection for object and pose recognition," in *International Conference on Computer Vision, 3D Object Recognition Workshop*, 2009.

[19] S. A. Hutchinson and A. C. Kak, "Planning sensing strategies in a robot work cell with multi-sensor capabilities," in *IEEE Transactions on Robotics and Automation*, 1989, vol. 6, pp. 765–783.

[20] S. J. Dickinson, H. I. Christensen, J. Tsotsos, and G. Olofsson, "Active object recognition integrating attention and view point control," in *Computer Vision and Image Understanding*, 1997, vol. 3, pp. 239–260.

[21] B. Schiele and J. L. Crowley, "Recognition without correspondence using multidimensional receptive field histograms," *International Journal of Computer Vision*, pp. 31–50, 2000.

[22] M. Swain and D. Ballard, "Color indexing," *International Journal of Computer Vision*, pp. 11–32, 1991.

[23] H. Murase and S. K. Nayar, "Visual learning and recognition of 3D objects from appearance," *International Journal of Computer Vision*, pp. 5–24, 1995.

[24] S.D. Roy, S. Chaudhury, and S. Banerjee, "Isolated 3D object recognition through next view planning," in *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, 2000, vol. 1, pp. 67–76.

[25] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *International Conference on Computer Vision and Pattern Recognition*, 2006, vol. 5.

[26] E. N. Malamas, "A survey on industrial vision systems, applications and tools," *Image and Vision Computing*, pp. 171–188, 2003.

[27] D. Kim and R. Nevatia, "Recognition and localization of generic objects for indoor navigation using functionality," *Image and Vision Computing*, pp. 729–743, 1998.

[28] G. L. Foresti, "Object recognition and tracking for remote video surveillance," in *IEEE Transactions on Circuits and Systems for Video Technology*, 1999, pp. 1045–1062.

[29] J. W. H. Tangelder and R. C. Veltkamp, "A survey of content based 3D shape retrieval methods," in *Multimedia tools and applications*, 2008, pp. 441–471.

[30] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, 1988.

[31] J. Canny, "A computational approach to edge detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986, pp. 679–698.

[32] R. Schnabel, W. Roland, and K. Reinhard, "Efficient RANSAC for point cloud shape detection," in *Computer graphics forum*. Blackwell Publishing Ltd, 2007.

[33] K. D. Gremban and K. Ikeuchi, "Planning multiple observations for object recognition," in *International Journal of Computer Vision*, 1994, pp. 137–172.

[34] F. G. Callari and F. P. Ferrie, "Active object recognition: Looking for differences," in *International Journal of Computer Vision*, 2001, pp. 189–204.

[35] S. Abbasi and F. Mokhtarian, "Automatic view selection in multi-view object recognition," in *International Conference on Pattern Recognition*, 2000, vol. 1, pp. 13–16.

[36] A. Leonardis and H. Bischof, "Robust recognition using eigenimages," in *Computer Vision and Image Understanding*, 2000, pp. 99–118.

[37] L. Juan and G. Oubong, "A comparison of SIFT, PCA-SIFT and SURF," *International Journal of Image Processing*, pp. 143–152, 2009.

[38] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

[39] P. Moreels and P. Perona, "Evaluation of feature detectors and descriptors based on 3D objects," in *International Journal of Computer Vision*, 2007, vol. 73, pp. 263–284.

[40] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *European Conference on Computer Vision*, 2002, pp. 128–142.

[41] J. Wu, Z. Cui, V. S. Sheng, P. Zhao, D. Su, and S. Gong, "A comparative study of sift and its variants," *Measurement Science Review*, 2013.

[42] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *International Journal of Computer Vision*, pp. 333–356, 1988.

[43] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *The International Journal of Robotics Research*, pp. 1343–1377, 2011.

[44] E. Sommerlade and I. Reid, "Information-theoretic active scene exploration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[45] H. Yu and M. Bennamoun, "A phase correlation approach to active vision," in *Computer Analysis of Images and Patterns*, pp. 57–64. Springer, 2005.

[46] H. Chen and L. Youfu, "Data fusion for three-dimensional tracking using particle techniques," in *Optical Engineering 47.1*, 2008.

[47] H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz, "Appearance-based active object recognition," in *Image and Vision Computing*, 2000, pp. 715–727.

[48] F. Deinzer, J. Denzler, and H. Niemann, "Viewpoint selection - planning optimal sequences of views for object recognition," in *International Conference on Computer Vision*. 2003, pp. 65–73, Springer.

[49] F. Deinzer, J. Denzler, and H. Neimann, "On fusion of multiple views for active object recognition," *Pattern Recognition*, pp. 239–245, 2001.

[50] T. Arbel and F. P. Ferrie, "Viewpoint selection by navigation through entropy maps," in *Journal of Computer Vision*, 1999, pp. 248–254.

[51] N. Govender, J. Claassens, P. Torr, and J. Warrell, "Active object recognition using vocabulary trees," in *IEEE Workshop on Robot Vision*, 2013.

[52] F. G. Callari and F. P. Ferrie, "Active recognition: Using uncertainty to reduce ambiguity," in *International Conference on Pattern Recognition*, 1996, pp. 925–929.

[53] D. Pangercic, "Fast and robust object detection in household environments using vocabulary trees with SIFT descriptors," in *Conference on Intelligent Robots and Systems*, 2011.

[54] N. Atanasov, B. Sankaran, J. Le Ny, T. Koletschka, G. J. Pappas, and K. Daniilidis, "Hypothesis testing framework for active object detection," in *International Conference on Robotics and Automation*, 2013, pp. 4216–4222.

[55] N. Henze, T. Schinke, and S. Boll, "What is that? object recognition from natural features on a mobile phone," in *Workshop on Mobile Interaction with the Real World*, 2009.

[56] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–7.

[57] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. X. Han, "Contextual weighting for vocabulary tree based image retrieval," in *International Conference on Computer Vision*, 2011, pp. 209 – 216.

[58] C. Yuan, X. Li, W. Hu, and H. Wang, "Human action recognition using pyramid vocabulary tree," in *Asian Conference on Computer Vision*, 2009, pp. 527–537.

[59] D. Sabatta, D. Scaramuzza, and R. Siegwart, "Improved appearance-based matching in similar and dynamic environments using a vocabulary tree," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 1008 – 1013.

[60] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision*, 2000, vol. 2, p. 1470.

[61] I. H. Witten, A. Moffat, and T. C. Bell, *Managing gigabytes: compressing and indexing documents and images*, Morgan Kaufmann Publishers, 1999.

[62] H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz, "Active object recognition in parametric eigenspace," in *British Machine Vision Conference*, 1998, pp. 629–638.

[63] H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz, "A comparision of probabilistic, possibilistic and evidence theoretic fusion schemes for active object recognition," in *Computing*, 1999, vol. 62.4, pp. 293–319.

[64] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. Vol. 15, pp. pp. 1115, 1972.

[65] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," in *Pattern recognition*, 1981, pp. 111–122.

[66] V. F. Leavers, *Shape detection in computer vision using the Hough transform*, Ph.D. thesis, Department of Physics, Kings College London, 1992.

[67] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool, "Hough transform and 3D SURF for robust three dimensional classification," in *European Conference on Computer Vision*, 2010.

[68] A. Yao, J. Gall, and L. Van Gool, "A Hough transform-based voting framework for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2061–2068.

[69] M. Godec, P. M. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," in *Computer Vision and Image Understanding*, 2013, pp. 1245–1256.

[70] S. N. Srihari and V. Govindaraju, "Analysis of textual images using the Hough transform," *Machine Vision and Applications*, pp. 141–153, 1989.

[71] J. Denzler and C. M. Brown, "Information theoretic sensor data selection for active object recognition and state estimation," in *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 2002.

[72] A. Krause, A. Singh, and C.Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, 2008.

[73] S. Yu, B. Krishnapuram, R. Rosales, and R. Rao, "Active sensing," in *IEEE International Conference on Artificial Intelligence and Statistics*, 2009, pp. 639 – 646.

[74] C. E. Shannon and W. Weaver, *The matematical theory of communication*, Univresity of Illinois Press, 1959.

[75] E. Persoon and K. Fu, "Shape discrimination using Fourier descriptors," in *IEEE Transactions On Systems, Man and Cybernetics*, 1977, pp. 170–179.

[76] H. Kauppinen, T. Seppanen, and M. Pietikainen, "An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995, vol. 17, pp. 201–207.

[77] C. T. Zahn and R. Z. Roskies, "Fourier descriptors for plane closed curves," in *IEEE Transactions on Computers*, 1972, vol. 21, pp. 269–281.

[78] R. D. D. Leon and L. E. Sucar, "Human silhouette recognition with Fourier descriptors," in *International Conference on Pattern Recognition*, 2000, pp. 709–712.

[79] C. H. Teh and R. T. Chin, "On image analysis by the methods of moments," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1988, vol. 10.

[80] G. Taubin, *Recognition and positioning of rigid objects using algebraic and moment invariants*, Ph.D. thesis, Brown University, December 1990.

[81] Q. M. Tieng and W. W. Boles, "Recognition of 2D object contours using wavelet transform zero crossing representation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.

[82] H. S. Yang, S. U. Lee, and K. M. Lee, "Recognition of 2D contours using starting-point-independent wavelet coefficient matching," in *Journal of Visual Communication and Image Representation*, 1998, vol. 9, pp. 171–181.

[83] D. Zhang and G. Lu, "Content-based shape retrieval using different shape descriptors: A comparative study," in *IEEE*, 2001.

[84] D. Zhang and G. Lu, "A comparative study on shape retrieval using Fourier descriptors with different shape signatures," in *Intelligent Multimedia and Distance Education*, 2001, vol. 14, pp. 1–9.

[85] G. Lu and A. Sajjanahr, "Region-based shape representation and similarity measure suitable for content-based image retrieval," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999, pp. 164–174.

[86] P. J. Van Otterloo, *A contour oriented approach to shape analysis*, Prentice Hall, 1991.

[87] T. W. Rauber, "Two-dimensional shape description," Tech. Rep., University Nova de Lisboa, Portugal, 1994.

[88] R. C. Gonzalez and R. E. Woods, *Digital image processing*, Prentice Hall, 2002.

[89] J. M. Glover, "Probabilistic Procrustean models for shape recognition with an application to robotic grasping," M.S. thesis, MIT, 2008.

[90] C. B. Akgul, B. Sankur, Y. Yemez, and F. Schmitt, "3D model retrieval using probability density-based shape descriptors," in *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 2009, vol. 31.

[91] D. Macrini, C. Whiten, R. Laganiere, and M. Greenspan, "Probabilistic shape parsing for view-based object recognition," in *International Conference on Pattern Recognition*, 2012.

[92] B. Krishnapuram, C. M. Bishop, and M. Szummer, "Generative models and Bayesian model comparison for shape recognition," *International Workshop on Frontiers in Handwriting Recognition*, 2004.