

# Image-based Navigation in Real Environments Using Panoramas

Derek Bradley<sup>1</sup>, Alan Brunton<sup>1</sup>, Mark Fiala<sup>1,2</sup>, Gerhard Roth<sup>1,2</sup>

<sup>1</sup>SITE, University of Ottawa, Canada

<sup>2</sup>Institute for Information Technology, National Research Council of Canada

**Abstract** – We present a system for virtual navigation in real environments using image-based panorama rendering. Multiple overlapping images are captured using a Point Grey Ladybug camera and a single cube-aligned panorama image is generated for each capture location. Panorama locations are connected in a graph topology and registered with a 2D map for navigation. A real-time image-based viewer renders individual 360-degree panoramas using graphics hardware acceleration. Real-world navigation is performed by traversing the graph and loading new panorama images. The system contains a user-friendly interface and supports standard input and display or a head-mounted display with an inertial tracking device.

**Keywords** – Panoramic imagery, Image-based rendering, Immersive environments.

## I. INTRODUCTION

This research is part of the NAVIRE project [5] at the University of Ottawa. The goal of the project is to develop the necessary technologies to allow a person to virtually walk through a real environment using actual images of the site. This paper presents an image-based panorama rendering and navigation system that helps to achieve this goal. Our system generates cubic panoramas using a high resolution multi-sensor camera and then renders the 360-degree panoramas using hardware accelerated graphical techniques. Navigation in the real environment is achieved by localizing the panoramas on a common map and establishing connectivity among them. This creates a 2D graph topology that can be traversed interactively, simulating navigation along a user-defined path. The system includes a simple user interface with dynamic onscreen visual cues. Standard input and display devices are supported, along with a head-mounted display (HMD) that uses an inertial tracking device to track the user's view direction.

The remainder of this paper is organized as follows. Section 2 outlines related work in panoramic imagery and immersive environments. Section 3 describes our method to capture and generate cubic panoramas. Our panorama rendering and navigation techniques are presented in Section 4. Section 5 discusses some problems that were encountered and future work is presented. Finally, we conclude with Section 6.

## II. RELATED WORK

An initial framework for an immersive environment was given by Fiala [3] where the environment could be defined by

a set of panoramas captured by catadioptric cameras, and the panoramas were defined as cubes. Sampling the plenoptic function from a single point and storing it in a cubic format is an idea used often in computer graphics. Fiala chose this idea for his framework because it would allow for the creation of new interpolated panoramas for 4 of the 6 sides using the image warping results of Seitz and Dyer [6]. Fiala describes the cubic format as being efficient for rapid rendering in modern graphics cards, although it was not implemented in his paper.

In this paper we extend the work of Fiala in several ways, producing a more usable system. Most importantly the catadioptric cameras used to create the cube panoramas are replaced by a multi-sensor panoramic camera with much higher resolution. Fiala showed the limited resolution for several camera/mirror configurations, since the whole panorama is contained in an annular region in a single image. Additionally, we implement the graphics hardware acceleration, and add user-friendly map functionality. In our system, the user can see where they are and which direction they are facing in a small inset map, and they can alternate between views of full screen maps. In the prototype system of Fiala, one could get easily lost and disoriented.

Many previous systems use a single image sensor (CCD or CMOS 2D array) and a combination of lenses and mirrors (a catadioptric system) to capture the light from all azimuth angles and a range of elevation angles [3, 2, 1, 4]. The GRASP webpage<sup>1</sup> details many of these systems. The use of multiple image sensors provides a much better result. In our system, a higher resolution can be achieved with a significantly simpler optical system because multiple sensors are employed.

## III. PANORAMA IMAGE CAPTURE

Panoramic images are captured using the Point Grey Ladybug spherical digital video camera<sup>2</sup>. This multi-sensor system has six cameras: one points up, the other five point out horizontally in a circular configuration. Each sensor is a single Bayer-mosaicked CCD with 1024 x 768 pixels. The sensors are configured such that the pixels of the CCDs map approximately onto a sphere, with roughly 80 pixels overlap between adjacent sensors. An example of images taken from the Ladybug is shown in Figure 1.

<sup>1</sup> [www.cis.upenn.edu/~kostas/omni.html](http://www.cis.upenn.edu/~kostas/omni.html)

<sup>2</sup> [www.ptgrey.com](http://www.ptgrey.com)



Figure 1: An example of demosaicked images from the Ladybug camera.

These six images need to be composited together to form a panoramic image. This is done by projecting a spherical mesh into the six images and using the projected coordinates to texture map the images onto the mesh. The mesh can then be rendered using standard graphics techniques with any perspective view from the center of a common coordinate system for all six sensors. The overlap regions are fused by feathering and blending the overlapping images. Blending is performed with a standard alpha blending technique in OpenGL.

Given that any perspective view from the center of the Ladybug can be rendered, we generate cubic panoramas by rendering the six sides of a cube from this common center of projection. That is, we render six perspective views with 90 degree fields of view in both the positive and negative directions along the  $x$ -,  $y$ - and  $z$ -axes. Figure 2 illustrates the cubic panorama generated from the images in Figure 1. Notice that since no camera points downward, there is a section of the panorama that cannot be computed. This is represented by a black hole in the bottom face of the cubic panorama.

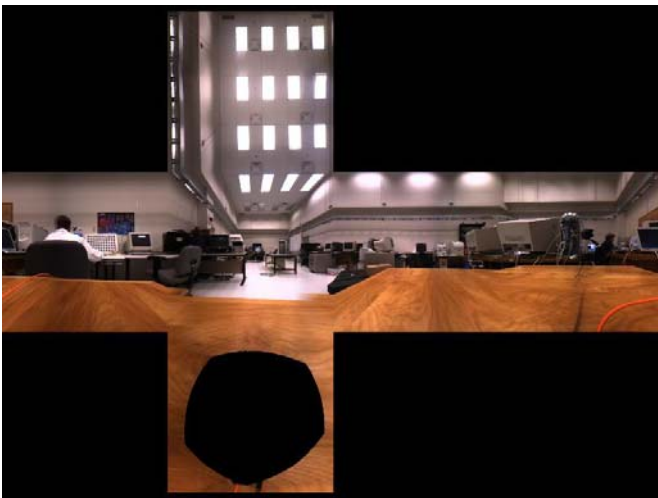


Figure 2: The images from Figure 1 projected into a cubic panorama.

#### IV. RENDERING AND NAVIGATION

Cubic panoramas are displayed in a “cube-viewer”, which uses accelerated graphics hardware to render the six aligned images of the cube sides. The position of the viewpoint for rendering is fixed at the origin of the 3D world. 6 quadrilaterals (12 triangles) are created and positioned in

axis-aligned static locations around the origin such that they connect to form a cube. The cube is centered at the origin, and the faces are oriented so that the face normals point inward. The 6 cubic panorama images are used as face textures and standard OpenGL texture mapping is used to render the cube quadrilaterals. This creates a 360-degree panorama from the perspective of the world origin. Although the viewpoint location remains static, the view orientation can be manipulated to see the panorama in any direction using polar coordinates.

The 360-degree view orientation is controlled in real-time using standard input (ie: mouse and keyboard) or an Intersense inertial tracker<sup>3</sup> mounted on a Sony i-glasses<sup>4</sup> HMD. Figure 3 shows the viewer display interface.



Figure 3: Display interface of the cubic panorama viewer.

The current pitch and heading of the user is displayed on the interface, as well as a 2D map of the environment. The environment consists of a number of panorama locations connected in a graph topology, indicated by white circles and black lines on the map in the bottom right corner of Figure 3, and enlarged in Figure 4. The current location is depicted by a red circle, and the current orientation by a red arrow. Users can move to any neighbouring panorama on the map. Blue arrows are rendered in the display on the panorama (and on the heading dial) to indicate possible directions of motion. An arrow becomes red when the user is facing that direction, to cue to the user that they may move forward (visible in Figure 3).

<sup>3</sup> [www.intersense.com](http://www.intersense.com)

<sup>4</sup> [www.sony.com](http://www.sony.com)

When a user moves to a new location, the new panorama is loaded from disk and displayed in the viewer. Recall that the actual cube and viewpoint location remain fixed in the 3D world. Moving to a new location is simulated by changing the face textures of the cube quadrilaterals from the previous panorama images to the new one. A global orientation is maintained during navigation by specifying an offset for each panorama cube. When a cube is loaded, the local orientation is determined by combining the global orientation and the cube offset. The local orientation is updated automatically when the new cube is loaded, so the user gains a sense of physically moving in the direction they were facing, even though their position does not change in the 3D world. Many panoramas are captured in sequences to prevent large jumps when navigating. Manipulating the local orientation within a panorama simulates moving your head in the real world. Moving to a new panorama in the system simulates walking in the real world. Cubes also contain GPS data for outdoor environments, which can be used for automatic localization on maps. The environment is stored in XML format, so that the individual panorama cubes can be easily extended to include additional data.



Figure 4: 2D connected graph of panorama locations.

## V. DISCUSSION AND FUTURE WORK

During the development of this system, some problems were encountered that will lead to future research. One of these problems is dealing with the overlapping regions of the 6 camera images. Currently, overlapping regions are combined by feathering and alpha blending. However, this can create unwanted ghosting affects in the resulting panorama. More complex stitching algorithms are currently being investigated to solve this problem. Future work will also be performed to automate the process of localizing, aligning and connecting panoramas on a common 2D map. Panorama compression is another issue of interest. Since many high resolution panoramas must be captured for smooth

navigation, a method to optimize disk usage and panorama loading speed will be developed. Currently JPEG compression of the camera images is available directly from the Ladybug camera and hardware compression of the cubic textures is supported using the built-in S3TC format, but further research will explore the optimal compression methods and proper usage of disk cache.

## VI. CONCLUSION

We present a system for virtual navigation in the real world using image-based rendering of panorama images. Our navigation system is based on the graph topology of many cubic panoramas, and we include a simple display interface. Our system can simulate viewing and walking in complex real environments without the need for full 3D modeling, which requires time-consuming steps and advanced resources to accomplish.

## REFERENCES

- [1] S. Baker and S. Nayar. A theory of catadioptric image formation. In IEEE ICCV Conference, 1998.
- [2] A. Basu and D. Southwell. Omni-directional sensors for pipe inspection. In IEEE SMC Conference, pages 3107–3112, Vancouver, Canada, Oct. 1995.
- [3] M. Fiala. Immersive panoramic imagery. In Canadian Conference on Computer and Robot Vision, 2005.
- [4] M. Fiala and A. Basu. Hough transform for feature detection in panoramic images. *Pattern Recognition Letters*, 23(14):1863–1874, 2002.
- [5] NAVIRE. <http://www.site.uottawa.ca/research/viva/projects/ibr/>.
- [6] S. M. Seitz and C. R. Dyer. View morphing. *Computer Graphics*, 30(Annual Conference Series):21–30, 1996.