

**Merging Static and Dynamic Visual Media
along an Event Timeline**

Kyratso Karahalios


S.B., Electrical Engineering
Massachusetts Institute of Technology
1995


M.Eng., Electrical Engineering and Computer Science
Massachusetts Institute of Technology
1995

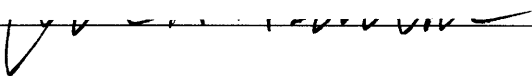
Submitted to the Program in Media Arts and Sciences
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences
at the
Massachusetts Institute of Technology

February 1998

© Massachusetts Institute of Technology 1998. All rights reserved.

Author 
Program in Media Arts and Sciences
November 12, 1997

Certified by 
Andrew B. Lippman
Associate Director, MIT Media Laboratory
Thesis Supervisor

Accepted by 
Stephen A. Benton
Chairman, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

FEB 11 1998

LIBRARIES

Merging Static and Dynamic Visual Media along an Event Timeline

Kyratso Karahalios

Submitted to the Program in Media Arts and Sciences
School of Architecture and Planning
on November 12, 1997
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences
at the
Massachusetts Institute of Technology

Abstract

The objective of this project is to develop tools that assist a user in the construction of three-dimensional spaces for interesting events. Both spatial and temporal information are used by incorporating different forms of media such as photographs, film, and video. The system enables the user to use different algorithms in different parts of the space in an attempt to use the most lucrative method for that specific portion. The result is an integrally formulated model. Acquisition, layout, organization, and navigation of the space will be performed with a graphical user interface.

Thesis Supervisor: Andrew B. Lippman
Title: Associate Director, MIT Media Laboratory

This research was supported by the Digital Life Consortium.

Merging Static and Dynamic Visual Media along an Event Timeline

Kyratso Karahalios

The following people served as readers for this thesis:

Reader _____ ✓ - - -

Walter Bender
Associate Director for Information Technology
MIT Media Laboratory

Reader _____ / - /

Michael Naimark
Research Scientist
Interval Research Corporation

Acknowledgments

As I move on to continue my studies, I would like to take the time to thank some people who have inspired and guided me along the way.

I want to thank Andy Lippman for again giving me the opportunity and endless resources to work on this project.

Walter Bender for his insight, his comments, and his enthusiasm.

Michael Naimark for sharing his knowledge and pointing me in directions I had never thought of.

Judith Donath for taking the time to help me focus on what I was doing.

Steve Waldman for just always being there and smiling, even while in Baltimore.

Ali Azarbayejani and Matt Antone for answering all my questions.

Chris Dodge for providing late night laughter and food.

Linda Peterson and Laurie Ward for their patience and support.

And most importantly, my mother, father, and brother, for their love, their encouragement, and their belief in me.

Contents

1	Introduction	15
1.1	What is an Event?	15
1.2	Exploring a Multi-Dimensional World	17
1.3	Thesis Overview	18
2	Background: 3-D Graphics and Modeling	21
2.1	Foundations of 3-D Graphics	22
2.1.1	Affine Transformations	23
2.1.2	Recording in 3D	24
2.1.3	Viewing 3D in 2D	25
2.2	Estimation and Modeling of 3-D Geometry	26
2.2.1	Stereo Vision	26
2.2.2	Using Vanishing Points to Recover Camera Position	27
2.2.3	Structure from Motion	27
2.2.4	CAD Based Modeling	28
2.2.5	Predicting Views from Existing Views	28
2.2.6	Single or Multiple Images?	28
3	Background: Navigating and Manipulating a Visual Database	31
3.1	<i>Moviemaps</i>	31
3.1.1	Video-disc Based <i>Moviemaps</i>	31
3.1.2	World Wide Movie Map	33
3.2	Synthetic Movies Derived from Multi-Dimensional Image Sensors	34
3.3	Virtual Reality Modeling Language, <i>VRML</i>	34
3.4	Virtual Worlds Project	35

3.5	Path Planning	35
3.6	Panoramas	36
3.6.1	QuickTime VR	36
3.6.2	Salient Stills	37
4	The Model Synthesis	39
4.1	Division of Modeling Techniques in SpaceMap	39
4.1.1	Three-Point Perspective Calibration with Triangulation	39
4.1.2	Shape from Motion	41
4.1.3	Modeling Considerations	42
4.1.4	Labeling Variations	42
4.2	Object Hierarchy	44
4.2.1	System Organization	44
5	The Function of the Interface	45
5.1	Acquisition and Grouping of Images	45
5.2	Labeling Features	45
5.2.1	Labeling Stills	45
5.2.2	Labeling Movie Frames	47
5.2.3	Making Movies	48
5.2.4	Storing Spaces	48
5.3	Model Constitution	48
5.4	Layout of Event Space	48
5.5	Navigation	49
5.5.1	Illusory Nature of Space and Time	49
5.5.2	Mapping	50
5.5.3	Paths of Navigation	50
6	Examples of Events	53
6.1	Model Conglomerate	53
6.2	Example of Event	53
6.3	A Changing Event	54

7 Summary and Future Work **59**

7.1 Summary 59

7.2 Future Work 59

 7.2.1 Implementation 59

 7.2.2 Navigation 60

A Projecting onto View Plane with Perspective Projection **61**

List of Figures

1-1	Map of Dealey Plaza	16
1-2	Process for creating an event	18
2-1	Geometric model of pinhole camera	24
2-2	The viewing volume	26
2-3	The general stereo problem	26
3-1	Aspen interface	32
3-2	An example of a Salient Still	38
4-1	Simple block diagram of the system	40
4-2	Ideal rotation domain for structure from motion	42
4-3	Comparison of models from two different labelings	43
4-4	The object hierarchy	44
5-1	The loading browser	46
5-2	Points and lines on a <i>Still</i>	47
5-3	The layout canvas	49
6-1	An example of parts vs. whole in creating the structure	54
6-2	A partial reconstruction made with two parts	54
6-3	Four views from one event	55
6-4	Picture of the medical center	56
6-5	Key legend	56
6-6	Two different views from a scene in motion	57

Chapter 1

Introduction

Noteworthy events attract a lot of attention and a lot of people. Whether it be for future reference or for personal reasons, these events tend to be captured using diverse forms of media. Newspaper photographers, television newscasters, and spectators with video cameras or still cameras may be present. Each of these forms of visual media record the event from a different perspective, and most probably, at a different instant in time. If there are many such images, they probably have some overlap.

It has also become increasingly easier to convert these visual images into digital form and to find images through search engines. Given this assortment of images, I created a system - *SpaceMap*, to facilitate the merging of the data into a coherent three-dimensional model of the space.

There exist many algorithms for building synthetic three-dimensional spaces from real images. However, most of these are tailored to a very specific situation, such as known camera calibration and specifically shaped objects. This system attempts to be flexible enough to allow for the use of different modeling techniques where they would best be suited as well as the extension of these models as more data becomes available. The movement through space will be achieved by interacting with a three-dimensional visual database.

1.1 What is an Event?

An event consists of a set of physical objects that occupy space-time. The physical objects themselves lie in a three-dimensional space. Forces act on these objects within this space-time domain and modify their state.

These events can be combined to form new events. Similarly, they can be decomposed into yet other events. The events in this system are created by combining events made from different algorithms. This piecewise approach allows for some creativity as well as flexibility in defining events.

To illustrate an example of an event for this system, let us consider the site of the JFK assassination. People had gathered at Dealey Plaza to watch the procession of the motorcade. Still photographers were present. Some took 35mm film, some black and white, and some took polaroids¹. Still others had 8mm home movies. The graininess and quality of the pictures was quite varied.

What is also known, although not to a precise degree of accuracy, are the relative locations of the photographers. The map in Figure 1-1 [29] shows the relative orientation of the space.

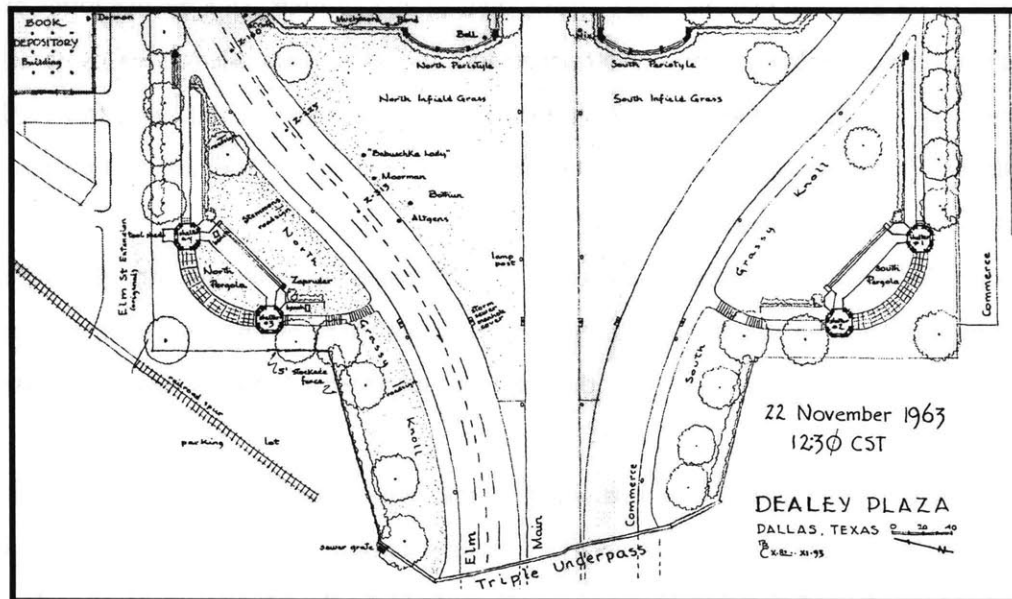


Figure 1-1: Map of Dealey Plaza

Using the map as a rough estimate we can try to build parts of the model with stereo techniques. Since the car is moving, structure from motion can be used to model parts of the car. Buildings and sidewalks can be reconstructed using vanishing points. These techniques alone may not fill the entire event space, but as new photos are added, the model may evolve incrementally.

¹The quality of most of these has unfortunately deteriorated by now.

1.2 Exploring a Multi-Dimensional World

The goal of this work is to create and explore a three-dimensional space. This space is a representation created from still images and sequences of moving images. The building of this space relies heavily on machine vision, or the ability of a machine to “see” and comprehend spatial relationships.

Due to the complexity involved in understanding vision, most vision systems impose a set of constraints to simplify the problem. Many of these constraints require knowing information about the space and the recording devices beforehand. This includes a reasonably accurate camera model, camera parameters such as focal length and location with respect to an object, the general orientation of the recorded objects, and some user feedback along the way.

In creating SpaceMap, many assumptions were also made. It assumes a pinhole model for recording, the space is made up of polygons, as well as some prior knowledge of the space. The user is required to label directions, as well as edges, surfaces, and certain key features common to the images.

Because of the limitations of this approach, the models have significant errors and gaps in regions where there is not enough information to reconstruct the scene. This thesis does not attempt to *solve* the vision problem², but to explore object manipulability, visibility, and synthesis. It seeks a simple, yet robust and meaningful way to represent a multi-dimensional space without confusing and disorienting the user.

The framework of the system is very modular. It allows for the inclusion of new algorithms without altering the existing framework of the model and without causing compatibility conflicts with already existing spaces that are located on the network. It also allows for the user to selectively include or exclude certain objects from a space and see how it affects the environment as well as compare the similarities and weaknesses of the different algorithms. Several alternatives are explored for simple and efficient exploration the space, from pre-selecting a viewpoint to using artifacts to aid in the navigation.

²By this I mean understanding the subject of human visual perception, and how to mimic this using a machine.

1.3 Thesis Overview

SpaceMap consists of four separate types of tools: tools for labeling images, estimating models, communicating between them, and navigating. By using these tools, one can input two-dimensional images and create a visual database that is navigable in a three-dimensional space. A simple block diagram of the process for creating events can be seen in Figure 1-2.

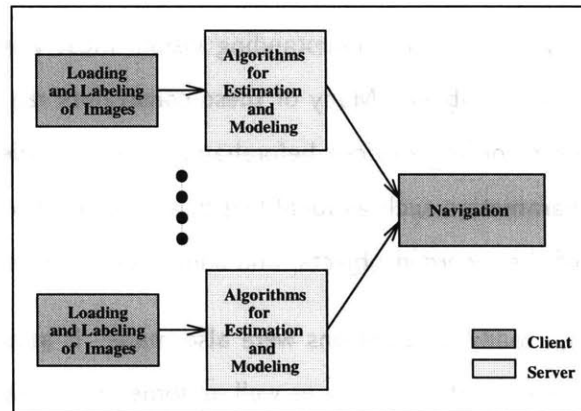


Figure 1-2: Process for creating an event

The knowledge that this thesis presents spans a wide spectrum.

- ◇ First, it provides a medium to experiment with and understand which algorithms are efficient in modeling different events. When creating an event, users interact with the system using a stand-alone java applet. Images are loaded from a local disk or URL into the system. The system is user-assisted. That is, the user manually labels feature points, dimensions, and surfaces in the respective images. Most of the computation exists on the server side in the form of c and c++ libraries. The user chooses which methods to use and where to use them. This is because different portions of the event are treated as separate objects. In cases where there is not enough information to place one object with respect to another, the user can specify distances between specified points.
- ◇ Secondly, it provides the tools to research how to meaningfully view an event. Once a model is created, it can be navigated using the mouse and keyboard. It also addresses the question of how much control the user should be given in the navigation as well as how to store and access the vast amount of data involved.

◇ Lastly, it is equipped with a framework to allow for a varied audience for these events. The finished models are then viewed within a web browser.

There are several potential problems to consider when implementing this type of system. One of them is how to view regions with occlusions. Another difficulty arises when trying to portray moving objects in a system which essentially depicts space as slices of time. Also, there are some objects which are difficult to model such as trees and people.

Chapter 2 presents a brief overview of 3-D graphics and the modeling terminology used throughout this work.

Chapter 3 discusses related work in graphic environments, navigation, and interface design.

Chapter 4 introduces the system, SpaceMap. It addresses how the system is organized and how the different pieces work together.

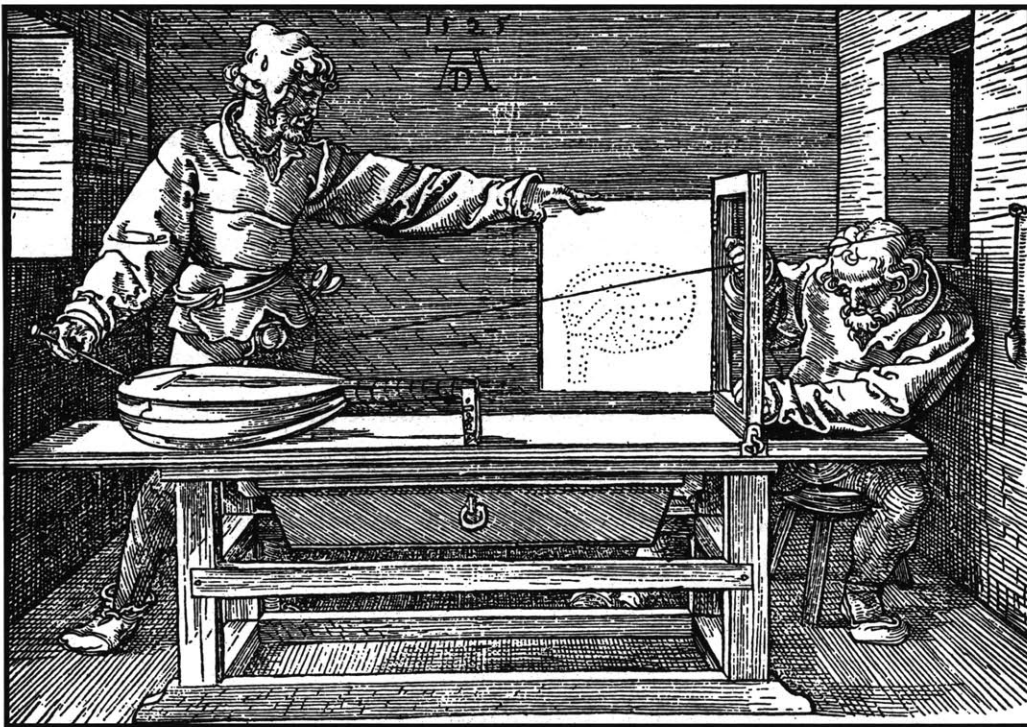
Chapter 5 describes the operation of the system and how the interface was structured around the framework.

Chapter 6 provides some examples of models created using the system.

Chapter 7 evaluates the system and discusses future work.

Chapter 2

Background: 3-D Graphics and Modeling



The computational portion of this work attempts to map a three-dimensional volume onto a two-dimensional surface. As early as 300 B.C., Euclid studied the geometry of three-dimensional objects and their two-dimensional projections. The Renaissance brought forth the first paintings with a “true” sense of perspective. Artists such as Dürer, Piero della Francesca, Viatore, Alberti, and Brunelleschi established a theoretical geometrical grounding for their work. Dürer published

some of this knowledge in his, “The Painter’s Manual 1525” [12]¹. In a draft of one of his books on painting, he writes:

But such books [by the ancients] we no longer have, and because they are irretrievable one must try to write a new one. And this has induced me to put down in the following pages that which I have learned, so that those who read it will be led to think more about such things and perhaps find a better and shorter route. And I will start this endeavor with measure, number, and weight. [27]

Leonardo da Vinci then paved the way for shading and depth perception. To date, there is no solution to the vision problem. The use of computers has provided a medium with which to experiment with *seeing* machines.

Computer vision has the potential to record information of the surrounding environment. Living systems use vision to navigate about their surroundings. Because it is so easy to input visual information into a computer, it is an obvious extension to attempt to navigate using the vision capabilities of a computer. Due to noise in measuring instruments, equations without closed-form solutions, lack of understanding of biological vision systems, we can see that this is a complex problem. Several assumptions and constraints are made among the various techniques to simplify the problem space.

This work builds upon the accumulation of data from various modeling techniques. In order to better understand how the three-dimensional model structure is created, this chapter begins with a brief introduction to 3-D graphics. It also introduces the terminology used throughout this chapter. The chapter then proceeds to describe several different models for estimating the three-dimensional structure of scenes.

2.1 Foundations of 3-D Graphics

A point \mathbf{p} in the projective space, P^3 , is described as the coordinate vector

$$\begin{bmatrix} x & y & z & w \end{bmatrix}^T \quad (2.1)$$

¹The drawing above is from Dürer’s, *Der Zeichner der Laute*. It is a perspective apparatus depicting a man drawing a lute.

including a scale factor ². It is assumed that the coordinate space is right-handed.

The following sections describe a brief overview of the transformations involved in 3-D graphics. The intent is to view mappings from 3-D to 2-D coordinates, and vice versa, simply as a composite of transforms.

2.1.1 Affine Transformations

An important property of an affine transform is that it preserves parallel lines. Examples of such transforms are translation, rotation, scale, and shear. In the defined space, such a transform is represented by a 4-by-4 matrix that maps the original point to a new point.

The 3D translation matrix,

$$T(d_x, d_y, d_z) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

transforms the point

$$p = \begin{bmatrix} x & y & z & 1 \end{bmatrix}$$

defined above into:

$$T(d_x, d_y, d_z) \cdot \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T = \begin{bmatrix} x + d_x & y + d_y & z + d_z & 1 \end{bmatrix}^T. \quad (2.3)$$

Analogously, a scaling matrix is represented as

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

²

$$\begin{bmatrix} x & y & z \end{bmatrix}^T \rightarrow \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$$

is a one-to-one mapping of affine space to projective space. Throughout the context of this work, will will assume we are in the affine space.

and a rotation matrix is represented as

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.5)$$

Three-dimensional shearing matrices allow for shear along the x, y, and z axis. The matrix,

$$SH_{xy}(sh_x, sh_y) = \begin{bmatrix} 1 & 0 & sh_x & 0 \\ 0 & 1 & sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

shears points along the z axis. Matrices that shear along the other axes can be created in a similar fashion.

These matrices have been presented as individual point transformers. Transforming lines, planes, and more complex surfaces is an obvious extension.

2.1.2 Recording in 3D

This section establishes the geometric model of our recording device. Figure 2-1 illustrates a simple pinhole camera model.

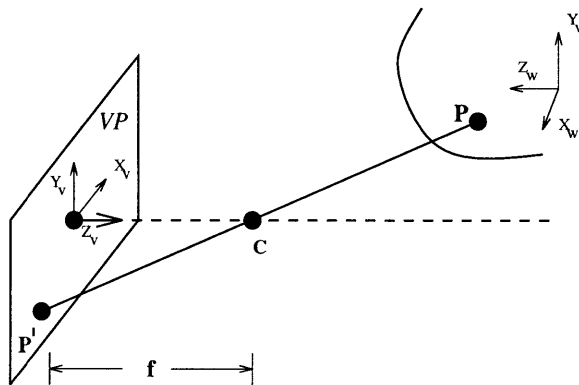


Figure 2-1: Geometric model of pinhole camera

In this model, all the points are projected through the optical center³, C . VP is the viewing plane or retinal plane where the image is formed through perspective projection. P is the three-dimensional point, and P' is the image in the viewing plane. f is the focal length.

Using this simplified model, mapping the world coordinates to the view plane coordinates is simply a series of several transformations. A more detailed derivation of the transformations may be found in Appendix A.

Camera Calibration

The projection from the 3-D world to the virtual plane can be described with a set of parameters: the intrinsic and extrinsic parameters. Intrinsic parameters include pixel shear, optical center, effective focal length, and radial and decentering lens distortion terms [8]. Extrinsic parameters are used to represent the camera's position from a fixed position in the world. Affine transformations and three-point perspective are just two of the methods that can be used for camera calibration.

2.1.3 Viewing 3D in 2D

Mapping onto a Computer Screen

The porthole into the event will ultimately be a rectangular portion of the computer screen. Therefore, the points in the viewing plane image that do not fit in the screen porthole are essentially discarded.

Clipping Planes

When using perspective projection, it is often good practice to bound the viewing volume. An object very far from the center of projection will appear indistinguishable. Similarly, an object very close to the center of projection will be scattered across much of the viewing window. This viewing volume is bounded by a *front clipping plane* and *back clipping plane*. These planes are parallel to the viewing plane and are located a distance, F and B , in front of and in back of the viewing plane, respectively (see Figure 2-2).

³Sometimes referred to as the focal point.

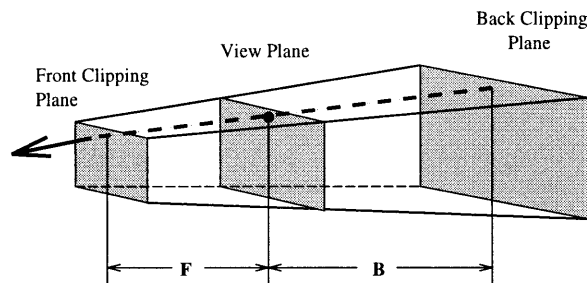


Figure 2-2: The viewing volume

2.2 Estimation and Modeling of 3-D Geometry

Many methods exist for creating photo-realistic and artificial virtual environments. This section briefly describes some of them. It begins with a method that requires previous knowledge of the separation distance of a stereoscopic camera. Other methods that require a priori knowledge might depend on knowing the focal length of the camera or the location of a specific calibration object in the image. Some of the other methods mentioned attempt to recover the camera parameters and then proceed to recover the points. These methods can be supplemented with known parameter information if it is available.

2.2.1 Stereo Vision

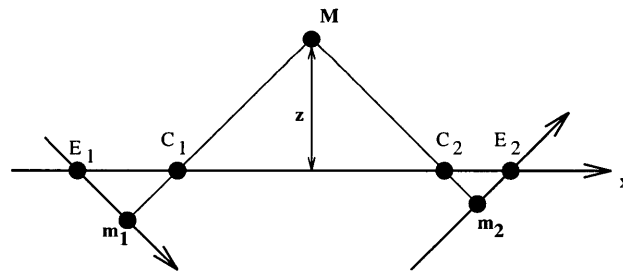


Figure 2-3: This is pictured in the $C_1C_2m_1m_2$ plane. C_1 and C_2 are the optical centers. $\langle C_1, C_2 \rangle$ is the optical baseline. m_1 and m_2 are the respective correspondences. E_1 and E_2 are the respective intersections of the optical baseline with the retinal planes. The disparity is defined as $d = 1/z$.

Stereo vision can be used for the reconstruction of three-dimensional coordinates. The approach can be used with two or more cameras. The more cameras, the better the reconstruction tends to be, for there will be more detailed geometric, epipolar, and physical constraints and a

larger viewing area⁴.

The first problem encountered in this approach is that of correspondence. The individual pixels, edges, curves, or other distinct features in each of the images must be matched. This task is inherently ambiguous.

Once a set of corresponding points meet the correspondence criteria, the three-dimensional coordinates of this match in the universal coordinate frame can be calculated. This is where the reconstruction occurs.

This method assumes the distance between the respective cameras is known or that it can be derived from an object of known size in the image. Figure 2-3 is a diagram of the general case for determining the universal point, M [13].

2.2.2 Using Vanishing Points to Recover Camera Position

One such method for determining the intrinsic parameters of each camera takes advantage of the invariance of some geometric properties to perspective projection. It relies on the detection of the vanishing points in images with three mutually orthogonal sets of parallel lines [8, 2]⁵. The directions and coplanar regions are often labeled by the user⁶.

The intrinsic camera parameters are derived from the three vanishing points in the image. Once the camera parameters are known, the three-dimensional structure of the space is recovered using point correspondences and triangulation.

An interesting benefit of such a method is that a model can be created from just one image. This technique is very reliable given that the scene contains essentially straight lines and planar surfaces which are known a priori by the user.

2.2.3 Structure from Motion

In this case, we first try to recover the camera motion using optical flow information of an image sequence. This method requires that features in the sequence be tracked or somehow labeled. Horn presents a least squares algorithm to determine structure from general motion [15].

⁴This depends on the placement of the cameras. There is a limit to which adding cameras adds needed information.

⁵[4] includes the elimination of radial and decentering lens distortion.

⁶Line detectors can be used to find straight edges, but they tend to not be very reliable; user-labeled lines give better results.

Another such method creates models from uncalibrated cameras. It also recovers camera motion, pointwise structure, and camera focal length. The user must provide some sort of feature tracking. By tracking these features, the user can recover the motion, focal length, and the three-dimensional coordinates of the feature points using the extended Kalman filter (EKF)⁷. The estimation handles translation as well as rotation. The user then segments the object into piecewise smooth pieces. The surface of the object is then texture mapped with the corresponding real images [5].

This technique becomes increasingly harder to use when there is more than one moving object in the sequence⁸.

2.2.4 CAD Based Modeling

Very reliable models can be created through the use of CAD systems. These require measurements of the scene's physical space or corresponding maps of the space. The corresponding portions of the image are then mapped onto the defined CAD specifications.

2.2.5 Predicting Views from Existing Views

This method attempts to avoid calibrating the camera altogether. It is similar to methods used in trinocular stereo and has proved to be quite effective. Two views should at least be weakly calibrated⁹. Given the fundamental matrices between any two views in a set of N views, the epipolar geometry of all N views can be derived. The viewpoint and retinal planes are then propagated through the reference views to predict a new viewpoint [20]. This method essentially represents the new viewpoint through algebraic relations to the existing viewpoints.

2.2.6 Single or Multiple Images?

Given a single two-dimensional image, instructing a computer to find the third dimension is an ambiguous task. That is why dimensions are labeled when calibrating a camera using vanishing points. When using stereo to recover structure, one must have at least two overlapping views from different viewpoints. Similarly when using motion to recover structure, one must have at least two overlapping views taken at a different time. When using stereo, a wide baseline tends

⁷Other methods may be employed as well.

⁸[11] proposes a solution to this case.

⁹Weakly calibrated implies that the epipolar geometry is known.

to give higher precision, whereas in motion, time intervals should not be too large in order to yield an accurate correspondence^{10 11}. Predicting views from existing ones also requires at least two views.

¹⁰They should not be too small either, for then there is essentially no motion.

¹¹It is also possible to manually label correspondences in both stereo and motion methods.

Chapter 3

Background: Navigating and Manipulating a Visual Database

The previous section dealt with the creation of models for the event space. This section examines methods for traversing this augmented space. That is, what is required from the user to get from one place to another, and how much control and interactivity is involved. SpaceMap builds on styles used in the interactive systems below.

3.1 Moviemaps

A moviemap provides a variety of routes for travelling in a virtual environment along a pre-planned tour [21, 24, 25]. The factors that can be controlled such as frame rate and direction may vary. These issues of control as well as traversal and data lookup are topics influencing SpaceMap. The following examples describe different movie maps, their interfaces, as well as different means used for navigating movie maps.

3.1.1 Video-disc Based Moviemaps

Aspen Moviemap

The first interactive movie map was made in Aspen. The Aspen moviemap provided more structured directions for navigation than does SpaceMap. It also encompassed a variety of input devices and was photorealistic.

The Aspen moviemap provided a means for acquiring and displaying spatial knowledge.

It allowed the user to pre-experience a “place” by exploring it interactively. It is an optical video-disc look-up-based system. Photographs were taken by placing four cameras with 90° horizontal field of view lenses, pointing in four orthogonal directions, on the roof of a van and taking pictures every ten feet while driving through the streets of Aspen. Straight footage was taken, by filming in both directions on every street. Turn footage was also taken by filming turns from every street onto every intersecting street. A technique for synthesized turns was also created using two perpendicular views from cameras and interpolating between them and also from 360° anamorphic panorama views [32]. This spatial continuity created by match-cutting resulted in a visually *seamless* environment.

One interacted with the moviemap via a touch screen monitor, joystick, and tablet. There was a vertical and a horizontal screen. The vertical screen displayed photographs that simulated navigation as if driving through the streets of Aspen. One could move left, right, forwards, or backwards. There was also the option to view an abstracted representation of Aspen from different viewpoints using QADAS, or Quick and Dirty Animation System. The horizontal screen displayed a map of Aspen. One could also reach a destination by using the map. Other data in the form of slide shows, photos of facades of buildings, and text were available to the user when they chose a point of interest.



Figure 3-1: Aspen interface

Navigating Aspen consisted of the user choosing one of several discrete directions and moving towards them at their own pace. It also allowed for navigation beyond just the space and into the buildings in the form of pre-annotated slide-shows and documents. This organization was ideal for the city of Aspen. The city had well defined boundaries and primarily orthogonal streets. The association between the landscape and the viewpoint direction enabled the majority of the users to become acquainted with the environment without losing a sense of orientation.

Other Moviemaps

One could choose a turning direction in the Aspen movie map by touching an arrow on a touch screen. They would then proceed to view the actual process of turning. Another technique used

in the movie map, the “Paris VideoPlan”, incorporated the metaphor of a *guide* to assist the user. It was a sidewalk view of the Madeleine district in Paris. A 35mm camera was used to take one frame every 2 meters along the sidewalk. Instead of filming and viewing every possible turn, a mime directed the user and represented the link from one locality to the next. The mime stood at the intersections and gestured at possible directions for turning. This introduced a cinematic continuity different from the spatial continuity represented in Aspen.

The concept of speed in movie maps is an interesting issue. Movie maps are made by shooting frames at known distance intervals. By increasing the frame rate, one appears to be moving faster. The “Golden Gate Videodisc Exhibit” was an aerial moviemap and used a trackball to control both speed and direction. A 35mm gyro-stabilized motion picture camera on a helicopter was used for filming a 10 by 10 mile span of the Bay Area always directed towards the center of the Golden Gate Bridge. Because the viewer always faced the bridge, there was no turning left or right. This interface created a very responsive feel. It provided more control, coherence, and allowed the user to traverse this locale at very high speeds.

The moviemap in Karlsruhe, Germany, “VBK: A Moviemap of Karlsruhe” took advantage of the 100 miles of track from the downtown area and into the Black Forest. The filming was done using a camera, with an 85° horizontal FOV, mounted in the front of a tram car. The rail ensured registration of the images in the Aspen style. The navigation was accomplished by controlling a throttle for speed and three buttons for direction while standing in front of a 16 foot wide screen. The elegant relationship between the making and the viewing of this moviemap not only provided a spatial coherence, but also the element of immersion into the scene.

“See Banff!” was the first stereoscopic moviemap. A set of unconnected routes were filmed in the Canadian Rocky Mountain Banff region. The hardware was later housed in a kinoscope design podium with an eyehood for viewing, a crank with force-feedback for moving, and a lever for choosing different scenes.

3.1.2 World Wide Movie Map

It is interesting to observe how a site is created and what each user feels are important characteristics of that particular site. In creating the event spaces with SpaceMap, different characteristics are emphasized by different users based on how they perceive that space. Similarly, in the World-Wide-Movie-Map, *WWMM*, “spaces” consisting of still images and movies are patched together.

The movies and links guide the user as they tour that space.

WWMM is a composite of the World Wide Web and the Aspen Movie Map. It is a “walk-through map” of sites around the world.

These sites can be created by users using a java interface and are supported by a distributed server architecture. The beauty of this is that different sites get “stitched” together by different people based on their own experiences. Unlike movie maps that exhibit spatial continuity throughout the space, WWMM exhibits conceptual continuity. The sites are composed of still images and movies that traverse portions of the site. A site may have many links and can also be connected to other sites [31].

3.2 Synthetic Movies Derived from Multi-Dimensional Image Sensors

SpaceMap uses sets of two-dimensional images to create models. To create models from these images, depth information must be retrieved as well as color information. In his PhD thesis, Bove [9] describes a range-sensing camera whose output is not two-dimensional frames, but rather a three-dimensional database of objects and their movement in the scene.

The camera used in the Aspen moviemap (see Section 3.1.1), incorporated the fact that frames were taken at known locations. This was crucial for recreating a coherent space. The range-sensing camera makes assumptions about optics, dynamics, lighting, reflectivity, and smoothness of objects. In this manner, a moving scene can be rendered from viewpoints other than that of the original camera.

3.3 Virtual Reality Modeling Language, VRML

The objects in SpaceMap are represented as capable of performing certain *acts* based on whether they are stills, movies, models, or discrete models within models. They are also meant to be viewed in a browser. A similar object-oriented approach appears in VRML. By dealing with objects in this manner, the issue of manipulating objects is addressed as well as the issue of navigating the space that contains them.

VRML differs from custom systems that describe three-dimensional environments in that it is an extension to the World Wide Web and is in theory, platform independent. The goal

is to attribute properties and actions to objects. These objects can then be manipulated over a network. VRML endeavors to create virtual environments with a level of multi-participant interactivity.

3.4 Virtual Worlds Project

An important concern when navigating into and away from spaces is maintaining a meaningful representation at all possible viewpoints. When changing the viewpoint and clipping planes at which a scene is viewed, the information in the model might not have much interesting information if the viewer is too close or too far. The following project attempts to provide relevant information about its world based on a chosen resolution and bandwidth.

The Virtual Worlds Project proposes a framework that supports multi-resolution simulation [23]. Depending on the view of the environment, different visual representations and models will be used. Instead of producing a model and then generating views from it, the representation is based on the location of the *observer*.

Objects are composed of views and sub-objects. Each object contains a formula describing what to show as a function of the observer. This method also attempts to optimize the information transferred on distributed simulators.

3.5 Path Planning

A common method for navigating a model, as is also done in this work, is to allow some input device to modify the camera parameters of the scene. By doing so, one can essentially move about the model by altering the viewpoint. However, allowing all possible views in a scene is not practical. Some views may lack surfaces due to insufficient information, or may be of no interest. Others may appear disorienting to the user.

Formulating an *interesting* path for the user might make a good first pass through the scene. The moviemaps addressed earlier offered an interactive pre-planned tour of a space. This planning had to be considered during the filming of the space.

In his PhD thesis, Steven Drucker devised a set of criteria to assist in the manipulation of the scene. By using a set of camera constraints, a path potential function is calculated. Using this function, the system also generates the shortest path covering designated places of interest. In this manner, different agents can be created to accommodate different users.

Another method for constructing paths of navigation is to remember common travelling patterns of many users. A match may indicate an interesting view or places of interest.

3.6 Panoramas

The Panorama¹ has evolved greatly through the years. Three or four hundred years ago in Italy, scenes of vast gardens were laid for viewing outside of windows [1]. Huge paintings of panoramas were also made on circular or cylindrical canvases and buildings for entertainment purposes. Today, there are cameras specifically made for taking panoramic photographs as well as algorithms that stitch panoramas together from several photographs. Their topology provides an interesting and efficient means for looking about a scene.

Panoramas have the property of capturing and displaying a view from a single point in space. They are two-dimensional. Panoramas can be cylindrical, spherical, or made up of multiple rectilinear images; they can be any subset of a 360° arc. A distinction is generally made between stationary and moving panoramas.

One of the topics addressed in this work is how to represent the dimension of time in this three-dimensional space. By including time, the problem is then restated as one of representing four-dimensions in a three-dimensional space. The following sections describe how motion is incorporated into the viewing of panoramas.

3.6.1 QuickTime VR

Apple Computer's Quicktime VR, *QTVR*,² is a commercial product used to navigate virtual environments. *QTVR* allows for the navigation of a scene through panning and zooming a two-dimensional panorama, as well as links to hotspots in the panoramas. The techniques used for the interactive visualization are a subset of anamorphic image processing [13][17].

In effect, the user sees a two-dimensional image from a single point of view. The result can be photo-realistic. Navigation is therefore seen as panning, zooming, and "branching" to the next location. In this approach, the user finds themselves enclosed in a bounded space with the option to look around or branch to a new space.

¹Sometimes referred to a cyclorama.

²There also exist several other similar commercial products.

A second mode of QTVR simulates viewing an object from different locations. In this approach, the user is located outside the space and is allowed to view around it.

Moving Movies

“Moving Movies” used motion picture projection to create a moving panorama. A camera on a turntable recorded a view. This view was then displayed with a projector. The camera and projector had the same FOV. Using the same turntable for both, the angular movement was also preserved³. Essentially, the projector mimicked the angular movement of the camera so that spatial coherence was maintained. In one installation, a living room was filmed on a turntable. The room and furniture were then painted white. In the playback, the projection moved along this three-dimensional canvas.

“Be Now Here (Welcome to the Neighborhood)” introduced stereoscopic panoramas. Four cities were filmed at different times of day with two cameras⁴. For playback, instead of a rotating projection, a rotating viewing platform was used. The panoramas included motion from local pedestrians as well as wildlife and automobiles [25].

3.6.2 Salient Stills

Analogous to representing a four-dimensional space in three dimensions, it is noteworthy to mention a method that is capable of recording time of a two-dimensional world in a two-dimensional image⁵.

A *Salient Still* [28] captures a two-dimensional representation of the world at an instantaneous moment in time. It does so by warping an object from one view onto the same object from another consecutive view⁶. This warping process creates a visual effect, that depending on the viewing sequence, can record the passage of time. The tiling of images is conceptually similar to panoramas made by QTVR or panoramic cameras, however, in allowing an extra degree of freedom in the time domain, the image captures a history of not only the space, but of the movement within it.

Figure 3-2 is a Salient Still. In this picture the camera “paints” the portion of the scene that

³Alternatively, the lens FOV can be scaled with respect to the rotational speed.

⁴The cities were Jerusalem, Dubrovnik, Timbuktu, and Angkor, the four cities in the UNESCO “In Danger” list.

⁵This can be restated as representing a three-dimensions in two dimensions.

⁶Salient Stills are generally created from a series of consecutive views from the same camera.

it sees. The picture is also a recording of the motion of the camera. Different results can be obtained depending on which filters one uses. Looking at the picture, one can see that objects that were in motion appear slightly blurred. Salient Stills work best when the scenes are affine.

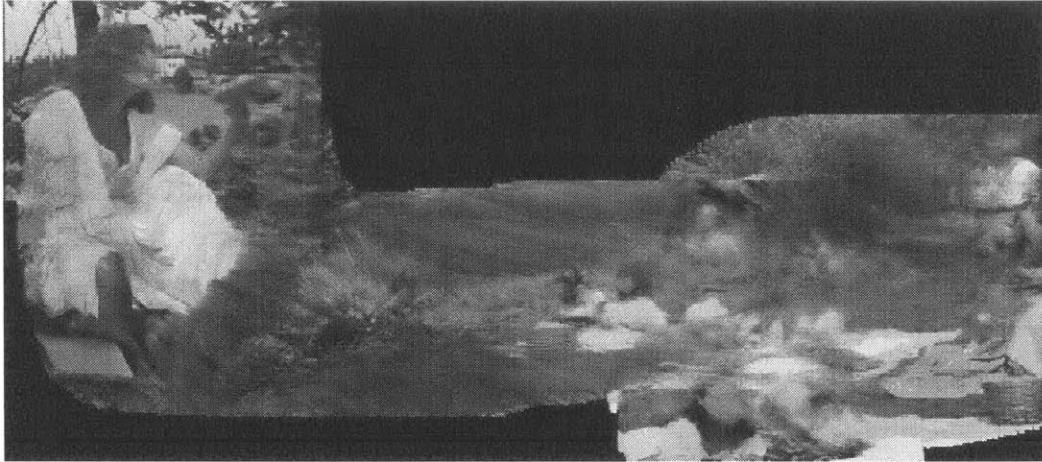


Figure 3-2: An example of a Salient Still

Chapter 4

The Model Synthesis

Several methods for modeling objects were mentioned in Chapter 2; different navigating techniques were discussed in Chapter 3. This chapter begins to describe a set of procedures that combines these two concepts into a set of tools called SpaceMap.

SpaceMap allows for the organization of two-dimensional images from different perspectives. This two-dimensional input undergoes a transformation and becomes one three-dimensional output that is traversable. It is also possible to add the dimension of time.

4.1 Division of Modeling Techniques in SpaceMap

This section describes how SpaceMap facilitates the merging of the different models. SpaceMap is split into two basic parts: composition and navigation. A model of an event may be a composite of several different models. Each distinct model, however, may be created using a different algorithm. Once a model is created, it “acts” as any other model regardless of how it was created.

This layer of abstraction divorces the composition stage from the navigation stage. The algorithms consist of c and c++ libraries that are located on a server and accessed via native calls from the client (see Figure 4-1). The two algorithms currently used in SpaceMap are briefly described below.

4.1.1 Three-Point Perspective Calibration with Triangulation

This is an extension of the vanishing point method described in Section 2.2.2. It depends heavily upon the user for the labeling of directions in order to determine the vanishing points. There

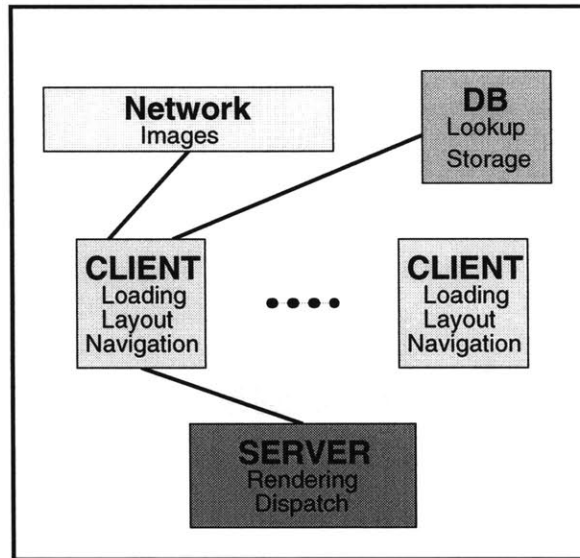


Figure 4-1: Simple block diagram of the system

are three classes of features that must be labeled: points, lines, and surfaces.

A point is defined as an x and a y coordinate in the 2-D image. A line is defined as a set of two points¹. A set of lines and points defines a surface.

Once these features are labeled, the method first estimates the camera's intrinsic and extrinsic parameters. It then proceeds to estimate the polyhedral scene structure.

Estimation

The focal point can be easily computed given that three vanishing points are found in the image. The remaining two intrinsic parameters, the x and y offsets of the center of projection, are the coordinates of the focal point with respect to the viewing plane.

Once the intrinsic parameters are found, the extrinsic parameters, three rotation and three translation components, can be estimated. A more detailed derivation of these parameters can be found in [7].

Building the Structure

Since the surfaces in the image were labeled, each of the surface normals can be estimated. There are three criteria for the recovery of the structure as described in the *origami fixing*

¹Most of the lines labeled tend to be edges of objects in the image.

algorithm [8].

- The scene itself must be an *origami*² scene.
- At least one of the surfaces must contain a known 3-D position of a feature or line.
- Two or more directed lines must be recoverable from each surface.

When these conditions hold, the surface normals can be calculated, and the scene can be reconstructed iteratively.

4.1.2 Shape from Motion

Most shape from motion algorithms require a preprocessing stage for the tracking of specific features, usually involving specialized hardware. In this implementation, the features will be tracked by the user. This is because feature tracking is a difficult problem in itself, especially with grainy images.

The primary construction then occurs by estimating the 3-D motion of these feature points. In downloading sequences from a network, the user usually has no camera information, therefore, as in the previous model, camera parameters must be determined. This adds an extra level of complexity to the problem.

The camera model used here is the pinhole camera model mentioned in Section 2.1.2, however, it is enhanced to allow for all possible focal length singularities including a focal length of infinity. Points are characterized as absolute points from a chosen reference frame and a respective depth.

For motion-based estimation, the 3-D geometry of the N original feature points must be recovered from an arbitrary video sequence. This is accomplished through the use of a *generalized extended Kalman filter* (GEKF). This is a non-linear mapping from a Euclidian space to an error space. A more detailed description of this manifold-tangent model can be found in [3].

The novelty of this approach is in its camera model, for it maintains both the relative and internal orientation of the camera states. By doing so, it can easily map the scene parameters from frame to frame even when the focal length is unknown.

²An origami scene is defined as a scene composed entirely of polyhedral surfaces.

4.1.3 Modeling Considerations

Each of the two methods above is tuned for a specific conditions. For example, the vanishing point method works well with images with straight lines. For events that depict urban settings, this method yields better models.

The structure from motion method is more selective. Recovering structure from a static scene while a camera is panning will not work. There do not exist enough distinct views of non-coplanar elements, and this would result in insufficient information for a unique interpretation. If an object is moving during a pan, a part of the moving object may be recovered. The ideal case would be the one pictured in Figure 4-2, where the camera rotates about the object within a ninety degree path. The path is limited so that the points can be tracked as the camera moves³. For these reasons, modeling objects using motion proved to be more difficult from random footage.

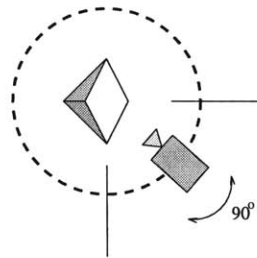


Figure 4-2: Ideal rotation domain for structure from motion

4.1.4 Labeling Variations

With both of the above methods, there is an extensive amount of user assistance, primarily in the labeling. Because of the differences in user interaction, there is an accountable variance in models created from the same image. These variances are propagated throughout each stage of the algorithm.

Figure 4-3 shows two different labelings of the same still image. The method used here to create this model relied on vanishing points. There is quite a discrepancy in the resulting models. Given enough time using tools such as these, the user develops a certain level of skill and distinct style in model creation. This becomes evident in the portrayal of events as well.

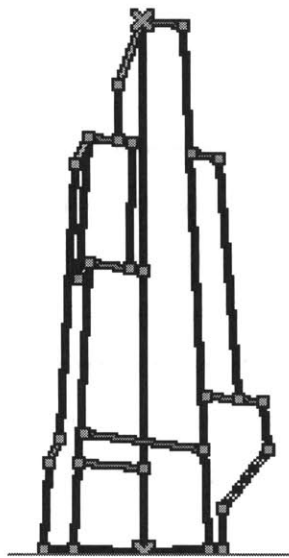
³Similarly, the object can rotate while the camera is steady.



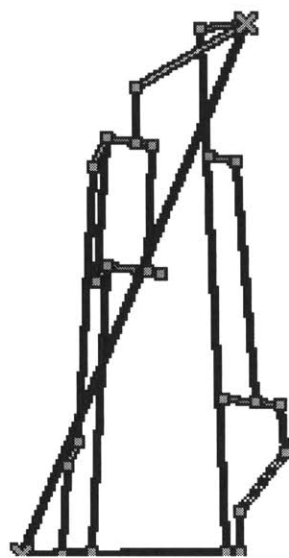
(a) Labeling A



(b) Labeling B



(c) Model from A



(d) Model from B

Figure 4-3: Comparison of models from two different labelings

4.2 Object Hierarchy

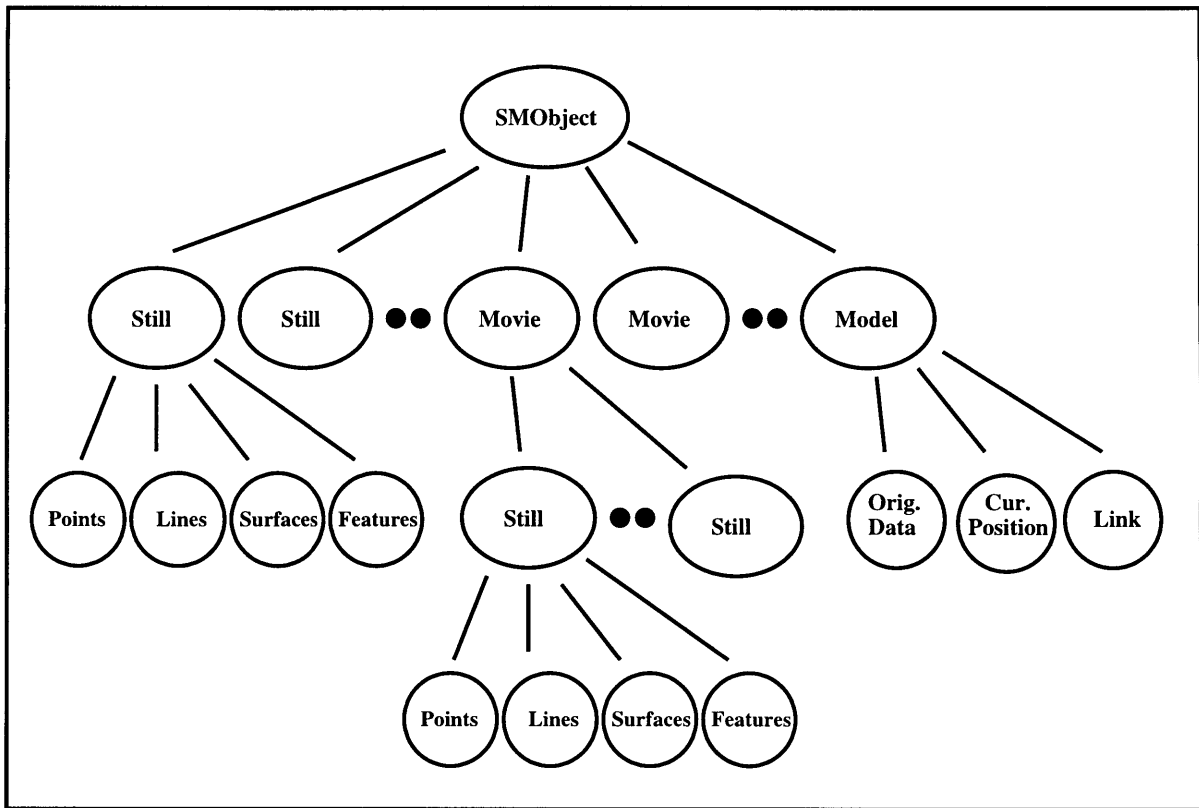


Figure 4-4: The object hierarchy

The object hierarchy of SpaceMap can be seen in Figure 4-4. From this, we can see how the different models merge into one representation. Every single entity exists as an *SMObject*. The method described in Section 4.1.1 relies on the properties of a *Still*. A *Still* encapsulates the name, location, group, and labeling of an image. The shape from motion method described in Section 4.1.2 used a movie or image sequence. A *Movie* is defined as a series of *Stills*. It, therefore contains the same labeling features. It is also possible to use both methods on any one of the images. A *Model* is the result of the 3-D estimation algorithm. It is used for navigating.

4.2.1 System Organization

The above hierarchy is hidden and maintained in the client side of the system. The client is responsible for accessing images from the network, finding images in the database, and sending data to the server. The server consists of c and c++ libraries, while the client is written in java.

Chapter 5

The Function of the Interface

This section discusses some of the functionality of the interface in SpaceMap. It begins with the creation of the events and proceeds to the interface of the layout and navigation.

5.1 Acquisition and Grouping of Images

The system's input consists of stills and movies. These images are currently loaded in terms of URL's. The images for each event are grouped together under a space name. Figure 5-1 shows the loading panel of the system. Once an image is added to the space, a thumbnail is added to that space. The panel also allows the user the option of labeling that image as a still, labeling it as a movie, or creating a movie. The canvas containing the image allows the user to pan and zoom to better label the image.

5.2 Labeling Features

There are two labeling panels: the still label panel and the movie label panel. When the image loading panel is first brought up, the still panel is the default labeling panel (see Figure 5-1)¹.

5.2.1 Labeling Stills

When labeling stills, the user must first label points, then lines, and then surfaces.

¹Specifically, the point labeling still panel.

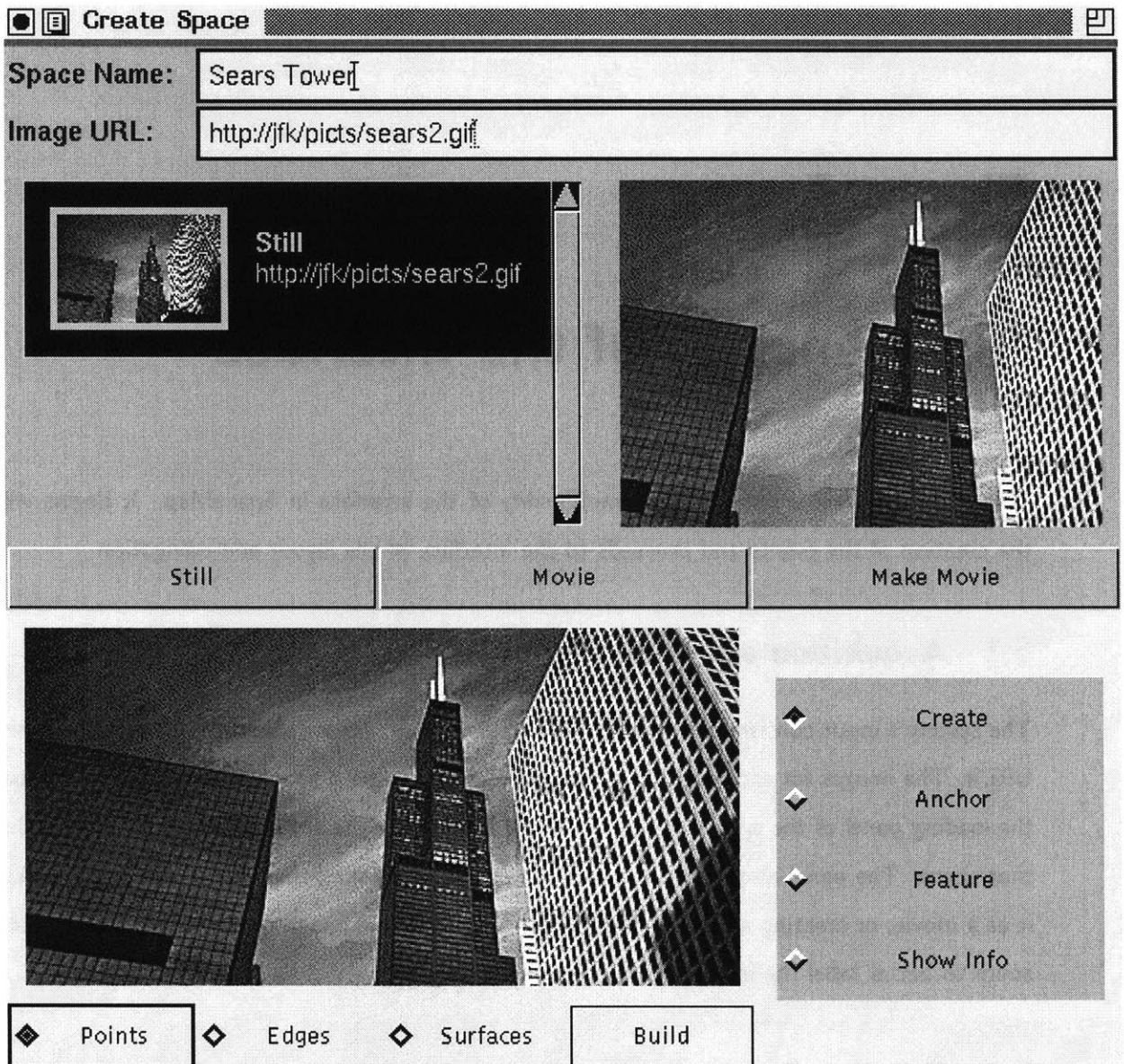


Figure 5-1: The loading browser

Points

To label points, one must choose the create option. Then, points are labeled simply by clicking. An anchor is a special point used by the system to specify a certain location such as the origin or points on the axes. If anchors are not chosen, then the system chooses defaults. A feature is there for future implementation if using multiple non-sequential images of the same scene.

Show Info mode shows the coordinates of a point.

Lines

When labeling lines, the user first clicks on the beginning point and then on the end point. It is important to label the x, y, or z directions of the lines correctly.

Surfaces

Surfaces are defined as a chosen set of lines and points. They are then given a name tag for reference².



Figure 5-2: Points and lines on a *Still*

5.2.2 Labeling Movie Frames

Labeling features in movie frames is similar to labeling points in still images. The major concern here is to label the points in the same order in each of the images. This can become quite complicated as the number of points increases.

²The labeling of points, lines, and surfaces is analogous to the labeling in [2].

5.2.3 Making Movies

A movie in SpaceMap is defined as a sequence of stills. To make a movie the user loads all the necessary images into the movie making panel. They can then be reordered by moving the thumbnails in the thumbnail panel. There is also the option of viewing the movie³.

5.2.4 Storing Spaces

The spaces are stored in a serialized file. Each space knows which stills and movies belong to it as well as their labelings. When the model is created, it too is attached to its specific space along with its position and orientation.

5.3 Model Constitution

A model is a set of three-dimensional points. Once a model exists, it can be rotated, translated, or scaled.

The models are created solely on their geometrical features such as points, lines, and surfaces. By basing the computation of the structure on these primitives, we bypass the problem of dealing with the problems posed with images of different resolution and picture quality. In a sense, these models are a representation of space that is revised and made consistent with mathematics.

5.4 Layout of Event Space

Figure 5-2 shows the labeling of the Sears Tower. Similarly, to augment the space, the user can proceed to create objects of the other two buildings in the image. Since this image does not contain any information about how these buildings are oriented or connected with respect to each other, the user must provide this knowledge.

This spatial correspondence is done in a three-dimensional layout canvas. It is here, hopefully, that the conflicts of contradictory and differential space can be resolved. The individual objects can be modified and shuffled independently. Once the event is set, the canvas is locked and can be used for navigation. The layout canvas can be seen in Figure 5-3.

The individual objects as well as the entire canvas are manipulated with the keyboard and mouse. The objects can be translated, rotated, and scaled.

³This approach is similar to the approach used in *WWMM*.

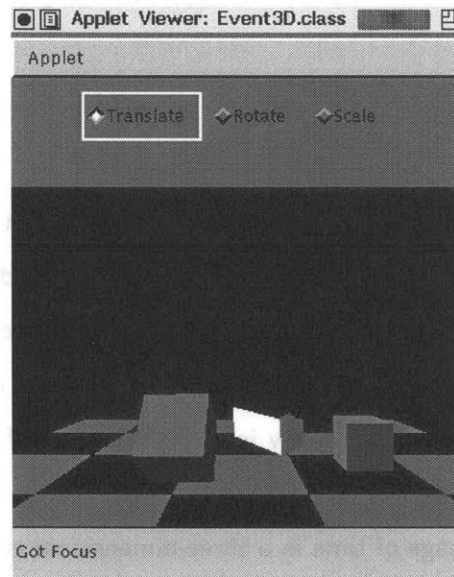


Figure 5-3: The layout canvas

5.5 Navigation

5.5.1 Illusory Nature of Space and Time

Real space and time are continuous and perceived as infinite in bounds. The events modeled here are discrete bounded space-time intervals. They are self-contained. In mapping events from real space to model space, there exist both qualitative and quantitative limitations.

Both spaces, however, beg the question - how do we perceive time and space? Both time and space are dependent on the movement and position of the user. In real space, since one cannot be in two places at once, one views an event at an instantaneous moment in time from only one viewpoint. When viewing an event in model space, one can view the same instant from several viewpoints. This is because of the ability to halt time, reverse it, or repeat it in model space.

As mentioned earlier, different users created different event models given the same images. Similarly, different observers have different notions of space and time when viewing the same event.

In making models of real events, one aspect of this work is to see how different navigating systems affect the viewing of different events, and how different viewers choose to best navigate them.

Many assumptions have been made to create this navigable medium. They are briefly described below.

5.5.2 Mapping

In mapping from real space to model space, one of the biggest variances lies in the relative scales of objects and the relative distances between them. The different objects in the models may have been created using different algorithms or images from cameras further away. Another difference is the concept of the unit of time in model space. It may have a linear mapping in the recording of the event, but it may vary in the playback or if the user chooses to alter the time of the event.

In order to trace the passage of time in a three-dimensional event, one first has to represent time in this three-dimensional model space. Depending then on how this is done, how does one steer through this multi-dimensional space that may be changing?

Most algorithms discard moving objects in the creation of models. This greatly simplifies the spatial correspondence in the images and results in a model at a specific instance in time.

Variances on this theme tried to incorporate the time variable within the model itself. An example of this is described in the following chapter.

5.5.3 Paths of Navigation

Most of the models originally created were static. With these static models, different navigation interfaces were created to see which was more efficient and more natural.

The first interface consisted of buttons that allowed the user to turn left and right, rotate left and right, move up and down, and move forward and backward. Another version of this interface allowed for continuous movement of the prior command until a “brake” was pressed. A compass-like symbol was also used in place of the buttons. A keyboard mapping eventually became a more convenient navigation control.

With these controls, the user could essentially go wherever they chose. The only constraints were the bounded clipping planes and the surfaces of the objects in the model. The system became a little disorienting when the user steered off the “floor” of the event, for it could be difficult to find their way back.

Another approach consisted of pre-planning a route through the event. This may be a route with the most objects in its path or a path which the creator found the most interesting. An

extension to this would be to gather the paths of the users and derive a path that reflects the most favored routes.

In navigating events with moving objects, the first interface allowed the user to choose a viewpoint and then control a scrollbar that represented a relative measure of the elapsed time. The user could move forwards or backwards. This proved quite effective. When something was moving, it proved more effective to steer around the entire event as opposed to steering directly into it.

The following chapter pictures some examples of individual models and how they were assembled. It also shows the interfaces that were used for navigation. The rendering of the surfaces onto the objects was not done for the navigation due to the complexity of the environment and speed.

Chapter 6

Examples of Events

This chapter analyzes and evaluates some of the modeling techniques and tours of the system. Perhaps the biggest cause for disorientation in the tour is the lack of texture and model discontinuities.

6.1 Model Conglomerate

The events are built as conglomerates of models. Given the two pictures of the same building from different viewpoints in Figure 6-1, their combined structure can be seen in Figure 6-2.

There is no visual information of the top of the building or the other sides. In some respects it resembles a movie-set. By navigating behind this model, there is some disorientation, especially when the model is seen in its inverted state. Texture mapping would eliminate this considerably.

6.2 Example of Event

Figure 6-3 depicts four different views from a simple static event. The people shown in the scene appear the same from every viewpoint¹. Figure 6-3(a) shows the event from a distance. Figure 6-3(d) is a view from inside one of the models looking out. The trees in the image are randomly generated and placed to correspond with the scene using the layout canvas. One of the original images used in the above event can be seen in 6-4. This particular interface is navigated using the scheme in Figure 6-5.

¹The people were segmented from the images by hand.



(a) Image A



(b) Image B

Figure 6-1: An example of parts vs. whole in creating the structure

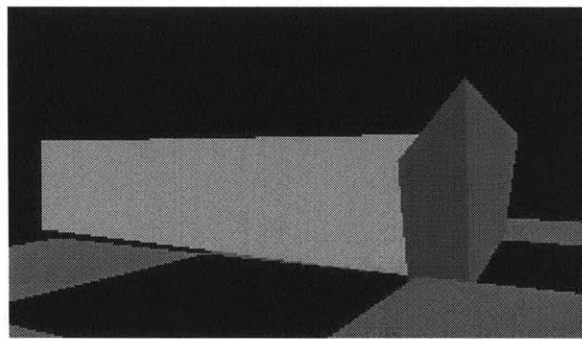
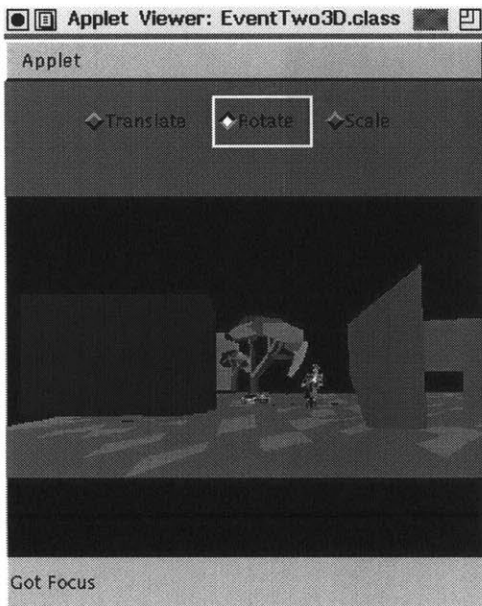


Figure 6-2: A partial reconstruction made with two parts

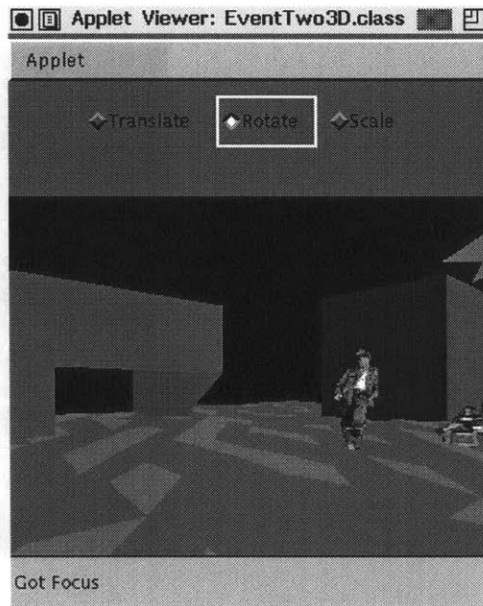
6.3 A Changing Event

In representing the element of time, what moves or changes within the space must be shown. This is currently done through the use of a sprite. The event shows the space with moving the entity displaced at different intervals in its own time and space. In this case it is a person.

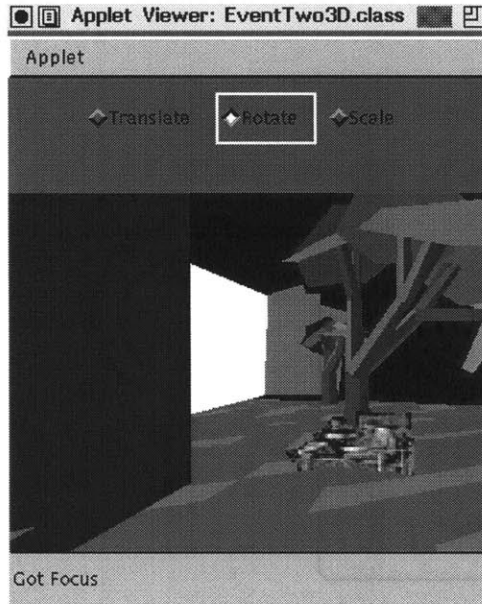
The image of the moving person is a two-dimensional image that moves from its starting location to its final location. People tend to be very difficult to model. In this case, the person's orientation is the same from every viewpoint, even when in motion. As time unfolds in space, the person traces the path set by the user. Another option shows the moving object at several points of its path but with incrementally different levels of transparency. The image becomes more opaque as it moves forward in time.



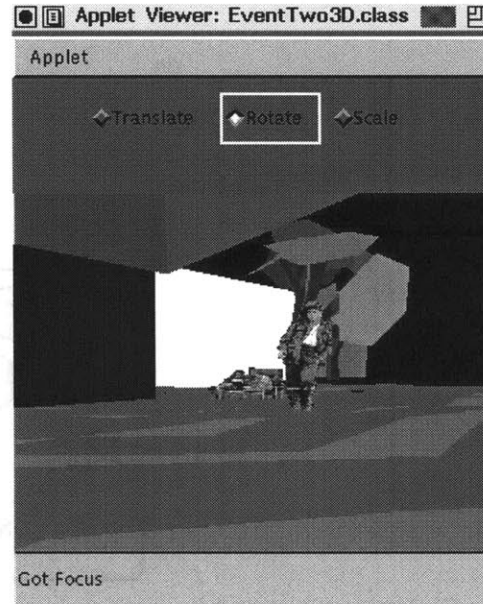
(a)



(b)



(c)



(d)

Figure 6-3: Four views from one event

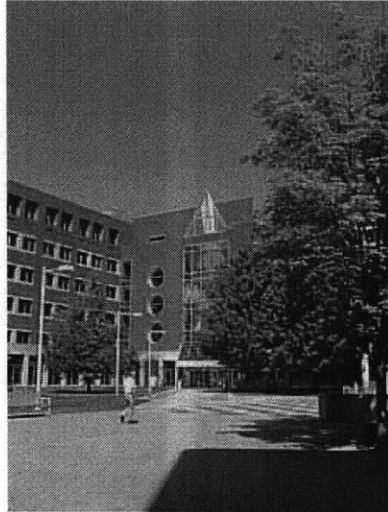


Figure 6-4: Picture of the medical center

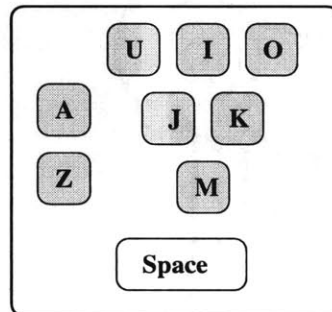
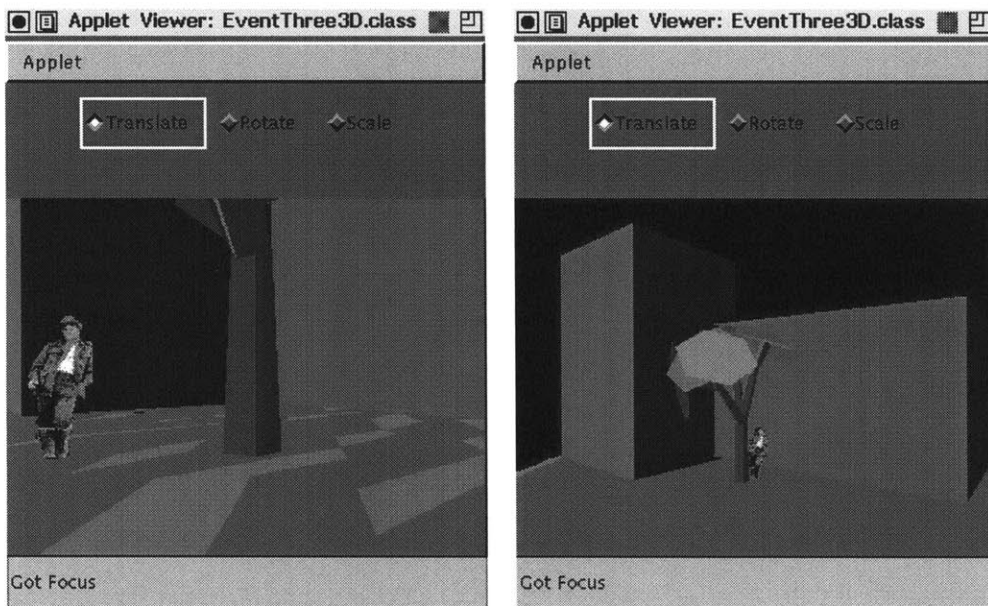


Figure 6-5: I=forward, M=back, K=right, J=left, O=rotate right, U=rotate left, A=up, Z=down, and Space=stop.



(a) Image A

(b) Image B

Figure 6-6: Two different views from a scene in motion

Chapter 7

Summary and Future Work

7.1 Summary

SpaceMap is a tool. Its purpose is to create a coherent space from a collection of still images and movies. This system succeeds in combining its many inputs into one self-contained environment.

It also allows for the user to produce and manipulate invented spaces using the models made from the SpaceMap tools or from VRML objects. It is, however, far from complete. The following section begins to address some areas for further examination.

7.2 Future Work

7.2.1 Implementation

Image Loading

Currently, stills and movies are stored in terms of their respective URL's. This has the obvious drawback of not being able to retrieve the same image at a later time due to URL changes or server inaccessibility. Although very memory intensive, one solution to this is to store the imagemaps as each URL is loaded.

Introducing New Algorithms

The addition of more modeling algorithms would also enhance the system. The output of these algorithms is also a major consideration. Ideally, the output should consist of three-dimensional points, lines, surfaces, and an index of the points and lines that reside on each surface. For

texture mapping purposes, the textures would be grabbed from the original images in the data base.

Simplify Interface

Interfaces evolve through many trials and many users. The current method of labeling frames in a movie is rather difficult. It is very time consuming and it relies heavily on the user. Incorporating a good feature tracker would simplify this process immensely, especially for the motion algorithm. Also, the rotations in the layout canvas should be made consistent.

7.2.2 Navigation

Adding Texture to the Models

Texture mapping of the models would greatly enhance the user's awareness of the space. It would allow for better orientation and conceptual reality.

Extreme Viewpoints

Currently, the extreme far and close-up viewpoints are bounded. At the farthest distance from the space, the user can see the entire space. At the closest distance, the user is restricted by the *walls* of the objects in the space. There is a sense of spatial opacity. It would be useful to provide a different graphical representation to link different spaces together at remote distances. At close distances, the user could be allowed more access to the objects in terms of textual data, links, or relevant images.

Time Element

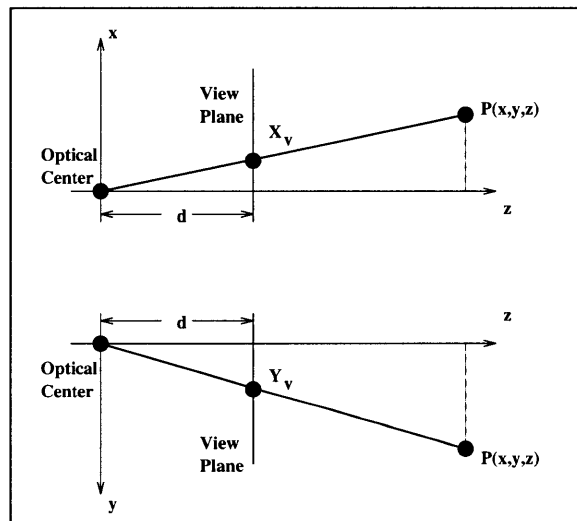
Time poses quite a paradox in this work. In the beginning of the process, the element of time is used in creating the static structure of objects from movies. In the final product, time must be recovered through user intervention.

That by which things are thus dated is available environmentally and yet not restricted to the world of equipment with which one currently concerns oneself. It is rather the case that in the world the enviroing Nature and public environment are always discovered along with it.

-Martin Heidegger

Appendix A

Projecting onto View Plane with Perspective Projection



$$\frac{x_v}{d} = \frac{x}{z}, \quad \frac{y_v}{d} = \frac{y}{z} \quad (\text{A.1})$$

$$x_v = \frac{d \cdot x}{z} = \frac{x}{z/d}, \quad y_v = \frac{d \cdot y}{z} = \frac{y}{z/d} \quad (\text{A.2})$$

M_{per} is the transformation of A.2 as a 4×4 matrix:

$$M_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \quad (\text{A.3})$$

Bibliography

- [1] The cyclorama. *Scientific American*, November 6, 1886.
- [2] Matthew E. Antone. Synthesis of navigable 3-d environments from human-augmented image data. Master's thesis, Massachusetts Institute of Technology, 1996.
- [3] Ali J. Azarbayejani. *Nonlinear Probabilistic Estimation of 3-D Geometry from Images*. PhD thesis, Massachusetts Institute of Technology, February 1997.
- [4] Ali J. Azarbayejani, Tinsley Galyean, Bradley Horowitz, and Alex Pentland. Recursive estimation for cad model recovery. *2nd CAD-Based Vision Workshop*, February 1994.
- [5] Ali J. Azarbayejani and Alex Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6), June 1995.
- [6] D.H. Ballard. Generalizing the hough transform to detect arbitrary shapes. 1980.
- [7] S.C. Becker and V.M. Bove Jr. Semiautomatic 3-d model extraction from uncalibrated 2-d views. *Proceedings SPIE Visual Data and Analysis II*, 2140, February 8-10 1995.
- [8] Shawn C. Becker. *Vision-assisted modeling for model-based video representations*. PhD thesis, Massachusetts Institute of Technology, February 1997.
- [9] V. Michael Bove Jr. *Synthetic Movies Derived from Multi-Dimensional Image Sensors*. PhD thesis, Massachusetts Institute of Technology, June 1989.
- [10] R.T. Collins and R.S. Weiss. Vanishing point calculation as a statistical inference on the unit sphere. *Machine Vision and Applications*, 3, 1990.
- [11] Trevor Darrell, Ali J. Azarbayejani, and Alex Pentland. Robust estimation of multiple models in the structure from motion domain. *CVPR*, 1994.

- [12] Albrecht Dürer. *The Painter's Manual 1525*. Abaris Books, Inc., 1977.
- [13] Olivier Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, 1996.
- [14] Henry N. Holtzman. Three-dimensional representations of video using knowledge based estimation. Master's thesis, Massachusetts Institute of Technology, 1991.
- [15] Berthold Klaus Paul Horn. *Robot Vision*. The MIT Press, 1986.
- [16] <http://www.inworldvr.com/>.
- [17] K. Kanatani. Statistical analysis of focal-length calibration using vanishing points. *IEEE Transactions on Robotics and Automation*, 8(6), December 1992.
- [18] Leslie Lamport. *LATEX, A Document Preparation System*. Addison-Wesley Publishing Company, 1994.
- [19] Brenda Laurel. *The Art of Human Computer Interface Design*. Addison Wesley Publishing Company, 1990.
- [20] Stéphane Laveau and Olivier Faugeras. 3-d scene representation as a collection of images and fundamental matrices. *INRIA, Rapport de recherche*, 4(2205), June 1994.
- [21] Andrew Lippman. Movie maps: An application of the optical videodisc to computer graphics. *Computer Graphics Proceedings, Siggraph'80*, 1980.
- [22] E. Lutton, H. Maitre, and J. Lopez-Krahe. Contribution to the determination of vanishing points through the hough transform. *IEEE PAMI*, 16(4), April 1994.
- [23] Brock D. Michel. A 3d environment for an observer based multiresolution architecture. *Spring 1997 - Simulation Interoperability Workshop*, 1997.
- [24] Robert Mohl. *Cognitive Space in the Interactive Movie Map*. PhD thesis, Massachusetts Institute of Technology, September 1981.
- [25] Michael Naimark. A 3d moviemap and a 3d panorama. *SPIE*, 1997.
- [26] Patrick Niemeyer and Joshua Peck. *Exploring Java*. O' Reilly and Associates, Inc., 1996.
- [27] Hans Rupprich, 1966. Dürers Schriftlicher Nachlass. p.104 Berlin.

- [28] Laura Teodosio. Salient stills. Master's thesis, Massachusetts Institute of Technology, 1984.
- [29] Richard B. Trask. *Pictures of the Pain*. Yeoman Press, 1994.
- [30] Shimon Ullman. *High-Level Vision*. The MIT Press, 1996.
- [31] Steve Waldman, 1997. Many talks with Steve regarding object organization.
- [32] Steven Yelick. Anamorphic image processing. Bachelor's Thesis, Massachusetts Institute of Technology, 1980.