FlyCam: Practical Panoramic Video and Automatic Camera Control

Jonathan Foote and Don Kimber
FX Palo Alto Laboratory, Inc.
3400 Hillview Avenue
Palo Alto, CA 94304
{foote, kimber}@pal.xerox.com

Abstract - **This paper describes computationally and materially inexpensive methods for panoramic video imaging. Digitally combining images from an array of inexpensive video cameras results in a wide-field panoramic camera, from inexpensive off-the-shelfhardware. We present methods that both correct lens distortion and seamlessly merge images into a panoramic video image. Electronically selecting a region of this results in a rapidly steerable "virtual camera." Because the camera is fixed with respect to the background, simple motion analysis can be used to track objects and people of interest. We present methods of motion analysis and algorithms for automatic camera control that mimic the actions of a human operator, using inexpensive and widely available hardware.**

I. Introduction

Teleconferencing and video capture of events such as lectures and meetings require a human operator to orient, zoom, and focus the video or motion picture camera. Automating this task would have many applications in business, education, and entertainment. This paper presents FlyCam, a system that generates a seamless panoramic video images from multiple adjacent cameras. The name alludes to the compound eyes of insects that form sophisticated images from an array of cheap sensors. FlyCam component cameras are mounted on a rigid substrate such that each camera's field of view overlaps that of its neighbor. The resulting images are aligned and corrected using digital warping, and combined to form a large composite image. The result is a seamless high-resolution video image that combines the views of all cameras. Because cameras are mounted in fixed positions relative to each other, the same composition function can be used for all frames. Thus the image composition parameters need only be calculated once, and the actual image composition can be done quickly and efficiently, even at video rates.

Because a FlyCam is fixed with respect to the background, straightforward motion analysis can detect the location of people in the image. This can be used to electronically "pan" and "zoom" a "virtual camera" by cropping and scaling the panoramic image. In this system, an appropriate camera view can be automatically determined by finding motion of human images. Thus the system can serve as an automatic camera operator, by steering a real or virtual camera at the most likely subjects. For example, in a teleconference, the camera can be automatically steered to capture the person speaking. Also, it is possible for remote viewers to control their own virtual cameras; for example, someone interested in a particular feature or image on a projected slide could zoom in on that feature while others see the entire slide.

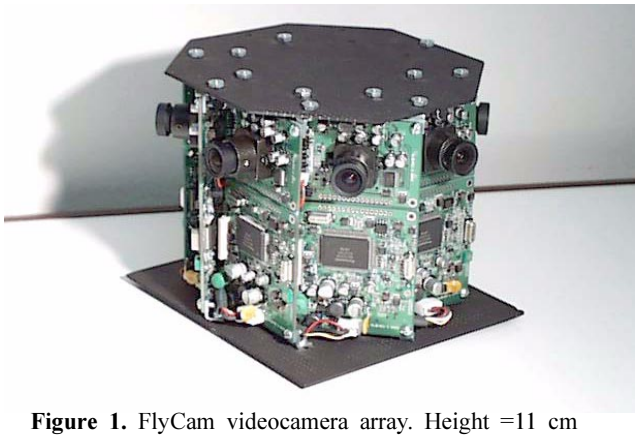

Figure 1. FlyCam videocamera array. Height =11 cm (4.2 inches)

II. Technical Details

A. FlyCam Construction

The philosophy behind FlyCam was to achieve computationally reasonable panoramic imaging with a minimum of expensive or special-purpose equipment. To this end, a FlyCam is composed of inexpensive (< $150) miniature color video board cameras. Figure 1 shows a FlyCam prototype constructed from five video cameras. Though cameras are mounted as close together as practical, they do not share a common center of projection (COP). Cameras are mounted on the faces of an octagon 10 cm wide, thus each camera is angled at 45 degrees to its neighbors. An octagon was chosen primarily for simplicity of construction; many other geometries and configurations are possible and might well improve on the current design. It is not necessary to align or optically calibrate the cameras in any way, as long as their fields of view overlap slightly. Each camera has a 3mm lens offering a near 90-degree field of view, thus the component camera images overlap considerably (for reasons described later). The small focal length results in substantial radial distortion, however it also yields a large depth-of-field and thus all objects are in focus from a distance of a few centimeters to infinity. The resulting FlyCam is lightweight and compact, though there exist smaller board cameras that could be used for an even more compact array.

B. Piecewise image stitching

We use a piecewise perspective warping of quadrilateral regions to both correct for lens distortion and to map images from adjacent cameras onto a common image plane so they can be merged. First, a number of image registration points are determined by

imaging a structured scene and manually identifying the points in the different camera images that correspond to each registration point in the scene. In practice, we image a grid of squares, and use the corners as registration points. The four corners of each square form a quadrilateral "patch" in the image of each camera. Every image patch is then warped back to a square and tiled with its neighbors to form the panoramic image.

Bilinear transformations are used to warp each quadrilateral patch. Each patch is mapped into a square "tile" in the panoramic image. Given that the tiles are square with corners at known coordinates, the equation below transforms the coordinate system *u*, *v* to the warped coordinate system *x*, *y*.

$$\begin{bmatrix} x & y \end{bmatrix} = \begin{bmatrix} uv & u & v & 1 \end{bmatrix} \begin{bmatrix} a_3 & b_3 \\ a_2 & b_2 \\ a_1 & b_1 \\ a_0 & b_0 \end{bmatrix}$$

For each patch, the 8 unknown coefficients of the transformation matrix are determined by solving the simultaneous equations given by the two coordinates of each of the four corners. The four points in each patch have 8 scalar values to solve for the 8 unknown parameters.

To calculate a pixel value in the warped coordinate system *x, y*, the above equation is inverted by solving for *u, v* in terms of *x, y* [1]. This allows for what is termed "inverse mapping." For every pixel in the warped coordinate system, the corresponding pixel in the unwarped system is found and its value is copied. Because the warping is a continuous function rather than discrete, the reverse mapping will generally yield non-integral unwarped coordinates. For this reason, the pixel value is calculated from its immediate neighbors using bilinear interpolation. Because the necessary warping is never extreme for this application, there is no need for additional interpolation or filtering to reduce aliasing effects. Though this is straightforward to implement, we use the Intel Performance Library subroutines as they are tuned for high throughput on MMX processors.

C. Border patch cross-fading

The luminance across cameras will not be even, primarily because the component cameras have "auto-iris" functions that adapt their gain to match the available light. Component cameras imaging a scene with variable lighting will tend to have different gains, hence patches imaged by adjacent cameras will have different luminances. Thus even when the panoramic image is geometrically correct, seams will be apparent from the brightness differences across cameras [2]. We minimize this problem by the simple measure of cross-fading edge patches. Redundant patches at are used at the edge of each camera -- that is, at camera borders, the same patch is imaged from each neighbor camera. Because these patches are then corrected to a square of known geometry, they can be combined by cross-fading them. The pixel value in a patch is given by a linear combination of the component patches, such that pixels on the left come from the left camera, pixels on the right come from the right camera, and pixels in the middle are a linear mixture of the two. This proves quite effective for hiding the camera seams, to the extent that they can be difficult to detect even when the observer knows where to look.

Figure 2. Raw camera images, showing "patches"

D. Finding registration points

The process of finding the registration points is somewhat tedious: the precise coordinates of many points must be recorded across multiple images. On the other hand, this need only be done once. For a registration image, we used a grid of squares placed approximately one meter from the FlyCam. Each square in the grid becomes a warping patch as described above. To facilitate the process, we placed the FlyCam on a slowly rotating platform and recorded the resulting raw video. We recorded grid points from one column of the grid from the first frame, then identified the number of frames that corresponded to angular displacement of one square. Recording grid points from the same column from frames spaced regularly at this interval gave us wide-field registration data without the effort of constructing a wide-field registration image. The final patch grid is shown in Figure 2: not counting overlapping border columns, the grid is 5 X 15 for a total of 75 patches in the final image. This is sufficient to give a good correction for the radial lens distortion; note how distorted vertical features in Figure 2 are straightened in Figure 3.

E. Optical and stereo issues

In keeping with our "better, faster, cheaper" philosophy, no attempt is made to align component cameras to a common center of projection. In any case, it is not practical to achieve a common COP without elaborately aligned mirrors or other optical apparatus. Thus the panoramic image will have imperfections due to disparity between the cameras. We minimize this in several ways. First of all, there is no disparity for objects near the distance of the registration image. Because the baseline distance between component cameras is quite small, an object can move far from the optimal distance without noticeable disparity. Blending the border patches reduces the disparity artifacts even further. For our teleconferencing application, subjects never get close enough to the FlyCam that disparity is noticeable. Distant objects at infinity have a slight disparity (a small number of pixels), but typically

Figure 3. Composite panoramic video frame

these are smooth walls so such artifacts are invisible. Current work is to use the disparity "bug" as a feature: by calculating a measure of disparity it is possible to segment foreground (teleconference participants) from the background [4].

F. Resolution and computation requirements

Our FlyCam prototype operates at 1/4 the potential resolution. Four video signals from the component cameras are combined using a commercially available "quad processor" that tiles each image at 1/4 resolution into a quadrant of the resulting image. This means that only one video card is necessary when the quad processor is used. Also, the raw video can be recorded in one stream without synchronization issues. We have found this useful for developing the tracking algorithms of Section IIIA. Panoramic image generation using 1/4 resolution images and the Intel Performance Library routines takes only 15% of the CPU (of a 300 MHz Pentium II at 15 fps), thus there are plenty of cycles left for the more sophisticated image processing described in the next section. Even using full resolution leaves 40% of the CPU free for other tasks.

III. "VIRTUAL" VIDEO CAMERAS FROM PANORAMIC IMAGES

.An immediate application of the panoramic image is to select a normal-aspect "virtual" view by cropping the large image. Virtual cameras can be panned/zoomed virtually instantaneously, with none of the limitations due to moving a physical camera. In addition, an unlimited number of different views are available at any one time, unlike a physical camera. We have built a FlyCam server application that functions as a virtual webcam, allowing each client to request an individual view from the panoramic image. Figure 4 shows the client; a new virtual camera is steered by clicking on the panoramic image or the left/right arrows, controls, while the"+" / "-" zoom controls have the obvious functionality. Unlike other webcams that use a steerable camera, every client can choose their own unique combination of pan, tilt, and zoom.

A. Virtual camera control using motion analysis

Though a user can easily select a desired image, we hope to eliminate human input entirely for a truly automatic teleconferencing system. To this end, we have implemented a simple automatic camera control algorithm to select an appropriate virtual camera view. Because the FlyCam is fixed with respect to the background, motion analysis does an excellent job of detecting interesting foreground objects, such as people. Motion is determined by computing the absolute value of the frame-to-frame pixel differences. This is thresholded to remove small motions and quantization noise. The total number of non-zero pixels in the thresholded image is a good measure of the motion in the FlyCam's field of view. The first and second spatial moments of the thresholded difference image are calculated. The first spatial moment is a good estimate of the centroid of a moving object, while the second spatial moment estimates the spatial distribution of motion in the panoramic image.

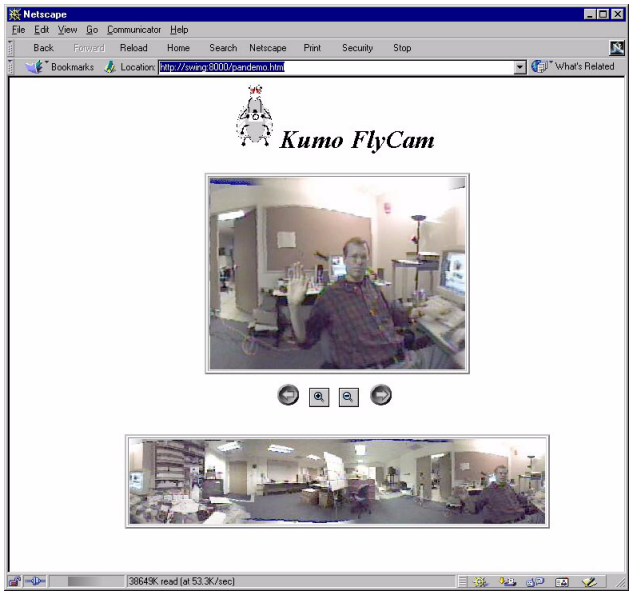

Figure 4. FlyCam webcam application

Straightforward heuristics control the virtual camera based on the motion analysis. The virtual camera is set to follow the first spatial moment, that is the center of any moving object. The new image coordinates x at frame t are set to a function of the coordinates form the previous frame and the new estimate of the motion centroid $\hat{x}$ as follows:

$$x_t = \alpha x_{t-1} + (1-\alpha)\hat{x}$$

The parameter $\alpha, 0 \le \alpha < 1$ serves as "inertia;" if it is large the virtual camera will move only slowly towards the motion. A moderate value of α serves both to mimic the dynamics of a physical camera and to smooth jitters due to noise inherent in the motion analysis. This update equation is subject to several ad-hoc conditions. If the second moment is very large in magnitude, it indicates more than one object is moving so the camera location is not altered. Once the camera has been changed, a timer is started. The.effect of this timer is to discourage long shots of the same scene, which are visually uninteresting. As the timer value increases, the motion change threshold is decreased. This can be done in such a way that the mean or the statistical distribution of shot lengths matches some pre-determined or experimentally determined parameters. Another camera change resets the timer. The net effect is to encourage human-like camera operation. For example, if the virtual camera has been imaging a particular speaker for some time, it becomes more likely that the camera will cut away to capture a listener nodding in agreement. This behavior adds to the realism and interest of the video and mimics

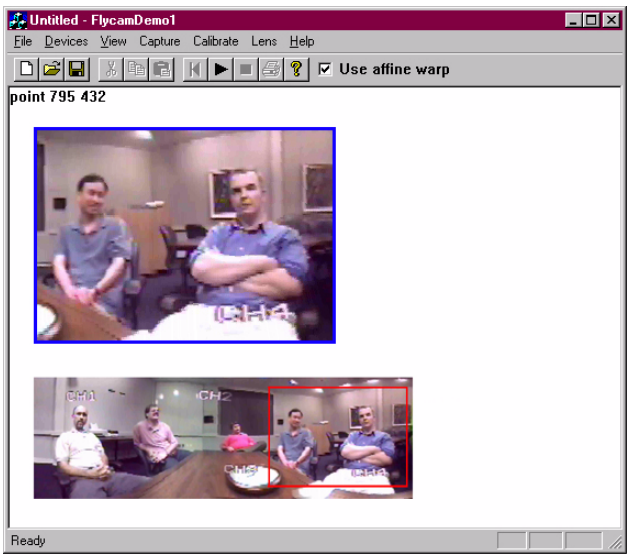

Figure 5. Automatic camera steering from motion.

the performance of a human operator. Figure 5 shows our automatic camera application running on a recorded video stream.

IV. Related Work

There has been considerable prior work on combining multiple images into a panoramic scenes; enough that limited space precludes a fuller set of references. Many approaches have been to compose existing still images into a panorama that can be dynamically viewed [5,6], or by compositing successive video frames into a still panorama [7]. Because all these techniques involve computationally expensive image registration (that is, aligning images with unknown displacements) none of these techniques can be done practically at video rates. In contrast, the system presented here uses cameras with fixed, known alignments, so displacements need not be calculated at all.

A group at Columbia has created an omnidirectional digital camera using curved mirrors [8]. In this system, a conventional camera captures the image reflected from a parabolic mirror, resulting in a hemispherical field of view. Digitally processing the reflected image allows the construction of distortion-free images for any user selected portion of the acquired omnidirectional image, albeit at limited resolution. The drawback of this approach is that subimages extracted from the hemispherical image will be limited in resolution to a small fraction of the single camera, and the necessary image warping will be extreme to regenerate unwarped images. In contrast, the system presented here has virtually unlimited resolution at all viewing angles. If more resolution is desired, the system can be configured with additional cameras. A group at UNC uses 12 video cameras arranged in two hexagons, along with a mirror apparatus to form a common COP. The UNC group devised a similar approach to panoramic image composition, though using the texture mapping hardware of a SGI O2 [2]. Another group at Columbia has taken a similar approach using an array of board cameras. Instead of piecewise image warping, a table lookup system directly warps each image pixel into the composite panorama [3].

There is no shortage of prior research in person tracking systems. Many are commercially available, including ones built into steerable cameras. Systems based on steerable cameras must compensate for camera motion as well as the event when a face goes out of view of the camera. This is much less of a problem for a panoramic camera with a motionless and much wider field of view. At least one system uses a panoramic image from a hemispherical mirror to point a steerable camera [9].

A. Future Work

Besides investigating other camera configurations and resolutions, we are also investigating the use stereo disparity to improve person tracking. We are in the process of integrating a face tracking system developed at Harvard University with FlyCam for more robust automatic camera control. The Harvard system uses a combination of image cues and audio source location from a microphone array to steer a conventional camera [10].

References

[1] G. Wolberg, *Digital Image Warping*, IEEE Computer Society, Press, 1992

[2] A. Majumder, et al., "Immersive teleconferencing: a new algorithm to generate seamless panoramic video imagery," in *Proc. ACM Multimedia 99*, Orlando, FL, pp. 169-178, 1999.

[3] R. Swaminathan and S. Nayar, "Non-metric calibration of wide-angle lenses and polycameras," in *Proc. Computer Vision and Pattern Recognition*, June 1999

[4] Darrell, T., Gordon, G., Woodfill, W., Baker, H., "A magic morphin mirror," in *SIGGRAPH '97 Visual Proceedings*, ACM Press. 1997.

[5] S. Chen and L. Williams, "View interpolation for image synthesis," in *Computer Graphics (SIGGRAPH'93)*, pp.279-288, August 1993.

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

[7] Teodosio, L., and Bender, W., "Salient Video Stills: content and context preserved," in *Proc. ACM Multimedia 93*, Anaheim, CA, pp.39-46, 1993.

[8] Nayar, S., "Catadioptric omnidirectional camera." In Proc. of *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Peurto Rico, June 1997

[9] Huang, Q., Cui., Y., and Samarasekera, S., "Content based active video data acquisition via automated cameramen," in *Proc. IEEE International Conference on Image Processing (ICIP) '98*

[10] Wang, C., and Brandstein, M., "A hybrid real-time face tracking system," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) '98,* IEEE