The Role of Fixation and Visual Attention in Object Recognition

by

Aparna Lakshmi Ratan

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 1995

Signature of Author ....
Department of Electrical Engineering and Computer Science
Jan 20, 1995

Certified by ....
Eric Grimson
Associate Professor of Computer Science
Thesis Supervisor

Accepted by ....
F. R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

The Role of Fixation and Visual Attention in Object Recognition

by

Aparna Lakshmi Ratan

Submitted to the Department of Electrical Engineering and Computer Science
on Jan 20, 1995, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

This research is a study of the role of fixation and visual attention in object recognition. In this project, we built an active vision system which can recognize a target object in a cluttered scene efficiently and reliably. Our system integrates visual cues like color and stereo to perform figure/ground separation, yielding candidate regions on which to focus attention. Within each image region, we use stereo to extract features that lie within a narrow disparity range about the fixation position. These selected features are then used as input to an Alignment-style recognition system. We show that visual attention and fixation significantly reduce the complexity and the false identifications in model-based recognition using Alignment methods. We also demonstrate that stereo can be used effectively as a figure/ground separator without the need for accurate camera calibration.

Thesis Supervisor: Eric Grimson
Title: Associate Professor of Computer Science

Acknowledgments

I would like to thank my thesis supervisor Prof. Eric Grimson for his support and advice throughout this project. I also wish to thank Greg Klanderman for the use of his image processing library, Patrick O'Donnell for his help with the head-eye system, Ian Horswill and Pamela Lipson for their feedback on the initial drafts of the thesis and Kenji Nagao for his helpful comments and discussions.
Finally, thanks to my parents and to my sisters Maithreyi and Aishwarya for all their enthusiasm, support and encouragement.

Contents

Chapter 1

Introduction

Model-based object recognition involves finding an object in a scene given a stored description of the object. Most approaches to model-based object recognition extract features like points and lines from the model and the data and identify pairings between model and data features that yield a consistent transformation of the model object into image coordinates. If we have a cluttered scene as in Figure 1-1 and have no indication of where the object is in the scene, then we have to try all possible pairings of model and image features in order to solve for the correct transformation that aligns the model features with the image features. The large number of pairings makes this search combinatorially explosive. Most of the search is unnecessary and irrelevant since it involves trying pairings of features from different objects in the scene that couldn't yield the correct transformation.

There have been several methods suggested in the literature to reduce the unnecessary search involved in recognition. We now discuss the effects of clutter on the performance of some of these recognition methods. Methods that explore a tree of interpretations using constrained search techniques to find consistent interpretations of the data relative to the model (e.g. [18]) have an exponential expected case complexity in the presence of scene clutter. If the clutter can be made relatively small, however, the expected search complexity is reduced to a low order polynomial [18]. There are other recognition methods known as minimal alignment methods (e.g. [29], [64]), which find a small number of corresponding features between model and data and use the associated transformation to align the model with the data for verification. These methods have worst case complexity that is polynomial in the number of model and data features, an improvement over the constrained search methods mentioned above. The complexity is still a function of scene clutter, however, so in practice clutter can slow down these methods significantly. In both cases, scene clutter also contributes to the number of false alarms that must be handled [23].

All the studies (e.g. [18]) on the search space complexity and the effects of scene clutter on it suggest that we need a way to reduce the number of features in the scene and restrict the search to relevant data subsets in the scene while avoiding extraneous information provided by clutter. For example, in the Figure 1-1(b), if we could find the area in the scene that contains the object (Figure 1-2(a)), then the number of features to be tried in the scene reduces considerably (from 500 to 20 in this case). If we use minimal alignment for recognition ([29]) then we need three corresponding points between the model and image to compute the transformation in order to align the model with the data. Given a set of model and data features, we have to try all possible triples of model and data points and verify the associated alignments. In the example we have 20 model features, 500 features in Figure 1-1(b) and around 20 features in Figure 1-2(a). The number of alignments to be tried between the model and image in Figure 1-1(b) is $500^3 * 20^3$ or $(1 * 10^{12})$ and the number of alignments between the model and image in Figure 1-2(a) is $20^3 * 20^3$ or $(6 * 10^7)$. Also, by focusing on features coming from a single object (with the properties of the object we are looking for), we reduce the number of false positives.

Keeping these issues in mind, it is convenient to divide object recognition into three tasks which serve to illustrate the different complexity issues that arise in recognition. These tasks are selection, indexing and correspondence:

- Selection : Selection is the problem of identifying regions in the image that are more likely to come from a single object.

- Indexing: Given a library of object models, indexing refers to the task of determining which model corresponds to the selected subset of the image.

- Correspondence: Correspondence refers to finding a match between individual model features and image features.

Previous work suggests that selection is one of the key problems in recognition ([17], [18]) since it reduces the expected complexity of recognition and keeps the false positives under control. Grimson shows in [17] that the expected search complexity (using a method called constrained search) can be reduced from exponential to a low order polynomial when all the edge features are known to come from a single object. Selection can be used to improve the performance of other recognition methods (e.g. [30] among others) as well.

The aim of this project is to investigate the role of visual attention and fixation in the selection phase of object recognition. Visual attention refers to selecting out portions of the scene on which to focus the resources of visual processing. Fixation is the mechanical movement of the eyes such that both eyes are pointed to and accommodated at the same point in space. In this thesis, we present a method to reduce the complexity and control the false identifications in model-based recognition by using several simple visual cues in conjunction to focus attention on and fixate selected regions in the scene that are likely to contain the target object.

We show that by

1. using a combination of cues (color and stereo in this case) to perform figure/ground separation into regions on which to focus attention,

2. using stereo to extract features that lie within a narrow disparity range about the fixation position within the salient regions, and

3. using visual attention to control these cues

we can reduce the search involved in the recognition process and find target objects efficiently (with a marked reduction in the complexity of the search space) and reliably by improving the correctness of the solution (i.e. reducing the number of false positives and false negatives).

Chapter two describes the solution proposed in general. Chapter three describes the highlighting of target regions using color. Chapter four describes the process of zeroing in on target regions using stereo and the processing at finer level of resolution to give the final set of selected features that are fed into the recognition engine. Chapter five describes the alignment and verification steps. Chapter six explains how the system was tested and shows results. Chapter seven includes the discussion and conclusions.

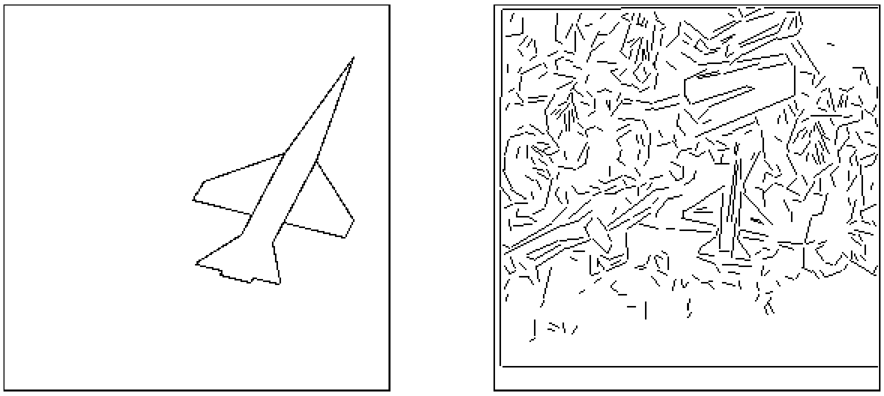

Figure 1-1: (a) The model object (b) Cluttered scene

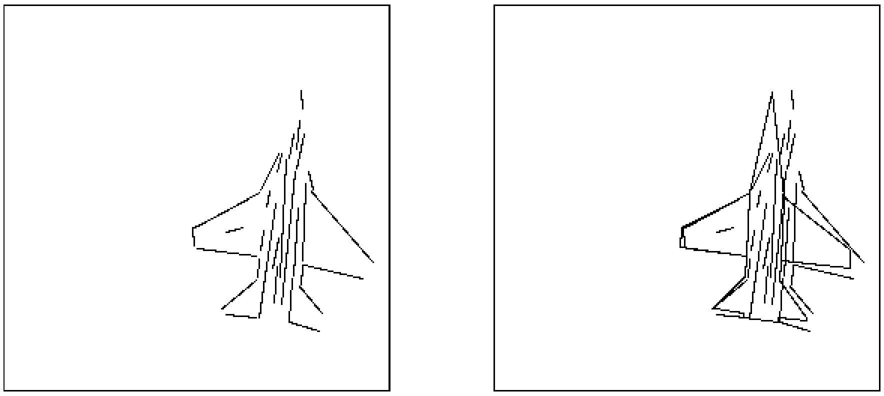

Figure 1-2: (a) Selected region from 1-1(b), (b) Model aligned with object

1.1 Difficulty

Humans don't have difficulties in recognizing partially occluded objects efficiently and reliably in cluttered scenes but the same task is challenging for computers. If we have an ideal situation with perfect image data of an object isolated from the background, then there are many techniques (e.g. [20], [29] among others) for recognizing the object and its pose. In most normal scenes, however, there are additional problems introduced when only a portion of the object is visible (occlusion) and when most of the data in the image does not come from the target object (spurious data due to scene clutter). Figure 1-1(b) is an example of a cluttered scene where most of the data in the image does not come from the object given in Figure 1-1(a). Thus, the recognition system needs to identify the object in the cluttered scene and match the subset of the data belonging to the object with the model to determine the pose of the object.

The recognition process is further complicated by the presence of noise in the sensory data. Noisy sensor data often means that the features extracted from the image are not perfect. For example, in Figure 1-2(a), some of the features extracted from the image are fragmented while others are missing. This implies that we cannot compare attributes like angles and lengths of model and data features exactly. The problem of extracting good features from the image is also affected by the lighting conditions in the scene. If the lighting conditions were carefully controlled, such as in factory environments, then we can get good features reliably but in most common scenes the illumination conditions are not known and specularities and shadowing effects make the task of extracting good features difficult. Thus, a good recognition system has to be able to work reliably with noisy data, under varying illumination conditions in day to day scenes without being affected by occlusion and clutter due to spurious data.

1.2 Motivation and Related Work

Solving object recognition directly for computers is too hard a problem. However, effective segmentation makes a significant difference to the complexity of later stages of recognition. We are thus interested in approaching the object recognition problem using efficient segmentation techniques to make it feasible. The approach we are taking, using visual attention to direct the eye to focus on the object of interest, suggests a way to achieve a fast and restricted type of scene understanding. If the object is present in the scene, then focusing attention on the visual features that describe the object helps isolate a region in the image that could contain the object. This kind of selection greatly reduces the search in the correspondence stage where the image data is matched with model data using schemes like Alignment [29] or Linear Combination of Views [64].

1.2.1 Visual Attention

While there is enough evidence to prove that object selection is a complex task for a machine to perform, it is interesting to note that humans seem to have no difficulty in selecting out parts of a scene that contain relevant or interesting information with regard to the task being performed. This ability of humans to select out relevant parts of a scene relating to a particular task is known as visual attention. This observation has motivated the use of visual attention in object recognition. Hurlbert and Poggio in [28] suggest how the concept of visual attention can be used to reduce the combinatorial search in recognition. There have been a number of computational models [59], [15], [34] of attention proposed in the literature that use this idea. All these models are based on the model of visual attention proposed by Treisman in [62] as a result of psychophysical experiments. The Treisman model consists of several low level feature maps which could be combined using a selection filter. The computational models of attention ([34], [15] and [59]) mentioned above use different strategies to combine and control the feature maps. In Koch and Ullman's model, the feature maps are combined using a "Winner Take All" mechanism where the network locates the region that differs the most from its neighbors with respect to some property. All the "conspicuity" values are combined into a global saliency map and the network finds the maximum conspicuity value in the global map. The most conspicuous location is where attention is focussed. Clark and Ferrier [15] combined the feature maps by assigning a weight to each feature map and combining them using a linear combination of these weighted features. Syeda-Mahmood [59] uses an arbiter module that combines the feature maps and maintains separate saliency maps until the arbiter stage. The idea of using feature maps to represent low level processing of information can be traced back to Marr [37] where he uses the primal sketch to expose low level image features and Triesman [62] in her model of attention among others. Thus, we see that visual attention gives a convenient way to combine and integrate information provided by several visual cues in order to perform selection.

1.2.2 Active Vision

Fixation plays an important role in biological and machine vision, especially in binocular stereo. As Ballard mentions in [6], the human eye is different from cameras in that it has much better resolution in a small region around the optical axis. This region is called the fovea. The resolution over the fovea is much better than in the periphery. An interesting feature in the design of the human visual system is the simultaneous representation of a large field of view and local high acuity. The human eye has the ability to quickly move the fovea (saccade) to different spatial locations. Another feature of the human visual system is that the complete visual field is not stabilized. The region that is stabilized lies near the point of fixation which is defined as the intersection of the two optical axes. Thus, we see that humans make use of an elaborate gaze control system with the ability to foveate a target.

An active (animate) vision framework takes advantage of fixation and keeps the fovea over a given spatial target (gaze control), changes focus and changes point of view while investigating a scene. The "active vision" paradigm has been discussed in papers such as [1], [6],[5] among others. Most of the work in the field of active vision has been concerned with low level tasks like gaze control [1], [52], [51], [15], [16]. The importance of camera movements and adjustment of imaging parameters in stereo vision has been investigated by Ballard in [7], Abbot and Ahuja in [2] and Bajcsy in [4]. Knowledge of verging geometry has been used by Krotkov et al. [35] to address calibration issues. A system that integrates information from focus, vergence angle, and stereo disparity over multiple fixations to get accurate depth estimates was proposed by Abbot and Ahuja [2]. Vergence control has been used by Olson [50] to simplify stereopsis by limiting the disparity range to provide relative depth information over single fixations to be used in building qualitative descriptions for recognition. Controlled eye movements have been used to obtain geometric information for camera calibration [10]. All these applications of active vision use the ability to control the position of the cameras in order to obtain additional visual constraints to simplify various tasks.

1.2.3 Active-Attentive Vision

We would like to use the active vision framework to perform higher level tasks such as model-based object recognition. Recognition can be more robust using active and attentive vision since we have the ability to obtain multiple views and can ignore irrelevant information. Ferrier and Clark in [15] suggest a form of "active-attentive" vision to focus attention on parts of the scene that is important to the task at hand. In their paper, they give a framework for combining feature maps using active vision techniques. However they focused more on the low level issues of building the head and gaze control. Bober et al. [8] actively control the sensor based on the goal to be accomplished. Their system architecture divides a visual task into the categories of camera control, focus of attention control and selection of a suitable recognition strategy and they stress the close interaction between the goal, sensor control and the visual task. Horswill in [26] uses a task based approach to perform a higher level task. He exploits knowledge about the environment to simplify visual and motor processing in an agent that performs the specific tasks of navigation and place recognition.

1.3 Our Approach

Our system is similar in spirit to Ferrier and Clark [15] and Bober et al. [8] in that it investigates the role of fixation and visual attention to perform the higher level task of object recognition. We illustrate the effectiveness of using an active-attentive control mechanism to do efficient figure/ground separation in the domain of model-based object recognition in cluttered scenes using alignment style recognition techniques.

We use the visual cues of color and stereo in conjunction to show that by combining different cues, we don't need the individual cues to be very accurate. We demonstrate this as follows:

- we show that rough color measures can be used to roughly segment the data without the need for a complex color constancy model.

- We also show that stereo can be used effectively as a figure/ground separator without calculating absolute depths [24]. Thus, we don't need accurate camera calibration. If we consider selection to be the important part of recognition, and do 3D recognition from 2D by using techniques like linear combination of views [64], then we don't need accurate 3D data for correspondence. This means that we can use relative depths to get feature subsets in the same depth range and avoid extracting absolute depth information entirely. This is useful and interesting for several reasons [24].

1. There has been some physiological evidence to show that the human system does 3D recognition from 2D views.
2. As shown by Grimson in [24] and Olson in [50], small inaccuracies in camera parameters can lead to large errors in depth.

If we are interested in finding roughly contiguous 3D regions then it is useful to fixate on a target and search for matching features within some disparity range about that point. Thus, matching features yield a candidate object. This is similar to the working of the human stereo system where matching disparities are restricted to a narrow band about the fixation point (Panum's limit).

The project uses a head-eye system which can pan and tilt. The head initially scans the room and finds regions that could potentially contain the target object using visual cues like shape, color, texture etc. Once it has found candidate regions in the image, it investigates these regions in detail and feeds the selected output into a recognition engine.

The algorithm uses a variant of the Marr-Poggio-Grimson stereo algorithm, which uses a coarse to fine control strategy. The initial segmentation layer uses object properties like color and texture to mask the edges that are in regions of interest in the left and right images. The stereo algorithm is run on the reduced set of segments. Filtering using cues like color and texture reduce the number of features to be matched by the stereo matcher considerably. The stereo algorithm finds a focal edge in one image that has a unique match in the other image and uses this edge to zoom in and fixate the eyes on the target. The second layer runs the stereo algorithm on a pair of high resolution images to get segments that match in a narrow disparity band around the target edge. Since the eyes are fixated on the target, most of the matched segments come from the target object. The resulting matched segments are used as input to the recognition engine.

Our system has the following properties:

- It is simple and easy to use.
- It reduces the complexity of the search space for recognition considerably by focusing attention on regions in the scene that contain the target object, thus selecting regions containing the object before doing alignment.
- The system works efficiently (by reducing the search space at the time of recognition) and reliably (with few false positives and false negatives) in cluttered scenes.
- The system combines rough color measures (without a complex color constancy model) with stereo measures (without the need for accurate camera calibration) to perform selection.

The proposed solution is discussed in greater detail in Chapter 2.

1.3.1 Example describing the stages to solution:

The goal of the project is to find a target object (e.g. the plane in Figure 1-3) in a room by analyzing pictures taken by a pair of cameras on a platform that can pan and tilt. The algorithm proceeds by taking an image of the scene (Figure 1-4(a)) and finding the line-segment approximations to the edges detected in the image (Figure 1-4(b)). These line segments are the features that we use for recognition. As we discussed earlier in this chapter, we want to find the model object in this scene by using Alignment-style recognition techniques (e.g. [29]) where we find 3 corresponding points between the model (Figure 1-3) and the image (Figure 1-4(b)) to compute the transformation that aligns the model with a hypothesized instance of the object in the image and then verify that hypothesis by comparing the transformed model with the image data. The central problem in this method of hypothesis construction is finding corresponding sets of model and image features. In our example, where there are roughly 500 features in the scene (Figure 1-4(b)) and 20 model features (Figure 1-3), the number of alignments to be tried is on the order of 10^{12}. We also notice that there is considerable spurious data (data that does not belong to the object) in Figure 1-4(b) which contributes to false identifications. We have implemented a system that reduces the number of alignments to be tried during recognition significantly and controls the false matches by focusing attention on relevant data subsets using the visual cues of color and stereo.

Once we get all the features in the image (Figure 1-4(b)), we use the color of the target object to select out regions in the image that could contain the target object and retain only those features from Figure 1-4(b) that fall within these selected regions. The color algorithm is discussed in Chapter 3. The features remaining after the color filter has been applied are shown in Figure 1-5. The number of alignments to be tried at this stage is 10^9. Figure 1-5 gives us a set of regions on which to focus future resources since they are likely to contain the target object.

As we discussed earlier in this chapter, by using several simple cues in conjunction we don't need the individual cues to be very accurate and we can reduce the number of false identifications. We use stereo as a second visual cue in our system. The stereo algorithm is run over a pair of images containing the features that remain after the color filter. The stereo algorithm, which is discussed in greater detail in Chapter 4, isolates a distinctive edge (measured as a combination of length and intensity contrast) in the left image (Figure 1-5) with a unique match in the right image. This enables the cameras to fixate the edge and obtain a new set of images such that the region around the fixation point is examined at high resolution (in greater detail). Figure 1-6 gives the resulting features that are examined at finer resolution after the cameras have fixated some edge from Figure 1-5. At this stage, we notice that the target object in Figure 1-3 is included in the region that is being examined at finer resolution. The stereo algorithm is run again on the high resolution images to find matching features in a narrow disparity range about the fixation point. Since the cameras are fixated on the target object, most of the matched edges come from the target object as seen in Figure 1-7(a). These selected features in Figure 1-7(a) are fed into an Alignment-style recognition engine which is discussed in Chapter 5. The results of aligning the model in Figure 1-3 with the selected target object in Figure 1-7(a) are shown in Figure 1-7(b). The number of alignments that had to be tried using the features in Figure 1-7(a) are on the order of 10^7 which is a significant improvement over the 10^{12} alignments that had to be tried in Figure 1-4(b). Also, since the selected features come from the target object, we reduce the number of false identifications due to spurious data.

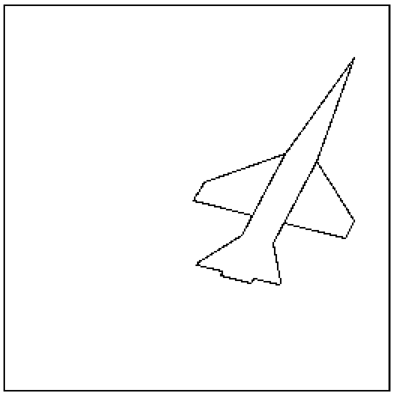

Figure 1-3: The geometric model of the object to be found. Model has 20 features.

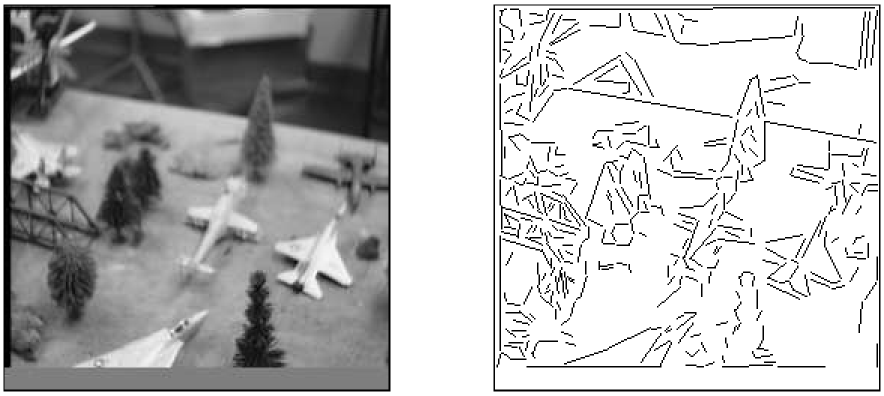

Figure 1-4: (a) Initial gray image. (b) Segments in initial image. Number of features = 500. Number of alignments = $20^3 * 500^3 = 10^{12}$

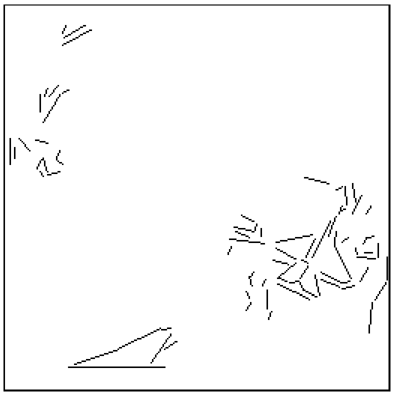

Figure 1-5: Regions to focus attention after color filter has been applied. Number of features = 70. Number of alignments = $70^3 * 20^3 = 10^9$

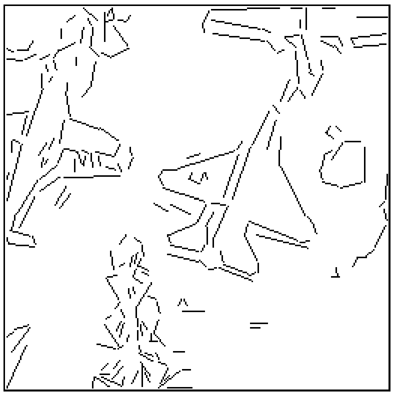

Figure 1-6: Foveated region after the stereo algorithm is run to determine where to fixate the eyes. Number of features = 300. Number of alignments = $300^3 * 20^3 = 10^{11}$. Note that we are looking at a region of interest (region that could contain the model object) from the initial scene in greater detail.

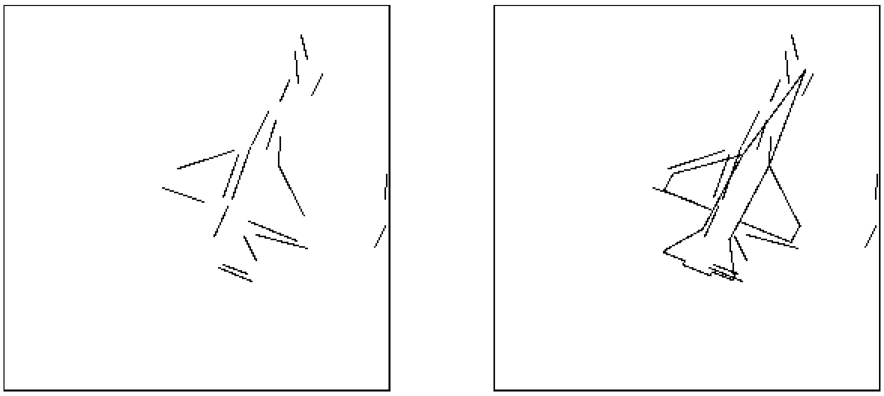

Figure 1-7: Selected dataset and results of aligning the model with the selected dataset. Number of features in the selected dataset = 25. Number of alignments $= 25^3 * 20^3 = 10^7$

Chapter 2

Solution proposed

A common approach to model-based object recognition is to hypothesize the pose of a known object in the image and then verify it to localize the object in the image. This involves finding correspondences between model and image features. If we have no information about the location of the object in the scene, then all pairings between model and image features have to be tried and the large number of pairings make the search for the correct set of corresponding features combinatorially explosive. Most of this search is useless, especially when pairings between different objects are tried. If a recognition system had information about data subsets that are likely to come from a single object, then the search for matching features can be restricted to relevant data subsets that are likely to lead to the correct solution, and false identifications caused by extraneous information due to scene clutter can be avoided. As we saw in Chapter 1, the problem of isolating regions belonging to a single object in an image is termed the selection (figure/ground separation) problem and has been recognized as a crucial problem in model-based recognition ([17], [18]). In this chapter, we will discuss a system that implements figure/ground separation by combining the following two themes.

- Merging multiple visual cues in order to achieve figure/ground separation.
- Using active vision metaphors to direct the figure/ground separation.

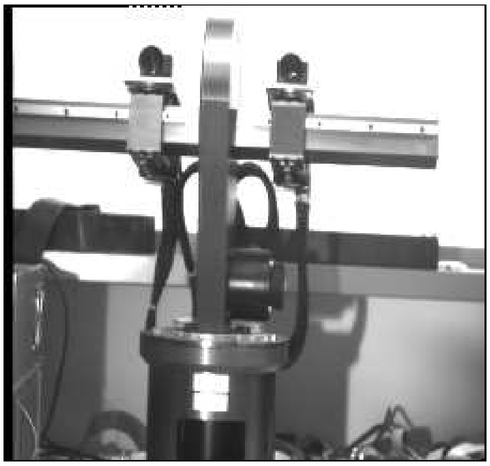

Figure 2-1: The head eye

We use the system to find a small target in a cluttered environment quickly by focusing its resources on regions in the image that are likely candidates to contain the object.

2.1 The Overall System

Our active attentive visual system consists of a two camera, eye-head system which can pan and tilt and which lets each camera verge (Figure 2-1). The system has a pan range of $\pm 75^o$, a tilt range of -80^o to 90^o and individual eye vergence range of $\pm 25^o$. Figure 2-2 illustrates the overall flow of control in the system, i.e. how various visual cues (e.g. color, texture etc.) can be integrated and controlled within an active vision framework. We describe the elements of the system in more detail below.

The goal of the system is to efficiently find a target object in a cluttered environment with minimal false positives. Scanning the entire field of view at high resolution is impractical, so we use a coarse-to-fine strategy whereby we use visual cues to quickly isolate regions of the image that are likely to contain the object. There are many cues that can be used for this purpose. They could be shape based cues like edges and 3D shape information from motion and stereo or they could be region based cues like color, texture, etc. Since most recognition techniques use shape based cues, we need to extract such shape based features from the scene before recognition.

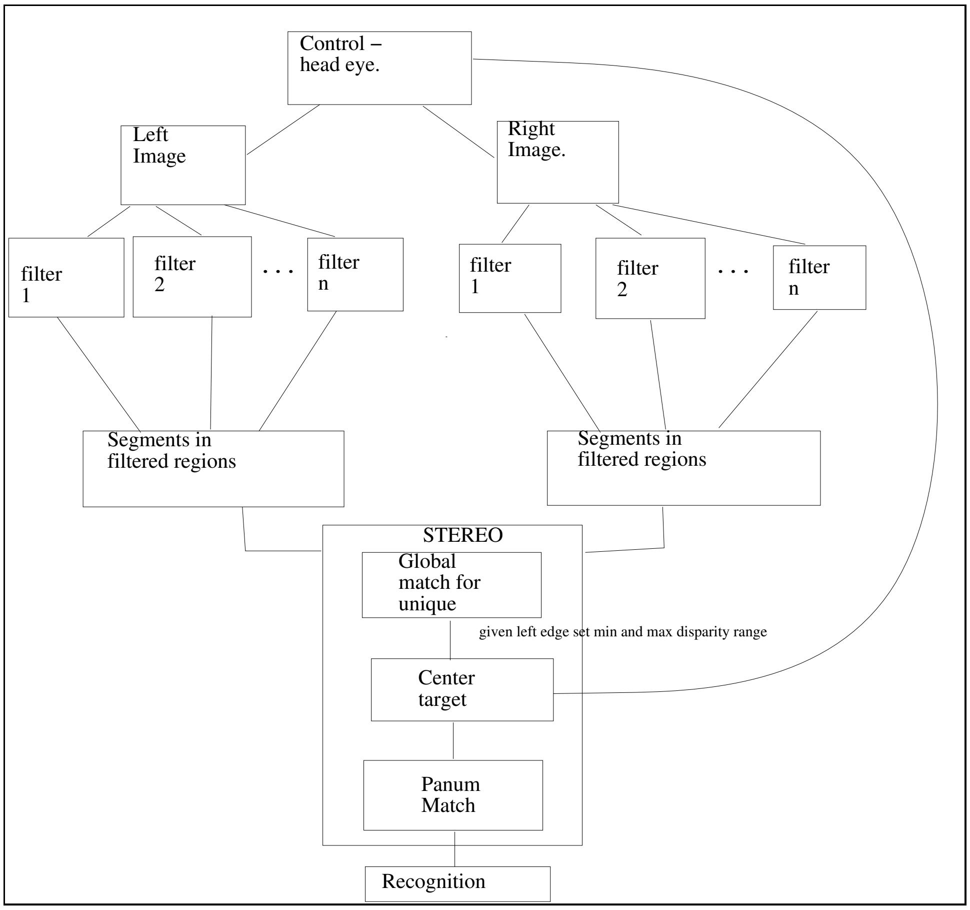

Figure 2-2: Overall flow of control in the system.

However, shape based cues are not sufficient to control the complexity of the problem when used on their own. Consider a case where we have $m = 50$ model features and $n = 500$ data features. If we use Alignment-style recognition techniques (e.g. [29]) where we find 3 corresponding points between the model and the image to compute the transformation that aligns the model with a hypothesized instance of the object in the image and then verify the hypothesis by comparing the transformed model with the image data, then the number of alignments that have to be tried is $O(m^3n^3)$ which gives us on the order of $1.5 * 10^{13}$ cases to be tried in this example. Thus, we see that we need more information to control the number of alignments that have to be tried. We can use information provided by the region based cues like color and texture to reduce the number of features that have to be tried. Region based cues are useful in selection (figure/ground separation) since they provide us with a method for roughly comparing properties of the model with the data, so that we can exclude extraneous information without ignoring any relevant target regions. In this system, we investigate the possibility of combining cues to direct the fixation of the eyes on candidate regions and analyze these regions at a finer level to select out target features that can be fed into a recognition engine. Another advantage in using multiple cues is that the individual cues can be inaccurate.

In our system, we use Alignment-style[1] recognition techniques to find targets in a large room. The room is scanned by pan and tilt movements of our eye-head system to select different viewing points. As shown in Figure 2-3, we use color information and stereo edge information to reduce the number of target features that have to be verified by the recognition system. Details on why we chose these cues and how we decided to combine them in an active vision framework are described in later chapters. The various stages in the working of our system are described below.

[1] Alignment-style recognition techniques ([29], [64]) find a small number of corresponding features between the model and the image to compute the transformation that aligns the model with a hypothesized instance of the object in the image and then verifies the hypothesis by comparing the transformed model with the image data.

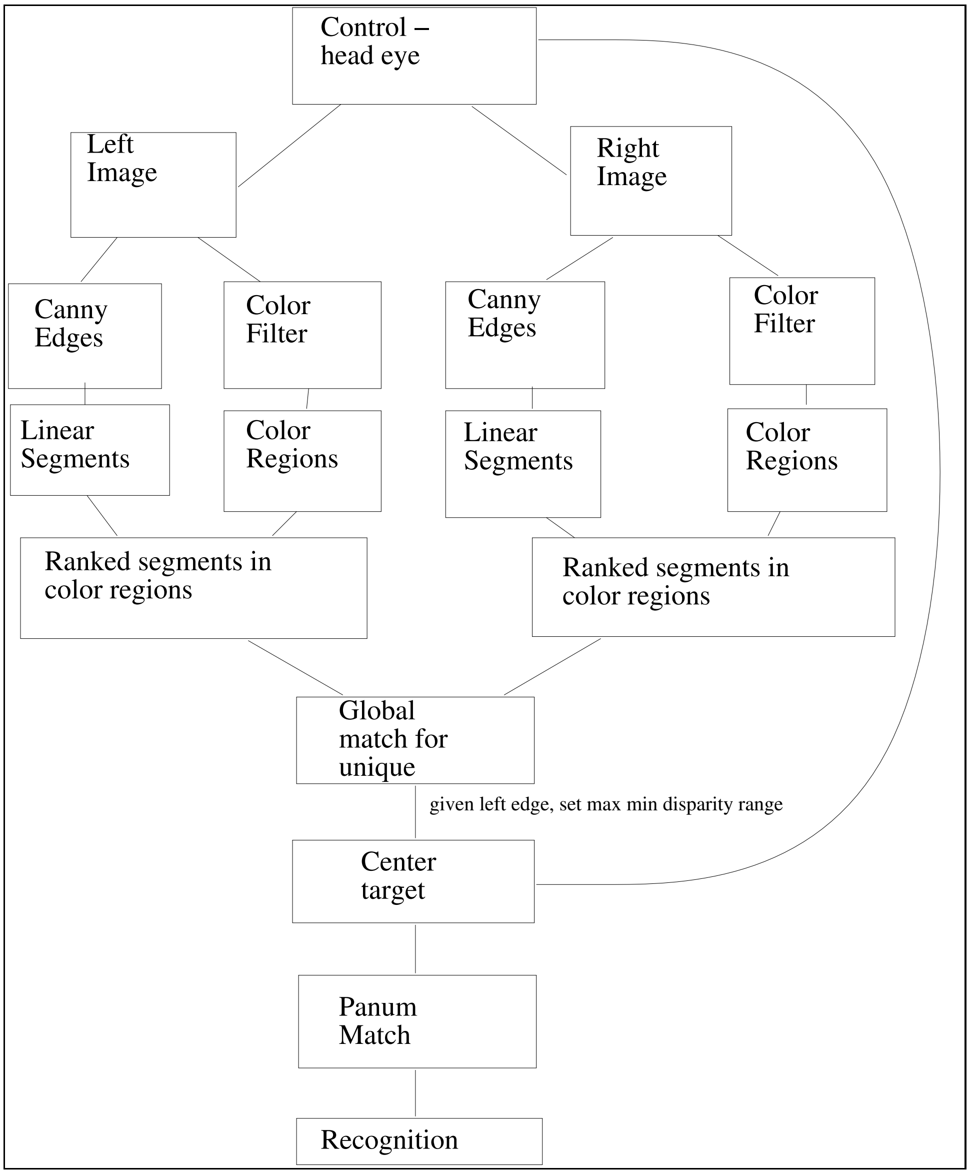

Figure 2-3: The various stages in the working of our system.

2.1.1 The various stages in the working of our system

1. An outer loop scans the area using a head eye system with small increments in pan and tilt angle.

2. At each head position, a coarsely sampled left and right image are extracted.

3. Linear edge segments are extracted from both images.

4. Each feature is a line segment and is described by:

   - Normal to the edge.
   - Offset from the origin.
   - Base and end points.
   - Tangent from base to end.
   - Length of edge.
   - Mean intensity on either side.
   - Mean hue and saturation on either side.
   - Right and left neighbors.
   - List of possible matches.
   - Index number.

5. Using the color information of the target object, the left and right images are segmented into regions (ellipses) that could contain the target object, i.e. regions that have roughly the same color as the target object.

6. Keep only the features that fall within the salient regions extracted above.

7. Potential matches between features in the left and right images are computed in the following way using a stereo matcher [24] that is described in greater detail in Chapter 4. If the images are registered so that the epipolar lines are horizontal and coincide with the scan lines then the stereo matching problem

is simplified since a feature in the left image can only match any feature along the corresponding horizontal scan line in the right image.

Every pair of matching features in the left and right images must satisfy the following matching constraints.

(a) They must have the same contrast sign (whether there is a dark to light transition or a light to dark transition at the edge).

(b) They must have roughly the same orientation.

(c) A significant fraction of the left edge must have sufficient epipolar overlap with the right edge.

(d) They must have roughly the same intensity, hue and saturation values on at least one side of the edge.

(e) The arrangement of neighboring edges at one of the endpoints is roughly the same.

In addition to the matching constraints given above, the algorithm takes advantage of the following global constraints [41] in order to get a focal edge in the left image with a unique match in the right image.

(a) The continuity constraint which says that the world consists of piecewise smooth surfaces. Hence, applying the continuity constraint to a given match (L,R) will yield a large number of likely correct matches within the neighborhoods of L and R if the initial match is correct, and a small number of likely incorrect matches otherwise.

(b) The uniqueness constraint which says that there can be only one match along the left or right lines of sight.

If the focal edge in the left image has only one match in the right image, this is accepted as the correct match. Otherwise, if the left edge has more than one match in the right image, the algorithm scans a neighborhood about the ambiguous match, looks at nearby matched segments and accepts the best match based on the recorded matching information.

Thus, the stereo algorithm finds distinctive segment-features (that lie in the salient color regions from step 5) in the left image which have unique matches in the right image, as measured over the full range of possible disparities. The distinctiveness is measured as a combination of the length and contrast of the feature. Such features could serve as focal trigger features which can be used to fixate the cameras.

8. The disparities associated with each target (trigger) edge in the left image and its matching edge in the right image are used to verge the cameras. This is done by panning and tilting the head so that the corresponding 3D feature is centered between the cameras. The cameras are then moved so that the left edge is centered in the left camera and the matching right edge is centered in the right camera. This gives a simple fixation mechanism where the trigger feature is fixated and foveated in both cameras.

9. A finely sampled (high resolution) pair of images is taken. Salient regions are selected using the color properties of the model as in step 5 and edges within the salient regions are extracted. The edges are matched under the restriction that a match is sought only to within a small depth of field about the fixation point. All edges that have a match at this narrow depth of field, together with their neighboring edges (edges that lie close to them) in the image form the input to the recognition engine.

10. Alignment [29] is used to determine if the target object is present among the selected edges. The results of the alignment are saved and step 8 is repeated with the next trigger feature for the cameras to fixate (from step 7). Once all the trigger features have been fixated in turn, the best result alignment result is saved. If this result indicates the presence of the object in the scene, the system returns the model aligned with the image, otherwise the system returns with a message that it could not find the target object in the scene.

The system is being used to find a target object starting with a given head position and going through the steps mentioned above. The cues used at the moment are color and stereo. The final output is the model aligned with the image or a message that the target object was not found in the scene.

2.2 Evaluation of the system

One way to determine the success of a system is by its performance on some task. In order to evaluate the active attentional system, we need to determine if the selection mechanism succeeded in selecting regions relevant to the task. We use the task of model-based recognition for demonstrating the effectiveness and efficiency of our system. In particular, we evaluate its performance in playing the game "Where's Waldo". In this game, a target object is placed within the domain of the system. The goal is to find the object quickly and correctly. As mentioned before, combining cues and fixating on a set of candidate regions in the image that are likely to contain the object help speed up the recognition process. The system's performance can be evaluated by

1. noting if the regions selected as input to the alignment do indeed contain the target object i.e. noting the number of false positives and false negatives.

2. constructing tables to indicate the reduction in search at the end of each processing stage of the system.

The efficiency of the system can be investigated by running the system on a variety of objects with different levels of scene clutter and noting the number of possibilities explored and the number of false positives and false negatives with and without the various modules.

Chapter 3

Color to preselect focal regions

While we could just use shape based cues like stereo and motion to achieve selection for model-based recognition, we would like to demonstrate the effectiveness of combining shape based cues like stereo with region based cues like color in controlling the combinatorics of recognition methods. Shape cues, in contrast to color, tend to be highly resolution dependent and extracting shape dependent features (e.g. corners) may require elaborate processing. In this chapter, we describe the simple method used to extract rough regions in the image that could contain an instance of the target object based on its color properties.

3.1 Motivation

Color is a useful cue in object recognition for the following reasons:

- it is an identifying feature that is local and is fairly independent of view and resolution.
- it is a strong cue that can be used in locating objects in a scene. Psychophysical experiments conducted by Treisman [62] show that color is used in preattentive visual processing.
- it is useful in segmentation since it gives region information and if specified correctly can be relatively stable to changes in orientation and illumination

conditions, as mentioned by Swain and Ballard in [58].

- a color region in an image tends to come from a single object and thus features within a color region can be grouped together to describe an instance of the object in the image.

We use a color-based description of a model object to locate color regions in the image that satisfy that description. Color information in a model has been used to search for instances of the model in an image in works such as [58] and [60] among others. Swain and Ballard [58] represent the model and the image by color histograms and perform a match of these histograms to locate objects. Syeda-Mahmood [59] developed a model of color saliency to perform data and model driven selection. We use a simple blob-coloring algorithm to roughly segment the image into connected components with color properties similar to the color properties of the model. As explained in chapter 2 (Figure 2.3), the color algorithm serves as a filter to restrict the stereo correspondences to relevant regions in the image (Figure 3-2). Our simple approach causes false positive identifications but we can tolerate these errors in the system since color is used in conjunction with stereo and the combination of cues helps to weed out some of these false targets.

3.2 Color Labeling Algorithm

The target object is modeled by building histograms of its component colors and representing each color by 6 values which correspond to the mean hue, saturation and value and the standard deviation of the hue, saturation and value from the mean. The algorithm to preselect regions in the image is a simple sequential labeling algorithm which finds the connected components in the image that match the color description of the model and represents the connected components by best fit ellipses. Since we do not attempt to model color constancy, we assume that the color of the light source does not change drastically in our experiments. While this simple algorithm has been sufficient to illustrate the importance of color in selecting regions to focus attention, we can generalize it to histogram matching approaches to color segmentation (e.g. [58]) or other color saliency algorithms (e.g [60]) to obtain the same results.

3.2.1 Algorithm

- Input: A model color description, input HSV Image (p)
- Output: A list of ellipses represented by their center, area, orientation, major and minor axes, and an image of labeled regions (out)
- Description: The input image (p) is an HSV image. We scan the image row by row and collect all the pixels whose hue and saturation are within 3 standard deviations of the model hue and saturation. Since the intensity values can change drastically with changing lighting conditions, they were not used in the match. We find the connected components of regions in the image that are of the model color by looking at the 8-connected neighborhood of each pixel that is of the model color. If $p(i, j)$ is of the model color and one of its 8-connected-neighbors has already been labeled, $out(i, j)$ gets that label otherwise $out(i, j)$ gets a new label.

As we scan the image row by row, if $out(i, j)$ is labeled then we add $1, i, j, i^2, i * j$ and j^2 to the accumulated totals of area, first moment x, first moment y and second moments a, b, c respectively for each connected component. At the end of the scan, the area, center and orientation of the bounding ellipse for each connected component can be calculated as follows.

$$\begin{aligned} \text{Center}_{\text{X}} &= \frac{x}{\text{area}} \\ \text{Center}_{\text{Y}} &= \frac{y}{\text{area}} \end{aligned}$$

The orientation of the major axis is given by

$$\sin 2\theta_{min} = \frac{+2b}{\sqrt{(4b)^2 + (a-c)^2}}$$

$$\cos 2\theta_{min} = \frac{a-c}{\sqrt{(4b)^2 + (a-c)^2}}$$

$$\begin{aligned} E_{max} &= 0.5(a+c) + 0.5(\sqrt{(4b)^2 + (a-c)^2}) \\ E_{min} &= 0.5(a+c) - 0.5(\sqrt{(4b)^2 + (a-c)^2}) \end{aligned}$$

The major and minor axis of the ellipse that has the same first and second moments of inertia is defined by α and β, where α is in the same direction as the axis of least inertia and α is greater than β. α and β are given by

$$\begin{aligned} E_{max} &= \frac{\pi}{4} * \alpha^3 * \beta \\ E_{min} &= \frac{\pi}{4} * \alpha * \beta^3 \end{aligned}$$

$$\begin{aligned} \alpha &= (\frac{(E_{max})^3}{E_{min}})^{\frac{1}{3}} * (\frac{4}{\pi})^{\frac{1}{4}} \\ \beta &= (\frac{(E_{min})^3}{E_{max}})^{\frac{1}{3}} * (\frac{4}{\pi})^{\frac{1}{4}} \end{aligned}$$

The object colors are modeled by building histograms of HSV values for the target object, and getting the mean and standard deviations of the distributions of all the colors on the object. The object can be made up of different colors. Once we have the colors of the object modeled, we can apply the algorithm to a color image to isolate regions that are likely to contain the object based on their color.

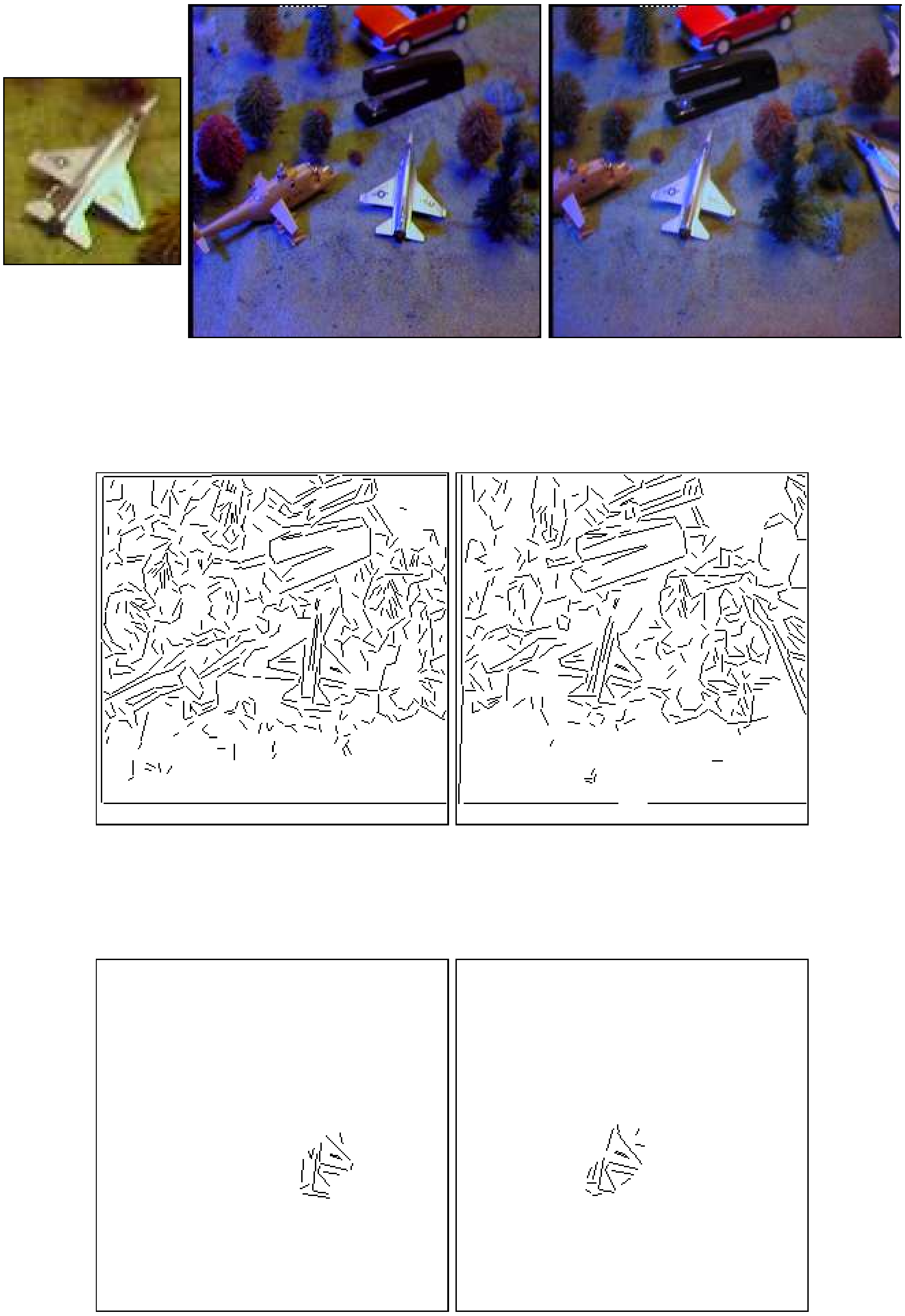

Figure 3-1: Pictures starting from top left. (a) The model, (b) left image, (c) right image, (d) segments in left image, (e) segments in right image, (f) and (g) results from left and right images after applying color filter. Note that the color filter misses many segments on the object due to the change in the color of the light source.

Figure 3-2: Pictures starting from top left. (a) The model, (b) left image, (c) right image, (d) segments in left image, (e) segments in right image, (f) and (g) results from left and right images after applying color filter. The color segmentation is not perfect but it reduces the number of segments that have to considered for the stereo match considerably when compared to (d) and (e) by restricting it to relevant data subsets.

3.3 Why is color alone not sufficient?

In model driven selection, specifying the color of the model object is often not sufficient to get perfect segmentation since we have to account for specularities, shadows and inter reflections that cause the image region containing the model to appear fragmented. Specularities occur as bright white streaks in images of objects with shiny surfaces (e.g. metallic surfaces) under normal lighting conditions. There are methods suggested in the literature that can remove specularities [33] by analyzing the clusters formed when a specular region and its adjacent color region are projected into color space. Another problem with using color for segmentation is one of achieving color constancy or a stable perception of color of varying lighting conditions. There has been some work in the literature to correct for the chromaticity of the illuminant ([42], [47] among others). We have tried to avoid using a complex color constancy model in our method since we are interested in a quick way to roughly segment the scene into regions that could contain the model object. As expected, our simple method causes false identifications with extreme changes in the lighting conditions (Fig 3-1). For example, when the color of the light source is changed drastically like in Figure 3-1(b) and 3-1(c), the algorithm misses parts of the object as shown in Figure 3-1(f) and 3-1(g) and in some cases gets no regions at all. We currently assume normal light sources (e.g. tube lights, halogen lamps etc.) in indoor environments where the lighting conditions do not change drastically.

Color may not provide perfect segmentation due to artifacts like specularities, shadows, etc. but it can be used effectively to isolate relevant regions in low resolution images under normal illumination conditions. These rough regions are useful to focus future visual processing on relevant data sets, thereby reducing the complexity and increasing the reliability of the recognition process. For example in Figure 3-2, the segmentation with our simple color algorithm does not isolate the object perfectly but the isolated region is enough to focus future processing on relevant subsets of the image.

Chapter 4

Stereo in Selection

4.1 Role of stereo in selection instead of 3D reconstruction

If we consider model-based recognition to be a matching of model to data, we could use stereo to give us 3D data that can be matched with a stored 3D model. On the other hand, if we follow up on our argument in Chapter 1 that selection plays a critical part in recognition, then stereo can be used to identify data subsets that are in the same depth range (i.e. subsets that do not have large variations in disparity) and help in selecting parts of the data that are likely to belong to the same object. In this section, we argue that stereo is better suited for figure/ground separation than for 3D reconstruction. In section 4.2 we describe how stereo is used for 3D reconstruction and in section 4.3 we discuss the sensitivity of 3D reconstruction to changes in camera parameters. We describe a stereo algorithm that is modified for selection in section 4.4.

4.2 Stereo for 3D reconstruction

Traditionally, stereo has been used for 3D reconstruction in the following way:

- Pick a point in the left image.

- Find the corresponding point in the right image that is a projection of the scene point as the one picked in the left image.

- Measure the disparity between the left and right image points.

- Use disparity and the relative orientation of the two cameras to determine the actual distance to the imaged scene point. Solving for the distance requires the geometry of the cameras to invert a trigonometric function.

A 3D reconstruction of the scene is obtained by computing the distance to a large number of scene points using the method above.

There have been a number of stereo algorithms in the literature which modify this basic algorithm by using distinctive features like edges, corners or brightness patches and by suggesting different constraints to search for corresponding features in the two images (e.g. epipolar constraint, orientation of features, etc.). Most of the research in stereo stresses that the hard part in recovering depth using stereo by matching features and using trigonometry to convert disparity into depth lies in the matching process (correspondence problem). This is true provided we have ways to determine the camera parameters accurately. The methods suggested to find the camera parameters (e.g. [63]) have been shown to be unstable [65].

4.3 Sensitivity of depth to camera calibration

We note the main results of Grimson's analysis of the sensitivity of depth reconstruction from stereo disparities to changes in camera parameters in [24].

Consider a camera geometry (Figure 4-1) with baseline b, two cameras verged such that each makes an angle of $\alpha_l = \gamma$ and $\alpha_r = -\gamma$ respectively with the line perpendicular to the baseline, each camera has focal length f, and the disparities are d_l and d_r (offset of the projected point in each image from the projection centers).

If we represent the computed depth Z at a point in terms of interocular spacing ($Z' = \frac{Z}{2b}$) and use disparities as angular arcs ($d'_r = \frac{d_r}{f}, d'_l = \frac{d_l}{f}$) then

$$Z' = \frac{1 + \gamma(d'_r - d'_l)}{(2\gamma - (d'_r - d'_l) + 2\gamma d'_r d'_l)}$$

We would like to know how uncertainty in measuring the camera parameters affects computed depth. Grimson uses a perturbation analysis in [24] to show that three parameters can lead to large errors. These parameters are

- The location of the two principal points.
- The focal length.
- Gaze angles.

Errors in locating the principal points lead to large errors in computed depth, e.g. for an object that is 1 meter away from the camera, errors on the order of 10 pixels lead to 10% errors in depth. The current methods for computing principal points [36] have residual errors of about 6 pixels.

Errors in computing focal length result in small errors in relative depth for nearby objects ($\frac{Z}{2b} \approx 10$). Larger disparities lead to larger errors. Thus, if the object is roughly fixated then disparities on the object are small and the depth error is small ([24],[50]).

Errors in computing the gaze angles lead to large errors in relative depth. An error of 1^o leads to a 34% relative depth error for nearby objects ($\frac{Z}{2b} \approx 10$) and a 0.5^o error in gaze angle causes 17% error in relative depth (Figure 4-2).

Thus, if we don't estimate the principal points and gaze angles accurately, we get large errors in our computed depth. These errors in computed depth vary nonlinearly with actual depth. If we are trying to recognize an object whose extent in depth is small compared to its distance, then the effect of the error is systematic and the uncertainty becomes a constant scale factor on the computed depth. If the object has a relative extent in depth on the order of a few percent, then the uncertainty in computing depth will skew the results, causing problems for recognition methods that match 3D data with stored 3D models. Thus, we see that the sensitivity of computed

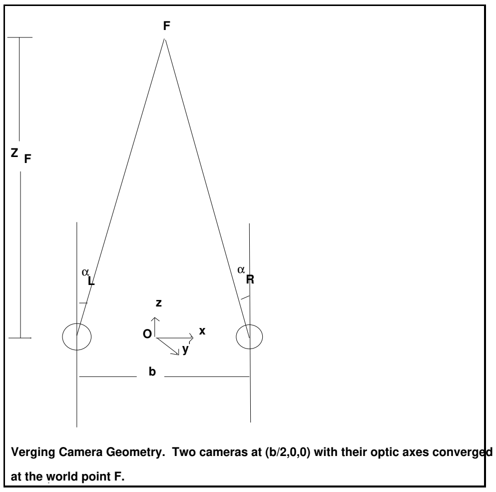

Figure 4-1: Camera geometry with baseline b, two cameras with focal length f verged such that each makes an angle of α_l and α_r with the line perpendicular to the baseline.

depth to camera parameters cause problems for 3D recognition due to large errors in depth and due to distortions in relative depth.

4.4 How can stereo be used without accurate camera calibration?

Among the standard applications of stereo are the tasks of navigation and recognition. Faugeras [14] has argued that a scene around a moving robot can be constructed and maintained without careful camera calibration. They avoid reconstruction by using relative coordinate systems. In this work, we would like to illustrate a similar idea for the role of stereo in recognition.

We have argued in Chapter 1 that selection plays an important role in recognition. If stereo is used for selection instead of 3D reconstruction then we could avoid explicit 3D input for 3D object recognition by using view based recognition schemes like [64]. These view based recognition schemes use stored 2D views of a model to generate a hypothesized image that can be compared to the observed image.

From sections 4.3 and 4.4 we can conclude that

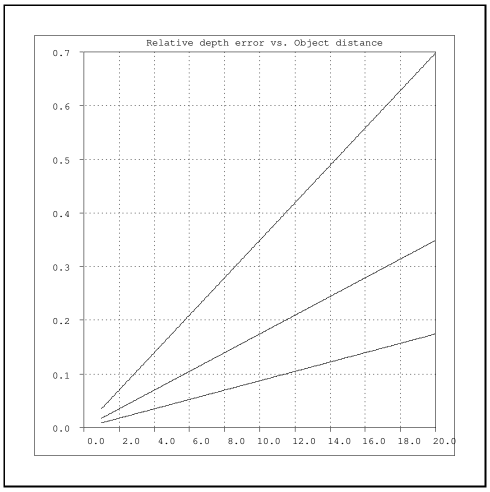

Figure 4-2: Plots of the percentage error in depth as a function of object distance (in units of interocular separation). Graphs represent errors in computing gaze angles of 1, 0.5 and 0.25 degrees, from top to bottom. This figure is taken from [24].

- small inaccuracies in measuring camera parameters result in large errors in depth.
- Selection is a critical part of object recognition and we can avoid explicitly computing 3D distances if we use stereo for selection. This new role of stereo allows us to do recognition without the need for careful camera calibration.

4.5 Geometry Of Verging Systems

As shown in figure 4-1, the verging system has the following camera geometry. The system has two cameras that can rotate about the x axis and an axis parallel to the y-axis. The axes of rotation pass through the nodal points of the cameras. Thus, the projection of a world point (X, Y, Z) to image coordinates $X_L = (x_L, y_L)$ and $X_R = (x_R, y_R)$ is given by

$$X_L = f\left(\frac{(X+\frac{b}{2})\cos\alpha_L - Z\sin\alpha_L}{(X+\frac{b}{2})\sin\alpha_L + Zcos\alpha_L}, \frac{Y}{(X+\frac{b}{2})\sin\alpha_L + Z\cos\alpha_L}\right)$$

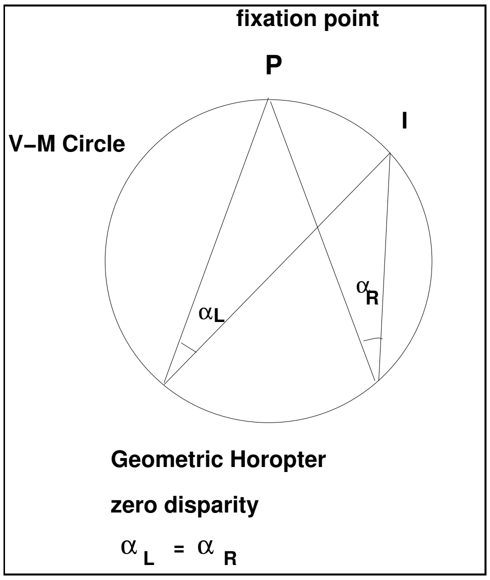

Figure 4-3: The Geometric Horopter.

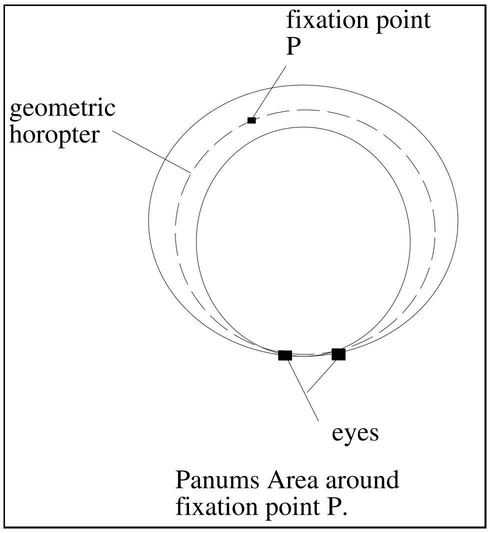

Figure 4-4: Panum's Area. The dotted circle is the zero disparity locus (geometric horopter) and the region between the two solid rings is the area in which disparities are less than $\pm 14'$ (panums area)

$$X_R \quad = \quad f\left(\frac{(X-\frac{b}{2})\cos\alpha_R + Z\sin\alpha_L}{-(X-\frac{b}{2})\sin\alpha_R + Z\cos\alpha_R}, \frac{Y}{-(X-\frac{b}{2})\sin\alpha_R + Z\cos\alpha_R}\right).$$

Solving for (X, Y, Z) given $(x_L, y_L) = (x_R, y_R) = (0, 0)$ gives the fixation point F:

$$F \quad = \quad \left(\frac{b(\tan\alpha_L - \tan\alpha_R)}{2(\tan\alpha_L + \tan\alpha_R)}, 0, \frac{b}{(\tan\alpha_L + \tan\alpha_R)}\right).$$

4.5.1 Isodisparity Contours and the Geometric Horopter

The set of world points that give rise to image points with a fixed horizontal disparity d is given by $x_L - x_R = d$. When d is 0, this set forms a circle called the Veith Muller circle or the geometric horopter. This circle passes through the nodal points of the cameras and the fixation point and is independent of the individual camera angles α_L and α_R (Figure 4-3). Isodisparity contours when $d \neq 0$ can be approximated by circles in the region of central vision provided d is small or $\alpha_L \approx \alpha_R$.

4.5.2 Panum's Area

Panum's area refers to the narrow range of disparities over which humans are able to achieve stereo fusion easily (Figure 4-4). This limit on disparities implies that the fusible region in humans is restricted to a narrow range of depths about the fixation point and that the stereo system fails over most other parts of the scene. A verging stereo system resembles the human stereo system in that fixates a target and searches for matching targets over a narrow range of disparities around the fixation point (referred to as Panum's area). A verging system can be used as a resource that provides extra information about fixation points.

4.6 Using stereo to focus on target regions

Stereo can be used for selection by using the property that nearby points in space project to nearby points in the images and occupy a narrow range of disparities. Thus, we can use the disparity of some pair of matching features coming from an object to fixate the object and find all other matches around the initial match that lie in the same disparity range. These matched features that have similar disparities are likely to come from the fixated object.

If we are using stereo for figure/ground separation since computing distance reliably without accurate camera calibration is difficult, then the algorithm should be able to do the following.

- Detect features that are close to each other in the image that lie within some depth band.
- Center the matching features in the left and right images so that neighboring parts of the same object are visible in both images.
- Choose target features to foveate and fixate.

4.6.1 Description of the Stereo Algorithm

The main problem in the matching process is finding a unique match and this depends on the control mechanism used by the algorithm. Most of the stereo algorithms in the literature have been used for reconstruction and were designed to find as many matches as possible over a wide range of disparities. One of the main problems in stereo matching lies in determining what constitutes a unique match. Stereo algorithms that try to find matches over a wide range of disparities (on the order of hundreds of pixels) face difficulties in trying to guarantee a unique match based on local attributes of features, like contrast and orientation. One solution to this problem is to use attributes of nearby features [3], [46], [40] and another is to alter the control strategy.

Since we are interested in finding roughly contiguous 3D regions to select out groups of image features that are likely to come from a single object, we use a control method fixates a target, searches for matching features in some narrow disparity range ($\pm\delta$) around the fixation point, and collects all the matching features in this disparity range as the selected features. This is similar to the working of the human stereo system where the fusible range of disparities is restricted around the fixation point (Panum's area).

The stereo algorithm implemented here is a modified version of Grimson's stereo matcher ([20]). It is similar to earlier stereo algorithms [3] and [46], [40], [37], [41] and uses ideas about the human stereo system, Panum's area and the role of eye movements in stereopsis as discussed in [37], [41], [24] and [50].

The stereo algorithm does the following:

- decides what features to match in the two images,
- decides how the matching is to be done,
- uses a coarse to fine mechanism in an active vision framework, where it fixates on a candidate feature at the coarse level and match features within a narrow disparity range around that point to get regions that probably come from the same object in 3D-space at a finer level of resolution (Figure 4-5).

4.6.2 Features for the stereo matching process

The features used for the stereo match are line segments obtained from intensity edges by running a split and merge algorithm [53]. Each segment is described by its end-points, HSV (hue, saturation and intensity) values on either side, distance and arrangement of its neighboring segments.

4.6.3 Stereo Matching Constraints

There are several constraints obtained from physics and gemoetry that can be used in the matching process.

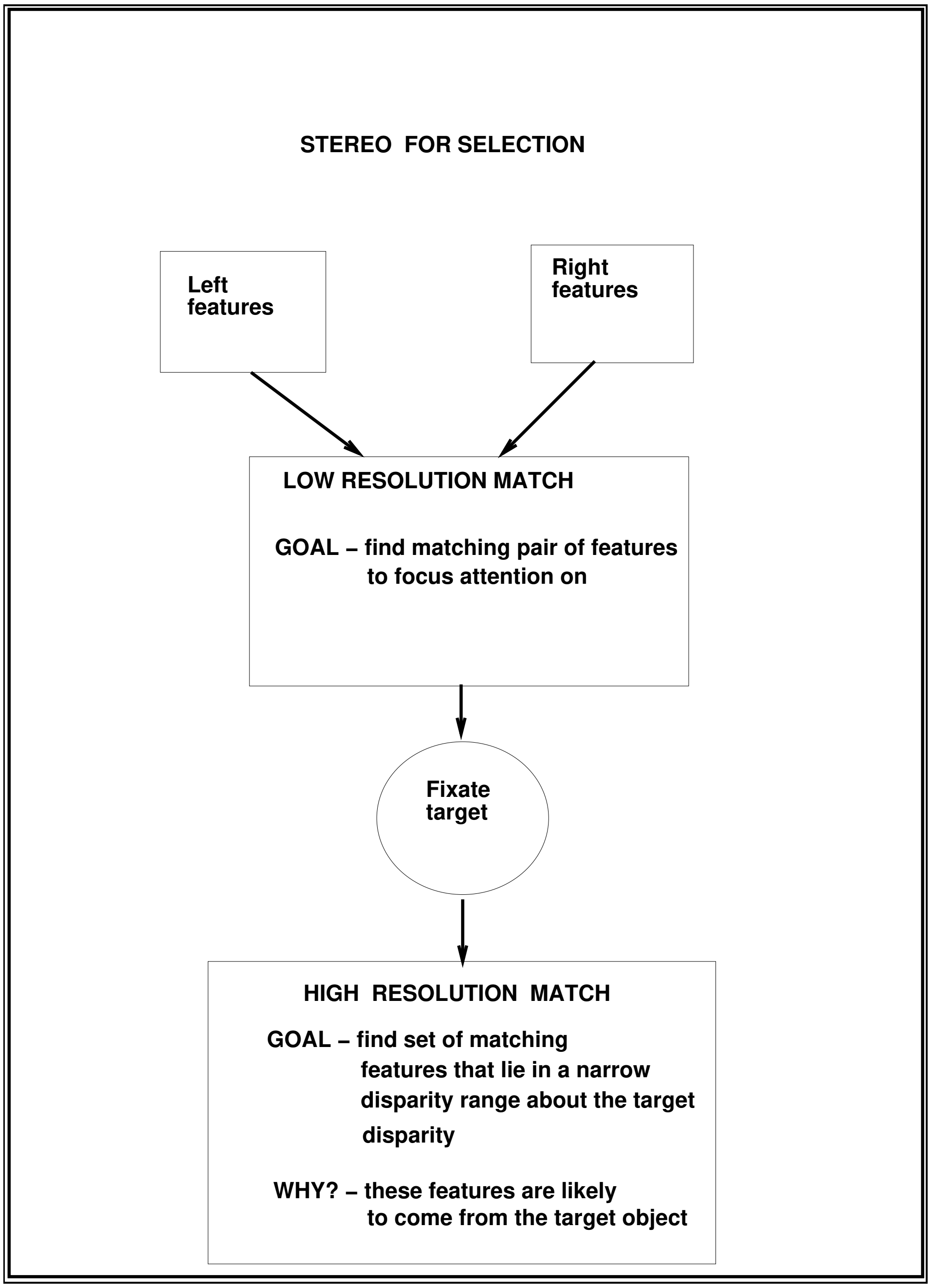

Figure 4-5: Stereo Algorithm.

- Epipolar Constraint:

  This ensures that the images P_l and P_r of a world point P must lie on corresponding epipolar lines. An epipolar line is the intersection of the plane containing the two lens centers with the image plane. The epipolar lines in one image all radiate from one point where the line through the two lens centers cuts the image plane. An object imaged on the epipolar line in the left image can only be images on the corresponding epipolar line in the right image.

- Local Constraints :

  This ensures that matching segments have similar intensity, color, contrast, orientation and overlap.

- Constraint on neighbors :

  If two segments coming from some object match, then the geometry of the neighboring segments in the left and right images are similar and the neighboring matches lie in a small disparity range.

- Uniqueness Constraint:

  If the best match for left segment (L) in the right image is R then the best match for R in the left image is L.

4.6.4 Similarity Measure

The similarity measure $S(m,n)$ describes how good a potential match is based on the similarity of local properties like length, orientation, contrast etc. $C(m,n)$ and the similarity of neighboring matches that lie within a narrow disparity range δ about the disparity of the potential match.

$$S(m,n) = C(m,n) + \sum_{k \in neighb(m)} \left(\frac{1}{dist_{m,k}} \max_{l \in neighb(n)} \frac{C(k,l)}{dist_{n,l}} * \Delta disp_{m,n,k,l}\right)$$

$\Delta disp_{m,n,k,l} = 1$ if $disp_{m,n} - disp_{k,l} \leq \delta$

$$C(m,n) = \text{LENGTH}_{\text{sim}} + \text{ANGLE}_{\text{sim}} + \text{INTENSITY}_{\text{sim}} + \text{OVERLAP}$$

4.6.5 Algorithm

1. Get the intensity edges from the left and right images. Get segments from the intensity edges by running a split and merge algorithm [53] on the edge images.

2. Each segment is described by its end points, HSV (hue, saturation and intensity) values on either side of the segment and the distance to neighboring segments

3. Low Resolution Match:

   For a distinct (long, with high contrast and of the object color) feature in the left image find a unique match in the right image. The unique match is found using the whole range of disparities. Every pair of matching features in the left and right images must have the same contrast sign (whether there is a light to dark transition or a dark to light transition at the edge), roughly the same orientation, roughly the same intensity, hue and saturation values on one side of the edge, roughly the same arrangement of neighboring edges at one of the endpoints and a significant fraction of the left edge must have sufficient epipolar overlap with the right edge. The best match is is the one that maximizes the similarity measure (section 4.6.4) in both directions (i.e. from the left image to the right and vice versa). This feature is used to fixate the cameras. If there are several distinctive features in the left image with unique matches in the right image, we fixate each of these features in the order they were found.

4. Adjust the pan and tilt angles of the cameras to foveate and fixate the target feature in the left and right images. Verge the cameras so that the target feature is centered in both the images. The verging of the cameras leaves the optic axes non-parallel so that the epipolar lines are no longer along the scan lines of the image. Resection (reproject) the images so that the optic axes are parallel or

else center the feature in one of the cameras by adjusting the pan and tilt angles and leave the optic axes parallel. At present, the feature is centered in the left image using pan and tilt of the head while leaving the optic axes parallel.

5. High Resolution Match:

   Search for features having a unique match within a narrow disparity band $\pm\delta$ (Panum's limit) about the target disparity due to fixation. Features that match outside this range of disparity are ignored. The matching criteria remain the same as in step 3.

6. This set of features obtained gives us the region selected from the image as a candidate region containing the model object.

7. Save the selected features and fixate on the next feature obtained from step 3.

8. Once all the candidate features from step 3 have been explored and the respective features collected in step 5, we pass the groups of features obtained in step 5 to a recognition engine [29] that aligns the model with the selected feature set and verifies if the object is present in the image or not.

Chapter 5

The Recognition System

The previous chapters discussed the development of a selection mechanism using color and stereo cues. The selection was done to improve the performance of a recognition system. In this chapter, we describe

- how the results of the selection module can be evaluated using the recognition system,
- the recognition system,
- the integration of the attentional selection module with a recognition system,
- how its performance can be improved by using attentional selection.

5.1 Why build a recognition engine for the system?

We need a recognition engine for the system in order to assess the performance of selection. In the previous chapters, we discussed how selection helps reduce the search involved in recognition. We saw that in the worst case, selection reduced the combinatorics of the recognition system significantly, but in practice, we might not have objects with color and long edges and other information that would select out the region containing the object accurately. Thus, we have to account for errors in the selection mechanism and see how it affects the recognition process in terms of false positives and false negatives.

5.2 The Recognition System

There are a number of recognition systems in the literature [29], [21], [22], [64], [9] that recognize rigid objects using a geometric description of the model. These systems have a geometric description of the model in terms of features like points and lines. They extract similar features from the image and find the correspondence between the model and image features to compute a transformation that projects the model onto the image. The difference between the various recognition methods lies in the way in which they approach the combinatorics that results from examining all matches between model and image features to get the correct transformation. We have correspondence-space based methods [21], [22], [9] that explore the space of all possible matches between the model and data features and pruned the search space by using geometric constraints on the model and image features [21] or by using distinctive features on the model to guide the search [9]. Another set of methods for recognition are alignment-based methods [29], [64] that explore only a part of the interpretation by matching a small number of model and image features that are sufficient to compute the transform that aligns the model features with the image features. We used an alignment-based recognition system. These methods use a minimal set of corresponding features to produce a transformation that aligns the model with the image. These methods tend to have problems in cluttered environments. We show the advantages of using attentional selection while using alignment-methods to recognize objects in cluttered environments.

5.3 Recognition Using Alignment

The recognition system we built uses an alignment-based method developed by Huttenlocher and Ullman [29]. The design of the recognition system involved picking features to match, building the model and choosing a method for verification.

5.3.1 Alignment Method

In this method, the model is represented as a list of 3D-points. The description of the alignment method follows [29].

Definition 1

Given 3 non-collinear points a_m, b_m and c_m in the plane and three corresponding points a_i, b_i, and c_i also in the plane, there exists a unique affine transformation[1], $A(x) = Lx + b$ where L is a linear transformation and b is a translation such that $A(a_m) = a_i$, $A(b_m) = b_i$ and $A(c_m) = c_i$.

Definition 2

Given three non-collinear points a_m, b_m and c_m in the plane and three corresponding points a_i b_i and c_i in the plane, it is shown in [29] that there exists a unique transformation, $Q(x) = Ux + b$, where U is a symmetric matrix and b is a translation vector, such that $\Pi(Q(a'_m)) = a_i$, $\Pi(Q(b'_m)) = b_i$, $\Pi(Q(c'_m)) = c_i$, where $v' = (x, y, 0)$ for any $v = (x, y)$, and Π is the orthographic projection onto the $x - y$ plane.

Computing the transformation

As shown by Huttenlocher in [29], we can use the following algorithm to compute Q and the two-dimensional affine transform A given three pairs of corresponding points (a_m, a_i), (b_m, b_i), (c_m, c_i) where the image points are in two-dimensional image coordinates and the model points are in three-dimensional object coordinates.

1. Rotate and translate the model so that the new a_m is at (0,0,0) and the new b_m and c_m are in the $x - y$ plane. Get all the model triples off line.

[1] An affine transformation in a plane is linear and can account for uniform rotation, translation, scaling, skewing and shearing. An affine transformation has 6 parameters.

2. $b = -a_i$ is the translation vector. Translate all image points so that the new a_i is at the origin.

3. Solve for the linear transformation L using

$$Lb_m = b_i$$

$$Lc_m = c_i$$

4. Solve for c_1 and c_2 using

$$c_1 = \pm\sqrt{\frac{1}{2}(w + \sqrt{w^2 + 4q^2})}$$

$$c_2 = \frac{-q}{c_1}$$

where

$$w = l_{12}^2 + l_{22}^2 - (l_{11}^2 + l_{21}^2)$$

(l_{ij} are the elements of the linear transformation matrix L) and

$$q = l_{11}l_{12} + l_{21}l_{22}$$

5. We can now compute two symmetric matrices sR^+ and sR^- that differ by a reflection.

$$sR^+ = \begin{pmatrix} l_{11} & l_{12} & (c_2l_{21} - c_1l_{22})/s \\ l_{21} & l_{22} & (c_1l_{12} - c_2l_{11})/s \\ c_1 & c_2 & (l_{11}l_{22} - l_{21}l_{12})/s \end{pmatrix}$$

where

$$s = \sqrt{l_{11}^2 + l_{21}^2 + c_1^2}$$

sR^- is identical to sR^+ except that terms $r_{13}, r_{23}, r31$ and r_{32} are negated. The image coordinates of a transformed model point are given by the x, y coordinates of x' where

$$x' = sRx + b$$

with translation vector b, scale and rotation sR.

This method for computing the transformation is relatively fast, but since there is no automatic way to build 3D models of objects, building full 3D models of the objects manually is tedious.

5.4 Recognition Using Linear Combination Of Views

We discovered that building 3D-models for all objects is not a feasible idea and thus tried out another recognition method where the model is represented by a set of 2D views. This is the recognition method using linear combination of views [64].

5.4.1 Linear Combination Of Views

In this method, the object is represented as a small set (3) of 2D-views and full correspondence is provided between these views. A description of the method follows.

Let O be a rigid object. Let P and P_1 be two 2-D images of O such that P_1 is an out of plane rotation of P. Let O' represent O following a 3D affine transformation and P' is a new view corresponding to O' under orthographic projection

$$O' = AO + T$$

where A is a linear transformation matrix, and T is the translation vector. If u_1, u_2 and t_x, t_y represent the first two rows of A and T respectively, we can express the coordinates of a point (x', y') in the new view as

$$(x', y') = (u_1 \cdot p + t_x, u_2 \cdot p + t_y)$$

If r_1 is the first row of R, and e_1 and e_2 represent the first two rows of an identity matrix, then e_1, e_2 and r_1 span R^3 if they are linearly independent vectors so that any vector u_1 can be expressed as a linear combination of these 3 basis vectors

$$u_1 = a_1 e_1 + a_2 e_2 + a_3 r_1$$

$$u_2 = b_1 e_1 + b_2 e_2 + b_3 r_1$$

and using the above two equations we get

$$(x', y') = (a_1 x + a_2 y + a_3 x_1 + a_4, b_1 x + b_2 y + b_3 y_1 + b_4)$$

where $a_4 = t_x$, $b_4 = t_y$ and (x_1, y_1) are the coordinates of (x, y) in view P_1. Thus if the correspondence between four points (x, y), (x_1, y_1) and (x', y') in views 1,2 of the model and the new image view are known, the coefficients (a_i, b_i) for i = 1,2,3,4 can be solved. When the correspondence between the two model views is known, these coefficients can be used to align all the points of the first model view with the new view and the alignment can then be verified.

5.5 Picking features for recognition

In order to benefit from the alignment method, we need a few distinguishing features that are relatively stable and are sufficient for performing alignment [20]. If we consider computing alignments using all points along the contour of the model and data as features, then we have to try a large number of alignments even for a simple model. If we consider using end-points of line segments in the model and data as our features, then we have to cope with uncertainty in the position of the end-points in the data due to edge fragmentation or occlusion and errors in the location of end-points could lead to many incorrect alignments. Corner features are good for alignment since an object generally has only a few corners. Corners also tend to be spread out over an object and give better alignment results than features that are close to each other.

In our system, we approximate the curves in the edge image by line segments and use the junction points where two line segments meet is considered a corner feature. We also use the orientations of edge segments to induce virtual corners [29]. Figure 5-1 shows an example of a virtual corner induced at the point of intersection of two extended edge contours. Let a and b be two data points with orientation vectors a_i and b_i and A is the line passing through a in the direction a_i, B is the line through b in the direction b_i. It has been shown in [29] that if the distance from the two edge points a and b to the intersection point c is large then a small error in either of the two orientation vectors causes a large positional error in the location of c.

The corner features give us a reasonable set of features for alignment. However, since our selected data consists of a group of line segments, we could have used a combination of points and lines to compute the alignment transform (e.g. [48]) as well.

Once the alignment transform has been computed we use the line segments as features to verify the alignment. The line segments are described by their length, orientation, the hue, saturation and intensity on either side of the segment in the image.

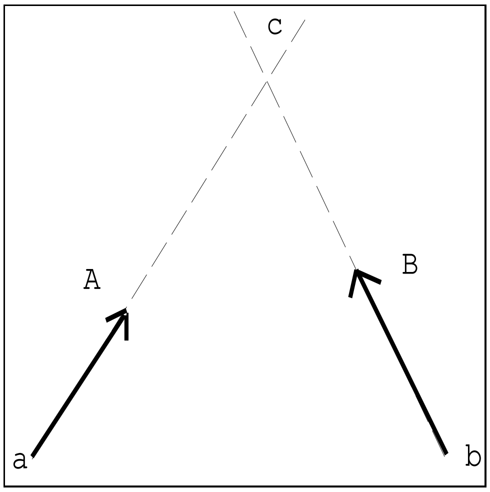

Figure 5-1: Virtual point c at the intersection of two extended edge contours.

5.6 Complexity of the matching process

5.6.1 Alignment

In the absence of any other information, we have to try all possible triples of model and data features in order to align the model with an image. If we have m model features and n data features, there are $\binom{m}{3}$ model triples and $\binom{n}{3}$ data triples, each of which may define a possible alignment of model with image. Thus each of these $O(m^3 n^3)$ alignments needs to be verified by transforming the model to the image and checking for additional evidence of a match.

5.6.2 Linear Combination of Views

As we saw in section 5.4, we need four corresponding points in the model and image to compute the alignment transform using this method. Thus, we need to try $O(m^4 n^4)$ alignments. Although 4 corresponding features are sufficient to compute the linear combination coefficients, we need around 7 matching features to get an accurate estimate of the parameters. This increases the number of matches to be tried from $O(m^4 n^4)$ to $O(m^7 n^7)$.

In both the methods given above, we can reduce the number of matches to be tried by using additional constraints like color and intensity information around the feature, the angle at the corner, etc.

5.7 Verification

Once the alignment transform has been computed, we have to determine whether or not the transformation brings the transformed model in correspondence with an instance in the image. We use the oriented line segments in the model and image in the verification process. We verify the alignment going from the model to the data and from the data to the model. For each transformed model segment we find data segments of the same orientation and roughly the same length that lie within $\pm\delta = 10$ of the projected model segment. If there are multiple segments that satisfy the above conditions then the image segment that matches the projected model segment is the one that *minimizes* S where S is given by

$$S = \frac{\Delta\text{dist}}{\delta} + \frac{\Delta\text{orient}}{180} + \frac{\Delta\text{length}}{\text{model-length}} + \frac{\Delta\text{l-hue}}{360} + \Delta\text{l-sat} + \frac{\Delta\text{r-hue}}{360} + \Delta\text{r-sat}$$

The Δ terms indicate difference between the projected model line and the image line with respect to distance, length, orientation, hue and saturation on either side of the edge. The denominators are the normalizing factors. Once a matching image segment is found, it is removed before verifying the next model segment. The fraction of model segments with matched image segments is noted. The same process is repeated from every image segment to find the matching model segment and the fraction of image segments with matching model segments is noted. Both these fractions have to pass a threshold for the projected model to be considered in alignment with the data (i.e. for the object to be recognized in the image).

5.8 Integrating the results of selection with recognition

The selection module of the system returns a set of selected segments where each segment is described by its length, orientation and the hue, saturation and intensity on either side. The corners extracted from this selected set of segments are used to drive the alignment process. The segments with all their attributes are used to verify the results of the alignment and prune the number of alignments that have to be tried. We have described two alignment style recognition techniques here. We implemented both since the linear combinations method breaks down if we have planar objects.

5.8.1 Model Representation

The model representation varies depending on whether we use the alignment [29] or the linear combination of views [64].

1. Alignment for planar objects: The model is a set of line segments that represent the measured contours of the model object.

2. Linear Combination Of Views: The model is represented as a set of two views of the object. Four corresponding points between the two views are specified. A complete set of corresponding segments representing the contours of the object in the first two model views is also stored. These segments are projected into the image using the transformation matrices obtained by using 4 corresponding points in the three views (the two model views and the image view consisting of the selected segments).

5.8.2 Features

The features used for aligning the model with the image are the corners as specified in section 5.5. In the case of alignment we cycle through all possible triples of data corners to find the three corresponding model and data points to compute the transformation as given in section 5.3. In the case of linear combination of views, we cycle through all possible sets of four features to find the four corresponding points in the three views in order to compute the transformation as given in section 5.4.

5.8.3 Verification

Once the transformation is computed, the projected model segments are aligned with the data segments and the verification method described in section 5.7 is used to determine if the alignment is good enough. In addition we used the distance between the centroids of the model and data features as an initial test to prune hypotheses where the projected model segments lie far outside the selected region. Many of the candidate alignments can be easily filtered out by using rough scale factors and checking for alignments resulting from unstable basis points. These initial tests help us to avoid wasting time on doing the segment verification for alignments that are clearly wrong.

5.8.4 Refinement

Once we have found a correct solution, we refine it to get a better correspondence of the model and data by using a least squares minimization technique (Powell's Method). We minimize the normalized sum of the distance from every model segment to the closest data segment of the same length and orientation and the normalized sum of the distance from every data segment to the closest model segment of the same length and orientation. The refinement of the final pose of the solution improves the alignment of the model and data features.

5.8.5 Problems

The features that we are using don't give us perfect alignments since the selected data often has fragmented segments or some missing segments which makes it hard to get points in the model and image that correspond exactly. Due to noisy data, the minimal number of corresponding points is often not good enough to give perfect alignments using these recognition methods. For example while using the linear combination of views, although 4 corresponding points in the three views is sufficient, this is not enough to get a good alignment. In practice we need 7 corresponding points in the three views and finding 7 corresponding points is very time consuming even with 20 model and data segments ($20^7 * 20^7 = 4e^{16}$ alignments). Another weak point in this recognition module is the verification process. Although the verification gives the correct answer most of the time, it gives rise to false positives when objects have similar shape or there are occluded features. These false positives are discussed in greater detail with the results in Chapter 6.

Chapter 6

Results

In this chapter we report the results of experiments done to test the system described in chapter 2. We discuss the task used to test the system and describe typical scenes and models that were used. The experiments show the reduction in the search as a result of using the attention mechanism. We also discuss the reliability of the system and issues concerning false positives and false negatives.

6.1 Description of the models and test scenes

The system was used to perform the equivalent of playing "Where's Waldo", i.e. it was required to find a small target in a cluttered environment quickly by focusing its resources on regions that are most likely to contain the target object. The models we used were colored objects. The objects did not have much texture on them and were placed in a cluttered airfield scene indoors in a lab. The scene also had other distractor objects with features resembling the model features. The distractor objects had similar color and shape as the model objects. The scene was imaged using two color CCD cameras mounted on a head-eye system. The orientation of the target and the intensity of the lights in the room were varied while testing the system. The starting position of the head was also varied so that the target object was either totally visible in both the left and right images, partially visible in both images, visible in one eye and not visible in the other eye or not visible in both eyes. Figure

6.1 shows a typical scene containing the model.

6.2 Experiments

The reliability of the system was tested by running it 50 times on scenes containing different target objects, in several orientations under varying lighting conditions. Tables 6.1 and 6.2 show the search reduction at each stage in the number of segments and tables 6.3 and 6.4 show the reduction in the number of alignments at the various stages for the same 50 runs. The results of some of the runs are discussed below.

Experiment 1 - Figures 6-1–6-12

Figures 6-1–6-6 show the results at different stages of one complete run of the system where the head was initially positioned so that the target object was present in the image taken by the left eye and absent in the image taken by the right eye. The head initially takes a pair of coarse resolution images. The images are processed to find edges and approximate the edges by line segments. The color filter is applied and only those line segments that fall in regions that have color properties similar to the color properties of the target object are retained. The results of applying the simple color filter are not perfect but they are useful in guiding the stereo match by reducing the number of correspondences that have to be tried by the stereo algorithm. The stereo matching is done on the segments retained after applying the color filter and the head is turned so as to center the focal edge resulting from the stereo match in the left image. The head takes two more images and extracts the 256 * 256 array about the focal match to obtain the images shown in figure 6-5. The segments are extracted from these images and the stereo match is done in a narrow disparity range about the disparity of the focal match (fixation disparity). The resulting set of segments shown in figure 6-6(a) are those that are likely to contain the target object. The selected segments are fed into an alignment-based recognizer and the results of the alignment are shown in figure 6-6(b). The poor results of the alignment indicate that the selected set of segments did not belong to the target object. The head moves a little and repeats the scanning process. Figures 6-7–6-12 show the results of the run after the head has moved. The target plane is visible in both the left and right images. Figure 6-9 shows the segments after the color filter has been applied. Figure6-10 shows the images taken after the head has moved to center the matched focal edge resulting from running the stereo algorithm. Figure 6-11(a) shows the set of segments likely to contain the target object after the second stereo match is done on a narrow disparity band around the center of the images shown in figure 6-10. These segments are fed into the recognition engine and figure 6-12(b) shows the results of aligning the transformed model and the selected segments. The good alignment implies that the selected segments came from the target object. The system stops scanning the room further since the object has been found.

Experiment 2 - Figures 6-13–6-21

Figures 6-13–6-21 illustrate the processing at the various stages for a different orientation of the same object (plane) and different intensity of the light source. In this example, the target object is found in two fixations. Figures 6-13–6-15 show the initial scene and segments before and after the color filter. Figures 6-16–6-18 show the results of the first fixation when the eyes foveate a distractor object (plane) that has the same color properties as the target object. Figure 6-18(b) shows the results of the alignment which indicates that the object is not found in the selected set of segments. The head turns and fixates on the next focal edge which foveates the correct target object as seen in figure 6-19. Figures 6-19–6-21 show how the target object is recognized with this fixation.

Experiment 3 - Figures 6-22–6-30

Figure 6-22–6-30 show the various stages of the processing on a different object (red car). Figure 6-22 shows the initial images. Figure 6-23 and 6-24 show the segments before and after the color filter has been applied. The target object is missing some edges and is hard to spot in figure 6-22 but the target object is clearly visible at higher resolution in the foveated images in figure 6-28. This example illustrates how fixation guided by visual attention helps in recognition by examining interesting regions in the image in greater detail (at higher resolution). The recognition was done using linear combination of views in this case. There are two fixations. The first fixation investigates the orange object in 6-22. The selected features after the first fixation are rejected by the recognition engine. The second fixation investigates the red car in **??** and the results of aligning the transformed model view with the set of segments selected as likely to contain the target object (segments in figure 6-24(a)) are shown in figure 6-30(b). In this example, the target object (car) is found by the system.

Experiment 4 - Figures 6-31–6-36

Figures 6-31–6-36 show the results of the various stages of processing when the system finds a simple planar object.

Experiment 5 - Figures 6-37–6-42

This example illustrates the fact that individual cues do not have to be very accurate when several cues are used in conjunction. Figures 6-37–6-42 show the results of the various stages of processing when the color of the light source is changed by covering the light source with blue paper. Figure 6-37 shows the initial color images and figure 6-38 shows the segments. Figure 6-39 show the segments remaining after the color filter has been applied. The results of the simple color filter are not good due to the change in color of the light source and a lot of segments on the object are missing. In this example, the few segments remaining in the two images after the color filter had been applied were enough to get a stereo match and figure 6-40 shows the foveated images after the head turned to center the matched edge in the left image. Figures 6-41 and 6-42 show the segments in the foveated images, the selected segments and the results of aligning the transformed model with the image. Figure 6-42(b) shows that the target object has been found. This example illustrates the advantage of using multiple cues by showing how the system recovers from the bad performance of the color filter by using stereo. It also shows that the simple color algorithm (without a color constancy model) described earlier is unstable when the color of the light source is changed drastically.

Discussion of the results recorded in tables 6.1–6.4

Tables 6.1 and 6.2 show the reduction in the number of features (segments) at the various stages in the running of the system on 50 runs. The columns from left to right represent the following. *L-segs* gives the number of segments in the coarse left image, *R-segs* gives the number of segments in the right coarse image, *Col-L-segs* and *Col-R-segs* gives the number of left and right segments after color filter has been applied, *Foc-L* and *Foc-R* gives the average (over number of fixations) number of segments in the foveated left and right images, *Final* gives the number of selected segments, *Ans* says if the object was found (F) or not found(NF), *Fix.* gives the number of fixations it took to find the object, *Rt* says if the answer given by the system is correct or wrong.

Tables 6.3 and 6.4 show the reduction in the number of alignments that have to be tried at the various stages of the system on the same 50 runs described in tables 6.1 and 6.2. The format of the tables and the columns representing the various stages are identical to tables 6.1 and 6.2 described above.

The tables indicate that the system works efficiently (by reducing the number of matches that have to be explored by the recognition system) and reliably (by correctly finding the target object when it is present in the scene and returning without finding the target object when it is not present in the scene). Out of the 50 trials, there were 7 false identifications (5 false positives and 2 false negatives). As an example, let us discuss the first trial when the system finds the target object. In this trial, the system finds features to fixate the cameras and examines them in the order they were found by the stereo algorithm. At each fixation, the system takes a pair of images and finds matches between the two images in a narrow disparity range about the fixation point. These matched edges are fed into the recognition engine which transforms the model and verifies whether the selected segments represent an instance of the target object or not. In trial 1, the system finds the target object correctly and it took three fixations before the segments selected were verified as representing the target object. Now, let us examine trials when the system correctly determines that the target object is not present. In trial 2, there are 7 targets that the system fixates on and none of the 7 foveated regions contained an instance of the target object. Thus, at the end of this trial, the system correctly determines that the target object is not present in the scene. Trial 6 is another example where the system correctly determines that the target object is not present in the scene. Trial 6 differs from trial 2 in that it finds no regions in the image with color properties similar to that of the target object and as a result has no targets to fixate. Thus, in trial 6, the system correctly decides that the target object is not present in the scene immediately after the color filter is applied. The examples discussed above illustrate the fact that system gives the correct answer most of the time. We now discuss the cause of the false positive and false negative identifications made by the system from the tables 6.1 and 6.2.

We see that in trials 7, 8, 11, 38 and 42 of tables 6.1 and 6.2, the system gave us the wrong answer by finding the wrong target (false positive). In all of these cases, the cause for the false positive was a weak verification system in the recognition engine. We discuss an example scenario where the initial scene had one or more distractor objects with similar shape and color properties as the target object and the selected segments from the distractor object were recognized as belonging to the target object. Figures 6-43–6-45 describe an example of a false positive due to the weak verification system. Figure 6-43 is an example of an image where there is a plane with similar color and shape properties as the model. Figure 6-44(a) shows the segments extracted from Figure 6-43. Figure 6-44(b) shows the selected segments. Figure 6-45 shows the model aligned with the selected segments. This alignment in Figure 6-45 verified the hypothesis in Figure 6-44(b) as an instance of the target object in the scene and the recognition engine identified the wrong object as the target object due to the weak verification system.

Trials 32 and 48 in table 6.2 are examples of false negatives identified by the system. In both these trials, the system could not find the target object when it was present in the scene. In trial 32, the color of the lights in the scene was changed drastically from white to green. Our simple color algorithm (which does include a color constancy model) did not give any color regions and thus the system returned without finding the object. In trial 48, the color and the intensity of the light source were changed drastically. The color algorithm found some regions in the left and right images but missed several features on the target object in both images. The stereo algorithm couldn't find any focal feature in the left image with a match in the right image to fixate the cameras and this led to a false negative identification.

Table 6.5 summarizes the results recorded in tables 6.1–6.4 by giving the average reduction in the number of features and the number of alignments at every stage of the system.

Table 6.1: Table with results on 30 runs. The columns from left to right represent the following. *L-segs* gives the number of segments in the coarse left image, *R-segs* gives the number of segments in the right coarse image, *Col-L-segs* and *Col-R-segs* gives the number of left and right segments after color filter has been applied, *Foc-L* and *Foc-R* gives the average (over number of fixations) number of segments in the foveated left and right images, *Final* gives the number of selected segments, *Ans* says if the object was found (F) or not found(NF), *Fix.* gives the number of fixations it took to find the object, *Rt* says if the answer given by the system is correct or wrong.

No	L-segs	R-segs	Col-L-segs	Col-R-segs	Foc-L	Foc-R	Final	Ans	Fix.	Rt
1	533	406	93	61	568	288	32	F	3	Y
2	528	443	100	78	514	252	29	NF	7	Y
3	413	388	112	90	538	353	20	F	5	Y
4	505	396	180	95	507	368	22	F	7	Y
5	524	390	231	80	482	352	26	F	3	Y
6	510	412	112	100	-	-	-	NF	0	Y
7	530	398	201	64	507	400	27	F	3	N
8	545	512	120	95	517	500	30	F	2	N
9	515	490	107	98	507	380	28	F	2	Y
10	520	485	126	120	490	485	26	F	3	Y
11	524	512	134	97	509	382	32	F	2	N
12	528	434	108	78	524	408	26	F	5	Y
13	543	478	124	64	510	400	25	NF	6	Y
14	505	460	109	96	520	360	28	F	3	Y
15	516	492	104	83	507	353	33	F	1	Y
16	520	491	98	95	500	410	27	F	4	Y
17	525	465	138	76	487	398	25	F	3	Y
18	502	498	102	68	510	387	28	F	2	Y
19	531	512	86	77	-	-	-	NF	-	Y
20	504	386	108	90	516	444	29	F	4	Y
21	514	412	98	67	504	490	26	F	2	Y
22	509	399	101	85	443	404	26	F	3	Y
23	510	501	112	66	506	424	27	F	3	Y
24	508	392	78	65	482	356	29	F	2	Y
25	508	498	106	75	412	400	26	F	3	Y
26	491	376	94	33	378	320	40	NF	2	Y
27	585	492	138	239	520	519	51	F	6	Y
28	498	497	55	75	616	570	45	F	2	Y
29	624	501	172	86	594	517	61	NF	7	Y
30	534	521	100	102	483	456	21	NF	3	Y

Table 6.2: Table with more results. The columns from left to right represent the following. *L-segs* gives the number of segments in the coarse left image, *R-segs* gives the number of segments in the right coarse image, *Col-L-segs* and *Col-R-segs* gives the number of left and right segments after color filter has been applied, *Foc-L* and *Foc-R* gives the average (over number of fixations) number of segments in the foveated left and right images, *Final* gives the number of selected segments, *Ans* says if the object was found (F) or not found(NF), *Fix.* gives the number of fixations it took to find the object, *Rt* says if the answer given by the system is correct or wrong.

No	L-segs	R-segs	Col-L-segs	Col-R-segs	Foc-L	Foc-R	Final	Ans	Fix.	Rt
31	503	506	97	51	508	380	22	F	3	Y
32	518	467	-	-	-	-	-	NF	0	N
33	513	386	118	100	540	353	20	F	5	Y
34	487	490	-	-	-	-	-	NF	0	Y
35	532	494	123	90	492	382	23	F	3	Y
36	540	523	120	104	510	500	25	NF	2	Y
37	532	460	101	84	501	500	27	F	4	Y
38	515	524	85	100	517	533	32	F	2	N
39	518	480	78	98	507	480	28	F	2	Y
40	525	483	96	110	540	515	26	F	3	Y
41	534	522	114	97	519	502	22	F	2	Y
42	525	504	130	108	514	518	28	F	5	N
43	498	487	-	-	-	-	-	NF	-	Y
44	478	470	106	102	510	460	21	F	3	Y
45	540	500	120	93	485	453	30	F	1	Y
46	543	481	78	95	490	476	24	F	4	Y
47	522	470	83	76	487	498	23	F	3	Y
48	500	496	20	23	-	-	-	NF	-	N
49	511	517	76	87	532	512	42	NF	4	Y
50	490	381	-	-	-	-	-	NF	-	Y

Table 6.3: Table with results on 30 runs showing the number of alignments at the various stages. The columns from left to right represent the following. *L-coarse* gives the number of alignments in the coarse left image, *R-coarse* gives the number of alignments in the coarse right image, *Col-L* and *Col-R* gives the number of left and right alignments after color filter has been applied, *Foc-L* and *Foc-R* gives the average (over number of fixations) number of alignments in the foveated left and right images, *Final* gives the number of alignments after selection, *Ans* says if the object was found (F) or not found(NF), *Fix.* gives the number of fixations it took to find the object, *Rt* says if the answer given by the system is correct or wrong. The number of model segments is 20.

No	L-coarse	R-coarse	Col -L	Col-R	Foc -L	Foc-R	Final	Ans	Fix.	Rt
1	1.2e12	5.3e11	6.4e9	1.8e9	1.4e12	1.9e11	2.6e8	F	3	Y
2	1.1e12	6.9e11	8e9	3e9	1e12	1.2e11	1.9e8	NF	7	Y
3	5.6e11	4.6e11	1.1e10	5.8e9	1.4e12	3.5e11	6.4e7	F	5	Y
4	1e12	4.9e11	4.6e10	6.8e9	1e12	3.9e11	8.5e7	F	7	Y
5	1.1e12	4.7e11	9.8e10	4e9	8.9e11	3.4e11	1.4e8	F	3	Y
6	1e12	5.5e11	1.1e10	8e9	-	-	-	NF	0	Y
7	1.2e12	5.0e11	6.49e10	2.09e9	1.04e12	5.12e11	1.5e8	F	3	N
8	1.29e12	1.07e12	1.38e10	6.85e9	1.1e12	1e12	2.1e8	F	2	N
9	1.09e12	9.4e11	9.8e9	7.5e9	1.04e12	4.3e11	1.7e8	F	2	Y
10	1.1e12	1.1e11	1.6e10	1.3e10	9.4e11	9.1e11	1.4e8	F	3	Y
11	1.15e12	1.07e12	1.9e10	7.3e9	1.05e12	4.4e11	2.6e8	F	2	N
12	1.17e12	6.53e11	1e10	3.79e9	1.15e12	5.4e11	1.4e8	F	5	Y
13	1.28e12	8.74e11	1.53e10	2.10e9	1.06e12	5.12e11	1.25e8	NF	6	Y
14	1.03e12	7.79e11	1.04e10	7.08e9	1.12e12	3.73e11	1.76e8	F	3	Y
15	1.10e12	9.53e11	9.00e9	4.57e9	1.04e12	3.52e11	2.87e8	F	1	Y
16	1.12e12	9.47e11	7.53e9	6.86e9	1.00e12	5.51e11	1.57e8	F	4	Y
17	1.16e12	8.04e11	2.10e10	3.51e9	9.24e11	5.04e11	1.25e8	F	3	Y
18	1.01e12	9.88e11	8.49e9	2.52e9	1.06e12	4.64e11	1.76e8	F	2	Y
19	1.20e12	1.07e12	5.09e9	3.65e9	-	-	-	NF	-	Y
20	1.02e12	4.60e11	1.01e10	5.83e9	1.10e12	7.00e11	1.95e8	F	4	Y
21	1.09e12	5.59e11	7.53e9	2.41e9	1.02e12	9.41e11	1.41e8	F	2	Y
22	1.05e12	5.08e11	8.24e9	4.91e9	6.96e11	5.28e11	1.41e8	F	3	Y
23	1.06e12	1.01e12	1.12e10	2.30e9	1.04e12	6.10e11	1.57e8	F	3	Y
24	1.05e12	4.82e11	3.80e9	2.20e9	8.96e11	3.61e11	1.95e8	F	2	Y
25	1.05e12	9.88e11	9.53e9	3.38e9	5.59e11	5.12e11	1.41e8	F	3	Y
26	9.47e11	4.25e11	6.64e9	2.87e8	4.32e11	2.62e11	5.12e8	NF	2	Y
27	1.60e12	9.53e11	2.10e10	1.09e11	1.12e12	1.12e12	1.06e9	F	4	Y
28	9.88e11	9.82e11	1.33e9	3.38e9	1.87e12	1.48e12	7.29e8	F	2	Y
29	1.94e12	1.01e12	4.07e10	5.09e9	1.68e12	1.11e12	1.82e9	NF	7	Y
30	1.21e12	1.13e12	8e9	8.48e9	9.01e11	7.58e11	7.4e7	NF	3	Y

Table 6.4: Table with more results

No	L-coarse	R-coarse	Col -L	Col-R	Foc -L	Foc-R	Final	Ans	Fix.	Rt
31	$1.02e12$	$1.04e12$	$7.30e9$	$1.06e9$	$1.05e12$	$4.39e11$	$8.52e7$	F	3	Y
32	$1.11e12$	$8.15e11$	-	-	-	-	-	NF	-	N
33	$1.08e12$	$4.60e11$	$1.31e10$	$8.00e9$	$1.26e12$	$3.49e11$	$6.40e7$	F	5	Y
34	$9.24e11$	$9.41e11$	-	-	-	-	-	NF	-	Y
35	$1.20e12$	$9.64e11$	$1.49e10$	$5.83e9$	$9.53e11$	$4.46e11$	$9.73e7$	F	3	Y
36	$1.26e12$	$1.14e12$	$1.38e10$	$9.00e9$	$1.06e12$	$1.00e12$	$1.57e8$	NF	2	Y
37	$1.2e12$	$7.78e11$	$8.24e9$	$4.74e9$	$1e12$	$1e12$	$1.57e8$	F	4	Y
38	$1.09e12$	$1.15e12$	$4.91e9$	$8.00e9$	$1.11e12$	$1.21e12$	$2.62e8$	F	2	N
39	$1.11e12$	$8.85e11$	$3.80e9$	$7.53e9$	$1.04e12$	$8.85e11$	$1.76e8$	F	2	Y
40	$1.16e12$	$9.01e11$	$7.08e9$	$1.06e10$	$1.26e12$	$1.09e12$	$1.41e8$	F	3	Y
41	$1.22e12$	$1.14e12$	$1.19e10$	$7.30e9$	$1.12e12$	$1.01e12$	$8.52e7$	F	2	Y
42	$1.16e12$	$1.02e12$	$1.76e10$	$1.01e10$	$1.09e12$	$1.11e12$	$1.76e8$	F	5	N
43	$9.88e11$	$9.24e11$	-	-	-	-	-	NF	-	Y
44	$8.74e11$	$8.31e11$	$9.53e9$	$8.49e9$	$1.06e12$	$7.79e11$	$7.41e7$	F	3	Y
45	$1.26e12$	$1.00e12$	$1.38e10$	$6.43e9$	$9.13e11$	$7.44e11$	$2.16e8$	F	1	Y
46	$1.28e12$	$8.90e11$	$3.80e9$	$6.86e9$	$9.41e11$	$8.63e11$	$1.11e8$	F	4	Y
47	$1.14e12$	$8.31e11$	$4.57e9$	$3.51e9$	$9.24e11$	$9.88e11$	$9.73e7$	F	3	Y
48	$1.00e12$	$9.76e11$	$6.40e7$	$9.73e7$	-	-	-	NF	-	N
49	$1.07e12$	$1.11e12$	$3.51e9$	$5.27e9$	$1.20e12$	$1.07e12$	$5.93e8$	NF	4	Y
50	$9.41e11$	$4.42e11$	-	-	-	-	-	NF	-	Y

Table 6.5: Table summarizing the results in tables 6.1-6.4. This table gives the average number of segments in the left and right image together with the average number of alignments that have to be tried at the various stages of the system. The average number of fixations before the target object was found is 3.

Stage	Left Image	Right Image	Number of Alignments ($\times 10^8$)
Initial segments	480	450	10000
After color	108	90	100
Focal segments	500	400	10000
Selected segments	25	25	1

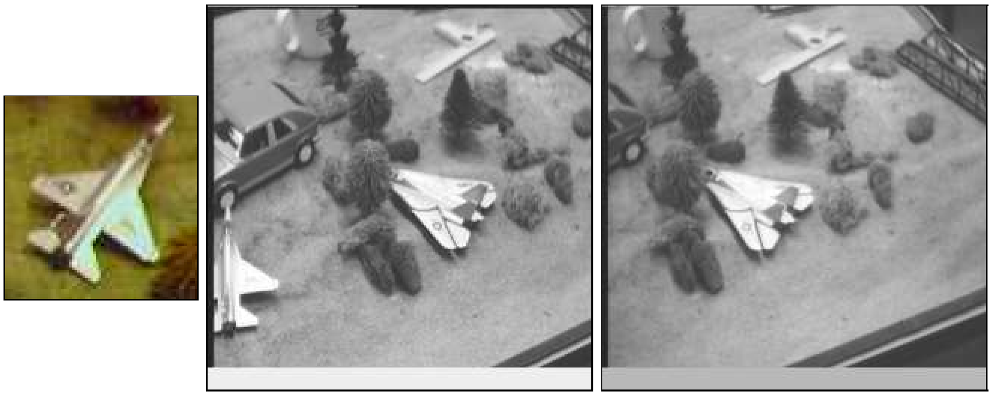

Figure 6-1: a) Target object b) Left image c) Right image

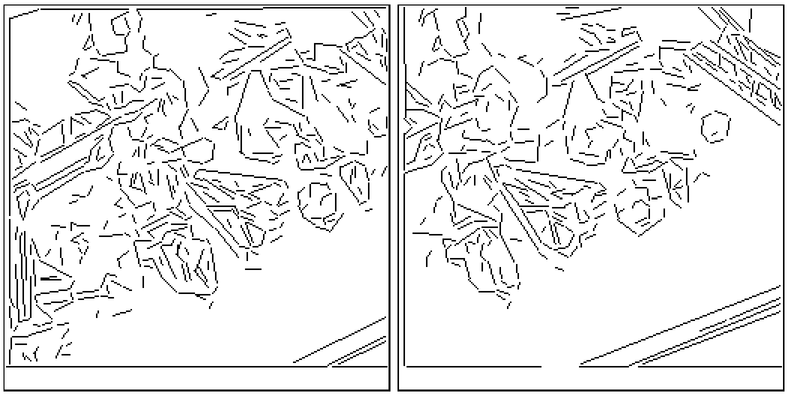

Figure 6-2: Segments from left and right images

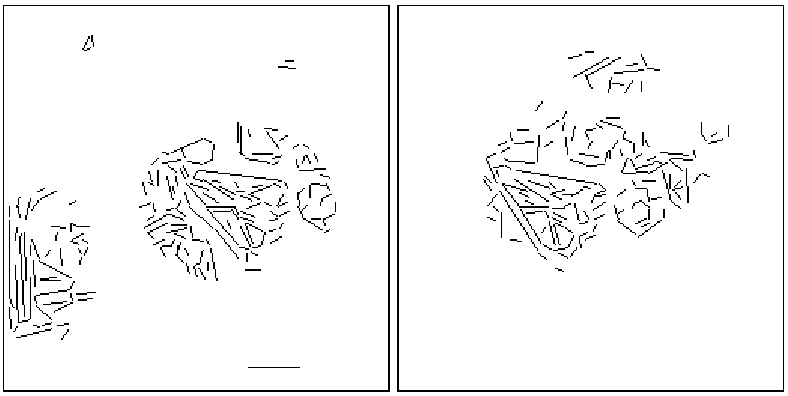

Figure 6-3: Segments from left and right images after applying color filter.

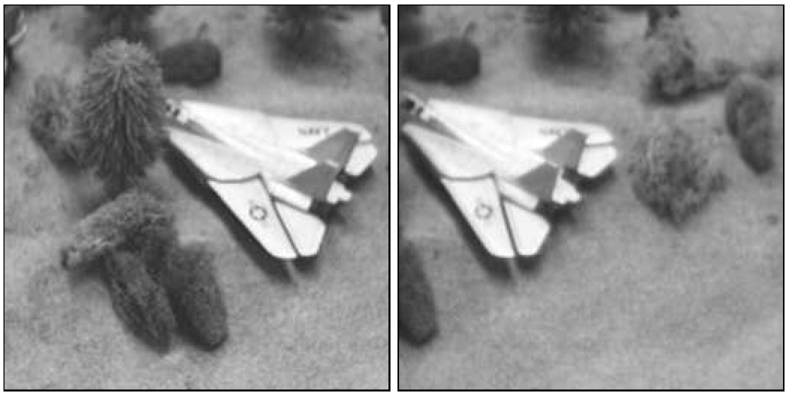

Figure 6-4: a) Foveated left gray image. b) Foveated right gray image.

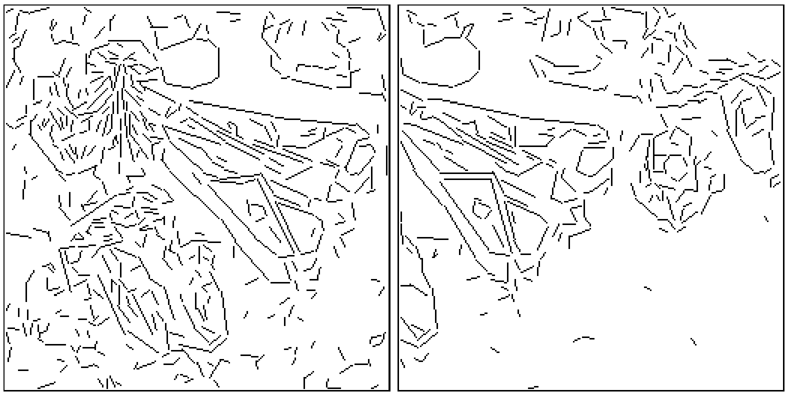

Figure 6-5: Segments in the foveated left and right images.

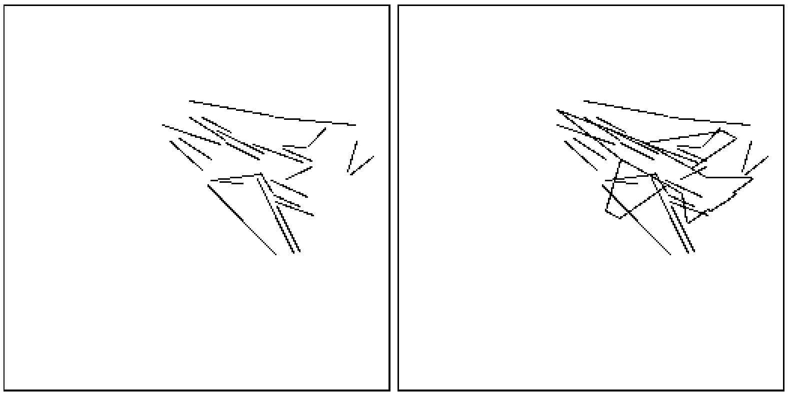

Figure 6-6: a) Selected segments b) Selected data aligned with model. As we can see, the alignment is not good enough and the object is NOT FOUND in the given scene.

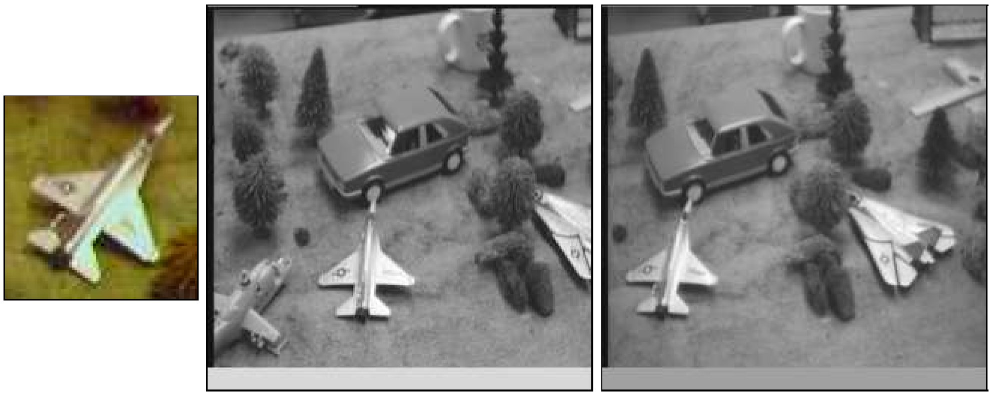

Figure 6-7: a) Target object b) Left image c) Right image

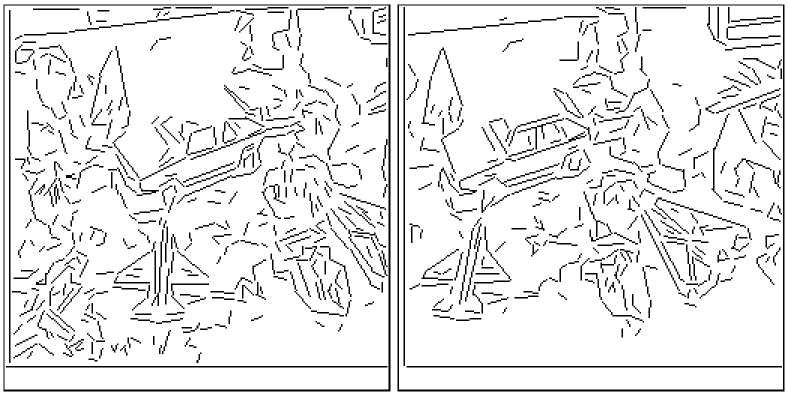

Figure 6-8: Segments from left and right images

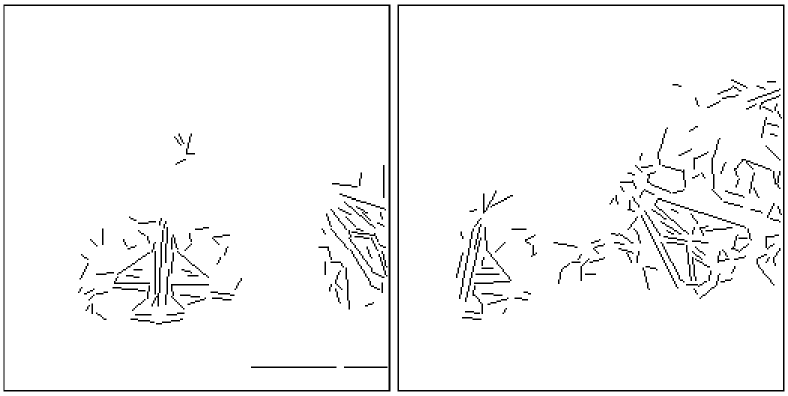

Figure 6-9: Segments from left and right images after applying color filter.

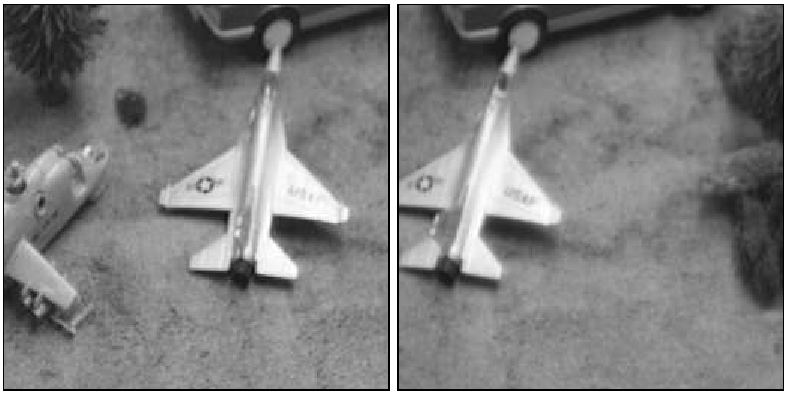

Figure 6-10: a) Foveated left image. b) Foveated right image.

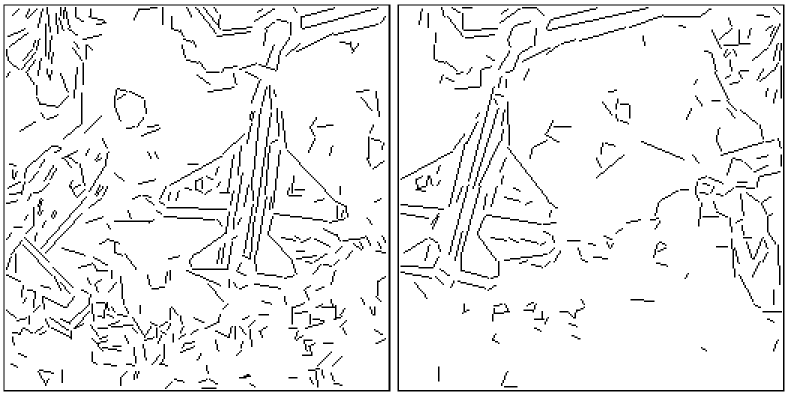

Figure 6-11: Segments in the foveated left and right images.

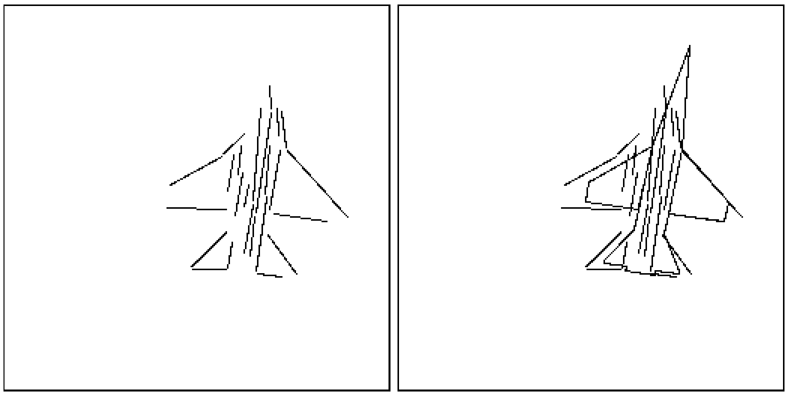

Figure 6-12: a) Selected segments b) Selected data aligned with model. As we can see, the alignment is good enough and the object is FOUND in the given scene.

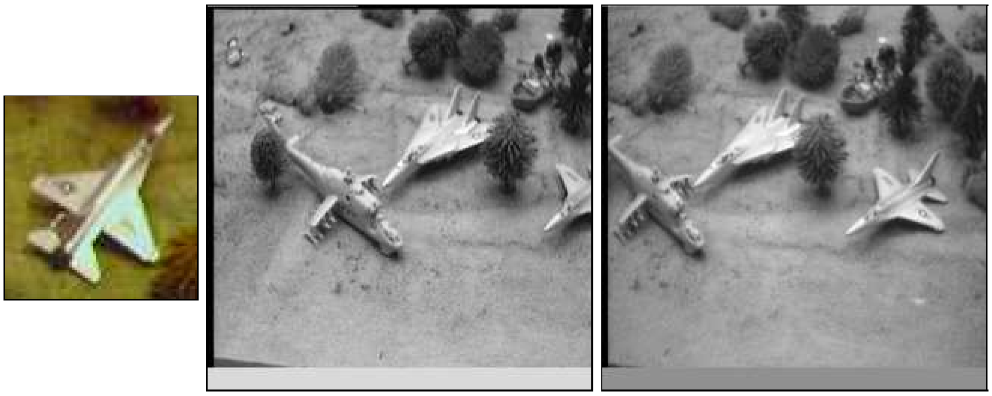

Figure 6-13: a) Target object b) Left image c) Right image

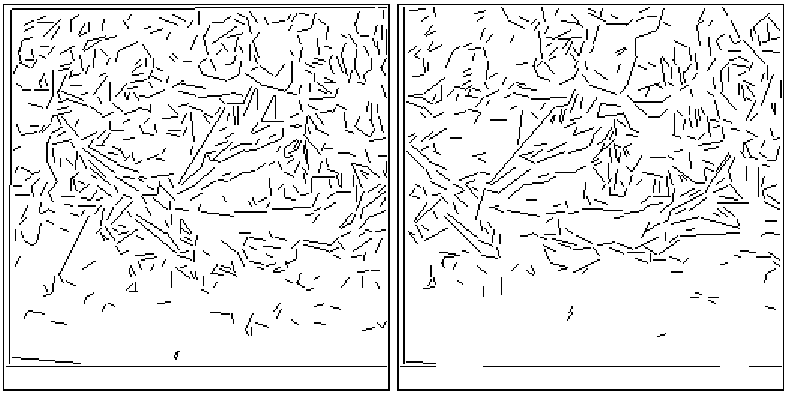

Figure 6-14: Segments from left and right images

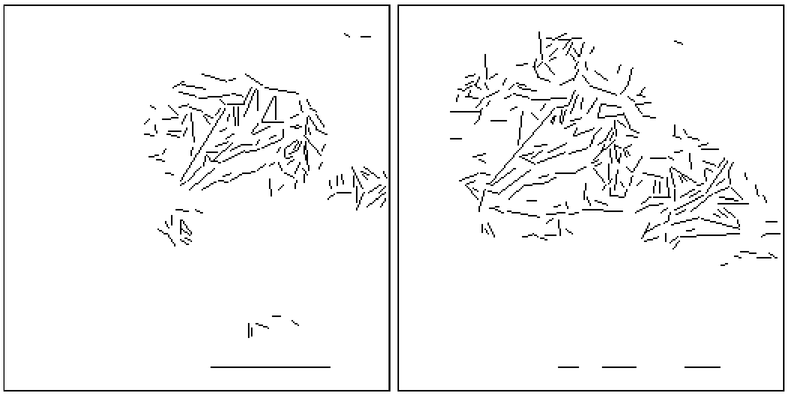

Figure 6-15: Segments from left and right images after applying color filter.

Figure 6-16: FIRST FIXATION: Foveated left and right images.

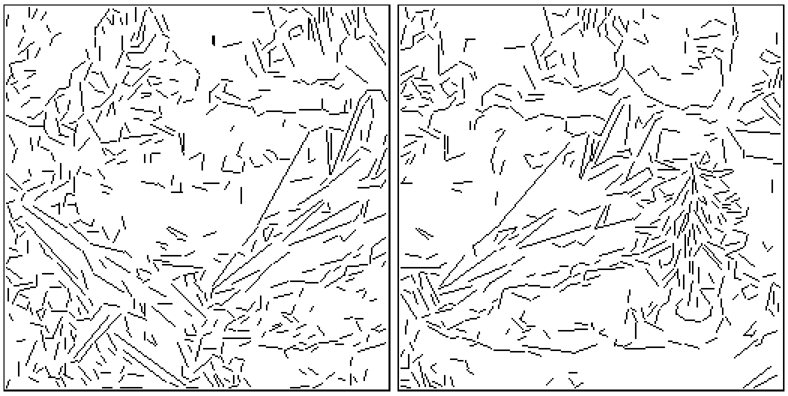

Figure 6-17: FIRST FIXATION: Segments in the foveated left and right images.

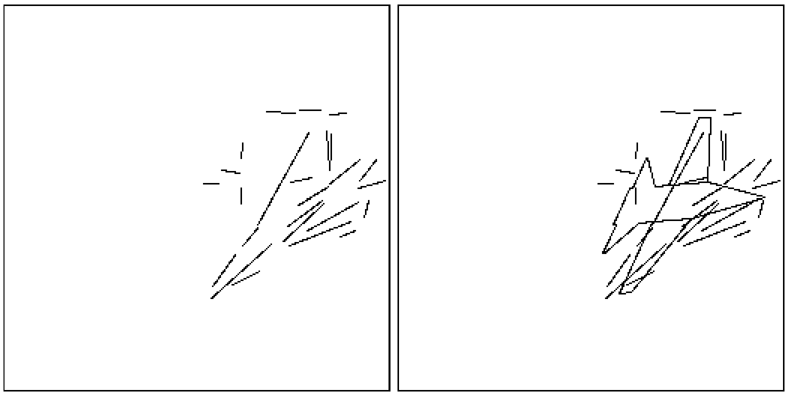

Figure 6-18: FIRST FIXATION: a) Selected segments b) Selected data aligned with model. As we can see, the alignment is not good enough and the object is NOT FOUND in the given scene.

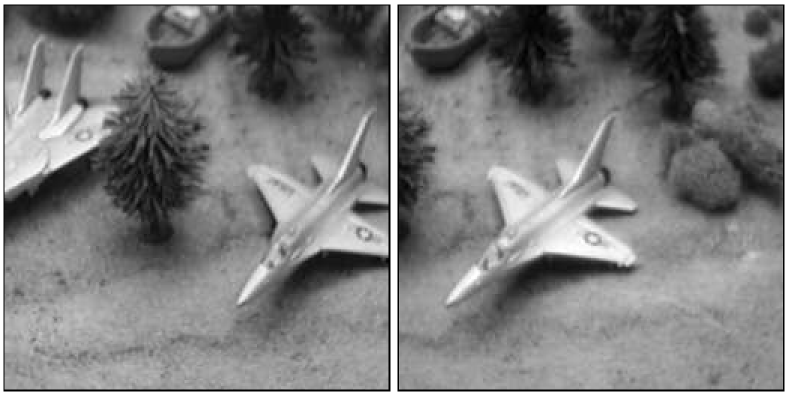

Figure 6-19: SECOND FIXATION: a) Foveated left image. b) Foveated right image.

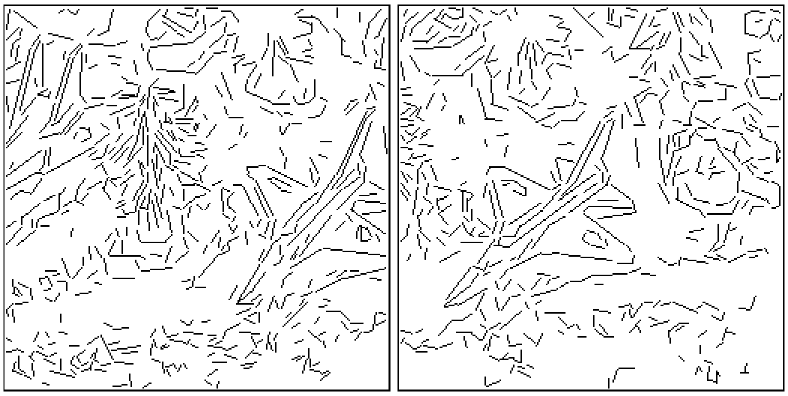

Figure 6-20: SECOND FIXATION: Segments in the foveated left and right images.

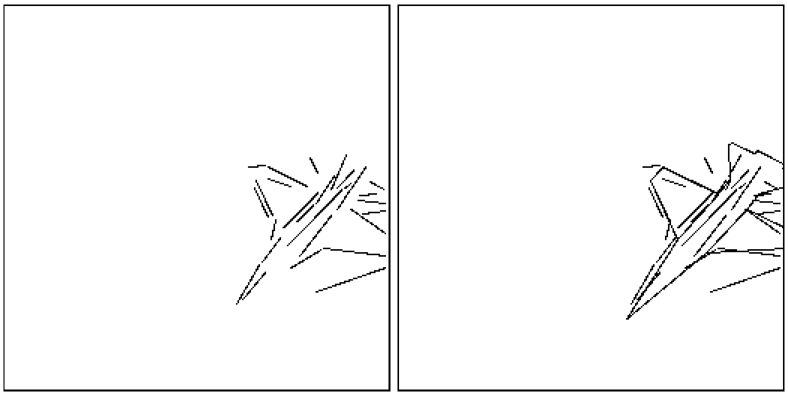

Figure 6-21: SECOND FIXATION: a) Selected segments b) Selected data aligned with model. As we can see, the alignment is good enough and the object is FOUND in the given scene.

Figure 6-22: a) The target object. b) Left image. c) Right image.

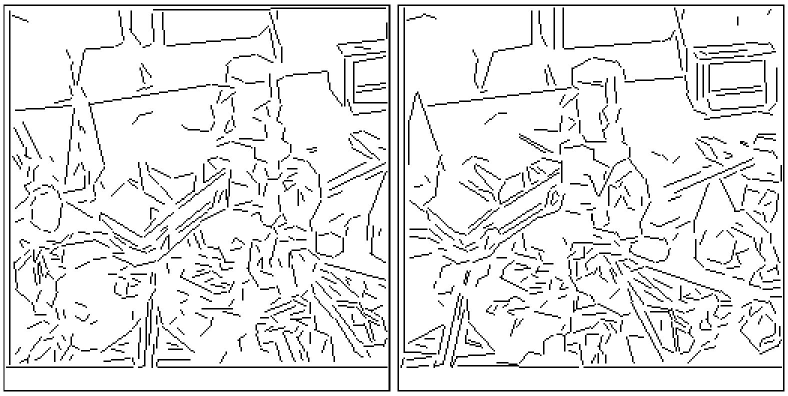

Figure 6-23: Segments from left and right images.

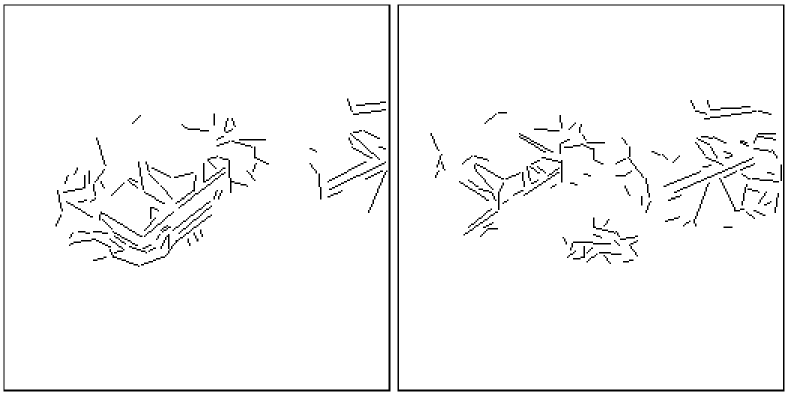

Figure 6-24: Segments from left and right images after applying color filter.

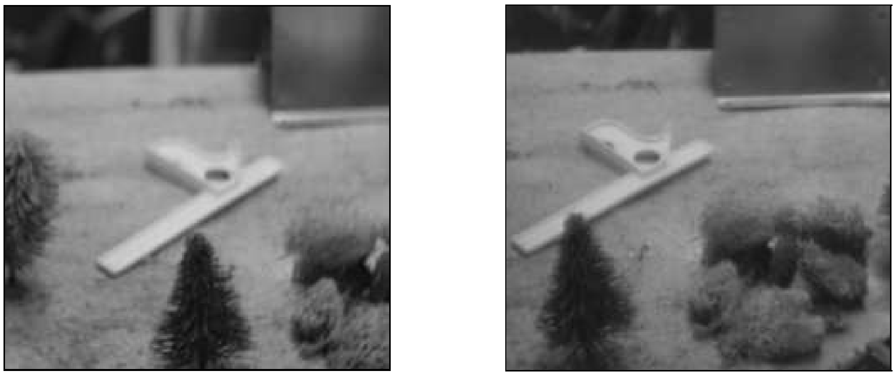

Figure 6-25: FIRST FIXATION - High resolution images extracted around a unique match from Figure 3.

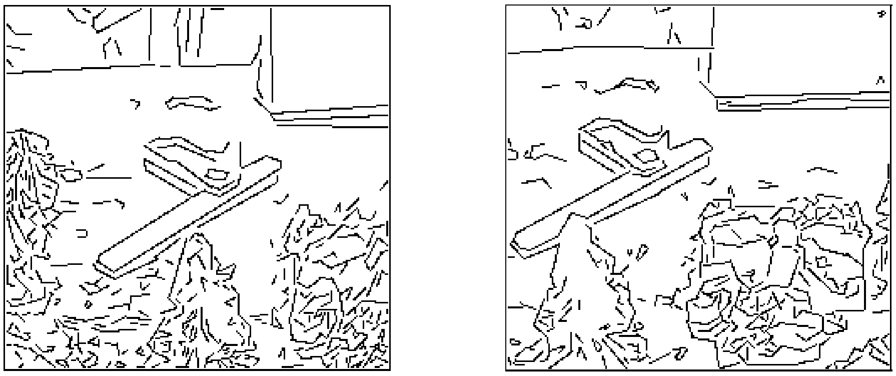

Figure 6-26: FIRST FIXATION - Segments from the high resolution images

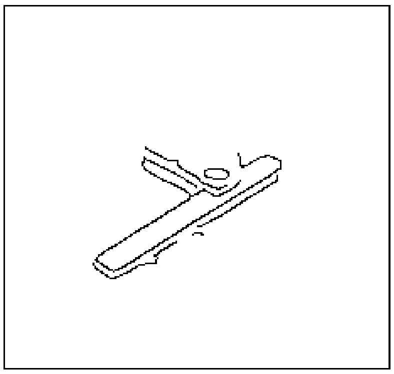

Figure 6-27: FIRST FIXATION - Selected segments. The selected segments are NOT recognized as an instance of the model. The system fixates on the next target edge.

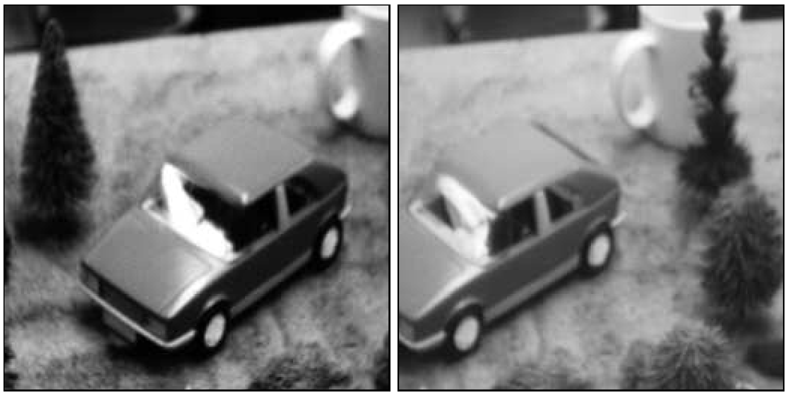

Figure 6-28: SECOND FIXATION: a) Foveated left image. b) Foveated right image.

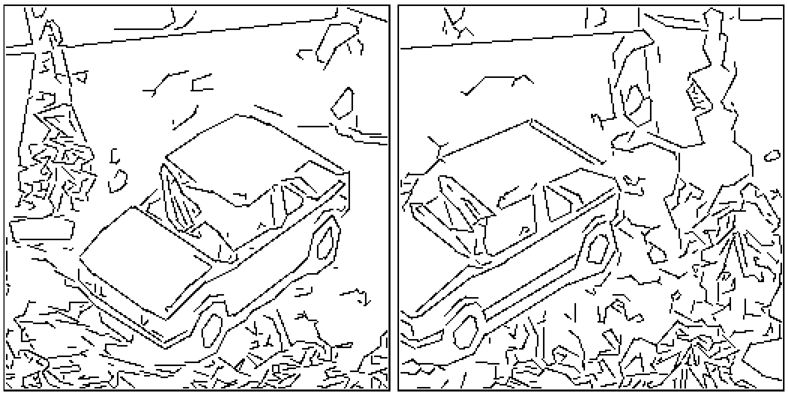

Figure 6-29: SECOND FIXATION: Segments in the foveated left and right images.

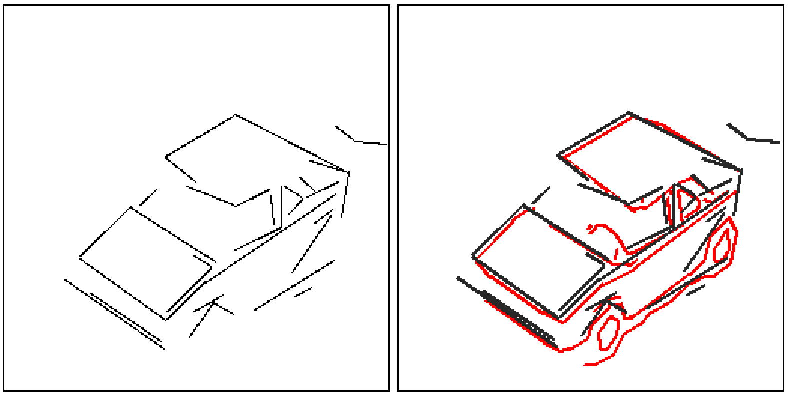

Figure 6-30: SECOND FIXATION: a) Selected segments b) Selected data aligned with model. As we can see, the alignment is good enough and the object is FOUND in the given scene.

Figure 6-31: a) Target object b) Left image c) Right image

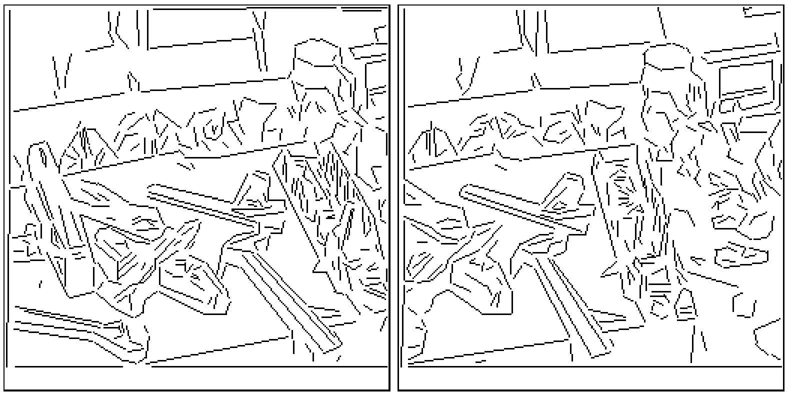

Figure 6-32: Segments from left and right images

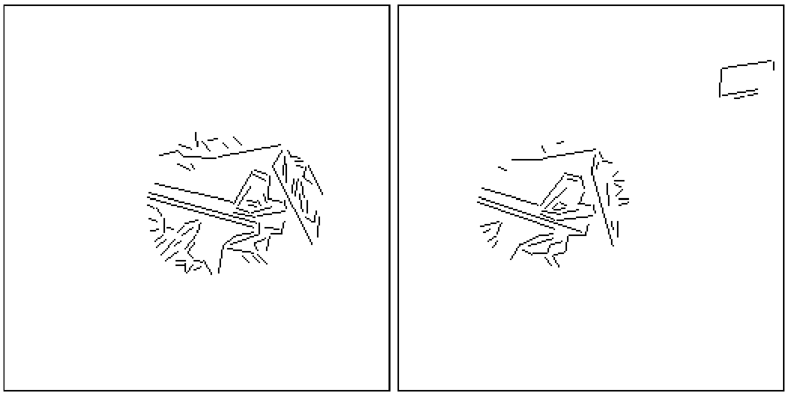

Figure 6-33: Segments from left and right images after applying color filter.

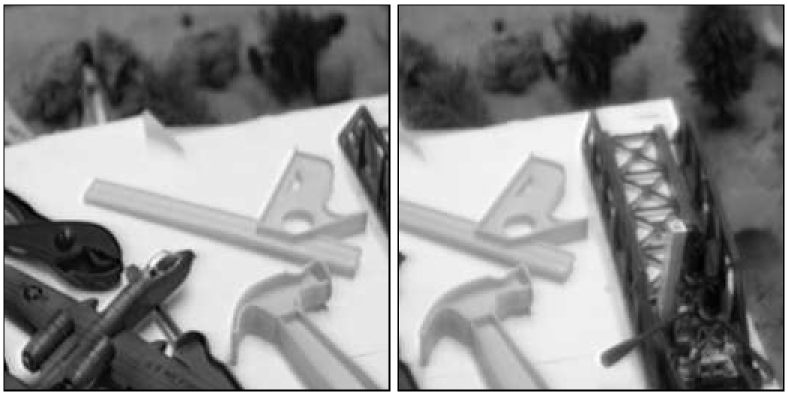

Figure 6-34: a) Foveated left image. b) Foveated right image.

Figure 6-35: Segments in the foveated left and right images.

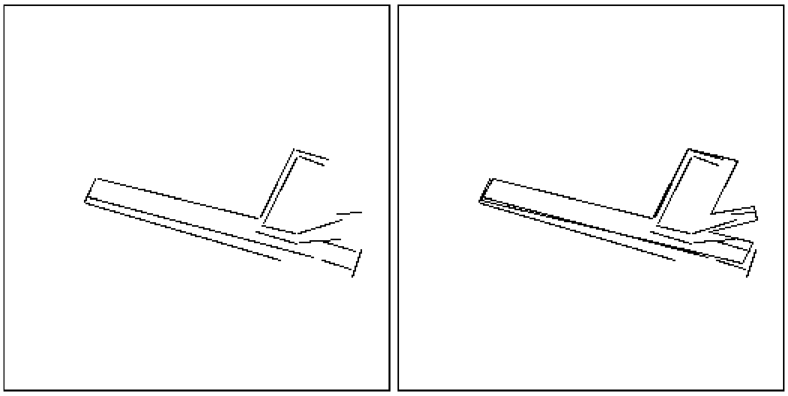

Figure 6-36: a) Selected segments b) Selected data aligned with model. As we can see, the alignment is good enough and the object is FOUND in the given scene.

Figure 6-37: a) The target object. b) Left image when color of light source is blue. c) Right image when color of light source is blue.

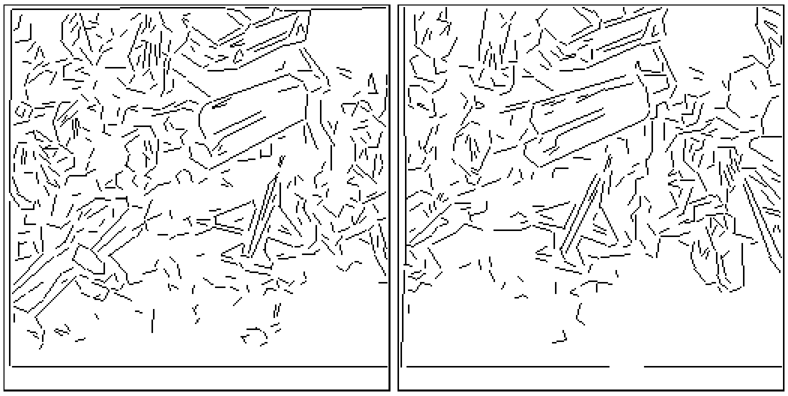

Figure 6-38: Segments from left and right images

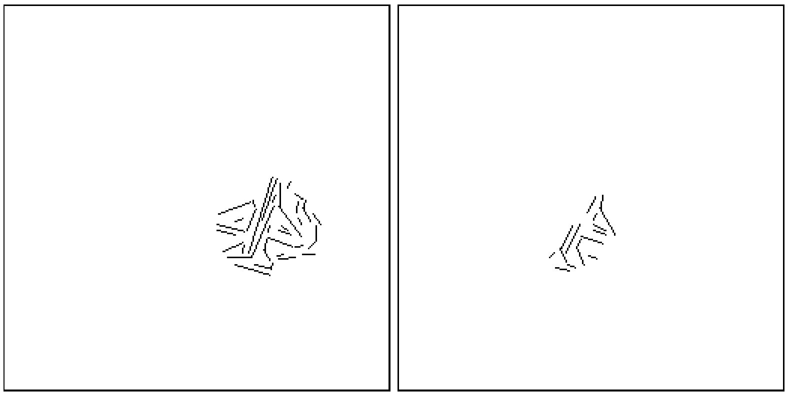

Figure 6-39: Segments from left and right images after applying color filter. Note that the color filter misses many segments on the object due to the change in the color of the light source.

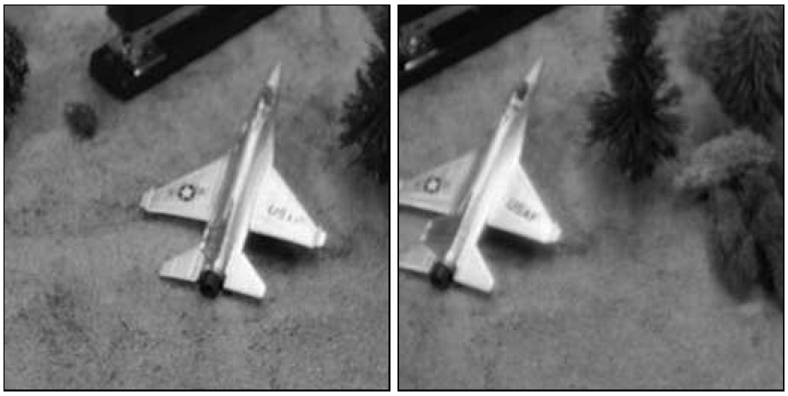

Figure 6-40: a) Foveated left image. b) Foveated right image.

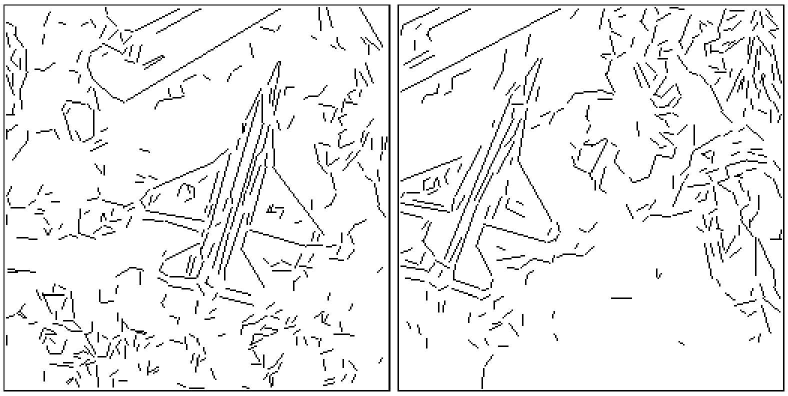

Figure 6-41: Segments in the foveated left and right images.

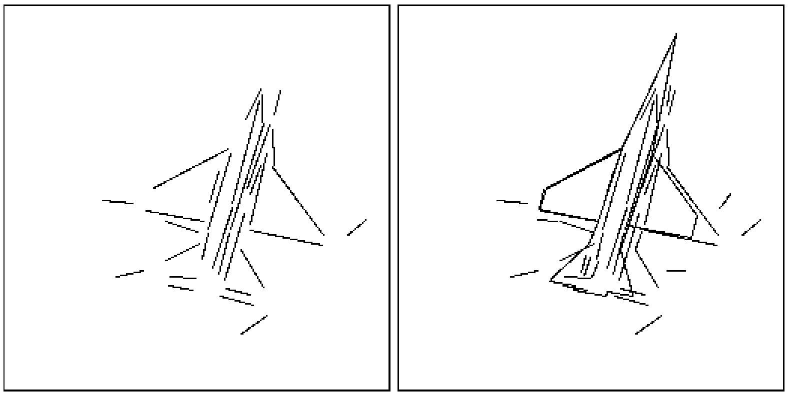

Figure 6-42: a) Selected segments b) Selected data aligned with model. As we can see, the alignment is good enough and the object is FOUND in the given scene.

Figure 6-43: Image with a distractor plane of similar color and shape as the model plane.

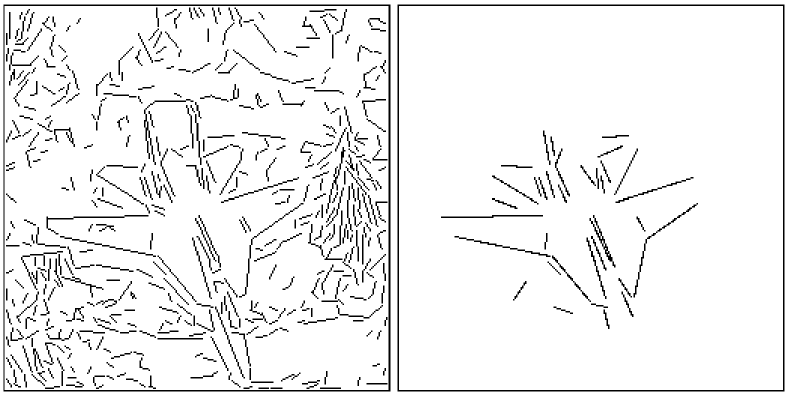

Figure 6-44: a) The segments in the image. b) The selected segments.

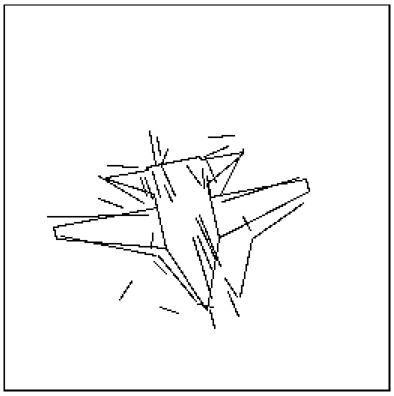

Figure 6-45: The model aligned with the selected segments. The verification system accepted this as a good alignment and gave us a false positive.

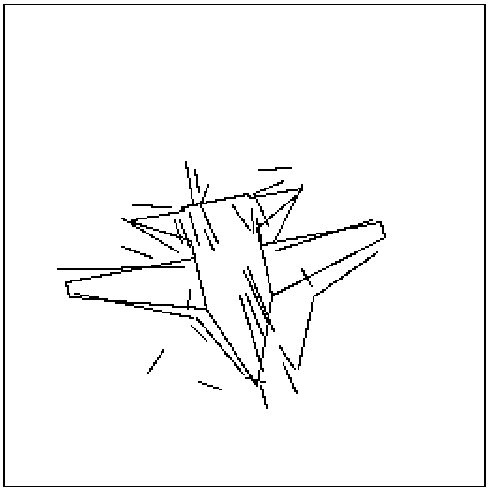

Figure 6-46: Example of a false positive.

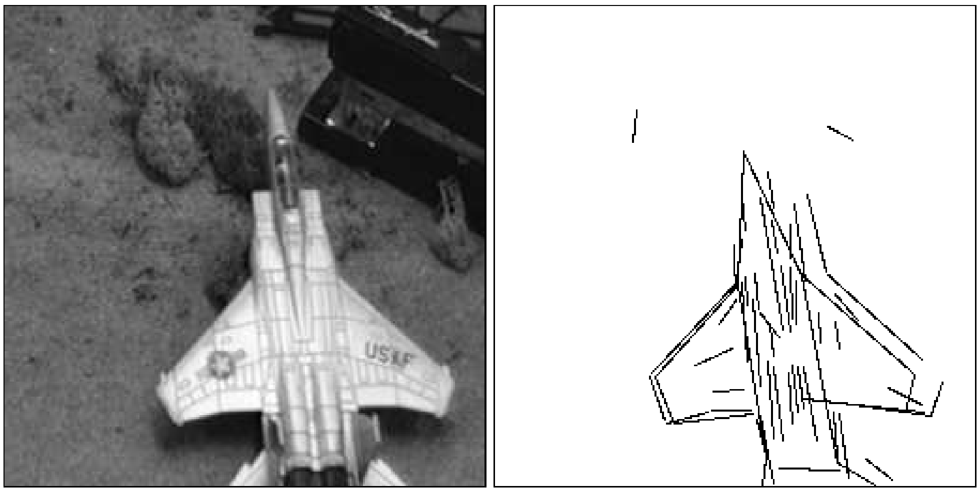

Figure 6-47: Example of a false positive.

6.3 False positives and negatives

The reasons why the system gives false positives and false negatives are the following:

- The verification scheme is not good enough and we get false positives when the shapes of the objects are similar as shown in figures 6-46 and 6-47. The current verification algorithm checks to see if the transformed model, aligned with the selected data segments, crosses a certain threshold based on the criteria discussed in Chapter 5. If the alignment score is greater than some threshold, it means that the hypothesized data segments represent an instance of the model in the image. The problem with this verification scheme is in setting a good threshold so that the presence of the target object is verified whenever it is present in the scene and other objects are rejected. If there is an object in the scene that is similar in shape to the target object (e.g. Figure 6-44), then differentiating between the object and the target object is a difficult task for our current verification scheme since the alignment scores of both objects will pass the threshold. Figure 6-46 is an example of the verifier accepting the wrong object as the target object. Figure 6-47 is another example of a false positive. In this case, the set of selected features from the distractor object that were fed into the recognition engine were incorrectly verified as representing an instance of the target object in the image.

- The system gives false negatives if the selection fails to locate any instance of the object in the image. For example, if the lighting conditions in the room change drastically and the color of the target object in the scene appears very different from the modeled colors, our simple color algorithm does not find any regions and the system returns without finding the target object. The system also fails to find the object in the scene if the stereo algorithm is unable to find matches between the left and right images. This occurs when the object appears very different in the left and right images or when the object is occluded in one of the images.

- The system gives false negatives in the case of severe occlusions when more than half of the bounding contours are missing. In this case there is not enough evidence to show that the selected segments represent an instance of the target object in the image. For example, if only 10% of the target object is visible in the image and all the visible features are selected as belonging to the target object, then all the selected segments may align perfectly with the transformed model but there will be a large number of model segments without corresponding data segments. Since the verifier needs a certain fraction of model segments to be aligned with data segments and a certain fraction of data segments to be aligned with model segments to accept a match, it rejects the selected segments in the above example and the system fails to locate the instance of the object in the image.

Chapter 7

Summary and Conclusions

7.1 Summary

In this project, we have attempted to show that focus of attention and fixation play an important role in selecting out candidate regions in the scene that could contain a target object in model-based recognition. In Chapter 1, we began with a discussion of the effect of scene clutter on recognition systems. We saw that scene clutter impedes the performance of recognition methods (e.g. [17], [29], [64] among others) and also contributes to the number of false alarms that have to be handled. We argued using previous results [18] that a key component in efficient object recognition is selection or figure/ground separation before model matching. We discussed how selection produces features in the scene that are likely to come from a single object (the target object) with minimal amount of spurious data, and how these selected features can be filtered by the recognition system to isolate the instance of the target object exactly. We then went on to show that effective and efficient selection can be achieved when several independent cues are used in conjunction.

In this project, we have used visual attention mechanisms [59] to integrate the visual cues of color and stereo in order to perform selection and focus the resources of the recognition engines onto relevant data subsets and we have used active vision techniques to direct the selection process. We have built an active-attentive vision system to support the higher level task of model-based object recognition.

Specifically, we have built a system that searches for objects in a cluttered room. The system uses color and stereo as visual cues to select out candidate regions in the scene that could contain the target object and these regions are fed into an Alignment-style[1] recognition engine which verifies whether or not the object is present in the selected region. The system illustrates how simple color measures can be used to roughly segment the image into regions that are likely to contain the target object. It also shows that stereo can be used effectively as a figure/ground separator without the need for explicit depth calculations and accurate camera calibration [24]. The results in Chapter 6 show that the system performs reliably in cluttered scenes with different objects under varying lighting conditions. Thus, this system demonstrates a method for doing efficient selection which reduces the complexity of the recognition process significantly and keeps the false identifications under control.

While we have shown that our system can find a target object correctly in a cluttered indoor scene, there are still places for improvement. These include:

- We can further reduce the false positives by improving the verification method for the recognition system.
- We can add additional cues like rough texture measures in the selection process to improve performance in scenes which have little color information.
- We can refine the final pose of a solution further by ranking the alignment features and using the best features in a least squares minimization.
- We could also use other features besides edges (e.g. centroids of data clusters) in alignment [49].
- We can take advantage of additional constraints, like a rough estimate of ground plane for example, to make the recognition more robust.

[1] Alignment-style recognition techniques ([29], [64]) find a small number of corresponding features between the model and the image to compute the transformation that aligns the model with a hypothesized instance of the object in the image and then verifies the hypothesis by comparing the transformed model with the image data.

7.2 Future Directions

We have shown that a system that selects features from a target object quickly and reliably in a scene is useful in controlling the explosive search involved in recognition. Such a system can be applied directly to a binocular robot moving around in the environment to help it recognize landmarks, avoid obstacles and perform other tasks which require recognizing specific objects in the environment. An active-attentive vision system is more robust and computationally efficient than a static vision system on a mobile robot since it allows the robot to change its visual parameters to acquire relevant information from the scene to solve the specific task that it has at the time. A mobile robot with an active vision system also has the ability to obtain multiple views which helps greatly in performing model-based object recognition. We would like to use our system on a mobile, binocular, vision-based robot that is required to recognize and fetch objects in the environment. The system would enable the robot to use multiple cues to focus its attention on relevant visual information in the scene in order to recognize target objects efficiently.

While the current system has been used in recognition of objects using video images, we could also extend it to other kinds of images (e.g. SAR images) in applications like automatic target recognition where the system has to analyze large amounts of data. Even though visual cues like color and stereo may not be applicable in this domain, the principle of focus of attention on relevant data subsets can be used effectively to locate targets quickly and reliably.

Bibliography

[1] Y. Aloimonos, "Active Perception", *Lawrence Erlbaum Assoc., Publishers*, 1993.

[2] A.L. Abbott and Narendra Ahuja, "Surface reconstruction by dyanamic integration of focus, camera vergence, and stereo", *ICCV* 1988.

[3] N. Ayache and B. Faverjon, "Efficient registration of stereo images by matching graph descriptions of edge segments", *IJCV 1(2)*, 107-131, 1987.

[4] R. Bajcsy, "Active Perception vs Passive Perception", *Proc. Third IEEE Workshop on Computer Vision*, 55 - 59, Oct 1985.

[5] R. Bajcsy and M. Campos, "Active and Exploratory Vision", *CVGIP: Image Understanding*, Vol. 56, 31-40, July 1992.

[6] D. Ballard, "Animate Vision." *Artificial Intelligence*, Vol 48, 57-86, 1991

[7] D. Ballard, "Eye Movement and Visual Cognition", *Proc. Work. on Spatial Reasoning and Multi Sensor Fusion*, 1987

[8] M. Bober, P.Hoad, J.Mataas, P. Remagnino, J.Kittler, J. Illingworth, "Control of perception in an Active Vision System: Sensing and Interpretation", *IROS*, 1993.

[9] R.C. Bolles and R.A. Cain, "Recognizing and locating partially visible objects: The local feature-focus method", *International Journal of Robotics Research*, 1(3):57-82 48, 1982.

[10] C. Brown, "Progress in Image Understanding at University Of Rochester", *DARPA IU Workshop*, 73-77, 1988.

[11] W. Ching, P. Toh, K. Chan and M. Er, "Robust Vergence with Concurrent Detection of Occlusion and Specular Highlights", *ICCV*, 384 - 394, 1993.

[12] R. Deriche and G. Giraudon, "Accurate corner detection: An analytic study", *ICCV*, 66-70, 1990.

[13] F. Ennesser and G. Medioni, "Finding Waldo, or Focus of Attention Using Local Color Information", *CVPR* 711-7112, 1993.

[14] O.D. Faugeras, "What can be seen in three dimensions with an uncalibrated camera rig?", *Second ECCV*, Italy, 563-578, 1992.

[15] J. J. Clark and N. Ferrier, "Modal control of an attentive vision system", *Second ICCV*, 524 - 523, 1988.

[16] D.J. Coombs, "Real Time Gaze Holding in Binocular Robot Vision", *PhD. Thesis, Univ Of Rochester*, 1992

[17] W.E.L. Grimson, "The combinatorics of object recognition in cluttered environments using constrained search", *Proc. of the International Conference on Computer Vision*, 1988.

[18] W.E.L. Grimson, "Object Recognition by Computer: The Role of Geometric Constraints." *Cambridge: MIT Press*, 1990.

[19] W.E.L. Grimson, "A computer implementation of a theory of human stereo vision", *Phil. Trans. Roy. Soc. London*, vol B 292, 217-253, 1981.

[20] W.E.L. Grimson, "Computational experiments with feature based stereo algorithm", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 7, Jan 1985.

[21] W.E.L. Grimson and T. Lozano-Perez, "Model based recognition and localization from sparse range or tactile data", *Intl. Journal Of Robotics Research* , 3(3): 3 - 35, 1984.

[22] W.E.L. Grimson and T. Lozano-Perez, "Localizing overlapping parts by searching the interpretation tree", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):469-482, July 1987.

[23] W.E.L. Grimson and D.P. Huttenlocher, "On the sensistivity of the Hough Transform for Object Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):255-274, July 1990.

[24] W.E.L Grimson, "Why stereo vision is not always about 3D reconstruction", *MIT AI Lab Memo 1435*, 1993.

[25] W.E.L. Grimson, A. Lakshmi Ratan, P.A. O'Donnell, G. Klanderman, "An Active Visual Attention System to Play "Where's Waldo"?", *Proc. of the Workshop on Visual Behaviors*, 85-90, 1994.

[26] I.D. Horswill, "Polly: A vision based Artificial Agent", *Eleventh Natl. Conf on AI*, 824 - 829, 1993.

[27] R. Horaud and T. Skordas, "Stereo correspondence through feature grouping and maximal cliques", *IEEE Pattern Analysis and Machine Intelligence*, **11**, 1168–1180, 1989.

[28] A.Hurlbert and T. Poggio, "Visual attention in brains and computers", *Technical Report, Aritificial Intelligence Lab, M.I.T., AI-Memo-915*, June 1986.

[29] D.P. Huttenlocher and S.Ullman, "Object Recognition Using Alignment", *Proc. First Intl. Conf.Comp. Vision*, 109-111, 1987.

[30] D.P. Huttenlocher, "Three-dimesional recognition of solid objects from a two-dimensional image", PhD thesis, Artifical Intelligence Lab, MIT, *AI-TR-1045*, 1988.

[31] N.H. Kim and A.C. Bovik, "A contour based stereo matching algorithm using disparity continuity", *Pattern Recognition*, **21**, 515–524, 1988.

[32] L. Kitchen and A. Rosenfeld, "Gray-level corner detection", *Pattern Recognition Letters*, 95-102, December 1982.

[33] G.J. Klinker, S.A. Shafer, and T. Kanade, "Using a color refection model to seperate highlights from object color", *ICCV*, 1987.

[34] C. Koch and S. Ullman, "Selecting one among the many: A simple network implementing shifts in selective visual attention", *Technical Report, Aritificial Intelligence Lab, M.I.T., AI-memo-770*, Jan 1984.

[35] E.Krotkov, K. Henriksen and R. Kories, "Stereo ranging with verging cameras", *IEEE Trans. on Pattern Analysis and machine Intelligence*, 1489-1510, 1989.

[36] R.K. Lenz and R.Y. Tsai, "Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3-D Machine Vision Metrology", *IEEE PAMI*, 10(5), 713-720, 1988.

[37] D.Marr, "Vision", *Freeman :SanFrancisco*, 1982.

[38] J.E.W. Mayhew and J.P. Frisby, *3D Model Recognition from Stereoscopic Cues*, MIT Press, 1991.

[39] E.S. McVey and J.W. Lee, "Some accuracy and resolution aspects of computer vision distance measurements", *IEEE Pattern Analysis and Machine Intelligence* **4**(6):646–649, 1982.

[40] S.B. Pollard, J.E.W. Mayhew, and J.P. Frisby, "PMF: A stereo correspondence algorithm using a disparity gradient limit", *Perception* **14**, 449–470, 1985.

[41] D.Marr and T. Poggio, "A Theory Of Human Stereo Vision", *Proc. Royal Society Of London B 204*, 301-328.

[42] L.T. Maloney and B. Wandell, "Color constancy: a method for reccovering surface spectral reflectance", *J. Opt. Soc. Amer. A*, 3(1), 29-33.

[43] J.E.W. Mayhew and J.P. Frisby, "Psychophysical and computational studies towards a theory of human stereopsis", *Artificial Intelligence* **17**(1–3): 379–386, 1981.

[44] J.E.W. Mayhew and J.P. Frisby, *3D Model Recognition from Stereoscopic Cues*, MIT Press, 1991.

[45] E.S. McVey and J.W. Lee, "Some accuracy and resolution aspects of computer vision distance measurements", *IEEE Pattern Analysis and Machine Intelligence* **4**(6):646–649, 1982.

[46] G. Medioni and R. Nevatia, "Segment-based stereo matching", *Computer Vision, Graphics, and Image Processing*, **31**: 2–18, 1985.

[47] C.L. Novak and S.A. Shafer, "Supervised color constancy using a color chart", *Carnegie Mellon University, Technical Report*, CUM-CS-90-140.

[48] D. Shoham and S. Ullman, "Aligning a model to an image using minimal information", *Proc. 2nd Intl. Conf. of Computer Vision*, 1988.

[49] K. Nagao and W.E.L. Grimson, "Object Recognition by Alignment using Invariant Projection of Planar Surfaces", *Intl. Conf. on Computer Vision*, Vol. 1, 861-864, 1994.

[50] Olson, "Stereopsis Of Verging Systems", *ICCV*, June 1993.

[51] K. Pahlavan and J. Eklundh, "A Head-Eye System - Analysis and Design", *CVGIP: Image Understanding*, Vol 56, 41-56, 1992.

[52] K.Pahlavan, T.Uhlin, J.O.Eklundh, "Dyanamic Fixation", *Fourth ICCV* 412 - 419, 1993.

[53] T. Pavlidis, *Structural Pattern Recognition*, Springer-Verlag: New York, 1977.

[54] T. Poggio et al., *The MIT Vision Machine, AI Tech Report*

[55] L. Robert and O.D. Faugeras, "Curve-based Stereo: Figural Continuity and Curvature", *CVPR*, 57–62, 1991.

[56] I. D. Reid and D. W. Murray, "Tracking Foveated Corner Clusters using Affine Structure", *ICCV*, 76 - 83, 1993.

[57] W.S. Rutowski and A. Rosenfeld, "A comparison of corner detection techniques for chain coded curves", *Technical Report*, Univ. Of Maryland, Tech. Report No. 263, 1977.

[58] M.J. Swain and D. H. Ballard, "Indexing via Color Histograms", *ICCV*, 623 - 630, 1990.

[59] T.F. Syeda Mahmood, *AI Tech Report*, 1992.

[60] T.F. Syeda Mahmood, "Data and Model driven Selection using color regions", *Proceedings of the European Conference on Computer Vision*, 321-327, 1992.

[61] M. Tistarelli and G. Sandini, "Dyanamic Aspects of Active Vision", *CVGIP: Image Understanding*, Vol 56, 108-129, July 1992.

[62] A. Treisman, "Selective Attention in Man", *Brit. Med. Bulletin.*, 20:12-16, 1964.

[63] R.Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses", *IJRA*, 3(4), 323-344, 1987.

[64] S.Ullman and R.Basri, "Recognition by Linear Combination of Models", *IEEE Pattern Analysis and Machine Intelligence*, 13(10), 992 - 1006, 1991.

[65] L.B. Wolff, "Accurate Measurement of Orientation from Stereo using Line Correspondence", *CVPR*, 410-415, 1989.

[66] M.S. Wu, and J.J. Leou "A Bipartite Matching Approach for Feature Correspondence in Stereo Vision", *Pattern Recognition Letters*, Vol. 16, No. 1, 23–31, 1995.