FlyAbout: Spatially Indexed Panoramic Video

Don Kimber
FX Palo Alto Laboratory, Inc.
3400 Hillview Ave. Bldg 4
Palo Alto, CA 94304
kimber@pal.xerox.com

Jonathan Foote
FX Palo Alto Laboratory, Inc.
3400 Hillview Ave. Bldg 4
Palo Alto, CA 94304
foote@pal.xerox.com

Surapong Lertsithichai
Harvard Design School
48 Quincy Street
Cambridge, MA 02138
surapong@post.harvard.edu

ABSTRACT

We describe a system called FlyAbout which uses spatially indexed panoramic video for virtual reality applications. Panoramic video is captured by moving a 360° camera along continuous paths. Users can interactively replay the video with the ability to view any interesting object or choose a particular direction. Spatially indexed video gives the ability to travel along paths or roads with a map-like interface. At junctions, or intersection points, users can chose which path to follow as well as which direction to look, allowing interaction not available with conventional video. Combining the spatial index with a spatial database of maps or objects allows users to navigate to specific locations or interactively inspect particular objects**.**

Keywords

Panoramic video, spatial databases, video maps, interactive video, virtual reality

1. INTRODUCTION

There have been many attempts to recreate the experience of being in a particular place. Traditionally, this was done only through art and descriptive prose. The invention of photography provided a great step forward in the ease and accuracy with which places could be depicted. Motion pictures and video recording provided another important improvement. However, photographs or motion pictures are inherently passive media. They provide the user with particular views of the space, but only those chosen by the photographer or cinematographer in advance. The end user has no control over what is seen and when. Even if video is panned through different views from a given point, or is taken in motion along some path, the user's experience is of 'going along for the ride', but not of choosing where to go or to where to look on that ride.

Ideally, reconstructing the experience of 'being there' would allow the user to freely move around, and choose where to look. One approach to this requires a computer model of the space, and real time view rendering as a user virtually navigates through the space [1]. In most cases this approach is limited by the fidelity of the underlying 3D model, and cannot provide photorealistic views of the space even with computationally expensive rendering techniques. A promising recent approach is to record sufficient image data to approximate the flow of light in any direction and through any point in the space. This allows a view to be constructed for any particular position. Image data is represented in structures called "light fields" [2] or "lumigraphs" [3]. However, representing this data requires capturing four-dimensional images, which is prohibitively expensive to record and store for most practical cases.

Two image-based approaches have been developed which partially address the problem. One is panoramic imagery, which allows a user to freely chose which direction to gaze from a fixed point of view. Examples include Quicktime [4], IPIX [6], and BeHere [15]. Another approach is to associate video with paths the user may follow, or images with points along those paths [11]. FlyAbout combines these approaches, by associating panoramic images with points along spatial paths. The result is spatially indexed panoramic video. Though recording panoramic images at many finely spaced points could approximate it, in practice it is easier to record time-indexed panoramic video taken along one or more paths.

In this paper, we describe FlyAbout, a lightweight system for interactive navigation of a photorealistic representation of space. The FlyAbout system offers immersive, interactive navigation of a space without the expense of generating and rendering VR models. Spatial images are captured in a new media type called a VideoMap, which differs from conventional video in that it is both panoramic and spatially indexed.

Table 1. Comparisons of Spatial Capture Technologies

View:	*Positional Degrees of Freedom:*		
	0 *(discrete points)*	***1*** *(paths)*	***2,3***
Fixed	Photographs	Aspen Project [11], Views of Golden Gate [16]	CG rendered images.
Panoramic	QTVR [4], IPIX[6]	*FlyAbout*	CG based VR, Lightfields

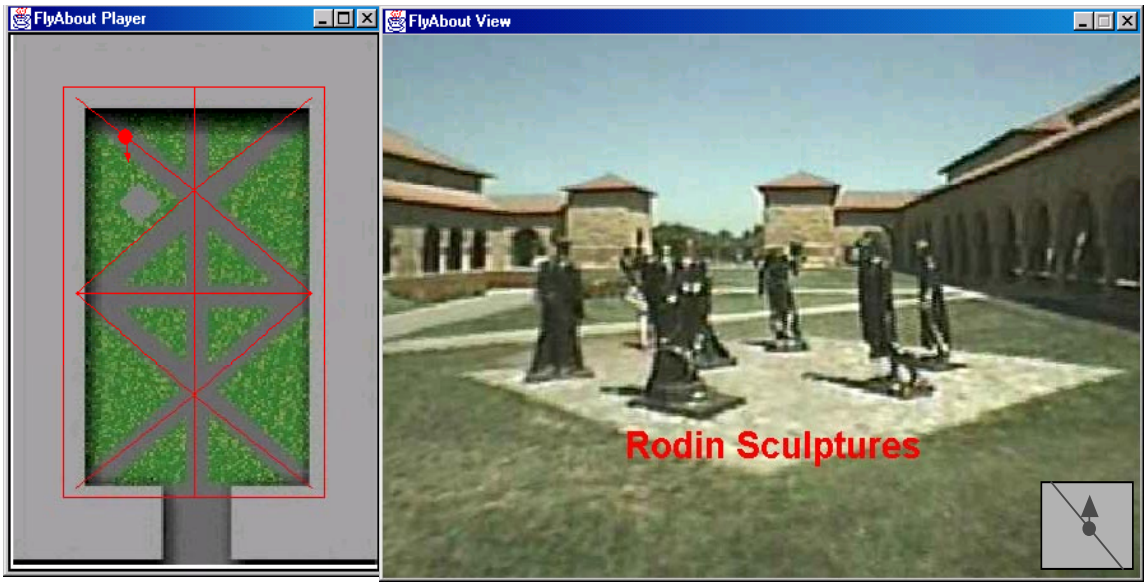

Figure 1 FlyAbout Video Player with Map.

FlyAbout player for viewing Stanford courtyard. Virtual paths correspond to actual paths through the courtyard. Paths are shown as red lines on map. The red circle indicates current view position, and protruding arrow indicates gaze direction. A small "heads up display" map gives cues to the available navigational choices. Objects from a spatial database are captioned in red on the image, at a location determined from the view angle and the object's coordinates.

Table 1 shows a conceptual map of various spatial capture approaches. There is a gap between still panoramic images (look anywhere from one point) and general 3D rendering (look anywhere from any point). The FlyAbout system falls neatly within that gap, allowing the user to look anywhere, but from a limited—yet still rich—range of points. This is a very practical compromise between viewpoint constraints and complexity. We can offer the user an experience almost as good as a "fly anywhere" system without the tremendous cost of capturing and rendering a full 3D modeled environment.

2. RELATED WORK

A number of systems exist for panoramic still images. One of the first systems in wide use was is QuickTime VR [4]. A more recent approach is IPIX [14], where wide-angle still images are recorded, and unwarped in a postprocessing step. BeHere delivers panoramic video over the Internet via a wide-angle lens and a streaming system [15]. None of these applications are spatially indexed. Some work has been done with spatial indexing, but without full seemless panoramic images presented as a continuous stream. Examples are the artistic works by Michael Naimark [13],[16], and the front, rear and side view images used in the Aspen project [11]. DizzyCity.com provides multiple panoramic views from New York City street corners (using IPIX images), but does not permit motion along paths from point to point. The work described in [11] also included images recorded with a special panoramic lens, but requiring a special projection display for viewing. In contrast, the work presented here combines spatial indexing with full panoramic digital video. Previous work also includes panoramic video recorded along interesting routes, but the paradigm for presenting this to users has been traditional time-based video [19]. Interesting work has also been done to derive spatial map information from the panoramic image [10].

In researching this paper, we have found that Michael Naimark and Andrew Lippman (among others) anticipated many FlyAbout innovations in the Aspen project [11]. We acknowledge the pioneering work done there, which is even more impressive given the expensive and cumbersome analog video equipment of two decades ago. In many ways, FlyAbout can be seen as a lightweight, inexpensive, and updated reincarnation of that work.

3. THE FLYABOUT SYSTEM

The FlyAbout system captures panoramic video from a 360° camera that is moved through a space of interest. The recorded video is spatially indexed and presented through an interface we call a VideoMap player. FlyAbout explicitly trades the time index for a spatial index. Paths can be navigated through a VideoMap by spatial relations such as "move forward" rather than the typical "play, stop, rewind" metaphors of traditional temporal video. Additionally, objects and locations can be inspected using a map-like interface: clicking on a map item brings the view to that particular location or object. The FlyAbout system displays both the panoramic video and an integrated spatial map that indicates the current viewpoint and gaze direction, as shown in Figure 1. Figure 2 shows the architecture of the FlyAbout system that is described in more detail below. A VideoMap may also contain text information, as seen in Figure 3. Navigation may be controlled through the panoramic video window, the spatial map, or the text display.

Figure 3. VideoMap Web Interface.

A web based VideoMap of Lake Tahoe, California. The upper right window displays the current view, which may be controlled by mouse or keyboard. The map at left shows available paths in red, and indicates the current position with a dot. The bottom frame provides information about the current view, and can also be used to control navigation via text hyperlinks.

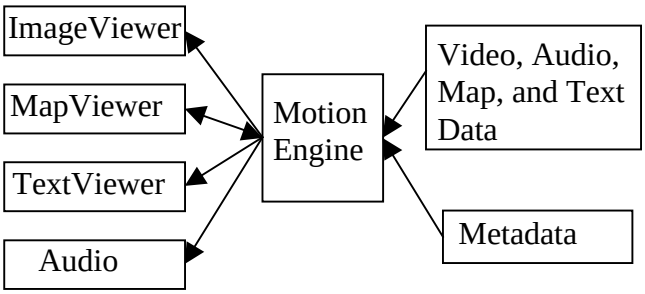

Figure 2. Overview of VideoMap Player.

VideoMap Player consisting of a motion engine and multiple media players

A VideoMap is constructed by recording the panoramic video and acquiring associated spatial metadata. The video is then processed and indexed, and rendered by a motion engine and VideoMap player. This allows users to virtually navigate the space. Section 3.1 describes VideoMap construction in greater detail.

3.1 Creating VideoMaps

Any panoramic camera can be used for recording VideoMaps, provided it can capture a horizontal 360° field of view. Though our VideoMaps could accommodate fully spherical panoramic video, we currently use a cylindrical projection in which views of the zenith and nadir are not available. This is for usability as well as practical reasons: there is typically little interest in viewing the sky or floor. Also the practical problems of capturing a fully spherical panoramic view should be obvious: where do you put the tripod, (let alone the videographers), so the view is not obstructed? Current panoramic camera designs include those based on wide-angle lenses [6],[15] mirrors [5] or multiple cameras [7],[9]. We used "FlyCam," a 4-camera 360° system developed at our laboratory [17], and shown in Figure 4. Because a FlyCam is compact and lightweight (only a few pounds, including mounting) it is relatively easy to rig various mobile camera platforms, including a dolly for indoor locations and outdoor footpaths, as well as a vehicle-mounted platform for roads. A car with a sunroof was especially handy for the latter. Four streams of video, one from each camera, were captured on four separate DV camcorders. (Though not essential, a fifth video

stream was also recorded, having a reduced-resolution camera image in each quadrant produced by a quad processor. This last stream was useful for several purposes: synchronization, sanity-checking, and to produce a reduced-resolution VideoMap.)

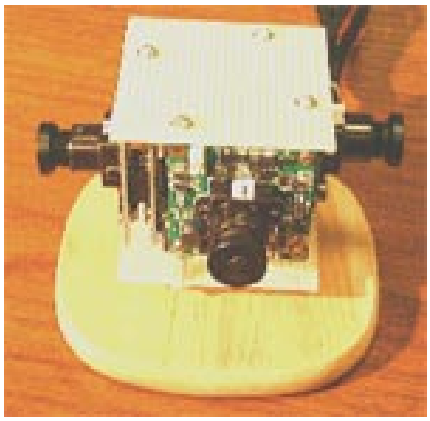

Figure 4. FlyCam used for Panoramic Recording.

When using a multiple-camera panoramic imaging system such as FlyCam, the multiple video streams must be synchronized. Several methods of synchronization are possible, including free-running, where the video recordings are not synchronized. Free-run recordings are synchronized in postproduction by finding a common start time, for example using a camera flash visible in all cameras. Though the individual time bases might drift, in practice they are stable enough to be nearly frame-accurate for recordings of an hour or less. Alternatively, recordings may be synchronized using a timecode generator, either on the audio track or as part of the recorder transport. Another alternative is to synchronize the time base of the video cameras using either "genlocking" or a time base corrector, and finding a common start frame. In this case all recordings will be frame-accurate with respect to each other. For our most recent recordings, we write SMTPE timecode on one audio track for each video recording, and those timecodes are used in postproduction to synchronize frames.

Spatial metadata is acquired in concert with the panoramic video, describing the spatial location of the camera at any given moment. This can easily be done using available Global Positioning Satellite (GPS) equipment or any of a number of other methods. Relevant text may also be collected and marked with either time or, equivalently, position. During playback, this metadata assists navigation by displaying spatial location on a generated map, or is integrated with an existing map or spatial database. Relevant text or hyperlinks can be displayed at the appropriate positions or views. Map data may be taken from existing maps, aerial photographs, spatial databases, or other available images, for example a building floorplan. Given GPS or similarly accurate spatial data, a simple map of the camera paths can be rendered directly from the data.

Camera paths are conveniently represented as data tuples of the form (x,y,a,t) where x and y are the location of the camera, a is the orientation angle of the camera, and t is the time. (Height (z) data can also be captured if relevant). Such a tuple could be given for every frame of video, however, assuming the camera motion is relatively smooth, a much smaller number of tuples can be provided, and linear interpolation can be used to estimate camera position and orientation at intermediate points. Spatial sensors, such as a GPS unit, may produce the spatial metadata automatically. Other sources of spatial data include radio beacons or dead reckoning from acceleration or vehicle speed and heading sensors. Alternatively, spatial data could be produced by image processing of the panoramic images, for example by using triangulation from known points to determine camera position. In simple cases, it is feasible to produce the spatial metadata by hand, given a map, known paths, and landmarks visible in the video. In the absence of specific position data from external sources, position estimates can be derived from analysis of the panoramic video data. The speed and heading of camera motion can be estimated from analysis of "optical flow" in the image [12].

Ideally, the panoramic video should be collected when there are no people or other objects moving through the space. These objects will appear to move backwards when the video path is traversed in a direction opposite to the direction it was captured in. However, if the camera is moved slowly, so that many images are available for any given position, a median filter or other image processing could be used to eliminate the transient objects. Alternatively, a single path can be captured in both directions; during playback the one with the same direction of travel is displayed to the user.

3.2 Rendering & Motion Engine

In multiple camera systems such as FlyCam, the recorded video streams must be stitched together before they can be presented as a smoothly pan-able virtual view. In our FlyCam system this is done by first mapping each camera view, modeled as a perspective projection (like a pinhole camera), onto a portion of a cylinder. The details of this procedure can be found in [17]. In the cylindrical projection, the images from adjacent cameras overlap slightly; cross fading these regions make camera border artifacts less conspicuous. Though the FlyCam system can do this processing in real time, it is done off-line for creating Video Maps which need not be available in real time.

Offline, video from the DV camcorders are uploaded as AVI files over a FireWire interface. The resolution of the images is 720x480. The multiple AVI files are rendered into a motion jpeg file that contains a complete 2400x480 panoramic image for each time frame. (The horizontal image resolution is somewhat less than 4 X 720 because of image overlap.) During playback, the selected view is cropped from the full panorama. Naïve cropping results in distortions as the cylindrical projection is mapped onto a planar display: straight lines will appear curved. It is straightforward to map the cylindrical projection back to a perspective representation so that straight lines, such as building edges, appear straight. However, we find that a cylindrical projection is often not objectionable, particularly while viewing natural outdoor locations where there are few straight lines.

Because the VideoMap interface of FlyAbout is spatially rather than temporally oriented, a core component of the system is a 'motion engine'. In traditional video, including panoramic video, frame images are ordered by time, and motion corresponds to increasing time. A typical interface for a panoramic video appliance has primitives like:

```
Interface VideoController {
   Play(speed); // in seconds per second
   Stop();
   SeekToTime(time);
   GetCurrentTime();
   SetGazeDirection(angle);
   SetFieldOfView(angle);
};
```

Video frames are sent to a display, and this interface can be used to control the speed, or position of the video. A speed of unity means the video is played at normal speed, but other speeds could be used to play at multiples of real time, including fast-forward, slow motion, or even reverse for negative values. The interface also allows the view direction and degree of zoom to be changed.

The FlyAbout controller sends frames to a display in a similar manner, but is spatially based. Rather than playing video by moving forward in time, video is played by moving along a path at some velocity. The interface looks something like this:

```
Interface FlyAboutController {
   Move(speed, navigationInfo);
   // speed in meters/sec
   Stop();
   MoveAhead(distance, navigationInfo);
   MoveTo(position);
   GetCurrentPosition();
   SetGazeDirection(angle);
   SetFieldOfView(angle);
};
```

The `Move` function indicates motion along the current path at a given speed. The `NavigationInfo` structure specifies parameters that control the subtleties of navigation. For example, it indicates the preferred direction, and what to do when an intersection is reached. The default rule is that motion always proceeds in the direction closest to the preferred direction. `NavigationInfo` can also specify a mode in which the preferred direction is updated to match the forward direction along the path. This means that motion will always be as 'straight as possible', and when an intersection is reached, the path chosen will be the path as close as possible to current direction.

The FlyAbout controller consists of a motion engine, which continually updates its viewpoint and gaze direction based on the current motion and navigational input. The controller maps positional information back to temporal information, using the spatial index to get the actual time-based video frames.

3.3 Displaying FlyAbout Video

Presenting a VideoMap to the user can be done in a number of ways. However, a naïve playback of the full panorama is not particularly useful, and in fact can be counterintuitive. When displayed on a conventional planar display, an unwrapped cylindrical panoramic image is not easy to interpret, especially when moving: one quadrant of the image appears to be approaching, one receding, and the other two moving past in opposite directions.

We prefer to display a virtual view towards a particular direction, by cropping the cylindrical panorama to a normal aspect ratio and mapping to a planar projection. The user may then change their view direction to pan around the full panoramic image, and may also choose to zoom in or out. For a more compelling experience, a large virtual view can be displayed on a plasma or projection screen. The FlyAbout architecture allows multiple simultaneous views from the same point. So two or more large displays in a room could present multiple virtual views at a fixed relative angle, for example to simulate multiple windows on a vehicle. We have also implemented FlyAbout playback using a head mounted display, with a head tracker used to control the gaze direction. This gives a photorealistic, immersive "virtual reality" experience.

3.4 VideoMap Player

Our primary user interface is a VideoMap Player that displays the cropped panorama alongside a spatial map (as shown in Figure 1). The map indicates the users viewpoint and gaze direction; clicking on the map is one way to navigate. Our system supports VideoMaps in a standalone player as in Figure 1, or as a dynamic web page, as shown in Figure 3. In addition to the video and map windows, text links can also be included. Navigation can then be controlled using video, map, or text.

Dragging the mouse in the video window controls gaze direction, and together with keyboard commands can control forward or backward motion. We have experimented with a variety of interface configurations, and have found that using mouse dragging to control gaze, and arrow keys to control motion, is adequate in many cases. Key commands can also be given to stop or reverse direction. Mouse dragging together with the shift key, can be used to precisely control motion or speed. We have implemented a joystick control, in which the joystick is moved in the direction of desired motion, and rotation of the joystick is used to rotate gaze direction. Alternatively the mouse can be used in the map window to drag the location (indicated by a 'you-are-here' dot) or gaze direction. Popup menus in the map window can be used to move the view as near as possible to a selected location. Clicking on map objects can also move the gaze direction to that object.

Text can also provide additional information to the user. As the viewpoint moves, the text display can be updated to provide descriptive text. Conversely, text hyperlinks can change position or gaze direction. Links may be extracted from a GIS database, for example to indicate locations of nearby hotels. A text link to a particular hotel could move the viewpoint to a position near that hotel. Additionally, a URL can be generated at any time for the current viewpoint and used to 'bookmark' points of interest.

Figure 2 shows the basic player structure, where the motion engine, display, and UI are all implemented as one program in a single address space. This system has been implemented and can render VideoMaps stored on a local hard drive, CD or DVD. A networked FlyAbout system has also been implemented, where the VideoMap is presented in a web browser, as shown in Figure 3. In the current implementation, panorama images are updated by server push. A streaming video format could be used as well, although the buffering must be kept small enough so that latency does not ruin the sense of interactivity. The map viewer is implemented as a Java applet. A simple HTTP protocol controls the motion engine by sending HTTP GET requests with special

control URLs. This allows the interface to be controlled by any number of sources, for example a Java applet, links in an HTML page, or by any other device or system that can generate HTTP requests.

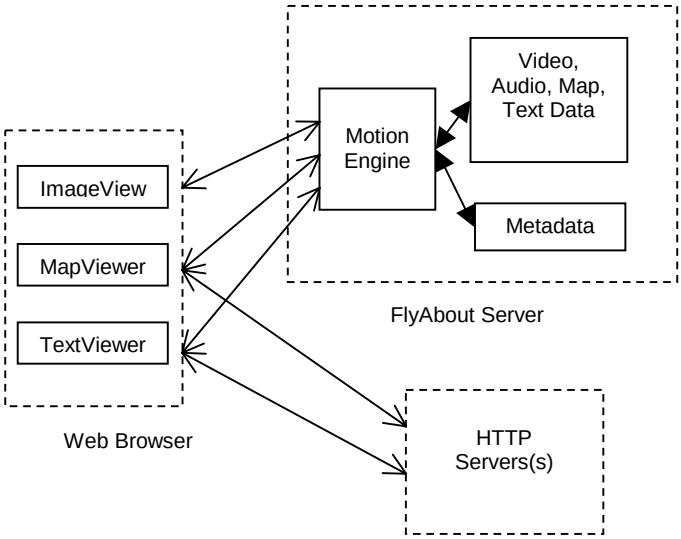

Figure 5. Web based VideoMap.

The FlyAbout HTTP Server. The client browser image is updated by server push, and the client's interactive map viewer is implemented as Java applet. Text associated with the current location is displayed in another frame.

3.5 Navigating VideoMaps

When viewing VideoMaps, users may control their motion in addition to controlling the gaze direction. On any given path, the user may move in either of two directions along the path, and may control the speed. We are experimenting with a variety of navigation controls such as a mouse, a joystick, and pedals. When motion brings the view position to a path intersection, users may indicate which of the several paths they will take. There are a variety of possible interfaces, for example "left, right, straight" buttons that are activated when there is a directional choice to be made. For some applications, the user can control the direction of motion independently of the gaze direction. This is useful when examining a particular object by navigating around it. In this case, controls can allow the user to "snap" the gaze to a particular object, such as a sculpture or building, so that it is in constant view as the user moves around it. This latter feature is particularly useful in situations such as a sculpture garden or car sales lot where objects, rather than the scenery or paths, are the interesting points.

In most applications, there is no real need for completely independent path direction and gaze direction control. A simpler approach is to use the gaze direction as the preferred direction at an intersection. When motion brings the view position to an intersection, the path closest to the current view direction is chosen. Another simplified navigation mode uses the current direction of motion as the preferred direction. In this case the rule is to go as 'straight as possible' at an intersection. This can be useful for a 'hands off' tour in which the motion proceeds continuously along a path.

It is important for the user to be aware of navigational choices, in particular where intersections are located and the paths available from them. In many cases this may be obvious from the view. For example, if the VideoMap is collected along roads or obvious paths, then it should be made clear to the user where they may turn onto an alternate path or if they are constrained to a single path. (For example in the courtyard shown in Figure 1, the paths available for virtual navigation match the footpaths through the courtyard and are clearly visible in the video.) However, it is not obvious what paths are available in the absence of visual cues. Also, for footage collected along a road or driveway, some turnoffs or driveways may not be accessible. In these cases, the user could determine their choices by consulting a map interface, as in Figure 3. However consulting a map may distract from the immersive experience. An alternative is to show a small map segment oriented to the users current gaze direction. This can be seen in the bottom right corner of Figure 1

Another useful function is to allow the user to specify an object or location to look at rather than a specific direction. As the user moves around various paths, their gaze remains fixed on the object. We have implemented this as a 'focus' location that can be set to any point in the map. This makes it easy to view an object from any position. This functionality is similar to "object movies" of QuickTime VR, which provide views of a single object from multiple directions. However, unlike object movies, it is not necessary to carefully point the camera at the object when the video is collected. Furthermore, some path segments may be used as part of the paths for navigating around more than one object. This functionally is particularly useful for e-commerce or other applications, for example in a car sales lot. If cars are arranged in a grid, then capturing FlyAbout video along all grid borders then lets the FlyAbout system automatically generate a view of any car from any angle.

A particularly powerful aspect of this system is the ease with which hyperlinks can be made between video images and spatial objects. When integrated with a spatial database of object coordinates, it is simple geometry to find the best location and gaze direction to view any object. Thus clicking on a map object can bring up a view, or in reverse, it is easy to answer the question "what am I looking at" by finding the closest object to the gaze vector. An ultimate application of this is to automatically annotate the video image, for example by overlaying text as in Figure 1, where the Rodin sculptures are indicated.

4. EXAMPLE VIDEO MAPS

We have experimentally produced several VideoMaps for a variety of different spaces and paths including a conference room, an outdoor courtyard and indoor lobby, a courtyard at Stanford University with a number of intersecting paths, and a highway around Lake Tahoe, California.

4.1 Kumo Conference Room

We produced our first VideoMap of a conference room at our laboratory. The camera was mounted on a dolly with swivel wheels, so we were able to keep the camera orientation constant as a total of two minutes of video was collected. This web based VideoMap is shown in Figure 6. The room map used was a top view of a 3D room model that had been derived from an architectural blueprint.

Figure 6. VideoMap of Conference Room.

4.2 Courtyard with Fountain

The first outdoor VideoMap we produced was along a path around a fountain on Xerox campus in Palo Alto. About 5 minutes of video was collected, again with a constant camera orientation. Chosen paths included some that went up stairs, and also a short segment entering and exiting the lobby of a building. Capturing paths along stairways required more teamwork then ingenuity: two people lifted the camera dolly and climbed the stairs as smoothly as possible. Figure 7 shows the VideoMap; the view location is at the fountain viewing the main building. Figure 8 shows the same VideoMap, although here the view is at the lobby entrance. The two different views are reflected in the different view location of the two maps.

4.3 Stanford Courtyard

We produced a VideoMap of a courtyard adjacent to the Main Quad on the Stanford campus. Our first attempt, on a busy Saturday, was hampered by too many visitors. Seeking a less crowded time, our next attempt was during a morning hour; however the low angle of the sun caused objectionable artifacts when directly in the camera view. (This is a general problem with panoramic imaging of any type.) We had best success at a mid morning on a weekday; a view of the resulting VideoMap was shown in Figure 1. The courtyard has a number of paved paths across a lawn, and we collected video along each of the paths, for a total of about 15 minutes of video taken in four shots. The paths were paved with brick, causing some shaking of the camera. The effect during playback is not completely objectionable, and suggests walking rather than smooth motion or extreme shaking.

Figure 7. FlyAbout VideoMap of Courtyard.

Figure 8. VideoMap of Courtyard.

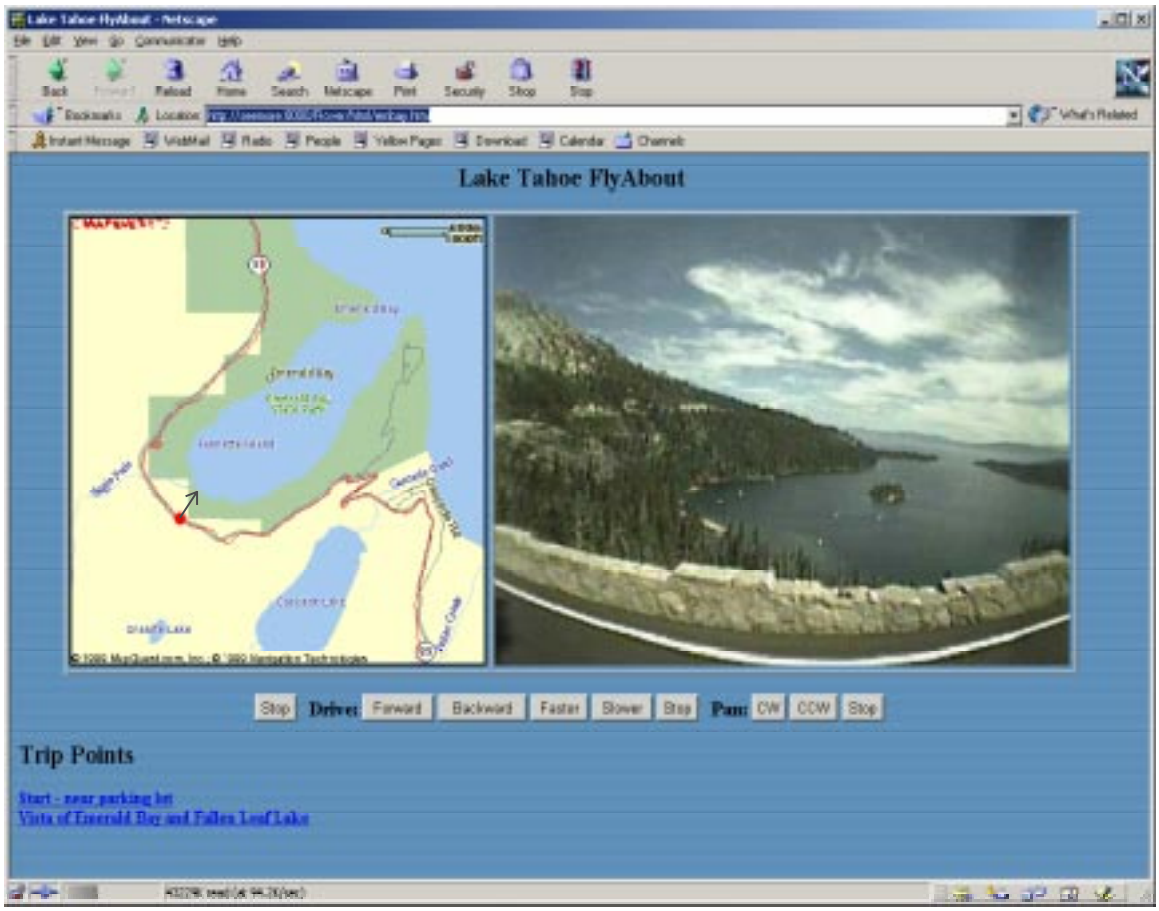

Figure 9. VideoMap of Emerald Bay.

A web based VideoMap of Emerald Bay in Lake Tahoe, California. The upper right window displays the current view, and may be controlled by mouse or keyboard actions. The map at left shows available paths, and indicates the current position and view direction. The text display in the bottom frame provides information about the current view, and can also be used to control navigation via hyperlinks.

Two aspects of the Stanford courtyard made it an interesting choice for a VideoMap. In particular, the courtyard had a rich set of paths, including several intersections where six paths meet. This was useful for prototyping a navigation interface, and had the added benefit in more navigational choices for the user, and thus a more interesting VideoMap. Additionally, because the video was shot along just those paths a person would normally walk, the choices of where to go was enhanced by the visible paths.

4.4 Lake Tahoe

We produced a VideoMap from a drive along the western side of Lake Tahoe. The video was shot on a Monday in late August. We started in Tahoe City around 11am, and drove south to Stateline, collecting about 90 minutes of video. The camera was mounted on a tripod extended from the sunroof of a compact car. A GPS unit was used to record spatial metadata on a laptop computer. In hindsight, it might have been better to drive in the opposite direction so that the vehicle was closer to the guardrail. This would have improved the camera views overlooking the lake.

The Tahoe VideoMap covers a large enough area that a multiscale interface is valuable. Figure 3 shows a large-scale view that includes the entire lake as well as the path around it. The red 'you-are-here' marker can be dragged along the path, moving the view along the drive. Precise viewpoint positioning is difficult with a map at this scale. A 'zoomed in' map can be chosen for a given region, which is more useful for precise navigation. Zooming and re-centering are controlled by a popup menu over the applet showing the map. The maps are retrieved as needed from a web based map provider. [20] Figure 9 shows a map expanded to just the Emerald Bay region. We also added text hyperlinks to Tahoe VideoMap that allow movement to preselected places. These links were added manually but could easily be extracted from GIS database.

5. DISCUSSION & FUTURE DIRECTIONS

We have experimented with several different VideoMaps and features of the FlyAbout system. From our initial experiments, it seems that VideoMaps are well suited to "virtual visits" of remote locations. Because they are spatially rather than temporally based, they are not appropriate for capturing activities. Obvious VideoMap applications are travel and tourism, real estate, or documenting interesting spaces like museums, scenic areas, or any places with scenic or educational value.

There is no inherent reason that VideoMaps could not be produced for entire cities or regions, and integrated with other VideoMaps at any scale. For example a more comprehensive VideoMap of Lake Tahoe could allow tourists to virtually drive to one or more hotels or resort destinations, and then to walk through those resorts to view their grounds and amenities. Capturing all streets of an entire city is certainly feasible, and the navigational abilities of a VideoMap would let users virtually drive any desired route. We plan to build applications which will help us better understand what capabilities are needed for large and multiscale VideoMaps.

We are also improving the VideoMap interface in a number of ways; for example direct integration with spatial databases or GIS data. It is straightforward to add navigational features, such as automatically generated routes for virtual tours. Integrating this with a mapping service could provide a VideoMap preview of driving directions to a chosen destination, showing what turns and landmarks would look like en route.

The VideoMap interface is capable of more advanced interaction. For example, users could select objects in the video view and get text information about them, see their locations on the spatial map, or move their viewpoint close to those objects. On the production side, we are in the process of building a six camera FlyCam for higher image quality VideoMaps, and are working on image processing methods for automatic generation of spatial metadata.

6. CONCLUSION

In conclusion, we have presented a spatial capture that is far more interactive than existing panoramic imaging systems, yet is far more practical than a complete 3D rendering solution. Few other approaches are lightweight enough to give a similar sense of presence over the Web, or scalable enough to cover large and dense geographical areas. Letting the user choose location and view, and direction at intersections gives an immersive and interactive component that is not present in static panoramas, even of motion video. In addition, integrating spatial location with the time index automatically makes the system "object-centric" from spatial metadata. This allows the panoramic images to be accessed by object, rather than by time index or spatial location, which removes a significant navigational burden from the user. An object can be viewed simply by clicking on its map representation, rather than by having to find and specify its time location and view direction in the panoramic video.

7. REFERENCES

[1] H. Rheingold, *Virtual Reality*, N.Y.: Summit, 1991.

[2] M. Levoy and P. Hanrahan. Light field rendering. In *Proc. SIGGRAPH* '96.,1996.

[3] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The Lumigraph," in *Proc. SIGGRAPH'96*, pp. 43-54, 1996.

[4] S. Chen and L. Williams, "View Interpolation for Image Synthesis." *Proc.SIGGRAPH'93*, pages 279-288, August 1993.

[5] S. K. Nayar. "Catadioptric Omnidirectional Video Camera." In *Computer Vision and Pattern Recognitio*n, pages 482–488, June 1997.

[6] IPIX, the Interactive Pictures Corporation, http://www.ipix.com

[7] M. Nicolescu and G. Medioni , "Electronic Pan-Tilt-Zoom: A Solution for Intelligent Room Systems," in *Proc. IEEE International Conference on Multimedia and Expo*, August 2000.

[8] FullView wide-field panoramic camera, http://www.panoramtech.com/product2000/sources/FV360.htm

[9] Immersive Media, http://www.immersivemedia.com

[10] H. Kawasaki, K. Ikeuchi, M. Sakauchi, "Automatic 3D City construction System using Omni Camera," in *Proc. IEEE International Conference on Multimedia and Expo*, August 2000.

[11] A. Lippman, "Movie-Maps: An Application of the Optical Videodisk to Computer Graphics" *Computer Graphics* **14**(3), July 1980

[12] S. Beauchemin and J. Barron. The computation of Optical Flow. *ACM Computing Surveys*, Vol. 27, No. 3, 1995.

[13] M. Naimark, "VBK - A Moviemap of Karlsruhe", in: Tommorrow's Realities – In *Proc. SIGGRAPH* '91, Las Vegas 1991

[14] IPIX Corporation. http://www.ipix.com .

[15] BeHere Corporation. http://www.behere.com .

[16] M. Naimark, "Views of the Golden Gate", Exhibit at the Exploratorium, San Francisco, Fall 2000.

[17] J. Foote and D. Kimber: "FlyCam: Practical Panoramic Video," in *Proc. IEEE International Conference on Multimedia and Expo*, vol. III, pp. 1419-1422, August 2000.

[18] DizzyCity Corporation. http://www.dizzycity.com

[19] T. Pintaric and U. Neumann. "Demonstration of Immersive Panoramic Video," *ACM Multimedia 2000*, Los Angeles.

[20] Mapquest.com, Inc. http://www.mapquest.com