Scene Reconstruction Using Accumulated Line-of-Sight

by

Christopher P. Stauffer

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

Master of Science in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1997

© Massachusetts Institute of Technology 1997. All rights reserved.

Author ....
Department of Computer Science and Engineering
May 9, 1997

Certified by ....
W. Eric L. Grimson
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

JUL 24 1997

Eng.

Scene Reconstruction Using Accumulated Line-of-Sight

by

Christopher P. Stauffer

Submitted to the Department of Computer Science and Engineering
on May 9, 1997, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

Abstract

Standard scene reconstruction methods attempt to determine the positions which visible elements of the environment occupy in space. Instead of determining where things are, we choose to model where things are not. We model the open space in a scene.

This is accomplished by tracking a human participant moving in the environment. Based on the position of the tracked object and where it is visible to the cameras, a 3-D vacancy grid is modified which represents the empty space in the environment. This method can produce useful spatial models itself or can be used to complement other reconstruction methods.

This thesis begins with a survey of scene reconstruction followed by a general description of this method and its advantages and disadvantages. Its application to a single camera system is illustrated by an example. The numerous considerations for extending to multiple camera systems are discussed. Possible applications and extensions are outlined and conclusions are drawn.

Thesis Supervisor: W. Eric L. Grimson
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I would like to thank both Eric Grimson and Tomas Lozano-Perez for all their help and guidance over the past two years. I feel somewhat undeserving of the challenges and help they have given to me. They have both contributed greatly to my recent success and happiness. I cannot express how much I have benefited from being part of the environment that they, and the other MIT Artificial Intelligence facaulty, have created.

I thank Gideon Stein and Jeremy DeBonet for their part in implementing the vision library which was used in this work and for their friendship. Their comments and comradery were accompanied by those of other members of the Artificial Intelligence Laboratory community, including Greg Galperin, Carl DeMarcen, Mike Leventon, Erik Miller, Carlin Vieri, Rodney Daughtrey, Polina Golland, and the AI Hockey Team.

I especially like to thank my undergraduate advisor, Paul Cooper, who not only helped me decide where I wanted to be today, but helped me get here. I would also like to thank my high school swimming coach, Mr. Downey, who taught me the value of perseverance and modesty. I thank Parul Matani for her patience, friendship, and love.

Most of all, I would like to thank my parents, William and Ruth Stauffer, and family, Bill, Jenny, Carol, and my grandmother, Hazel, without whose continuing love and support, this thesis would not have been possible. They helped me acquire the tools I needed to find challenge, happiness, love, and balance in my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In order for computer algorithms which can effectively approximate human-level visual tasks to be realized, they must either be restricted to known, structured environments or be capable of modeling the uncertainties of the environment. This is one of the prime motivations for the overwhelming interest in scene reconstruction. Scene reconstruction refers to systems which attempt to use one or more sensors in an environment to create a geometric interpretation of the scene. Sensors can be cameras, laser range-finders, sonar arrays, etc. Some possible geometric interpretations are exact reconstructions, projective geometry reconstructions, or occupancy/vacancy evidence grids.

In this thesis, we investigate an approach to creating a vacancy model of an active scene from any number of unique, static cameras. This is a common situation in static camera systems, including most human-computer interaction and surveillance systems. A model of the scene is of particular interest in these cases, because these systems are used to observe and perhaps to understand the actions of the participants in these environments.

Unfortunately, there is not enough information available at any instant to robustly model the scene without making strong assumptions. There are many cases where it is even impossible for humans to disambiguate a scene based on a single image. This was the motivation for looking into using tracking information to aid in reconstruction.

In fact, our method depends on the observation of the moving participants to create a model. It leverages tracking and visibility information into an evolving model of the open space in a scene. Essentially, it determines what areas of the space must be empty in order to match its observation of the moving object at a particular location. It does this by accumulating vacancy evidence within the volumes between the visible parts of the object and the center of projection of the corresponding camera. This models the unoccluded space from all available viewing angles. The complete method involves tracking objects in the available views, using the tracking information to determine the relationships between the views, determining the objects' relative position, and altering a 3D occupancy model of the environment to match the observations.

This method will be useful for tracking, scene understanding, and visualization applications. While the resulting model will not be an exact geometric representation of the scene, it will capture important functional properties of the scene. From this representation, it is possible to have expectations of which parts of moving objects should be visible or occluded, to determine which parts of the scene are visible in each camera, to determine entrances and exits to the scene, and also to visualize the environment.

A scene reconstruction system should not only be evaluated based on what types of applications it may facilitate, but also based on the circumstances under which it is effective. To be broadly applicable, an application should work well in different environments with as few assumptions and restrictions as possible. For instance, some systems require calibrated sensor systems, completely static scenes, reasonably similar views, no occlusion, and/or very few areas which are relatively homogeneous. Much work has been devoted to circumventing these assumptions to make more robust reconstruction systems. The number and complexity of these considerations helps to justify the vast quantity of research done in this area[3, 4, 6, 8, 9, 10, 19, 21, 23, 24, 28, 31].

To make more generally applicable systems, researchers have proposed increasing the number of views with multiple-camera arrays or moving acquisition systems, using active sensing systems, or increasing resolution to decrease local ambiguity. These all

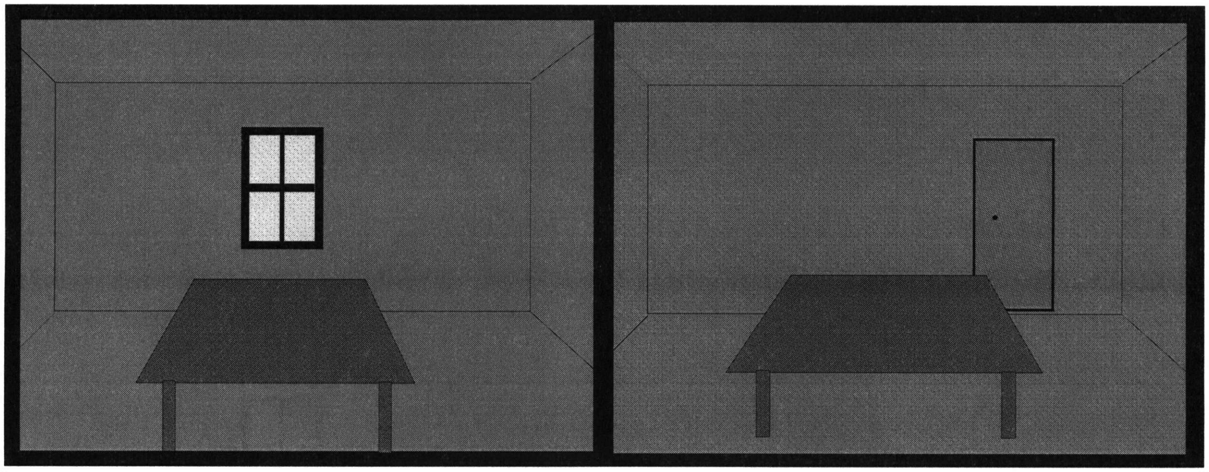


Figure 1-1: *This figure illustrates two views of a scene which are inherently ambiguous. Without additional information there is more than one interpretation of relative camera position.*

may result in improvements, but often it is not desirable to add or change equipment depending solely on the needs of one algorithm and often better equipment is either too expensive or not yet available.

The method described in this thesis attempts to create a camera-relative model of a scene based on how objects present in the scene over time. There are two obvious assumptions in this method. First, moving objects must be present within a static environment which can be localized. While requiring moving objects restricts use of this method, many of the applications of the resulting model are centered on understanding of the moving objects.

The second restriction is the necessity for observation over time. Having a model which is constantly updated based on current observations is also useful because most scenes are not completely static. If the model is no longer sufficient to explain the observations, the model can be altered to remove the contradiction.

While integrating information over time restricts use of this method to relatively static scenes with static cameras, it also adds robustness. For instance, certain problems are not solvable by human observers without the information gained over time. Figure 1-1 illustrates that in scenes which contain symmetries and don't contain ample distinguishing features between unique views, finding relative positions of static cameras can be an ill-posed problem. Even humans can find it difficult to calibrate two distinct views of a scene based on single views.

Our method stands to be more robust to unknown camera placements by integrating correspondences derived from tracking moving objects. Our method makes no assumptions about the similarity in appearance between the available views. It can integrate color, black and white, infrared, and other sensing systems. Also, partial occlusions will only hinder the system in cases where it greatly affects the localization of the object.

Chapter 2 outlines related work in reconstruction. Chapter 3 gives a general description of this method and its advantages and disadvantages. In Chapter 4, a single camera system is described followed by a discussion of the resulting representation. Chapter 5 covers the numerous considerations for extending to multiple camera systems. Chapter 6 discusses possible applications and extensions. The final chapter summarizes this work.

Chapter 2

Related Work

Visualization, recognition, object tracking, and object modeling applications have motivated much of the work in scene reconstruction. This chapter will cover some of the primary methods that are used to reconstruct scenes and efforts to overcome their weaknesses. This will include discussion of methods which model open space in an environment and methods which explicitly model the environment.

2.1 Modeling Open Space in an Environment

Reconstructing the open space in an environment is a problem that is of particular importance to mobile robot systems. Path planning, obstacle avoidance, and navigation in unknown and unstructured environments often requires a world model of which areas are occupied and which are unoccupied.

There are a number of methods of sensing range to the closest object in a direction. A survey provided by Jarvis[17] discusses some methods of obtaining range data, including striped lighting, grid lighting, depth from occlusion, texture gradient, focusing, some "shape from" methods, stereo, motion, triangulation, and time-of-flight range-finders.

To develop and maintain an effective world model, mobile robot platforms have been developed which integrate their depth readings from different positions to make more effective models of the open space[7, 33, 26]. Others have attempted to integrate occupancy evidence using multiple wide-angle stereoscopic images [21]. All these methods include some form of evidence grid. Each cell in the evidence grid accumulates either occupancy evidence, vacancy evidence, or both.

Although these methods bear a similarity to our method, they involve introducing a moving sensor into the environment. Our goal is to create a system which can use stationary, passive sensors(cameras) to model a scene based on activity within that scene without introducing strong restrictions on their placement.

2.2 Explicit Modeling of the Environment

The area of passive scene reconstruction has been dominated by the many variants of stereo[14, 9, 20, 24, 23, 28, 30, 31]. A survey of stereo methods can be found in [6].

Other methods exist for very limited domains. There are many "shape from" methods, including Shape from Shading[13] and Shape from Texture. These methods are generally not useful for scene reconstruction unless the scene can be considered a single surface with at least one property which varies with depth or orientation. In general, scenes have many discrete, non-uniform surfaces.

In general terms, stereo methods take different views of an environment, find correspondences in those views, determine how those views relate to each other, and use the correspondences and the relationships between the views to hypothesize the true positions of the correspondences. Arguably, the crucial step in stereo computation is establishing the correspondence between homologous features- features which are projections of the same physical identity in each view.

Regardless of which variant of stereo is used, correspondences can only be drawn for points in the scene which are visible in at least two different views. Therefore, to model a scene effectively, it would be necessary to have many, distinct views of a scene which show all points which need to be reconstructed with at least one redundancy. Researchers have tried to create more complete models by using structure from motion[3] or large arrays of cameras giving different views[31]. Since structure from motion usually necessitates introducing a moving acquisition system into the environment and processing many video streams at once presents other difficulties, these solutions add undesirable complexity.

Even assuming ample redundancy in the views, it is necessary to determine numerous, robust correspondences. Without numerous correspondences, the resulting reconstruction will be very sparse. Without robust correspondences, there will be significant errors in the relationships between the views and errors in the reprojection of the points. Most scenes have regions with little or no discernible features. Elements as common as walls, carpets, tabletops, sky, and small "windowed" regions often lack significant features. For these regions, an accurate depth cannot be determined.

Assuming the scene contains ample features to create a satisfying reconstruction, there still exists the difficulty of determining which are matching features. One solution to this problem is to manually define the correspondences in all the views and to define which of the points determine surfaces. Although this can result in very good models, the amount of work and knowledge necessary use this type of system undesirable.

If the feature determination and matching is not done by human inspection, it is necessary to choose a feature and define a measure of similarity. Lines, points, patches of the image, corners, and LEDs are just some of the examples of features which have been used. Regardless of the feature, mismatches occur.

Researchers have proposed more complex solutions to this problem. There are methods which use more than two cameras in an effort to reduce mismatches[28, 8, 31]. Additional views give additional constraints for calibrating the camera and make it possible to verify the correspondences based on their occurrence in many views.

Some systems use motion to find and track correspondences [20, 24, 30, 4, 3]. This includes moving acquisition systems and moving objects. The chance of a mismatched feature is reduced by tracking the features through the sequence of images. Relative camera position through time can be derived as a result of the fairly robust correspondences.

Stereo algorithms have difficulty whenever the views are very discrete, the views are not sufficiently redundant, or there are homogeneous regions. The efforts to reduce the errors resulting from these problems have mostly resulted in more intrusive, complex techniques. Our effort is directed towards creating a system which does not require extremely complex acquisition systems and gains robustness from observation of moving objects within the scene.

Chapter 3

Accumulated Line-of-Sight

This chapter outlines the general method with less attention to the specific details of any particular implementation. The next chapter gives an example of a single camera implementation. The following chapter discusses implementation issues of a multiple camera system.

3.1 The Method

The method described in this section is a very general framework for creating a system that can model its environment based on its observations of moving objects within the environment. It can be used with varying success in any system with any number of static cameras. Systems of multiple cameras with moving objects in their regions of overlap add significant robustness, but even rough single camera systems can be implemented, as shown in Chapter 4.

In its most general form, this method involves determining the moving objects, determining relative positions of the objects, updating the model, and displaying the model. This section discusses each step and the trade-offs inherent to different camera configurations. The desired result is a model which consistently explains the visibility or occlusion of objects in the environment. A single camera reconstruction will result in a camera relative projective model of the first occlusion along the line-of-sight of each pixel. The multiple camera reconstruction will result in a spatial model of

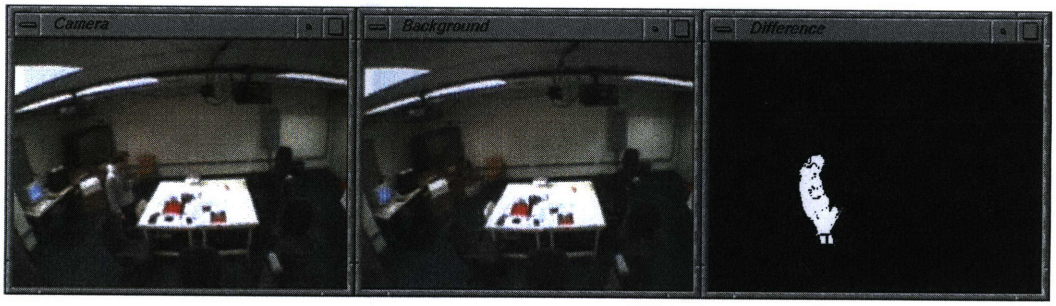


Figure 3-1: *An example of backgrounding showing (left) the current frame, (middle) the background, and (right) the thresholded difference between the background and current frame giving the pixels which are significantly different from the background. The human figure which is not part of the background is visible in the difference.*

vacancy regions which explain the visibility of objects to all cameras.

3.1.1 Determining Moving Objects

Finding unique, static features without specific knowledge of the scene is difficult, but finding unique, moving regions in static views can be accomplished with some reliability, Systems have been developed for object tracking based on edge detection[12, 15, 29], optical flow[2, 25, 27], corner detection[32], backgrounding[12, 22, 5], and information specific to a particular context[16]. Edge detection tracking and corner tracking generally involve locally determining the best match for an evolving model of the edges or corners are part of the moving object. Optical flow attempts to track regions of steady flow. Backgrounding refers to a general set of techniques which attempt to model the static background and use its difference from the current image to find the pixels which are not part of the background.

Because it is desirable to know which pixels are part of a moving object, the most effective method is backgrounding, or image subtraction. Backgrounding can involve taking a single image before any objects are introduced, averaging recent frames, or only updating a background pixel if the color has remained constant for a certain period of time. A more complex backgrounding method is used in the example described in the following chapter.

Once the background has been determined, the current frame is subtracted from the background giving an error signal which can be thresholded to produce a binary image in which 'on' pixels signify points at which the current frame significantly differs from the background(see Figure 3-1. Running some variant of connected components[14] on the binary image will give the regions which may represent moving objects.

This step is performed for every camera in the system. For each camera, a binary image similar to the thresholded difference image in Figure 3-1 is produced. The binary image is further processed to determine all connected regions of pixels. The connected regions of pixels which are not part of the background from each camera are used in the following steps.

3.1.2 Determining Relative Position of the Object

Depending on the method used to approximate the depths, some moving regions may have to be filtered out. It is more important to minimize false corresponding regions than to maximize true corresponding regions, because the location of mismatches could appear to be part of occupied space. This will adversely affect the model. Removing noisy components which do not correspond to objects, components which represent more than one object at different depths, and components caused by changes in the environment often results in more robust models.

The procedure of approximating the depth for each object is dependent on the camera position, number of cameras, types of occlusions, and timing of cameras. Some possibilities are triangulation in disparate views, stereo, or approximation from a characteristic of the object.

If only one view is available, it is necessary to approximate the position of the object by using a characteristic of the object that correlates to its approximate depth. Some characteristics which can correlate to depth are size, height, centroid, top, and bottom of the region. For instance, the depth of a sphere of known radius is approximately equal to $k/diameter$ or $k/(area)^{1/2}$ where k is defined by the camera parameters and the known radius. In other cases, localizing part of the object that is known to lie on a plane will give a better estimate. This is discussed further in the example in the next chapter.

With a single stereo pair, the depth of points in the object could be approximated by locally averaging over the depth of features within the region of movement. This would be more robust than simply using some approximation which is a function of a characteristic of the object in the image. Unfortunately, using a stereo pair with a small baseline can result in large discretizations in estimated depth while using a larger baseline can result in errors due to partial occlusions or mismatched features.

More robust results can be gained using triangulation with multiple cameras in very different positions. If a significant region of the object is visible to at least two of the views, the error in the resulting relative position should be on the order of a fraction of the size of the object. If a significant region of the object is not visible or there is another reason to believe that the localization may have error, the object can be neglected. This is discussed further in Chapter 5.

With a single camera system, the relative position of the object is represented by the depth of the object and the position of its centroid in the image. The position of the centroid in the image defines a unique ray along which the centroid of the object lies. The depth defines how far along the ray the centroid lies. With a multiple camera, the relative position of the object is the approximate three dimensional location of its centroid in space. Given the positions of the cameras, it is possible to determine the distance to the object to within a constant scale factor.

3.1.3 Updating the Model

The initial assumption of the algorithm is that the space surrounding the scene and the cameras is completely occupied. If no movement occurred, this hypothesis would remain valid. In this case, most visual routines could completely neglect this camera because it contains no information about moving objects or the environment.

As moving regions and their corresponding relative positions are observed, the model must be updated to match these observations. If according to the model an object should be occluded, but it is visible, that is evidence against the occlusions in the model. Eventually, the model will be complete enough to explain all the occlusions observed.

In a single view case, this model can be simplified by modeling a lower bound, d_{lb}, on the depth corresponding to the first occluder, d_o, along the line-of-sight of each pixel. It will always be the case that $d_{lb} < d_o$, because there can never be a visible object at that point which is exactly at or just farther than the depth of the occluder in the scene. Although, it will never be possible to determine the exact location, our algorithm will be relatively certain that it will not be closer than d_{lb}. This is the most complete model which can be achieved without the additional information of where an object is expected to be seen but is not visible. This simplification reduces the time and complexity of evolving the model. Further discussion can be found in the next chapter.

In the multiple view case, it is necessary to have all the cameras in one global reference frame. Other investigators have created systems which can calibrate cameras based on moving objects[1], so this will only be tersely handled here. Once all cameras are calibrated, it is possible to begin modeling the scene. This is discussed further in Chapter 5.

The single camera implementation produces a "depth" map which represents a lower bound of the first occlusion for each pixel. The multiple camera implementation produces a three dimensional evidence grid which can be processed to produce a model of vacant and occupied space.

3.1.4 Displaying the Model

For a single view reconstruction, the model has been simplified as a lower bound to the depth of the first occlusion along the line-of-sight of each pixel. This depth map is similar to the output of many stereo systems. It can be displayed by projecting each point in a 3-D triangular mesh outward by the corresponding distance in the depth map using a simple projection model. The current background image can be used to texture map the 3-D mesh to aid in understanding the model. This is illustrated further in Chapter 4.

For a multiple view reconstruction, the representation is a three dimensional evidence grid. It is possible to display the evidence grid as a dense volume of small polygons between adjacent vertices with transparency values derived from the vacancy probabilities of each of their vertices. Unfortunately, it is difficult to display due to hardware limitations. Small gains can be won by removing polygons which are almost transparent.

An alternative is to model the space discretely as either the open space or occluding volumes. A simple method for displaying the open space or occluding volumes is to threshold the vacancy evidence grid above or below a certain value, respectively. The resulting voxels can be displayed as a cubes centered at the remaining voxel positions.

To reduce the aliasing of this method, a more exact approximation of the surface could be determined by stitching together vertices derived by linearly interpolating to find the approximate location of the epi-surface between each adjacent voxel which are not both vacant or occupied. A more thorough coverage of three dimensional rendering techniques can be found in the medical literature[18].

3.2 Advantages and Disadvantages

Many of the characteristics of this modeling technique can be considered either an advantage or a disadvantage. This section will outline a number of considerations and state under which circumstances they could be considered an advantage or a disadvantage.

3.2.1 Dependency on Correspondences

This method depends on correspondences of moving objects not homologous features in the scene. Whereas most every camera pair has homologous features, not all cameras have moving objects. Without moving objects this method will produce nothing. Without sufficient homologous features, other methods will also fail to robustly produce models.

But in a scene with moving objects, virtually unlimited correspondences can be found. If there are not sufficient correspondences at time, t, to achieve relative certainty in camera placement, the process can wait until more correspondences are found. This may make this method more robust to arbitrary camera placements and scene with relatively few features.

With the correspondences gained over a period of time, there should be a capability of calbrating cameras which are only partially overlapping views, cameras which are very widely placed, and even sets of cameras whose views do not necessarily overlap. These possibilities open up many new and interesting questions which are discussed further in Chapter 5.

3.2.2 What is Being Modeled?

Most modeling methods attempt to model the exact location of all visible elements of the scene. If this can be achieved effectively, the scene can be reproduced from any position. Our method models the open space in the scene. With many cameras, this will approximate an effective model of the scene.

Even with a single camera the depth map has some very desirable properties. The depth map gives a lower bound on the depth to every pixel, not only the pixels which have significant features. Given a moving region and a corresponding depth, it will be possible to determine if it is adjacent to an occluding contour for that depth. If it is adjacent to an occluding contour, it is probable that the object is not entirely in view and should be analyzed more specutively.

Chapter 4

Single Camera Reconstruction

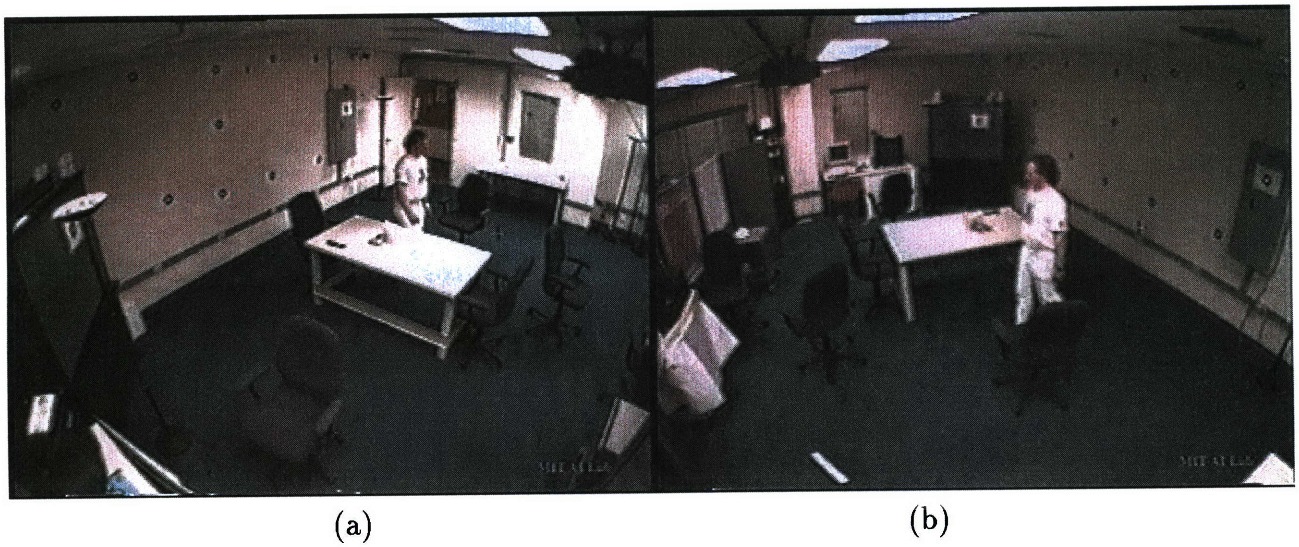
The goal of reconstruction from a single view is to create a dense representation of the depth of the visible parts of the scene relative to that view. This representation should determine a depth for each pixel, not only for distinguishable features. Our representation is a lower bound on the distance to what a pixel represents in a scene.

This particular implementation was motivated by difficulties which arose from efforts in the Human Computer Interface(HCI) Room at the MIT AI lab. The HCI Room is an attempt to create an environment which allows groups of people to interact with the computer unhindered by traditional keyboards, displays, or mice. There are two cameras involved in the tracking and visual interpretation of the rooms' participants. Figure 4-1 shows an example image from both cameras.

Because the HCI room contains tables, chairs, windows, doors and other occluding boundaries, simple methods for tracking, localizing, and analyzing the actions of participants can be confused. Since the location of elements of the environment are not available, the room must be capable of modeling the environment to determine where potentially confusing elements of the environment are located.

4.1 The method

The method described below involves segmenting moving objects from the background, filtering objects which can't be localized, approximating the depth of the


(a)

(b)

Figure 4-1: *A typical view from the (a) left and (b) right camera in the HCI room. Each illustrates the difficulties in tracking and analyzing participant's actions.*

remaining objects, updating the representation, and displaying the resulting model.

More specifically, the segmentation involves a backgrounding procedure which includes modeling the most probable appearance of the background based on the statistical properties of the pixels. Of the objects which are different from the background, some can be localized and others can't. Because our depth approximation procedure assumes that the object is a standing human, it is necessary to filter out objects which may not be standing humans. Once the silhouettes of standing humans can be determined over time, it is necessary to make an assumption to determine their approximate depths. Finally, the silhouettes and depths are used to update the model.

The output of the single camera reconstruction is a projective model of the scene build only from the information available in one camera. Figure 4-2 shows the models resulting from two independent processes using one of the cameras available in the HCI room.

4.1.1 Segmenting Moving Objects

Because robust, updatable backgrounding is desired, standard backgrounding methods were insufficient. Using a single background image taken before object are in-

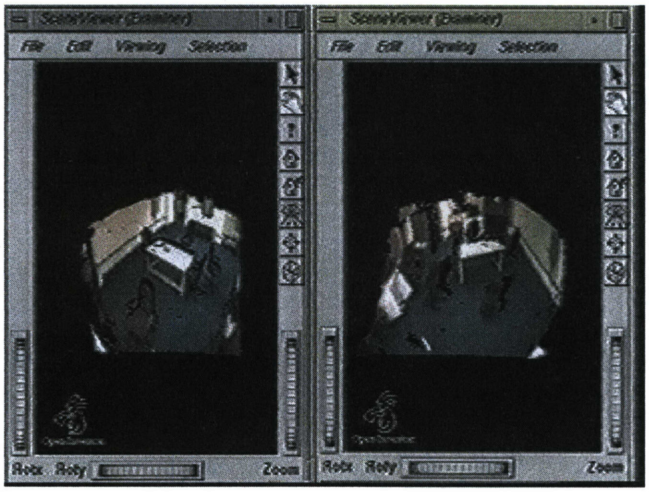


Figure 4-2: *The reconstruction from both cameras.*

troduced is undesirable because any changes in the environment will accumulate and eventually cause serious errors. For instance, shuffled papers, a moved coffee cup, or a jacket left in the corner will introduce connected components which will continue to exist unless the scene is returned to the exact initial condition. While they exist these falsely detected moving components will be incorporated into any object which comes into contact with them causing errors in the representation.

There also exist many dynamic methods of backgrounding which involve maintaining an evolving background. Two simple methods are averaging recent frames and adopting a new pixel from the current image if that pixel has been different from the background for a significant number of frames. Unfortunately, both these methods suffer from moving objects significantly affecting the appearance of the background. This can leave significant after-images from objects which are stationary for long enough to affect the background which results in regions which do not correspond to true objects.

In an effort to keep moving human figures from altering the background image, a new heuristic was adopted. This heuristic is that if the recent values of a particular pixel are modeled by a number of Gaussian distributions, the most probable model of the background pixel is the Gaussian distribution which has occurred more frequently and varied less in recent history. With this heuristic, people in the scene can only become part of the background if they stay in one position and the pixels which represent them vary less than the background at that point for a significant amount of time. Of course, static rigid objects can more easily become part of the background.

The backgrounding procedure involves modeling the N most probable Gaussian distributions which best explain the recently observed values of each pixel. N must be large enough that models with high probability are generally not lost. If N is too high, it becomes prohibitively expensive to maintain the models. In this example, N was chosen to be 5. The Gaussian model includes a mean, variance, and frequency statistic which are maintained for each stored value.

Qualitatively, a model is more probable if it has occurred frequently enough and did not vary significantly in the recent past. For example, the most probable background model for a pixel which represents a blue wall in a static scene will be a Gaussian centered at "blue" with a small variance and a large occurrence value, because often the pixel is blue and it varies only as much as the noise in the sensor. If the pixel turns green as a result of a person with a green shirt entering the scene, the background will not adopt the green model unless the person can remain in the same position for enough time to have the model's frequency statistic outweigh its larger variance.

Initially, the N models are given a frequency of zero which makes all of them improbable and they are quickly replaced by the first values which appear. As their statistics are altered, they are ordered with the most probable background model first.

For every frame, each pixel is checked against the N stored pixel models in order of their probability until a match occurs. A match of the image pixel, $I_{i,j}$, to the corresponding pixels' m*th* model is defined as:

$$Match(i,j,m) = \begin{cases} 1 & \text{if } (I_{i,j} - \mu^m_{i,j})^2 < k\sigma^m_{i,j} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

where $I_{i,j}$ is the value of the pixel at the position (i,j), $\mu^m_{i,j}$ is the stored mean value for the m*th* model for that pixel, $\sigma^m_{i,j}$ is its stored variance, and k is a constant which defines a relative threshold for the pixel error. This adjusts the threshold on a model by model(pixel by pixel) basis which accounts for different surfaces which vary different amounts. For instance, because lighter surfaces, monitors, and light sources have greater variation than dark, shadowed, and "washed out" regions in video, dark regions of the room should be more sensitive to pixel variations.

Depending on the type of camera and environment, the pixel value and pixel difference can be defined differently. Our color cameras were in a fairly controlled environment with good lighting and few shadows, so a simple RGB representation was sufficient. Our difference was defined as the length of the vector of differences. But HSV, normalized RGB, or some other mapping may produce better results in certain situations.

If no match occurs, the least probable stored pixel model is replaced by the current color with the default variance and frequency characteristics. If a match occurs, the matched model is updated as follows:

$$\mu^{m\,\prime}_{i,j} = (1-\alpha_1) * \mu^m_{i,j} + (\alpha_1) * I_{i,j} \tag{4.2}$$

$$\sigma^{m\,\prime}_{i,j} = (1-\alpha_2) * \sigma^m_{i,j} + (\alpha_1) * (\mu^m_{i,j} - I_{i,j})^2 \tag{4.3}$$

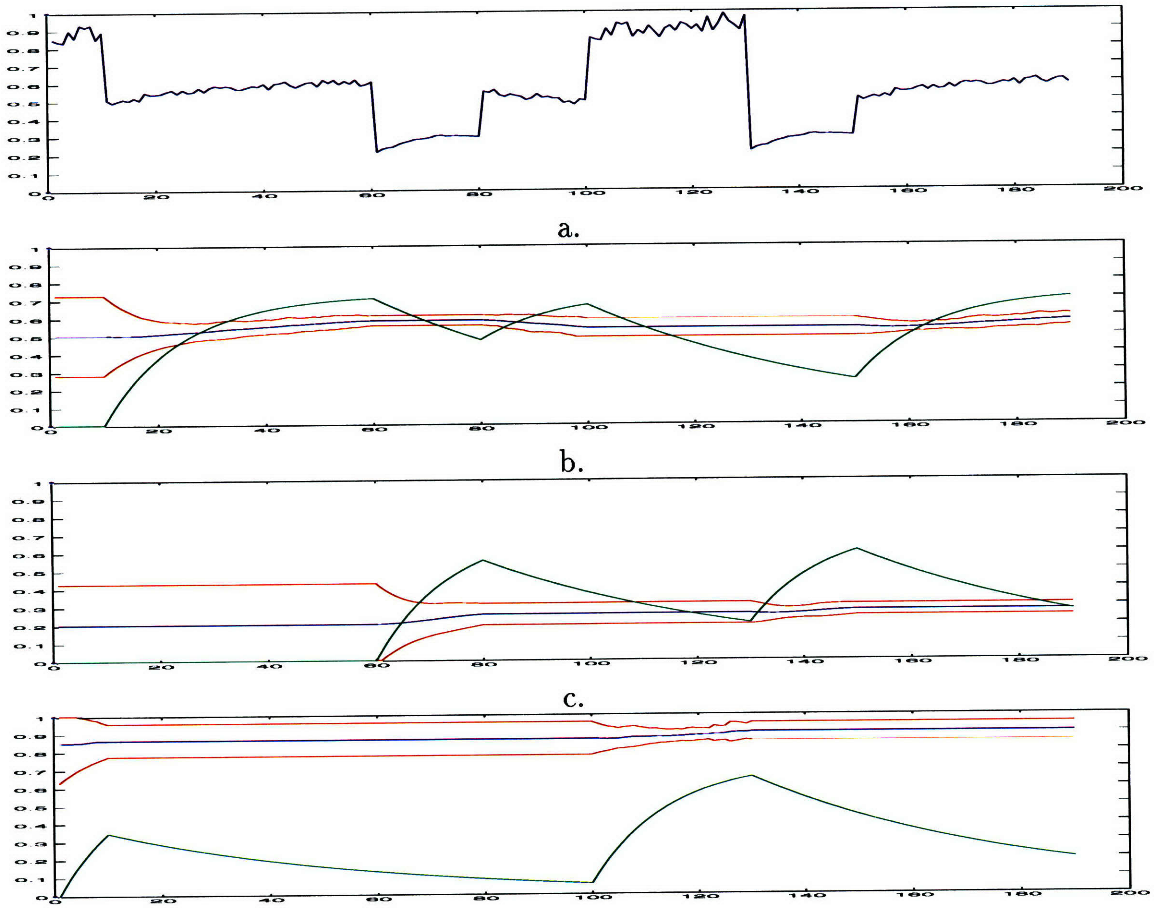
$$f^{m\,\prime}_{i,j} = (1-\alpha_{o1}) * f^m_{i,j} + \alpha_{o1} \tag{4.4}$$

where $f^m_{i,j}$ is its occurrence probability and α_i is the learning values for each function. If $\alpha_1 = \alpha_2 = 0$, the model for each stored value will be static with initial value and default variance. If $\alpha_1 \neq 0$, the value will track long term trends in the background color. With $\alpha_2 \neq 0$, σ will approximate the variance of the samples which are classified in the mth model.

After all the models which matched have been updated, the occurrence frequency of all models is decreased:

$$f^{m\,\prime}_{i,j} = (1-\alpha_{o2}) * f^m_{i,j} \tag{4.5}$$

where α_{o2} determines how fast the frequency statistic decreases. As a model is seen



a.

b.

c.

d.

Figure 4-3: *a. An series of pixel values as received over time. b-d. Three dynamic, Gaussian models of the pixel values showing the mean(blue), range of values(red), and occurrence frequency(green).*

less, its occurrence frequency decreases slowly. Eventually, a model will become improbable because no example of that model has occurred.

A simple example of a three model system is shown in Figure 4-3. When an example of a particular model is observed, its frequency increases and its mean and variance are altered accordingly. When no example for a particular model is observed, its mean and variance remain constant, but its frequency decreases. In the sequence, the top model is the most probable for this sequence because its variance is smallest and it occurs most often. The second model has a noticeably higher variance and the third model does not occur often, making them less probable.

Figure 4-4 shows the execution of the backgrounding method in the room. The

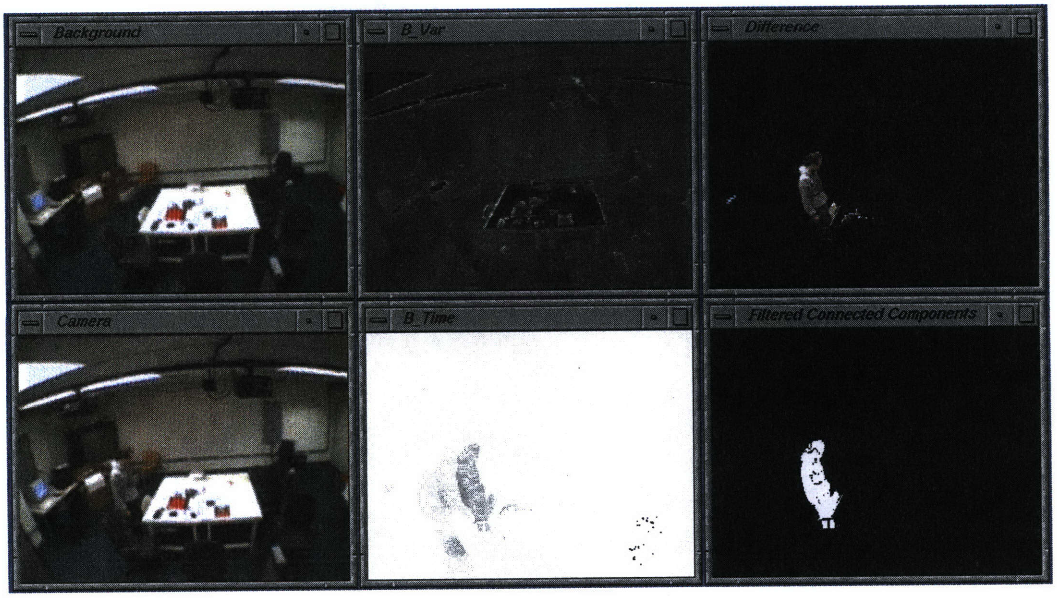


Figure 4-4: *(left column) The current background and the current image. (middle column) The variance and frequency for the model of each pixel in the background. Lighter pixels in the variance image correspond to pixels which vary more. Lighter pixels in the frequency image correspond to pixels which have more consistently occurred in the recent past. (right column) The binary image of pixels which are not part of the background and the filtered connected components.*

background appears identical to the current image except that it does not include the person. The background image is also a time-filtered version of the most probable background. The variance is higher in brighter regions (including the monitor), except the areas which are "washed out." Unfortunately, despite the higher variance in the region of the monitor, there is still a small residue from the flicker. The frequency is slightly decreased where the person is standing, but will quickly recover when the person moves from that area. The filtered component includes the entire region in which the person is visible, including through the arm of the chair. That often allows depths to be determined even through windowed regions in the scene (under tables, chair legs, etc.).

Using adaptive pixel models allows this procedure to adapt while limiting the chance that moving objects will be included in the model of the background. The ability to adapt adds robustness to long term changes in the environment. The backgrounding heuristic helps limit the number of falsely detected objects due to an incorrect background model or noisy regions in the image.

The output of this algorithm is a binary image which represents the pixels that are not part of the background. This binary image is "on" where the current pixel did not match with the most probable model and is "off" where the current pixel matched to most probable model. We used connected components[14] to segment the binary images into connected regions.

4.1.2 Approximating the Objects' Depths

Reconstruction using a single camera necessitates making an assumption. In the past, single camera reconstructions have assumed: the scene is made of a single, uniform, Lambertian surface; the spatial characteristics of texture change as their depths change; or objects can be recognized and their relative sizes can be used to determine relative depths. Another means is to determine a characteristic which correlates with depth.

With relatively rigid objects in an environment with few occlusions, the most obvious characteristics which correlate with distance are the area of projection, height, or width. Unfortunately, humans are very deformable and are often partially occluded to the camera, so these are not good indicators of depth. The assumption that our single camera method uses is that it is possible to detect a single, standing, human figure and to approximate its depth based on the position of the top of the head in the image.

Because our method assumes that the connected component is a standing human, it is necessary to apply filters to remove regions which may not be localized or may not correspond to moving objects. Although the backgrounding procedure limits the number of falsely detected objects, there are still connected regions which do not correspond to real objects. The first step filters objects based on their size. Objects less than 5% or greater than 200% the size of an average human are neglected. Objects which are near the ceiling in a region above the highest possible head position are also removed. Figure 4-4 shows an example where reflections and monitor flicker connected components have been filtered out.

The remaining objects are localized by assuming that the top of their head lies near a plane which is parallel to the floor. As the point on the plane moves up in the image, the object is assumed to be farther from the camera. Using this approximator, the distance to the object can be approximated by

$$distance \propto \frac{1}{Y_{top} - Y_{horizon}} \tag{4.6}$$

where Y_{top} is the verticle position of the topmost pixel of the person and $Y_{horizon}$ is a horizon line for the top of the head at infinite distance which is determined by experimentation.

This distance may be underestimated in the case of people sitting, bending over, or being particularly short, but it should not be over estimated. Underestimation does little to alter the model because we are modeling the lower bound of the depth of the first occlusion, while overestimation can cause depths to be greater than their actual distance. The distance metric must be defined with respect to the tallest person which enters the environment. The difference in height of the people can cause errors in the distance estimates resulting in shorter people appearing to not touch the floor. In the case of this experiment, a difference of 2 inches in height for a person in the middle of the room would result in approximately 6 inches error in depth.

4.1.3 Updating the Depth Map

As stated in the previous chapter, the initial assumption is that the space surrounding the scene and cameras is completely occupied. In our simplified model, this corresponds to setting the lower bound for the distance to the first occlusion for each pixel to zero. Because we are modeling a lower bound, the initial condition has no information about the scene.

Once the connected regions and their corresponding depths have been determined, it is possible to test whether the current scene model contradicts our current observations. For example, if a person is standing 10 feet from the camera and is behind an occlusion which is at 8 feet(e.g. a table), the region where that person is detected

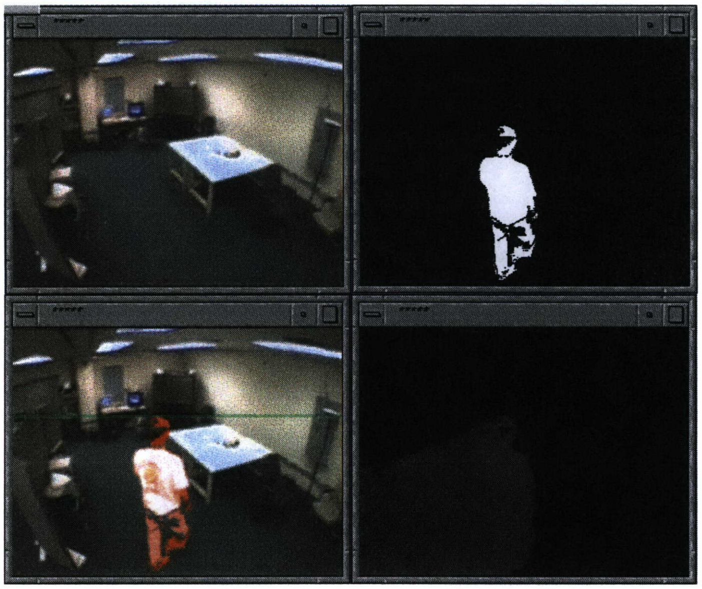


Figure 4-5: *Output of the program run in the example. (upper left) the current background image. (lower left) the current frame. (upper right) the connected component which remained after filtering. (lower right) the current depth map in a very early stage of development. As the depth map is refined, it nears the approximate depth of the scene.*

should not include the pixels which represent the table, otherwise the person would have to be in front of the table.

We model a lower bound on the first occlusion along the line-of-sight of a pixel. If that lower bound, $d_{occlusion}$, is closer than an object visible in that pixel(i.e. $d_{occlusion} < d_{object}$) there is a contradiction. Therefore, $d_{occlusion} >= d_{object}$.

Figure 4-5 shows the output of the program at a very early stage. Almost nothing is known about the depth of the scene for most of the pixels, except along the path were the person has recently appeared. Figure 4-7 shows a more developed model.

4.1.4 Displaying the Model

At any point, the model can be displayed as a 3-D projective surface. This surface is a composed of tiling polygons with vertices at adjacent pixel positions. Two viable tilings are square tilings and triangular tilings. Square tilings require more complex graphics capabilities, because the four vertices are not necessarily coplanar. Because of the number of polygons necessary, our implementation used a triangle tiling which

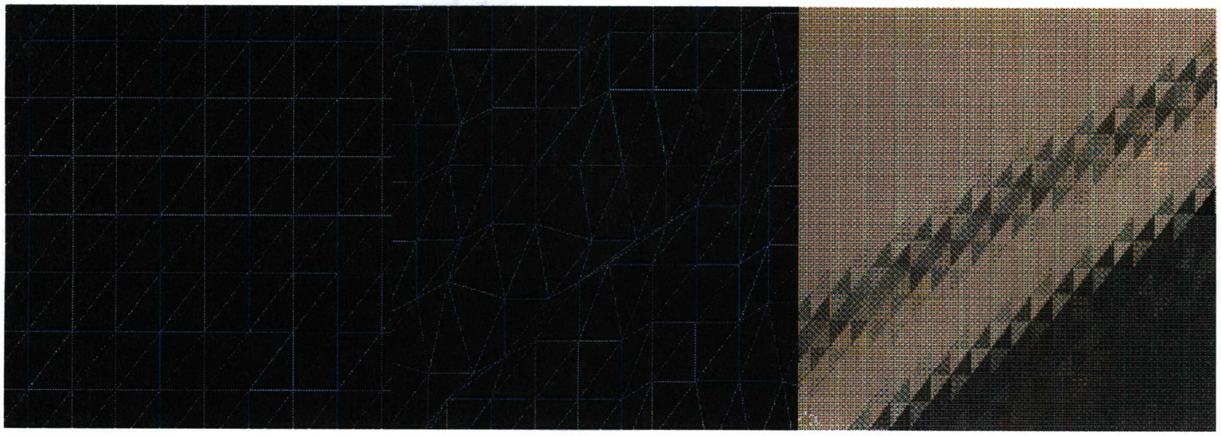


Figure 4-6: *A patch of a flat triangular mesh surface, a projected mesh surface, and a texture-mapped surface*

can be rendered in real-time.

The mesh surface connects adjacent points on the scene model. The location of these points is determined by projecting a point corresponding to each pixel a distance which corresponds to the distance for that pixel using a standard projective model. After projecting the mesh it can be texture-mapped with the background model by giving the entire polygon the average color of its vertices. Figure 4-6 shows this process for a small patch of the model. With an efficient graphics engine, the individual polygons could be texture-mapped which would reduce the aliasing evident in Figure 4-6.

Finally, polygons for which there is no depth information or which correspond to large depth changes which occur near occluding boundaries are removed for visualization purposes. Figure 4-7 shows a depth map and the corresponding 3-D projective surface model.

4.2 Results

The same process was run on the two available cameras in the HCI room resulting in two projective reconstructions from each camera. For this experiment, a person walked around the room for 5 minutes without changing the environment.

Figure 4-5 shows the output of the program running on the left as it maintained

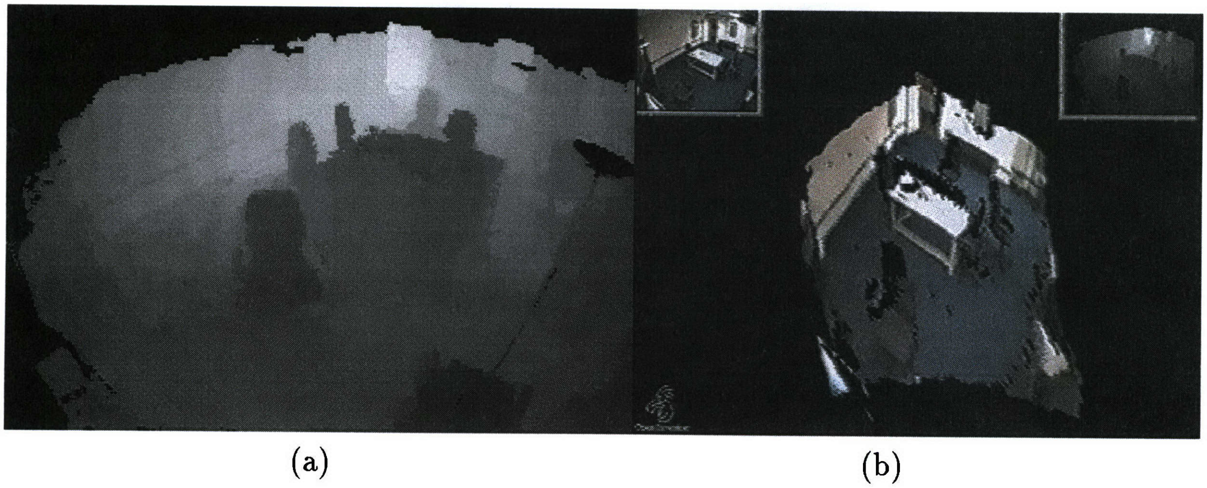

(a) (b)

Figure 4-7: *(a) The depth map from our example (b) The final output which shows the background used for texture, the depth map used for reprojection, and the resulting model. (center) the resulting model in 3-D.*

a background, segmented moving objects, filtered objects whose depth could not be approximated, approximated the depth of the remaining objects, and updated the model accordingly.

The model appears to have captured many important aspects of the environment. Occluding contours are very pronounced. Although the furniture is very salient, there is little depth information within the furniture. They appear somewhat flat in this single camera model.

The walls and floors are smooth and flat. The general shape of the scene is correct. The doors and windows are at visibly different depths than the wall they are on, despite the fact that there are few features in those regions. Because no one has been located on the ceiling or in front of the standing halogen lamp, no depth information has been gained there.

Looking at the average depth of furniture in the room is gives a more quantitative measure of the results. Table 4.1 shows the approximate depths of the object relative to the average depth of the corresponding region(Figure 4-8) in the model.

In the region of the standing halogen lamp, the depth remains 0 because the upper part of the lamp has not been occluded. The other regions are relatively uniform, because the space has been defined by the depth of the person who has occluded

Table 4.1: Average Depths of Regions Compared to True Depths

Region Label	Region	Approx. True Depth	Normalized Model Depth($d_{i,j}$)	Corrected Model Depth ($k * d_{i,j}$)	Ratio
A	halogen lamp	6'	- (zero)	-	*undef*
B	chair near halogen lamp	5'	.145	4.8'	.962
C	second chair	10'	.306	10.1'	1.014
D	table and surrounding chairs	14'	.431	14.3'	1.021
E	farthest chair	20'	.639	21.2'	1.059
F	back wall near door	25'	.784	26.0'	1.040
G	hallway outside the door	29'	.910	30.1'	1.040
H	back wall near window	22'	.628	20.8'	.945
I	hallway outside the window	26'	.733	24.3'	.935

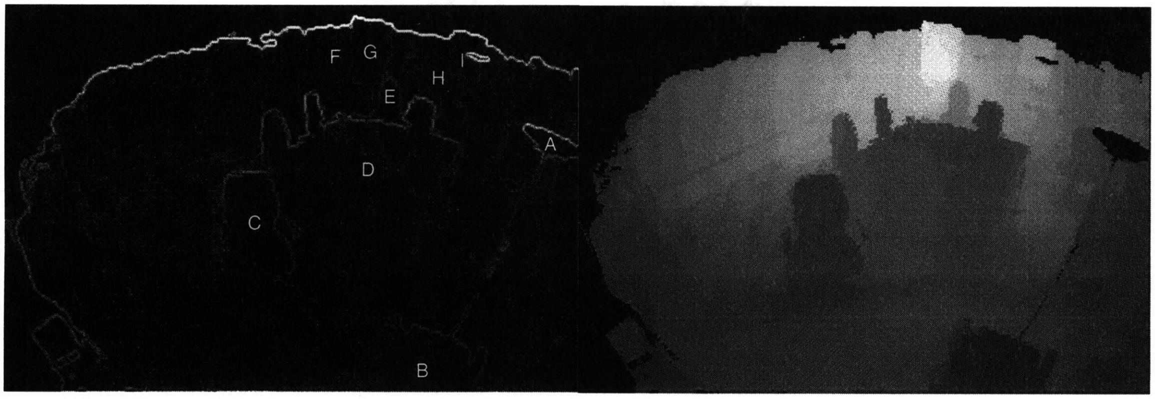


Figure 4-8: *The regions used to evaluate depth map.*

them. A comparison of the average depth of the model within these regions and the approximate average depth to visible regions of the model gives a more quantitative feel for the correctness of the resulting model.

In general, although the structure of the individual elements of the scene is not present, the general structure of the entire scene is evident. For a single camera reconstruction, these results are quite surprising.

Chapter 5

Multiple Camera Reconstruction

This chapter discusses the issues of implementing a self-calibrating multiple camera system which can automatically reconstruct a scene based on movement within the scene. While this method is more complex than the single camera model, it is not dependent upon making an assumption which greatly restricts is use.

It assumes only that there are multiple, similar, synchronized cameras which are directed towards a certain region in space in which objects move and are partially visible in at least two cameras. It does not assume that the cameras have a certain distribution in the scene (small displacement, small rotations, etc.) or that the scene has certain visual or structural properties (large scale textures, many objects, etc.).

5.1 The Method

As described in Chapter 3, this process involves determining the visibility of moving objects, localizing the moving objects, updating the model, and displaying the model. With multiple synchronized cameras, some these steps are significantly different than the single camera implementation discussed in Chapter 4.

The goal of this methods is to leverage visibility of objects in multiple views into a three dimensional model of open space in a scene. With multiple cameras, this model can potentially be very complete.

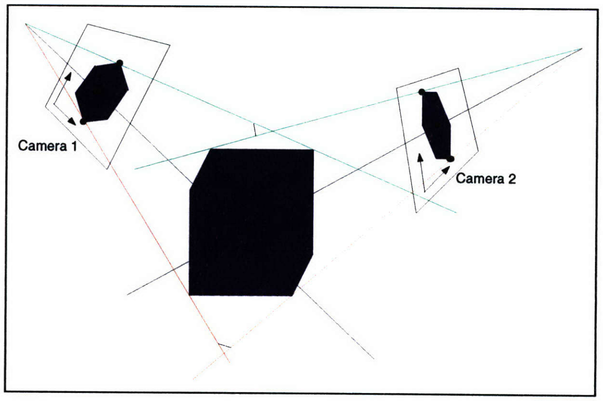


Figure 5-1: *Two cameras and the corresponding rays for the uppermost point, lowermost point, and centroid of the object in the image. Note that there is no direct correspondence between the highest point in one image and the highest point in the other. If this is not apparent, imagine Camera 2 undergoing a 180 degree in plane rotation. This switches the uppermost and lowermost point.*

5.1.1 Determining the Visibility of Moving Objects

Determining the visibility of moving objects with multiple cameras is accomplished by the same method as a single camera reconstruction. This is outlined in detail in Chapter 4.

In addition to a binary mask of the moving regions, the moving regions can be processed to determine properties which may be useful in determining correspondences, determining the reliability of the correspondences, or localizing parts of the object. For instance, the centroid of the object, the size of the object, whether the object is adjacent to an occluding boundary, and a description of the object(e.g. color histogram, first moments, average brightness, clustered color regions and locations, speed, direction, etc.).

5.1.2 Localizing the Moving Objects

Localizing the moving object involves finding correspondences in multiple views, using the correspondences to calibrate the cameras, and using the calibrated cameras to determine the relative position of the cameras on the objects. After calibration, it is possible to iterate this process by removing obvious outliers from the correspondences and repeating the process.

Establishing Correspondences

Once the visibility is determined, it is necessary to localize a feature of the object in at least two views. Unfortunately, the top, bottom, left side, or right side of an object in different cameras do not correspond to the same point in space. For example, with two views of a box(see Figure 5-1), the top pixel in both views corresponds to different corners, as does the bottom pixel. The left and right edges will also correspond to different edges in most cases. Because these optical rays do not correspond to the same feature in space, they would produce poor camera calibration and overestimates and underestimates of depth.

Without explicit knowledge of the camera placement the most consistent feature is the centroid of the pixels in the perspective projection of the object. This is obvious when looking at a roughly spherical object in N views. This approximation is usually accurate except in cases of shapes with unusual projections and occlusions.

Without occlusion, the centroid of the projection of the most objects should closely correspond to the centroid of the true object. This will not be true for all cases, but it is the best method of establishing correspondences. As an extreme example, a sailboat from one angle may appear to be a boat with only a mast but from another angle may have a full sail. In this case, the ray from the centroid in the first image centroid will pass close the center of mass of the boat and mass while the other will pass nearer the center of mass of the sail. These cases are relatively rare and can be detected as poor correspondences once a rough camera calibration is determined. Using the centroid results in relatively small error for objects such as people, cars, trucks, push carts, and animals.

With occlusion, the centroid of the visible regions of the object should be calculated within approximately the maximum dimension of the object. The difficulties resulting from occlusion are similar to the the difficulties resulting from shapes which have very different perspective projections. The reliability calculated after rough camera calibration has been determined will also be relatively low for large occlusions.

Given a single moving object in the scene, correspondences of moving regions are easily determined. With multiple moving objects in a scene, there is the additional a problem of determining which objects correspond to which in each view. Depending on what type of cameras are available, additional information can be derived from the images of the objects which can aid in disambiguating matches. This process would involve determining a description of the object and finding a distance measure for that description. Some possible descriptors are: color histogram, moment ratios, average brightness, average motion, "speed", and clustered color regions. These measures may help to determine a set of correspondences that are more likely to be true correspondences which will be extremely useful in robustly calibrating the cameras.

Calibrating the Cameras

Researchers are currently investigating methods which require only point correspondences that can be used to find not only the epipolar geometry of a set of cameras but estimates of the internal parameters and relative motion between views[19, 10]. Since we do not have a priori knowledge of the scene, but do have many correspondences, these methods are ideal for parameterizing the cameras.

Using one of these methods, we can determine both the intrinsic and extrinsic parameters for the cameras up to a similarity transform. Because some of the correspondences will be affected by occlusions or unusual projections, the camera model can be refined by by removing or devaluing the correspondences which are outliers.

5.1.3 Updating the Model

Once the cameras have been calibrated, the three dimensional evidence grid must be updated to explain the visibility of the objects. This process involves verifying the correspondences, determining relative position of the objects, and accumulating vacancy evidence.

Verifying correspondences involves attempting to determine whether features in a set of images correspond to the same homologous feature- in our case, a moving object- and further that they correspond well enough to effectively localize that object. The error in the correspondences can be approximated by the distance between the lines of projection of the corresponding centroids. If the centroids correspond to a single point in space, the projection rays of the centroids intersect in space, resulting in no error. Using multiple views, the false correspondences can be reduced.

For the consistent correspondences, the approximate position can be determined by finding the closest point to the line of projection of the centroids. The relative depth of the object is the distance between the camera and the object. If the correspondence has no error, that point is the intersection of the line of projection of the centroids.

To update the model, we can trace a the voxels that make up a line which passes from the camera in the direction of each visible pixel to a depth of the corresponding object. At each voxel, the vacancy evidence is increased. At any time this vacancy evidence grid may be visualized.

5.1.4 Displaying the Model

It is possible to display the evidence grid as a dense volume of small polygons between adjacent vertices with transparency values derived from the vacancy probabilities of each of their vertices. Unfortunately, it is difficult to display large numbers of semi-transparent polygons due to hardware limitations. Small gains in computability can be won by removing polygons which are almost completely transparent.

An alternative is to model the space discretely as either the open space or occluding volumes. A simple method for displaying the open space or occluding volumes is to threshold the vacancy evidence grid above or below a certain value, respectively. The voxels that have sufficient vacancy or occupancy evidence can be displayed as a cubes centered at the remaining voxel positions.

To reduce the aliasing of this method, a more exact approximation of the surface could be determined by stitching together vertices derived by a local, linear interpolation that approximates the location of the epi-surface between each adjacent voxel which are not both vacant or occupied. There are many variants of this in medical imaging literature. An example of such a method is the Marching Cubes Algorithm[18].

Chapter 6

Future Work and Extensions

6.1 Future Work

It is possible to overcome some of the limitations resulting from assumptions that were used to implement this method. Often surveillance systems have cameras which pivot. Many scenes have structural elements which move or change over time. The depth assumption made in the single camera model limits it use. This section discusses how to overcome some of the limitations resulting from those assumptions.

6.1.1 Incorporating Pivoting Cameras

Pivoting cameras necessitate a more complex backgrounding scheme. Given a camera which pivots around its optical center, it is still possible to do backgrounding. The process would involve determining the colineation and radial distortion parameters which bring the current image into correspondence with a background image that is a mosaic of all the viewing angles.

Once this colineation has been determined, one could interpolate the values for the background pixels from the current image. These values would be used to update the Gaussian models and to determine moving regions as discussed in Chapter 4.

There is the additional consideration that at one time, only one window of the background for that camera is visible. This could easily be modeled as a moving occlusion mask which is overlaid onto the background. This mask would allow the system to deal with objects that are moving off camera, but which would be visible at other camera angles.

6.1.2 Dynamic Scene Modeling

Unfortunately, most scenes are not completely static. Walls can be added or removed. Trees can grow leaves or fall down. Statues and buildings can be erected. Our model is capable of determining new areas which are vacant, because it constantly updates itself as necessary. Unfortunately, there is no simple mechanism for determining new occlusions.

Three methods could be implemented which would aid in updating the model in these situations. First, the depths could be occasionally reset to zero. The model would have to be rebuilt. The old model could be used until the new model develops.

Second, the depth along a particular pixels' line-of-sight can be linked to the background model for that pixel. For instance, the single camera implementation could include the approximate lower bound on the depth of the first occlusion in its Gaussian model for the background pixel. When a background pixel changes to a new value, the depth can be reset. If an object was moved to a new location, as the new pixels of a object are incorporated into the background, they will be reset to zero and the modeling will continue.

The third possibility is to decrease the vacancy evidence grid values slowly as the model is developing. In this manner, occasional mismatches and other errors will eventually be "forgotten." This will also result in errors resulting from less frequently occupied regions of space losing their vacancy evidence over time.

6.1.3 Single View Reconstruction with Stereo

The assumption used in the single camera implementation limits the use of that system to particular types of environments. If the single camera was replaced by a stereo rig, a more robust method for determining the depth of an object could be used.

The same backgrounding method could be used to determine which regions correspond to moving objects. A stereo algorithm could be used only on those regions to determine the approximate depth of features within those regions. Because occlusions of the moving regions will not be part of the moving region, only objects' self-occlusions will cause ordering problems to search for matching features.

Once the depth of the features in the region have been determined, the depth of the entire region can be approximated by locally averaging the depth of the closest features in that image. If the matches in a particular region are not consistent, the higher variance of the features in the region can be used to determine that the local average is not an effective approximation of the depth.

In this manner, any object could be used to carve out space- not only space that can be occupied by a human standing in an environment. A book thrown on the table or a hand reaching behind a desk could be used as effectively as a moving human figure. Using a stereo rig to approximate the depth of an object could greatly increase the applicability of this system.

6.2 Extensions

Some applications that this method can facilitate are different than applications of standard reconstruction methods. While it can be useful for visualization applications, it is more useful for applications in which issues of occlusion and robust scene models are more important.

6.2.1 Automatic Event Extraction

An effective model of the scene may facilitate analysis of the actions of the participants in the environment. For instance, the operator of a surveillance system may want to be warned dangerous conditions or things which have not occurred before. He could specify that he would like to be notified if an object passes through a certain door or if a large object is approaching the area. The computer could also model the actions it has observed in the past and tell the operator when something new or unusual occurs(e.g. a person falling out a window, a truck on the White House lawn, etc.).

With a three dimensional model or a projective model of the scene, it is possible to determine features in an environment. For instance, fitting planes to the projective model in our single camera example can produce a reasonable segmentation of the scene into structural elements(walls, chairs, tables, lamps, doors, windows, etc.).

The position of these features in space can be used to define spaces objects can occupy. When an object is being tracked, it's path can be stored as a list of regions it occupied as it passed through the room. This greatly reduced representation of tracking data can be used to learn activity patterns or to define conditions under which the operator should be warned.

6.2.2 Compression and Synthetic Movies

Video surveillance systems generally have very large amounts of video and long periods in which few events occur. This makes them a prime candidate for compression. Any standard of video compression could result in large compression ratios.

Compressing the background and the moving objects separately could increase the compression ratio even more. Because this method produces a model of which pixels occlude which regions of space, a moving object could be stored as a bounding box, a depth, and an image with a transparency mask.

The bounding box would reduce the size of the region which needed to be stored. The depth would allow all values of the occluded pixels to be neglected in compression. And the transparency mask would determine which pixels were part of the object, allowing the compression algorithm to neglect background pixels within the bounding box.

A simple example is a parking lot surveillance system which observes an automobile passing through it. If the automobile passes behind trees as it moves straight through the lot, standard compression methods would model the change in all the pixels in which the automobile appears. Algorithms could exploit that fact that the large changes would occur near the automobile. When the automobile passed behind the trees, the local change in pixel value would be very high.

A more effective backgrounding scheme would maintain a static background that modeled the occlusions which can occur. This would leave only the appearance of the car to be modeled as it moves through the scene which can be significantly compressed.

Creating synthetic movies is similar application. By simply overlaying the moving regions from a camera in a similar relative position into the background model of another camera, a new video can be constructed. The mask that models which of the pixels of the moving object should be visible at that depth is intersection of the moving region and the background regions which do not have occlusions at a depth less than that moving region. This binary mask would be used as a blending function of the two videos.

6.2.3 Visualization Applications

One application of reconstruction systems is visualization of the scene. Once the method recreates a scene, the scene can be visualized as a three dimensional model. Often this is considered the primary goal, not just a means of qualitative evaluation of the model. As a result, reconstruction algorithms are often critiqued based on how accurately they recreate the scene.

While our method results in a fairly complete model for areas of the scene in which moving objects are visible, it does not have any depth information where moving objects do not appear. But that lack of depth information means that an occlusion exists for that region or that none of the objects that have been localized have been found in that region.

Chapter 7

Conclusion

This thesis describes a general method for reconstructing a scene using segmentation of moving objects in any number of static views. While standard reconstruction methods attempt to model the location of the visible parts in the scene, this method models the open spaces. Having a model of the open space in a scene as defined by the objects which move in the scene constrains locations where moving objects can exist and where objects at certain depth can be observed. It can also produce effective models for visualization of the scene.

A single camera implementation is described which illustrates the effectiveness of this method despite using a simplifying assumption to determine the depth of the objects. The resulting model has little detail within individual elements of the scene, but it effectively models the general shape of the room. This method did not produce large errors for regions with little or no texture and windowed regions and effectively localized features within the room.

While multiple camera systems would be more difficult to implement, the a multiple camera system would involve fewer assumptions and could potentially be more robust. By using the abundant correspondences that can be observed given sufficient time, the cameras could be robustly calibrated. Using multiple cameras, the correspondences can be verified and the resulting model will be more complete.

Although the systems described are widely applicable, future investigation could lead to incorporating pivoting cameras, modeling dynamic scenes, and single view reconstruction with a stereo rig. The model can be used in interesting applications, including scene visualization, symbolic event abstraction, compression, and synthetic movie generation.

It is evident that this method was motivated by difficulties in implementing tracking systems, because it not only models visibility of objects as they move, but it relies on the moving objects to create that model. That is what defines this method and what sets it apart from other methods.

Bibliography

[1] Ali Azarbayejani and Alex Pentland. "Camera self-calibration from one moving point correspondence," MIT Media Laboratory, Cambridge, MA. 1996.

[2] F Bartolini, V Cappelini, and A Mecocci. "Counting People Getting In and Out of a Bus by Real-Time Image Sequence Processing," *Image and Vision Computing*, Vol. 12, No 1. Pp. 36-41. January/February 994.

[3] R. C. Bolles and H. H. Baker and D. H. Marimont. "Epipolar-Plane Image Anlysis: an Approach to Determining Structure From Motion," *International Journal of Computer Vision*, Vol. 1, pp. 7-55. 1987.

[4] J. Costeira and Takeo Kanade. "A Multi-body Factorization Method for Motion Analysis," *Fifth International Conference on Computer Vision(ICCV'95)*, pp. 1071-1076, IEEE Computer Society Press, June 1995.

[5] Kenneth M Dawson-Howe, "Active Surveillance Using Dynamic Background Subtraction," *Technical Report No. TCD-CS-96-06*, University of Dublin, Trinity College, Ireland, 1996.

[6] U. R. Dhond and J. K. Aggarwal. "Structure from stereo–A review," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol 19, No. 6 pp. 1489-1510, Nov/Dec. 1989.

[7] Alberto Elfes. "Sonar-Based Real-World Mapping and Navigation," *IEEE Journal of Robotics and Automation*, pp. 233-249, IEEE Computer Society Press, June 1987.

[8] O. Faugeras and B. Mourrain. "On the Geometry and Algebra of the Point and Line Correspondences between N Images," *The Fifth International Conference on Computer Vision(ICCV95)*, pp 951-956, IEEE Computer Society Press, June 1995.

[9] F. P. Ferrie and M.D. Levine. "Integrating Information from Multiple Views," *IEEE Workshop on Computer Vision*, pp 117-122. IEEE Computer Society, 1987.

[10] R. I. Hartley. "An Algorithm for Self Calibration from Several Views," *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 908-912, Seattle, WA, 1994.

[11] R. I. Hartley. "In defence of the 8-point Algorithm," *Fifth International Conference on Computer Vision(ICCV'95)*, pp. 1064-1070, IEEE Computer Society Press, June 1995.

[12] D. Hogg. "Model-based Vision: a Program to See a Walking Person," *Image and Vision Computing*, Vol. 1, No. 1,pp. 5-20. February 1983.

[13] B. K. P. Horn. "Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View," *MIT AI Laboratory Technical Report 232*, November, 1970.

[14] B. K. P. Horn. *Robot Vision*, pp. 66-69, 299-333. The MIT Press, 1986.

[15] D. P. Huttenlocher, J. J. Noh, and W. J. Rucklidge. "Tracking non-rigid objects in complex scenes," *International Conference on Computer Vision(ICCV93)*, pp 93-101, 1993.

[16] S. Intille and A. Bobick. "Visual Tracking Using Closed-Worlds," *Proc. of the Fifth International Conference on Computer Vision.* MIT, Cambridge, MA, pp. 672-678, June 20-23, 1995.

[17] R. A. Jarvis. "A Perspective on Range Finding Techniques for Computer Vision," *IEEE Trans. Pattern Anal Machine Intelligence*, Vol. PAMI-5, March 1983.

[18] W.E. Lorensen, H.E. Cline, "Marching Cube: A High Resolution 3-D Surface Construction Algorithm", *Computer Graphics* **21**(3), 1987, pp. 163–169.

[19] Q. T. Luong and O. D. Faugeras. "An Optimization Framework for Efficient Self-Calibration and Motion Determination," *12th IAPR International Conference on Pattern Recognition*, Vol. 1, pp. 248-252. IEEE Computer Society Press, 1994.

[20] L. McMillan and G. Bishop. "Plenoptic modeling: An Image-Based Rendering System," *Computer Graphics(SIGGRAPH'95)*, pp. 39-46, Aug. 1995.

[21] Hans P. Moravec. "Robot Spatial Perception by Stereoscopic Vision and 3D Evidence Grids," *CMU-RI-TR-96-34*. CMU, September 1996.

[22] D Murray and A Basu. "Motion Tracking with an Active Camera," *IEEE Trans. PAMI*, Vol. 16, No. 5, pp. 449-459. May 1994.

[23] Yuichi Nakamura, et. al. "Occlusion Detectable Stereo - Occlusion Patterns in Camera Matrix," *Proceedings Computer Vision and Pattern Recognition '96*, pp. 371-8, 1996.

[24] M. Okutomi and T. Kanade. "A Multiple Baseline Stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 4, pp. 353-363, April 1993.

[25] S. Reddi and G. Loizou. "Actively Keeping a Moving Target at the Center of the Field of View," *IEEE Trans. PAMI*, Vol. 17, No. 8. pp. 765-776. August 1995.

[26] Michael Reed and Peter K Allen. "Automated Model Acquisition Using Volumes of Occlusion," *IEEE International Conference on Robotics and Automation*, Minneapolis, MN, April, 1996.

[27] S. Smith and J. Brady "Optical Flow Based Moving Object Detection in a Moving Scene," *IEEE Trans. PAMI* Vol. 17, No. 8. pp. 814-820. August 1995.

[28] Gideon Stein and Amnon Shashua "Direct Methods for Estimation of Structure and Motion from Three Views," *AIM-1594*, MIT. Cammbridge, Massachusetts, June 1995.

[29] G Sullivan. "A Priori Knoledge in Vision," *Computer Vision: Craft, Engineering, and Science*, pp. 58-79. Springer-Verlag, Berlin, 1994.

[30] R. Szeliski and S. B. Kang. "A Direct Method for Visual Scene Reconstruction," *IEEE Workshop on Representations of Visual Scenes*, Cambridge, Massachusetts, June 1995.

[31] Takeo Kanade, et. al. "A Stereo Machine for Video-rate Dense Depth Mapping and Its New Applications," *Proceedings of 15th Computer Vision and Pattern Recongnition Conference*, San Francisco, California, June 18-20, 1996.

[32] H. Wang and M Brady. "Real-time Corner Detection Algorithm for Motion Estimation," *Image and Vision Computing* Vol. 13, No. 9. pp. 695-703 . November 1995

[33] B. Yamauchi. "Mobile Robot localization in Dynamic Environments Using dead Reckoning and Evidence Grids," *Proceedings of IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, April 1996, pp. 1401-1406.