An Adaptive Framework for Sensor Planning in a Coordinated Multi-agent Environment

by

Lovell Hodge

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Systems Design Engineering

Waterloo, Ontario, Canada, 2001

0-612-65247-5

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

The objective of this research is to develop an automated system for multiple sensor planning based on the coordinated decisions of independent, intelligent agents. The problem domain is such that a single sensor system might not be able to provide adequate information for a given sensor task. Hence, it is necessary to incorporate multiple sensors in order to obtain complete information. The overall goal of the system is to perform feature inspection on one or more target features within a static modeled environment. In this system, the sensors are mobile, each agent controls the position of a sensor and each agent has the ability to communicate with other agents in the environment.

The system includes a case based reasoning system that enables the agents to learn previous sensor arrangements and apply them to similar scenes. This decreases the amount of communication that is necessary to arrive at a solution. The agents may be trained off-line if necessary, but are also quite capable of learning cases online.

The experiments demonstrate the feasibility of the system when using multiple mobile cameras as the sensor suite. Each camera is controlled by an agent and the vision task is the coverage of one or more target objects in a cluttered scene.

The system provides an efficient and reliable method to accomplish the sensor

planning necessary to facilitate such tasks as feature inspection and feature detection of stationary targets. The use of agents as autonomous controllers provides a level of re-usability and scalability not normally found in other sensor planning systems. Such a system may be used in environments where the deployment of sensors needs to be an automated process due to potential hazards or where the configuration of the system needs to be changed frequently.

Acknowledgements

I would like to thank my supervisor Professor Mohamed Kamel for his encouragement, guidance and support. With his supervision, this experience was both rewarding and pleasant.

I am indebted to Professors Robin Cohen and Andrew Wong from the University of waterloo, Professor Hamada Ghenniwa from University of Western Ontario and Professor Tarek Sobh from the University of Bridgeport for their advice and constructive criticisms.

My sincere gratitude goes to my wife Andrea for her patience and support and my daughter Alana whom, although too young to understand, is a constant source of inspiration. A special thanks go to my parents in law, Betty and Terrence who never doubted my ability to accomplish this undertaking. To Joan Daniel whose support was never ending and never wavering, my sincere thanks.

To Millar Drummond, Toni Sattolo and Rolf Normandin who provided me with the opportunity for success in life, thank you.

I would like to extend a special thanks to my colleagues at the PAMI lab at the University of Waterloo for their help and support. Special thanks are due to Nayer Wanas.

Finally, I would like to acknowledge the financial support that was provided by the University of Waterloo and Materials and Manufacturing of Ontario.

Dedication

This thesis is dedicated to the loving memory of my Grandparents, John and Christine Hodge. Your love will be with me forever.

Contents

1 Introduction **1**

1.1 Sensor Planning 1

1.2 Agents 5

1.3 Multi-agent Environments 6

1.4 Statement of the Research Problem 8

1.5 Organization and Scope of the Thesis 10

2 Literature Review **11**

2.1 Introduction 11

2.2 Sensor Planning 11

2.2.1 The Synthesis Approach 14

2.2.2 The Generate and Test Approach 17

2.2.3 The Expert System Approach 24

2.2.4 Agent Based Systems 25

2.3 Discussion of the Different
Approaches to Sensor Planning 31

3 Background 36

3.1 Introduction . . . 36

3.2 Viewpoint Parameters . . . 36

3.2.1 Depth of Field . . . 37

3.2.2 Resolution . . . 40

3.2.3 Field of View and Visibility . . . 45

3.3 Multi-Agent Systems . . . 51

3.3.1 Decision Theoretic Agents . . . 57

4 The Model 61

4.1 Introduction . . . 61

4.2 The CAD Model . . . 62

4.3 Camera Viewpoints . . . 65

4.4 Data Generation . . . 66

4.5 The Agent Model . . . 68

4.5.1 The Sensor Model . . . 70

4.5.2 Action Mechanism . . . 71

4.5.3 The Knowledge Base . . . 72

4.5.4 The Communication Mechanism . . . 78

4.6 The Decision Module . . . 82

4.6.1 The Mental Model . . . 84

4.6.2 The Action History . . . 84

4.6.3 The Coordination Algorithm . . . 85

4.6.4 The Conflict Resolution Mechanism . . . 94

4.7 The Coordination Algorithm 97
4.8 Theoretical Basis for Agent Behaviour 100
4.9 Case Based Learning 101
4.10 Summary 105

5 Experimental Results **106**

5.1 Introduction 106
5.2 Single Target Coverage 107
5.2.1 Increasing the Number of Cameras 119
5.3 Multi-Target Coverage 122
5.3.1 Agent Communication 130
5.3.2 Increasing the number of Cameras 131
5.4 Learning to Improve Efficiency 137
5.4.1 The Case Based Reasoning System 137
5.5 Discussion 152

6 Conclusions and Future Research **155**

6.1 Contributions 156
6.2 Limitations 159
6.3 Future Research 160

A Object DXF Representation **164**

B Agent Data Generation **167**

C Inter-agent Interaction **169**

Bibliography 172

List of Tables

4.1 Data for Viewpoint 1 75

4.2 Data for Viewpoint 2 76

5.1 Initial Positions of Cameras 116

5.2 Final Camera Positions 116

5.3 Best Results from the Exhaustive Search 117

5.4 Final Camera Positions 121

5.5 Final Camera Positions Using 2 Cameras 129

5.6 Final Camera Positions from Exhaustive Search 129

5.7 Trace of Agent Communication 132

5.8 Final Camera Positions Using 3 Cameras 132

5.9 Initial and Final Decisions by the Agents 139

5.10 Performance Measures Without Learning 143

5.11 Performance Measures With Learning 144

5.12 Initial and Final Decisions by the Agents 145

5.13 Final Camera Positions Using Translated Scene without Learning . 145

5.14 Performance Measures Without Learning for Translated Scene . . . 149

5.15 Final Camera Positions Using Translated Scene With Learning . . . 149
5.16 Performance Measures With Learning for Translated Scene 151
5.17 Performance Measures With Learning on Translated Scene 152

A.1 DXF Single Face Description 165
A.2 Facet Information from DXF representation 166

List of Figures

2.1 An Object Surrounded by a Geodesic dome 19

3.1 Depth of Focus 38

3.2 Camera Resolution 41

3.3 Geometric Constructs 42

3.4 Camera Field of View 47

3.5 Object Occlusion within Camera Field of View 48

3.6 Facet Orientation within the Camera Field of View 49

3.7 Computation of Facet Orientation 50

4.1 CAD Representation of a Sphere 63

4.2 A Cube Model and its Triangulation 64

4.3 Bounding Polyhedron with Partial Voxelization 67

4.4 Basic Model of the Proposed Agent 69

4.5 Simple Scene using 2 Cameras 74

4.6 Scene from Viewpoint 1 74

4.7 Scene from Viewpoint 2 75

4.8 The Main Components of the Decision Module 83

4.9 Flowchart of the Coordination Algorithm 86

5.1 CAD Model of Target Object 107

5.2 Rendered Model of Target Object 108

5.3 Rendered CAD Model of Scene 109

5.4 Bounding Polyhedron for Camera Range of Motion 110

5.5 Target Visibility from Various Viewpoints 111

5.6 CAD Model of Scene with Initial Camera Positions 112

5.7 Initial Camera Views 113

5.8 Final Camera Views 114

5.9 CAD Model of Scene Showing Final Camera Positions 115

5.10 Utility Values of the Best Results from Exhaustive Search 118

5.11 Camera Views for the Optimal camera Positions 119

5.12 Camera Views for Sub-optimal Positions 120

5.13 Final Positions using 3 Cameras 121

5.14 Final Views for 3 Cameras 123

5.15 Multi-Target CAD model 124

5.16 Multi-Target Rendered CAD model 125

5.17 Total Target Visibility per Candidate Viewpoint 125

5.18 Initial Camera Positions 126

5.19 Initial Camera Views 127

5.20 Final Camera Positions 127

5.21 Final Camera Views 128

5.22 Final Positions for 3 Cameras 133

5.23 Final Views for 3 Cameras 135

5.24 Coverage of Distinct Target Vertices 136

5.25 CAD and Rendered Model of Scene 139

5.26 Target for Inspection 140

5.27 Camera Positions Relative to Scene 141

5.28 Final Views for 3 Cameras 142

5.29 Original Scene Relative to Bounding Polyhedron 146

5.30 Translated Scene Relative to Bounding Polyhedron 147

5.31 Final Views for Translated Scene without Learning 148

5.32 Final Views for Translated Scene With Learning 150

B.1 Generation of Agent Data 168

C.1 Agent Interaction 170

Chapter 1

Introduction

1.1 Sensor Planning

In recent years, there has been much research interest in the field of sensor planning. The focus of this research is, mainly, the quantification and optimization of the relationship between the sensors and the object that is being observed by the sensors [1]. Such a relationship, if known, can indicate the reliability of the task directed sensing function being carried out. Much of the work carried out in sensor planning has dealt with dynamically changing sensor configurations in such a way as to achieve the optimal sensing arrangement for a particular sensing task. The optimality of the arrangement is based on some measure of visibility or reliability and the sensing goals are usually the measurement of geometric and/or physical features of the environment. The overall goal of sensor planing is to automatically generate the proper sensor configurations given any known a priori information about the environment. Such information may be in the form of CAD models or adjacency graphs or any other type of representation where geometric and topological features of the environment may be represented.

Most of the research carried out in the area of sensor planning has centered around vision tasks which are usually allocated to systems containing cameras and laser range finders[1]. Feature inspection is a very popular task for such systems. The aim is to have the system automatically determine the various sensor parameters that would allow all features of interest to be simultaneously visible at the correct focus and magnification. There are presently many computer vision systems that rely on a great deal of human intervention to determine the optimal placement of the cameras for a particular vision task. For example, in a robotic vision system that controls an assembly line, the manufacture of a particular product may require the placement of cameras in an appropriate configuration in order to facilitate an inspection task. However, the appropriate camera parameters and positions are usually obtained by means of lengthy trial and error methods. In addition, such systems are invariably inflexible and subject to error due to unforeseen factors or events such as slight alterations of the environment. Such systems tend to function efficiently for a particular situation, but have to be reformulated for novel situations.

A sensor planning system is therefore a means of alleviating the bottleneck associated with human controlled computer vision based inspection. Such systems have been designed to utilize knowledge about the environment such that novel tasks are carried out without human intervention. The sensor planning system can be used to automatically position and orient the cameras as well as the light sources. In addition, the automated control of the camera optics such as zoom, focus and other parameters decreases the overall complexity of the system for the human operator.

Sensor planning techniques have been applied in the areas of automated visual inspection systems [2, 3], as well as robotics [4]. Well known systems such as General Automatic Sensor Planning (GASP) [3] and the Machine Vision Planner (MVP) [5] utilize geometric models of the environment and models of the sensors to derive the viewing positions based on the specified task. Other systems such as SAUSAGES[1] utilize sensor planning techniques for the guidance of autonomous vehicles and the control of camera movement associated with such vehicles [6].

In each case, the system may either contain a single mobile sensor or multiple sensors capable of independent movement. For example a sensor planning system may control a single camera attached to a robot arm that has many degrees of freedom [7]. Alternatively, the sensor planning system may consist of multiple sensors, each attached to a mobile platform [8]. The traditional approach to the implementation of such sensor planning systems is based on the centralized control of one or more cameras. The control algorithms may utilize a variety of methods including constraint optimization [5], expert systems [9] and candidate viewpoint space search [7]. The centralized execution of these algorithms do however, possess some inherent disadvantages as summarized below.

- Since the entire system depends on a single processing node, a failure of this node can lead to failure of the entire system.

[1]Developed at Carnegie-Mellon University . The Plans for Coordinated Sensor movement are stored and executed by this system

- Increasing the number of cameras requires more complex programming of the centralized control software.
- The processing time required for a given task may be directly proportional to the number of cameras in the system. More efficiency could be realized if the tasks were executed concurrently.
- A centralized control system may be inadequate for implementation in hardware due to size and computational resource constraints.

For sensor planning systems involving multiple cameras, the disadvantages may be addressed by distributing the sensing task and processing requirements amongst the individual cameras. Each camera therefore would become an integral part of an autonomous problem solving module that we refer to as an agent. This approach relies on communication amongst the individual camera modules to achieve the degree of coordination necessary to accomplish the given sensing task.

The general objective of this dissertation is to develop a framework for the coordination of such a distributed autonomous system of agents. In order to preserve the autonomy of the system, the individual agents must be able to reason about their individual plans with respect to the overall task of achieving a particular sensing goal. In the following section, we introduce the concept of agency and the advantages that are characteristic of a distributed control methodology.

1.2 Agents

The definition of an agent is one of much discussion and diversity within the research community. Most of the definitions are domain dependent and hence the term agent is most accurately defined within the domain to which it is applied. One unifying statement that can be made on this matter is that an agent is an entity that can perceive and affect its environment [10]. An agent can possess capabilities that represent some degree of autonomy. Such capabilities may include but are not limited to:

Communication The agent should be able to communicate with other agents or with a human.

Actuators The agent should be able to affect its environment

Intelligence The agent can adapt to changes in its environment or learn about its environment in such a way that its behavior is improved over time.

Knowledge The agent may possess some knowledge of the environment in which it resides. This knowledge may be static or dynamic depending on the capabilities and the tasks assigned to the agent.

It is important to note that although intelligence is not a necessary condition for agency, it contributes greatly to the degree of autonomy exhibited by the agent. In this research we depend on an agent's ability to make rational decisions both from an individual and a collective perspective. Such decisions can be influenced by the level of intelligence demonstrated by the agent. Hence the notion of intelligent agents is an important one.

Agents may be classified as *static* or *mobile*, depending upon whether or not they move around in their environment. Agents may also be classified as *deliberative* or *reactive*. Deliberative agents possess an internal reasoning subsystem which allows them to engage in planning and negotiation in order to achieve coordination with other agents [11]. Reactive agents essentially react to stimuli from their environment without the need for an internal reasoning subsystem.

1.3 Multi-agent Environments

An environment that consists of a group of agents that cooperate to jointly solve problems is known as a *Multi-agent* environment. In such an environment, the aim is to take advantage of the collective problem solving ability of the group since no one agent has the capability of solving a particular problem on its own. Multi-agent environments offer many advantages over single agent environments, among which are the following.

- Problems solved by a group of agents can be significantly more complex than those solved by a single agent.

- The programming complexity of the individual agents is reduced since each agent may have simpler functions and problem solving capabilities.

- A multi-agent environment offers a higher degree of fault tolerance since the entire system does not depend on a single agent.

- Such an environment lends itself to parallel execution and may therefore offer improvements in time critical applications.

In a multi-agent environment, conflicts amongst agents may arise due to interdependencies amongst the agents [12]. As a result, there must exist some method of conflict resolution that is generic enough to resolve most, if not all possible conflicts. In addition, there needs to be an efficient method of coordinating the activities of the individual agents so that the overall goal is eventually achieved. Such coordination requires that the agents communicate amongst themselves, hence the additional necessity of an efficient and robust communication system. These are but a few of the issues that characterize a multi-agent environment that do not necessarily occur in a single agent environment.

Despite the inherent increase in complexity over single agent systems, multi-agent systems have been applied with great success to a wide variety of problem areas. These areas include air traffic control [13], robotic vision systems [2] and flight reservation systems [14]. However, from a traditional perspective, sensor planning and intelligent multi-agent coordination and control have been very distinct research areas. The recent interest has been fueled by demands from the military and industrial sectors for more intelligent active sensing systems. As a result, intelligent sensing systems that rely on inter-agent cooperation have been developed for use in so called hazardous environments [6]. The combination of these two distinct areas of research offers tremendous advantages over traditional sensor planning methodologies. In this research, we utilize the collective capabilities and problem solving skill of multiple agents to find the required sensor positions for a particular sensing task.

1.4 Statement of the Research Problem

The goal of this research is to develop a framework for sensor planning based on the collective computational capabilities of collaborative agents. Such a framework would provide the necessary structures, coordination algorithms and learning algorithms such that the "appropriate" sensor configurations may be generated with "improved efficiency" over time for a particular sensing task. This research utilizes a homogeneous group of intelligent agents to efficiently control the deployment of a group of cameras so as to obtain maximal visual coverage of one or more targets being observed.

The requirement of multiple cameras may be due to other objects in the scene occluding or partially occluding the target object. Multiple cameras may also be necessary when multiple spatially distinct targets are under simultaneous regard or the large size of the target object may require multiple fields of view for maximal coverage. By planning the sensor configurations for maximal target coverage, the resulting views can be used for image processing applications including inspection of one or more features of the target.

In designing such a framework, there are important criteria that must be considered and should be addressed within the framework. These criteria include the following.

Scalability and Re-usability The framework should allow for the addition of new sensors such that they may be incorporated into the existing group of sensors automatically and with minimum effort by the user.

Coordination The system must be coordinated such that the plans of the individual agents contribute in a positive way towards the global utility of the system. Conflicts must therefore be resolved in an efficient and productive manner.

Fault Tolerance In case of sensor failure, the system should automatically reconfigure so that the sensing task can be achieved.

Efficiency The system must be efficient in finding an overall sensing plan for a particular sensing task. Therefore the system should produce a solution for a particular sensing task in "an acceptable period of time" given a sufficient amount of resources.

Learning Ability The system must learn to improve its performance with experience.

Convergence The system must be able to converge either to a particular solution or a state where it informs the user that a solution is not possible given the current resources.

Such a system can contribute significantly to ongoing research in sensor planning in a variety of ways. An agent based sensor planning system can easily reconfigure itself to allow for changes in the environment. The system would be more efficient than simple trial and error in providing robust sensor configurations for a particular sensing task. In addition, the system is scalable since more sensors may be added to the group without the need for extensive intervention by the user.

1.5 Organization and Scope of the Thesis

Although many types of sensors are possible. we have attempted to limit the scope of this thesis by focusing on the use of cameras as the characteristic sensor. Furthermore. the thesis is concerned with planning the viewpoints of the cameras in a modeled environment.

The following chapters expand on the concepts introduced in the above sections. Chapter 2 provides a comprehensive survey of the research currently being carried out in sensor planning. In addition. this chapter also provides a summary of the fundamental theories concerning multi-agent coordination and planning. The chapter characterizes some of the issues that must be considered when utilizing multiple agents. such as communication. learning and knowledge representation.

Chapter 3 presents the theoretical foundations pertinent to this thesis from the the areas of optics solid geometry.

Chapter 4 presents the proposed framework of the multi-agent sensor planning system. In this chapter we describe in detail the components of the developed system with careful attention to the role played by each of the subsystems involved.

Chapter 5 provides an example of the results that are possible from the system presented in this thesis. The examples were chosen to illustrate the variety of data models that can be accommodated by the system. In addition. each model serves to highlight important capabilities of the system. Chapter 6 highlights the main contributions of this work and places the framework design in perspective relative to the previous work carried out in this area. This chapter also provides suggestions for future research.

Chapter 2

Literature Review

2.1 Introduction

In this chapter we explore the current state of the art in both sensor planning and intelligent agent systems. The chapter attempts to present the current research in sensor planning within the context of the various approaches to this problem. We then present the major work being carried out in the area of multi-agent systems. Finally the research being undertaken that attempts to unify multi-agent technology and sensor planning is presented.

2.2 Sensor Planning

The research being carried out in the area of sensor planning has traditionally been focused on three general areas of application, namely scene reconstruction, model-based object recognition and feature detection. These areas essentially differ in the amount of knowledge that is known a priori and also the vision task to be achieved.

In scene reconstruction, the goal is to reconstruct a model of the scene by incrementally sensing the unknown world and amalgamating the successive sensor readings into a partial model. The next best sensor configuration is based on the knowledge gained about the world so far. There are several parameters that determine the effectiveness of the next sensor configuration. For example, a sensor configuration may be chosen based on its superior ability to explore the largest region of unexplored space. In this problem very little knowledge about the world is known a priori. There has been considerable research done in this area that focuses on the criteria that determines the next best sensor configuration and the integration of the partial scenes [15, 16].

Sensor planning research in model-based object recognition has focused mainly on the sensing tasks required to determine the identification of an object and its pose. The approach usually employs a hypothesize and verify methodology whereby hypotheses regarding the identity and pose of the object are generated based on the initial sensor input. These hypotheses are then verified by some predefined metrics and new sensing configurations are proposed based on the most accurate hypotheses. An excellent overview of this approach can be found in the published work of Hutchinson et al [4].

In addressing the problem of feature detection, the goal is to automatically determine the optimal sensor parameters that would offer the most information about one or more features of a known object in a previously determined pose [1]. There is usually a significant amount of a priori information available to the system.

It is this knowledge of the object in question that determines the decisions made by the system. The features being observed must meet certain requirements as set out in the vision task. These requirements usually include (but are not limited to) the need for the observed features to be focused, correctly magnified, and un-occluded by any part of the object being observed or by other objects in the scene. There is a substantial amount of research that has been carried out in this area. The emphasis is on developing algorithms for automatically planning the sensor parameters for various vision tasks. Most famous is the work done by Tarabanis et al [17, 18, 19]. Other related work includes that undertaken by Sakane et al[20], Cook et al [6] and Trucco et al [3].

The research presented in this thesis concerns the third application domain, that of feature detection. Hence we will necessarily limit our literature review to sensor planning research in this area. As previously mentioned, there are many systems that attempt to provide solutions to the general problem of sensor planning as applied to feature detection. From a very high level perspective, the basic difference between these systems lies in the method used for determining the actual sensor parameter values that will achieve the particular feature detection task. Augmenting the categorization imposed by Tarabanis and Allen in their survey of sensor planning methods [1], we present the following four categories of sensor planning methods.

1. The Synthesis Approach

2. The Generate and Test Approach

3. The Expert Systems Approach

4. The Agent Approach

2.2.1 The Synthesis Approach

In the synthesis approach, the sensor parameters, object properties and the sensing tasks are described as analytical relationships. The sensor configurations are subsequently obtained from these relationships by analytical means. Classical implementations of this approach include the *Automatic Sensor and Illumination Planning System* developed by Cowan et al [21, 22] at the robotics laboratory of SRI International and the Machine Vision Planner or *MVP* system developed by Tarabanis et al [5]. In these systems, the goal is to automatically synthesize the desirable camera views of a scene based on geometric models of the environment, models of the vision sensors and, models of the task to be achieved. In both systems, the general approach is to find the locus of viewpoints that satisfy each of the following task constraints.

- *Feature Visibility* The features to be inspected must be not be occluded by each other or by other objects in the scene.
- *Focus* The features must be in focus from any viewpoint chosen from the locus of admissible viewpoints.
- *Field of View* The features must be in the field of view of the camera.
- *Resolution* The features must be spatially resolvable to a given specification from any viewpoint within the locus of admissible viewpoints.

The loci of admissible viewpoints that satisfy each requirement are then intersected to find the locus of globally admissible viewpoints that simultaneously satisfy the task requirements. Although both systems rely on the same general approach, they differ fundamentally in the number and type of parameters that are planned and the methodology used to satisfy all the constraints.

In the SRI system by Cowan et al. the task requirements or constraints are satisfied individually by an iterative search technique. The method determines the locus of viewpoints in 3D space that satisfy the constraint being considered. This set of viewpoints is obtained by iteratively building the region that satisfies the constraint. Once the locus of viewpoints is produced for each constraint, they are intersected to find those viewpoints that satisfy all the constraints simultaneously. Hence a generalized set of viewpoints is synthesized from the individual loci.

The camera optical settings such as focal length f and aperture a are not planned by the system but are chosen a priori. In addition the orientation of the camera is set to the centre of a sphere that circumscribes the region of interest. This reduces the number of planned parameters and assists in the efficient convergence of the system.

In contrast, the MVP system formulates the problem as a constraint satisfaction problem consisting of eight variables. The planned parameters are three positional degrees of freedom $r_0(x, y, z)$ and two orientational degrees of freedom in the form of pan and tilt angles. In addition, the distance between the back of the lens to the plane on which the image is formed (back nodal point to image plane distance) d, the focal length f and the aperture of the lens a are also planned by the system.

As will be illustrated in the following chapter, these parameters affect the depth of focus and field of view of the camera, hence their importance to the achievement of the vision task.

For each task constraint, the admissible region is bounded by a hyper-surface which is described by an eight dimensional vector. The combination or synthesis of these individual regions produces a locus of viewpoints that satisfy all the constraints simultaneously based on the planned parameters. The idea is to then find the optimal parameters within this locus of generalized viewpoints.

As an optimization problem, the analytical relationships that model the vision task constraints are used as the constraints of the optimization process and the objective function is some metric of the distance between a candidate generalized viewpoint and the bound described by the combined hyper-surfaces. Since each task constraint is modeled analytically, the locus of viewpoints that satisfies each constraint is expressed as an inequality function g_i of the parameters being planned as in equation 2.1.

$$g_i(r_0, \bar{v}, d, f, a) \geq 0 \tag{2.1}$$

Where $\bar{v}$ is the vector describing the viewing direction or orientation of the camera and the parameters r_0, d, f, a are as previously defined. Each inequality g_i specifies the relationship between the planned parameters based on the constraint being referred to. In other words, given a particular set of parameters, each inequality specifies how well that set of parameters satisfies a particular constraint. Hence, $i = 1..n$ where n is the number of constraints. The optimization function

F is thus a weighted sum of the inequality functions. This is expressed in equation 2.2

$$F = max(\alpha_i g_i) \quad For \quad i = 1..n \quad subject \quad to \quad g_i \geq 0. \tag{2.2}$$

In both the SRI and MVP systems, the emphasis is on the utilization of a single camera, an illuminating source and a centralized planning mechanism. The systems both utilize CAD models of the scene where objects within the scene are modeled as convex or concave polyhedra. The MVP system offers some advantages over the SRI system in terms of the robustness of the solution due to the fact that all the camera parameters are planned explicitly. However the SRI system offers some advantages in terms of efficiency since fewer camera parameters are planned by the system.

2.2.2 The Generate and Test Approach

In the *generate and test approach*, sensor parameter values are generated and then evaluated based on some predetermined criteria. The space of possible sensor parameters is usually discretized and heuristics are employed to limit the search space. Usually, the object is in a known pose and surrounded by a tessellated sphere which provides the discretized set of possible viewing positions in 3D space.

Systems that employ the generate and test approaches include the HEAVEN system by Sakane et al [7, 8], the Illuminator Control Expert (ICE) system by Yi et al [23], the General Automatic Sensor Planning (GASP) system by Trucco et al [3] and the viewpoint planning system developed by Roberts et al [24].

In the HEAVEN system, the object under observation is surrounded by a sphere with its center at the geometric center of the target object. The sphere also circumscribes an icosahedron whose triangular facets are projected onto the surface of the sphere. The result of this projection is the tessellation of the surface of the sphere by equilateral triangles. Each triangle may be subsequently subdivided to produce 4 triangles thus creating a finer tessellation. The resulting tessellated sphere is referred to as a geodesic dome [25] and is illustrated in figure 2.1. Using the center of each facet as a viewing point, a ray is passed from this center to the surface of the target object. All intersections of the ray with the surface of the target object can be computed. If there is an occluding object in the path of the ray, then the ray would intersect such a surface prior to intersecting the target surface. Hence any occluding surfaces can be identified.

The HEAVEN system uses a distance measure to rank each facet within an occlusion free region for a particular sensing task. An occlusion free region is essentially a region on the surface of the sphere where the rays projected to the target object are not intersected by any other object in the scene. Facets that are close to the border of the occlusion free regions are ranked lower than facets that are near the center of an occlusion free region. The distance measure utilized is the negated inner product of the ray from the center of the facet under consideration to the center of the nearest occluded facet. Once these facets have been ranked, they are then sorted by decreasing order of their rank. The sensor is a single camera mounted on a robot manipulator (the so called eye in hand configuration). This is placed at the intersection of the highest ranked facets, the facets occupied by the workspace of the manipulator and any user specified facets that provide additional

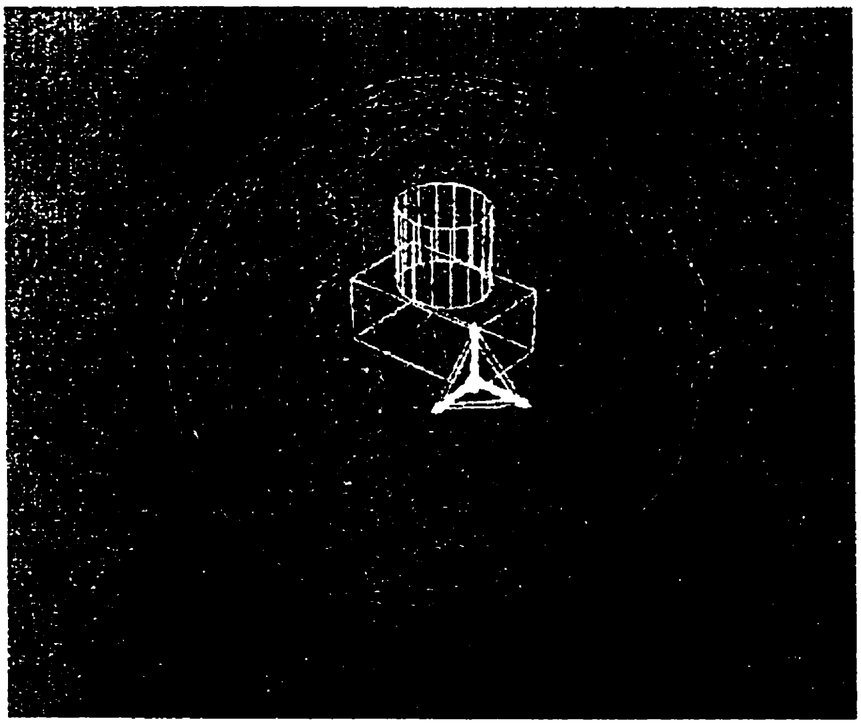


Figure 2.1: An Object Surrounded by a Geodesic dome

information to the problem.

The ICE system utilizes the same method for generating possible viewpoints on the surface of a tessellated sphere. However, this system also plans the position of the illumination source in addition to the camera position. The camera and illumination source positions are planned separately and independently of each other. Hence the criteria utilized for obtaining the best positions are also different. In order to plan the camera position, the system ranks the candidate viewpoints based on edge visibility. This refers to the length of an edge on the target object that is not occluded compared to the total length of an edge. The camera is positioned such that the total number of complete edge segments visible is maximized.

The illumination planning portion of the ICE system utilizes an independent optimization process. The system optimizes the so called edge contrast parameter. This is a measure of the difference in reflected light intensity between neighbouring regions in an image of the target taken from a candidate viewpoint. By utilizing faces of the target that meet at the edge under consideration, the contrast for an edge may be evaluated using a finite number of points along the edge. The resultant contrast graph represents the variation in contrast along the edge and is used to determine the contrast distribution. This function is then used to assess the optimization criteria specified as:

- The ratio of the portion of an edge in which a given contrast threshold is exceeded as compared to the total length of the edge.
- . The amount by which the threshold is exceeded over that portion of the edge that exceeds the threshold.

We note here that the Illuminator planning takes place once the camera planning has been completed. Hence the illumination planning does not alter the camera position. Due to the inherent interdependencies of these two subproblems, it is possible for the solution obtained to be suboptimal.

Other systems that follow the generate and test approach include the General Automatic Sensor Planning *GASP* system developed by Trucco et al [3] and the viewpoint planning system by Roberts et al [24]. The GASP system focuses on the optimal planning of viewpoints for objects commonly found in manufacturing and can accommodate both range and intensity image sensors. The optimality of a given sensing configuration is based on a weighted combination of feature visibility and measurement reliability criteria.

The information required to compute the visibility and obtain measurements on a given feature is stored in a CAD model. The CAD model encodes shape information and provides reference measures. The system relies on the manipulation of Feature Inspection Representations (FIRs) which, at the basic level, contain the best viewpoint from which a single intensity or range camera can obtain a desired measurement on a given feature. More complex inspection tasks can be carried out by combining the FIRs into inspection scripts. For example, the system can inspect multiple features using a single sensor by finding a region in 3D space from which multiple features are co-visible or by finding the shortest path in 3D space through which a single sensor can view each feature in succession from its optimal viewing position as specified by the FIR. The system utilizes a composite traveling salesman algorithm [26] to find the required shortest path. In addition. The system is also capable of utilizing a stereo pair of sensors to inspect single or multiple features as previously described.

In keeping with the generate and test methodology, the GASP system utilizes a geodesic dome centred at the centroid of the object to generate the candidate viewpoints. The visibility of the various features in the CAD model is computed off-line and stored in the feature representation format. The optimality of a given viewpoint is defined as shown in equation 2.3.

$$o = K_v v + k_r r \tag{2.3}$$

The coefficients k_v and k_r indicate the relevant importance of the visibility v of the feature and the reliability r of the measurements obtained from a given viewpoint. We note here that $k_v, k_r \in [0, 1]$ and $k_v + k_r = 1$. The online efficiency of the system depends on the complexity of the vision tasks to be carried out since the FIRs are computed off-line.

The vision planning system designed by Roberts et al [24] was motivated by the fact that in many cases, object inspection and object recognition cannot be performed adequately from a single image. Hence multiple views of the object are needed to adequately cover the surface of the object. The system therefore selects a minimized number of views that allow each object face to be adequately viewed according to specified constraints.

The system obtains a solution in two phases. In the first phase, the system computes a search space for the viewpoint planning,. This search space is represented by a graph where the nodes correspond to the faces of the target object and the arcs connect nodes (faces) that simultaneously satisfy all constraints. For example, if the constraints are that the faces must be visible and in focus, then there is an arc connecting any two faces that are both visible and in focus from a given viewpoint. This information is generated from a CAD model of the object and visibility information is obtained by considering a finite set of possible candidate viewpoints.

The system then computes the largest set of faces that are visible from a candidate viewpoint. This set of faces is removed from the graph and the process is repeated for the remaining nodes. The resulting subsets of the candidate viewpoint list form an approximation to the set of maximally connected subgraphs or cliques.

The second phase of the system involves the actual viewpoint acquisition. The system accommodates three methods for viewpoint acquisition as listed below.

1. View acquisition using an eye in hand camera.
2. View acquisition using a fixed camera and turntable.
3. View acquisition using a stereo vision system.

Using an eye-in-hand configuration requires that the camera is mounted on a robot arm that has enough range of motion to position the camera at any viewpoint on the surface of the surrounding view sphere. The output from the previous stage provides a list of object surfaces that are visible from a given viewpoint. The set of viewpoints from which all faces of the object are visible is obtained by intersecting the individual visibility regions of the faces of the object. The system then finds the best viewpoint within this set by choosing the viewpoint that has a minimum angle to all the face normals that are visible. This point is used as the viewing position at which the eye-in-hand system is positioned.

The fixed camera and turntable setup consists of a camera in a fixed location and oriented towards the centre of the view sphere. The object is placed on a turntable and the only motion is the rotation of the object in fixed angular increments. The camera therefore forms a horizontal circle on the surface of the view sphere due to the rotation of the sphere with the object. The candidate viewpoint region is obtained as described above but with the added constraint that the viewpoints considered must also lie on the circumference of the circle traced by the camera as the sphere moves relative to the camera.

For the stereo camera system setup, the candidate viewpoints are generated in the same manner as the single camera case. However, the viewpoints considered are the set of non-coincident points that simultaneously provide an un-obstructed view of a particular feature. Hence the system must remove all the candidate viewpoints corresponding to the position of one camera that do not guarantee that the features are also visible in the other camera.

2.2.3 The Expert System Approach

The expert system approach relies on the encoding of an expert's knowledge as to the best lighting and viewing configurations for particular sensing tasks. The user inputs information on the object or feature to be observed and the expert system outputs the appropriate lighting and/or viewing recommendations. Examples of such systems include the *LIGHTING ADVISOR* created by B. G. Batchelor [9].

and a similar system developed by A. Novini [27].

The information required by these systems include the reflectance characteristics of the object and the type of feature that is to be emphasized. The program then displays a line drawing of the appropriate lighting condition. The system by Novini also gives advice on the image processing operations that should be used to extract certain types of features. It is important to note that these systems only provide qualitative information on the type of lighting that would be most appropriate for the particular task. For example, the systems would determine whether the object should be illuminated from the front or rear to provide the best conditions for feature inspection. Extensions to these systems also suggest the particular viewing method to be used. However, they do not provide any suggestions as to the exact spatial configuration of the cameras or illuminators for inspection. The idea is to address the problem from a qualitative perspective derived from a catalogue of possibilities.

2.2.4 Agent Based Systems

The previous sections have presented systems that utilize either a single camera or a set or cameras (as in the case of stereo vision configurations) that are explicitly controlled by a central planning algorithm. In this section we present sensor planning systems that rely on distributed control for the concurrent planning of several sensors. Each sensor is locally controlled by a problem solving entity or agent. The definition of the term agent is very much influenced by the problem domain for which the agent is designed. However, from the perspective of the following systems, an agent is, at the very basic level, a computing entity that has the resources to solve or attempt to solve a given computational problem. The degree of agency attributed to a computing entity really depends on the observable levels of intelligence, pro-activeness, communication abilities and autonomy demonstrated by the entity in various problem domains [28].

The application of agents to sensor planning is based on the ability of agents to autonomously coordinate their actions in an effort to achieve the optimal or at least functionally acceptable sensor configuration for the given sensing task. Each sensor is controlled by a single agent and the agents can communicate with each other by way of messages through some underlying communication medium. There are several general coordination schemes that have been developed for coordinating groups of agents. These and other fundamental agent theories will be presented more rigorously in the following chapter. However, we present here an overview of the systems that utilize this approach.

Durfee et al [29] have developed a sensor surveillance system based on a network of semi-autonomous problem solving agents. The system is known as the *Distributed Vehicle Monitoring Testbed* or DVMT. Each agent controls an acoustic sensor and is capable of communicating with the other agents in the network. The sensing task is to identify, locate and track patterns of vehicles moving through a two dimensional space based on their acoustic signatures. The agents cooperate by generating and exchanging tentative partial solutions based on the local acoustical data obtained from their sensors. By iteratively exchanging and refining these partial solutions, the network eventually converges to an overall solution.

The partial solutions generated by each agent are hypotheses that describe the belief of the node as to the time stamped location (where the vehicle was at certain times), the type of vehicle and the confidence in the hypothesis. A complete solution details the position and identification of the vehicle at a given time or over a period of time. Each sensor only covers a small portion of the problem space and may have overlapping fields of view with other sensors. Hence the hypotheses are based on local information only. The agents communicate with each other and refine their hypotheses through a blackboard system based on the HEARSAY II architecture [30, 31].

The agents achieve the coordination necessary for their task through the use of organizational structuring. An organizational structure specifies a set of long term responsibilities and interaction protocols for each agent. The establishment of this structure is accomplished during the creation of the network. In the case of DVMT, the organizational structure defines an area of interest for each agent within the sensor space. Although the decision to transmit or receive information concerning a local hypothesis is made by the agent, the organizational structure imposes some guidelines as to when to transmit or receive a hypothesis. This is based on the importance of the sensed data within the area of interest of the agent. For example, a hypothesis created and transmitted by an agent would carry a higher confidence rating if the vehicle is believed to be in the centre of the area of interest of the agent as opposed to being close to, or outside of the boundary of the area of interest. The disparity in confidence exists since the sensor may be tracking ghost data when the vehicle is close to the boundary of its area of interest instead of the true vehicle data. By coordinating their influence on the iterative construction of the final solution, the agents can collectively solve the vehicle monitoring task without the need for complete knowledge of the environment.

Okoshi et al [32] have demonstrated a multi-agent model-based system for feature inspection. The system consists of seven agent processes running on three workstations. Three PUMA 560 manipulators provide the dexterity for a camera, and two light sources. There are also two mobile robot vehicles with mounted cameras. The remaining agents are image processing agents running on workstations and they provide the image processing capabilities. The goal is to remove a valve handle and nut from a valve assembly and inspect the valve sleeve for water leakage. The system uses robot vision to to determine the rotation angle of the valve handle, verify that the handle is grasped by the manipulator and finally, inspection of the valve for water leakage.

Each agent can send messages to the other agents. The system is coordinated by means of a contract net protocol [33]. This protocol allows an agent to broadcast requests for assistance in performing a particular task. Any agents that are capable of providing assistance to the soliciting agent offers bids. The bids are received and analyzed and a contract is awarded to the agent with the most qualified bid based on some given criteria.

In this system, the agent controlling the manipulator and camera broadcasts a message requesting assistance of a lighting agent to provide the optimum lighting conditions for image processing in order to determine the rotation angle of the valve. The contract is awarded to a light source agent. After the handle has been grasped by the manipulator. the agent then broadcasts a request for a camera agent to verify that the handle has been grasped. The camera agent with the winning bid must then position its camera such that an un-obstructed view of the grasped handle is obtained. Once the handle has been removed by the manipulator, another contract is awarded to a camera agent to position its camera so that the valve sleeve can be inspected. The image obtained is then passed on to an image processing agent for analysis. It is important to note that the camera parameters (position and orientation) are computed off-line prior to the activation of the system of agents. The contracts were awarded to the camera agents based on their proximity to the planned viewing position at the time when the bids were requested.

Another variation on the agent approach to sensor planning developed by Cook et al [6, 34] relies on decision theory to coordinate the sensor planning amongst multiple autonomous vehicles executing a military mission as a part of DARPA's[1] unmanned ground vehicle program. The idea is to allow a group of autonomous vehicles equipped with cameras to select optimal viewing locations and camera pan and tilt angles in order to gain the maximum information during a surveillance task.

The system relies on three levels of coordination to accomplish both surveillance and target tracking tasks. On the first level, the areas of interest to the group is decided upon by the (human) mission leader. An observation point refinement algorithm is used to select optimal observation points from which vehicles can observe a specified area of interest. The algorithm utilizes polygonal descriptions

[1] Defense Advanced Research projects Agency

of the areas of interests to make its selection. The decision as to which observation points to choose is based on a utility measure that characterizes each candidate observation point in terms the probability of detection of the group of vehicles and the estimated amount of information that can be obtained from the observation point.

The area surveyed by any of the ground vehicles is divided into segments or fields of view. The time spent observing a particular field of view is dependent on a priori information such as the probability of finding a target in a particular field of view. Thus the fields of view are weighted based on their importance and these weights are updated after every mission. The decision by an agent as to the order in which its field of views are observed is made at the local level without consultation with the other agents.

The third level of control utilizes distributed decision making in order to perfrom target confirmation, security hand-off and health checks. Target confirmation requires the input form all agents whose camera is in line of sight with the target. Hence, the agent that detects a target can request confirmation of the target from the other agents. All target confirmation information is communicated to the requesting agent. If an agent is tracking a moving target it may request that its security surveillance responsibilities be temporarily handed over to another agent not involved in the tracking process. The group may also need to reconfigure itself if a particular agent either communicates a failure to the group or does not respond to periodic health checks by the mission leader. This reconfiguration may change the formation of the group and/or reassign particular areas of interest that were the responsibility of the failed agent.

2.3 Discussion of the Different Approaches to Sensor Planning

The preceding sections have presented an overview of the various methods developed for the planning of one or more sensors. Each method provides a distinct contribution to the available methodologies. In order to adequately discuss the advantages and disadvantages of these methods, we must present some general criteria by which we can measure the suitability of the methods to a given generic sensor planning problem. The criteria is based on the issues that affect any sensor planning system. Such issues include but are not limited to the following:

Scalability Can more sensors be easily added to the system for more complex sensing tasks?

Reliability Can the system provide the user with some confidence measure of its output?

Heterogeneity Can the system accommodate different types of sensors?

Adaptability Can the system adapt from one task to another so that it can learn from experience?

Conflict Resolution Is the system capable of resolving potential conflicts, for example severely overlapping fields of view or contention for a single viewpoint.

Multi-Feature Inspection Can the system inspect multiple features concurrently?

Efficiency How efficient is the system in providing a solution to a particular sensing task?

Fault Tolerance Is the system capable of recovering from failure of one or more of its components?

In general, the single sensor and stereo vision systems such as those described in the synthesis and generate and test approaches are robust approaches but are not necessarily capable of meeting the demands of a wide variety of sensing tasks. In situations where the sensing tasks require varying amounts of sensors, the inherent difficulties in scalability in the synthesis and generate and test approaches become apparent. There are many situations where a single sensor would be inadequate for the task, for example, if the feature being inspected is too large to fit in the field of view of a single camera or is severely occluded by other objects in the scene. Another situation that requires the use of multiple sensors occurs if there are multiple spatially separated features to be inspected concurrently. Scalability is therefore an important issue. However, the current systems that utilize the synthesis and generate and test approach are not easily scalable. More explicitly, these systems do not account for the interdependencies that result amongst sensors in a multi-sensor system. Such interdependencies include contention for candidate viewpoints (resource allocation) and information redundancy as a result of overlapping fields of view.

The GASP system and the Vision system developed by Roberts et al [24] approach the problem of multi-feature inspection by moving a camera (or two cameras in the case of a stereo vision setup) sequentially through a list of viewing positions, resulting in the camera coverage of the entire set of features to be viewed. However, since this is a sequential process, the efficiency of the system is significantly less than one that allows concurrent viewing.

Both the generate and test and the synthesis methods provide some measure of viewpoint optimality and hence the reliability of a given viewpoint is known. The systems can also accommodate various types of sensors although not concurrently, due to the fact that they are essentially single sensor systems. There is no provision however for the systems to learn from experience. Hence the efficiency of the systems essentially remain constant regardless of the number of problem cases presented. The other important issue affecting these systems is that of fault tolerance. Since there are only at most two sensors, any failure of a sensor or the centralized planning algorithm would result in the failure of the system as a whole. In situations where the fault tolerance is an important issue, a more decentralized approach to planning would be necessary.

The expert system approach does offer some advantages in terms of adaptability and heterogeneity since it depends on a rule base. For example, the rules could be updated to provide better suggestions based on experience. In addition, the rules could be adapted for various types of lighting and sensor configurations. However, since the system is based on a qualitative approach to solving the sensor planning problem, there is no apparent reliability measure. Also such systems are not easily scalable nor do they possess the degree of autonomy present in the other approaches since they rely heavily on the availability of user knowledge encoded as rules.

The agent approach to sensor planning presents some significant advantages over the other approaches in addressing the aforementioned issues. From a general perspective, the most obvious advantages of this approach are based on the scalability, efficiency and fault tolerance issues. By allowing each sensor to be independently controlled by an autonomous agent, the planning algorithms are by default decentralized. This improves the fault tolerance of the system. In addition, more sensors can be added or sensors taken away without the need for extensive reprogramming of the system. The concurrent execution of the sensor planning algorithms provides an improvement in the efficiency of the system over the sequential generate and test systems especially in the case of multi-feature inspection.

The multi-sensor system developed by Cook et al [6] for military surveillance illustrates the ability of a multi-agent sensing system to adapt its behavior based on experience. However, there is considerable user intervention in the decision making process of the agents. For example, the areas that are to be surveyed and the geometric formation of the ground vehicles are just some of the aspects of the problem decided upon by human operators.

The vehicle monitoring test bed developed by Durfee et al [29] illustrates the ability of the agent based approach to overcome the interdependencies amongst multiple sensors by the use of organizational structures. However, the use of organizational structures not only adds a notion of centrality to the system but indeed, such structures decrease the degree of autonomy of the agents. This is because the role of each agent is dictated by its designer a priori.

The agent based approaches considered so far do not fully exploit the possible rational decision making capabilities of the agents. This capability can decrease the amount of user intervention necessary to solve a given sensor planning problem. By incorporating agents that are more autonomous, we can perhaps increase the efficiency of the problem solving process while still obtaining acceptable solutions to the problem. The remainder of this thesis explores this possibility, by providing a framework for agent controlled multi-sensor planning that relies more on the rationality and communication abilities of the agents to coordinate their efforts. In addition we explore the possibility of both self learning and rote learning to improve on the efficiency with which a given problem is solved.

Chapter 3

Background

3.1 Introduction

In this chapter we explore the mathematical foundations required for the computation of optical constraints used in the planning of camera viewpoints. These constraints include visibility, focus, resolution and depth of field computations. The chapter also discusses some fundamental theories on multi-agent cooperation and coordination and presents a general framework for distributed constraint satisfaction algorithms that form the basis of the work presented in the rest of the thesis.

3.2 Viewpoint Parameters

In general, vision tasks require that the quality of the image obtained is sufficient for the task at hand. This is usually achieved through some image enhancement process by which the features required are enhanced and ultimately extracted. The quality of the image obtained depends not only on the optical properties of the imaging system but also on the viewpoint from which the image is obtained. Since image acquisition is a computationally less expensive process than image enhancement, it would be more advantageous to devote some computational effort to the determination of the appropriate viewpoint parameters. This could result in less effort required for the image enhancement process.

The set of viewpoint parameters typically contains the position and orientation, in terms of pan and tilt angles, of a camera for a given vision task. However, the set can also contain the optical parameters associated with the chosen viewpoint and camera setup. Such optical parameters include the focus and aperture setting as well as the depth of field, the field of view and the feature resolution constraints of the camera system. In this section we present the method of computation for the various optical parameters.

3.2.1 Depth of Field

The focal length of a lens determines the point at which the image of an object is in perfect focus. However, a camera is also capable of acquiring clear pictures of objects at varying distances, providing that these objects are within the depth of field of the camera lens system. This range of distances is a result of the finite area of the photo-receptors of the image plane. Each photo-receptor will accept a point of light of area less than or equal to that of the photo-receptor. If a point object is located at a distance such that the size of the resulting point image is less than

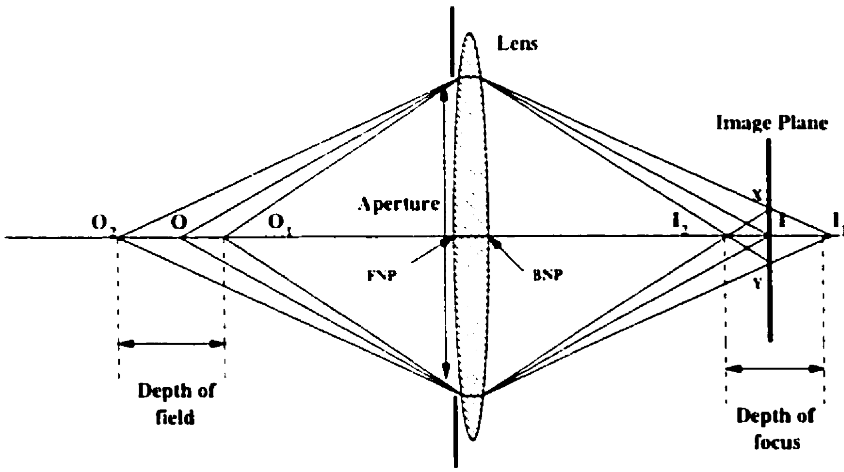


Figure 3.1: Depth of Focus

or equal in area to that of a single photo-receptor, then the object will be in focus [35].

Referring to figure 3.1, consider a point O in front of a camera lens A which produces an image I on the image plane of the camera. The front and back nodal points of the lens are shown as FNP and BNP respectively in the figure. These are the points through which the principal axis of the lens passes. The image plane consists of an array of sensor elements or pixels arranged in N rows and M columns. The points XY represent the diameter of a circle around I within which all point images are less than or equal to the size of the individual photo-receptors and are therefore in focus. This is called the circle of least confusion [35, 22] or blur circle. An observer will see all points within the circle as reasonably sharp points. Point objects whose point images fall outside the circle will be blurred. Now rays from the lens aperture meet at the points I_1 beyond I and also at I_2 in front of I. The point images I_1 and I_2 correspond to the point objects O_1 and O_2 on either side of O as shown. Hence the images of O_1 and O_2 are in acceptable focus on the image plane since they are on the edge of the circle of least confusion. Hence, the distance O_1O_2 is referred to as the depth of field and the distance I_1I_2 is known as the depth of focus.

If a point object is placed at a distance D from the lens centre and the focal length of the lens is f, then the distance of the resulting image will lie at a distance d from the lens centre where d is related to the object distance and the focal length the Gaussian Lens formula of equation 3.1.

$$\frac{1}{d} + \frac{1}{D} = \frac{1}{f} \tag{3.1}$$

The near and far limits of the depth of field D_1 and D_2 corresponding to the positions of points O_1 and O_2 in figure 3.1 respectively, can be computed as shown in equations 3.2 and 3.3.

$$D_1 = \frac{Daf}{af - c_b(D - f)} \tag{3.2}$$

$$D_2 = \frac{Daf}{af + c_b(D - f)} \tag{3.3}$$

Where a is the size of the aperture and c_b is the diameter of the blur circle. f and D are as previously defined. From equations 3.2 and 3.3 we note that if the blur circle is of constant size and the aperture is made smaller, the depth of field increases. Similarly if the aperture is made larger, the depth of field decreases.

Hence the limits of the depth of field can be adjusted by changing the aperture of the lens system.

3.2.2 Resolution

For a given vision task, it is important to know the approximate size of the smallest feature in the scene that can be discerned by the vision system. This is usually expressed in terms of pixel resolution. That is to say, a given feature on an object should appear as a minimum number of picture elements on a sensor. Given a choice of possible viewpoints, a reconfigurable vision system can eliminate those viewpoints that do not allow this constraint to be met.

The method used for the computation of the resolution of an object on the image plane is based on the procedure developed by Tarabanis et al [35]. This method is based on the lower bound of the number of pixels occupied by the edges of the target object. Cowan [22] has also presented a method for computing the resolution of an object. This method is based on the lower bound of the angle subtended by an edge from a point on a polygonal surface. The method used in this thesis is that developed by Tarabanis et al.

Figure 3.2 shows a line segment. AB of length l as imaged by a camera with perspective centre located at O and whose image plane is at a distance d from the perspective centre along a viewing axis OZ'. A perspective centre is a point through which all rays are assumed to pass through. However. in reality. this is not usually the case unless the front and back nodal points of the lens coincide. For

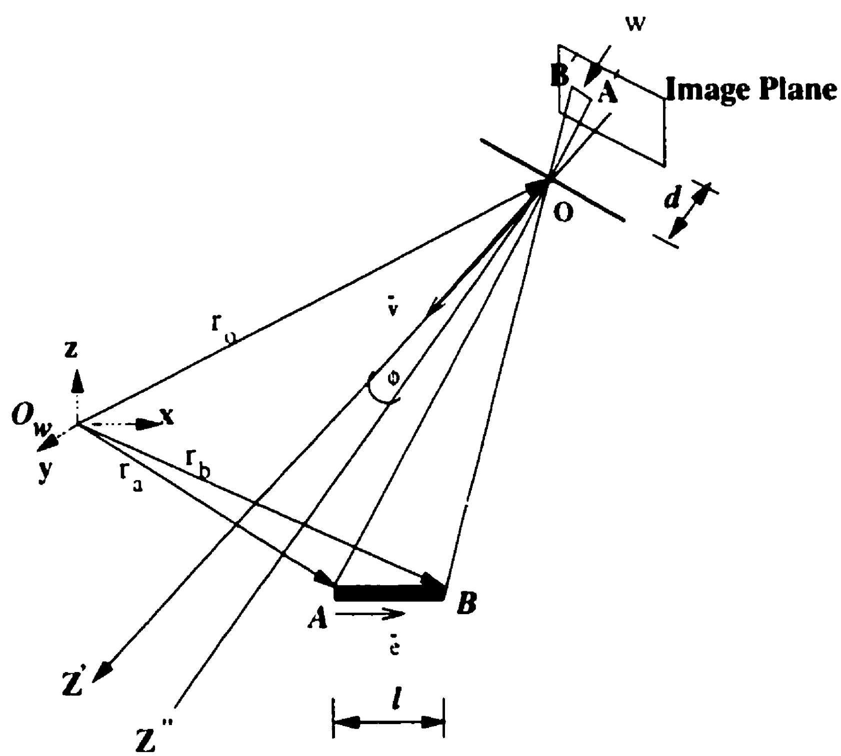


Figure 3.2: Camera Resolution

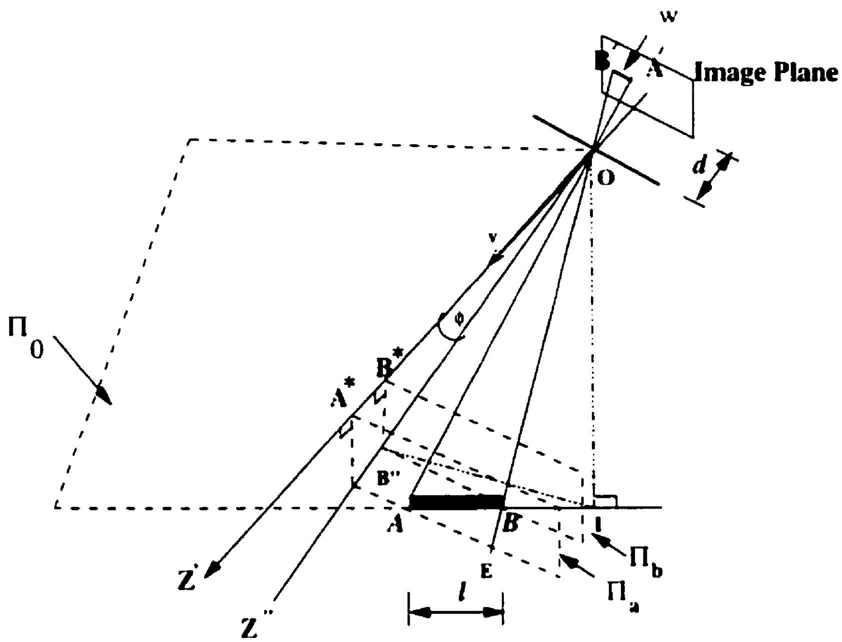


Figure 3.3: Geometric Constructs

the sake of simplicity, the diagram illustrates only one lens centre but the ensuing analysis assumes that the front and back nodal points are distinct.

The vector $\bar{v}$ is the unit vector along the viewing axis of the camera and $\bar{e}$ is the unit vector along the line segment AB. $A'B'$ is the image of the line segment AB formed on the image plane. The length of the image is specified by w. Hence, the objective here is to derive the relationship between the length l of the line segment AB and the length w of the image segment $A'B'$. Using this relationship, we can determine those viewpoints from which $A'B'$ will occupy a minimum number of pixels w on the image plane.

The geometric constructs used to derive such a relationship are shown in figure

3.3. A^{*} is the point of intersection between optical axis OZ' and the plane Π_a that passes through the point A and is perpendicular to the optical axis. The plane Π_b passes through point B and is perpendicular to the optical axis. Pi_b intersects the optical axis at the point B^{*}. Point E is the point of intersection between OB and the plane Π_a. The planes Π_a and Π_b are essentially parallel projections of the image plane such that the projections intersect point A and point B respectively. Hence the triangle formed by OAE is similar to that formed by the points $OB'A'$. If we were to align both triangles with the optical axis, then from similarity we obtain the equation 3.4.

$$\frac{e}{d} = \frac{AE}{OA^{*}} \tag{3.4}$$

Let I be the point of intersection of a line drawn from O to the line containing AB such that the angle OI and AB is a right angle. Also, let Π_0 be the plane of O, A and B. We can then project the optical axis through the angle ϕ to form the projection line OZ'' on the plane Pi_0. The point B'' is therefore the projection of point B onto the line OZ''. It can be shown that the triangle formed by the points OIB'' is similar to the triangle formed by the points ABE. Hence, we can derive the following equations based on similarity.

$$\frac{(AE)}{(OI)} = \frac{(AB)}{(OB'')} \tag{3.5}$$

Since $AB = l$, equation 3.5 can be written as:

$$\frac{(AE)}{l} = \frac{(OI)}{(OB'')} \tag{3.6}$$

Also, from figure 3.3 we see that:

$$(OB'') = \frac{(OB^{*})}{\cos\phi} \tag{3.7}$$

Using the results of equations 3.6 and 3.7 to substitute for (OB'') and (AE) in equation 3.6 we obtain:

$$\frac{w}{dl} = \frac{(OI)\cos\phi}{(OA^{*})(OB^{*})} \tag{3.8}$$

We can substitute for (OI), $\cos\phi$, (OA^{*}) and (OB^{*}) as follows:

$$(OA^{*}) = (\vec{r_a} - \vec{r_0}) \cdot \vec{v} \tag{3.9}$$

$$(OB^{*}) = (\vec{r_b} - \vec{r_0}) \cdot \vec{v} \tag{3.10}$$

$$\cos\phi = \frac{\{\|\vec{e} \times (\vec{r_0} - \vec{r_a})\|^2 - [(\vec{e} \times (\vec{r_0} - \vec{r_a})) \cdot \vec{v}]^2\}^{1/2}}{\|\vec{e} \times (\vec{r_0} - \vec{r_a}\|} \tag{3.11}$$

$$(OI) = \{\|\vec{r_0} - \vec{r_a})\|^2 - [(\vec{r_0} - \vec{r_a})) \cdot \vec{e}]^2\}^{1/2} \tag{3.12}$$

From equations 3.9 - 3.12 we can express the resolution constraint as an inequality in vector form.

$$\frac{\|\vec{v} \times [\vec{e} \times (\vec{r_0} - \vec{r_a})]\|}{[(\vec{r_a} - \vec{r_0})) \cdot \vec{v}][(\vec{r_b} - \vec{r_0})) \cdot \vec{v}} \geq \frac{w}{dl} \tag{3.13}$$

3.2.3 Field of View and Visibility

In order to plan camera viewpoints where the target object can be properly positioned within the image produced by the camera, we must be able to ensure that the target object is within the camera's field of view. In addition, even though an object may be within the field of view of the camera, it may be occluded by other objects within the scene. In this section, we address the computation of both the field of view of the camera and the visibility of an object that is within the field of view. We take the approach that for an object to be visible it must be within the field of view of the camera and un-occluded by any other object in the scene. In addition, an object may be partially visible from the point of view of the camera in two situations. The first situation is that the object lies partially within the field of view of the camera and the second situation is that the object lies totally within the field of view of the camera, but it is partially occluded by some other object within the scene.

Figure 3.4 illustrates the field of view cone formed by a typical camera. The back and front nodal points of the lens are shown as BNP and FNP respectively. The angle α is the angular separation of the boundaries of the field of view. This depends on I_{min}, the minimum dimension of the image plane (width or height) and d, the distance of the image plane from the back nodal point of the lens. The angle α is computed as shown in equation 3.14.

$$\alpha = 2tan^{-1}(I_{min}/2d) \tag{3.14}$$

The vector $\vec{Z}$ is the viewing axis of the camera, based on the camera coordinate system. For simplicity we have assumed that the field of view of a camera is a right circular cone. However, in reality the field of view is rarely symmetrical. It is usually a flattened rectangular cone. We use the minimum angular dimension in order to ensure that only the space actually visible to the camera is considered to be within the field of view of the camera using the simplified right circular cone. From the figure we note that object A is outside the field of view of the camera and object B is inside the field of view and un-occluded. Object B is therefore visible. However, object C is not visible since, although it is within the field of view of the camera, it is occluded by object B. Also, object D is only partially inside the field of view cone and hence it is only partially visible.

In order to facilitate the recognition of partial visibility, the visibility of an object is determined by the amount of its surface that is visible to the camera from a given vantage point. The surface of each object is tessellated by triangles and all vertices and edges form a vertex list and an edge list respectively. Hence, we can determine the number of vertices that are visible on an object as compared to the total number of vertices of the object. Figure 3.5 shows an object A, that has been tessellated into triangles. From the figure we can see that object A is totally within the field of view of the camera. However, it is only partially visible since vertex a is occluded by object B.

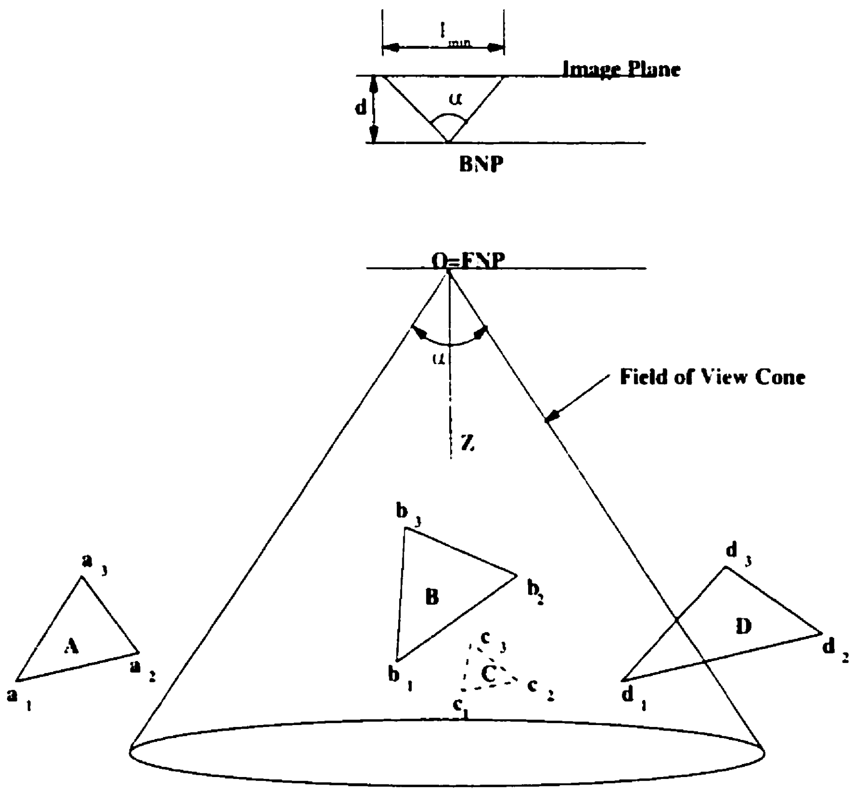


Figure 3.4: Camera Field of View

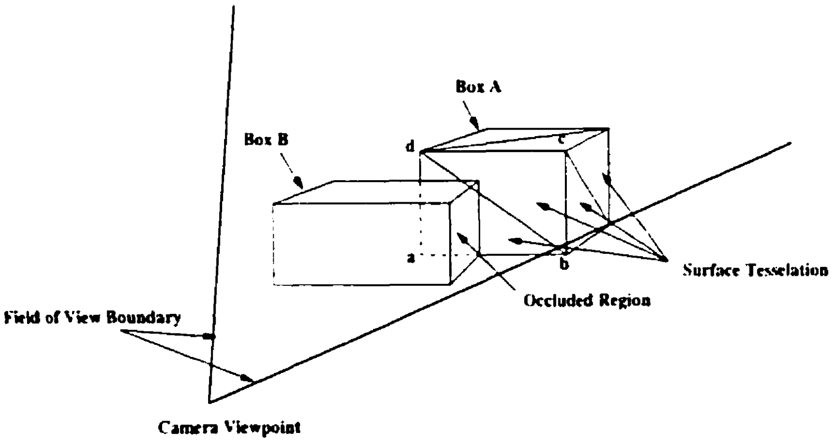


Figure 3.5: Object Occlusion within Camera Field of View

If an object is occluded (either totally or partially) by another object in the scene, then some vertices of the facets of the occluded object will not be visible from the camera viewpoint. Hence, any rays projected from such vertices to the camera viewpoint will intersect one or more facets of the occluding object. By testing for this intersection with other objects in the scene, we can determine exactly those vertices that are occluded on the target object.

Another aspect of the visibility problem is that of surface orientation. Given any triangular facet of an object, we need to ensure that although the vertices of the object are visible, the surface of the object is also visible. Consider the object facets shown in figure 3.6. If we project rays from the vertices of facet A to the camera viewpoint, the rays are within the boundaries of the field of view. However, due to the orientation of the facet, the surface of the facet is co-linear with the rays and hence not visible form the camera viewpoint. Facet B is oriented such that the

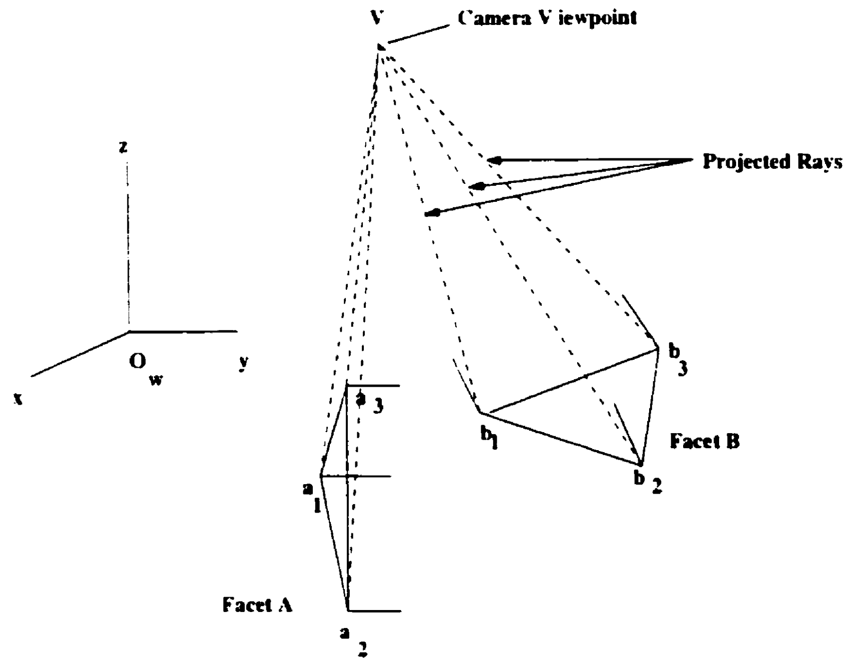


Figure 3.6: Facet Orientation within the Camera Field of View

surface is more visible from the same camera viewpoint. In order to determine the orientation of the facet surface, we utilize the angle σ between the normal to the plane of the facet and the ray projections from the vertices of the facet.

If the average angular separation is close to zero, then the facet is oriented such that the the surface is visible. However, if the average angular separation is closer to 90 degrees, then the facet is considered to be co-linear and hence the surface is not visible. To facilitate this computation, we chose the tessellation of the surface of the objects such that the normal to the facets are always pointed in the direction away from the surface of the object. If the average angular separation of the projected rays and the normal is greater than 90 degrees or negative, then the surface of the

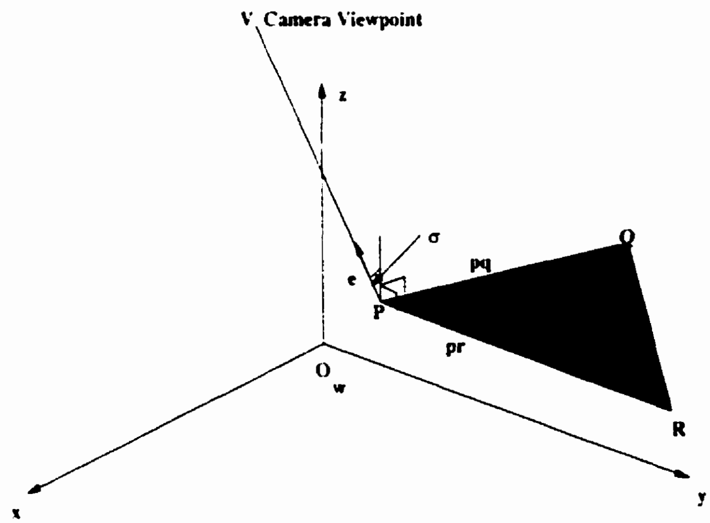


Figure 3.7: Computation of Facet Orientation

facet is oriented away from the camera viewpoint and that facet is not considered visible. In practice, we choose a threshold θ such that $0 \geq \theta < 90$. The average angular separation σ is then compared to theta. Only facets whereby $\sigma \leq \theta$ for all projected rays are considered to be non co-linear and hence visible.

The values of σ and the normal to the facet are computed as follows. Consider a facet with vertices positioned at coordinates $\vec{P} = (p_x, p_y, p_z)$, $\vec{Q} = (q_x, q_y, q_z)$ and $\vec{R} = (r_x, r_y, r_z)$ in 3D space relative to the world coordinate system as shown in figure 3.7. The unit vector $\vec{e}$ is the vector along the ray projected from the point P to the viewing position of the camera $\vec{V}$.

We compute the vectors $\vec{PQ}$ and $\vec{PR}$ as follows:

$$\vec{PQ} = \begin{pmatrix} q_x - p_x \\ q_y - p_y \\ q_z - p_z \end{pmatrix} \tag{3.15}$$

$$\vec{PR} = \begin{pmatrix} r_x - p_x \\ r_y - p_y \\ r_z - p_z \end{pmatrix} \tag{3.16}$$

The normal vector $\vec{n}$ to the facet is the cross product of $\vec{PQ}$ and$\vec{PR}$.

$$\vec{n} = \vec{PQ} X \vec{PR} \tag{3.17}$$

Hence we can compute the angle σ by finding the dot product as in equation 3.18.

$$\sigma = \cos^{-1} \vec{e} \cdot \vec{n} \tag{3.18}$$

3.3 Multi-Agent Systems

In this section we examine the fundamental theories and issues concerning multi-agent systems that provide the basis for the research presented in this thesis. The application of multi-agent technology to any problem domain is accompanied by its own unique set of requirements. Among these requirements are methods for coordinating the group of agents including but not limited to negotiation. conflict resolution and resource allocation schemes. In addition. effective and efficient communication amongst the agents is an important ingredient for facilitating the coordination of the group. Hence the protocols and pragma employed are also important to the success of the system in solving the problem at hand.

Many researchers have explored the problem of multi-agent coordination in variety of problem domains. As a result, several very relevant and plausible definitions for coordination have been established. For example, Ghenniwa and Kamel [12] have argued that coordination is a solution to the problem of interdependency. The authors define interdependencies as goal-relevant interrelationships between actions taken by various agents. Durfee and Montgomery [36] have defined coordination as the distributed search through a space of possibly interacting behaviours of individual agents and groups of agents to find a collective behaviour that satisfactorily achieves the agents' most important goals.

These definitions point to the importance of dealing with the interdependencies that may arise amongst the agents during the course of their actions. This can give rise to conflict situations amongst the agents. Conflicts arise when the agents choose incompatible actions, either because they base their decisions on different or incomplete information, or because they are trying to achieve different, possibly conflicting goals. Hence the notion of conflict resolution is important and fundamental to achieving a coordinated system. To resolve conflicts, systems of agents must interact, exchange information and possibly modify not only their actions but also their goals. These interactions are usually part of an overall negotiation process [37].

Researchers have attempted to address the notion of coordination from two main perspectives: experimental methods and formal methods. In the experimental based approach, the concept of coordination has been examined within a particular application domain for which a particular testbed has been developed. In the formal method, a more theoretical approach defined by mathematical models of such concepts as beliefs, intentions etc. have been developed. Since this research centres around the experimental approach to coordination, this approach is reviewed below. However, there are several publications that describe a more formal approach to coordination including the work carried out by Halpern and Moses [38] and Cohen and Levesque [39].

Within the context of experimental methods, there exist paradigms that further categorize the experimental approaches based on a priori assumptions. For example, the Functionally Accurate, Cooperative paradigm (FA/C) is based on the assumptions that the agents have common communication protocols, languages and representations of the environment. In addition the agents can assess the global effect of their collective behavior. The Functionally Accurate refers to the ability of the agent to produce acceptable plans even with inconsistent or incomplete data. The Cooperative refers to the agents' ability to interact with each other to revise and extend their tentative plans [12]

Research based on the FA/C paradigm includes the use of organizational models where the designer can specify the role of each agent, with whom it can interact with and the authority of the agent. As a result, the flow of information amongst the agents and their activities are controlled to a large extent by the organizational model of the system. Since each agent has defined roles, problems can easily be decomposed based on the roles of the agents. Such structures are usually static. One of the well known examples of systems that utilize this approach is the research carried out by Durfee et al in the Distributed Vehicle Monitoring Testbed (DVMT) system [29].

Improvements on the organizational model centred around a Partial Global Planning approach [40]. Here the aim was to make reasoning about coordination the responsibility of the agent and not the organizational structure. Hence, although the long term roles and capabilities of the agents were still defined by the organizational structure, the individual agents were able to reason to some extent about the other agents' plans. However, the structure still dictated between whom and when the exchange of these plans or meta-level information takes place.

The Contract-Net paradigm [33, 41] is another experimental approach to coordination which emphasizes the notion of task allocation. In this approach, task allocation is not predefined but determined as an agreement between a manager and a contractor. A manager agent forms a contract while the contractor agents bid on these contracts. Based on the bids received, the manager can assign a contract or task to a specific agent. The importance of the manager was offset by improvements to this paradigm in work carried out by Khun et al [42] where a more decentralized model was proposed. In this model, the society of agents as a whole was involved in the task decomposition and allocation without the help of a recognizable manager.

Another important experimental approach to the coordination problem was proposed by Yokoo and Durfee [43]. Here the authors address a specific class of problems known as constraint satisfaction problems(CSP). Yokoo and Ishida [44] define a constraint satisfaction problem as one that involves finding a goal configuration that satisfies all constraints defined for the problem, rather than finding a path to the goal configuration. More formally, we can define a CSP as m variables $x_1, x_2, ...x_m$, that obtain their values from domains $D_1, D_2, ...D_m$ respectively and a set of constraints on their values. A constraint is defined as a predicate whose parameters are the possible values of one or more of the variables. Hence, the constraint P_k is defined as $P_k(x_{k1}, x_{k2}, ...x_{kj})$ defined on the Cartesian product $D_{k1} X D_{k2} X ... X D_{kj}$. In this class of problems, the effective coordination of the system can depend on the efficient allocation of available resources or the feasible assignment of values to a set of variables, where each variable, or a subset thereof, is the responsibility of a given agent in the group. For example, from a sensor planning perspective, each agent is responsible for the assignment of the position and orientation of a camera under its control.

One approach to the problem of constraint satisfaction centres around the use of asynchronous backtracking algorithms [43]. These algorithms allow agents to run concurrently and asynchronously while providing a coherent framework for their execution and problem solving. The algorithm presented by Yokoo et al, assumes that each agent involved in the CSP has been assigned a priority. This could be based on simple alphabetic ordering of the agents or a more involved ordering process depending on the nature of the problem. Each agent then chooses a tentative value assignment to the variable or variables under its control and communicates its tentative value assignment to neighbouring agents. A neighbouring agent is one with which there is direct communication with the sending agent. An agent changes its current value assignment if it is not consistent with a higher priority process. The priority of the processes is common knowledge amongst the agents. If such a change is not possible (for example no consistent values are possible) then the lower priority agent must generate a "nogood" which is communicated to the higher priority process. The higher priority process would then change its value.

Each agent maintains the current value assignments of the other agents in the group and this forms the local view of the state of the system. The information concerning the current assignments is passed along by some communication protocol between neighbouring agents. It is possible for an agent to have an obsolete assignment for another agent. In this case, if a lower priority agent generates a nogood, the higher priority agent must also generate a nogood based on the request of the lower priority agent to change its value. Hence, before changing its value, the higher priority agent must verify that the lower priority agent has generated the nogood using the current assignment information.

We note here that the priority of the asynchronous backtracking algorithm is predetermined. Since higher priority agents have preference over the assignment of values initially, then a bad decision by a higher priority process could mean that the lower priority agents need to perform an exhaustive search in order to reverse the bad decision. As a result, some researchers have proposed methods of reducing the chances of the higher priority process making a bad initial decision. Such methods include the min-conflict heuristic and asynchronous weak-commitment search [44].

The former method is a value ordering heuristic. In other words, when a value is to be selected for a variable, the value that has the minimum number of conflicts with the other variables is selected. The latter method dynamically orders the priority of the agents so that a bad decision can be revised without exhaustive search. Each agent is given an initial priority of zero. The agent with the larger priority value will have the higher priority. If all agents have the same priority, they can revert to the priority based on alphabetical ordering. During problem solving, if an agent a_i with a priority k cannot find a value consistent with its local view of the state of the system, then that agent would send "nogood" messages to the other agents and increment its priority value. Also, the agents try to avoid previously encountered "nogood" situations. Eventually, another agent with previously higher priority will have to change its variable assignment in order to find an assignment consistent with that of agent a_i.

3.3.1 Decision Theoretic Agents

Irrespective of the type of coordination method employed, it is important for agents to make rational decisions when deciding on a course of action. An agent may need to refer to past experience as well as present circumstances when deliberating. In this section we examine some of the results obtained from research carried out in combining probability theory and utility theory to produce a decision theoretic agent.

A decision theoretic agent has the capability of making decisions even when given uncertain information and conflicting goals. Decision theory is essentially based on the maximization of an expected utility. This expected utility is a real number which describes the preference of an agent for a particular state. A given utility function can therefore be used to determine the behaviour of a particular agent.

In a nondeterministic environment, an action A on state S can produce several possible outcomes. Let $P(outcome_i(A))$ represent the probability that action A produces outcome i, where i ranges over all the possible outcomes. Let $U(S)$ be the utility or desirability of state S. Let E represent the summary of the agent's knowledge of the current state of its perceivable environment. The Expected Utility of the outcome produced by action A given evidence E is given by:

$$EU(A|E) = \sum_i P(outcome_i(A)|E)U(outcome_i(A)) \tag{3.19}$$

The principle of *Maximum Expected Utility* states that a rational agent should choose an action that maximizes its expected utility [10].

From the above we see that it is necessary for the agent to have some notion of the utility of the possible outcomes of its actions and the probability of these outcomes. This may be obtained from a percept history which can provide the statistical information necessary to compute or at least estimate the probabilities of the outcomes. The utility of the states can be obtained from the utility function which essentially defines the agent's behaviour. According to Russell and Norvig [10], if an agent's utility function accurately reflects the performance measure by which the agents behaviour is judged, then by maximizing its utility function, the agent would maximize its performance score when averaged over all possible environments in which the agent is acting. This idea is the central idea behind the maximum expected utility principle.

Suppose for a given action A on state S, there exists the possibility of two resulting states S_1 and S_2 such that the state S_1 occurs with probability p and state S_2 occurs with probability $1 - p$. We can utilize the following notation to express the agents preference for a particular state.

$S_1 \succ S_2$ State S_1 is preferred to state S_2.

$S_1 \sim S_2$ The agent is indifferent between S_1 and S_2.

As in formal logic we may impose constraints on the preferences. For example, the constraint of transitivity specifies that if $S_1 \succ S_2$ and $S_2 \succ S_3$ then $S_1 \succ S_3$. In order for an agent to be rational, the preferences of the agent must satisfy this constraint. Other constraints include order-ability, continuity substitutability and monotonicity [10].

According to the utility principle, if an agent's preferences obey the above constraints or axioms of utility, then there exists a real valued function U that operates on states such that $U(S_1) \geq U(S_2)$ if and only if state S_1 is preferred to state S_2 and $U(S_1) = U(S_2)$ if and only if the agent is indifferent to states S_1 and S_2.

In the situation where there are multiple factors that may affect the utility of a given state, the utility of an outcome for each factor may be combined to produce the overall utility for a particular state. For example, let $x_1, ...x_n$ represent the factors that affect the utility of state S. Then the utility of state U(S) can be given by

$$U(S) = f[u(x_1), ...u(x_n)] \tag{3.20}$$

where f is a simple additive function and u represents the individual utilities of the factors or attributes.

When making decisions, a rational decision theoretic agent can therefore utilize not only the current state of the world but also the experience it has gained from perceiving the world and interacting with other agents in the same environment. This gives the rational agent the ability to adapt its decisions based on new information obtained from other agents or the environment. By utilizing the principles of utility theory, an agent's decisions can reflect not only the current state of the system but also any experience it might have gained during deliberation.

Chapter 4

The Model

4.1 Introduction

The goal of this research is to design a system that can automatically generate the correct sensor configuration for a particular sensing task. The design should be extendible to any number and type of sensors. The system should also be able to accommodate multiple concurrent feature inspection[1], a problem not specifically addressed in the previous systems. Additionally the system should be coordinated and autonomously improve its performance with experience.

The assumptions made in this proposal are threefold and expressed as follows:

1. The agents have access to CAD models of the environment (in whole or in part) which contain precise measurements and geometric information about the environment including pose.

[1]The system can be used to inspect more than one feature at the same time.

2. The sensors are mobile with pre-definable ranges of motion. For example, the sensors may be attached to robot arms as in the eye hand configuration, or may have some other method of mobility in the 3D space.

3. Each sensor can be explicitly modeled such that all the parameters of the sensor are known a priori.

Having established the assumptions, we can utilize the collective computational ability of multi-agent systems to provide the means to automatically generate the correct sensor configuration in a multi-sensor environment. In this chapter, we illustrate an agent model that can coordinate its activity with other agents and improve its performance with experience while accomplishing a specified vision task. We begin with an examination of the form and type of data and structures available to the agents and the data collection method employed.

4.2 The CAD Model

We utilize a CAD[2] model of the environment under scrutiny to encode the spatial and geometric information required by the system [45]. The CAD model is then converted to a DXF format[3] consisting of the triangulation of all faces of all objects within the scene. The DXF file lists these triangles as lists of vertices grouped by face since each face may consist of one or more triangles. Curved surfaces are approximated by triangulation as shown in figure 4.1. From this type of

[2]Computer Aided Design

[3]Drawing eXchange Format

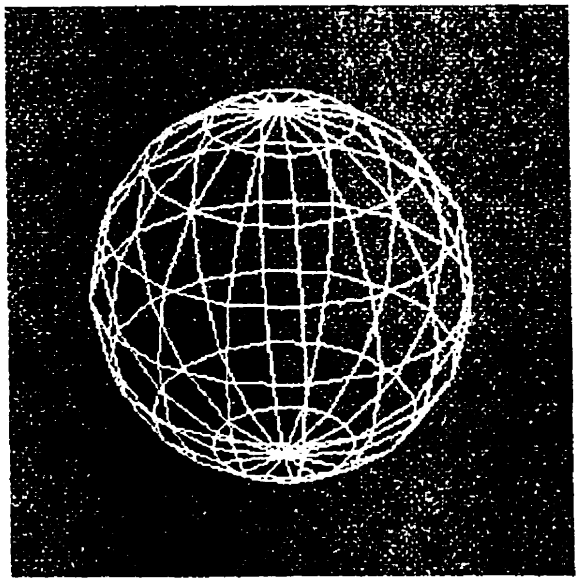


Figure 4.1: CAD Representation of a Sphere

representation, we can extract both edge and vertex information. Each triangle vertex is represented as a positional vector originated at the origin of the world coordinate system. Figure 4.2 shows a cube and the corresponding triangulation of the cube as represented in a DXF file. Each face is represented by the vertex list of the triangles that constitute the face. A sample listing of the DXF representation for one face of the cube shown in figure 4.2 along with the extracted facet information can be found in Appendix A.

We can calculate the length of any edge of any triangle as follows: Let triangle A be represented by the vertices $(\vec{r_a}, \vec{r_b}, \vec{r_c})$ such that $\vec{r_a} = a_x, a_y, a_z$, $\vec{r_b} = b_x, b_y, b_z$ and $\vec{r_c} = c_x, c_y, c_z$. In general, let k_x, k_y, k_z be defined as the XYZ components of the vector $\vec{r_k}$. Suppose we wish to compute the length of the edge segment $r_a\vec{r}_b$.

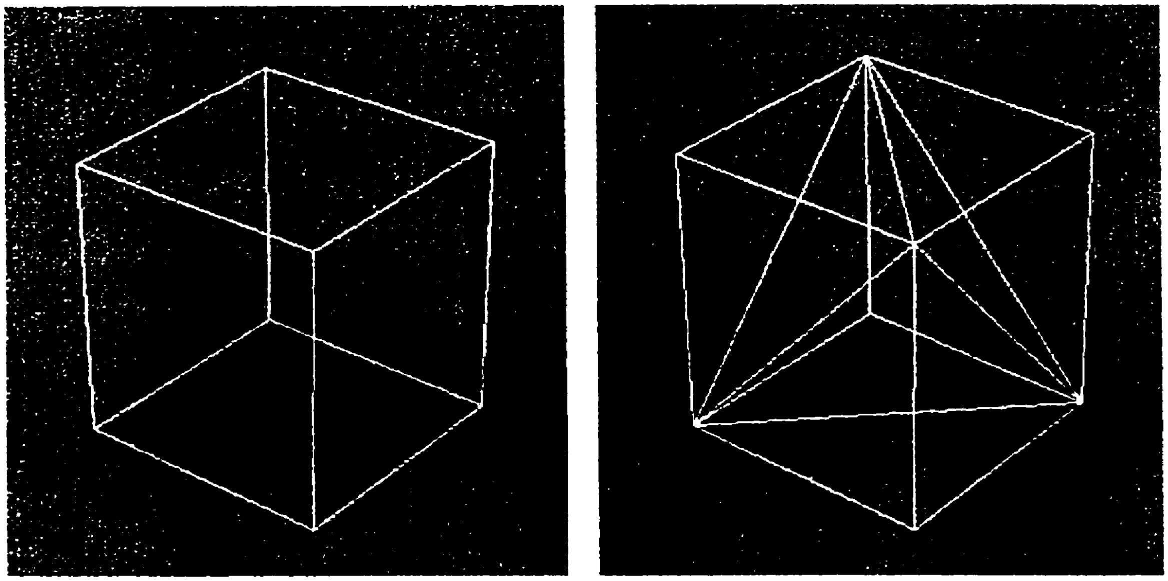


Figure 4.2: A Cube Model and its Triangulation

We first obtain a vector $\vec{d} = \vec{r_b} - \vec{r_a}$. We then compute the length of vector $\vec{d}$ as shown in equation 4.1.

$$|\vec{d}| = \left| \begin{array}{c} d_x \\ d_y \\ d_z \end{array} \right| = \sqrt{d_x^2 + d_y^2 + d_z^2} \tag{4.1}$$

Every non-zero vector $\vec{d}$ can also be normalized so that its length is equal to the unit vector $\vec{d_0}$. The normalized vectors allows us to compute the angles between two vectors in 3D space as shown in the previous chapter.

$$\vec{d_o} = \frac{1}{|\vec{d_0}|}\vec{d} = \begin{matrix} d_x/|\vec{d}| \\ d_y/|\vec{d}| \\ d_z/|\vec{d}| \end{matrix} \tag{4.2}$$

The edge and vertex information contained in the DXF file is thus sufficient for the computation of any higher level information required by the system. In the following sections we examine the knowledge structures that are used to store this information and the agent model that makes use of the stored information.

4.3 Camera Viewpoints

The system is based on the generate and test paradigm previously described in chapter 2. However, we do not utilize a geodesic dome or view sphere as described in the literature. Such a structure limits the camera positions to the surface of the dome and hence possibly more advantageous viewpoints in 3D space would be omitted from consideration. In order to generate a finite list of possible viewpoints, we utilize the boundary of the range of motion of the camera in the world coordinate system. Given a camera mounted in the traditional hand eye configuration [7], the camera has a range of motion along the three principle axes of the world coordinate system.

We can bound this range of motion by a polyhedron as shown in figure 4.3. Here we see a camera attached to the end of a robot manipulator and the corresponding bounding polyhedron. The polyhedron can then be subdivided into equally sized smaller polyhedrons or voxels. The centre of each voxel is a candidate viewpoint.

Figure 4.3 shows the subdivision of only a portion of the bounding polyhedron that is closest to the object being observed. The size of the subdividing voxels determine the number of viewpoints that are generated. Hence, for coarse subdivision we can choose a larger voxel size, while for finer subdivision we can choose a smaller voxel size. The granularity of the subdivision chosen depends on the field of views of the cameras.

The number of candidate viewpoints generated is based on the field of view of the camera involved. Generally, for cameras with a large field of view, the number of candidate viewpoints could be reduced since small movements of the camera will not necessarily result in a significant change in the scene. Cameras with smaller fields of view would require additional candidate viewpoints since a small change in the position of the camera could result in some objects moving in or out of the field of view.

4.4 Data Generation

Once the camera viewpoints have been generated, we obtain information regarding the depth of focus and visibility for each vertex in the target object. The depth of focus and visibility information is computed as described in sections 3.2.1 and 3.2.3 respectively. In addition, the resolution of the image on the image plane for each of line segments that constitute the target object is computed as described in section 3.2.2. The assumption is made that the optical characteristics of each camera is known a-priori. This information constitutes the sensor or camera model for the agent and is described in detail in section 4.5.1.

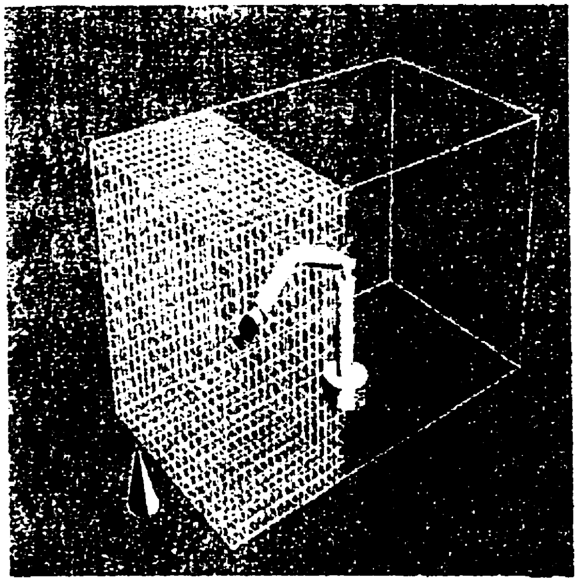


Figure 4.3: Bounding Polyhedron with Partial Voxelization

Using the sensor model. the agent can extract the geometric information required from the CAD model or alternatively, this information can also be provided off-line by an external pre-process. The resulting geometric information is stored in a database that is accessible by the agent controlling the camera. The data is in the form of an n-tuple where n is the number of features that are extracted from the DXF file. For our purposes we have chosen the following feature set.

1. *Viewpoint*: Contains the XYZ coordinates of the camera viewpoint with respect to world coordinates.

2. *View Orientation*: Contains the XYZ coordinates of the direction of the viewing vector related to the camera viewpoint with respect to the world coordinate system.

3. *Target Facet ID*: Each facet of the target object has a unique ID assigned for the purpose of identification.

4. *Facet Visibility*: The number of rays projected from the facet vertices to the camera viewpoint that are not occluded by any other object in the scene.

5. *FOV*: The number of vertices of the target facet that are within the field of view of the camera.

6. *Resolution*: The number of edge segments of a given target facet that meet the resolution constraint.

7. *DOF*: The number of vertices of the facet that are within the depth of focus of the lens.

8. *Facet Orientation*: Whether or not the facet is oriented such that its surface is visible or not.

4.5 The Agent Model

We present in this section a description of the agent model and the algorithms utilized for the coordination and adaptation of the agent with respect to the sensor planning problem. In order to facilitate the scalability and re-usability of the system in terms of the number of cameras involved in the planning process, we adopt an agent model that is generic enough for easy replication. However, the model may be tailored to suit the needs of specialized sensors or a specific sensor planning problem. In this system, each agent controls a single camera.

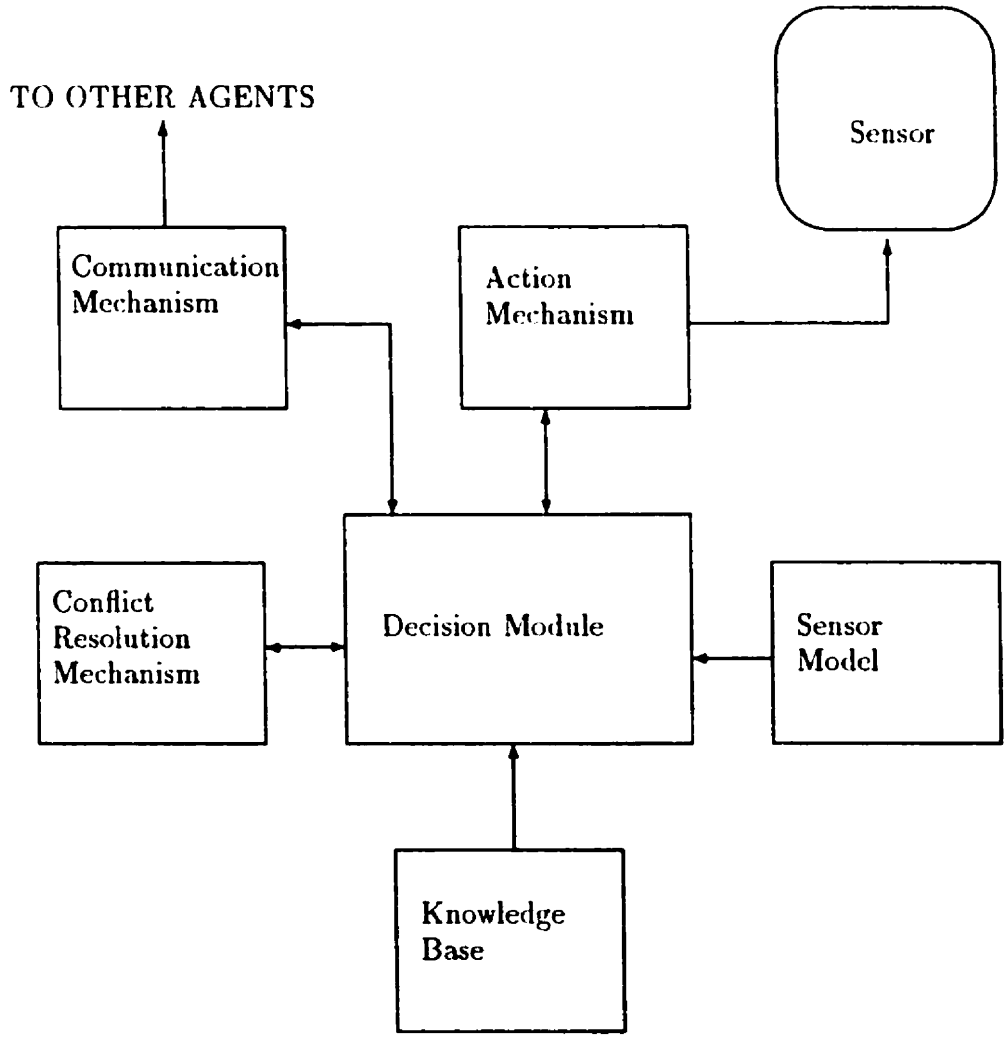


Figure 4.4: Basic Model of the Proposed Agent

Figure 4.4 shows the main modules of the agent. A description of each module follows.

4.5.1 The Sensor Model

The sensor model describes the capabilities and characteristics of the sensor. For example, in the case of a camera, the sensor model would contain such information as the aperture of the lens, the focal length and the range of mobility of the camera and any other information relevant or unique to the use of the sensor by the controlling agent. The information currently utilized include the following.

1. *Range of Motion* $(\delta x, \delta y, \delta z)$: This refers to the bounds on the motion of the camera relative to the coordinate world axes XYZ. This information is used to create the bounded polyhedron that is discretized to produce the candidate viewpoints as described in section 4.3.

2. *Lens Aperture Setting* (a): This refers to the diameter of the entrance pupil of the lens system.

3. *Focal Length* (f): The focal length of the lens.

4. *Back Nodal Point to Image Plane Distance* (d): The distance between the back nodal point of the lens and the image plane.

5. *Minimum Dimension of the Image Plane* (I_{min}): This may be lesser of the width or height of the image plane.

6. *The Minimum Dimension of the Blur Circle on the Image Plane* (c_b): Used as described in section 3.2.1 for the computation of depth of field of the lens system.

4.5.2 Action Mechanism

An action is defined herein as a change made to the parameters that specify the configuration of a sensor. The action mechanism is an interface to the machinery that implements such changes on the actual sensor. This provides a uniform interface to the decision module and abstracts it from the intricacies of the actual sensor mechanics. For example, in order to move the camera to a specified position, the decision module would simply give the coordinates to the action mechanism. It is up to this mechanism to provide the necessary commands to get the camera there. The action mechanism may be as complex or as simple as the situation warrants. The level of complexity depends on the type and capabilities of the actuator. In this thesis, the assumption is made that the cameras are mounted on robot manipulators in a *hand eye* configuration in order to achieve the necessary mobility in 3D space. Although we do not address the notion of path planning in order to position the camera at the desired coordinates, designing the action mechanism as an independent subsystem provides the level of abstraction necessary for additional computation to accomplish the required path planning.

It is also possible for the cameras to be limited in some component of the overall range of motion. For example, if the cameras are attached to mobile robot that is capable of movement along a surface such as a floor or table top, then the movement of the associated camera is limited to two dimensions. The action mechanism in this case serves as the interface between the controlling agents and the actuators of the mobile robot.

The above description assumes that the camera is being positioned by some automated means. In the case that a human operator is positioning the camera, then the decision module would provide the human operator with the appropriate camera coordinates.

4.5.3 The Knowledge Base

The knowledge base contains the CAD information about the scene that the agent utilizes in order to decide on the appropriate actions. The main purpose is to provide the decision module with the necessary measurements that would facilitate the computation of the appropriate utility values for a given viewpoint. The knowledge base is initially supplied to the agent. However, the agent will only need to be aware of the portion of the environment that it can affect. Hence partial knowledge of the environment is admissible and desired in order to decrease the storage requirements of each of the agents.

The actual information describing the scene may be of two types. The first type is the actual CAD information consisting of a DXF file with the vertex lists of the objects in the scene as described in section 4.4. The targets in the scene are clearly labeled beforehand so the agent has complete information regarding the targets. Using this information, the agents can extract the necessary feature information such as the target vertices that are in view and in focus and the corresponding edge segments that are resolved from a given vantage point. This information is obtained by utilizing the methods described in the previous chapter for computing field of view, resolution, depth of focus and visibility. Once this information is obtained, it is stored in the knowledge base as a feature data file. This process is carried out prior to the start of the problem solving phase for each agent.

Alternatively, the agents may be given the resulting feature data file directly instead of the CAD model. In this case the feature data is produced by an off-line feature extraction process that provides each agent with the data relevant to its range of motion. The agents would then use this data as they would if it had been produced during its pre-processing feature extraction stage. By restricting the feature data used by the agent to the targets and objects within its region of influence, an agent need only have partial knowledge of the scene provided that this knowledge is sufficient to enable the agent to make rational decisions. This becomes important for very large scenes where one or more regions of the scene may be inaccessible by the field of view cone of the camera due to the large size of the objects involved. In this dissertation, the region of influence of an agent's sensor is obtained by including all objects in the scene that are that are within the bounding polyhedron containing all the field of view cones from the candidate viewpoints that constitute the range of motion of the camera. This is possible since we set the orientation of the camera towards the centroid of the target in the scene. For more than one target, the region of influence becomes the union of the resulting bounding polyhedra.

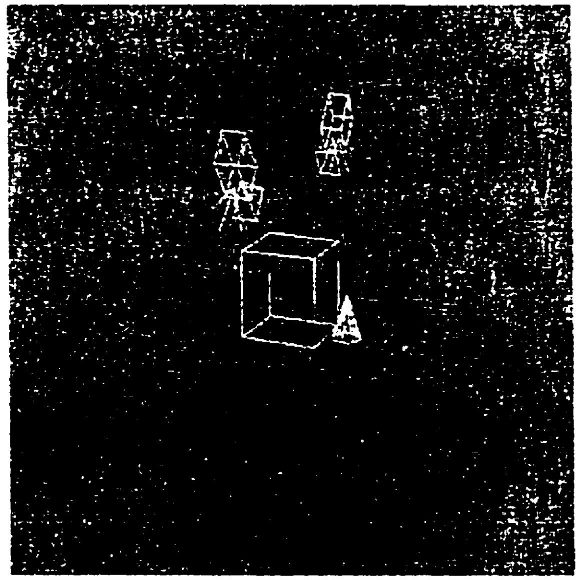


Figure 4.5: Simple Scene using 2 Cameras

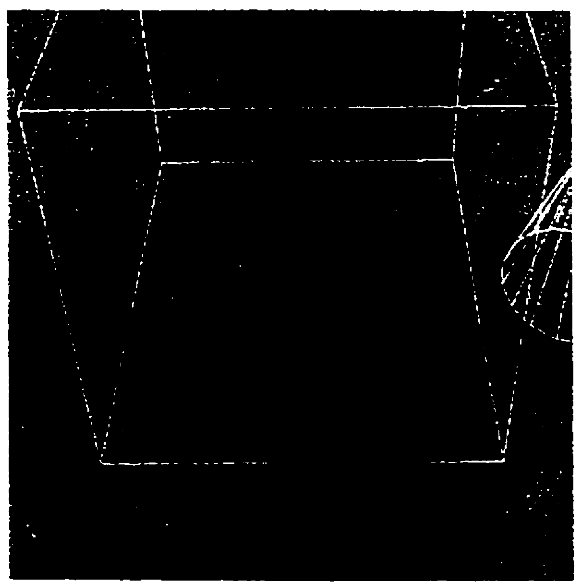


Figure 4.6: Scene from Viewpoint 1

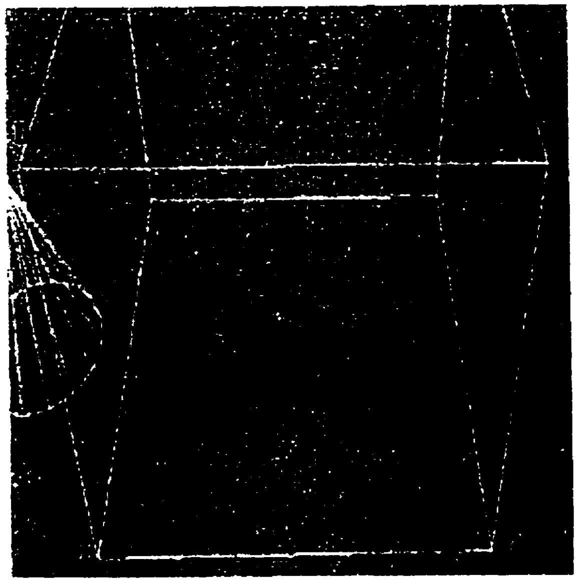


Figure 4.7: Scene from Viewpoint 2

ID	Facet	VIS	DOF	RES	FOR
1	ABF	3	3	3	3
2	FEA	3	3	3	3
3	ACD	1	3	3	3
4	DBA	2	3	3	3
5	CDH	0	3	3	3
6	HGC	0	3	3	3
7	AEG	1	3	3	0
8	GCA	2	3	3	0
9	BDH	1	3	3	0
10	HFB	2	3	3	0
11	EGH	1	3	3	3
12	FEG	2	3	3	3

Table 4.1: Data for Viewpoint 1

ID	Facet	VIS	DOF	RES	FOR
1	ABF	0	3	3	3
2	FEA	0	3	3	3
3	ACD	2	3	3	3
4	DBA	1	3	3	3
5	CDH	3	3	3	3
6	HGC	3	3	3	3
7	AEG	1	3	3	0
8	GCA	2	3	3	0
9	BDH	2	3	3	0
10	HFB	1	3	3	0
11	EGH	2	3	3	3
12	FEG	1	3	3	3

Table 4.2: Data for Viewpoint 2

As an example of the feature extraction process, consider the scene shown in figure 4.5. Two cameras are oriented towards the centroid of the cube. From the viewpoints chosen, some vertices are occluded or outside the field of view of each camera as shown in the corresponding camera views of figures 4.6 and 4.7 corresponding to viewpoints 1 and 2 respectively. Specifically we note that in view 1, vertices C and D are outside the field of view of the camera and vertices G and H are occluded by face $ABEF$ of the cube. Similarly, in view 2, the vertices A and B are outside the field of view of the camera and the vertices E and F are occluded by the face $DCHG$.

The corresponding feature data is shown in table 4.1 and table 4.2. The attributes of visibility(VIS), Depth of Focus (DOF), Resolution (RES) and Facet Orientation (FOR) are computed for each triangular facet of the cube. The numbers represent the total number of vertices that meet the constraint for a given facet except in the case of the resolution attribute which refers to the number of edge segments. Visibility is considered to be a complex attribute since it depends on the following rules:

1. If a vertex is outside the field of view it is not visible.

2. If a vertex is inside the field of view but occluded by the same object or another object in the scene then it is not visible.

3. If a vertex A is co-linear with another vertex B of the same or different triangular facet in the scene such that a ray projected from vertex A to the camera viewpoint V passes through vertex B prior to reaching V then vertex A is not visible.

This information essentially describes the view obtained by the camera from a geometric perspective. Hence from table 4.1, we see that the facets CDH and HGC are not visible from viewpoint 1 but they do satisfy the constraints of depth of focus and resolution. Since these facets constitute the face $DCHG$, then face $DCHG$ is not visible from the given viewpoint. Intuitively, visibility is the most important attribute since even if a vertex is visible and out of focus, the camera lens system can be changed to bring the vertex into focus if necessary. However, in this system, the agents would prefer a viewpoint that satisfies all of the constraints and this is based on the assumption that the camera optical properties are specified and set prior to runtime.

The facet orientation field specifies whether or not a given facet is oriented such that the surface normal of the facet is within an angular threshold of the viewing direction of the camera. The data indicates that the surface of facets AEG, GCA, BDH and HFB are not oriented within this threshold. Hence as far as the system is concerned, these surfaces are not visible even though the other constraints have been met.

4.5.4 The Communication Mechanism

The communication mechanism essentially allows the agent to communicate with other agents via a prearranged protocol or suite of messages. The type of message sent and the information contained therein is ultimately decided upon by the decision module. The agents utilize a protocol based on the Knowledge Query Manipulation Language (KQML) specification [46]. This specification provides a concise and easily implemented protocol for inter-agent communication. All the information necessary for the correct interpretation of the message is included in the message itself.

The format of the protocol used in this thesis is as follows:

Message ID This is a monotonically increasing number generated by the sending agent. This helps to determine the order in which the messages should be read by the receiving agent.

Sender ID The identification of the sender.

Receiver ID The identification of agent to whom the message is addressed.

Message Type The message type determines response of the receiving agent to the received message.

Message Content The information being sent to the receiving agent.

The sender ID and the message ID together form a unique identifier for each message received by an agent. Messages can be sent to a particular agent by including the agent ID in the receiver ID field or by specifying the receiver ID as *. There are three categories of messages that are communicated amongst the agents. The categories are queries, solicited assertions and unsolicited assertions.

Queries consist of the following message types:

RNR Random Number Request. Using this message type, an agent can request a random number from another agent.

PING If agent a has not received a communication from another agent b within a specified time period, agent a may send a *PING* query to see if agent b is still active. Agent a will not send any further communication to agent b until a response is obtained from agent b. A *PING* query solicits an immediate response by the agent. If no response is received, that agent is assumed to be inactive. If an active agent receives a *PING* request from another agent, it responds with an *ACK* or positive acknowledgment described below.

FP Final Position. This message type is used when an agent needs to solicit the agreement/disagreement of its final choice of viewpoint from the group of agents. The agents may reply with an *ACK* or agreement with the choice of final position or a *NAK* which indicates disagreement with the choice of final position. If at least one agent replies with a *NAK*, the receiving agent is obliged to reconsider its choice of final position and continue the negotiation process. If all agents respond in agreement with the agent's final position then the receiving agent can terminate further negotiation with the group. The sensor controlled by the agent is then positioned at the viewpoint chosen by the agent. This is accomplished via the action mechanism. Alternately, the position chosen can be communicated to a human operator via the action mechanism.

Solicited assertions are messages that are in response to a given query such as those presented above. The solicited assertions used in this protocol are:

ACK A positive response/agreement to a *PING* or *FP* query.

NAK A negative response/disagreement to an *FP* query: an agents desire to choose a particular viewing position.

RND The response to a request for a random number or *RNR* message type.

Unsolicited assertions are broadcast messages that provide information about a specific agent to the rest of the group of agents as soon as that information becomes available. These types of messages are generated as a result of a change in the agent's state or decision. For example a change in state occurs if the agent terminates its negotiation and a change in decision occurs if the agent changes its camera viewpoint.

The unsolicited assertions used in this protocol are:

FVL A message containing the coordinates of the camera position desired by the agent and the list of vertices of the target object that satisfy the constraints of visibility, field of view, depth of field, resolution and orientation. The message also contains the utility measure of this viewpoint which is a measure of the preference of the agent for the chosen viewpoint.

ALU Adjusted Local Utility. This message type signifies the communication of utility information that has been adjusted due to the receipt of previously unknown information from the other agents in the group.

TERM This message type signifies that the agent has decided to terminate further communication with the group. Such a message type is usually generated when the agent has decided that further negotiation will not yield any significant improvement to the current result. As mentioned previously, the agreement of the other agents must precede the agents decision to terminate its negotiations.

The design of the communication system is based on two very important assumptions. The first is that all messages take a finite amount of time to reach the recipient. Hence no messages are lost. This assumption can be justified for the purpose of our experiments since in a real world implementation, the proper transport protocols could be put in place to assure that messages are guaranteed to be delivered or retransmitted if necessary. The second assumption is that the time taken for a message to travel from sender to receiver can vary from message to message. The latter assumption is based on the fact that the dissemination of messages is highly implementation dependent. For example, if all the agents are executed on a single processor system, messages may not have equal delivery times due to the effects of time slicing. In our model therefore, it is possible for agents to make decisions based on outdated information. It is the responsibility of the coordination mechanism to ensure that such decisions are recognized and rectified in an appropriate and efficient manner.

4.6 The Decision Module

The decision module is actually a subsystem consisting a several components that interact to allow the agent to arrive at a rational decision based on the current state of the system and also past decisions. The configuration is illustrated in figure 4.8. It consists of a coordination algorithm that makes the decisions as to which viewpoint the agent chooses, a mental model that keeps track of the current state of the system and an action history that records the actions of the agent. The conflict resolution system interacts with decision module when a conflict arises in an effort to solicit a decision that would resolve the conflict. In this problem domain, a conflict arises when two or more agents decide to occupy the same viewing position or alternatively, when the distance separating two is below a user defined threshold. In both cases, this situation would result in unacceptable overlap amongst the fields of views of the cameras. The conflict resolution process is described in detail in subsequent sections. What follows is a description of each of the components of the decision module and their interactions and effect on the decision making process of the agent.

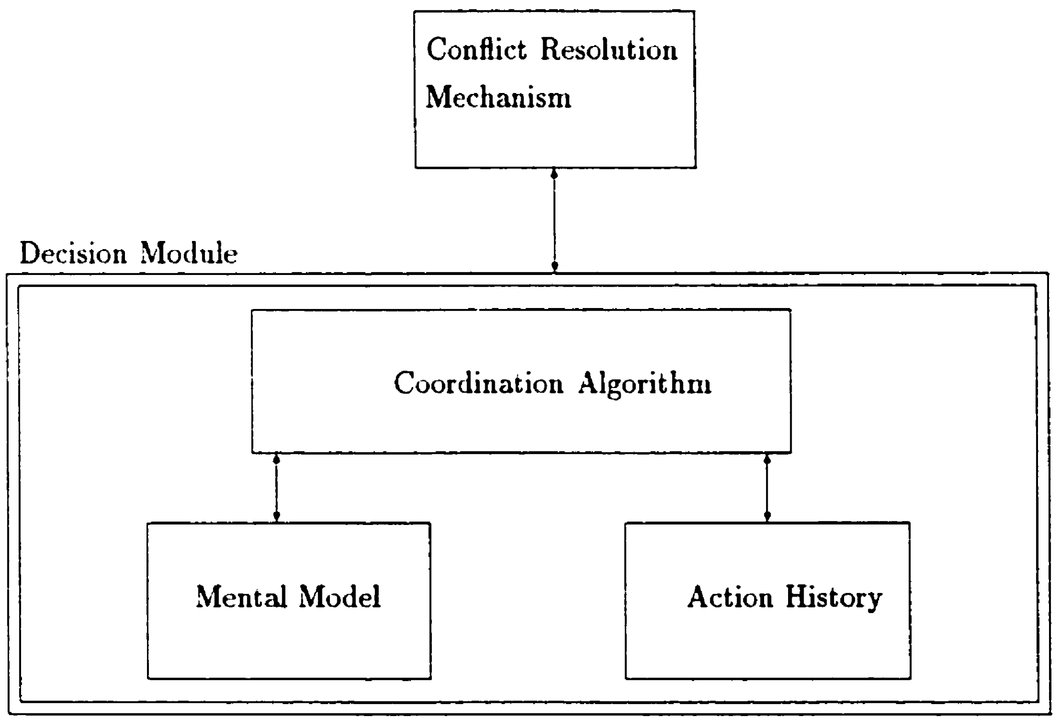


Figure 4.8: The Main Components of the Decision Module

4.6.1 The Mental Model

The mental model serves to provide a means of collecting information on the decisions of other agents within the group. The data is collected during the negotiation process from the other agents by means of the communication protocols previously described. The information stored in the mental model consists of the current intentions of the agents with reference to their choice of viewing position, their communicated contribution to the global utility and the identifiers of those facets of the target that meet the task constraints from their choice of viewing position. If an agent is no longer communicating with the rest of the group, this is also indicated in the mental model of the rest of the agents in the group. In addition, the mental model of any given agent indicates all those agents with which its current choice of viewing position is in conflict. This is referred to as the current conflict list.

4.6.2 The Action History

The action history database maintains a record of all the actions or choices of the agent and the corresponding reward or utility of the action. The database also contains a number representing the number of agents that were in conflict with that choice of action. This information allows the agent to generate an informed hypothesis concerning the possibility of a conflict occurring when a decision is made that is very similar or identical to a previous decision in its action history. Previous decisions that ultimately produced low utility values or conflict situations will have a lesser chance of being repeated during further negotiations. This assists in preventing oscillating behaviour where an agent may continuously oscillate between decisions that were initially expected to yield a high utility but which were subsequently proven to be in fact bad decisions.

4.6.3 The Coordination Algorithm

The core of the decision module is the coordination algorithm or CA. This module is responsible for generating the actions and the communication to the other agents. It is also responsible for invoking the conflict resolution mechanism when necessary and maintaining all the associated histories and databases within the decision module. The coordination algorithm is also responsible for recognizing when an agreement has been reached amongst the agents. Figure 4.9 illustrates the algorithm in flowchart form.

In order to choose one action over another, the coordination algorithm relies on the notion of a utility function as defined in section 3.3.1. Based on this definition, we present here a description of utility that is specific to this system. The algorithm utilizes two type of utility measures, namely the local utility and the global utility. These are described as follows.

The local utility of an agent's action μ_l is defined as the reward computed by the agent based on the degree to which the particular choice of viewing position meets the constraints of the sensing task and avoids potential conflict.

The global utility μ_g refers to the degree to which the combined actions of the agents meets the requirements of the sensing task. An agent is initially aware only of its local utility. However, as the negotiation process proceeds, the agent ob-

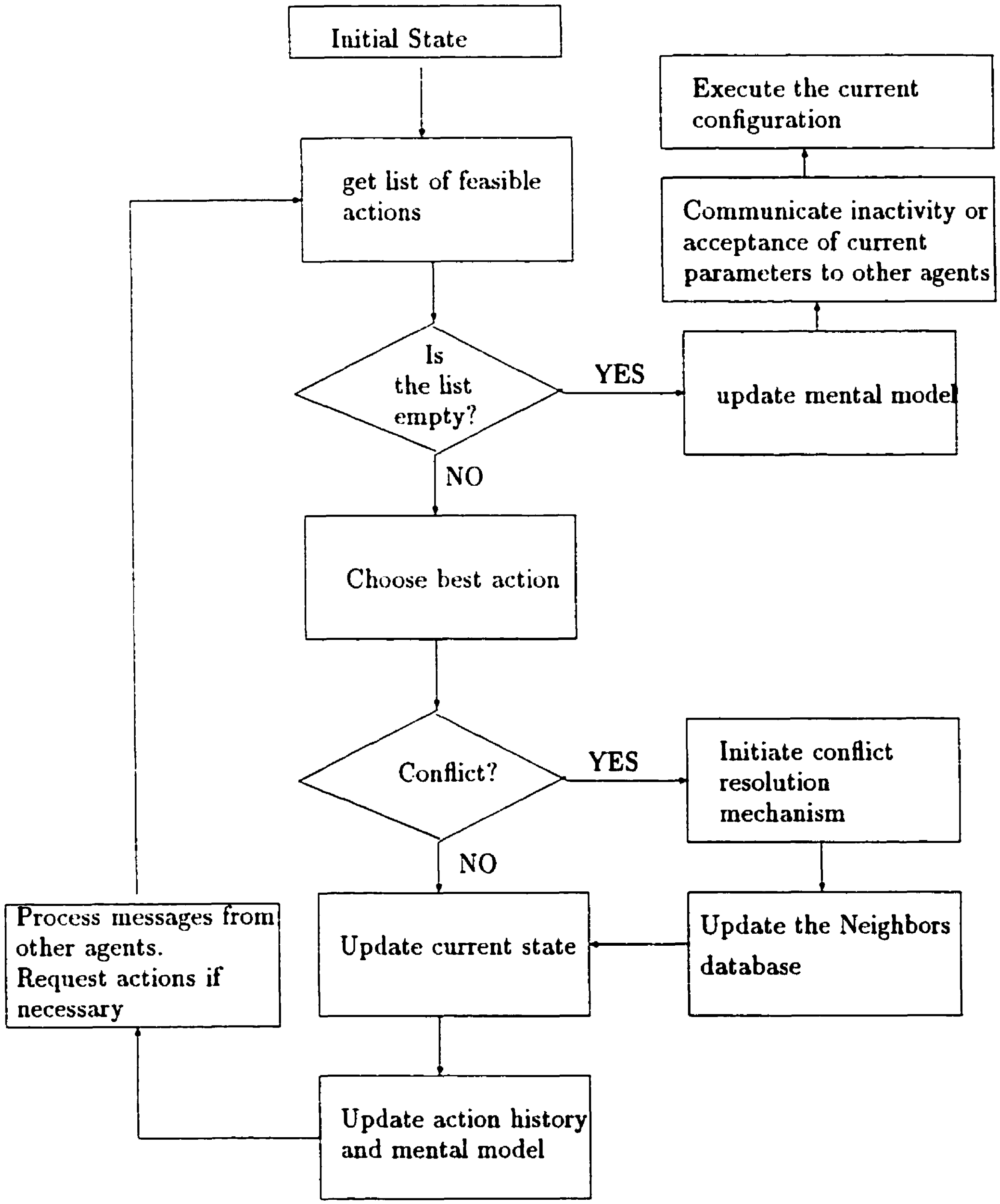


Figure 4.9: Flowchart of the Coordination Algorithm

tains information from the other agents concerning their respective desires in terms of viewing positions. Using this information, the agents can begin to formulate hypotheses about the global utility of their combined actions.

Let μ_l^i represent the local utility as computed by agent i. The local utility is based on the number of vertices that meet the task requirements of visibility, field of view, depth of focus, surface orientation as well as the number of edge segments of the target facets that meet the resolution requirement. We can express the local utility as in equation 4.3.

$$\mu_l^i = \beta_1 \frac{V_{fv}}{V_{total}} + \beta_2 \frac{V_{fov}}{V_{total}} + \beta_3 \frac{V_{dof}}{V_{total}} + \beta_4 \frac{E_{res}}{E_{total}} + \beta_5 \frac{F_{or}}{F_{total}} \tag{4.3}$$

Where, V_{fv} is the number of vertices of the target object that are not occluded by any other object in the scene or any part of the target object. V_{fov} is the number of vertices of the target object that are within the field of view of the camera. V_{dof} refers to the number of vertices of the target object that are within the depth of focus of the camera. E_{res} refers to the number of edges of the target object that are within the limits of resolution of the camera. The denominators V_{total} and E_{total} refer to the total number of vertices and edges respectively of the facets that constitute the target object. F_{or} refers to the number of facets that have a surface normal within a given angular range of the viewing direction of the camera. F_{total} refers to the total number of facets.

The weights $\beta_1, \ldots, \beta_5$ indicate the relative importance of the individual constraints to the computation of the local utility. This provides a mechanism for specifying the behaviour of the agent. By increasing or decreasing the respective weights, the designer can specify the level of importance of any of the task constraints prior to the onset of negotiations. Hence, an agent with a relatively high weight associated with visibility would tend more to viewing positions that provided visibility for the target vertices, while at the same time, ignoring the fact that the same vertices may not be in focus or the edge segments may not be resolved. The local utility therefore provides a measure as to how well the chosen viewing position satisfies the constraints of the task.

Similarly, we define the global utility as the weighted sum of the individual local utilities of a set of agents $A = \{a_1, a_2, \ldots, a_m\}$ as illustrated by equation 4.4.

$$\mu_g = \sum_{i=1}^{m} w_i \mu_l^i \qquad (4.4)$$

The weight of each local utility w_i specifies the importance of that agent in contributing to the global utility. The contribution of an agent to the global utility is referred to as the agent's confidence. Therefore, the weight w_i directly affects an agent's confidence. Hence, during negotiations, agents with higher respective weights associated with their local utilities would tend to keep their decisions in a conflict situation and allow the less confident agents to change their desires more readily.

Once a sensing task has been broadcast to the agents, the decision module selects a list of possible actions that satisfy its constraints locally. For example, the agent would choose viewing positions that satisfy the visibility, field of view, resolution, depth of focus and surface orientation constraints as previously described. Out of this list of possible actions the coordination algorithm then chooses the "best" action based on the maximization of a local utility. Any prior information such as the probability of causing a conflict can be utilized in calculating the local utility. However, in the initial stages, this information would not be available. Hence the selfish maximization of the local utility is, at this point, the highest priority for the agents and therefore the choice of viewpoint is made using a greedy selection criteria [47].

The agent's choice of action depends on its preference for a specific action. We can express preference in terms of the entities that are fundamental to decision theory, namely utility and probability. Here a rational agent such as ours will choose an action that will yield the highest expected utility averaged over all possible outcomes of the action. This is known as the principle of Maximum Expected Utility[10] and is explained in section 3.3.1. Hence, not only is the expected utility of an action important but of equal importance are the probabilities that the action will (not) cause a conflict as computed by equation 4.6 and that the same action will increase the expected global utility. We can therefore adjust the expected utility of any particular action by these probabilities. This provides a more robust decision making system and allows for the experience of the agent to influence its decisions.

The prior information that is necessary to formulate such probabilities associated with decision making comes from the data acquired over time in the action history, and the mental model of each agent. Therefore, the probability of an action causing a conflict and the probability of the same action improving the global utility is learned over time. Hence the correctness of the decisions made by the agent can be improved over time.

When an agent chooses a particular viewing position, it may do so without any knowledge of the effect that its choice will have on the other agents. Hence, it initially calculates an expected local utility for that viewing position. However, the agent must communicate with the other agents to inform them of its choice and the corresponding parts of the target object that is within its field of view. It does so via the FVL message type as previously defined.

Upon receipt of another agent's intended camera position, an agent checks to see if the sending agent's desired action is in conflict with its own desired action. If there is a conflict, the conflict is resolved using either of the methods described in section 4.6.4. If no conflict exists, or after a given conflict has been resolved, the agents proceed to refine their initial estimates of their local utilities by taking into consideration the information communicated to them by other agents within the group. This refinement takes into account any redundancy in the fields of view of proximal viewpoints and also the possibility of a viewpoint conflicting with another agent's choice. The local utility is therefore adjusted as follows.

$$\mu_l^i(t+1) = \mu_l^i(t) - \gamma \frac{V_{int}}{V_{total}}(1 - P_c) \tag{4.5}$$

$$P_c = \begin{cases} \frac{N_c}{s(N_{total}-1)} & \text{if } s \geq 1 \\ 1 - s & \text{otherwise} \end{cases} \tag{4.6}$$

Where $\mu_l^i(t)$ is the local utility of agent i calculated at time t and $\mu_l^i(t+1)$ is the local utility of agent i adjusted at time $t+1$. V_{int} refers to the number of vertices that are within the field of view of agent i and also in the field of view of one or more of the other agents. This gives an indication of the degree of overlap of the fields of view of two or more agents with respect to the total number of vertices V_{total}. The parameter γ is the utility adjustment parameter and it determines the degree to which the overlapping fields of view affect the local utility. The exact value of γ can be obtained by empirical observation for a given problem set. The variable s refers to the average separation (distance) between the desired position of agent i and the positions of other agents within the group for a particular viewing position. The variable N_c is the number of agents that have been in conflict with agent i and N_{total} is the total number of agents within the group. From equation 4.6 we see that the influence of the conflict adjustment is lowered if the agent's choice of viewing position is further from the other agent's choices. Therefore a highly advantageous but previously conflicting viewpoint may be chosen if it is not in close proximity to some agent's desired position. This computation provides the agent with a probability measure of its desired position being in conflict with another agent's desired position.

The expected local utility is adjusted based on the information received during communication with the other agents. An agent penalizes itself for viewing the same area of the target object as other agents and also tries to choose a viewing position that separates it from the viewing positions of the other agents. Under normal circumstances (no conflict occurring) the order in which the agents iteratively improve their choice of viewing position is based on the agents' confidence. Once an agent has adjusted its local utility, it broadcasts this information using the ALU message type. The agents then compare this adjusted local utility to their own. The agent that can contribute the maximum amount to the global utility is allowed to commit to its choice of viewpoint. The other agents automatically assume that such a commitment will be made. If two or more agents are contributing to the global utility by the same amount then the agents rely on their seniority ordering to decide which agent moves first. The concept of seniority is explained in section 4.6.4. This set of social laws provides a level of organization to the system which would otherwise be imposed by the designer. Hence the autonomy of the system is maintained.

An agent will only change its viewing position if such a change will solicit a significant contribution to the global utility. Hence, given viewing positions $P_1 = (x_1, y_1, z_1)$ and $P_2 = (x_2, y_2, z_2)$ and a threshold ϕ, an agent i will change its position from P_1 to P_2 only if the inequality of equation 4.7 is satisfied.

$$\|E(\mu_l^i \| P_1) - E(\mu_l^i \| P_2)\| > \phi \tag{4.7}$$

Where $E(\mu_l^i \| P_x)$ is the expected local utility at position P_x.

The expected change is used here since the agent's computations are based on its current mental model of the world. If an agent cannot find a position that would improve on its current contribution to the global utility by an amount greater than ϕ , the agent broadcasts an FP type message. This indicates to the other agents that the sending agent intends to make its current position its final position. Upon receipt of this message type, each receiving agent revises its local utilities and orders their possible viewing positions accordingly. If there is any viewing position that can yield a higher utility than the current position of the receiving agent. such that the change in utility is greater than ϕ. then that agent responds with a NAK or negative acknowledgment and the negotiation process is repeated. However, if all the receiving agents agree that there is no other position more advantageous to them. they all respond with an ACK or positive acknowledgment. This terminates the negotiation session and the coordination algorithm.

If an agent changes its current viewpoint, it must immediately inform the other agents by broadcasting an FVL message type. Upon receipt of the FVL message type. all receiving agents will reevaluate their local utilities. Hence in a system with extremely overlapping ranges of motion of the cameras, is possible for an agent that initially thought that it had chosen a very advantageous position based on its local utility measure. to find out that the position is not so advantageous once more information is obtained from the group. This allows the agents with initially high expected utilities to backtrack and possibly choose a position that can better contribute to the global utility based on new information. Since agents will only change viewpoints if the change in utility is greater than a given threshold, this prohibits the agents from oscillating between viewpoints that have similar expected utility values.

An agent may be unable to send an FVL if it looses communication with the group. Each agent maintains a list of the other agents that are actively participating in the negotiation process. If no message is received from a particular agent in the group over a specified time period, a PING message is sent to that agent. An acknowledgment to the PING message renews the entry in the agents list of participating agents. If no acknowledgment is received after a specified number of PING broadcasts, the sending agent removes the inactive agent from its list of active agents. The negotiation process therefore proceeds as if the inactive agent never existed.

If a desired action causes a conflict with another agent, then both agents will invoke their conflict resolution mechanisms as described below. The agent whose desired action causes the largest gain in utility is committed to that action. The rest of the group accepts the effects of this action as the current arrangement set. If after a series of negotiations, all agents broadcast acceptance messages, then the current arrangement set becomes the agreement set. The mental models of the agents are updated and the arrangement is executed by the action mechanisms of the agents.

4.6.4 The Conflict Resolution Mechanism

Although each agent would try to choose actions that are not in conflict with other agents, the situation is expected to arise where conflict is unavoidable. In this system, such situations are more likely to occur when the range of motion of the cameras controlled by the agents severely overlap, or the number of possible viewing locations is relatively small. The conflict resolution mechanism aims to provide a means of resolving such circumstances. The basic idea behind conflict resolution as proposed in this research is the idea of confidence. Formally the concept is defined herein as follows:

An agent is *confident* if according to its utility evaluation functions, its choice of action will produce a change in global utility that is considerably more than the average expected changes in global utility of the other agents with which it is in conflict. More formally, an agent is confident if the following is true.

$$E(\Delta\mu_g^i + \alpha) > \frac{1}{n-1}\sum_{j\neq i} E(\Delta\mu_g^j) + \epsilon 1 \leq j \leq n \tag{4.8}$$

Where, n is the number of agents. $E(\Delta\mu_g^i)$ is the expected change in global utility by the action of agent i. ϵ is a small positive constant such that $0 < \epsilon < 1$ and α is a small random positive constant such that $0 < \alpha < 1$. α is included as a means of adding a small amount of randomness to the system in an effort to avoid deadlock situations.

The proposed conflict resolution algorithm common to all agents is presented below. Assume that the agent under scrutiny is agent i.

Let CONFLICT-LIST be the list of n agents with which agent i is in conflict.

WHILE NOT EMPTY CONFLICT-LIST do

If agent i is confident:

$$E(\Delta\mu_g^i + \alpha) > \frac{1}{n-1}\sum_{j\neq i} E(\Delta\mu_g^j) + \epsilon 1 \leq j \leq n \tag{4.9}$$

then agent i commits itself to that action

else

if agent i lacks confidence, i.e.

$$E(\Delta\mu_g^i + \alpha) < \frac{1}{n-1}\sum_{j\neq i} E(\Delta\mu_g^j) - \epsilon 1 \leq j \leq n \tag{4.10}$$

then agent i re-assesses its choice of action and proposes a new action. If this action is not in conflict with any other agents desire, then agent i updates its mental model and broadcasts its new desired position to the rest of the group.

END-WHILE

The resolution of conflict situations depend on the ability to prioritize the agents such that the group of agents know explicitly which agent will change its viewing position in order to resolve the conflict. As mentioned in chapter 3, it is possible to impose such a priority on the system prior to the negotiation process. However, this decreases the autonomy of the system. By using the notion of confidence, we allow the agents to decide the agent priority without human intervention.

In a deadlock situation where it is impossible to decide on a least confident agent, the agents can utilize a random number to decide which agent should make a different decision so as to resolve the conflict. The agent with the highest random number has the highest seniority and similarly, the agent with the lowest random number has the lowest seniority. The agent with the lowest seniority will attempt to change its decision first. In the event that no decision can be made by the least senior agent that would resolve the conflict, then the least senior agent informs the most senior agent of the situation by disagreeing with its choice of action. This is accomplished by sending a negative acknowledgment (NAK) to the most senior agent.

The most senior agent would then choose a different viewing position based on the fact that the utility of the previous viewing position has been decreased by the probability of causing a conflict situation. Hence, the algorithm attempts to implement the concept that a confident agent should be allowed to commit to its chosen action while the less confident agents should reconsider their desired action. However, it also takes into account that the more confident agents may need to change their choice of viewing position in order to allow a more suitable arrangement to exist.

4.7 The Coordination Algorithm

The actual algorithm utilized by the agents for coordination is the same for all agents. This offers the advantage that the system is more easily scaled in terms of the number of agents involved in the negotiation process. We provide a description of the algorithm below.

Let Negotiate = TRUE. This allows the algorithm to begin.

Let γ be the local utility adjustment parameter.

Let ϕ be the minimum contribution that can be made to the global utility for negotiation to continue.

Step 1.0 Each agent calculates the utilities for each of the viewpoints accessible by the agent using equation 4.3.

Step 2.0 Order the viewpoints in decreasing order of local utility μ_l.

This information is also stored in the knowledge base. The values of the respective weights are set at runtime.

Step 3.0 Select a camera position from the subset of positions with the highest local utility and broadcast this to the group via the communication mechanism. This represents the agents preference to move to the selected position. Since each agent does this step, all agents would end up with the other agents' preferred camera positions.

Step 4.0 WHILE Negotiate = TRUE

Step 4.1 Update the mental model with the information obtained from the other agents in the group.

Step 4.2 Compare the list of visible features with the feature list from the other agents in the group.

Step 4.3 Adjust the local utility as follows:

$$\mu_l^i(t+1) = \mu_l^i(t) - \gamma \frac{V_{int}}{V_{total}} \tag{4.11}$$

Where, V_{int} is the number of visible features in common with other agents. V_{total} is the total number of visible features and γ is the adjustment parameter.

Step 4.4 Calculate C_i, the contribution to the global utility by agent i's preference, using the following formula:

$$C_i = \sum_j w_j \mu_l^j - \sum_{j \neq i} w_j \mu_l^j \tag{4.12}$$

Step 4.6 Update the seniority score as described above.

Step 4.7 Search the communicated preferences for any conflict as defined above.

Step 4.8 Calculate the probability of conflict as described by equation 4.6

Step 4.9 The value for the probability of conflict, the contribution to the global utility and the other attributes described for the action history are stored in the action history database.

Step 4.10 FOR all preferences in conflict with agent i's preference,

do the following:

The conflict resolution mechanism is activated and the algorithm described in section 4.6.4 is used to resolve the conflict.

END-FOR

At this point the agent searches for a better choice of vantage point based on the now available evidence from the other agents. In order to make the decision as to which vantage point to utilize, the agent adjusts the utilities of all the possible viewpoints within its range of motion by the information available in the mental model and the action history. This is achieved as follows:

Step 4.11 Adjust the local utilities for each preference based on the number of common features visible using equation 4.5 and the probability of conflict computation from equation 4.6.

The agent then chooses the best action based on the adjusted expected utility and calculates the contribution to the global utility. As more information is obtained from other agents in the negotiation process, then the local decisions become more informed.

Step 4.12 IF the contribution to the global utility is less than the threshold ϕ, then

Step 4.13 Negotiation = .F.

Step 4.14 The agent broadcasts an acceptance message and its desire to execute the current preference.

END-IF

END-WHILE

Step 5.0 Once the agents have stopped their negotiation, their preferences can be executed by the action mechanism.

4.8 Theoretical Basis for Agent Behaviour

Upon careful examination of the behavioral characteristics of the agents, we can see that the viewpoint selection process is based on a greedy algorithm approach. The basic idea is to evaluate each viewpoint and select the best possible viewpoint from a local perspective and subsequently from a global perspective using information obtained from other agents within the group. The accuracy of the choices made depends on the maximization of the local utility computed based on information gathered from other agents in the group. Hence the utility calculation takes into account the interdependencies of the agents in the form of redundancy due to overlapping views. The probability of conflict computation takes into account the separation of a camera position from known choices of other agents' camera positions. This information allows a more informed decision to be made since choosing viewpoints that are close to another agent's desired position is in fact increasing the chance of redundant information. Hence the utility adjustment seeks to separate the agents' desired positions as far as possible while maintaining a high level of visibility of the target object. As we shall show in the following chapter, this approach does not necessarily lead to an optimal solution. However, from the experiments undertaken, it is possible for the system to obtain a solution that is functionally acceptable for the vision task at hand.

4.9 Case Based Learning

For any given problem instance, the agents are required to undertake a negotiation process that subsequently leads to some form of a solution. Given the same problem instance, the agents would begin at the same point in the solution space and repeat the same negotiation process, since the initial ordering of the candidate camera positions is the same for that problem. Alternatively, given a similar problem instance where the same scene has been translated or rotated (or both), the agents can utilize prior experience to arrive at an initial estimate of the solution set of candidate positions, and then begin the negotiation process from this point. The hypothesis is that this can improve on the length of time that the negotiation process requires.

The utilization of prior experience by the agents is made possible though a case based learning system. Since we are relying on CAD models of the scenes and not actual images from the cameras to select the appropriate case from the case base, we have to utilize features of the CAD model that allow the agents to perform this selection with an appropriate degree of accuracy. To do so we select a feature set that describes the properties of the scene that are unique to a given scene and can therefore serve as a unique identifier for that scene. The feature set chosen consists of three components as described below.

1. General Case Features

   Case ID: Each case description is given a unique identifier.

   Centroid Name: Each object in a scene has a centre of mass (centre of gravity). This is given a unique name in the case base.

   Num Vertex: The number of vertices of each object in the scene.

   Num Face: The number of faces of each object in the scene. For curved surface approximations we utilize the triangulation of the curved surface to represent the number of faces.

   Surface Area: The computed surface area of each object in the scene.

2. Centroid Separation

   Centroid Separation: A Euclidean distance measure of the spatial separation in 3D space of each centroid in the scene relative to every other centroid in the scene.

3. Prior Solution Information

   Camera Viewpoint: The viewpoint chosen by the agent.

   Centroid Name: Name of the centroid of each object in the scene.

Centroid Position: The coordinates of the centroid of each object in the scene with respect to the world coordinate system.

Angle: The angle between the rays projected from the origin of the world coordinate system to the camera viewpoint and from the origin to the centroid.

Distance: The distance between the camera viewpoint and the centroid.

Each agent maintains its own case base in keeping with the totally decentralized methodology used throughout. For any given problem scene. each agent can refer to its case base to verify whether or not the current scene matches any known scene. This is achieved by comparing the general case features and the centroid separation data of the current scene with those of the stored scenes. Only an exact match is considered. At this stage. the features are orientation invariant. That is to say, regardless of the orientation of the current scene. the features remain the same. Hence if the same scene is presented in a different orientation, the agents will be able to choose the correct scene from their case bases.

At this point it is possible for an agent to have several matching entries in its case base if the same scene has been encountered in different orientations. The refinement to this set is accomplished by choosing the stored scene that is most similar to the current scene using the centroid position information of the prior solution information as described above. The stored scene whose centroid positions are closest to the current scene is chosen. We use a simple Euclidean distance to facilitate this choice.

Once the most similar scene has been established. the agents need to make an initial guess or estimate of the initial camera positions for the current scene. The scene selected from the case base has an associated solution or camera position for the agent. By using the angle and distance of this camera position relative to the centroids in the stored scene, the agent can choose a set of camera positions in the current scene such that the angle and distance measures relative to the centroids of the current scene are either similar or equal to that of the stored scene. These measures represent the spatial arrangement of the final camera position relative to that of the objects in the stored scene. The initial estimate of the camera position in the current scene is based on choosing a viewpoint (or set of viewpoints) that would provide the same or similar spatial relationship. If more than one viewpoint can be chosen, then the agent randomly chooses a viewpoint from this set. Each agent performs this activity to provide its own initial estimate of camera position.

Since each agent maintains its own case base, the agents only need to store information relevant to itself. Hence, it is possible for an agent to be involved in a negotiation process where every other agent has prior knowledge of the given scene, however, the new agent does not. This can occur for example if an additional camera is added to the system after a solution has been agreed upon by the existing agents. The system must reformulate its solution to include the additional resource. In this case, the new agent will evaluate its candidate positions without any initial position estimate and start the negotiation process there. At the end of the process, each agent updates its case base. Hence learning is automatic. The new agent will now have knowledge of the new scene. However, it is also possible for the user to provide the agents with a pre-computed case base in order to improve the efficiency of the system for specific problem sets.

The utilization of the case base structure enables the agents not only to utilize prior knowledge to some advantage. but also it provides a means of preventing the agents from starting the search at the same point in the search space. a situation which invariably results in conflict. The initial computation of utility values can therefore be carried out based on the initial unique positions of the agents.

4.10 Summary

This chapter has presented a description of the agent model. the algorithms and structures used to achieve the planning of sensors in a stationary modeled environment. The chapter also describes the cased based learning system that allows the agents to adapt their negotiation process based on learned information. In the following chapter. we present the empirical results that justify the feasibility of the system and the corresponding analysis of these results.

Chapter 5

Experimental Results

5.1 Introduction

The primary objective of this thesis is to present the design of a framework for the autonomous coordination of a distributed system of mobile sensors. In this chapter we present the experimental results of such a system based on the model and theoretical foundations presented in the previous chapters. We also provide an analysis of these results in an effort to explain the underlying behavioral characteristics of the system. Although the feasibility of the system has been demonstrated using several problem sets, we present the results obtained for three very different sensor planning tasks that are characteristic of the variation in the problem sets used for empirical evaluation. In each case, the objective is to deploy the sensors in a positional configuration such that their combined perspectives provide maximal coverage of one or more target objects in the scene. Finally we compare the results produced by the system with the optimal camera arrangement obtained from an

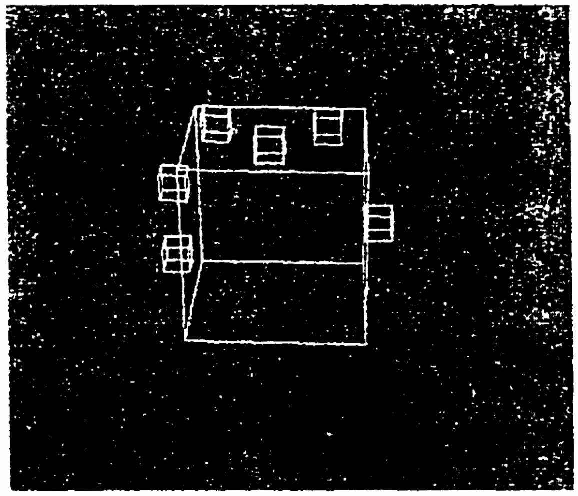


Figure 5.1: CAD Model of Target Object

exhaustive method. This allows us to obtain a measure of the performance of the agent based system.

5.2 Single Target Coverage

The first sensor planning task is to generate the camera positions necessary for the coverage of a single target partially occluded by other objects in a scene. Figure 5.1 shows a CAD model of the target and figure 5.2 shows the rendered target object from two different perspectives. This object is included in the scene as depicted in figure 5.3. As is apparent from the scene, the object is occluded from several vantage points. Hence, in order to completely view the surface of the target, we are forced to employ multiple cameras.

The range of camera motion can be represented by a bounding polyhedron

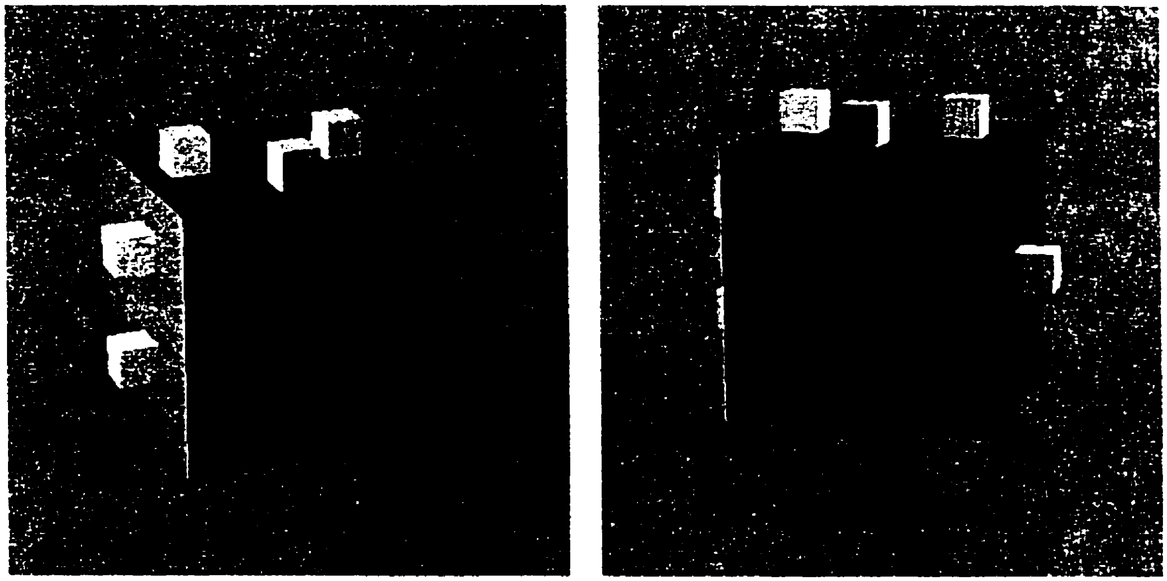


Figure 5.2: Rendered Model of Target Object

whose volume is subdivided into discrete voxels. The centroids of these voxels form the possible camera positions. We utilize a single bounding polyhedron for all cameras since this gives the maximum intersection of possible ranges of motion for the cameras and correspondingly, maximizes the interdependencies amongst the agents. Figure 5.4 illustrates the bounding polyhedron for the scene.

To illustrate the level of occlusion of the object. we refer to figure 5.5 which shows the percentage of the target object that is visible from the set of possible camera positions. The graph includes only those camera positions where some portion of the target object is visible. The graph was obtained by finding the ratio of the number of vertices visible to the total number of vertices of the target object expressed as a percentage. The orientation of the camera from each of the vantage

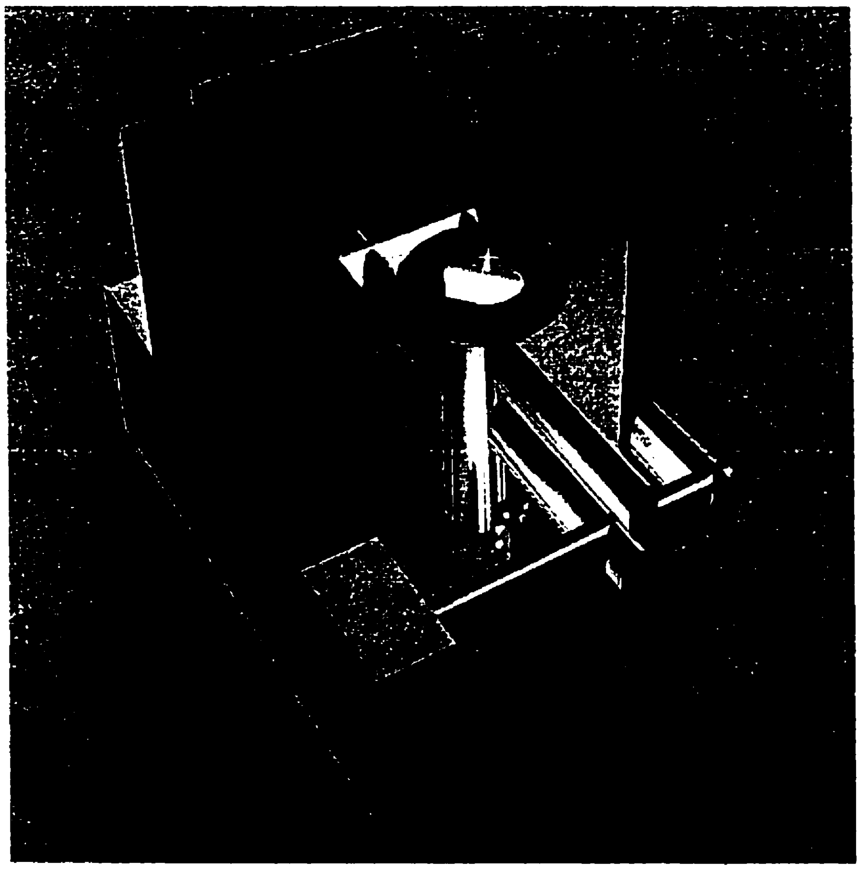


Figure 5.3: Rendered CAD Model of Scene

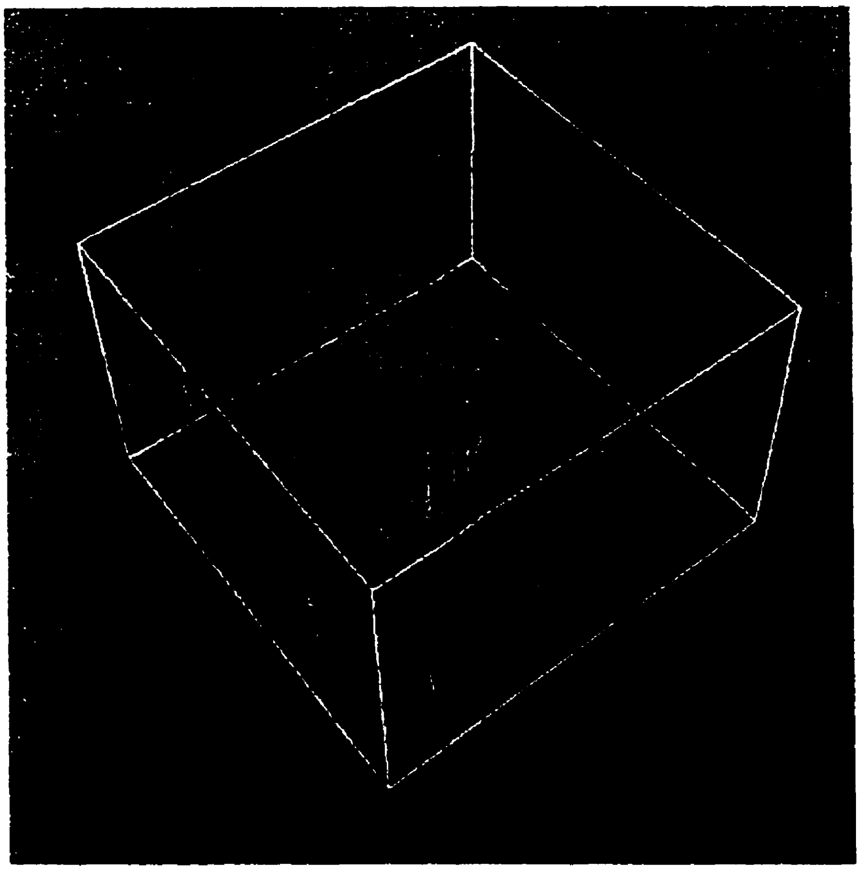


Figure 5.4: Bounding Polyhedron for Camera Range of Motion

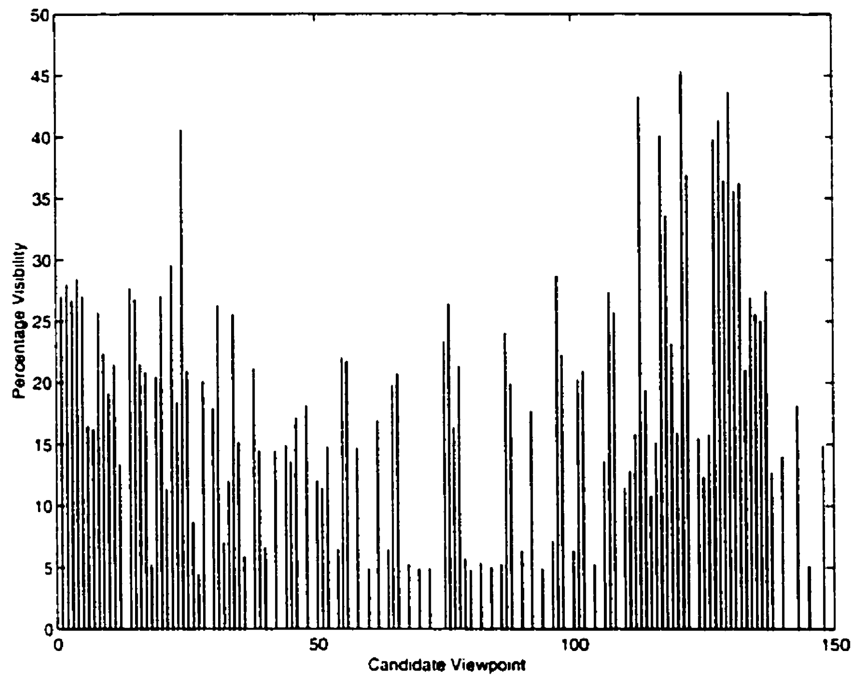


Figure 5.5: Target Visibility from Various Viewpoints

points is assumed to be towards the centroid or center of gravity of the target object. The vantage points from which the object is totally occluded by other objects in the scene are not shown. From this graph, we can see that if we were to position a camera at the best viewing position with the corresponding orientation towards the centroid of the target, we would be able to view about 45 percent of the target vertices.

We employ two cameras for observing the target object in the scene. Figure 5.6 illustrates the initial positions of the two cameras relative to the scene. The orientation of the cameras is fixed to the centroid of the target object.

From the initial views obtained from both cameras, we see that the target object

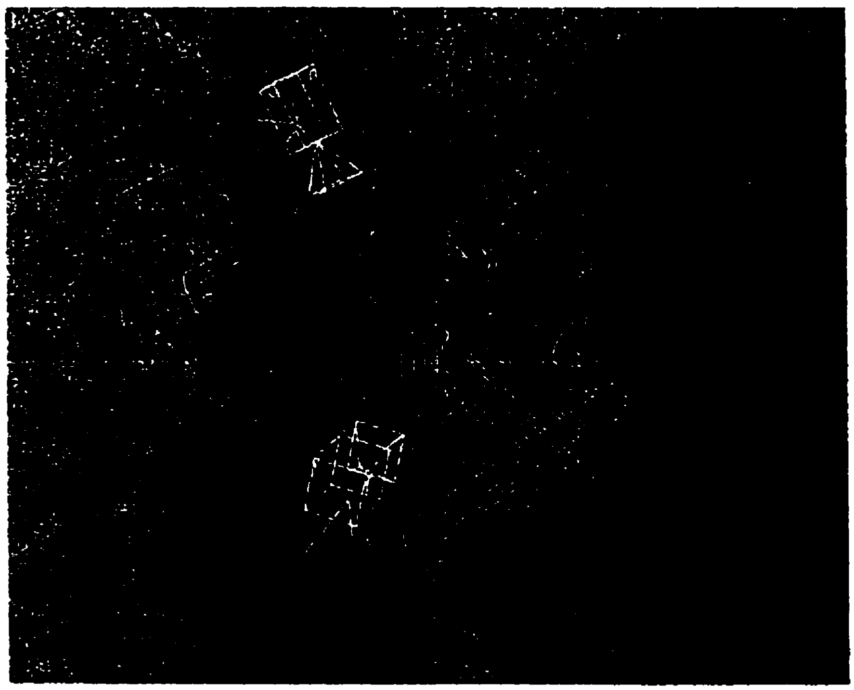


Figure 5.6: CAD Model of Scene with Initial Camera Positions

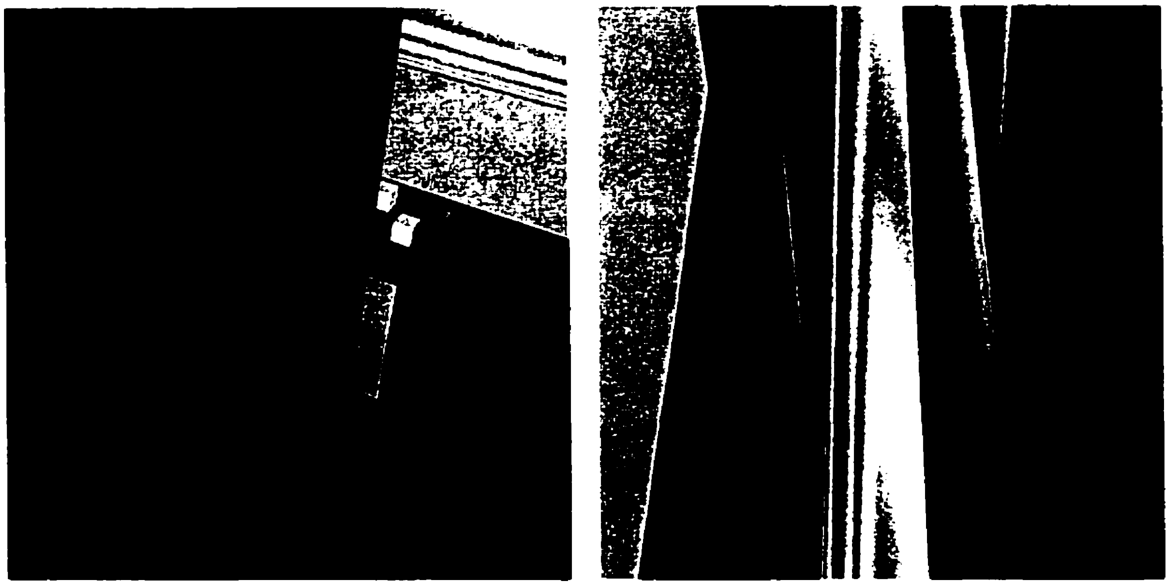


Figure 5.7: Initial Camera Views

is still occluded by objects within the scene. This fact is illustrated in figure 5.7. The objective therefore is to find suitable positions for the cameras such that the target object is least occluded by other objects within the scene. At the same time the system tries to minimize the redundancy in the views of the object by positioning the cameras such that the resulting intersection between the resulting field of views is minimized.

Figure 5.8 represents the views of the target obtained after the agents have autonomously positioned the cameras. It is apparent that there is a significant improvement in the amount of the surface of the target object target that is now visible. In addition, we can see that there is not a significant amount of redundancy in the resulting views. Figure 5.9 shows the corresponding positions of the cameras

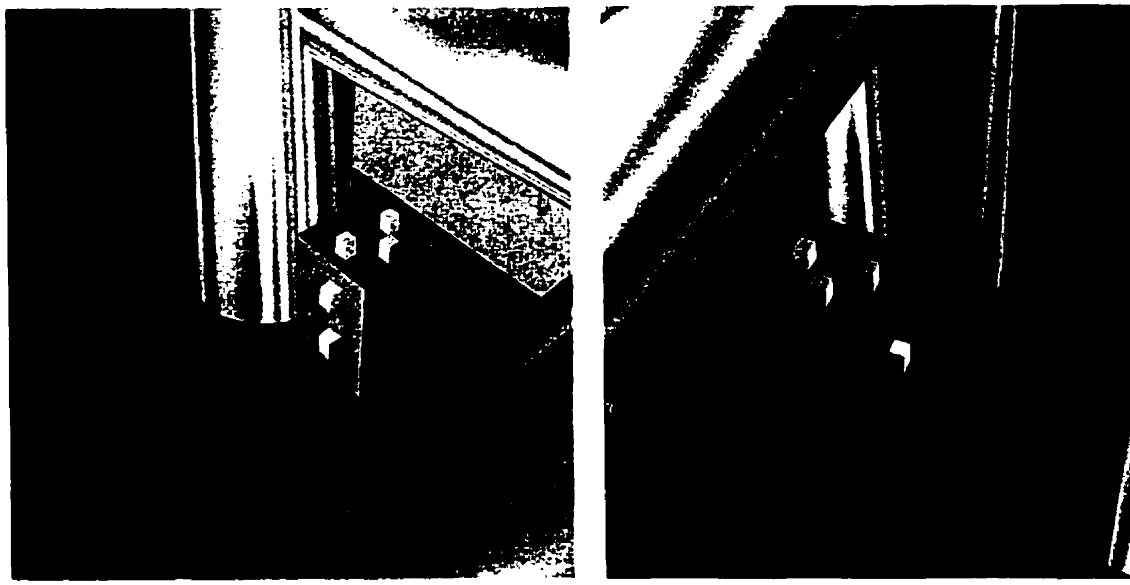
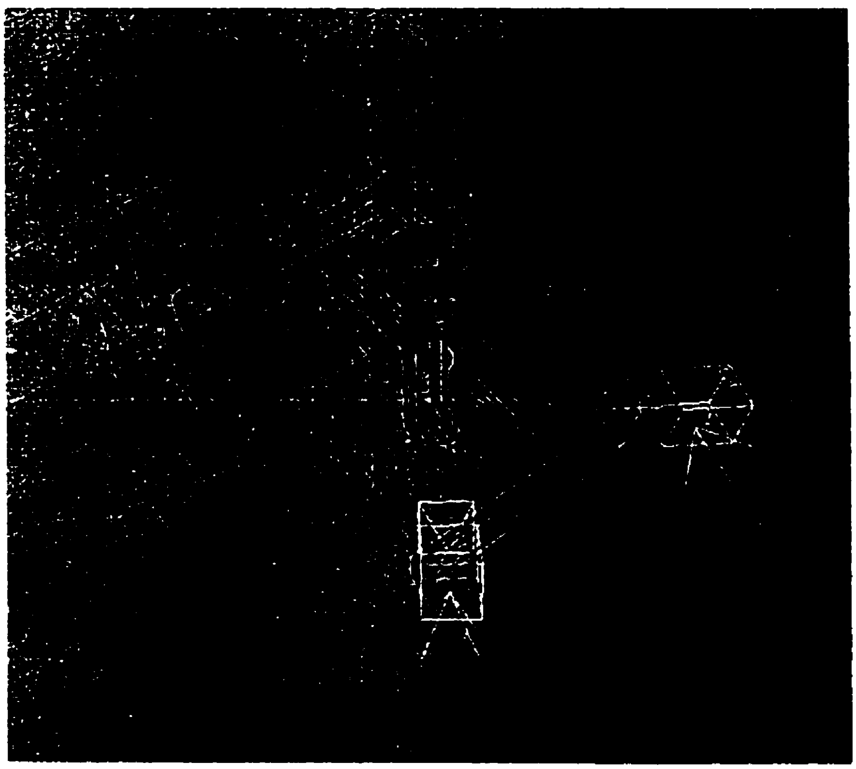


Figure 5.8: Final Camera Views

relative to the scene.

Tables 5.1 and 5.2 show the initial and final positions of the cameras respectively for the two camera system. In addition, table 5.2 shows the utility of the positions chosen by the agents. In order to assess the quality of the solution obtained, we utilize an exhaustive search method to arrive at the optimal viewpoints. Table 5.3 shows the first 16 camera position combinations along with their corresponding utility values obtained from the exhaustive search sorted in descending order of the utility value. The exhaustive search method required 11 minutes and 10 seconds to complete the search using only two cameras. Using the agent method, required only 2 minutes and 14 seconds for completion on the same computing platform. The solution obtained by the agent method corresponds to the solution



c

Figure 5.9: CAD Model of Scene Showing Final Camera Positions

Agent	X	Y	Z
1	1.2774	-1.9013	1.7271
2	-2.1734	0.0462	2.2083

Table 5.1: Initial Positions of Cameras

Agent	X	Y	Z	Utility
1	2.2064	-1.6976	1.7271	75.69
2	2.2064	0.8844	1.7271	33.80

Table 5.2: Final Camera Positions

number 7 in table 5.3.

The solid curve of the graph in figure 5.10 shows the behaviour of the global utility value for each of the camera position combinations considered by the exhaustive search. Note that only the distinct utility values have been included in the graph. The solid vertical line indicates the relative position of the solution obtained from the agent method to that of the solutions obtained by the exhaustive search. Although the agent method did not yield the optimal solution, it has succeeded in eliminating most of the other possible solutions that yield a smaller global utility value. The resulting solution by the agent method is thus functionally acceptable for the given vision task.

As a visual comparison of the quality of the results obtained form the exhaustive search, we show the views obtained from the optimal camera positions in figure 5.11. We also illustrate the views obtained from the camera positions that yield a global utility that is less than that obtained by the agent method. The sub-optimal set of camera positions chosen yielded a global utility of approximately 98. The

Solution	X1	Y1	Z1	X2	Y2	Z2	Utility
1	1.3457	0.8844	1.2265	2.2064	-1.6976	1.7271	114.73
2	2.2064	1.7451	2.2277	2.2064	-1.6976	2.2277	112.73
3	2.2064	1.7451	0.7258	2.2064	-1.6976	1.7271	111.80
4	1.3457	0.8844	1.2265	2.2064	-1.6976	2.2277	111.34
5	2.2064	1.7451	0.7258	2.2064	-1.6976	2.2277	110.42
6	2.2064	1.7451	2.2277	1.3457	-0.8369	1.2265	109.95
7	2.2064	0.8844	1.7271	2.2064	-1.6976	1.7271	109.49
8	2.2064	1.7451	2.2277	1.3457	-1.6976	1.7271	109.25
9	2.2064	1.7451	2.2277	1.3457	-1.6976	2.2277	109.25
10	2.2064	1.7451	2.2277	2.2064	-2.5583	2.2277	109.25
11	2.2064	1.7451	2.2277	2.2064	-2.5583	2.7284	109.25
12	2.2064	1.7451	2.2277	2.2064	-2.5583	3.2289	109.25
13	1.3457	0.8844	1.2265	1.3457	-0.8369	1.2265	108.56
14	2.2064	0.8844	2.7284	2.2064	-1.6976	2.2277	108.10
15	1.3457	0.8844	1.2265	1.3457	-1.6976	1.7271	107.87
16	1.3457	0.8844	1.2265	1.3457	-1.6976	2.2277	107.87

Table 5.3: Best Results from the Exhaustive Search

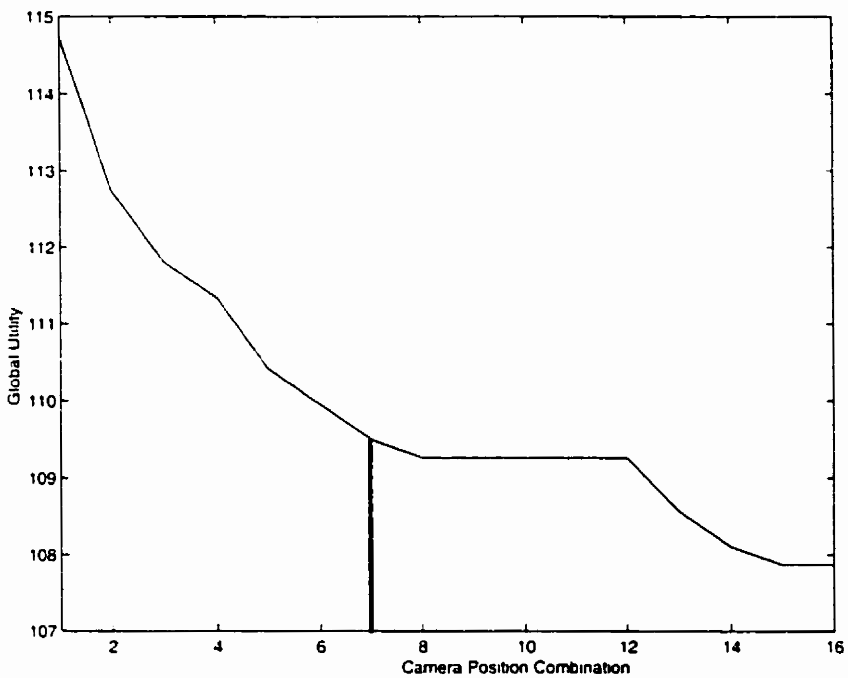


Figure 5.10: Utility Values of the Best Results from Exhaustive Search

Figure 5.11: Camera Views for the Optimal camera Positions

views obtained are shown in figure 5.12.

Figures 5.11 and 5.12 visually illustrate the difference in the quality of the solution. In figure 5.12 we see that the total visibility of the target is less than that shown in figure 5.8 the views obtained from the agents and 5.11 the optimal views produced by the exhaustive search method. Such a comparison illustrates that the agents are capable of eliminating most of the sub-optimal viewpoints during their negotiations.

5.2.1 Increasing the Number of Cameras

In this section. we show the results of adding another camera to the system. Since the agents are based on the same model and coordination algorithm. the incorpo-

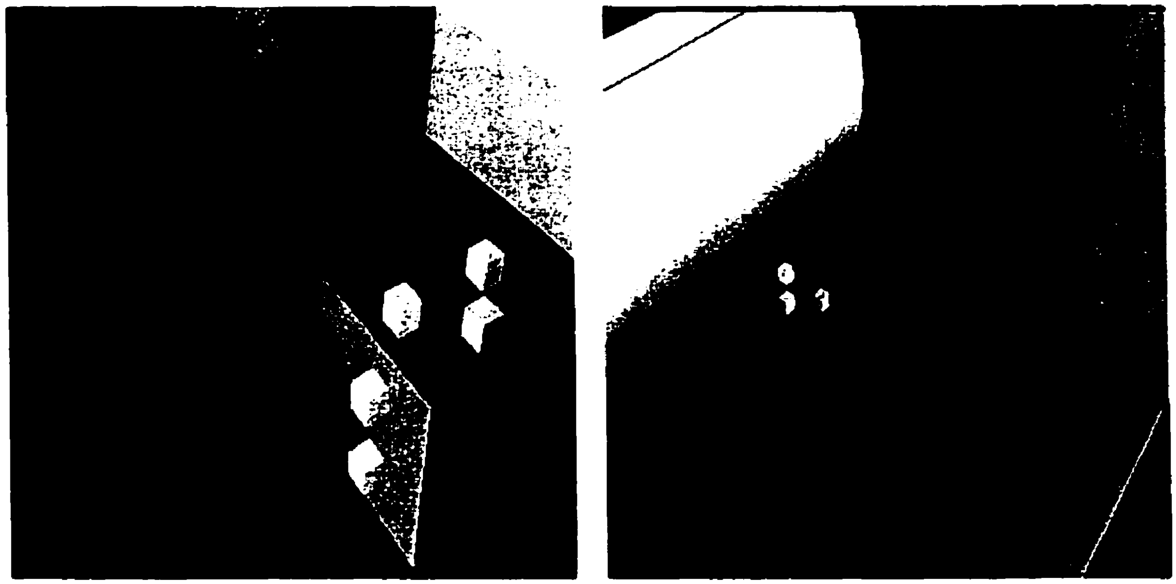


Figure 5.12: Camera Views for Sub-optimal Positions

ration of additional cameras is a trivial process. We simply duplicate the agent algorithms for each additional camera and update the agent's sensor model for the new camera. However, adding more cameras to the system does not guarantee that the amount of information obtained about the target will increase. This largely depends on number, size and orientation of the target object. For smaller single target tasks, two cameras may be sufficient to cover the surface of the target, while in other situations more cameras are required.

To illustrate this point, we refer to figure 5.13 which represents the arrangement of the 3 cameras after the conclusion of the agents' negotiation process. Their corresponding positions and utility values are shown in table 5.4.

The utility values shown in table 5.4 indicate that there is no significant im-

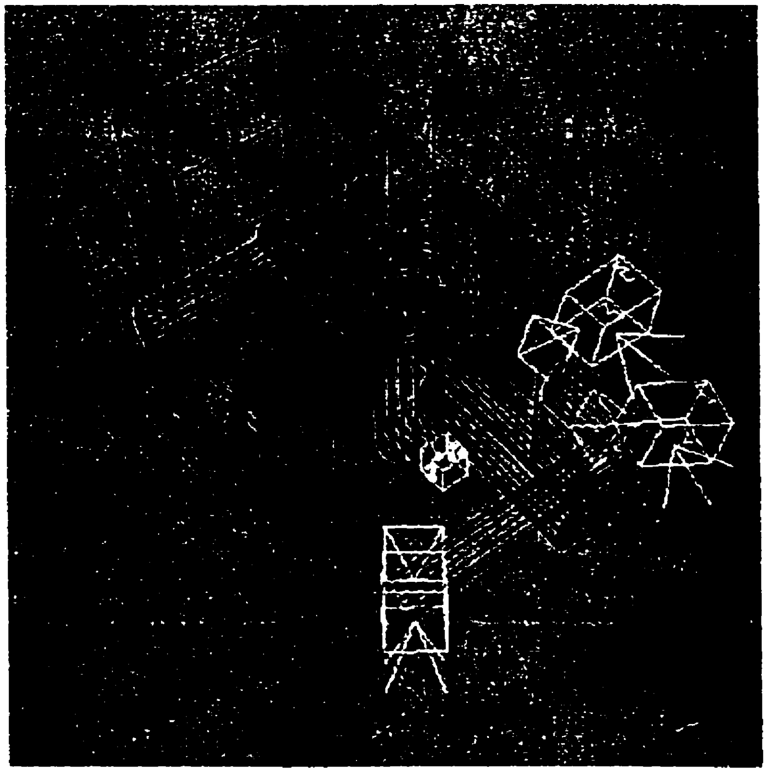


Figure 5.13: Final Positions using 3 Cameras

Agent	X	Y	Z	Utility
1	2.2064	-1.6976	1.7271	75.69
2	2.2064	0.8844	1.7271	18.15
3	1.3457	0.8844	2.2277	16.85

Table 5.4: Final Camera Positions

provement in the global utility as a result of adding a third camera. The global utility obtained was 110.69 as compared to the global utility obtained for the two camera system, which was 109.49. This fact is apparent in the views obtained from the three cameras shown in figure 5.14. The utilities of agents 2 and 3 were reduced as a result of the fact that the fields of view of their corresponding cameras are overlapping. Since we are inspecting a single target, most of the surface of the target can be covered with two cameras. The need for more than two cameras becomes more apparent when the target object is occluded by several objects, the target is of a large size or alternatively when there are multiple targets to be covered that are spatially separated. This situation is explored in the following section.

5.3 Multi-Target Coverage

The system may also be used to obtain information about multiple targets that may be partially or totally occluded by one or more objects in a given scene. The agent method is especially suited for multiple target problems since various numbers of cameras can be deployed to cover each target. Hence, this example serves to illustrate the advantage of this method over the single camera systems previously discussed.

The scene under consideration is shown as a CAD model in figure 5.15 with the targets rendered as solids for the sake of clarity. The objective is to deploy a set of cameras such that both targets are simultaneously visible and covered. It is not possible to utilize a single camera in this situation due to the occlusion and relative spatial location of the target objects. The rendered CAD model of the scene shown

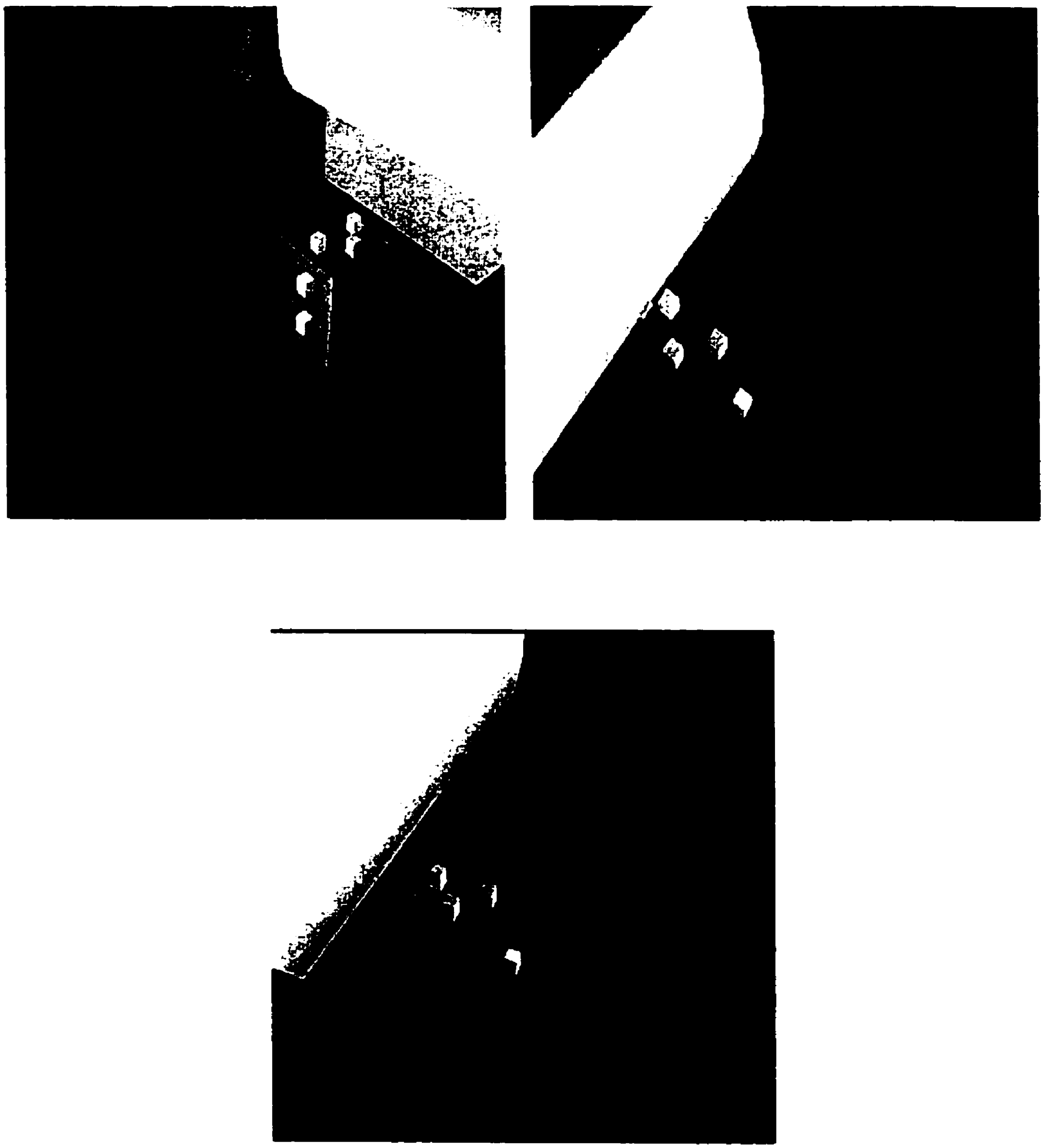


Figure 5.14: Final Views for 3 Cameras

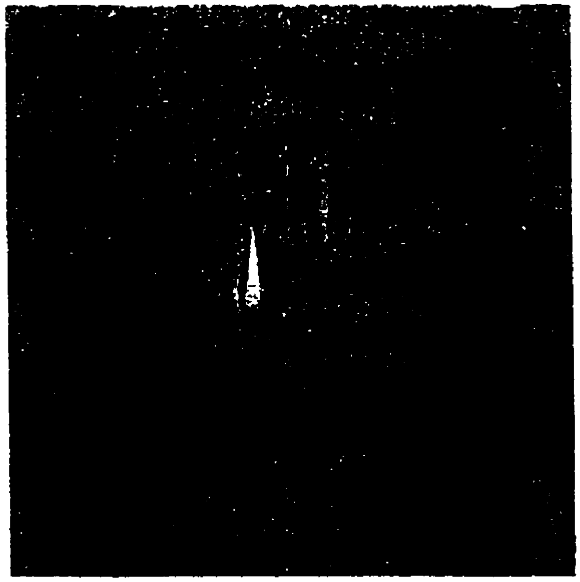


Figure 5.15: Multi-Target CAD model

in figure 5.16 clearly illustrates the occlusion of the rear target.

The graph depicted in figure 5.17 quantizes the total visibility of both targets from each of the candidate viewpoints. To compute the percentage of target visibility in this situation, we utilized the total number of distinct vertices visible from both targets from any candidate viewpoint as a percentage of the total number of distinct vertices of both targets. According to figure 5.17 we can view a maximum of 30% of both targets simultaneously from any one candidate viewpoint.

In order to view both targets simultaneously, we initially deploy two cameras, each controlled by an agent. The agents can choose the orientation of the camera based on the centre of mass of each of the target objects in the scene. Hence, any choice of viewing position also includes the centre of mass of the target object

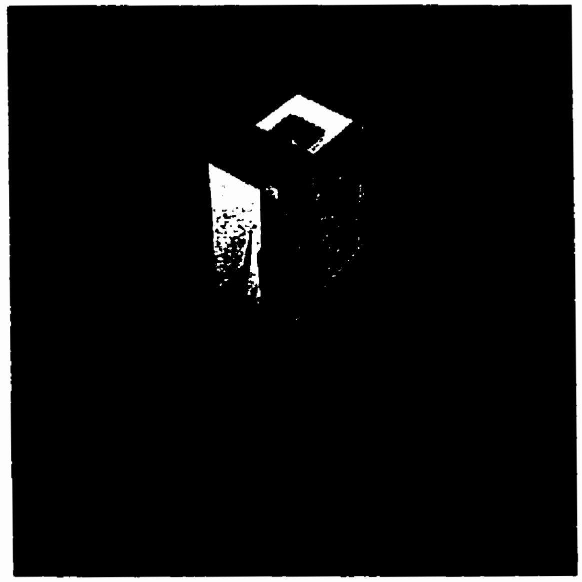


Figure 5.16: Multi-Target Rendered CAD model

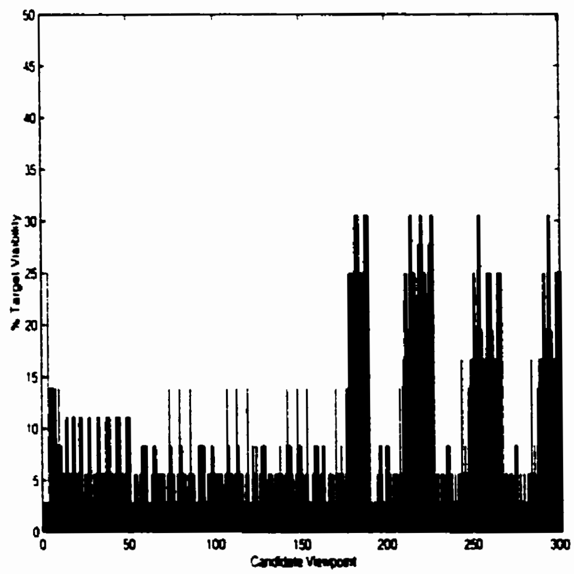


Figure 5.17: Total Target Visibility per Candidate Viewpoint

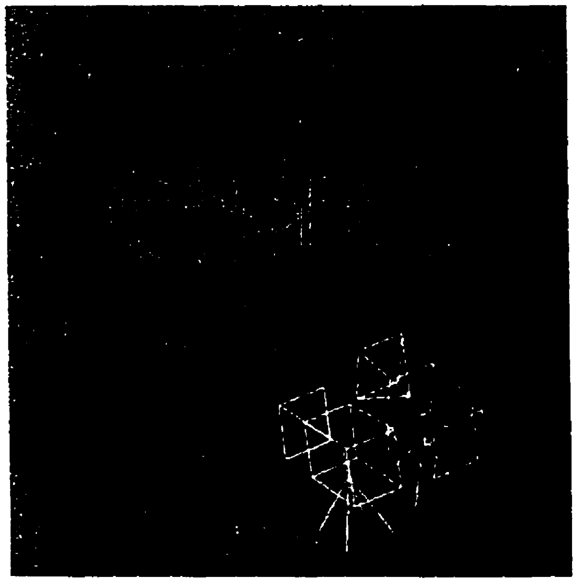


Figure 5.18: Initial Camera Positions

being observed by the agent. Figure 5.18 shows the randomly chosen initial camera positions relative to the CAD model of the scene. Figure 5.19 shows the views obtained by the cameras from their initial positions. The figures indicate that from the initial vantage points, the targets are either totally or partially occluded by other objects within the scene.

Figure 5.20 shows the final camera positions relative to the scene as agreed upon by the agents. We note here that the agents have decided to cover different targets in order to maximize the coverage of the set of targets and correspondingly, maximize the global utility measure. Figure 5.21 shows the actual views of the target objects obtained from each of the cameras.

The final positions of the cameras and their respective utility values are listed

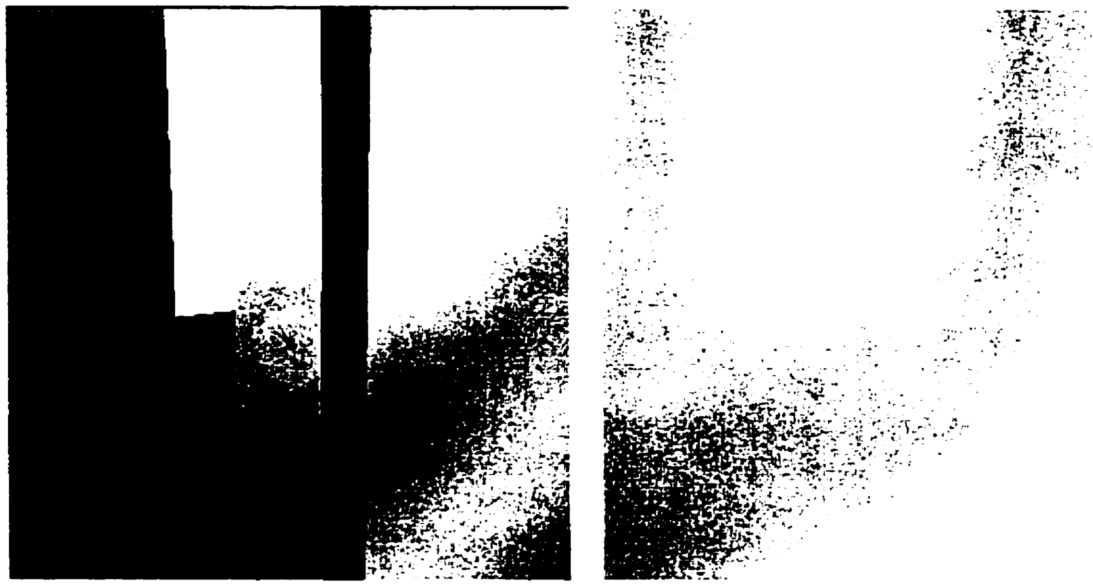


Figure 5.19: Initial Camera Views

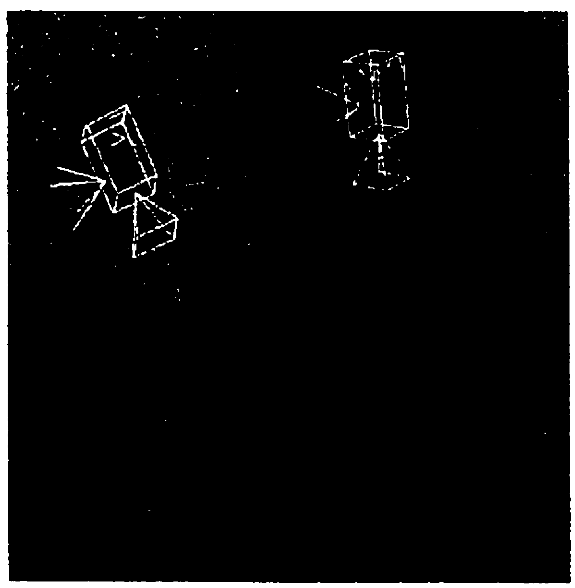


Figure 5.20: Final Camera Positions

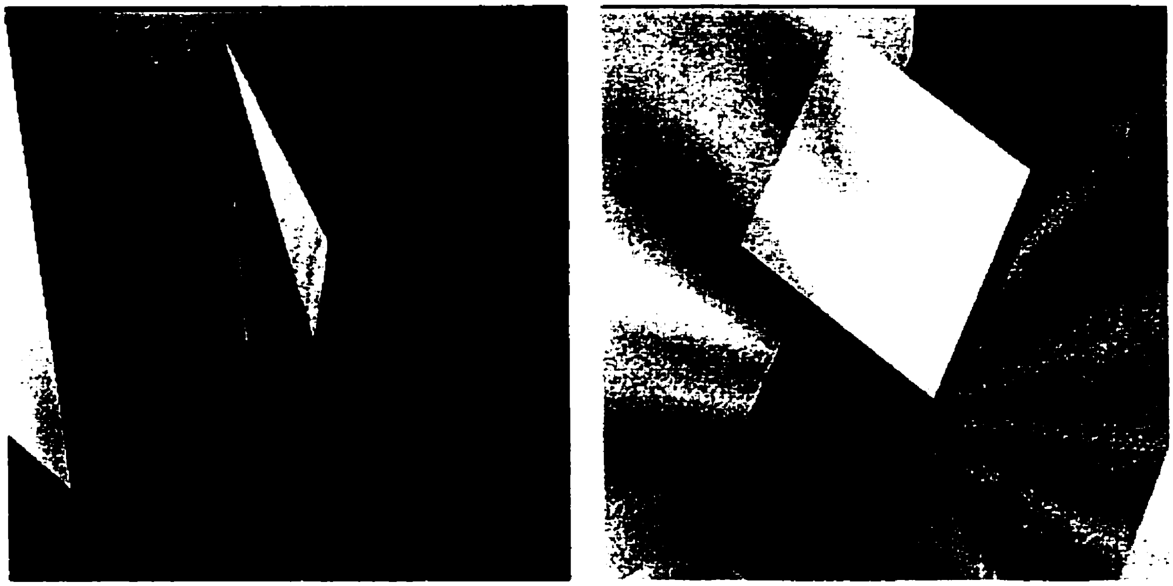


Figure 5.21: Final Camera Views

in table 5.5. In order to verify the correctness of the solution, we again utilized an exhaustive search algorithm to find the pair of viewpoints yielding the maximum utility. Table 5.6 shows that there are indeed 11 such possible solutions. We should also note that the solution obtained by the agents is in row 1 of the table. The agent method was significantly more efficient however, since the running time of the agent method was 2 minutes and 10 seconds while the running time of the exhaustive search method was 16 minutes and 42 seconds on the same computing platform. The results obtained from the exhaustive search show that the position of camera 2 given by the coordinates X2,Y2,Z2 remain the same for all the possible solutions yielding the highest utility level. In addition, the positions of camera 1, given by the coordinates X1,Y1,Z1 are symmetrical about the Y axis and differing

Agent	X	Y	Z	Utility
1	-0.6999	-0.20000	2.53147	22.92
2	2.20779	-0.25817	2.24387	31.25

Table 5.5: Final Camera Positions Using 2 Cameras

Agent 1			Agent 2		
X1	Y1	Z1	X2	Y2	Z2
-0.6999	-0.20000	2.53147	2.20779	-0.25817	2.24387
-0.6999	-0.20000	2.69915	2.20779	-0.25817	2.24387
-0.6999	-0.20000	2.86682	2.20779	-0.25817	2.24387
-0.6999	-0.20000	3.03450	2.20779	-0.25817	2.24387
-0.6999	-0.20000	3.20217	2.20779	-0.25817	2.24387
-0.6999	0.20000	2.36380	2.20779	-0.25817	2.24387
-0.6999	0.20000	2.53147	2.20779	-0.25817	2.24387
-0.6999	0.20000	2.69915	2.20779	-0.25817	2.24387
-0.6999	0.20000	2.86682	2.20779	-0.25817	2.24387
-0.6999	0.20000	3.03450	2.20779	-0.25817	2.24387
-0.6999	0.20000	3.20217	2.20779	-0.25817	2.24387

Table 5.6: Final Camera Positions from Exhaustive Search

only by the position of the Z coordinate of the camera.

In this situation, the algorithm resulted in a solution that is within the optimal set of possible solutions. The experiment shows that the agents can produce an optimal solution. However, due to the nature of greedy algorithms, this cannot be guaranteed for all cases. Hence we still focus on achieving a functionally accurate solution that is acceptable for the given sensing task.

5.3.1 Agent Communication

Table 5.7 shows the non-repeated messages sent by each agent during the communication process. An agent may repeat a message if there is no action or information available that changes its current mental model of the environment. The message type and content are therefore the same as the previous message. Such repeated messages have been removed from table 5.7. The line number column is for reference only. The *Type* column shows the type of message sent and the *From* column indicates the sending agent. The *Message Data* column illustrates only the camera position and corresponding utility for the sake of brevity. The format of the message data as shown is camera position (X,Y,Z), utility.

The agents initially chose the same position due to the fact that they try to selfishly maximize the utility. This is indicated by lines 1 and 2 of the table. The agents need to prioritize their decisions and in this case, they do so by randomly selecting their priority. This is done by requesting random numbers as shown in lines 3 and 5. The response to these requests are shown in lines 6 and 8. The initial utility values of the agents are adjusted to zero since they are in conflict. These adjusted utility values are also broadcast as illustrated in lines 4 and 7.

From the priority response, we can see that agent 1 is the agent with the highest priority. Hence, agent 2 chooses a different position. The position is chosen so as to maximize the utility and the distance away from the conflict position which is currently occupied by agent 1. The position change is broadcast in line 10. Agent 1 then responds with a reevaluation of its current position which now takes into consideration the position chosen by agent 2. The resulting utility is shown in line

11. Agent 1 also reevaluates its other possible positions at that time.

Line 12 shows that agent 2 is prepared to make its current position its final position. However, agent 1 disagrees with a NAK in line 13. This would only occur if agent 1 has found a position from which it can increase its current contribution to the global utility. Agent 2 re-asserts its position in line 14. Lines 15 and 16 show the new position found by agent1. Lines 18 through 21 result from the proposal and acceptance of these positions as final positions and line 22 shows the termination of negotiations by agent 2.

5.3.2 Increasing the number of Cameras

As an illustration of the scalability of the system, we incorporate a third camera. As previously mentioned, the scalability of the system is one of its main advantages. Hence adding another camera is a relatively trivial process. Figure 5.22 illustrates the deployment of three cameras on the same scene. From the relative positions of the cameras, it is apparent that the third camera has been positioned so that the second target (the rectangular post) is more visible in terms of the number of its constituent facets that are within the field of views of the cameras. As a result, the camera that was initially viewing this target has adjusted its position in order to accommodate the third camera. This becomes apparent if the position of chosen by agent 1 in table 5.8 is compared to that of camera 1 in the 2 camera case as listed in table 5.5 In the latter case, camera 1 had position (-0.69999,-0.20000,2.53147). This adjustment serves to limit the amount of redundancy between the two proximal camera views.

Line No.	Type	From	Message Data
1	FVL	agent2	0.08407000.-1.00000000,2.69915300,31.9444444
2	FVL	agent1	0.08407000.-1.00000000,2.69915300,31.9444444
3	RNR	agent1	R
4	ALU	agent2	0.08407000 -1.00000000.2.69915300.0.00000000
5	RNR	agent2	R
6	RND	agent2	1.25487681
7	ALU	agent1	0.08407000 -1.00000000,2.69915300,0.00000000
8	RND	agent1	3.78698574
9	FVL	agent2	0.08407000 -1.00000000.2.69915300.0.00000000
10	FVL	agent2	-0.69990600.0.20000000,2.36380200,22.9166667
11	ALU	agent1	0.08407000.-1.00000000,2.69915300,29.0694440
12	FP	agent2	-0.69990600.0.20000000,2.36380200,22.9166667
13	NAK	agent1	NAK
14	FVL	agent2	-0.69990600.0.20000000,2.36380200,22.9166667
15	FVL	agent1	2.20779100.-0.25817100,2.24387700.31.9444444
16	ALU	agent1	2.20779100.-0.25817100,2.24387700,31.2500000
17	FVL	agent2	-0.69990600.-0.20000000.2.53147700,22.916667
18	FP	agent1	2.20779100.-0.25817100,2.24387700,31.2500000
19	FP	agent2	-0.69990600.-0.20000000,2.53147700.22.916667
20	ACK	agent2	ACK
21	ACK	agent1	ACK
22	TERM	agent2	TERM

Table 5.7: Trace of Agent Communication

Agent	X	Y	Z	Utility
1	-0.6999	0.20000	2.36380	22.92
2	2.20779	-0.25817	2.24387	31.25
3	-0.6999	-1.0000	2.69915	35.41

Table 5.8: Final Camera Positions Using 3 Cameras

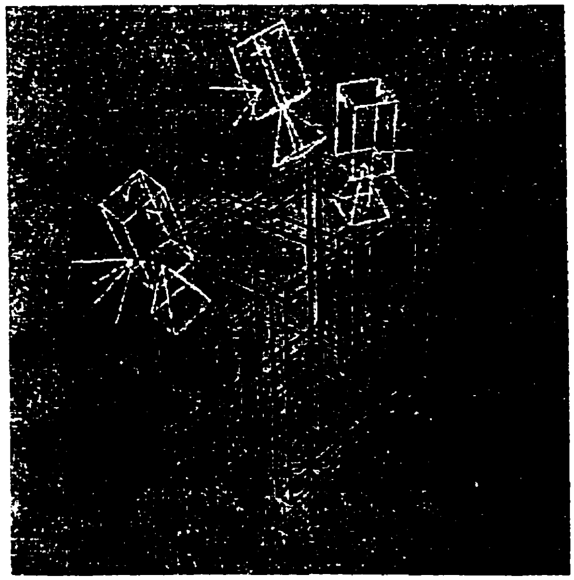


Figure 5.22: Final Positions for 3 Cameras

Figure 5.23 illustrates the resulting views of the three cameras. From the views we can see more of the surface of the second target as compared with the previous system that used only 2 cameras. From a quantitative perspective, we can represent the effect of adding another camera to the system by examining the number of distinct vertices of the set of targets that are visible in the union of all the camera fields of view. The graph of figure 5.24 shows the percentage visibility of the target vertices for one, two and three cameras. In this situation, we see that the deployment of three cameras yields more coverage of the surfaces of the target objects. This is due to the fact that initially using two cameras only allocated one camera per target. However, more of the surface of each target can be seen if more than one camera covered any of the targets. Hence the rise in visibility when a third camera was added.

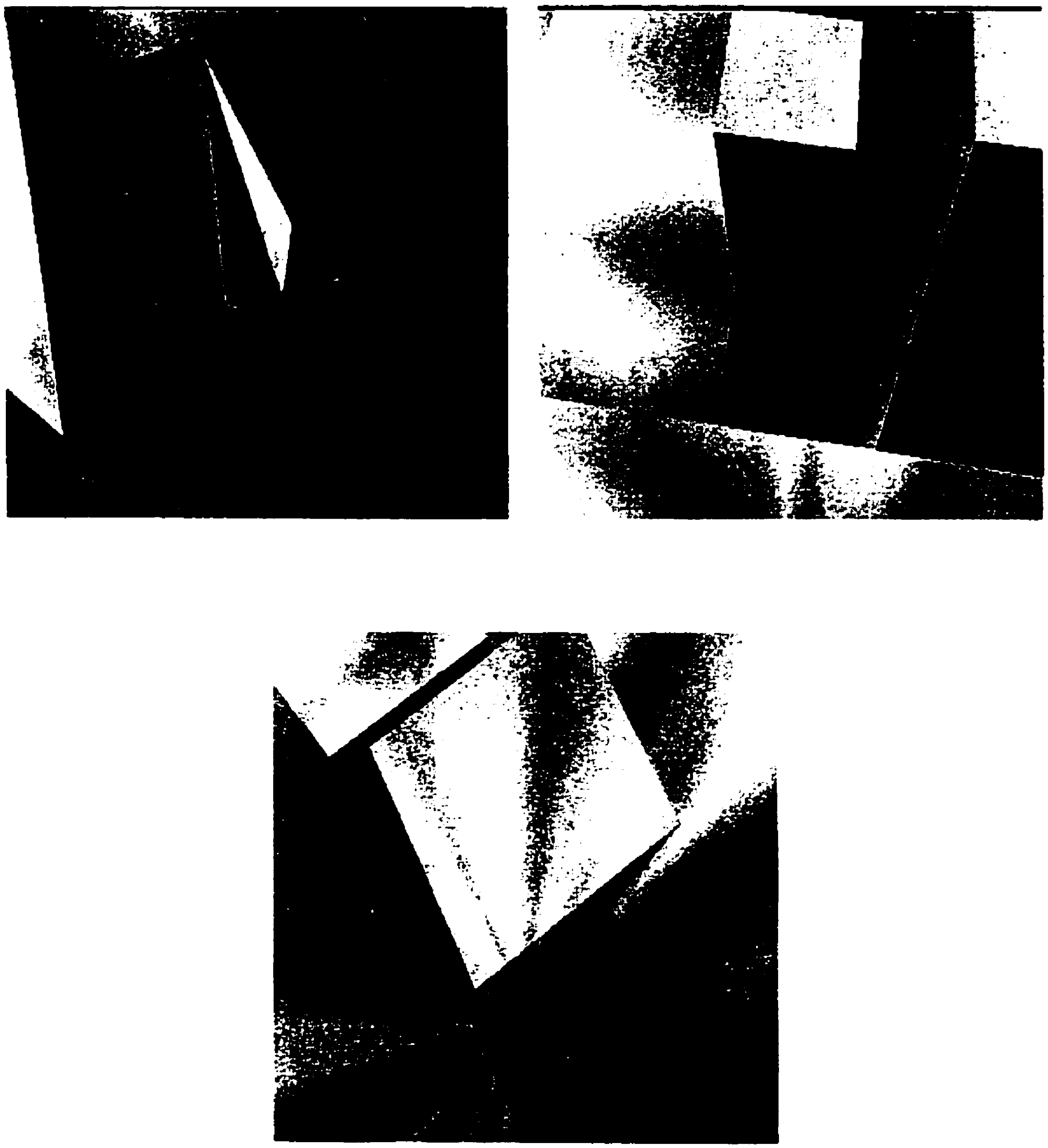


Figure 5.23: Final Views for 3 Cameras

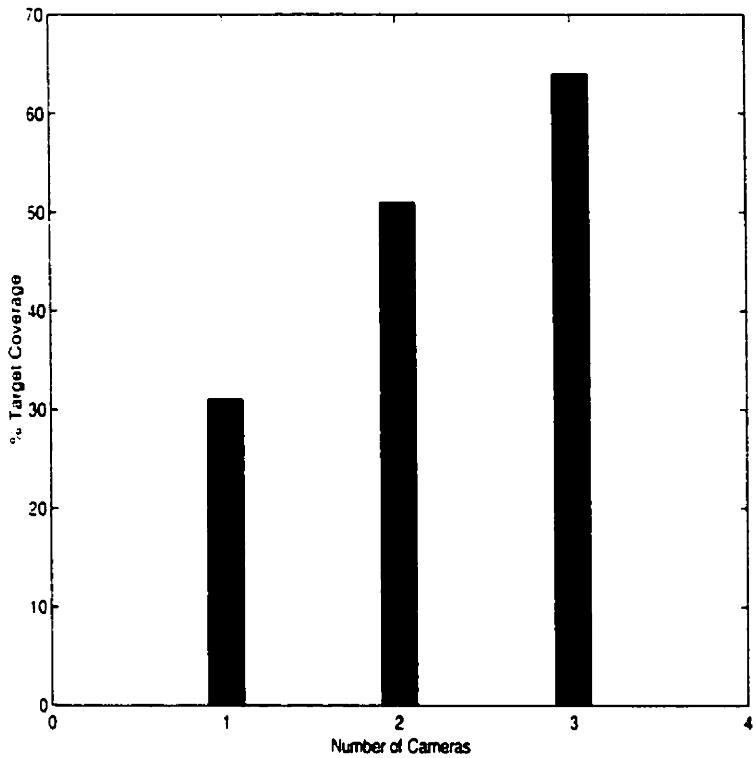


Figure 5.24: Coverage of Distinct Target Vertices

5.4 Learning to Improve Efficiency

5.4.1 The Case Based Reasoning System

The agent model previously presented incorporates a learning system based on the cased based reasoning approach. In this section, we illustrate the effects of learning on the time taken for the agents to arrive at a particular solution. Recall that the main objective of the learning system is to facilitate a shorter negotiation process by influencing the initial decisions of the agents. Normally, the initial decision of any agent is arrived at prior to the receipt of any communication from the other agents in the group. Hence initial decisions are based on the selfish desire to maximize the local utility. Due to the inherent interdependencies of the agent interactions, they invariably result in conflicts that must be resolved. By incorporating a learning system, the initial decisions of the agents can be made more informed and could therefore lead to shorter paths to the correct solution or at the very least avoid almost certain conflict.

The case based reasoning system establishes the following three scenarios.

1. The agents are presented with a problem for which they have no exact case match, nor do they have any previous experience that would facilitate an informed decision for an initial camera position. Therefore the agents choose an initial position that is based on selfish desire.

2. The agents have previously computed a set of camera positions for a previous scene that is a translational variant of the current scene. In this case the agents do not have a case that matches exactly. The agents therefore must

offer an informed guess for their initial decisions.

3. The agents have previously computed a set of camera positions for the current scene. In this case we have presented the agent with a problem for which it has exact prior experience. The agents can therefore utilize this experience to shortcut the negotiation process.

The first case corresponds to the method used by the agents in the two previous examples shown so far. Without any prior experience, the agents offer an initial guess based on selfish desire and then start the negotiation process. To illustrate this more effectively, we utilize another model as shown in figure 5.25. The target is shown in figure 5.26. We shall deploy three cameras to examine the target. Note that the target is partially occluded by other objects in the scene so we require multiple cameras for simultaneous coverage of the surface of the target.

Figure 5.27 shows the final positions of the cameras relative to the scene after the negotiation process was completed. Figure 5.28 shows the corresponding views of the target object obtained from each of the cameras. From the views we can see most of the target object. Table 5.9 shows the initial decisions and the final decisions of the agents. It is apparent from table 5.9 that the initial decisions of the agents were all in conflict. This is consistent with the fact that the agents make their initial decisions without any previous knowledge about the problem and also without any knowledge about the other agents' desires. The cameras all have the same range of motion, hence, the camera position that provides maximum utility is available to all the agents. The agents initially choose this position in an effort to

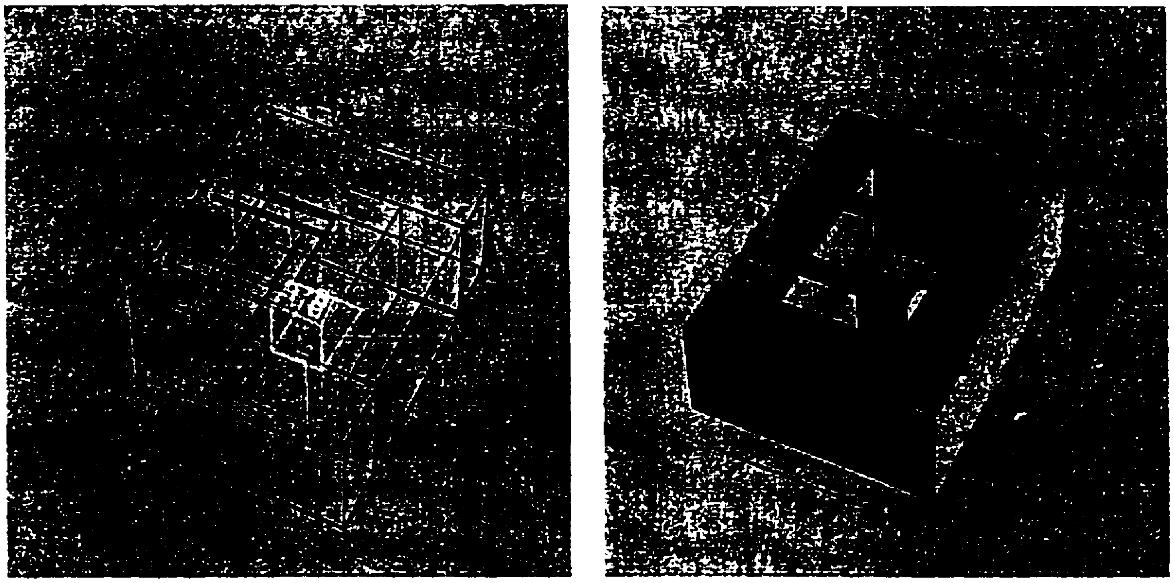


Figure 5.25: CAD and Rendered Model of Scene

maximize their utilities. The initial decision yields a maximum unadjusted[1] utility of 79.65. The resulting conflict situation must be resolved and as a result more communication is necessary.

We use the number of messages sent by the agents as a measure of the effort required to arrive at a particular solution. It is important to note that some messages

[1]The unadjusted utility does not take into account intersecting fields of view.

Agent	Initial Decision	Final Decision
1	1.6808,-0.3439,2.8606	-0.3362,0.3283,1.8941
2	1.6808,-0.3439,2.8606	-0.3362,-1.6886,2.5074
3	1.6808,-0.3439,2.8606	1.6808,-0.3439,1.8007

Table 5.9: Initial and Final Decisions by the Agents

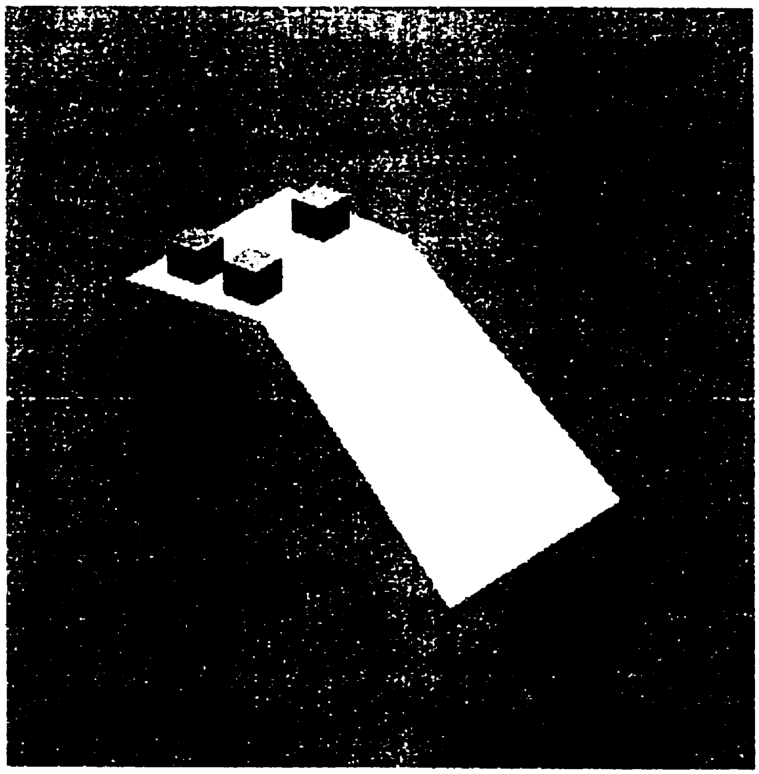


Figure 5.26: Target for Inspection

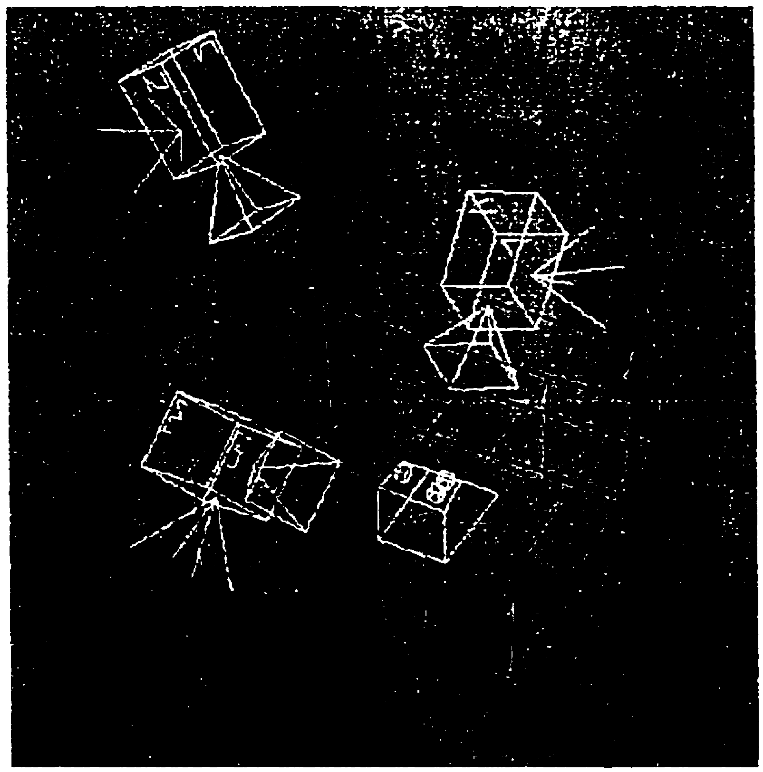


Figure 5.27: Camera Positions Relative to Scene

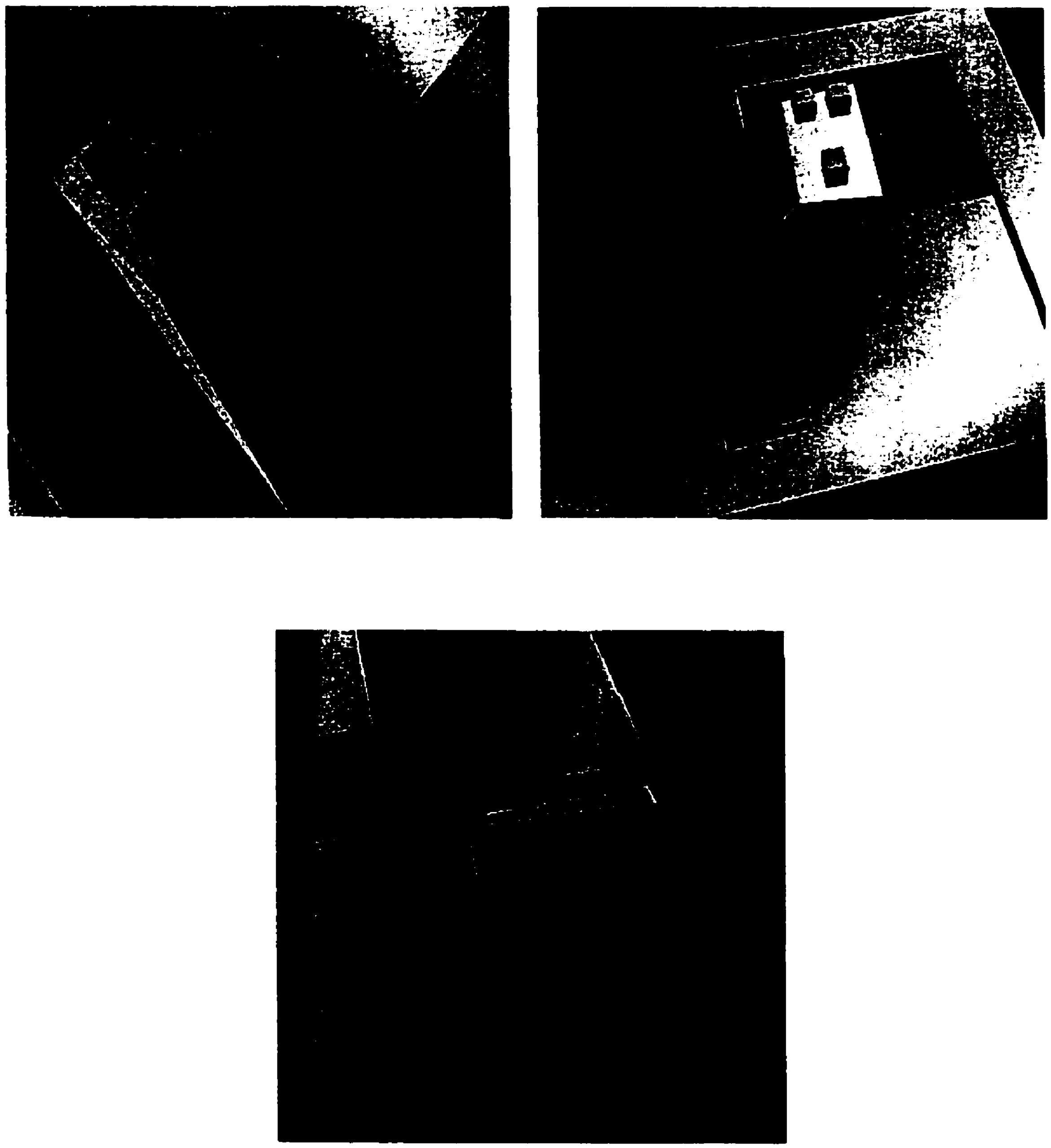


Figure 5.28: Final Views for 3 Cameras

Agent	No. of Messages	Final Utility
1	35	52.70
2	28	48.61
3	48	29.16

Table 5.10: Performance Measures Without Learning

are unavoidable such as FVL's. ALU's and other basic informational message types. However. these are consistent regardless of the nature of the problem being solved. Hence. any change in the number of messages sent by an agent will necessarily be as a result of the absence or presence of conflict situations or unacceptable camera positions.

Table 5.10 lists the number of messages sent by each agent for the aforementioned problem and their final utility values. This illustrates the performance measure of the system with no prior case experience.

To illustrate the effects of learning. the above problem was again presented to the agents. However. this time their initial decisions were made based on the fact that they had seen the exact problem before. Hence they already have a solution to the problem in their respective case bases. Table 5.11 shows the rather drastic decrease in the number of messages sent by the agents as compared to the initial results obtained in table 5.10. From an intuitive perspective. these results are to be expected since the agents have an exact match of the problem in their case base. Hence. their initial decisions are actually well informed to the extent that they are the final decisions. As table 5.12 shows. they arrive at the same final positions. The residual message counts serve the purpose of broadcasting,verifying and accepting the final camera positions.

In many cases, the agents may not have an exact match to the current problem in their cases bases as previously illustrated. This can occur for two reasons. The first reason is that the agents may not have previously solved a problem that utilized the same scene. A totally new scene would suggest that the agents have no prior information about the scene that they can utilize in making any initial decisions. This situation would require that the agents proceed with the normal course of negotiations to arrive at a solution. The second situation is where the scene has been rotated or translated relative to the bounding polyhedron that represents the range of motion of the cameras. In this case, the agents are capable of producing an initial estimate of the final positions of the cameras based on their previous solution to a similar problem.

As previously described, the initial estimate is based on finding positions for the cameras in the new scene such that their spatial arrangement with respect to the objects in the scene is similar to the spatial arrangement stored in the case base for the previously similar scene. The initial estimate is not intended to be a final solution, however, it does offer a spatially dispersed initial decision that does not require conflict resolution. This can reduce the amount of messages required

Agent	No. of Messages	Final Utility
1	9	52.70
2	13	48.61
3	11	29.16

Table 5.11: Performance Measures With Learning

Agent	Initial Decision	Final Decision
1	-0.3362,0.3283,1.8941	-0.3362,0.3283,1.8941
2	-0.3362,-1.6886,2.5074	-0.3362,-1.6886,2.5074
3	1.6808,-0.3439,1.8007	1.6808,-0.3439,1.8007

Table 5.12: Initial and Final Decisions by the Agents

Agent	X	Y	Z	Utility
1	-0.3362	0.3283	2.8607	81.25
2	1.0085	1.0007	2.0941	54.86
3	1.6808	-0.3440	2.8607	33.33

Table 5.13: Final Camera Positions Using Translated Scene without Learning

to arrive at the final solution.

The method is illustrated below for the same scene. Figure 5.29 shows the top view of the scene used in the previous experiment relative to the bounding polyhedron of the camera ranges of motion. Figure 5.30 shows the same scene after it has been translated by 0.8 metres and 0.6 metres in the X and Y direction respectively, relative to the world coordinate system.

As a control experiment, the agents first used the translated scene shown in figure 5.30 without any prior experience. That is to say, no information concerning the original scene or the translated scene was supplied to the agents. The resulting viewpoint positions and utility values that were obtained by the agents are listed in table 5.13. The corresponding views of the translated scene are shown in figure 5.31.

As before, we use the number of messages transmitted by an agent as an in-

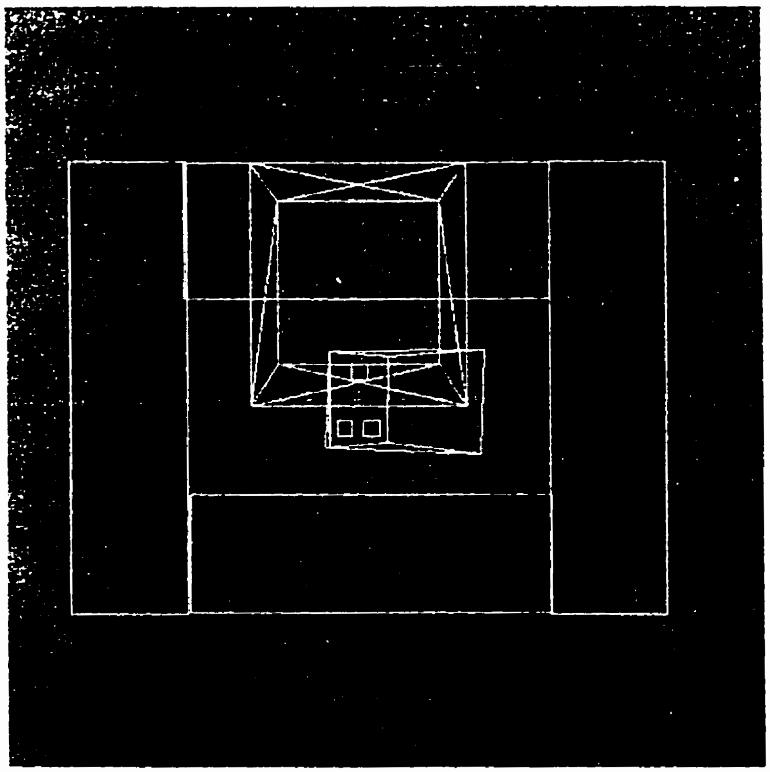


Figure 5.29: Original Scene Relative to Bounding Polyhedron

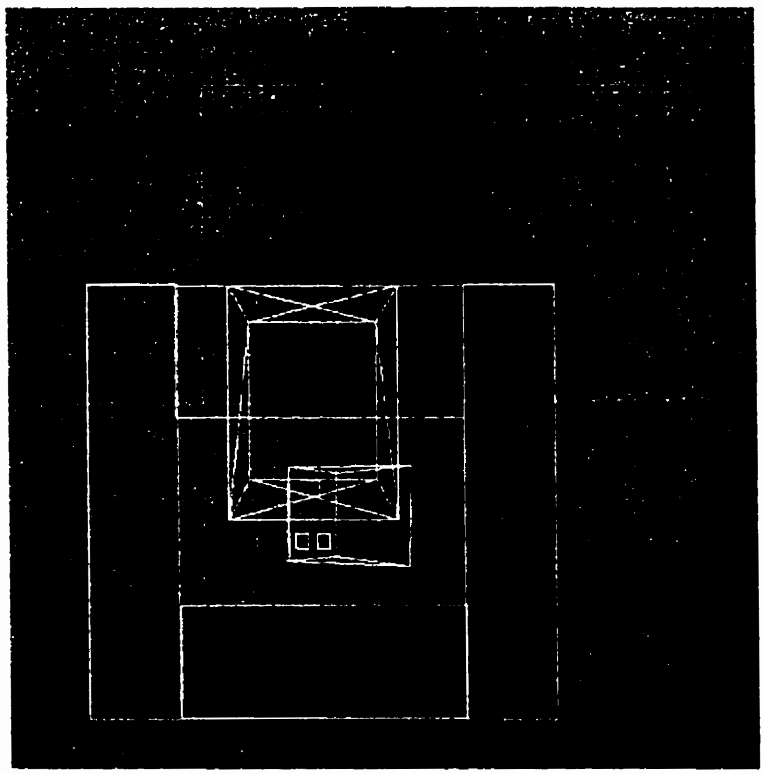


Figure 5.30: Translated Scene Relative to Bounding Polyhedron

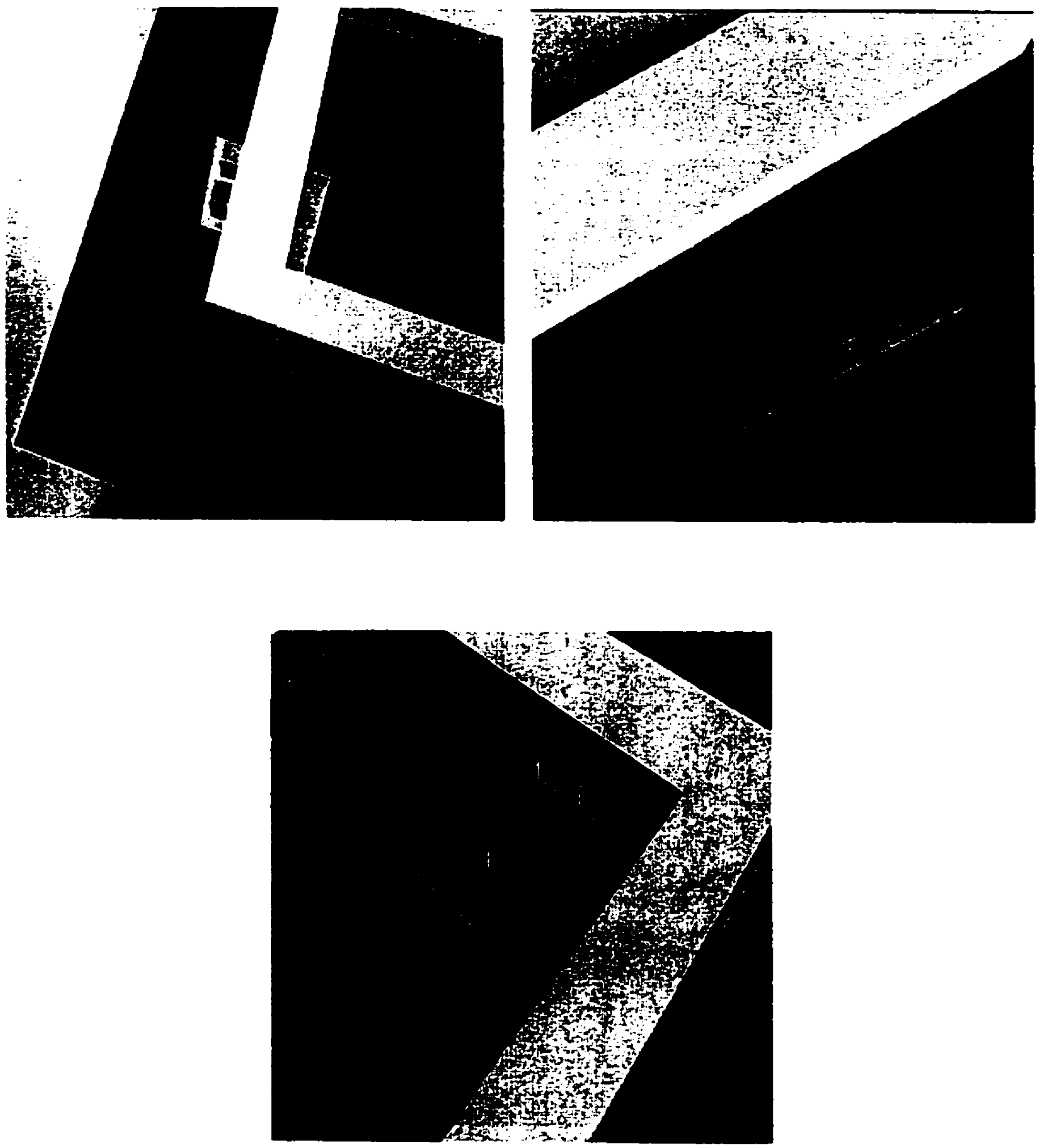


Figure 5.31: Final Views for Translated Scene without Learning

Agent	No. of Messages
1	26
2	33
3	34

Table 5.14: Performance Measures Without Learning for Translated Scene

Agent	X	Y	Z	Utility
1	-0.3362	0.3283	2.8607	81.25
2	1.6808	0.3283	2.8606	53.00
3	1.6808	1.6730	1.8008	22.52

Table 5.15: Final Camera Positions Using Translated Scene With Learning

dicator of the effort required to arrive at an acceptable solution. The number of messages sent for each agent is listed in table 5.14.

This experiment was then repeated with two different initial starting conditions. In the first situation, the agents were allowed to utilize the model of the original scene (prior to the translation) as shown in figure 5.29 as a basis for an initial estimate. Therefore, using the original scene, the agents estimated camera positions that would provide the similar spatial relationships between the cameras and the objects within the scene. The final positions arrived at by the agents are listed in table 5.15. The corresponding views are shown in figure 5.32.

The number of messages transmitted by each agent is listed in table 5.16.

The results indicate that there is a slight performance gain when using an initial estimate of the final positions. The agents required 16% fewer messages when an initial estimate was used. Another important observation concerns the actual final

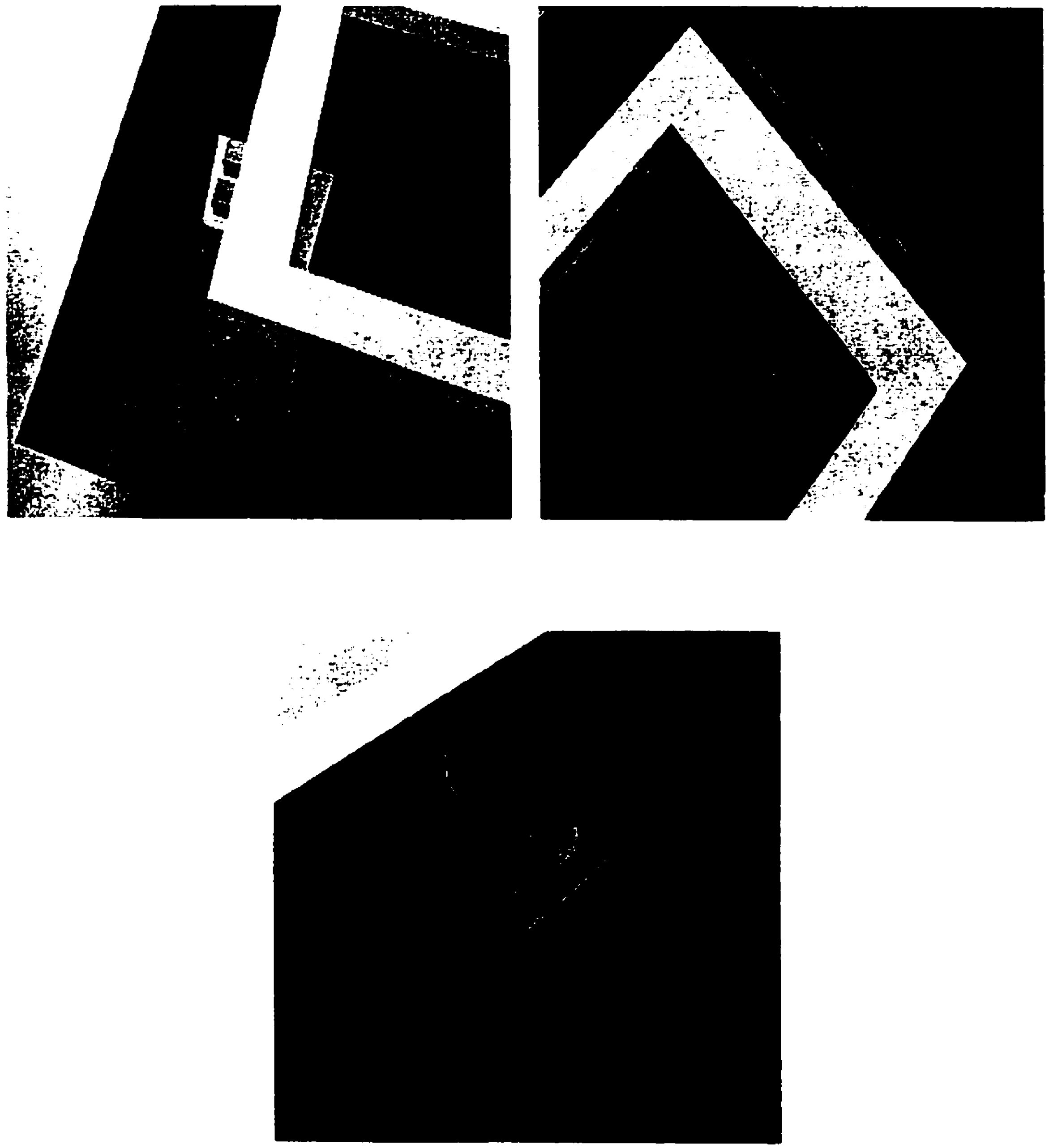


Figure 5.32: Final Views for Translated Scene With Learning

Agent	No. of Messages
1	23
2	26
3	30

Table 5.16: Performance Measures With Learning for Translated Scene

positions that were obtained. There is a difference in final positions listed in tables 5.16 and 5.14. This indicates that the initial position considered by the agents can influence the final outcome. This observation is consistent with the fact that the agents rank the possible camera positions differently depending on whether or not an initial estimate for a camera position is being considered.

In the case that no initial estimate is being considered by the agents, then the possible camera positions are evaluated with the assumption that no other agents exist. That is to say, the utility value assigned to a given camera position is done without consideration for its spatial relationship to any other possibly occupied camera position. The first position chosen by an agent will be a randomly chosen position that is a member of the set of positions with the highest utility value. Therefore, when no initial estimate is used, the agents start from the same point in the search space of utility values. As a result, repeating any given experiment without initial position estimates will yield the same final camera positions. However, in the case where an initial estimate is available, then all camera positions are initially ranked based on their spatial relations to the initial estimated positions of the agents. Hence, the difference of the utility values arrived at in both situations influences the subsequent choices of the agents when utilizing a greedy algorithm

Agent	No. of Messages
1	10
2	14
3	15

Table 5.17: Performance Measures With Learning on Translated Scene

approach.

In the second situation, we repeated the experiment using the same translated scene but we allowed the agents to use whatever knowledge they had acquired with regards to the translated scene and the results of their previous negotiations. As a result, the initial estimates of the agents corresponded to the final positions obtained from the previous experiment without learning. Hence the final positions are the same as listed in table 5.15. In this case, the agents had prior knowledge of the same scene and the initial estimates were actually the final positions chosen.

Table 5.17 shows the number of messages sent by each of the participating agents. The results indicate a 58% reduction in the number of messages required to arrive at a solution.

5.5 Discussion

We have illustrated the performance characteristics of the system for various problem sets. In general, the quality of the solution obtained relies on the applicability of the greedy selection approach to the problem set. In situations where the greedy approach to viewpoint consideration closely matches the characteristics of the optimal solution, then the results will be optimal or very nearly so. This is exemplified by the second experiment involving multiple targets as summarized in table 5.6. However, in situations where the solution space is characterized by many local suboptimal maximums, the approach presented here will provide a solution that is suboptimal to some degree. However, the advantage is that the solution is obtained at an efficiency much greater than that of an exhaustive search.

The point of reference (initial viewpoint chosen) serves as a basis for the evaluation of the utilities of all other viewpoints. Hence, the choice of initial viewpoint is usually based on the viewpoint yielding the highest initially evaluated utility. This is based on the expectation that such a viewpoint contributes in some way to the optimal or near optimal solution. When utilising an initial guess based on prior knowledge using the case based reasoning system, the assurance of initially choosing the best viewpoint is no longer available and as such can result in a decrease in the quality of the solution. The experiments have indicated that there is some trade-off inherent in the efficiency gained in finding a solution and the similarity of the problem to a previously encountered problem. In order to benefit from increased efficiency therefore, initial estimates based on a previously encountered problem, should be reserved for problem instances that are very similar to those stored in the case base. In cases that do not meet this criterion, the agents should start the negotiation process without the initial solution estimate.

The experiments described in this chapter serve to illustrate the feasibility of the agent approach to sensor planning with a representative of set of problem sets and scenarios. They demonstrate the ability of the system to provide solutions that are not necessarily optimal, but are nevertheless functionally acceptable for the task at hand. The inherent flexibility of the system encompasses not only the ease of incorporating additional cameras but also the effortless transition from single-target coverage to simultaneous multiple target coverage. Finally, the experiments have demonstrated the ability of the system to utilize prior knowledge to improve its efficiency by using a decentralized case based system. The following chapter summarizes the main contributions of this thesis and also provides a summary of the future work to be carried out.

Chapter 6

Conclusions and Future Research

The primary goal of this thesis was to develop a framework for the application of agent technology to the problem of planning multiple mobile sensors in a modeled environment. Such a framework consists of the necessary structures. coordination algorithms. communication protocols and learning algorithms for the autonomous generation of sensor position coordinates. We have specifically focused on situations where a single sensor may be infeasible for achieving the vision task. We have developed such a framework and have demonstrated its feasibility by experiment.

The main strengths of this system are based on its inherent flexibility and autonomy. Such a system would be ideal for real world applications where sensor planning is required in a flexible manufacturing or quality control environment. For example. the inspection of relatively large manufactured parts can be accomplished more efficiently by simultaneously deploying multiple cameras to cover the specified targets or target areas. In addition. since the sensing system requires minimal human intervention. the trial and error methods that contribute to the inefficiency of the inspection phase can be virtually eliminated. Another application of this system is the planning of camera positions by simulating the views that would be obtained for a given scene, prior to actually deploying the cameras. Once the number of cameras and the camera positions and orientations have been established by simulation, the actual cameras can be deployed to their respective positions. Again, this eliminates any costly trial and error process. In the rest of this chapter, we outline the major contributions of this thesis and the limitations of the proposed system. We conclude by providing some recommendations for future research in this fascinating area.

6.1 Contributions

The major contributions of this thesis are best explored with reference to the capabilities of the currently available systems for sensor planning as described in chapter 2. We necessarily limit our discussion to systems that operate in a static modeled environment since this is the operating environment for the system presented in this thesis.

1. Scalability of the Vision Planning System

   The systems that perform vision planning in a static environment are centred around single camera systems or stereo vision systems. These are not necessarily able to meet the demands of a wide variety of sensing tasks. In situations where more sensors are required, for example, covering large targets or multiple targets, such systems are not easily scalable. The system presented here provides a level of scalability that goes well beyond those systems previously discussed. A consistent and robust agent model along with precise communication protocols allows the agents to be easily replicated for the control of the additional sensors required for a more complex sensing task. This is achieved with the minimum amount of user intervention. Hence the autonomy of the system is maintained at all levels. Such a system can be applied to a much wider variety of sensing tasks.

2. Simultaneous Multi-feature Inspection

   The current systems are designed for the sensor coverage of a specific target feature. The coverage of multiple spatially related features is achieved by the sequential planning of viewpoints for the individual features. Hence the model is rotated or moved relative to the sensor to position the sensor at the next position to cover spatially distinct targets. This sequential process is inefficient and not suitable for situations where the simultaneous coverage of spatially distinct features is a necessity. The system described in this thesis alleviates this inadequacy by planning for all features simultaneously and deploying the sensors such that all features may be simultaneously covered by one or more sensors. This allows for a more efficient feature inspection process.

3. Improved Fault Tolerance

   The systems that we have discussed rely on the centralized computation of sensor positions. In the case of the agent based system developed by Okoshi et al [32], the actual planning of the camera positions are carried out off-line by a centralized process. In such systems, the failure of the central computing node implies the failure of the system as a whole. In our system, the failure of a computing node or agent results in the re-deployment of the other agents so that the sensing task can still be achieved. This is done autonomously and provides a level of fault tolerance that is not available in the existing systems.

4. Sensor Heterogeneity

   Since each of the sensors in the system is modeled by the controlling agent, it is possible to utilize sensors with various optical properties in the same sensing task. The exchange of information amongst the agents via a precise suite of messages provides a layer of abstraction to the underlying sensor properties and permits the use of different sensors. For example, cameras may have different focal lengths, resolution and fields of view. However, they can still be coordinated using the same agent models and protocols. This advantage does not exist in current systems.

5. Learning

   The idea of improving the efficiency of a sensor planning system over time through learning is completely absent from current systems that operate in a static modeled environment. In cases where the same problem or similar problems are repeatedly presented to such systems, the planning process is consistent in the time taken to arrive at a solution. In this system, we incorporate cased base learning as a means of improving the efficiency of the system either through autonomous learning (cases are recorded by the agents themselves) or by allowing the user to present case knowledge to agents within the system. In either situation, the system can improve on the time taken to reach a solution by utilizing this stored case knowledge and applying it to current problems. In addition, the case knowledge is decentralized and therefore each agent can learn based on its own experience. Hence new agents without any prior knowledge can still negotiate with more experienced agents in a given sensing task. Eventually, such newer agents will acquire case knowledge that is representative of their own experience.

In summary, the choice of an agent based approach to the sensor planning problem was made due to the inherent advantages of such a decentralized and flexible programming method over centralized single sensor systems. We have succeeded in providing the necessary negotiation and coordination algorithms that form the basis for the efficient and robust juxtaposition of two traditionally separate areas of research, namely, agent technologies and sensor planning. As presented in this chapter, such a union offers significant contributions to the field of sensor planning.

6.2 Limitations

The current limitations of this system are summarized below.

1. The agents must have complete knowledge of the scene

   The assumption made in this thesis is that the agents have a complete and accurate CAD model of the scene and this does not change during the course of negotiations. Hence there is no uncertainty in the information available to the agents. In some situations, this may not be possible to achieve and thus the system may not be able to provide an acceptable solution to the sensing task.

2. All possible camera parameter combinations are not considered

   In this system, the magnification, focus and orientation of the cameras are not planned by the agents. These parameters are pre-determined and utilized by the agent to plan the corresponding camera position. This approach can eliminate possible combinations of these parameters that can provide a more accurate solution to the problem at hand.

3. The planned position may be infeasible

   Although the set of candidate positions are based on the range of motion of the cameras, the combination of position and orientation may be infeasible from the perspective of the actual capabilities of the robotic manipulator. In such situations the user would have to eliminate the position from the candidate set.

6.3 Future Research

Although there has been a large body of research carried out in the application of agent technologies to various fields, we have not seen the same momentum in the application of agent technologies to sensor planning. This thesis presents a basis for the pursuance of improved algorithms in this application area and as such leaves a number of areas open to further research. Among these are:

1. Improving the quality of the solution

   Significant improvement can be made to the quality of the final solution obtained by the agents. As we are aware, a greedy approach does not guarantee an optimal solution. It would be beneficial to guarantee a degree of optimality of the solution such the result is not far from the optimal solution. This can be achieved through more informed search methods carried out by each agent in parallel such that the resulting arrangement set is a combination of the optimized decisions of each of the agents.

2. Dealing with missing information

   Currently the system does not deal with missing information within the scene model. Each agent has all the knowledge required to carry out its task. In cases where this is not possible, for example when only limited storage is available, model information may need to be distributed amongst the agents. In such a scenario, each agent would require a means of recognizing its need for information that it does not have and the ability to communicate a request for such information to other agents in the group. By distributing the model information amongst all agents, the storage requirements of each agent can be decreased.

3. Improving on the learning system

The learning method used in this thesis is based on the case base of information extracted from the CAD models used by the agent. It is also possible to incorporate image information such that the agents can search for positions that would result in the same image of the scene that was previously obtained or at least similar to that previously obtained. The idea then would be to utilize the CAD information as a starting point for a more refined search that could ultimately result in more accurate solutions.

4. Dealing with unexpected occurrences

   Currently, the system requires that every object in the scene be represented by the CAD model available to the agents. In many situations, it is possible for the physical scene to contain unexpected objects. In such cases, the agents should revise their decisions to accommodate the presence of such objects and hence plan views that do not include these objects. In order to accomplish this the agents would need to obtain feedback from the images that are taken from the planned positions. By comparing what is represented in the images to the expected scene as represented by the CAD model, the agents can determine whether or not an unexpected object is within their field of view. Once this is determined, the planned position can be revised accordingly.

5. Establishment of the limits to the coordination of the agents

   An important area that is not addressed within this thesis is the establishment of the limits of the coordination mechanism. In other words, how many agents and corresponding cameras can be added to the system before the coordination breaks down. Such limits can be established by empirical means by increasing the amount of agents and observing the effects on efficiency and convergence over a variety of problem sets. such benchmarks are important in establishing the range of problems to which this system is applicable.

Appendix A

Object DXF Representation

The DXF representation shown in table A.1 specifies the vertices in 3D space that comprise one face of a cube. The triangular facets represented are obtained from the 3D face information. Table A.2 shows the coordinates of the facets that were obtained from the representation in table A.1.

3DFACE
8
Cube
10
-0.508326
20
0.500507
30
0.010537
11
-0.508326
21
-0.500507
31
0.010537
12
0.492688
22
-0.500507
32
0.010537
13
0.492688
23
0.500507
33
0.010537
62
0
0

Table A.1: DXF Single Face Description

Facet	Vertex	Coordinates (X,Y,Z)
1	A	-0.508326,0.500507,0.010537
1	B	-0.508326,-0.500507,0.010537
1	C	0.492688,-0.500507,0.010537
2	D	0.492688,-0.500507,0.010537
2	E	0.492688,0.500507,0.010537
2	F	-0.508326,0.500507,0.010537

Table A.2: Facet Information from DXF representation

Appendix B

Agent Data Generation

Figure B.1 illustrates the main steps in generating the data required for each agent.

The formulation of the bounding polyhedron is achieved by considering the union of the ranges of motion of all the cameras involved in the sensing task. The bounding polyhedron is voxelated by considering equally spaced voxels within the volume starting at one face and continuing to the opposite face. The distance between such voxels is set by the user.

The identification of the target objects is achieved by labeling the target object in the DXF file as "Target". For scenes with multiple targets, the targets are labeled as Target1.Target2.. Targetn.

The resulting agent data contains the number of facets of the target that are visible and in focus and the number of edges for each facet that is resolved for all possible viewing positions. This is based on the camera being oriented towards the centroid or centre of mass of each of the targets.

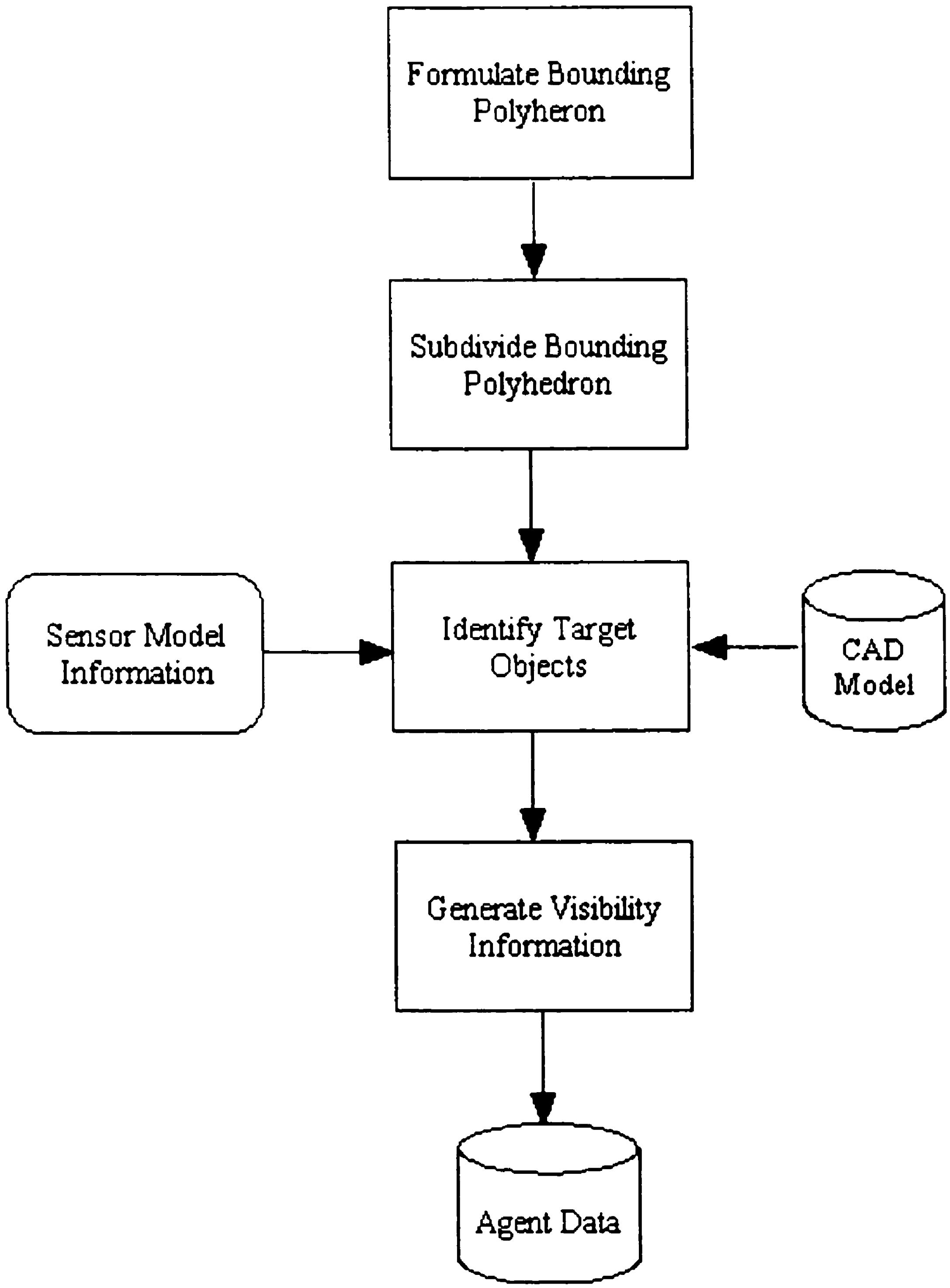


Figure B.1: Generation of Agent Data

Appendix C

Inter-agent Interaction

In order to communicate with other agents in the group. the agents must register with a common agent registry that is accessible by all agents. This registry maintains information about the agent such as whether or not the agent is currently online. the number of channels available for Dynamic Data Exchange (DDE) and the name of the agent. Every time another agent establishes a DDE connection with an agent. the number of channels available in the registry is decreased. each agent periodically scans the registry to ascertain whether or not any new agents have come online. Figure C.1 illustrates the relationship between the agent registry and the rest of the system.

The communication that takes place is achieved through the use of DDE as defined above. This is a pre-emptive process so when a message arrives, it is placed on a message queue. At a predetermined point in the agent algorithm, the messages are scanned. and the appropriate action is taken based on the message type. Since there is no synchronization process. the messages are scanned at different times

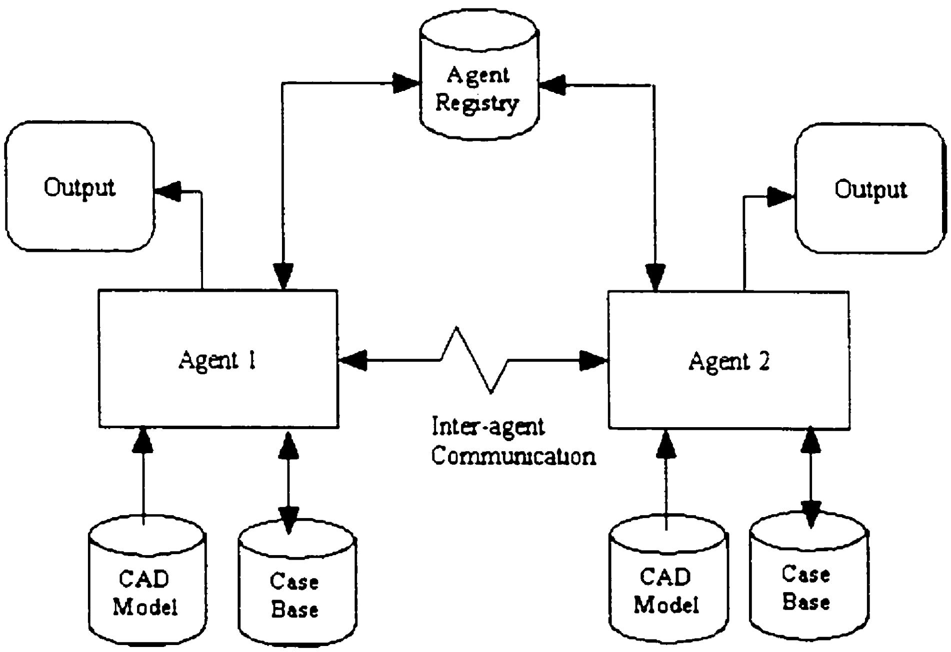


Figure C.1: Agent Interaction

since the agents operate asynchronously.

Some messages require an immediate response as is the case with the RNR (Request for Random Number) and PING message types. In this case, the message is not placed on the queue but interrupts the normal flow of the algorithm so that the appropriate response is immediately generated.

Bibliography

[1] P. K. Allen K. A. Tababanis and R. Y. Tsai. A survey of sensor planning in computer vision. *IEEE Transactions on Robotics and Automation*. 11(1):86 - 104. 1995.

[2] M. R. Korn and C. R. Dyer. 3D multiview object representations for model-based object recognition. *Pattern Recognition*. 20(1):91 - 103. 1987.

[3] A. M. Wallace E. Trucco and V. Roberto. Model-based planning of optimal sensor placement for inspection. *IEEE Transactions on Robotics and Automation*. 13(2):182 - 193. 1997.

[4] R. A. Cromwell S. A. Hutchinson and A. C. Kak. Planning sensing strategies in a robot workcell with multi-sensor capabilities. In *Proceedings of the IEEE International Conference on Robotics and Automation*. pages 1068 - 1075, 1988.

[5] P.K. Allen K. A. Tarabanis and R.Y Tsai. The MVP sensor planning system for robotic vision tasks. *IEEE Transactions on Robotics and Automation*, 11.(1):73 - 85. 1995.

[6] P. Gmytrasiewicz D. Cook and L. Holder. Sensor planning and cordination in multi-agent systems. In *Proceedings of the IEEE International Conference of Robotics and Automation*, 1991.

[7] M. Ishii S. Sakane and M. Kakikura. Occlusion avoidance of visual sensors based on hand eye action simulator system: HEAVEN. *Advanced Robotics*, 2(2):149 - 165, 1987.

[8] S. Sakane and T. Sato. Automatic planning of light source and camera placement for an active photometric stereo system. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1080 - 1087, 1991.

[9] B.G. Batchelor. A prolog lighting advisor. In *Proceedings of SPIE Intelligent Robotics: Systems and Applications*, volume 1193, pages 295 - 302, 1989.

[10] S. Russell and P. Norvig. *Artificial Intelligence. A modern Approach*. Prentice Hall Inc., 1995.

[11] Hyacinth S. Nwana. Software agents: An overview. *Knowledge Engineering Review*, 11(3):205 - 244, 1996.

[12] H. Ghenniwa and M. Kamel. Interaction devices for coordinating cooperative distributed systems. *Intelligent Automation and Soft Computing*, 6(2):173 - 184, 2000.

[13] N. Findler and R. Lo. An examination of distributed planning in the world of air traffic control. *Journal of Parallel and Distributed Computing*. 3:411 - 431. 1986.

[14] Peter Wayner. *Agents Unleashed A Public Domain Look at Agent Technology*. AP Professional. Massachusetts. 1995.

[15] C. I. Connely. The determination of next best views. In *Proceedings of the IEEE International Conference on Robotics and Automation*. pages 432 - 435. 1985.

[16] J. J. Clark and N. J. Ferrier. Modal control of an attentive vision system. In *Proceedings of the International Conference on Computer Vision*. pages 514 - 523. 1988.

[17] P.K. Allen K. A. Tarabanis and R.Y Tsai. Sensor planning with hierarchically distributed perception net. In *IEEE International Conference on Robotics and Automation*. pages 2663 - 2669. 1995.

[18] R.Y Tsai K. A. Tarabanis and S. Adams. Planning viewpoints that simultaneously satisfy several feature detectability constraints for robotic vision. In *Proceedings of the Fifth International Conference of Advanced Robotics*. 1991.

[19] R.Y Tsai K. A. Tarabanis and P.K. Allen. Automated sensor planning and modeling for robotic vision tasks. In *Proceedings of the 1991 IEEE International Conference of Robotics and Automation*, pages 76 - 82. 1991.

[20] H. Okoshi S. Sakane, T. Sato and M. Kakikura. A distributed sensing system with 3D model-based agents. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1157 - 1163, 1993.

[21] C. Cowan and P. Kovesi. Automatic sensor placement from vision tasks requirements. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 10(3):407 - 416, 1988.

[22] C. Cowan. Model-based synthesis of sensor location. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 900 - 905, 1988.

[23] R. M. Haralick S. Yi and G. Shapiro. Automatic sensor and light source positioning for machine vision. In *Proc. of the 10th Int'l Conf on Pattern Recognition*, pages 55 - 59, 1990.

[24] D. Roberts and A. Marshall. Viewpoint planning for complete three dimensional object surface coverage with a variety of vision system configurations. In *Proc. o fthe 9th British Machine Vision Conf.*, volume 2, pages 740 - 750, Southhampton UK, 1997.

[25] B. Klaus and P. Horn. *Robot Vision*. MIT Press McGraw Hill Book Company, 1986.

[26] A. Khan E. Lawler, J. Lenstra and D. Shmoys. *The Traveling Salesman Problem*. New York: Wiley, 1985.

[27] A. Novini. Lighting and optics expert system for machine vision. In *Proc. Optics, Illumination, Image Sensing*, pages 1105 - 1019, 1988.

[28] Alper Caglayan and Colin Harrison. *Agent Sourcebook: A complete Guide to desktop. Internet and Intranet Agents.* John Wiley and Sons Inc., 1997.

[29] V. Lesser E. Durfee and D. Corkhill. Coherent cooperation among communicating problem solvers. *IEEE Transactions on Computers*. 6:1275 - 1291. 1987.

[30] H Nii. Blackboard systems. In P. Cohen A. Barr and E. Feigenbaum, editors. *Handbook of Artificial Intelligence*, volume 4, pages 3 - 147. Addison Wesley, 1989.

[31] V. Lesser L. Erman. F. Hayes-Roth and D. Reddy. The HEARSAY II speech understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys*, 12:213 - 253. 1980.

[32] T. Sato H. Okoshi. S. Sakane and M. Kakikura. A distributed sensing system with 3D model-based agents. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. pages 1157 - 1163. 1993.

[33] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*. C-29(12):1104 - 1113, 1980.

[34] P. Gmytrasiewicz D. Cook and L. Holder. Decision theoretic multi-agent sensor planning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 18(10):1013 - 1023, 1991.

[35] K. Tarabanis and R. Tsai. Computing viewpoints that satisfy optical constraints. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 152 - 158, 1991.

[36] E. Durfee and T. Montgomery. Coordination as distributed search in a hierarchical behavior space. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1363 - 1378, 1991.

[37] N. Griffeth and H. Velthuijsen. Reasoning about goals to resolve conflicts. In *Proceedings of the International Conference on Intelligent and Cooperative Information Systems*, pages 197 - 204, 1993.

[38] J. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. In *Proceedings of the 3rd ACM Conference on Principles of Distributed Computing*, pages 50 - 61, 1984.

[39] P. Cohen and H. Levesque. Intention = choice + commitment. In *Proceedings of AAAI-87*, pages 410 - 415, 1987.

[40] E. Durfee. *Coordination of Distributed Problem Solvers, University of Massachusetts*. Kluer Academic Press, Boston, 1988.

[41] R. Davis and R. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20:63 - 109, 1983.

[42] H Muller N. Khun and J. Muller. Task decomposition in dynamic agent societies. In *International Symposium on Autonomous Decentralized Systems*, pages 165 - 171, 1993.

[43] T. Ishida M. Yokoo, E. Durfee and K. Kuwabara. Distributed constraint satisfaction for formalizing distributed problem solving. In *Proceedings of the Twelfth International Conference on Distributed Computing Systems*, pages 614 - 621, 1992.

[44] M. Yokoo. *Asynchronous Weak-commitment Search for Solving Distributed Constraint Satisfaction Problems*. Springer-Verlag, 1995.

[45] Daniel Olfe. *Computer Graphics for Design: From Algorithms to AutoCAD*. Prentice-Hall, 1995.

[46] Gerhard Weiss, editor. *Multi-Agent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 2000.

[47] Mark Allen Weiss. *Data Structures and Algorithm Analysis in C++*. Addison-Wesley, 1999.

[48] G. Fekete and L. Davis. Property spheres: A new representation for 3D object recognition. *IEEE Workshop on Computer Vision*, 13(2):192 - 201, 1984.

[49] S. Lee and X. Zhao. New sensor paradigm and its application to robot self-localization. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, volume 2, pages 462 - 467, 1995.

[50] V. Lesser M. Prasad and S. Lander. Retrieval and reasoning in distributed case bases. *Journal of Visual Communication and Image Representation: Special Issue on Digital Libraries*, 7(1), March 1996.

[51] M. Ishii S. Sakane and M. kakikura. Setup planning of active visual sensing for autonomous robots. *Journal of Robotics and Mechatronics*. 3(3):196 - 200. 1991.

[52] T. Sato S. Sakane, R. Neopld and Y. Shirai. Illumination setup planning for a hand-eye system based on an environmental model. *Advanced robotics*. 6(4):461 - 482, 1992.

[53] A. Phillips M. Johnston and P. Laird. Minimizing conflicts: A heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence*. 58(1-3):161 - 205. 1992.

[54] J. Gallier. *Curves and Surfaces in Geometric Modeling*. Morgan Kaufmann Publishers, 2000.

[55] H. Choset and J. Burdick. Sensor based planning part I: The generalized voronoi graph. In *IEEE International Conference on Robotics and Automation*, pages 1649 - 1655. 1995.

[56] P. K. Allen S. Abrams and K. A. Tarabanis. Dynamic sensor planning. In *Proceedings of the IEEE international Conference on Robotics and Automation*, pages 605 - 609, 1993.

[57] S. Abrams and P. Allen. Swept volumes and their use in viewpoint computation in robot work cells. In *Proceedings of the the IEEE International Symposium on Assembly and Task Planning*, August 1995.

[58] S. Abrams and P. Allen. Computing swept volumes for sensor planning tasks. In *Proceedings of the DARPA Image Understanding Workshop*. 1994.

[59] N. Carver and V. Lesser. A new framework for sensor interpretation: Planning to resolve sources of uncertainty. In *Proceedings of the National Conference on Artificial Intelligence*. pages 724 - 731. July 1991.

[60] H. Choset and P. Pinon. Coverage path planning: The boustrophedon decomposition. In *Proceedings of the International Conference on Field and Service Robots*. Canberra. Australia. 1997.

[61] B. Horling and V. Lesser. Using diagnosis to learn contextual coordination rules. In *Proceedings of the AAAI-99 Workshop on Context for AI Applications*. July 1999.

[62] R. Vincent D. Jenson. M. Atigetchi and V. Lesser. Learning quantitative knowledge for multiagent coordination. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*. August 1999.

[63] K. Nagatani and H. Choset. Toward robust sensor based exploration by constructing a reduced generalized voronoi graph. In *Proceedings of IROS'99*, 1999.

[64] S. Noh and P. Gmytrasiewicz. Towards flexible multi-agent decision making under time pressure. In *Proceedings of the Sixteenth International Joint Conference on artificial Intelligence*. pages 492 - 498. 1999.

[65] S. Noh and P. Gmytrasiewicz. Uncertain knowledge representation and communicative behaviour in coordinated defense. In *Workshop on Agent Communication Languages of IJCAI-99*, pages 119 - 128, 1999.

[66] S. Noh and P. Gmytrasiewicz. Implementation and evaluation of rational communicative behaviour in coordinated defense. In *Proceedings of the third International Conference on Autonomous Agents*, pages 123 - 130. ACM Press, 1999.

[67] V. Lesser N. Prasad and S. Lander. Learning experiments in heterogeneous multi-agent systems. In *Proceedings of IJCAI-95 Workshop on Adaptation and Learning in Multi-Agent Systems*, 1995.

[68] T. Sugawara and V. Lesser. Learning to improve coordinated actions in cooperative distributed problem solving environments. In *Machine Learning*. Kluwer Academic Publishers, 1998.

[69] D. Suryadi and P. Gmytrasiewicz. Learning models of other agents using influence diagrams. In *Proceedings of the Modeling: The Seventh International Conference*. Springer Wien, New York, 1999.

[70] B. Batchelor. A prolog lighting advisor. In *Proc. of SPIE Intelligent Robotics VIII: Systems and Applications Vol. 1193*, pages 295 - 302, 1989.

[71] G. Weiss, editor. *Distributed Artificial Intelligence meets Machine Learning*. Lecture Notes in Artificial Intelligence. Springer Verlag, 1997.

[72] R. Kakehi and M. Tokoro. A negotiation protocol for conflict resolution in multi-agent environments. In *Proceedings of the International Conference on Intelligent and Cooperative Information Systems*, pages 185 - 196, Rotterdam, The Netherlands, 1993.

[73] J. Zeng X. Lin and Q. Yao. Optimal sensor planning with minimal cost for 3D object recognition using sparse structured light images. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3484 - 3489, Minneapolis, MN, USA, 1996.

[74] V. Lesser M. Prasad and S. Lander. Learning experiments in a heterogeneous multi-agent system. In *IJCAI Workshop on Adaptation and Learning in Multi-Agent Systems*, pages 3484 - 3489, Montreal, Canada, 1995.

[75] G. Shivaram and G. Seetharaman. A new technique for finding the optical center of cameras. In *International Conference on Image Processing*, volume 2, pages 167 - 171, Chicago, IL, USA, 1998.

[76] G. Zlotkin and J. Rosenchein. Negotiation with incomplete information about worth: Strict versus tolerant mechanisms. In *Proceedings of the International Conference on Intelligent and Cooperative Information Systems*, pages 175 - 184, Rotterdam, The Netherlands, 1993.

[77] T. Lee and S. Ghosh. The concept of stability in asynchronous distributed decision-making systems. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 30(4):549 - 561, 2000.

[78] J. Archibold T. Gold and R. Frost. A utility approach to multi-agent coordination. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2052 - 2057, San Francisco, CA, USA, 2000.

[79] E. Marchand and G. Hager. Dynamic sensor planning in visual servoing. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 1988 - 1993, Leuven, Belgium, 1998.

[80] H. Schuler E. Zussman and G. Seliger. Analysis of the geometrical features detectability constraints for laser-scanner sensor planning. *International Journal of Advanced Manufacturing Technology*, 9(1):56 - 64, 1994.

[81] S. Koenig and Y. Smirnov. Sensor-based planning with the freespace assumption. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 3540 - 3545, Albuquerque, NM, USA, 1997.

[82] B. Mirtich H. Choset and J. Burdick. Sensor based planning for a planar rod robot: Incremental construction of the planar rod-hgvg. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 3427 - 3434, Albuquerque, NM, USA, 1997.

[83] Y. Ye and J. Tsotsos. Sensor planning for 3d object search. *Computer Vision and Image Understanding*, 73(2):145 - 168, 1999.

[84] R. Tsai and K. Tarabanis. Model-based planning of sensor placement and optical settings. In *Proceedings of SPIE - The International Society for Op-*

tical engineering Sensor Fusion II: Human and Machine Strategies, pages 75 - 89. Philadelphia. PA. USA. 1989.

[85] I. Stamos and P. Allen. Interactive sensor planning. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 489 - 494. Santa Barbara. CA. USA. 1998.

[86] A. Briggs and B. Donald. Automatic sensor configuration for task-directed planning. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1345 - 1350. Piscataway, NJ. USA. 1994.

[87] K. Tarabanis and R. Tsai. MVP sensor planning and modeling system for machine vision. In *Proceedings of SPIE - The International Society for Optical engineering Sensor Fusion IV: Control paradigms and data structures*, pages 255 - 266. Bellingham. WA. USA. 1992.

[88] S. Abrams and P. Allen. Sensor planning in an active robot workcell. In *Proceedings of SPIE - The International Society for Optical engineering Sensor Fusion IV: Control paradigms and data structures*, pages 267 - 277. Bellingham. WA. USA. 1992.

[89] L. Page and A. Sanderson. Robot motion planning for sensor-based control. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1333 - 1340. Piscataway, NJ, USA, 1995.

[90] Georg Glaeser. *Fast Algorithms for 3D graphics*. Springer-Verlag, 1995.

[91] C. Boutilier. Sequential optimality and coordination in multi-agent systems. In *Proceedings of the sixteenth International Joint Conference on Artificial Intelligence*, volume 1, pages 478 - 485, Stockholm Sweden, 1999.

[92] J. Pitt and A Mamdani. A protocol-based semantics for an agent communication language. In *Proceedings of the sixteenth International Joint Conference on Artificial Intelligence*, volume 1, pages 486 - 498, Stockholm Sweden, 1999.

[93] A. Briggs and B. Donald. Visibility-based planning of sensor control strategies. *Algorithmica*, 26(3-4):364 - 388, 2000.

[94] M. Reed and P. Allen. Constraint-based sensor planning for scene modeling. In *Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation. CIRA'99*, pages 131 - 136, Piscataway, NJ, USA, 1999.

[95] N. Okada; T. Nagata and Z. Hongbin. Planning for parts-picking tasks by using multi-sensor visual system. In *Proceedings of the 22nd International Conference on Industrial Electronics, Control, and Instrumentation*, volume 2, pages 737 - 742, New York, NY, US, 1996.

[96] J. Manyika and H. Durant-White. *Data fusion and sensor management: A decentralized information theoretic approach*. Ellis Horwood, 1994.

[97] Y. Labrou and T. Finin. Semantics and conversations for an agent communication language. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, volume 1, pages 584 - 591, Nagoya, Japan, 1997.

[98] D. Carmel and S. Markovitch. Exploration and adaptation in multi-agent systems: A model based approach. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, volume 1, pages 606 - 611, Nagoya, Japan, 1997.

[99] E. Durfee and Y. So. The effects of runtime coordination strategies with static organizations. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, volume 1, pages 612 - 619, Nagoya, Japan, 1997.

[100] A. Armstrong and E. Durfee. Dynamic prioritization of complex agents in distributed constraint satisfaction problems. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, volume 1, pages 620 - 625, Nagoya, Japan, 1997.

[101] Duncan Marsh. *Applied Geometry for Computer Graphics and CAD*. Spinger, 1999.

[102] Jacques Ferber. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley, 1999.

[103] Herbert Freeman. *Machine Vision for Three dimensional Scenes*. Academic Press Inc., 1990.

[104] Owen Ransen. *AutoCAD Programming in C/C++*. John Wiley and Sons, 1997.

[105] Ronald L. Rardin. *Optimization in Operations Research*. Prentice-Hall, 1998.

[106] Mongi A. Abidi and Rafael C. Gonzalez. *Data fusion in Robotics and Machine Intelligence.* Academic Press Inc., 1992.

[107] Qiang Yang. *Intelligent Planning: A Decomposition and Abstraction Based Approach.* Springer-Verlag, 1997.

[108] Joseph P. Bigus and Jennifer Bigus. *Constructing Intelligent Agents with Java: A programmers Guide to Smarter Applications.* John Wiley and Sons Inc., 1998.

[109] Gerhard Weib and Sandip Sen, editors. *Adaptation and Learning in Multi-Agent Systems.* Lecture Notes in Artificial Intelligence. Springer Verlag, 1996.

[110] J. Schmidt and H. Susse. 3D object recognition using model based invariants. In *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society*, pages 2034 - 2039, New York, NY, USA, 1998.

[111] Alan Paeth, editor. *Graphics Gems V.* AP Professional, 1995.

[112] L. Hodge and M. Kamel. Multi-sensor planning in a coordinated multi-agent environment. In *Proceedings of the World Automation Conference*, Maui, Hawaii, USA, 2000.

[113] L. Hodge and M. Kamel. A simulation environment for multi-sensor planning. *To appear in the Journal of the Society for Computer Simulation*, 1, 2001.