ON THE AUTOMATED SELECTION OF VIEWPOINTS FOR IMAGE BASED MAPS

Eric Bourque

Department of Computer Science
McGill University, Montréal

July 1998

A Thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements for the degree of
Master of Science

ABSTRACT

This thesis presents an approach to the automated construction of visual maps, in the form of walk-throughs, of an unknown environment. Our approach is based on the selection of informative viewpoints within the environment. These viewpoints are locations in the environment which correspond to views containing maximal visual *interest*. This approach to environment representation is analogous to image compression. Our goal is to obtain a set of selections resembling those made by a human observer given the same task.

Our computational procedure is inspired by models of human visual attention outlined in the literature on human psychophysics. We make use of the underlying edge structure of a scene, as it is largely unaffected by variations in illumination. Our implementation uses a mobile robot to traverse the environment, and then builds an *image-based* virtual representation of the environment, only keeping the views whose responses were highest. We demonstrate the effectiveness of our attention operator on both single images, and in viewpoint selection within an unknown environment.

RÉSUMÉ

Cette thèse présente une approche à la construction automatisée de scènes de réalité virtuelle basée sur des images d'un environnement inconnu en permettant à l'usager de contrôler ses déplacements. Notre approche est basée sur la sélection de points de vue informatifs situés dans l'environnement. Ces points de vue sont des emplacements dans l'environnement qui correspondent aux vues suscitant un intérêt visuel maximale. Cette approche à la repésentation de l'environnement est analogue à la compression d'images. Notre but consiste à obtenir un ensemble de sélections s'apparentant à celles qu'effectuerait un observateur humain dans une même tâche donnée. Notre procédure de calculs s'inspire des modèles d'attention visuelle rapportés dans la littérature de psychophysique humaine. Nous utilisons le contour de la structure sous-jacente d'une scène puisqu'elle n'est pas grandement affectée par les variations d'éclairage. Notre application a recours à un robot mobile pour effectuer la traversée de l'environnement, puis construit une représentation virtuelle de l'environnement basée sur des images, ne conservant que les vues évoquant le plus grand nombre de réponses. Nous démontrons l'efficacité de notre opérateur d'attention sur des images individuelles ainsi que sur la sélection de points de vue à l'intérieur d'un environnement inconnu.

ACKNOWLEDGEMENTS

My thanks are due to my supervisor, Dr. Gregory Dudek for his wonderful leadership and direction. Many ideas evolved from discussions in his office while countless people waited outside the door ...

Many thanks to Mike Daum, who, without knowing it, taught me to find the answers the hard way before asking.

I would like to thank Nicholas Roy, for always being willing to discuss research issues, and for being a good friend. Proof-reading this thesis while on a cross-country journey was certainly beyond the call of duty.

Philippe Ciaravola was instrumental in the experimental part of this research. Various software written by him was used extensively during the environmental experiments. He has also provided a few of the figures in this thesis. Thanks Phil.

Richard Unger, and Deeptiman Jugessur were quick to volunteer their time when I needed help getting a robot and various other things to the art gallery. Thanks guys.

I would like to extend my thanks to Lucia Carangelo, who not only helped with the translation of the abstract, but put up with my increased absence as the deadline approached, as well as my many thesis moods.

I would also like to thank the Canadian Centre for Architecture for being enthusiastic about having a robot in their gallery.

And finally, I would like to thank the members of the mobile robotics laboratory, who all, in one way or another, have contributed to this research.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

Introduction

This thesis presents a comprehensive approach to the *graphical* modeling of arbitrary environments. Using an exploring robot, we construct a navigable collection of images that captures the appearance of an environment. This constitutes, in effect, an image-based map. Because we do not want to retain very large amounts of data, we keep only the most *interesting*[1] views. This problem is akin to that accomplished by many tourists on their holidays: to recapitulate an excursion using a set of images. We will refer to this as the *vacation snapshot problem*.

1.1. Motivation

Graphical representations of an environment can be used for a wide range of applications. When these provide a realistic visual experience, they are frequently referred to as virtual reality (VR) representations, because they allow a user to experience realistic aspects of an artificial environment. Typical interfaces include: a cave[2], goggles with head trackers, data gloves, etc. The standard approach to creating VR representations consists of using an *a priori* manually-constructed 3D model of the environment for real-time graphical rendering from a desired viewpoint, typically called *model-based* VR. One factor limiting the utility of this type of VR modeling is that the construction of a realistic synthetic environmental model can be extremely

[1]The notion of interesting will be fully developed later on.
[2]A cave is a room in which several projectors are used to project rendered images onto the walls.

labor intensive – the modeling and texturing of a single object can take months.[3] In addition, the computational burden involved in rendering scenes for model-based VR can be substantial, often requiring specialized hardware for real time performance. Finally, obtaining a truly realistic result for an arbitrary environment remains exceedingly challenging.

An alternative technique called *image-based virtual reality* refers to the use of real image data (photographs) of an existing environment to create a VR environment. By using image data from a real environment, rendering overhead is minimized but data acquisition becomes increasingly important. One of the earliest examples of this technology was the *branching movie*: contiguous film clips that can be played in different orderings to provide a user-controlled walk-through [**44, 56**].

The type of image-based VR interface we employ in the work described here allows a user to view the scene from a fixed viewpoint, and to move discretely between precomputed viewing locations. Although the observer motion is currently constrained, image-based VR permits extremely realistic scenes to be displayed and manipulated in real time using commonplace computing hardware. There is also ongoing research on the *image-based rendering* of images; that is, the rendering of images associated with viewpoints that have never been explicitly sampled by using information extracted from nearby views [**31, 30, 11, 43, 5**]. The commercial product QuickTime VR (a trademark of Apple Computer) exemplifies the particular image-based VR user-interface discussed in this thesis [**10**].

Several authors have considered the use of exploring robots to map an unknown environment. While creating a 3D model using mobile robots is a tantalizing objective, it appears that the issues of maintaining metric accuracy, assuring accurate sensing of the surfaces and obstacles in the world, and performing the task efficiently

[3]The construction of a single complete textured model of an airplane in the film "Con Air" took two months [**57**], and the rendering alone in the Disney Production, "Toy Story" took 800,000 machine hours [**66**].

(in terms of time and cost) make construction of a true 3D representation unsuitable to many applications. The image-based map described here can serve as an appropriate substitute in many cases.

1.2. Building Image-Based Maps

The use of image-based VR addresses the shortcomings of limited realism and high computational load imposed by conventional model-based VR. Unfortunately, it only partially alleviates the intensive effort needed to create a VR world model: the acquisition of the requisite images to construct an image-based VR model still entails effort and expertise. In addition, if a scene must be modeled under different conditions or at different times, then the image acquisition processes must be carried out repeatedly. Furthermore, selecting suitable vantage points to produce an evocative and complete VR model is in itself an important issue, although it is often difficult in virtual reality applications to represent a scene such that the user is able to understand the topology of the environment.

In order to create an image-based virtual representation of an (unknown) environment, solutions are needed for several sub-problems:

(i) A technique must be available for covering (and exploring) free space.

(ii) An algorithm is needed to select specific regions discovered during the exploration that will serve as representative viewpoints.

(iii) Suitable images must be acquired and combined from the selected viewpoints.

(iv) A graphical interface technology is needed to display the image-based environment, and allow user interaction and feedback.

This thesis concentrates on the automated acquisition and construction of image-based VR models by having a robotic system select and acquire images from different vantage points within an unknown environment (steps (ii) & (iii)). The objective is to provide a fully automatic system for both the selection and acquisition of the

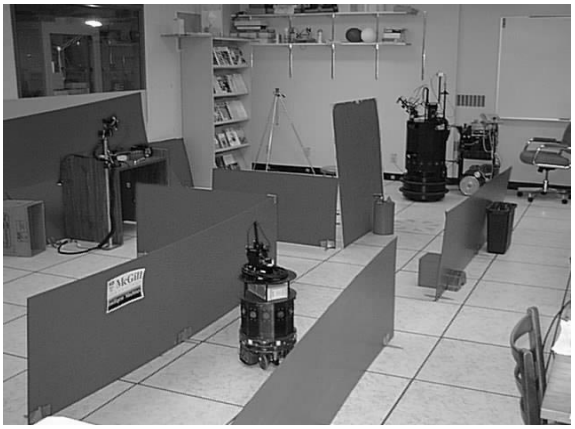


FIGURE 1.1. Robot traversing a simple controlled environment in our laboratory.

needed image data. In principle, this can be augmented by additional cues provided by a human operator.

The algorithm used to determine suitable vantage points within the environment is based largely on models of human environment exploration. We have developed a visual attention operator which quantifies the visual interest of a scene at various locations, thus allowing us to retain only the most interesting locations (in terms of certain criteria) in a scene. This is done to minimize the amount of data retained in order to reconstruct the scene in the VR model.

1.3. Applications

Image-based VR modeling appears promising in several contexts. An obvious class of application for this type of technique is to summarize a location for entertainment purposes: for example to capture and regularly update a locale for placement on a web site. A more prosaic application is the task fulfilled by a security robot: to capture images of an environment that must be surveyed regularly, either for threat detection or for data logging purposes.

Other task domains include those where the scene to be examined is either too remote, too dangerous, or inconvenient for a human operator to visit directly. As such, the potential application contexts overlap those for teleoperated robotics. A

recent example can be found in the VR system used to plan trajectories for the Mars Sojourner [**1, 2**]. In fact, many of the characteristics of the data acquisition problem to be accomplished by extra-terrestrial vehicles can be effectively captured by the *vacation snapshot problem*.

1.4. Outline

The following is an outline of this thesis:

Chapter 2 discusses the related work in the areas of autonomous robot exploration, human visual attention, as well as the image-based VR interface we are using. In this chapter we discuss the relevant work from the field of human psychophysics as well as cognitive psychology and promote its application to the *vacation snapshot problem*.

Chapter 3 discusses our approach to the vantage point selection problem, and also presents the motivation for selecting environment independent features of interest, or *interest points* within the environment. Our use of edge information is also discussed in this chapter.

Chapter 4 formalizes our method of evaluating interest points in a single image, using our interest operators. It then presents a method of combining the various operators.

Chapter 5 extends the techniques and methods presented in chapter 4 so that they may be used in a large scale environment as opposed to in a single image. It is in this chapter that we describe image mosaicing, as well as the image-based VR interface we are using. We also discuss the issues involved in using a continuous or *on-line* algorithm for vantage point selection.

Chapter 6 describes our exploration and modeling system architecture. It is in this chapter that we present our implementation using a mobile robot to create an image-based virtual representation of an unknown environment.

Chapter 7 presents our experimental results on single images and in real environments. The single image results are shown on photographs taken of various natural

scenes. The environmental experiments were first conducted in a simple controlled environment in our lab, and later in the uncontrolled environment of an art gallery.

Finally, we conclude with a discussion of the work, as well as an outline of the open questions and possible directions for future research in this area.

CHAPTER 2

Background

The question of what attracts our visual attention is by no means uncharted. Painters, graphic artists, and architects are all examples of professionals who have been well schooled in the various aspects of visual attention. Many researchers have examined the functioning of the human visual system in an attempt to understand and model its behavior. More recently, the field of computer vision has borrowed from human psychophysics, cognitive science, and neuroscience to better understand and simulate visual attention.

The multi-disciplinary work presented in this thesis is rooted in three distinct research areas. The relevant background from each will be discussed separately:

(i) autonomous robot exploration
(ii) visual attention
(iii) image-based virtual reality

It is the synthesis of the above techniques which allows us to build an image-based virtual reality representation of an unknown environment. In particular, we have developed a formal description of interesting views in the environment that we use to drive image acquisition.

2.1. Autonomous Exploration

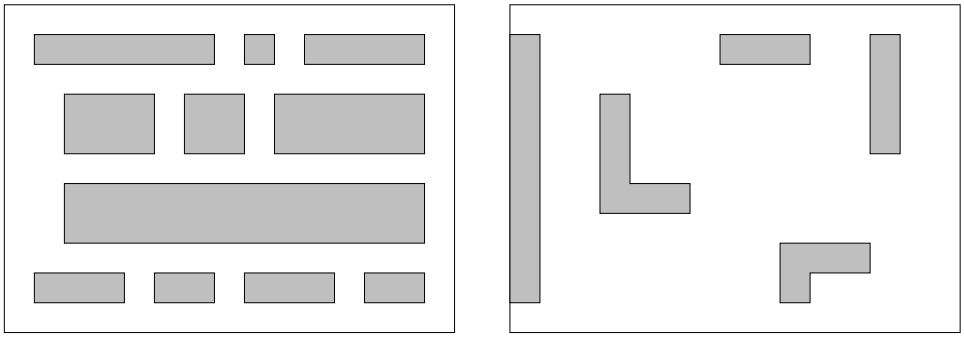
Because this thesis is primarily concerned with the selection of interesting locations in an environment and not with the problem of exploration per se, we will only touch briefly on the relevant work on autonomous environment exploration. A more exhaustive survey of the computational principles of mobile robotics is presented in [**19**].

In order for a mobile robot to explore (and possibly map) an environment, there are several sub-problems which must be addressed:

- exploration
- navigation/path planning
- obstacle avoidance
- localization
- environment representation
- sensor modeling

The exploration algorithm is perhaps the most important sub-problem, as it is responsible for making the high level decisions of where to proceed. The choice of where to go next is obviously task dependent, and in the case of mapping, we would like the robot to enter a previously uncharted area so that we may expand our previous knowledge of the environment. Several authors have devised algorithms for covering unexplored free space [**38, 37, 3, 20, 46, 62, 12, 64**], however, the details of these algorithms are outside the scope of this thesis. It is important to note that even in the case of image-based mapping of an unknown environment, the exploration algorithm used should be matched to the environment being explored. For example, an office-like environment and a stereotypical art gallery are more efficiently and effectively mapped if their exploration strategies cater to their geometry and some high level knowledge of their contents (figure 2.1). Consider the office environment: it would be desirable to map the free space, as this would provide the end user of the virtual environment with a better understanding of the layout and its contents. In contrast,

an art gallery displaying paintings would need to have the walls mapped, but the free space is relatively unimportant. This would suggest a wall-following type exploration strategy. Although the two environments could have been explored using an algorithm which guarantees complete coverage, fine tuning the algorithm to the environment and desired task can lead to a large time saving.



(a) Idealized office environment (b) Idealized art gallery environment

FIGURE 2.1. Two different environments which would benefit from specialized exploration strategies.

Once a goal has been selected by the exploration component, the robot must plan a safe path (perhaps with limited information) from the current location to the goal. This presents two separate issues: (1) a high level trajectory must be generated, and (2) the robot must not collide with obstacles along the way. Traditional approaches apply potential fields [**36**], or some form of global search, or a combination of the two. A complete discussion of robot motion planning can be found in [**42**].

Several methods exist for representing an environment. Two of these methods, topological mapping, and metric mapping, are quite popular in the field of mobile robotics. A metric map is a map where each location on the map corresponds to an exact location in the environment, and the distance between two points on can be measured directly off the map. A map of Canada would be an example of a metric map. A topological map, on the other hand, is a map where the relative position of

objects is represented through a series of links, but distances can not be read directly off the map. Most subway maps are examples of topological maps.

Because the nature of the map we are building is fundamentally topological, errors in positioning due to dead reckoning [**21, 7**] (errors which accumulate in the robot's estimated position vs. its actual position) can be tolerated, and we therefore are not concerned with robot localization. The accuracy of the internal positional estimate from the odometry sensors has proven more than sufficient for the construction of our map.

A suitable internal representation of the environment is needed in order for the above techniques to be realized. A common representation when metric accuracy is key is the occupancy grid [**50, 23**], a data structure based on a tessellation of the environment where each cell (region in the tessellation) is marked as empty, occupied or unknown based on sensory input. It is interesting to note that the tessellation (cell decomposition) need not be regular, and in fact there is a large reduction in storage space if quad-trees [**71, 34, 72**] or binary space partitioning trees [**65**] are used.

When the task being performed does not depend on metric accuracy, a topological representation is commonly used. This representation is particularly well suited to large environments, where the task at hand does not depend on precise knowledge of the environment at each possible location. The topological robot exploration strategy developed by Kuipers and Byun [**38**] creates a graph of the environment where the distinctive places form the nodes of the graph and they are connected by paths through space with equal distinctiveness (distinctiveness contours). In [**18**], Dudek proposes a representation which uses multiple abstraction levels, where each node of the topological graph can contain local metric information.

The implementation of a complete mobile robot system using some of the above techniques on a mobile robot in our lab is outlined in [**6**].

2.2. Visual Attention

Selective attention exists because we need a mechanism for prioritizing, or focusing on regions of interest relevant to some task due to limitations on processing ability, or memory load. The *vacation snapshot problem* captures this constraint by limiting to a small number of images with which to recapitulate the scene. This automatic processing can be task dependent - imagine the images retained by a stained glass artist, and an organ builder on a tour of the cathedrals in Europe. Although both subjects would have seen the same churches, they would certainly have focussed on different areas, and thus prioritized different views in their visual memory. In this thesis, we develop a model for simulating attention in a task-independent, and environment-independent fashion. Refer to section 3.2 for more details.

Perhaps one of the earliest "attention" operators, was that developed by Moravec. His operator was used to find possible matching points in a pair of images to calculate disparity. This operator detects points at which intensity values are varying quickly in at least one direction [**27**]. Other feature detectors used for the stereo correspondence problem include corner detectors, and curvature detectors.

Another feature detector which has been used for various tasks is a symmetry operator. One such operator, the annular operator, involves the comparison of edge information within a region in an image bounded by two circles of different radii. Edges which fall in the annular region enable the detection of circular correlation between edge segments [**32, 33**].

Schneider and Shiffrin suggest that information processing is divided into two distinct types of processes: (1) labile control processes, and (2) learned or inherent structural components [**60**]. They show that it is possible to train subjects to recognize certain inputs as targets, thus modifying their previous controlled processing and initiating automatic attention responses. These attention responses then direct attention (controlled processing) automatically to the target, regardless of concurrent inputs. It is this type of behavior that we simulate with our interest operators.

Rensink, O'Regan, and Clark have done some research on the way humans observe changes in a scene. They have found that we are able to immediately detect changes in a scene, provided there are no visual interruptions (we are constantly able to view the scene). They noted, however, that when there are brief visual interruptions (presented as blank frames in a movie sequence, for example) detection of such changes becomes extremely difficult. Their results indicate that attention is required to perceive change [**55**].

Several authors have shown there is an ability to shift specialized processing across the retinal image [**53, 51, 59, 14**]. Koch and Ullman suggest that there is a pre-attentive, or *early*, representation in the primate visual system composed of elementary features such as color, orientation, direction of movement, disparity, etc., which are searched in parallel, and a central representation which contains only the properties of the *selected* location in the retinal image. It is the mapping between these two phases (early and central representations) which constitutes their attentional model. They propose a set of "selection rules" that determine the next location which will enter the central representation with the predominant rule being based on the conspicuity of the location, that is, the distinctiveness of its properties relative to the properties of its neighborhood [**35**]. The latter is implemented using a *Winner-Take-All* network [**24**].

Tsotsos et. al. have outlined a selective tuning mechanism similar to that of Koch and Ullman [**70, 69**]. Their winner-take-all (WTA) algorithm is updated to better match the current understanding of the primate visual system. In particular, that attentional shifts do not take time proportional to their distance [**54**]; there is no attentional gradient – shifts of attention happen in constant time. Not only does their WTA outperform other methods, it is also near optimal in terms of efficiency from a biological standpoint.

For short-term attention, several featural dimensions have been identified that lead to pre-attentive "pop-out" and, presumably, serve to drive attentional processing [**68, 67**]. Likely feature maps used by humans for attentional processing include

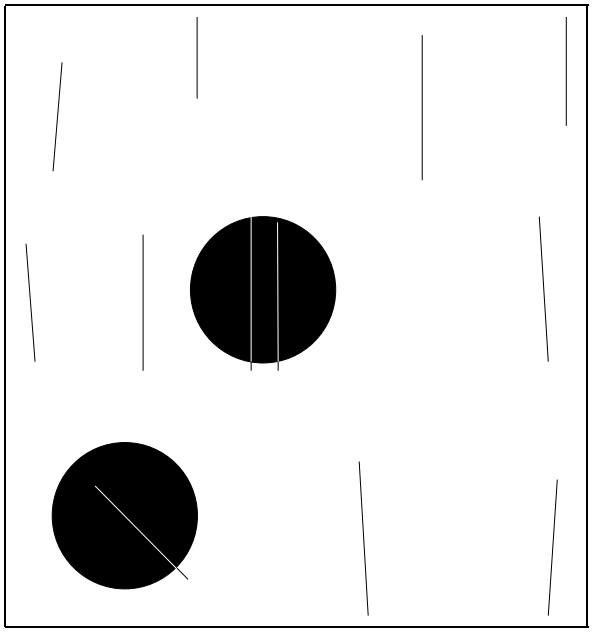


FIGURE 2.2. Example of the effectiveness of the interest operator on a simple image. The inverted (black) regions represent the selected locations. This stimulus and the results resemble those used and observed in tests of human visual pre-attentive vision.

those for color, edge orientation, and edge density. This notion of a statistical measure of image content is closely related to models of texture segmentation and texture discrimination based on global statistics, or the global aggregation of local measurements [**28, 52, 39**]. There has been substantial debate over the precise nature of psychophysical texture discrimination and whether it exploits first order statistics, second order statistics, learning, or other analyses [**26, 47, 58**]. In this thesis we are concerned with a specific application and hence we restrict our attention to a first-order model. Despite its simplicity, the behavior of our attention operator on simple stimuli resembles that predicted by the psychophysical literature (figure 2.2).

2.3. Image-Based Virtual Reality

The image-based VR interface we use requires cylindrical (360 degree) panoramic photographs. These types of images have been used for documentary purposes

from even before photographs were developed.[1] By exploring and selecting a *set* of panoramic images, we can capture most of the appearance of an environment. In principle, a suitable selection of panoramic images can serve to approximate the *light ray manifold* of a scene, and perhaps even permit the scene's reconstruction [**41**]. The Lumigraph, the Light Field, and the plenoptic array are related constructions that couple the reconstruction of a view in the scene to the sampling of its light rays [**25, 49**]. Scene visualization based on such methods is referred to as *image based rendering*. Our approach to visualization is based on collecting sample cylindrical panoramic images at locations selected by our *iconic-view selector* (an attention-like operator). Section 5.1 discusses the image-based VR interface we are using in detail.

[1]One instance of panoramic imagery that predates photography is the art of Hendrik Willem Mesdag and his associates. An example is a cylindrical room adorned by a panoramic painting c. 1880, on exhibit at Museum Panorama Mesdag in The Hague.

CHAPTER 3

Methodology

3.1. Approach

A primary bottleneck in the use of image-based VR is that the creation of models is time consuming and requires specialized expertise. The key issues in VR model development are: (1) the selection of suitable vantage points to cover the interesting aspects of the environment, and (2) the acquisition of suitably calibrated images from those vantage points. The image data is then post-processed to provide the image-based VR model. When this model consists of a collection of viewpoints in the environment, it is referred to as a *multi-node* model. The selected viewing locations form the nodes of a topological graph which determines the set of possible trajectories available to the user of the model. In the image-based VR interface we are currently using, the user experiences discontinuous motion between adjacent nodes in the topological graph, although the user can look in any direction from an individual node. In this thesis, we describe an approach to the fully automated creation of image based VR models of an environment with essentially no human intervention.

Our approach is based on using a small mobile robot to autonomously explore an unknown environment and collect the image data of interest (figure 3.1). We will presuppose that the robot travels along some trajectory through the environment, and that it can estimate its current (x, y, θ) position at any time. An overview of

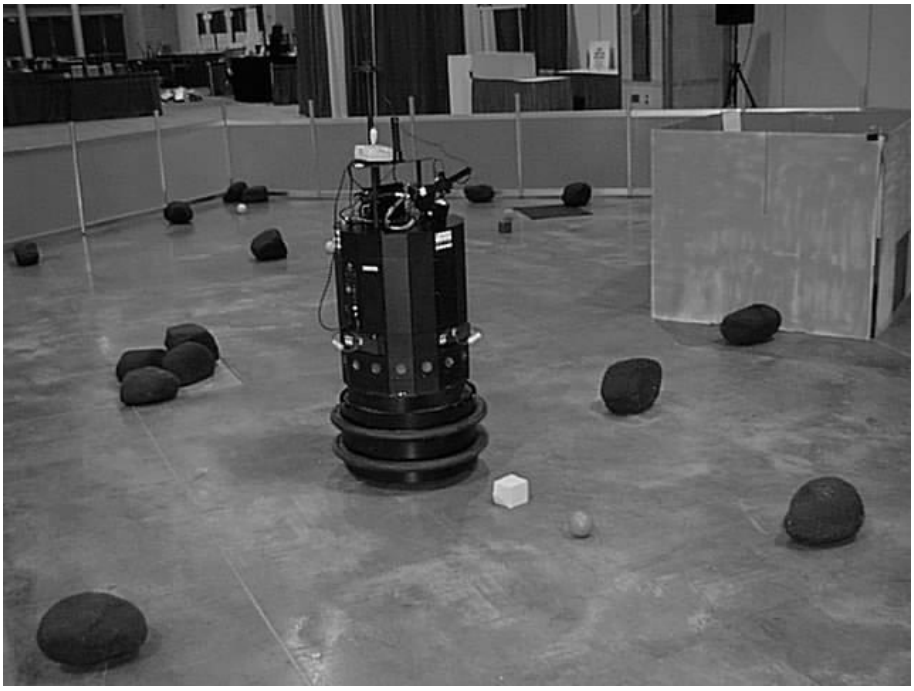


FIGURE 3.1. A picture of the Nomad 200 robot traversing an unknown environment.

exploration strategies is outlined in section 2.1. In principle, the exploration could even be manually controlled.

While the robot moves, it maintains an internal model of its own position. This model, based on dead-reckoning, can be corrected using external sensing or external landmarks. In practice, it is difficult to determine landmarks that are sufficiently general to function in any environment. As a result, while we use estimated metric positions to construct our VR model, these can be coarse estimates only; the map is fundamentally topological in nature.

Since our objective is to construct a virtual environment that is evocative of the original environment for human observers, our approach is inspired by models of human visual environment exploration as outlined in section 2.2. Thus, our approach is to compute a map over an image (perhaps a 3-dimensional image) of how much each point attracts attention. The extrema of this map provide a set of attentional features.

Our computational procedure for defining features is dependent on the edges present in an image. Edge structure has been used extensively in computational vision. These structures have an apparent psychophysical relevance in addition to the fact that they tend to encode geometric structure in the scene (eg. object boundaries, markings) [**48**]. Several extremely promising methods have been developed for grouping edge elements into high level features such as curves or closed contours [**74, 22**]. Doing this in a bottom-up, robust, stable and environment-independent manner, however, appears to be a problem that is not yet fully resolved. Nonetheless, the distribution of edge elements is clearly related to basic scene structure. Further, the edge element distribution has the advantage of being robust to variations in illumination.

It is with this in mind that we have formulated a metric for visual iconic-view selection based on the density and orientation of edge elements without grouping or segmentation. To focus attention at locales that are notable, our attention mechanism is driven to locations where the local edge element density and/or orientation differ substantially from that in the surrounding neighborhood. This has similarities to Koch and Ullman's major selection rule [**35**].

3.2. Environment-Independent Features

3.2.1. Paying Attention To What Is Interesting. Our approach to environment modeling using panoramic images is based on the idea of capturing views from locations of interest. This vague but compelling concept naturally leads to three different notions of "interesting" views in the context of a specific environment. These are:

(i) Views which would attract "early" visual attention in human observers based on preconscious mechanisms. Such views are those which would be selected by pre-attentive processing in, for example, a *search-light* model [**29**].

(ii) Views which are relevant to a specific task or functional model (this is closely related to "high-level" attention).

(iii) Views which capture the "typical" appearance of the environment.

In the present work we focus primarily on the first characterization of what is interesting: those tied to "pre-attentive" vision. This definition has the advantage of being closely related to existing models of human visual attention. In addition, using early vision as the basis for our interest operator permits a generic domain-independent solution to the mapping problem. While we do not propose a model of biological attention *per se*, we frame our discussion with biological attention as a starting point.

In practice, high-level attention and the selection of task-specific views (ii) is clearly of great significance, but it is a slightly different problem. It may be that, especially for biological systems, the selection of loci for low-level attention is a precursor of higher-level attention, and hence our solution based on early attention is a natural first step in building a more comprehensive solution. This hypothesis is consistent with several current theories on the role of attention in visual perception [**67**].

A variety of models for visual attention have been formulated with emphasis on both robotic systems and on human perceptual processing. Three primary issues in the computational modeling of attention have been of particular interest: *what* to look at (addressed primarily in the context of biological attention), how to *shift* attention from one location to another over time, and what *architecture* to use to combine information across space, scale, time and multiple feature spaces. Alternative approaches include the use of hierarchical top-down architectures [**70**, for example] or bottom-up data-driven methods [**73**, for example]. Our work involves making selections from discrete targets that are generated by an external process (the exploring robot). For this reason, we will not address the issue of shifting attention between alternative foci over time. For similar reasons, we select a simple architecture for directing attention based on an exhaustive examination of all possible fixation points. Our approach involves the use of an attentional operator at one or more spatial scales.

3.2.2. Edge-Based Maps. In keeping with the notion that attention is drawn to regions that are anomalous, and hence informative (in terms of a maximum entropy encoding), we look for regions that differ from the typical edge element

distribution. Psychophysics as well as neurobiology suggest that edge density and orientation are two key attributes of image data. We have thus identified four attributes of images that can be used to *rapidly* identify interesting regions.

- Edge element density: to what extent does the edge density in a local neighborhood differ from the mean density?
- Edge orientation: does the local edge orientation differ from the orientation distribution in a larger neighborhood?
- Density of perceptual groups: does the local density of certain perceptually relevant features differ from what is typical (for example, is there an unusual density of parallel lines) [**45**]?
- Color: is the local distribution of colors, as described by a color space histogram, substantially different from what is expected?

Each of these attributes appears to be both effective in practice and relevant to models of biological attention [**35**]. We will confine our discussion to the first two types of interest operator since they can be readily computed from simple achromatic edge element data.

Models of human pre-attentive visual feature detection suggest that a multiple-feature winner-take-all computation is likely to take place in driving biological attention. In contrast, we have also examined the use of a two dimensional (or multi-dimensional) operator that combines information across feature maps, as well as a winner-take-all scheme.

CHAPTER 4

Calculating Attention in a Single Image

As a precursor to the use of attention for selecting viewpoints of interest we will consider the use of attention to select regions of interest in a single (2D) image. The 2D analogy to our environment mapping process is the storage and recovery of the content of an image using a selection of sub-windows. In fact, we can define selecting a suitable window of a 2D image in a manner notationally isomorphic to the 3D problem. In the case where the distance to the objects in the environment approaches infinity (and hence we have parallel projection), the 2D problem and the environment mapping problem can be reduced to one another.

In order to formulate our attention operator, we must first devise a notational framework: we define a matrix **I** corresponding to the intensities of the image under consideration. We can then define a function $I_{x,y}(\phi, \theta)$ whose value is the intensity at location (ϕ, θ) in the sub-region of **I** starting at (x, y): $I_{x,y}(\phi, \theta) = \mathbf{I}_{x+\phi\ y+\theta}$.

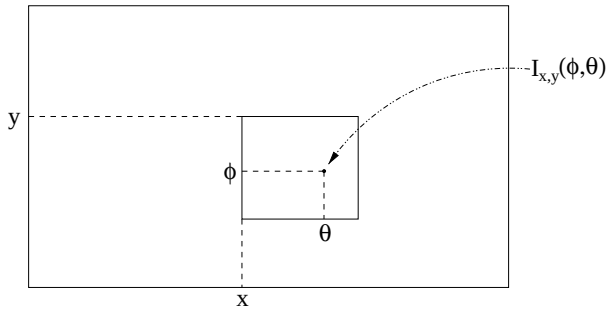


FIGURE 4.1. Organization of $I_{x,y}(\phi, \theta)$ in an image.

Although the resolution of the above function is limited to the size of the matrix **I**, we could devise a function which would return sub-pixel values through a host of interpolation techniques. It is also possible to increase the sampling resolution by using a higher performance intensity sensor. It is with these points in mind that we will assume $I_{x,y}(\phi, \theta)$ is continuous.

4.1. Density

Our first metric for computing visual attention is based on edge element density. Each element in the edge map $E(\mathbf{I})$ of image **I** has an intensity associated with the strength of the edge to which it belongs. We compute a density map $D(i, j)$ over the entire image by convolving a Gaussian[1] windowing operator of size $A \times B$ with the edge map. Each point in the map is divided by the total possible number of edgels, giving the following measure of density:

$$D(i,j) = \alpha \int_{j-\frac{B}{2}}^{j+\frac{B}{2}} \int_{i-\frac{A}{2}}^{i+\frac{A}{2}} e^{-\frac{(\phi-i)^2+(\theta-j)^2}{2\sigma^2}} E(I_{x,y}(\phi,\theta)) d\phi d\theta \tag{4.1}$$

with

$$\alpha = \frac{1}{\int_{-\frac{A}{2}}^{\frac{A}{2}} \int_{-\frac{B}{2}}^{\frac{B}{2}} e^{-\frac{l^2+m^2}{2\sigma^2}} dl dm} \tag{4.2}$$

Since we are interested in unusual locations, we define the interest, Γ, as the deviation from the mean $\hat{D}$ over the entire image:

$$\Gamma(i,j) = |D(i,j) - \hat{D}| \tag{4.3}$$

We then find the extrema of this map, that is, the locations with the highest deviations from the mean and define those as the most interesting locations, based on edge density alone. This involves an implicit assumption that the edgel density

[1] A Gaussian operator has desirable properties in terms of localization in both space and frequency space [**27**].

distribution is uni-modal, since otherwise we may occasionally obtain non-intuitive results. The extrema of this operator will typically be associated with edge junctions and other geometric "events" in typical indoor images, although they can also be associated with empty regions in textured images. See section 7.2.2 for examples of the use of the density metric.

4.2. Orientation

The second operator we use for computing attention is edgel orientation. Each entry in the orientation map $\Theta(\mathbf{I})$ is the orientation of the corresponding edgel in the edge map. We compute a local orientation signature $O(i,j)$ similar to the density map defined above, as follows. In order to select orientations that are maximally different from the typical orientation structure in the scene, we make a noise-insensitive estimate of the most likely orientation: a robust maximum.

Given a function,

$$\Phi(k,i,j) \qquad k \in [0,\pi) \tag{4.4}$$

which returns the number of edgels with orientation k in the local neighborhood of (i,j), we can compute the robust maximum orientation as follows:

$$\Phi^*(k,i,j) = \Phi(k \bmod \pi, i, j) \qquad k \in R \tag{4.5}$$

$$O(i,j) = \max_{k\in[0,\pi)} \int_{q-\frac{\omega}{2}}^{q+\frac{\omega}{2}} \Phi^*(k,i,j)\ dk \qquad \omega \in (0,\frac{\pi}{2}) \tag{4.6}$$

where ω is the width of the subsection of the orientation distribution we wish to consider. In practice, $\Phi(k,i,j)$ is also convolved with a Gaussian windowing operator. An example orientation distribution for a single image can be seen in figure 4.2.

Again, we are interested in unusual locations with respect to orientation, so we define interest as the deviation from the overall robust maximum orientation $\hat{O}$:

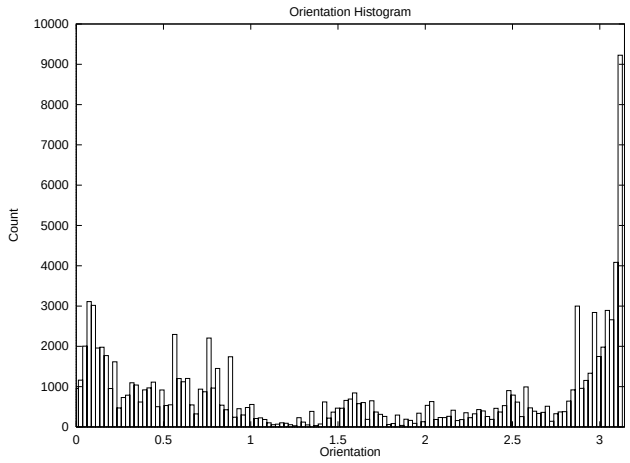


FIGURE 4.2. Histogram of different edge orientations in a single image. The plot wraps around at π radians.

$$\Omega(i,j) = |O(i,j) - \hat{O}| \tag{4.7}$$

We then find the extrema of this map which will be the local neighborhoods with the highest deviation from the maximum orientation, and define them as the most interesting locations based on orientation alone. Section 7.2.2 demonstrates the behavior of the orientation operator.

4.3. Combining Density and Orientation

A suitable function is needed to combine the information from the density and orientation operators such that we achieve results which are stronger than the results which each operator can provide independently. To produce a compound interest operator, we combine the individual interest ratings due to the individual measurements using an L_n metric:

$$C(i,j) = \sqrt[n]{(\gamma\Gamma(i,j))^n + ((1-\gamma)\Omega(i,j))^n} \tag{4.8}$$

where n is a constant and the value chosen for γ depends on the type of environment being sampled. By using a large value of n we obtain a behavior that resembles a winner-take-all scheme, while smaller values of n exploit combinations of features. We have also considered the use of multiple attention maps at multiple spatial scales, leading to feature detections $\mathcal{M}_f^s(i,j)$ where s and f are indices that specify the scale and feature type. In this case, we combine these maps using:

$$C(i,j) = \sqrt[n]{\Sigma_s \Sigma_f \gamma_f^s \mathcal{M}_f^s(i,j))^n}. \tag{4.9}$$

Section 7.2.2 demonstrates the effectiveness of the combined density-orientation operator with $n = 1$.

CHAPTER 5

Selecting Locations of Interest in the Environment

5.1. System specification

The image-based interface we use (QuickTime VR [**10**]), was one of the earliest systems to forgo the traditional modeling and rendering phases. Instead, by capturing a series of environment maps, it allows a user to *look around* a scene from fixed points in the environment. These points are pre-determined, and can not be changed once the VR model has been constructed. The latter is a severe limitation of the interface we are using, but as mentioned in chapter 1, there are various techniques currently under investigation for rendering environment maps from vantage points which were not explicitly sampled. Perhaps most notable, is Gortler et. al.'s Lumigraph [**25**], which allows arbitrary views of a limited space to be rendered quickly from a plenoptic function which has been reduced from 5D[1], to 4D by using surfaces surrounding objects in the scene. By limiting the interest to the light leaving the convex hull of a bounded object, the plenoptic function only needs to be represented along some surface surrounding the object.

The advent of improved image-based rendering techniques, or faster hardware, will not limit the usefulness or the necessity of the work presented here. Environment

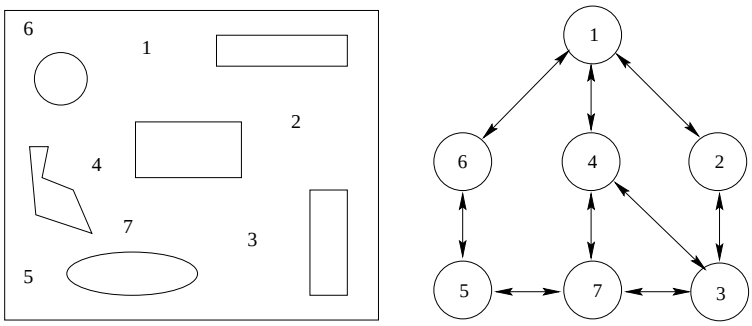
[1]For every (x, y, z) point in 3-space, there is a pixel corresponding to the ray projected along (ϕ, θ).

maps which are rendered from viewpoints which have not been sampled explicitly will always be lower quality than those which have been sampled[2], and thus the question of where to sample the environment remains valid. Secondly, the capture of image data to create a Lumigraph is non-trivial: one needs a *blue-screen* stage containing various calibration marks so that camera position may be accurately computed for each image. The method we outline for the automated acquisition of image-data could easily be modified to simplify the creation of a Lumigraph.

To construct an image-based model of an environment, we must first gather a set of images from each point $\mathbf{P}_i = (x_i, y_i)$ in the environment we wish to model. Because we are using a conventional camera, several images are required to span a complete cylindrical environment map. The set of images from each point is then tiled into a mosaic which can be subsequently mapped onto a viewing volume [**63, 10**]. In practice, the mosaic is produced by "stitching" or fusing all of the individual images from one sample location into a single composite image [**63**]. This involves registering consecutive images with one another using methods analogous to those used in stereo correspondence. In practice, this implies that camera rotation should be about the nodal point of the camera, that the scene should be static (or the sequence should be acquired as quickly as possible), that lighting should remain constant, and that camera motion must be minimized. These types of constraints, while conceptually trivial, substantially complicate the manual acquisition of image data for VR mosaics.

The shape of the panoramic image that is used can vary: both spherical and cylindrical projections have attractive properties, while cylindrical projections are predominant in existing applications. The latter gives the viewer a limited viewing hemisphere, in that information is lacking at the vertical extremes. For any viewing vector $\mathbf{v} = (r, \phi, \theta)$ where r represents the zoom factor and ϕ, θ are the Euler angles from a sampled location, the appropriate field of view can be mapped onto a planar surface for display [**10**].

[2]Assuming a real environment.



(a) A synthetic environment. The numbers correspond to the locations which were sampled.

(b) The associated topological graph.

FIGURE 5.1. A synthetic environment with the sample locations and its associated topological graph.

FIGURE 5.2. A snapshot of Apple Computer's *QuickTime VR* interface. The user can change their view by clicking and dragging in the desired direction of movement, and they can zoom in or out with the controls at the bottom of the window. If the user clicks within a *hot-spot* (the hot-spots are shown in rectangles for clarity), their view changes to the node containing that view. *Scene taken from* `http://www.apple.com/quicktime/samples/qtvr/multinode-frameset.html`.

The image data from the sampled location $\mathbf{P}_i$ in the environment now encompasses all possible viewing directions, within the constraints of the cylindrical map, and is defined as a *node*. To construct a navigable environment, several such nodes must be created, as well as a method defined for inter-nodal movement. In practice, one can define *hot-spots* within the images to create links in the topological graph

(figure 5.2). The desired result is to obtain a graph composed of nodes which encompass all the distinctive regions in the environment, as well as to provide a means of smooth navigation. That is, if two nodes are chosen which have no overlapping visual information, it would be desirable to have a node in between which would allow a smooth transition. A sample synthetic environment with the vantage points demonstrating these qualities is shown in figure 5.1. It is the automated selection of the nodal positions $\mathbf{P}_i$ which we will now develop further.

5.2. Notation

The *set* of all possible views or images obtainable from a fixed location in the environment can be described as a *viewing sphere* or spherical image. More specifically, for every ray projected from a location in R^3, in a direction along the unit sphere S^2 we can sample an intensity from the environment. This transformation can be expressed as:

$$M_{3D} : R^3 \oplus S^2 \longrightarrow R^+ \tag{5.1}$$

or

$$M_{3D}(x, y, z, \phi, \theta) = i \tag{5.2}$$

where (x, y, z) are spatial coordinates, (ϕ, θ) refer to the orientation of the light ray, and $i \in [0, i_{max}]$ is the intensity observed. This parameterization of light rays is related to the *light ray manifold* defined by Langer and Zucker [**40**] and the Lumigraph [**25**] as described above.

In our particular case, we have a camera mounted on a pan and tilt unit at a fixed location on a mobile robot. For the purposes of this thesis, let us assume that the robot is constrained to a flat floor, and thus we restrict the camera to a parametric

surface (x, y) topologically equivalent to a plane. This limits the origin of each ray to R^2, and we have the idealized 2D observer in a 3D world:

$$M_{2D} : R^2 \oplus S^2 \longrightarrow R^+ \tag{5.3}$$

or

$$M_{2D}(x, y, \phi, \theta) = i. \tag{5.4}$$

A minor variation is the case of an idealized camera which only pans, which is the case for the bulk of image-based VR. Since we are now dealing with a camera, as opposed to a single ray, the result of the transformation is an *image* or a set intensities given by a cone about the camera direction:

$$M_C : R^2 \oplus S \longrightarrow R^n \tag{5.5}$$

or

$$M_C(x, y, \phi) = \mathbf{I} \tag{5.6}$$

where $\mathbf{I}$ now denotes an n pixel image implicitly dependent on the field of view of the camera. Each pixel is also specified by equation 5.4. An entire *spherical* panoramic image $\mathbf{I}_{x,y}$ where each pixel is a ray corresponding to equation 5.4 is given by

$$M_S : R^2 \longrightarrow R^n \tag{5.7}$$

where n is the number of pixels in the image, thus leading to a parameterization of a *set* of images $\mathbf{I}_{x,y}$ whose pixels are specified by $\mathbf{I}_{x,y}(\phi, \theta)$ resembling the notation used in chapter 4.

5.3. Calculating Attention Revisited

We are now able to define a method for evaluating the various vantage points in the environment, in order to determine which will be part of the final VR scene. We extend our previous notion of "interestingness" in a single image from chapter 4 to apply to a viewpoint in the environment. We accomplish this task by applying our operator not to sub-regions of an image, but to the entire cylindrical panorama corresponding to the current (x, y) location in the environment. In other words, every location in the environment will have a measure of "interestingness", or information content. The extrema of this manifold will therefore be the regions which are most conspicuous, and their content will have been evaluated as being furthest from the mean. This is directly related to Koch and Ullman's major selection rule [**35**], with one substantial exception: in their model of selective attention, the feature which enters the central representation must be in the visual field. Because our "visual field" is all-encompassing, we must re-define a shift of attention to be a shift to another region in the environment. In their model, they encourage a shift of the visual field by inhibiting the currently selected feature. Similarly, in our model, the location chosen next is guaranteed to be different from the current chosen location (although it may be arbitrarily close), because we sort the points P_i in the environment by their decreasing interest value. The fact that two "attentional" locations may be an arbitrary distance apart more closely matches the model of attentional shifts presented by Tsotsos et. al.[**70**].

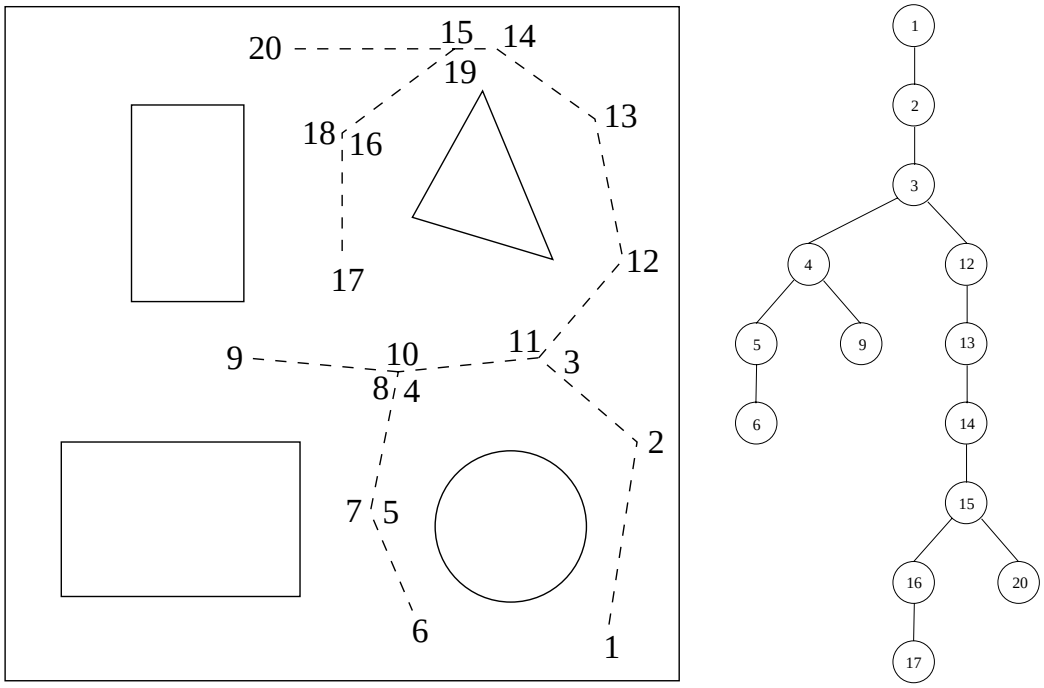
There may also be some additional constraints involved in the selection of locations in the environment. For example, we may want to limit the proximity of adjacent chosen locations, or perhaps force a selection in order to guarantee a smooth traversal through the environment in the VR model. Details of the topology of the nodal graph are presented in section 6.5.

5.4. Sampling & on-line performance

In the image-based mapping problem presented up to this point, we have presupposed that a characterization of the environment (in terms of typical statistics) is available at all times (this would suggest an off-line algorithm). In practice, as the robot moves through the environment it would be advantageous to make decisions when locations are encountered so that there is no need to either acquire and store immense amounts of data, or backtrack to re-visit selected locations to obtain the panoramic images. To do this, nodes must be selected based only on partial information on the statistical distribution of image content over the environment giving rise to an *on-line algorithm*. An on-line algorithm is one that can be used incrementally without a complete *a priori* problem specification. Assuming that the off-line algorithm performs well, we seek an on-line algorithm whose performance is a good approximation of that obtained with the off-line method. Alternatively, consider the vacation snapshot problem defined earlier (chapter 1): if a traveler had only 36 photographs she could take, and was visiting several places she had never seen, where should she take the pictures?

Consider the set of paths (for example hallways) that the robot navigates in a given environment and the locations at which sample views may be acquired. These locations can be used to define nodes (vertices) of a *geometric tree* over the trajectories of the robot (figure 5.3). Such a tree provides a one dimensional description of the trajectory of the robot (as it traverses the tree). In addition, we can index points on the tree by the fraction t of the total traversal already completed when a node is *first* encountered. Thus, the index t associated with a node indicates how much of the total knowledge of the environment is already available.

We can assure that the on-line algorithm exhibits arbitrarily good performance, as compared to the ideal of the off-line algorithm, by permitting the robot to backtrack. We can define the *forward interest* of a point from partial information as



(a) A synthetic environment with a trajectory. The numbers indicate the order in which the trajectory was followed. Note that some locations have several numbers since the robot was back-tracking.

(b) The associated geometric tree. Each node is labelled with the corresponding number in the trajectory when each location is *first* encountered.

FIGURE 5.3. A synthetic environment with a trajectory, and it's associated geometric tree.

$$\mathcal{C}_t(i,j) = |\hat{C}_t - C(i,j)| \tag{5.8}$$

where the subscript t denotes statistics computed from the initial fraction $t \in (0,1]$ of the entire data set. We define *on-line viewpoint selection with α-backtracking* as a variant of the off-line algorithm such that the best K non-overlapping points are selected as the exploration proceeds.

As each point is selected, a corresponding panoramic node is constructed. Interest values are also stored for all other points visited. As the exploration proceeds, t

increases and the forward interest values of previously visited locations may evolve. If a prior unselected point *which is no further back than a fraction α of the current trajectory length* becomes more interesting than one of the K selected points, the robot backtracks and uses it instead of the least interesting of the K points. As will be shown, the performance (in terms of the points selected) of this algorithm approaches the ideal as α approaches one.

We have conducted several simulations in order to experimentally verify the effectiveness of α-backtracking. Two such experiments will be shown here: (1) one with normal distribution random data, and (2) one with few local minima and maxima which is indicative of a simple scene. In (1), a path of length 4000 (units) was generated, and (2) had a length of 500 where each point along each of the paths was given an interest value. The distributions were normal distributions. The points along the paths which were chosen using the on-line algorithm with α-backtracking were then compared against those chosen by the off-line algorithm. In order to compare the two sets of points (we shall call them $\mathbf{P}_{\text{ideal}}$ and $\mathbf{P}_{\text{on-line}}$) chosen along each path, we used the following distance metric:

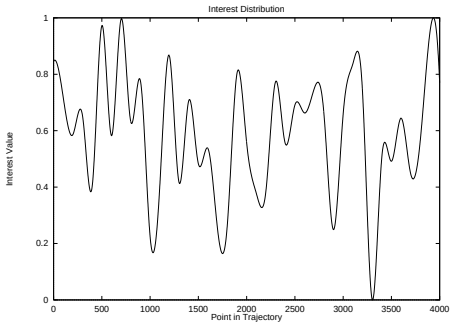
$$D_{\text{ideal,on-line}} = \Sigma_i |\ C(\mathbf{P}_{\text{ideal}_i}) - C(\mathbf{P}_{\text{on-line}_i})\ | \tag{5.9}$$

In other words, the measure of the effectiveness of the on-line algorithm using α-backtracking was the sum of the distances between each point in the off-line set and it's nearest neighbor in the on-line set in terms of their distinctiveness measure. This is analogous to the vacation snapshot problem: of two sets of photographs taken by different people who have traveled in the same areas (paths), which set of photographs is more interesting?

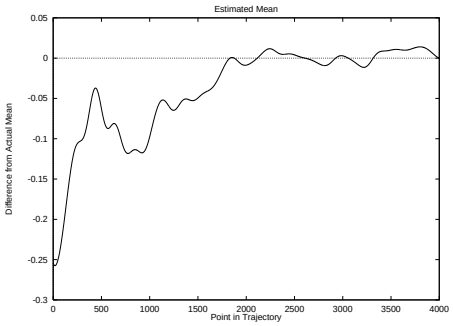
The results of the experiments are illustrated in figures 5.4, and 5.6. Note that the interest values along the path were inside the range $[0, 1]$, yet the *sum* of the differences of interest values from the resulting on-line choices compared to those chosen in the off-line algorithm is very low even for small values of α.

The number of points chosen (K) in the experiment for (1) was 300, and (2) was 15. When there is no proximity constraint, a small value of K compared to the total path length could lead to all of the choices arising in one area, thus providing non-intuitive results. Examples of the selections made along the paths are shown in figures 5.5, and 5.7.

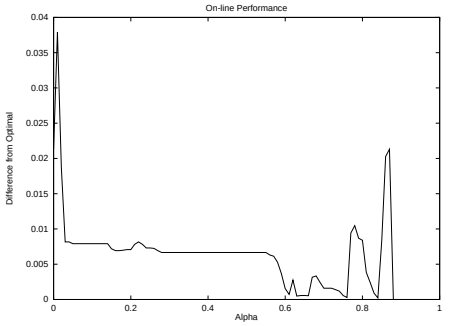
From these results, we can see that for moderate values of α, the on-line algorithm with α-backtracking can achieve results which are close to those of the off-line algorithm, without too much added path length (from the actual backtracking). This is a positive result since it allows the robot to only retain the image data for the top K points as the exploration proceeds, thus greatly reducing the needed storage space.



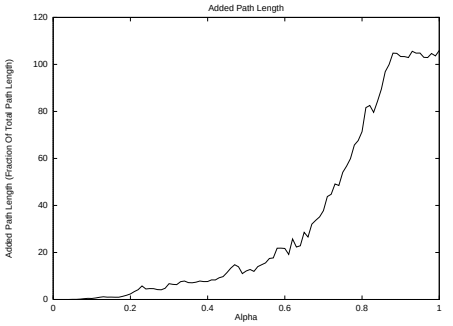
(a) The normal distribution random interest values along the trajectory.



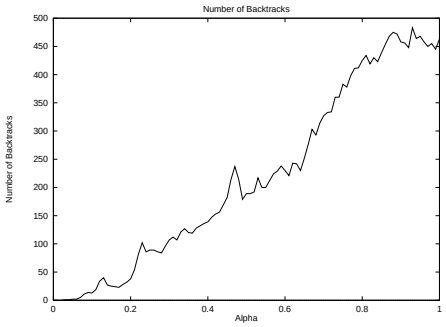
(b) The difference between the estimated mean and the actual mean as the trajectory is followed.



(c) A graph of the distance between the point sets chosen using the on-line and off-line algorithms for various values of α.

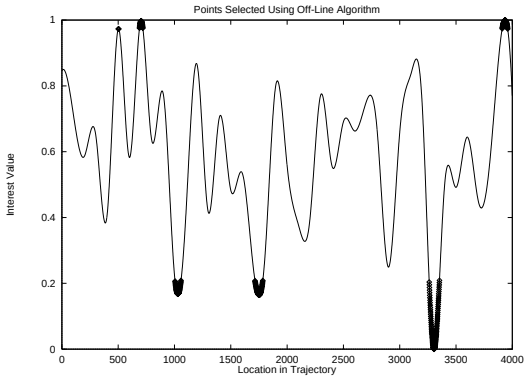


(d) A graph of the increased path length incurred due to backtracking for various values of α.

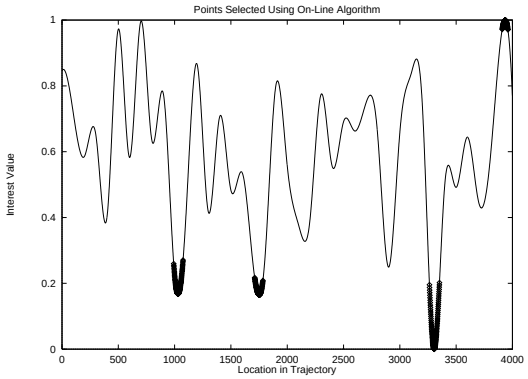


(e) A graph of the number of backtracks for various values of α.

FIGURE 5.4. Graphs representing the results of the α-backtracking simulation on normal distribution random interest data.

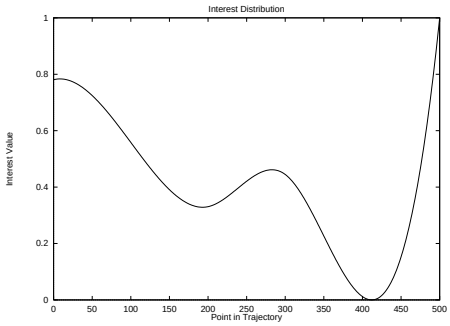


(a) Points selected along the path using the off-line algorithm.

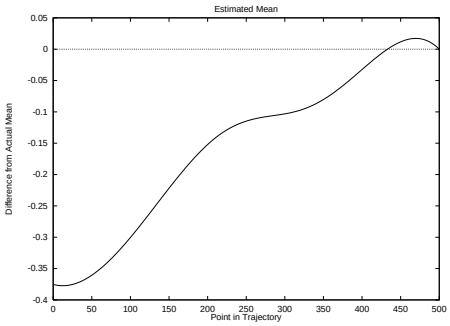


(b) Points selected along the path using the on-line algorithm with $\alpha = 0.15$

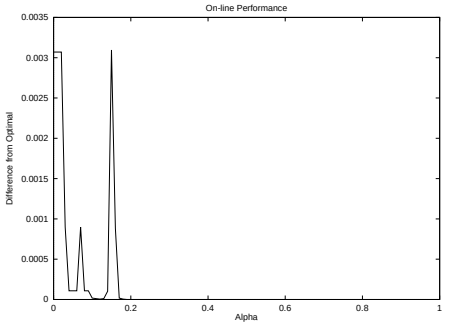
FIGURE 5.5. The points selected along the trajectory (shown in figure 5.4) are indicated by bold regions of the graph. The off-line selections are shown in (a) and the on-line with α-backtracking selections ($\alpha = 0.15$) are shown in (b). In the on-line case, the robot backtracked 25 times for a total added trajectory equal to the fraction 1.029493 of the total trajectory length. The error in terms of interest value was 0.007187. In each case, 300 points (out of a total of 4000 possible) were chosen.



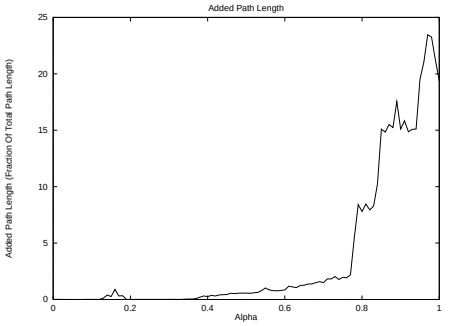
(a) The normal distribution interest values along the trajectory.



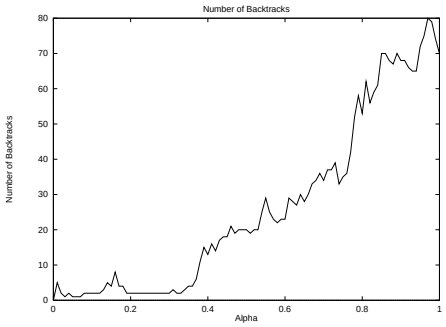
(b) The difference between the estimated mean and the actual mean as the trajectory is followed.



(c) A graph of the distance between the point sets chosen using the on-line and off-line algorithms for various values of α.

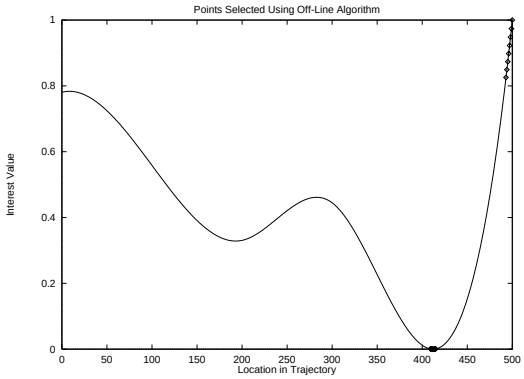


(d) A graph of the increased path length incurred due to backtracking for various values of α.

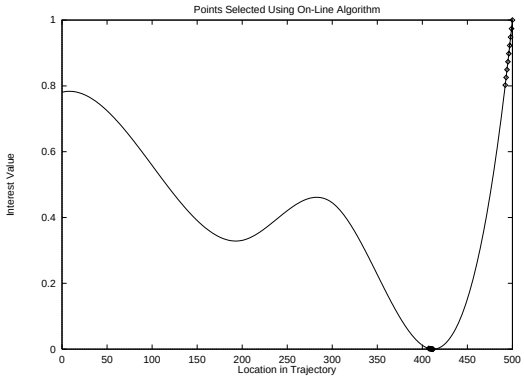


(e) A graph of the number of backtracks for various values of α.

FIGURE 5.6. Graphs representing the results of the α-backtracking simulation on normal distribution interest data.



(a) Points selected along the path using the off-line algorithm.



(b) Points selected along the path using the on-line algorithm with $\alpha = 0.1$

FIGURE 5.7. The points selected along the trajectory (shown in figure 5.6) are indicated by bold regions of the graph. The off-line selections are shown in (a) and the on-line with α-backtracking selections ($\alpha = 0.1$) are shown in (b). In the on-line case, the robot backtracked 2 times for a total added trajectory equal to the fraction 0.003992 of the total trajectory length. The error in terms of interest value was 0.025223. In each case, 15 points (out of a total of 500 possible) were chosen.

CHAPTER 6

Exploration and Modeling

The hardware system used for environment viewpoint selection was composed of a Nomadic Technologies Nomad 200 mobile robot with an on-board computer, and a NTSC CCD camera mounted in a 2-DOF Directed Perception pan and tilt unit on top of the robot. In practice, almost any type of mobile robot could be used, provided that the camera can be mounted at a height equivalent to that of an average human observer. This last constraint allows the sizes of objects in the final VR scene to be interpreted correctly in terms of a traditional human vantage point. In contrast, if the camera were placed at a height of 36", everything in the scene would look normal to a three year old child, but an adult might believe that everything in the scene was larger than it was in reality[1].

There are several software components that comprise the entire viewpoint selection system (figure 6.1):

- robot exploration: an environment-specific algorithm moves the robot through the environment,
- image acquisition: images are acquired at each vantage point to sample all possible orientations,

[1]Assuming that there were no objects that could be used to cue the observer to the actual size of objects in the scene.

- attention processing: a statistical measure of distinctiveness is computed and images are selected as locations where attention should be focussed,
- image stitching: sets of images from selected locations are joined together in a single cylindrical mosaic,
- nodal graph creation: a topological map is created and used to connect the cylindrical images producing a user interface in which a user can pan, tilt, zoom or translate (to an adjacent location).

The specifics of these subsystems are outlined below.

The final component in the system is a software package from Apple Computer Inc. which combines the stitched panoramic photographs and the topological representation from the nodal graph component to form a multi-node *QuickTime VR* movie[2].

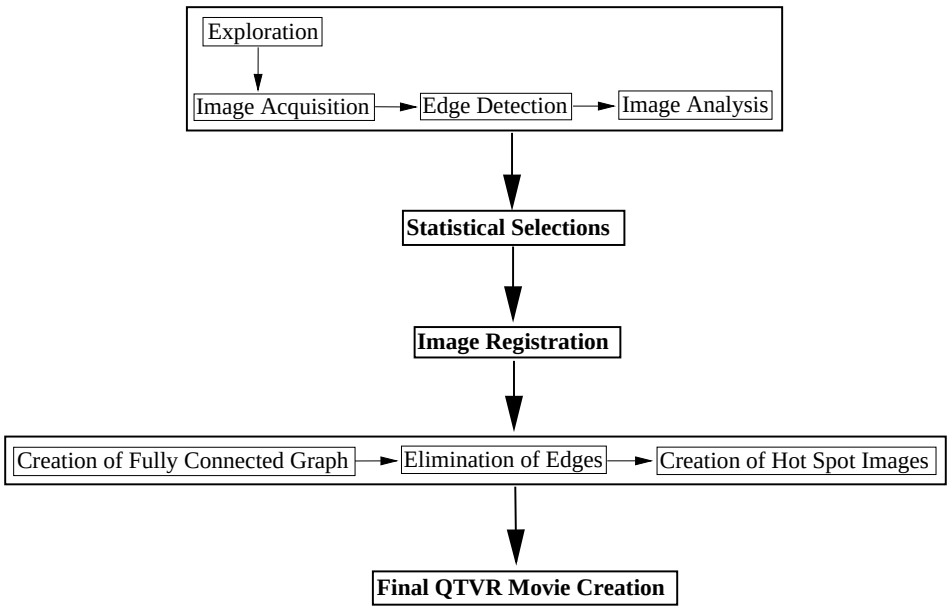


FIGURE 6.1. Architecture of the viewpoint selection system. The main boxes indicate the various software components, while the inner boxes indicate sub-processes within each component.

[2]The package is called *QuickTime VR Authoring Studio*, and may be obtained from Apple Computer Inc.

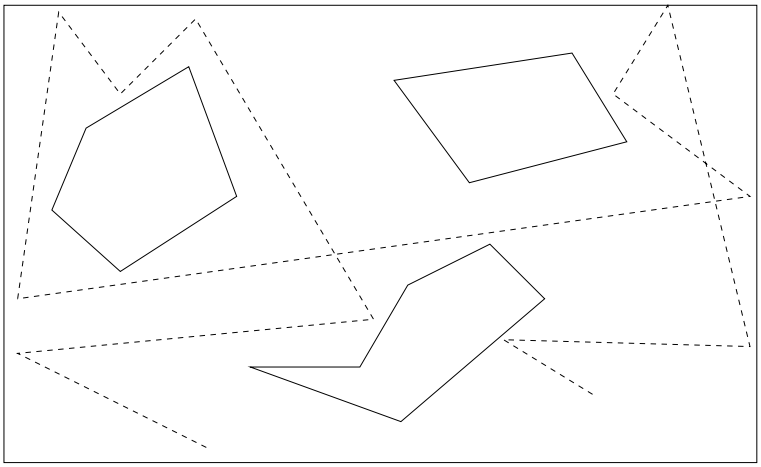


FIGURE 6.2. An example of how the "bouncing ball" exploration algorithm may proceed on a particular environment. The dotted line represents the trajectory of the robot.

6.1. Robot Exploration

In principle, any exploration algorithm may be used with the viewpoint selection system provided the environment is fully sampled, that is, that the robot visits each location in the environment. To exemplify this independence from any particular exploration algorithm, the system was designed using a "plug-in" architecture. This type of architecture allows the user to decide which exploration algorithm should be used for the current environment. The motivation for the plug-in exploration architecture was based on the potential variability in environments one will encounter and the fact that they might mandate environment-specific strategies as mentioned in section 2.1.

In practice, we have used a simple algorithm akin to a "bouncing ball[3]": the robot travels in a straight path until it is obstructed at which point it rotates by a random angle until it can once again move forward (figure 6.2). This is not the true behavior of a bouncing ball, however, we have chosen to rotate by a random angle due to the difficulty of robustly calculating the angle of incidence with a surface using sonar sensors. Although this algorithm does not exploit the layout of the environment it still

[3]We have also developed and tested additional exploration algorithms but they are outside the scope of this thesis.

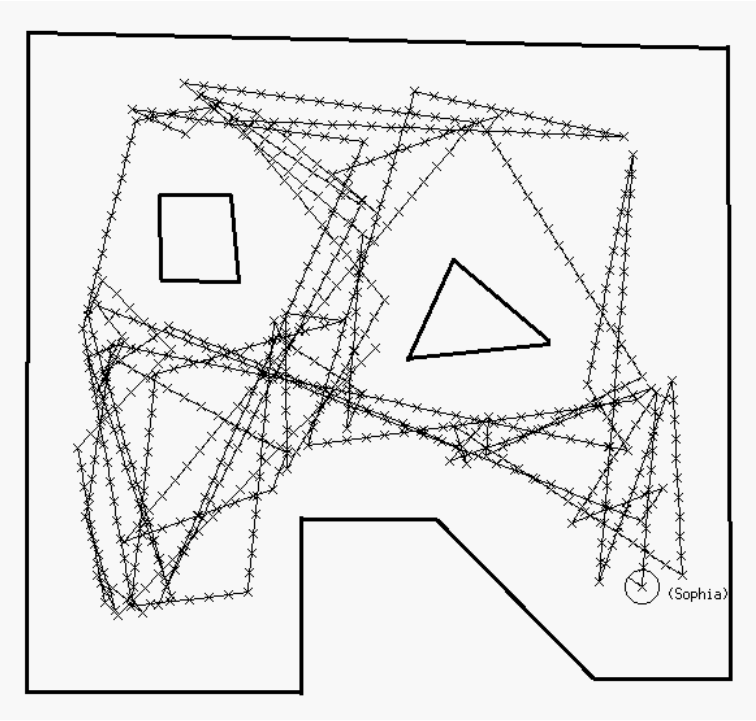


FIGURE 6.3. An example of how the bouncing ball exploration algorithm proceeded in simulation. Note that the robot maintained a predetermined safe distance from any obstacles in the environment.

manages to cover the free space quite well [**4**]. The latter is illustrated in figure 6.3: although the path is sometimes redundant, almost all of the free space is covered. Note that the robot maintained a predetermined safe distance from the surfaces in the environment.

6.2. Image Acquisition

As the robot explores the environment, video images are collected using a camera mounted on the top of the mobile robot. In order to minimize warping effects during stitching, we rotate the camera about its optical center or nodal point. To preclude the robot itself appearing in the images, the pan and tilt unit (PTU) is mounted above the front face of the robot (figure 6.4). This constrains the acquisition of the images to two half-cylinders since the robot itself would appear in the images of the back half. We acquire the images covering a span of 180 degrees with the PTU, we

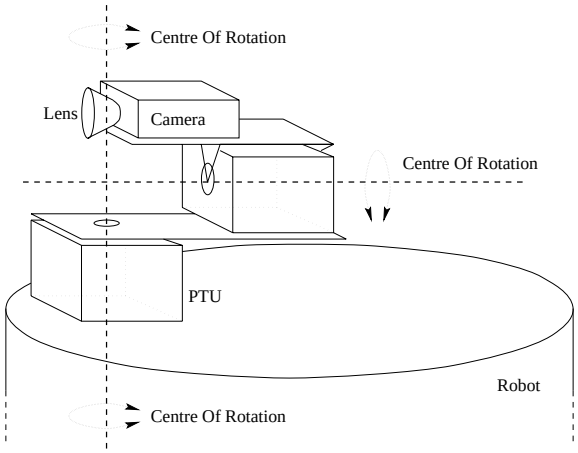
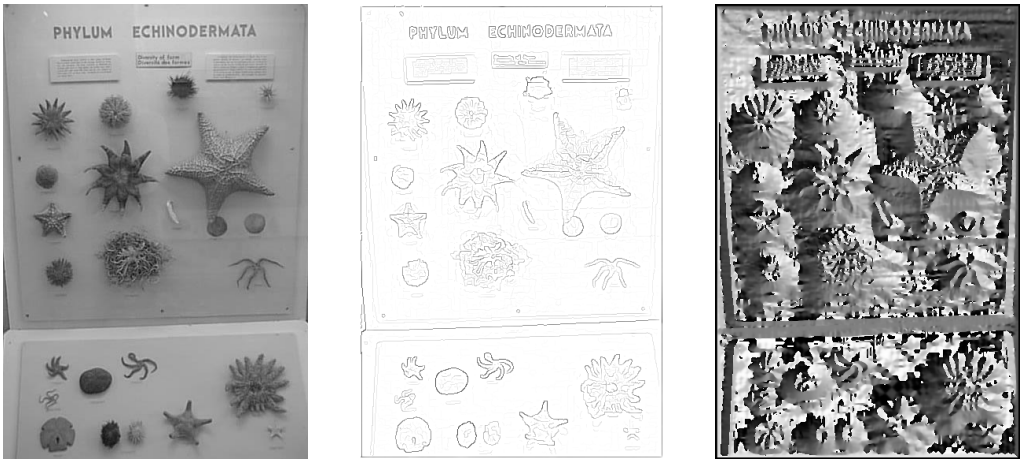


FIGURE 6.4. Camera position on the mobile robot. *Figure courtesy of Philippe Ciaravola* [**13**].

then rotate the robot by 180 degrees and translate the robot by its diameter, and then acquire the remaining 180 degrees with the PTU. This method provides minimal error about the optical center of the camera, and removes possible obstructions posed by the robot itself [**13**].

6.3. Attention Processing

The first phase of the image analysis process performs edge detection (the process of finding edges in an image) on the images acquired using the algorithm formulated by Canny [**9**], and later improved by Deriche [**16**]. This process returns an edge map, and an orientation map. The edge map is an array where each edge element corresponds to the strength of the edge in the original image. The orientation map is an array of the orientations of each of the edge elements in the edge map (figure 6.5). The interest functions are then evaluated on the edge and orientation maps as outlined in chapter 4. The values for the resulting images are sorted in order of decreasing absolute deviation from the mean, and the top n points (representing the extrema) in the density/orientation map(s) are chosen as the locations which will be part of the final graph.



(a) Input image (b) Edge map (c) Orientation map

FIGURE 6.5. An example input image as well as its edge and orientation maps. The intensity in the orientation image (c) corresponds to the angle of the edge element at that location.

6.4. Image Registration

To produce a panoramic image at each location, adjacent images taken with the same (x, y) position but different orientations must be fused together to produce a single cylindrical image. To solve this "mosaicing" problem we use cross correlation to find the best correspondence (figure 6.6). Observe that the problem is simplified by the fact that the images are acquired using only rotations about a fixed nodal point [**8, 63**]. Once the overlap between to adjacent images is found, the intensities are blended (averaged) to remove any seam which may be present [**13**]. An example of registering several images can be seen in figure 6.7.

6.5. Graph Creation

Because we wish to create a multi-node VR *scene*, the relationship between the panoramic photographs (environment maps) must be established, and a facility provided for the user to move between them. As mentioned earlier, *QuickTime VR* provides a facility for moving between nodes called *hot-spots* (figure 5.2). These are

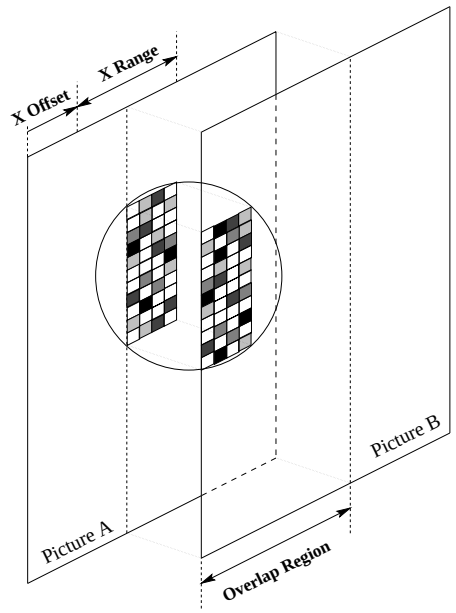


FIGURE 6.6. Finding the correlation between image A and B. The difference of intensities of the overlapping regions of the images are minimized. *Figure courtesy of Philippe Ciaravola* [**13**].

encoded using a mask over the panoramic image, with the value of the mask at the current location of the user's pointer determining which node will next be visited should the user to decide to move. In order to automate the creation of these masks, we construct a fully connected graph representing the selected locations in the virtual environment. Because each image is associated with a known pose (x, y, θ) in the plane[4] we are able to determine the arc lengths and positions in the mask which correspond to other nodes in the graph. Although localization errors are unavoidable even with correction techniques, we only require approximate positions to construct the topological representation.

Consider figure 6.8: assuming the radius of each panoramic image is fixed[5] we can compute the intersection range I_{AB} of the panorama B on the panorama A (in A's local orientation frame):

[4]We assume planar environments, although our approach could be readily extended to 3D environments.
[5]In practice the radii are fixed to a certain number of vertical scan lines (pixels).

(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4

(e) The resulting stitched image.

FIGURE 6.7. An example of the result of registering four adjacent images. *Figure courtesy of Philippe Ciaravola* [**13**].

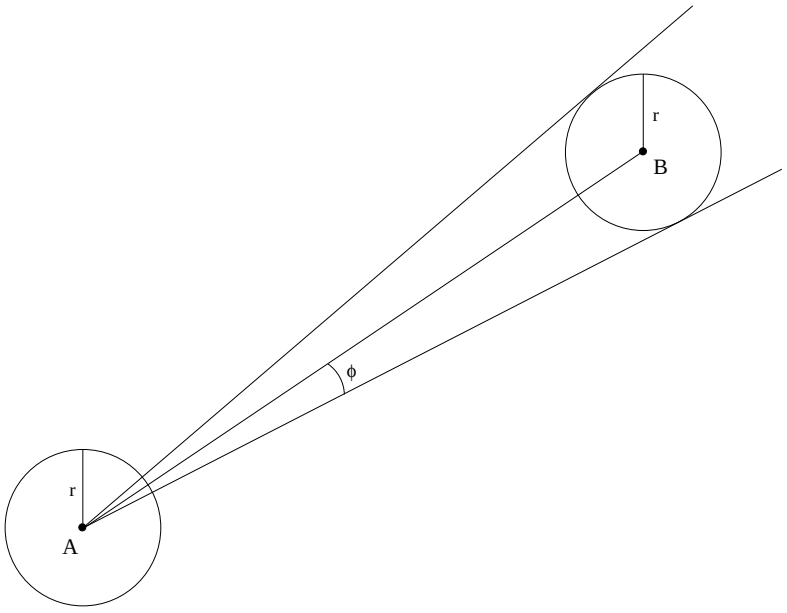


FIGURE 6.8. Calculating arc length of viewpoint B on viewpoint A.

$$\phi = \tan^{-1}\left(\frac{\sqrt{(A_x - B_x)^2 + (A_y - B_y)^2}}{r}\right) \tag{6.1}$$

$$\gamma = \tan^{-1}\left(\frac{B_y - A_y}{B_x - A_x}\right) \tag{6.2}$$

$$I_{AB} = [\gamma - \phi - A_\theta, \gamma + \phi - A_\theta] \tag{6.3}$$

where r is the radius of the panoramas.

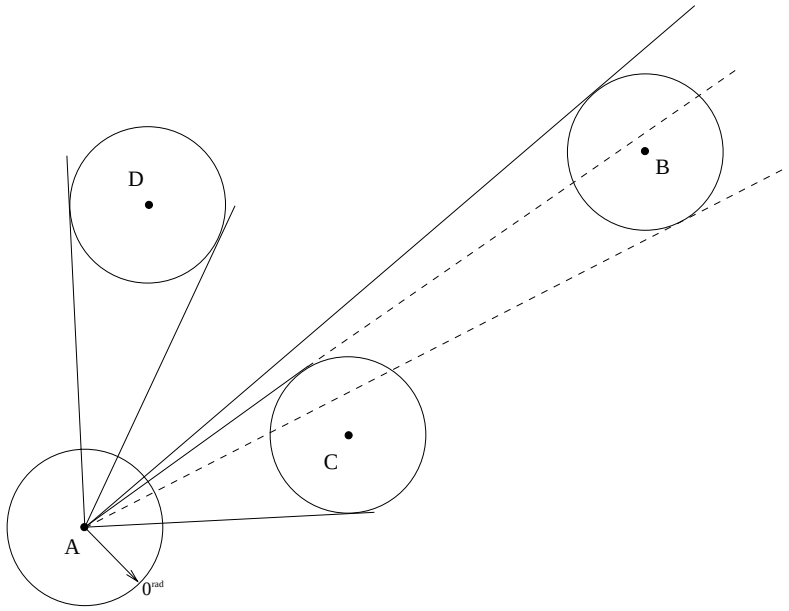


FIGURE 6.9. Viewpoint A's view of location B is obscured by intermediate location C.

A complication arises if there are viewpoints which occlude other viewpoints in the scene (figure 6.9). To account for this, we must process adjacent nodes in order of their increasing distance from the source node. If the arcs of two nodes overlap, we only keep the non-overlapping remainder of the arc for the occluded node. The resultant mask generated following this algorithm on the graph depicted in figure 6.9 is shown in figure 6.10.

This pre-computation provides a fully connected graph; however, since we do not build a map of the environment in the current exploration model, some adjacent nodes may be occluded by objects in the environment. We do not consider this to be

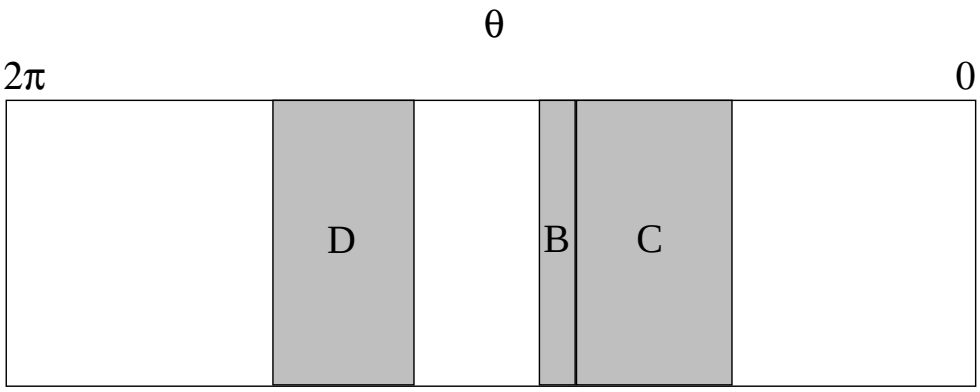


FIGURE 6.10. The connectivity mask associated with viewpoint A as depicted in figure 6.9.

a limitation – edges in the graph could be easily removed by the user before the final model synthesis.

6.6. Model Synthesis

At this point, the panoramic photographs representing the selected locations in the environment, as well as the *hot-spot* masks are used as input to the QuickTime VR Authoring Studio. This software produces an image-based environment in which a user can navigate using a "point-and-click" metaphor [**15**]. This VR interface technology can be used as a stand-alone application, or can be easily embedded within documents, such as web pages, or presentations.

CHAPTER 7

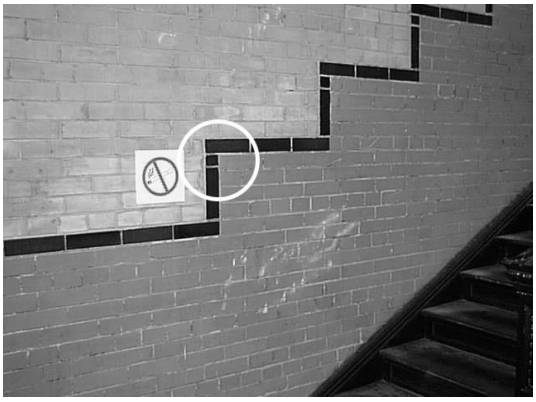
Experimental Results

7.1. Single Images

For the purposes of our single image experiments, we acquired several photographs of natural scenes using a digital camera. The goal of these experiments was to see how closely the points chosen in the images matched the points which would attract the focus of a human observer. In order to demonstrate the advantages of our interest operator, we chose photographs which had interest points which were well defined semantically, but which contained substantial texture, thus allowing them to be potentially confusing. As an example, consider the image displayed in figure 7.1: there is a lot of edge information present in the image due to the brick texture, however, the non-smoking sign, as well as the line dividing the two colors of brick are clear semantic tokens.

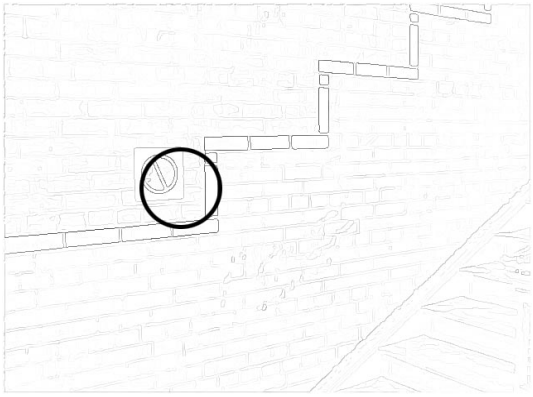
The approach we developed for this analysis was based on convolving the images using the kernels outlined in section 4, and subsequently sorting the resulting interest points $\mathbf{P}_i$ by their descending interest values. The top choices of the density, orientation, and the combined density-orientation operators are shown on sample images in figures 7.1, 7.3 and 7.4. For these images, the size of the operator was 100×100, $\sigma = 50$, $\gamma = 0.5$, and $\omega = 5$. The size of the operator was inspired by the approximate size of the semantic tokens in the images – each token is roughly between 50, and 120 pixels in size. Although in this particular example our choice is of operator size

(a) Density selection

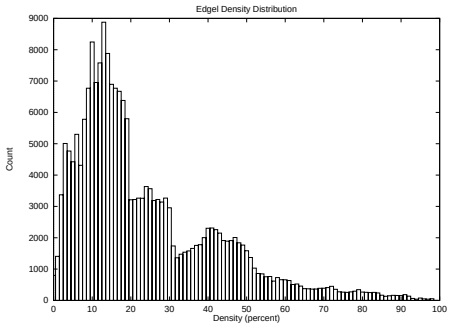


(b) Orientation selection

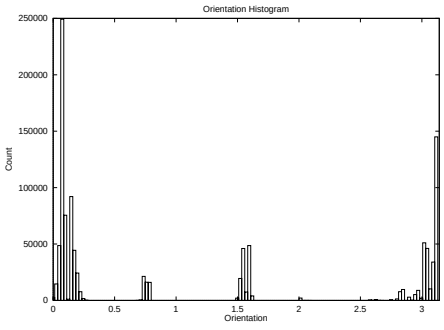
(c) Combined operator selection



(d) Edge map



(e) Edgel density distribution



(f) Edgel orientation distribution

FIGURE 7.1. Results of interest operator on a sample textured 2-D image. The associated interest map is shown in figure 7.2

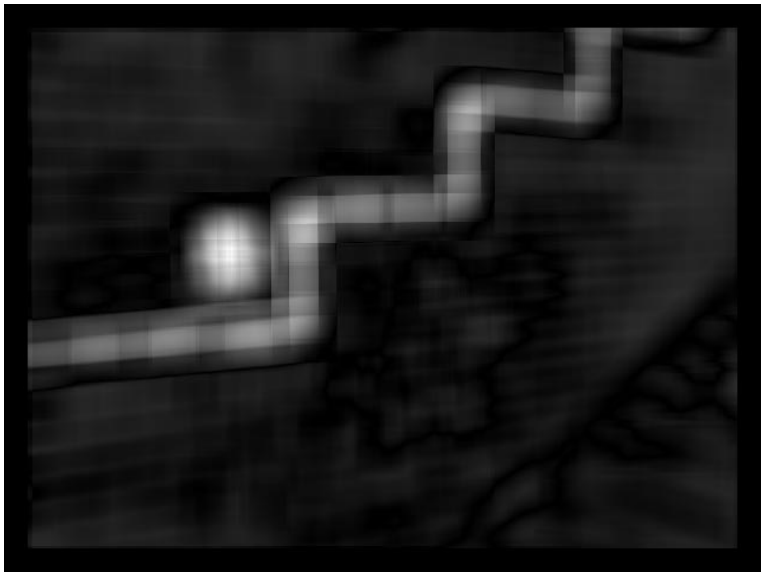


FIGURE 7.2. The interest map associated with figure 7.1. The intensities correspond to the interest values (darker is lower interest).

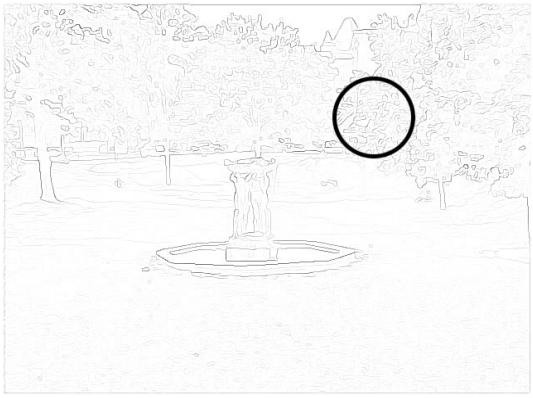
(a) (b)

FIGURE 7.3. Top two choices for another 2-D image.

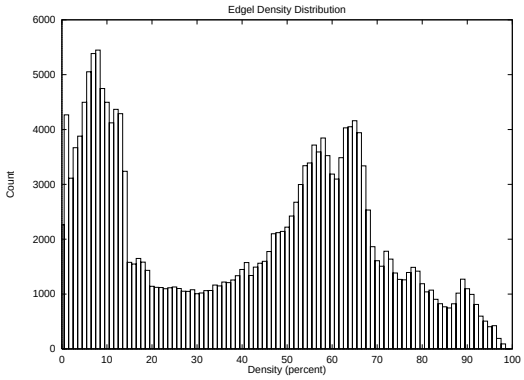
is ad hoc, our results suggest that the operator performs well at multiple scales, as is shown in figure 7.5.

The locations selected by the operator were clearly appropriate (figures 7.1,7.3 and 7.5). That is, they closely match what attracts the attention of a human observer.

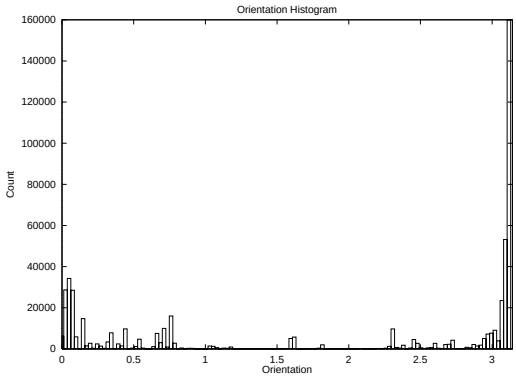
(a) Density selection.



(b) Edge map with density selection.



(c) Edgel density distribution.



(d) Edgel orientation distribution.

FIGURE 7.4. An example of a non-intuitive selection.

We have included one example where the operator chooses a non-intuitive point of maximum interest; a point on a background texture rather than a sculpture in the foreground (figure 7.4). In this example, the edge density distribution is not uni-modal, which is one of the assumptions for the correct behavior of our operators. Perhaps more importantly, this non-intuitive choice arises for two reasons: (1) the point of maximum interest is selected independent of functional or semantic attributes, and (2) the scale of the edge operator leads to the response being dominated by background texture (leaves and grass), rather than edges that arise from geometric structure. It may be useful to classify and detect such "irrelevant" texture, using, for

FIGURE 7.5. An example of the interest operator's selections at multiple scales.

example, the classification mechanism described in [**17**]. Note that the sculpture was chosen when the parameters of the edge operator, as well as the size of the Gaussian were modified accordingly.

7.2. Environmental

7.2.1. Overview. The experimental data for environmental selections discussed in this thesis was obtained at the Canadian Centre for Architecture, located in Montréal. Due to the fragile nature of the environment, the exploration was carried out manually. An approximate floor plan of the gallery as well as the trajectory followed by the robot can be seen in figure 7.6.

The gallery floor contained pedestals holding exhibits encased in glass, which were scattered about the environment. The walls contained many exhibits, usually of uniform size, spread around a room.

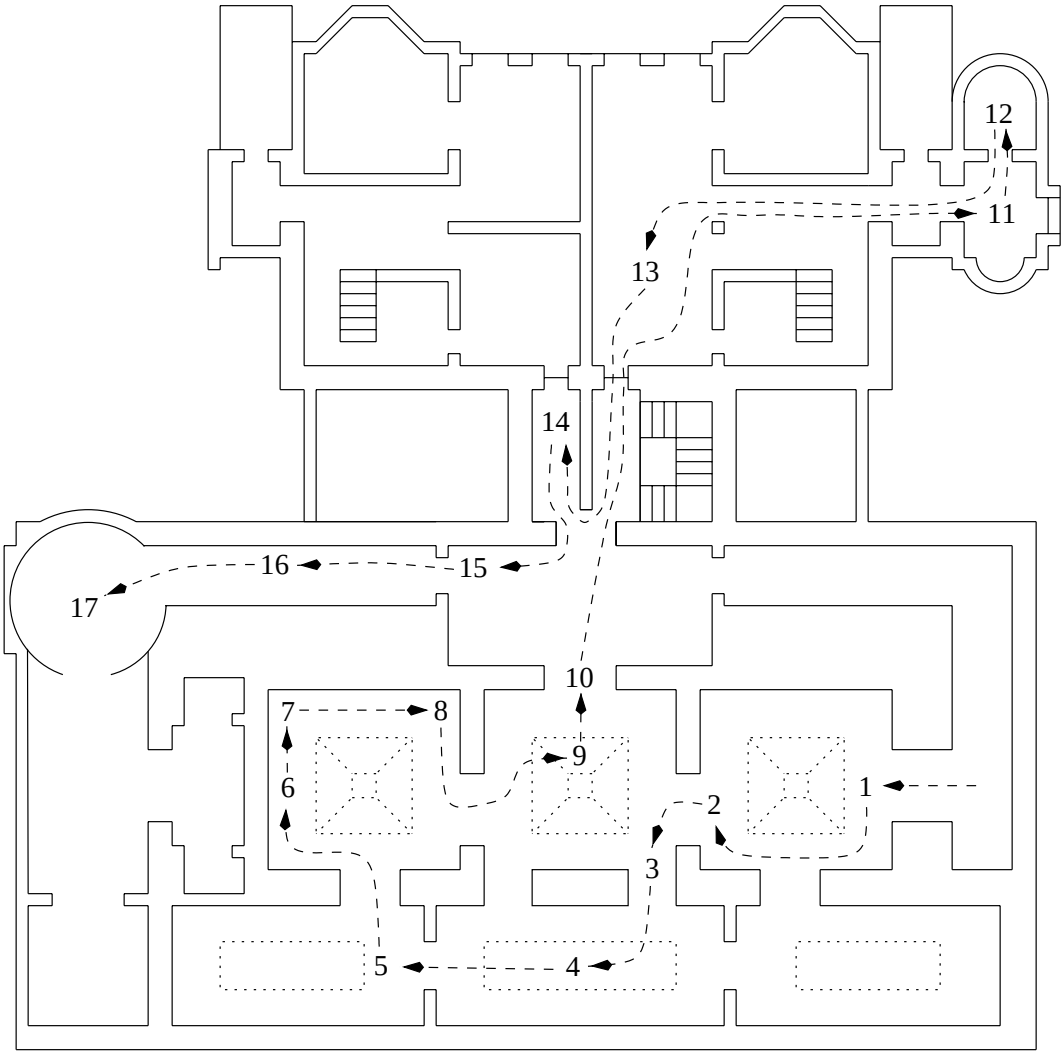


FIGURE 7.6. Approximate map of the gallery where the experimental data was collected. The numbers represent the locations in the environment where a set of images was acquired. The dotted lines represent the trajectory of the robot.

The path followed by the robot covered several rooms of the gallery, and included 3 rooms which were not part of the exhibit. The robot acquired 36 images at each of the 17 different locations along the trajectory. These images were processed as outlined in section 6. While the processing could, in fact, have taken place on-line in real time, image processing was completed off-line to allow alternative strategies to be evaluated on the same data set.

(a) (b) (c)

(d) (e) (f)

FIGURE 7.7. Selections made by the viewpoint selection system using the density operator alone.

7.2.2. Results. The individual locations (views) chosen in the panoramic images according to the density and orientation operators are shown in figures 7.7, and 7.8. The photographs in figure 7.7 demonstrate the effectiveness of the density operator. The images whose edgel density variance is highest show the areas in the gallery which were substantially different from the remainder of the gallery. Furthermore, the candidate with the lowest variance shows a region of a wall which

(a) (b) (c)

(d) (e) (f)

FIGURE 7.8. Selections made by the viewpoint selection system using the orientation operator alone.

contains little more than a brick-like texture (figure 7.10(a)). The effectiveness of the edgel orientation operator is illustrated in figure 7.8. Here, the candidates containing multiple curves such as the pattern in the painting, the frosted glass pattern in the door, and the marble fireplace, have the highest variance. This is due to the fact that the remainder of the gallery is dominated by rectilinear structures, as illustrated by the candidate whose variance was lowest (figure 7.10(b)): a series of photographs in

(a) (b) (c)

(d) (e) (f)

FIGURE 7.9. Selections made by the viewpoint selection system using the combined density-orientation operator.

frames. Unfortunately, it is difficult to convey the appearance of the entire environment in paper form - it is important to note that the views shown here represent less than three percent of the views seen in the environment.

The top candidates from the combined density-orientation operator provide an interesting and positive result – the top candidate in the distribution is the image *just between* the top density candidate, and the second orientation candidate. Equally

(a) (b) (c)

FIGURE 7.10. Selections placing at the bottom of the interest distribution.

appealing is the operator's choice representing minimal variance – the last image is convincingly boring! Figure 7.11 shows the entire panoramic images corresponding to the top five viewpoints in the environment chosen using the combined density-orientation operator.

(a) (b) (c) (d) (e)

FIGURE 7.11. Fully stitched cylindrical panoramic images corresponding to the selected views shown in figure 7.9.

CHAPTER 8

Conclusion

In this thesis, we have presented a comprehensive approach to the graphical modeling of arbitrary environments. Using an exploring robot, we have shown a method for constructing a navigable collection of panoramic images that captures the appearance of an environment in the form of an image-based virtual reality.

The key issue becomes one of how to automatically select the viewpoints in the environment to be used in the final model. In our work, these viewpoints are selected using an interest operator which selects viewpoints whose characterization in terms of visual features is atypical. This operator is based on the underlying edge structure of the scene, as the latter is robust in terms of variations in illumination, as well as camera parameters. More specifically, our operator is based on the edge element density distribution, and the edge element orientation distribution. Local areas which differ substantially from the mean in terms of either edge element density, or orientation are considered interesting, and evaluated for candidacy in the final VR scene. While our results are highly satisfying, there remain several interesting issues to be resolved. Foremost among these is the need for a formal characterization of performance for such an approach. Especially since we can now achieve effective and useful results, it is important to be able to evaluate alternative approaches in a consistent and reproducible manner.

A separate, more technical issue, is that our characterization of interesting views explicitly ignores the spatial sampling of the environment. For many real tasks, it may be desirable to achieve a somewhat uniform coverage of the environment (in terms of stored views). This seems like it can be readily achieved in practice, for example by constraining the minimum and maximum proximity of the stored sample views.

In some applications, especially those related to entertainment, it may be desirable to synthesize a view from a location for which no sampled images have been stored. This relates to *image based rendering* in which sample images (without a complete depth map) are used to render views from various locations. In some applications image-based rendering may eventually provide enhanced realism.

In this thesis, we have only touched briefly on the issue of scale. In ongoing work we are exploring this issue more fully. In practice, it appears that while a single-scale operator works surprising well for an environment with a limited depth range, interesting views should be selected across multiple spatial scales. This, in turn, suggests that it may be desirable to classify regions of observed views with respect to their content: textures of different types, geometric structures, or shading phenomena.

The interest operators developed in this thesis were intentionally designed to be environment and task-independent. While the results achieved using this model were satisfying, there are certain scenarios where one may desire an interest operator catered more specifically to the task. For example, in an art gallery, it may be more appropriate to develop a specialized operator for selecting paintings.

A final issue is the relationship between the active environment exploration carried out by the robot and the set of interesting locations selected. At present, our approach uses an exploration mechanism decoupled from viewpoint selection. A related interest operator is used in our lab, however, to select landmarks that can be used for robot localization [**61**]. The use of the interest operator to explicitly drive

exploration is something that might be of value in certain task domains, and we are exploring it further.

REFERENCES

[1] Paul G. Backes, *The mars sojourner rover*, IEEE Robotics and Automation **4** (1997), no. 3.

[2] Paul G. Backes, Kam S. Tso, and Gregory K. Tharp, *Mars pathfinder mission internet-based operations using WITS*, Proceedings of the IEEE International Conference on Robotics and Automation (Leuven, Belgium), vol. 1, May 1998, pp. 284–291.

[3] Tucker Balch and Ronald C. Arkin, *Communication in reactive multiagent robotic systems*, Autonomous Robots **1** (1994), no. 1, 27–52.

[4] Paul Beame, Allan Borodin, Prabhakar Raghavan, Walter L. Ruzzo, and Martin Tompa, *Time-space tradeoffs for undirected graph traversal by graph automata*, Information and Computation **130** (1996), no. 2, 101–129.

[5] J. Blanc and R. Mohr, *Towards fast and realistic image synthesis from real views*, Proceedings of the 10th Scandinavian Conference on Image Analysis (Lappeenranta, Finland), June 1997, pp. 455–461.

[6] Marc Bolduc, Eric Bourque, Gregory Dudek, Nicholas Roy, and Robert Sim, *Autonomous exploration: An integrated systems approach*, Proceedings of the AAAI Conference on Artificial Intelligence (Providence, RI), AAAI Press/MIT Press, July 1997, pp. 779–780.

[7] J. Borenstein, H. R. Everett, and L. Feng, *Where am I? Sensors and methods for mobile robot positioning*, Tech. report, University of Michigan, April 1996.

[8] Eric Bourque, Gregory Dudek, and Philippe Ciaravola, *Robotic sightseeing - a method for automatically creating virtual environments*, Proceedings of the IEEE International Conference on Robotics and Automation (Leuven, Belgium), vol. 4, May 1998, pp. 3186–3191.

[9] John F. Canny, *A computational approach to edge detection*, Transactions on Pattern Analysis and Machine Intelligence **8** (1986), no. 6, 679–698.

[10] Senchang Eric Chen, *QuickTime VR – An image based approach to virtual environment navigation*, Proceedings of the ACM SIGGRAPH (New York), ACM, 1995, pp. 29–38.

[11] Senchang Eric Chen and L. Williams, *View interpolation for image synthesis*, Proceedings of the ACM SIGGRAPH (New York), ACM, 1993, pp. 279–288.

[12] Howie Choset, Keiji Nagatani, and Alfred Rizzi, *Sensor based planning: Using a homing strategy and local map method to implement the generalized voronoi graph*, Proc. SPIE Conference on Mobile Robotics (Pittsburgh, PA), 1997.

[13] Philippe Ciaravola, *An automated robotic system for synthesis of image-based virtual reality*, Tech. Report CIM-TR-97-12, Centre for Intelligent Machines, McGill University, 1997.

[14] James J. Clark, *Spatial attention and latencies of saccadic eye movements*, Vision Research (In Press).

[15] Apple Computer, *QuickTime VR 2.0 authoring tools suite*, Apple Computer, 1997.

[16] R. Deriche, *Using canny's criteria to derive a recursively implemented optimal edge detector*, International Journal of Computer Vision **1** (1987), no. 2.

[17] Benoit Dubuc and Steven W. Zucker, *Indexing visual representations through the complexity map*, Proc. of the 5th ICCV (Cambridge, MA) (IEEE Computer Society, ed.), June 1995, pp. 142–149.

[18] Gregory Dudek, *Environment mapping using multiple abstraction levels*, Proceedings of the IEEE **84** (1996), no. 11.

[19] Gregory Dudek and Michael Jenkin, *Computational principles of mobile robotics*, Cambridge University Press, 1998.

[20] Gregory Dudek, Michael Jenkin, Evangelos Milios, and David Wilkes, *Robotic exploration as graph construction*, IEEE Transactions on Robotics and Automation **7** (1991), no. 6, 859–865.

[21] G. D. Dunlap and H. H. Shufeldt, Dutton's Navigation and Piloting, 557–579, Naval Institute Press, 1974, pp. 557–579.

[22] J.H. Elder and S. W. Zucker, *Computing contour closure*, Proc. 4th European Conference on Computer Vision (Cambridge, UK), vol. 2, 1996, pp. 399–412.

[23] Alberto Elfes, Autonomous Robot Vehicles, ch. Sonar-based real-world mapping and navigation, Springer, Berlin, 1990.

[24] J.A. Feldman, *Dynamic connections in neural networks*, Biological Cybernetics **46** (1982), 27–39.

[25] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen, *The lumigraph*, Proceedings of the ACM SIGGRAPH, August 1996, pp. 43–54.

[26] Rick Gurnsey and Roger Browse, *Asymmetries in visual texture discrimination*, Spatial Vision **4** (1989), 31–44.

[27] B. K. P. Horn, *Robot vision*, McGraw-Hill, New York, 1986.

[28] B. Julesz, *Visual pattern discrimination*, IRE Transactions on Information Theory **IT-8** (1962), 84–92.

[29] ———, *Early vision and focal attention*, Review of Modern Physics **63** (1991), no. 3, 735–772.

[30] Sing Bing Kang, *A survey of image-based rendering techniques*, Tech. Report CRL 97/4, Digital Equipment Corporation Cambridge Research Laboratory, August 1997.

[31] Sing Bing Kang and Pravan K. Desikan, *Virtual navigation of complex scenes using clusters of cylindrical panoramic images*, Tech. Report CRL 97/5, Digital Equipment Corporation Cambridge Research Laboratory, September 1997.

[32] Michael F. Kelly and Martin D. Levine, *A sampling strategy using multi-scale annular operators*, Tech. Report CIM-93-20, Centre for Intelligent Machines, McGill University, 1994.

[33] ———, *Symmetric enclosure*, Tech. Report CIM-93-01, Centre for Intelligent Machines, McGill University, 1995.

[34] A. Klinger, *Patterns and search statistics*, Optimizing methods in statistics (J. Rustagi, ed.), Academic Press, New York, 1971, pp. 303–337.

[35] Christof Koch and Shimon Ullman, *Shifts in selective visual attention: Towards the underlying neural circuitry*, Human Neurbiology **4** (1985), 219–227.

[36] D. E. Koditschek, *Robot planning and control via potential functions*, The Robotics Review 1 (O. Khatib, J. J. Craig, and T. Lozano-Perez, eds.), MIT Press, Cambridge, MA, 1989.

[37] B. Kuipers and T. Levitt, *Navigation and mapping in large-scale space*, AI Magazine (1988), 25–43.

[38] Benjamin J. Kuipers and Y. T. Byun, *A qualitative approach to robot exploration and map-learning*, Proceedings of the IEEE workshop on spatial reasoning and multi-sensor fusion (Los Altos, CA), IEEE, 1987, pp. 390–404.

[39] Michael S. Landy and James R. Bergen, *Texture segregation and orientation gradient*, Vision Research **31** (1991), 679–691.

[40] Michael Langer, *Diffuse shading, visibility fields, and the geometry of ambient light*, Proceedings of the Fourth International Conference on Computer Vision, May 1993, pp. 138–147.

[41] Michael Langer, Gregory Dudek, and Steven W. Zucker, *Space occupancy using multiple shadowimages*, Proceedings of the IEEE Conference on Intelligent Robotic Systems (Pittsburgh, PA), IEEE Press, August 1995, pp. 390–396.

[42] Jean-Claude Latombe, *Robot motion planning*, Kluwer Academic Publishers, Norwell, MA, 1991.

[43] Stéphane Laveau and Olivier Faugeras, *3-d scene representation as a collection of images and fundamental matrices*, Tech. Report 2205, INRIA, Sophia-Antipolis, France, February 1994.

[44] A. Lippman, *Movie maps: An application of the optical videodisc to computer graphics*, Proceedings of the ACM SIGGRAPH, 1980, pp. 32–43.

[45] David G. Lowe, *Perceptual organization and visual recognition*, Kluwer Academic Publishers, Boston, Mass., 1985.

[46] Vladimir J. Lumelsky, Snehasis Mukhopadhyay, and Kang Sun, *Dynamic path planning in sensor-based terrain acquisition*, IEEE Transactions on Robotics and Automation **6** (1990), no. 4, 462–472.

[47] Jitendra Malik and Pietro Perona, *Preattentive texture discrimination with early vision mechanisms*, Journal of the Optical Society of America **7** (1990), no. 5, 923–932.

[48] David Marr, *Vision*, W.H. Freeman, San Francisco, 1981.

[49] Leonard McMillan and Gary Bishop, *Plenoptic modeling: An image-based rendering system*, Proceedings of the ACM SIGGRAPH (Los Angeles, CA), ACM, August 1995, pp. 39–46.

[50] Hans P. Moravec and Alberto Elfes, *High resolution maps from wide angle sonar*, ICRA, 1985, pp. 116–121.

[51] Ulric Neisser, *Visual search*, Scientific American **210** (1964), no. 4, 94–102.

[52] H. C. Northdurft, *Orientation sensitivity and texture segmentation in patterns with different line orientation*, Vision Research **25** (1985), 551–560.

[53] David Noton and Lawrence Stark, *Eye movements and visual perception*, Scientific American **224** (1971), no. 6, 35–43.

[54] R. Remington and L. Pierce, *Moving attention: Evidence for time-invariant shifts of visual selective attention*, Perception and Psychophysics **35** (1984), no. 4, 393–399.

[55] Ronald A. Rensink, J. Kevin O'Regan, and James J. Clark, *To see or not to see: The need for attention to perceive changes in a scene*, Psychological Science **8** (1997), 368–373.

[56] D. G. Ripley, *Dvi - a digital multimedia technology*, Communications of the ACM **32** (1989), no. 7, 811–822.

[57] Jim Rygiel, *Digital effects in motion pictures*, Invited talk, Computer Vision and Pattern Recognition (1997).

[58] Dov Sagi, *The psychophysics of texture segmentation*, Early Vision and Beyond (T. V. Papathomas, ed.), MIT Press, Cambridge, MA, 1995.

[59] Peter A. Sandon, *Simulating visual attention*, Journal of Cognitive Neuroscience **2** (1989), no. 3, 213–231.

[60] Walter Schneider and Richard M. Shiffrin, *Controlled and automatic human information processing: I. Detection, Search, and Attention*, Psychological Review **84** (1977), no. 1, 1–66.

[61] Robert Sim, *Navigating by the stars: Robot positioning using attention*, Tech. Report CIM-98-1170, Centre for Intelligent Machines, McGill University, 1998.

[62] T. Skewis and V. Lumelsky, *Experiments with a mobile robot operating in a cluttered unknown environment*, Journal of Robotic Systems **11** (1994), no. 4, 281–300.

[63] Richard Szeliski, *Video mosaics for virtual environments*, IEEE Computer Graphics and Applications **13** (1996), no. 2, 22–30.

[64] Camillo J. Taylor and David J. Kriegman, *Vision-based motion planning and exploration algorithms for mobile robots*, IEEE Transactions on Robotics and Automation **14** (1998), no. 3, 417–426.

[65] W. C. Thibault and B. F. Naylor, *Set operations on polyhedra using binary space partitioning trees*, Proceedings of the ACM SIGGRAPH, 1987, pp. 153–162.

[66] Anne Thompson, *Toy wonder*, Entertainment Weekly **304** (1995).

[67] Anne M. Triesman, *Perceptual grouping and attention in visual search for features and objects*, Journal of Experimental Psychology: Human Perception and Performance **8** (1982), no. 2, 194–214.

[68] Anne M. Triesman and Garry Gelade, *A feature integration theory of attention*, Cognitive Psychology **12** (1980), 97–136.

[69] John K. Tsotsos, *Toward a computational model of visual attention*, Early Vision and Beyond (T. V. Papathomas, ed.), MIT Press, Cambridge, MA, 1995.

[70] John K. Tsotsos, Sean M. Culhane, Winky Yan Kei Wai, Yuzhong Lai, Neal Davis, and Fernando Nuflo, *Modelling visual attention via selective tuning*, Artificial Intelligence **78** (1995), no. 1-2, 507–547.

[71] J. Warnock, *A hidden-surface algorithm for computer generated half-tone pictures*, Tech. Report TR 4-15, Computer Science Department, University of Utah, Salt Lake City, UT, June 1969.

[72] K. Weiler and P. Atherton, *Hidden surface removal using polygon area sorting*, Proceedings of the ACM SIGGRAPH, 1977, pp. 214–222.

[73] Carl-Johan Westelius, *Focus of attention and gaze control for robot vision*, Tech. Report Dissertation/TR 379, Dept. of Electrical Engineering, Linkoping University, 1995.

[74] Lance Williams and David Jacobs, *Stochastic completion fields: A neural model of illusory contour shape and salience*, International Conference on Computer Vision, June 1995.

Document Log:

Manuscript Version 2 —
Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX — 8 January 1999

Eric Bourque

Mobile Robotics Laboratory, Center for Intelligent Machines, McGill University, 3480 University St., Montréal, Québec H3A 2A7 Canada, *Tel.* : (514) 398-2186
E-mail address: ericb@cim.mcgill.ca

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX