Panoramic Spherical Video -- The Space Ball

(James) Elliott Coleshill [1], Alexander Ferworn [2]

[1] Department of Computing and Information Science, University of Guelph, Guelph, Ontario, N1G 2W1
[2] The Network-Centric Applied Research Team (N-CART), Ryerson University, 350 Victoria St. Toronto, Ontario, M5B 2K3

Abstract. Techniques for synthesizing panoramic scenes are widespread. Such a scene can be automatically created from multiple displaced images by aligning and overlapping them using an image registration technique. The ability to generate panoramic scenes has many applications including the generation of virtual reality backgrounds, model-based video compression, and object recognition. These techniques--and consequently their associated applications share the restriction that all scenes are limited to a 360 degree view of the horizontal plane at the particular moment in time the images were taken.

Until recently, there has been little motivation to develop techniques for the presentation of complete spherical views in real time---scenes that present the entire potential visible fields of view, through time. With the advent of space exploration and associated micro-gravity environments, “up” and “down” are relative terms and locally fixed points of reference are difficult to come by. It may be useful to rethink how video is captured and presented to a user working in such an environment employing extended notions of what a panorama is.

The Panoramic Spherical Video (PSV) system described in this paper allows a user to view and pan through arbitrary angles of view including elevation and declination as well as providing the view in real time from an array of 16 synchronized CCD video cameras whose video output are selectively “stitched” together to provide a smooth transition between camera fields of view. In this way, the user can smoothly pan through all the fields of view that are generated by the system. All video processing is done in software--there are no moving parts.

Key Words. Panoramic Video, Image Stitching, Robotics, Omni-Directional Camera, Vision Systems

Introduction

Typically, each application relies on an authoring tool capable of registering individual images, aligning them, and finally stitching them to allow a panoramic affect through a 360 degree field of view. We have extended this functionality to support views above and below as well as allowing registration and stitching to occur in real time from video camera inputs. Our goal is to apply these techniques to various space exploration tasks.

V. Kumar et al. (Eds.): ICCSA 2003, LNCS 2667, pp. 51-58, 2003.

In this paper we describe a spherical panoramic video system for supporting micro-gravity applications. The system is a multi-camera network[1]. It will employ several CCD cameras wrapped around a small sphere that allows wide angle imaging with no moving parts. We also describe the software controls and user interface to the camera system.

Motivation--Space Application of PSV

Video cameras have always played a role in space flight and space applications. From the initial construction phase of the International Space Station (ISS) [2], video cameras mounted on the U.S. Space Shuttle [3], and the ISS robotic arm (Canadarm2) [4] were sufficient means to support monitoring the assembly and maintenance activities associated with the project. As the ISS expands with additional construction these cameras will be insufficient for the task. One alternative to equipping the ISS with innumerable inflexible fixed mount cameras is to provide a number of highly flexible panoramic video systems.

The ability to view an environment panoramically is useful in many space related applications such as supporting assembly and maintenance, extra vehicular activities (Spacewalking), video surveillance, and detection and tracking of objects moving within the environment.

During most missions assembly and maintenance operations must occur completely out of the field of vision of the Astronauts onboard, and the lighting conditions can vary substantially. To mitigate these concerns the National Aeronautics and Space Administration (NASA) have designed and deployed the Space Vision System [5][6]. This system is used to support various activities such as, docking, berthing, and mating operations. The Space Vision System consists of video cameras, lighting sources, video distribution units, and a core processing unit.

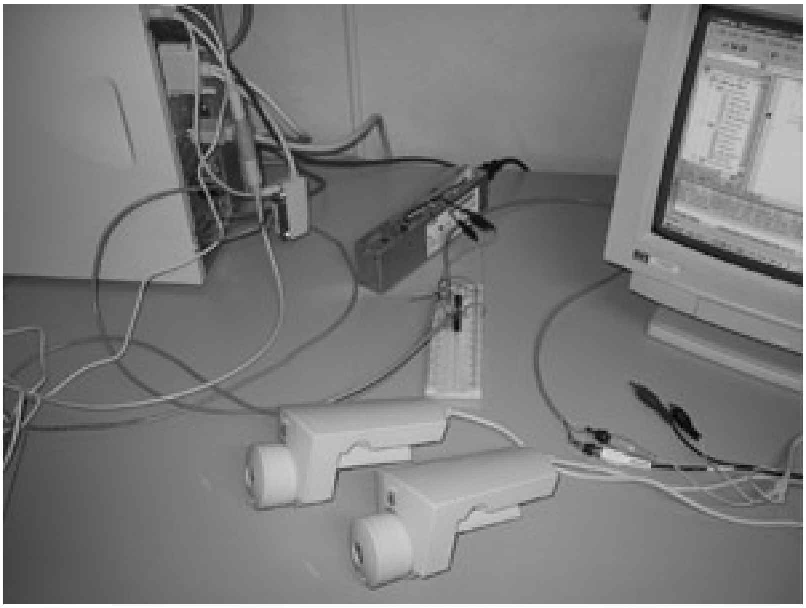
The Space Vision System employs hard mounted video camera devices located around the Space Station and on the Space Shuttle. To continue with support of assembly and maintenance activities a free floating video camera system is required. In December of 1997 NASA test flew the first free-flying robotic camera “ball”. The Autonomous EVA Robotic Camera (AERCam) [7] is used to perform visual and non-visual inspection activities around the outside of the ISS. AERCam is also used as a support tool for spacewalking Astronauts providing views of work being performed.

Unfortunately AERCam has only restricted utility in that its field of view is highly restricted. AERCam has only two color video cameras always pointing to the front of the vehicle. In order to change a view, many adjustments are necessary in the vehicles orientation making it quite cumbersome to operate. In addition, astronauts are provided with an additional source of concern as the device maneuvers in close proximity to their precarious positions.

The PSV will allow for a full environment view of the work area, as well as, reducing the amount of movement required by the free-flyer as scenes are generated in software using our proposed stitching technique. Movement to change the viewing angles will no longer be required.

Hardware Implementation Details

The current PSV system configuration consists of two analog video cameras connected to a multiplexer circuit board, which in turn is connect to a central processing unit.



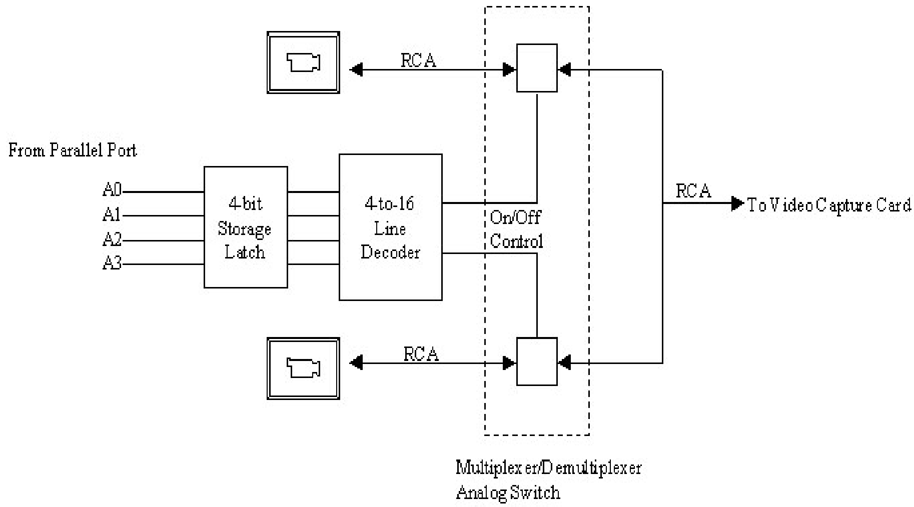
Current Setup

The multiplexer is connected to the computer via the parallel port and an RCA cable to a video capture card. The software controls the multiplexer by sending binary data through the parallel port. The decoder chip reads the binary signal and selects the requested camera by changing the on/off control line in an analog switch. This is shown in the diagram below.

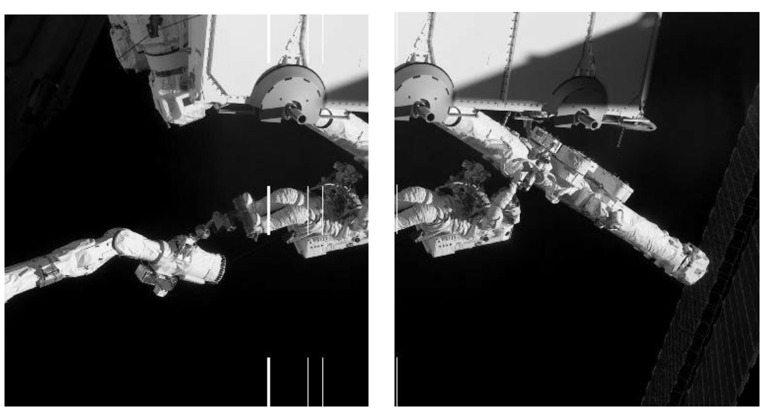
Panoramic Imaging

By combining a simple averaging algorithm as a similarity measure with a step search strategy we are able to scan two images taken by adjacent cameras to find the best possible stitch point.

We begin the image analysis by evaluating the similarity measure for six key points. The similarity measure used is a standard averaging filter. Each of these points consists of 50 pixels, where the average of all the pixels intensity values are calculated. Three points are located on the left side of the right image, and the other three are located on the right side of the left image. To avoid border issues the selected points are located two pixels within the borderline of the each image



Multiplexer Circuit Diagram



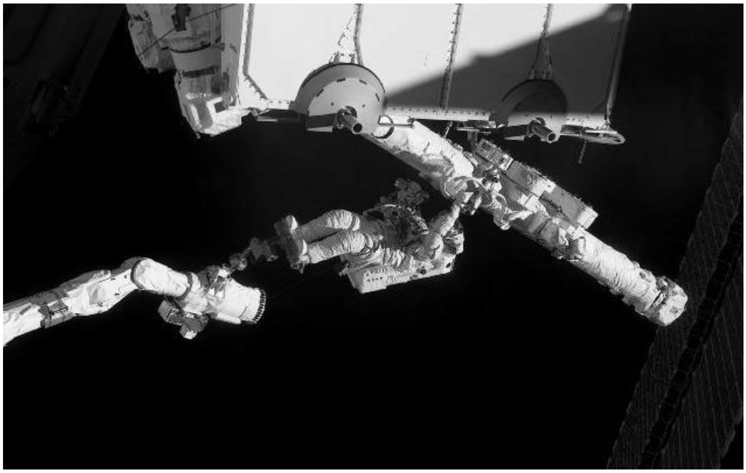
Possible Stitch Points

After obtaining the six points of interest, a scan of the left image is completed using its three points. As the points are moved across the image they are averaged and compared with the points located within the right image. If the averages are within a specified threshold then we consider this area a possible stitch point.

Once the entire left image has been scanned all the possible stitch points found are compared and the area where the majority are located is taken as the best stitch point.

The image merging process is straightforward. It calculates the width of the new image by taking into account the amount of overlap in the starting images. It then starts filling in the pixels of the new image with the corresponding pixels in the original images. The stitch point found determines which image pixels are to be used.

The entire process can be achieved in sufficient time to allow the video sources to be stitched in real time.



Final Stitched Image

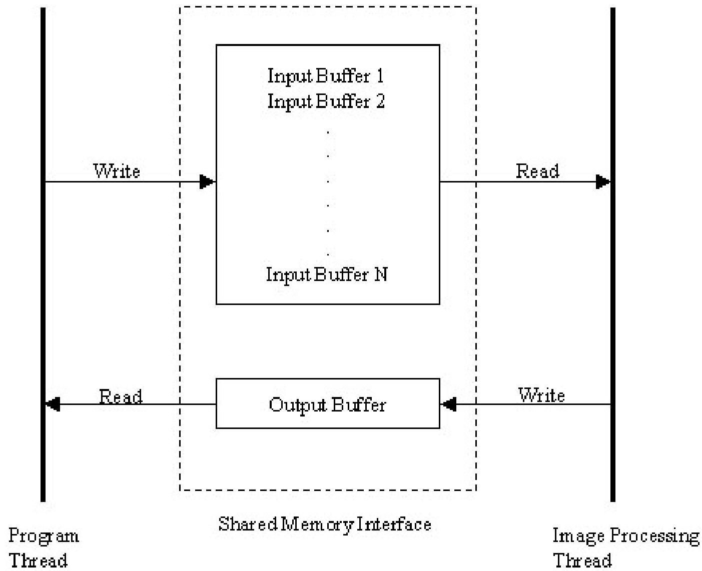
Software Implementation Details

We have written a prototype multi-threaded software system to control the current version of the PSV. The system currently functions in three control modes using two video cameras. Manual Stills, Single Camera Video, and Panoramic Video

- Manual mode allows the user to select an individual camera and take still pictures at the click of a button.
- Single Camera Video mode allows the user to select an individual camera and record an AVI video file from the current input stream.
- Panoramic Video mode allows the system to automatically control the multiplexer and generate two image panoramas.

The software consists of two main threads. The first thread refreshes the user interface, and handles the camera and multiplexer controls. It is responsible for generating an image graph for video streaming, buffering raw data into physical memory, and creating still images to be manipulated. The second thread controls the image processing and merging of the two images to produce the panoramic scenes.

To save time on raw data transfers from one thread to another a shared memory interface (SMI) was developed. The SMI has an input buffer that can store up to 5 seconds worth of raw image data coming from the video capture card, and an output buffer to handle the final panoramic scene to be displayed on the users screen.



Shared Memory Interface Diagram

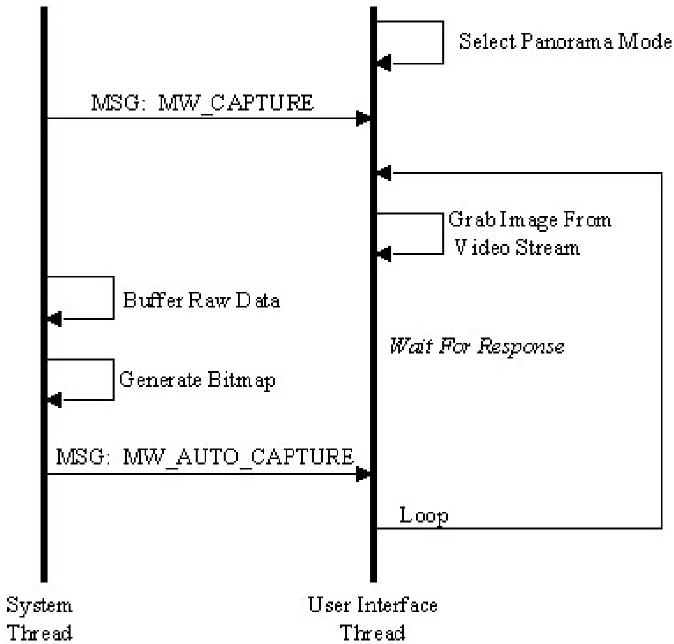
During every second of operation, the main thread buffers one image into the input buffer of the SMI for each connected video camera and refreshes the screen with the contents of the output buffer. At the same time the stitching thread is performing the image processing on the input buffers and writing the final panoramic scene to the output buffer of the SMI.

Since this is a multi-threaded application running in real time some synchronization is required. We used system events to control the processing time of each thread. Once the system is initialized, both threads continue uninterupted monitoring the system for specially defined events to occur. This approach allows the system to constantly buffer raw data from the capture card to the SMI, and refreshes the user interface with the latest generated panoramic scene in the output buffer of the SMI. Below is a timing diagram showing how the system events are used to control synchronization.

Results

As with any real-time application synchronization is a major concern. During initial development of the PSV system we ran into several synchronization and timing issues related to both the hardware and software.

At first we tried to build our multi-camera network using the Universal Serial Bus (USB). This proved to be too slow to handle the real-time control of multiple cameras. We turned to analog cameras and the use of a video capture card similar to that employed in the MAX project [8]. Using this approach and controlling the input signal with a multiplexer circuit proved to work much better. Since the software only has to communicate with the multiplexer and not each individual camera, we were able to reduce a 32-second delay to 600 milliseconds.



System Timing Diagram

We were forced to address various software synchronization issues as well. At first the software was running faster than it could control the hardware. This speed caused the system to buffer 16 of the same image from one camera rather than 16 different pictures from 16 different cameras. To solve this problem special system events were created allowing the software to synchronize with the hardware calls.

We are currently experiencing synchronization problems with the video streams themselves. The video cameras refresh rates have to be synchronized with the changing of the multiplexer circuit. At the moment the 16 images being buffered are a mixture from different cameras. We are currently constructing our own cameras provided with a field sync line. Using this field sync capability will allow us to synchronize the refresh of each camera with the changing of the multiplexer switch.

Conclusion & Future Work

In this paper we have described a robotic video camera system to support micro-gravity applications. The PSV system is used to capture panoramic scenes of a working environment and stream them to a user. This process is accomplished via a multi-threaded software system controlling multiple camera streams through a multiplexer circuit. Still digital images are captured from the raw video streams, stitched together using basic image processing techniques, and displayed to the user's screen.

The next stage of this work will concentrate on refining the design and development of a full sphere of cameras along with various software enhancements. The first step will be to integrate an additional 6 cameras into the system. These cameras will produce half of the spherical design. We intend to enhance our software by implementing a Real-Time "Attention" [9] algorithm to determine "interest" points in the panoramic image. We believe this is a tractable problem in the mostly black expanse of space. We intend the software to be able to monitor a spacewalking Astronaut and help focus the system on activities being performed.

Eventually we hope to deploy an entire 16 camera system with the software enhancements we have discussed. Because the "Ball" will allow "attention" to be focused in software it should be possible create a free-floating version requiring very little physical movement. While we have concentrated on developing the system for space applications we feel that modified versions of the PSV could be applied to areas where motion of the monitoring device is undesirable such as minimally invasive visual diagnostic tools for medical applications.

References

[1] C. Fermuller, Y. Aloimonos, P. Baker, R. Pless, J.Neumann, B. Stuart. Computer Vision Laboratory, University of Maryland. Multi-Camera Networks: Eyes from Eyes. IEEE, 2000

[2] NASA's International Space Station Website, http://spaceflight.nasa.gov/station

[3] NASA's Space Shuttle Website, http://spaceflight.nasa.gov/shuttle

[4] MD Robotics Website, http://www.mdrobotics.ca

[5] S. Miller, D. Withey. Neptec Design Group, Kanata, Canada. System Architecture for the Space Vision System.

[6] Neptec Design Group, Kanata, Canada. A Vision in Space: Astronauts Receive Better Position and Orientation Cues Using the Space Vision System.

[7] Howie Choset, Carnegie Mellon University, David Kortenkamp, Texas Robotics and Automation Center. Path planning and Control for AERCam, a Free-flying Inspection Robot in Space.

[8] A. Ferworn, R. Roque, and I. Vecchia, "MAX: Teleoperated Dog on the World Wide Web", proc. of the 2nd International Workshop on Presence, The University of Essex, Colchester, UK, 6-7 April 1999.

[9] Gal Sela, Martin D. Levine, Center for Intelligent Machines, McGill University. Real-Time Attention for Robotic Vision. Real-Time Imaging 3, PP. 173-194. Academic Press Limited, 1997.