Panoramic Video Capturing and Compressed Domain Virtual Camera Control

Xinding Sun*, Jonathan Foote+, Don Kimber+, B. S. Manjunath*

*Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106
{xdsun, manj}@.ece.ucsb.edu

+FX Palo Alto Laboratory, Inc. 3400 Hillview Avenue Palo Alto, CA 94304
{foote, kimber}@pal.xerox.com

ABSTRACT

A system for capturing panoramic video and a novel method for corresponding compressed domain virtual camera control is presented. It targets applications such as classroom lectures and video conferencing. The proposed method is based on the FlyCam panoramic video system that is designed to produce high resolution and wide-angle video sequences by stitching the video pictures from multiple stationary cameras. The panoramic video sequence is compressed into an MPEG-2 stream for delivery. The proposed method integrates region of interest (ROI) detection, tracking, and virtual camera control, and works on compressed domain information only. It first detects the ROI in the P (predictive coded) picture using only the macroblock type information. It then up-samples this detection result to obtain the ROI of the whole video stream. The ROI is tracked using a Kalman filter. The Kalman filter estimation results are used for virtual camera control that simulates human controlled video recording. The system has no physical camera motion and the virtual camera parameters are readily available for video indexing. The proposed system has been implemented for real time processing.

Keywords

Panoramic Video, Region of Interest, Virtual Camera Control, Tracking, Kalman Filtering, IIR Filter, MPEG-2.

1. INTRODUCTION

A typical scenario that this paper is concerned with is that of a speaker giving a lecture in a classroom/seminar or teleconference. The speaker may move around, stop, turn his body, or perform some gestures. One would like to obtain the ROI video sequence in the scene that includes the speaker.

The first problem considered here is the design of a system to capture the events. It is natural to use a panoramic camera system to capture the whole scene. Processing the panoramic video will obtain the ROI. The advantage of a panoramic system is that the speaker is always in the scene. This is the basis of the robustness of the system. In this paper, the FlyCam [6] system is used to produce panoramic video. Figure 1.a shows the frame from a panoramic video, and its corresponding ROI is shown in figure 1.b.

The second problem emerges when the panoramic video is compressed and stored or delivered to a client, where there is a need for virtual camera control. Here the MPEG-2 video compression format is considered. While the ROI can be processed on the raw panoramic video frames, the objective is to provide a fast and robust solution in the compressed domain.

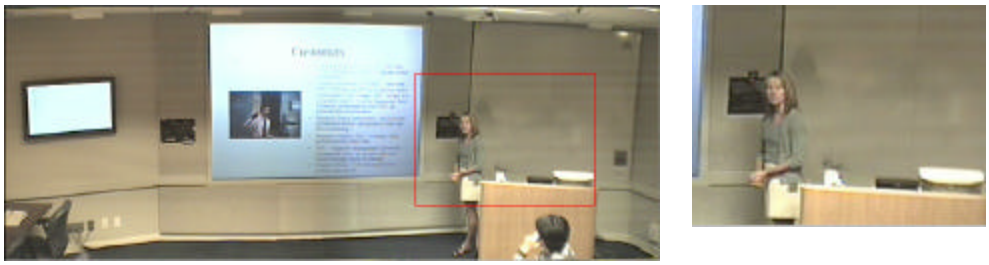
A novel method is proposed to process the panoramic video in the compressed domain. The method integrates ROI detection, tracking, and virtual camera control. First, it detects the ROI based on P picture macroblock information. Then, it up-samples the detection results to obtain the ROI of the whole video stream. The ROI is then tracked using a Kalman filter. The Kalman filter output is used to steer a "virtual camera" for displaying or recording ROI video. The Kalman filter output is smoothed to simulate the response of a human camera operator (this will be discussed in later sections). Since the panoramic camera is statically mounted, no physical camera control is needed.

The paper is organized as follows: Section 2 discusses the related work. To motivate the camera control discussion, section 3 summarizes the FlyCam system first presented in [6]. Section 4 details our main contribution--the compressed domain virtual camera control. Experiments and conclusions are given in section 5 and section 6, respectively.

2. RELATED WORK

There are some commercial as well as research systems that attempt to provide a solution for capturing events. Sony's EVI-

.



(a) Panoramic scene view (b) ROI output

Figure 1. An example of a panoramic scene and its ROI.

D30 camera [13] can be used to track moving objects, but it is often not robust. In particular, steerable cameras suffer from the drawback that the objects are difficult to track once they leave the camera's field of view.. Campbell and Bobick [3] use tokens to track object for event analysis. The token detection is more robust, but it also suffers from the same drawback. For example, when the subject turns his back and the token is out of the camera' filed of view, it will fail to track the subject.

Systems that stitch multi-camera video sequences have been designed to capture events. The advantage of this kind of panoramic systems is that the speaker never leaves the camera's field of view. Chen and Williams [4] and many others have developed systems that compose existing still images into a panorama that can be dynamically viewed. Teodosio and Bender [17] have developed a system that composites successive video frames into a still panorama. The above mentioned methods require image registration which is computationally expensive and thus limit their application to real-time video. Nayar [10] has created an omnidirectional digital camera using curved mirrors. A conventional camera captures the image from a parabolic mirror, resulting in a hemispherical field of view. But the subimages extracted from the hemispherical image will be limited in resolution to a small fraction of the single camera. Majumder et al. [8] use 12 video cameras arranged in two hexagons, along with a mirror apparatus to form a common center of projection (COP). They devised a similar approach to panoramic image composition using texture-mapping hardware. Swaminathan and Nayar [16] have 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. In a more recent work by Nicolescu and Medioni [1], a camera array is used for panoramic image composition. It can provide depth information. Foote and Kimber [6] have developed a system that is fast and targets minimum cost and general-purpose hardware.

Previous person-tracking efforts date back to the early 1980s. An example is O'Rourke and Badler's [12] work on 2-D kinematic modeling. In a more recent work, Darrell et al. [5] integrate stereo, color, and face detection with person tracking. Wang and Chang [19] have developed a system that can detect a face in an MPEG video based on DCT coefficients. The decompression process is avoided. Their detection rates reach 90 percent.

Since the main objective of the above systems is tracking or detection, the output of these systems is usually an object outline. Using raw tracking results to steer ROI selection usually produces objectionable jitter in the video output.

The method proposed here uses P picture macroblock type information for detection purposes. It requires even less computation to extract compressed information than that of [19] as it does not even require the decoding of DCT coefficients. It integrates detection, tracking and virtual camera control together to simulate the response of a human operator.

3. FLYCAM PANORAMIC VIDEO SYSTEM

The philosophy behind the FlyCam panoramic camera system design is to achieve computationally reasonable panoramic imaging with a minimal amount special-purpose equipment. The FlyCam system is composed of inexpensive miniature color video board cameras. The system generates panoramic video from multiple adjacent cameras in real time. Lens distortions are corrected and the images are stitched seamlessly by digital warping.

3.1 Hardware

Figure 3 shows pictures of two FlyCam prototypes constructed from multiple video cameras. Though cameras are mounted as close together as is practical, they do not share a common center of projection. Thus, it is not necessary to align or optically calibrate the cameras in any way as long as their fields of view overlap slightly. The resulting FlyCam is lightweight and compact, though there exist smaller board cameras that could be used for an even more compact array. Depending on the application, different numbers of cameras can be used to capture different kinds of scenes. The 180° panoramic view captured in figure 1.a was produced by using the system shown in figure 3.b.

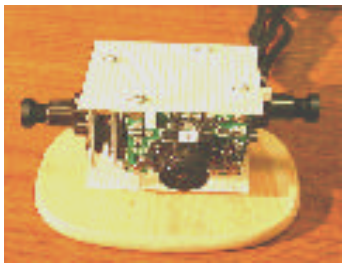
3.2 Piecewise Image Stitching

Piecewise perspective warping of quadrilateral regions is used to correct for lens distortion and to map images from adjacent cameras onto a common image plane so that they can be merged.

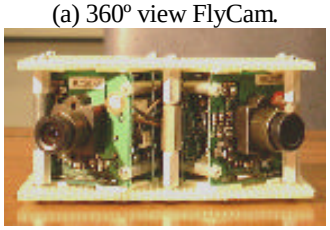
(a) Raw camera images showing patches.

(b) Composite panoramic video frame.

Figure 2. Raw camera images and composite panoramic video frame.



(a) 360º view FlyCam.



(b) 180º view FlyCam.

Figure 3. FlyCam examples.

First, a number of image registration points are determined by imaging a structured scene. The points in the different camera images that correspond to each registration point in the scene are manually identified. In practice, the corners of a grid of squares are used 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 then mapped into a square "tile" in the panoramic image. Given that the tiles are square, with corners at known coordinates, the transformation from (u,v) coordinate system to the warped coordinate system (x,y) is given by:

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

There are 8 unknown coefficients. Each point provides two constraints. Thus the four corners of a tile give eight equations to solve for the eight unknowns.

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) [20]. This allows for what is termed as "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.

3.3 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 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. Cross-fading or blending of edge patches minimizes this problem. Redundant patches are used at the edge of each camera: that is, at camera borders the same patch is imaged from each neighboring camera. Because these patches are then corrected to a square of known geometry, cross-fading can then be used to combine them. The pixel value in a patch is given by a linear combination of the component patches, such that in the panoramic image, pixels on the left come from the left camera, pixels on the right in the panoramic image come from the right camera, and pixels in the middle are a linear mixture of the two. This proves quite effective for hiding the cameras' seams, to the extent where they can be difficult to detect even when the observer knows where to look.

3.4 Optical and Stereo Issues

In keeping with the "better, faster, cheaper" philosophy, no attempt is made to align the 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. The panoramic image will have imperfections due to disparity between the cameras. Blending the border patches reduces the presence of disparity artifacts. For the classroom or teleconference applications presented here, subjects never get close enough to the FlyCam that disparity is noticeable.

4. VIRTUAL CAMERA CONTROL IN THE COMPRESSED DOMAIN

In some applications, the panoramic video has to be stored as digital video data. In other applications, it has to be delivered to a client, and thus, the virtual camera control has to be performed on the client-side. These videos are usually available in compressed video streams. A straightforward solution to this problem is to decompose the video and process it in the uncompressed domain, but it is not efficient. In the following, an efficient virtual camera control method is introduced.

4.1 General System Structure

A schematic of the virtual camera control process in the compressed domain is shown in figure 4. The input is a panoramic video in MPEG-2 format. First, the ROI is detected using the P frames in the MPEG-2 stream. This includes picking up the P frame from the video, detecting the ROI at a P frame, and then propagating the results to neighboring frames by up-sampling. Next, the output of the ROI detection is fed into a Kalman filter. The Kalman filter estimates the speed and position of the speaker. These estimated parameters are used for virtual camera control. The ROI output can be used to display the video (see section 5). It can also be used to extract the ROI video from the original panoramic video for storage purposes.

4.2 Detection of the ROI in the Compressed Domain

Compressed domain video processing can achieve fast speeds. While Zhang et al. [21] and many others use compression domain information as features for video data segmentation and indexing, very few efforts have been made to use them for detection

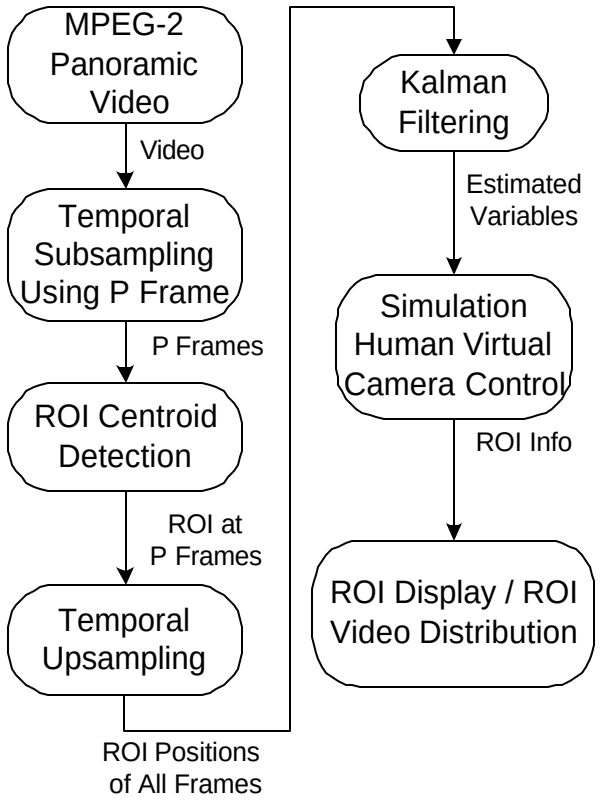


Figure 4. A schematic of the virtual camera control in the compressed domain.

purposes. An example of compressed domain face detection is given in [19]. The method proposed here is based on our previous work on motion activity detection [15].

4.2.1 MPEG-2 Motion Compensation

4.2.1.1 MPEG-2 Frame type

At this point, it is helpful to briefly review the MPEG-2 motion compensation scheme [7]. Motion compensation helps reduc temporal redundancy between the frames in a video sequence. There are three types of frames in the MPEG-2 video stream. They are the I picture, which is intra-coded, the P picture, which is coded using forward prediction, and the B picture, which is coded using both forward and backward prediction. The

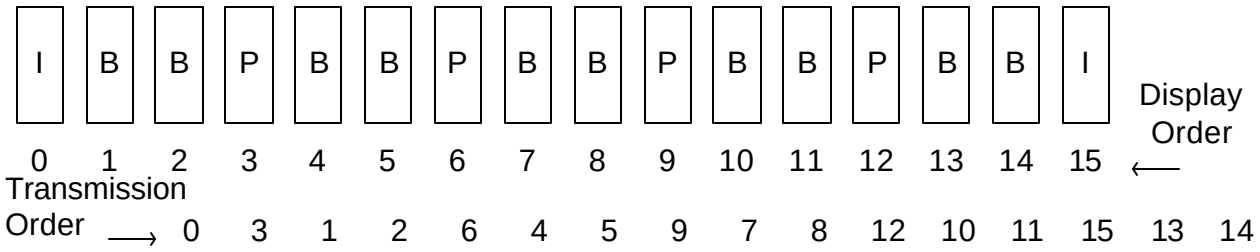


Figure 5. An example of MPEG group of pictures (GOP).

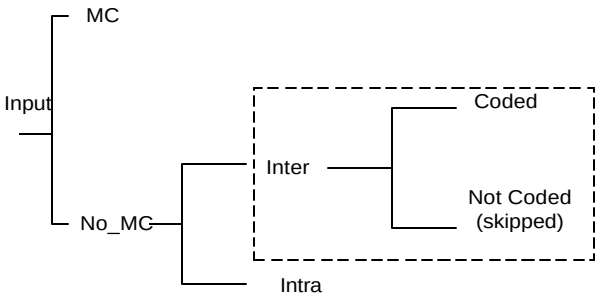
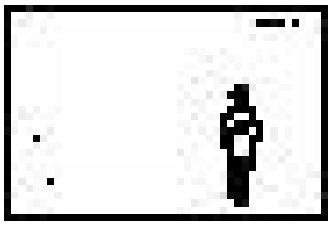


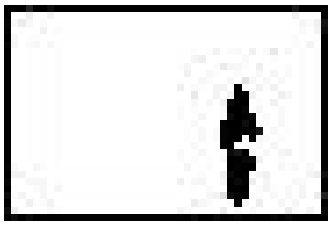
Figure 6. Coded macroblock types in MPEG-2 video P picture.

organization of these three types of frames is not normalized in the MPEG-2 standard and the choice is left to the encoder. Figure 5 shows a typical MPEG-2 sequence, which is also a group of pictures (GOP). Note that the display and transmission have different orders. For simplicity, only progressive video is considered for discussion, however, the proposed method also works on interlaced video if only one field of a frame is taken for processing.

(a) A P picture from an MPEG-2 video.



(b) Foreground detection of the object in the compressed domain.



(b) Foreground detection after median filtering.

Figure 7. Foreground detection for a P picture in an MPEG-2 Video.

4.2.1.2 No_MC Inter-Coded Macroblock Types

To exploit temporal redundancy, MPEG-2 adopts macroblock level motion estimation. In order to reduce the bit rate, some macroblocks in the P or B pictures are coded using their differences with corresponding reference macroblocks. Since only the P picture is used for later discussion of camera control, the following discussion applies to the P picture only. During motion compensation estimation, the encoder first searches for the best match of a macroblock in its neighborhood in the reference frame. If the prediction macroblock and the reference macroblock are not in the same positions of the frames, motion compensation is applied before coding. No_MC means non motion compensation. When a macroblock has non motion compensation, it is referred to as a No_MC macroblock. Generally, there are two kinds of No_MCs: one is the No_MC intra-coded and the other is the No_MC inter-coded. In the typical MPEG-2 encoder architecture, there exists an inter/intra classifier. The inter/intra classifier then compares the prediction error with the input picture elements (pels). If the mean squared error of the prediction exceeds the mean squared pel value then the macroblock is intra-coded, otherwise it is inter-coded. The No_MC intra-coded and inter-coded scheme can be obtained correspondingly. Complicated schema can be used to determine which way to code a macroblock. In general, the condition for a No_MC inter-coding can be written as:

$$\sum_{\mathbf{x}} \left(I_c(\mathbf{x}) - I_r(\mathbf{x}) \right)^2 < s \qquad (2)$$

Where $\mathbf{x} = (x,y)^t$ is the position of a point in the macroblock, s is the threshold, I_c is the current frame with a P picture type, and I_r is the reference frame which is either an I picture or a P picture.

Only MPEG-2 macroblocks of P pictures have No_MC inter-coded macroblocks. In fact, in a special case, when the macroblock perfectly matches its reference, it is skipped and not coded at all. To make illustration easy, the skipped frame is categorized the same as a No_MC inter-coded frame as shown in figure 6.

4.2.2 Detection of the ROI Centroid in P Pictures

Many methods have been proposed in the literature for object tracking. Since the primary objective is to capture a single speaker in a panorama, complex models such as those used in [5] and [12] are not needed. The speaker is modeled as a point object corresponding to the centroid of the body. The ROI output is a predetermined rectangular region that surrounds this point. Thus, the ROI detection basically detects the centroid of the body that is in the foreground of the scene.

4.2.2.1 Spatial Sub-sampling and Up-sampling of P Pictures

The MPEG-2 motion compensation scheme borrows its strategy from traditional region-based optic flow estimation, even though the motion vectors it provides are essentially not the same as

optic flow. In the case of a video where there is only one moving object, this motion compensation information becomes especially important.

If the center of a macroblock is to represent the whole block, then a sub-sampled image of the original frame can be obtained. Since the macroblock size is 16x16, the height and width of the sub-sampled image are 1/16th of the original frame height and width respectively. If an estimation of the centroid of the ROI in the sub-sampled image is obtained, it can then be up-sampled, i.e. the estimated centroid position (x, y) can be scaled by 16 to get the estimation of the original frame.

Sub-sampling tends to create aliasing effects if there are high frequency signals in the original image. That is why traditional motion estimation methods usually filter the images with a Gaussian filter before sub-sampling. An example of an application of a Gaussian filter to image can be found in Burt and Adelson [2].

4.2.2.2 Detection of the ROI Centroid

Given a point and its neighborhood, (here a point refers to the centroid of a macroblock and the neighborhood refers to the whole macroblock), the motion estimation can be formulated as minimizing the following summed square difference based on the intensity constancy [1]:

$$\sum_{\mathbf{x}} \left(I_c(\mathbf{x}) - I_r(\mathbf{x} - \mathbf{V}(\mathbf{x})) \right)^2 \qquad (3)$$

If the motion is zero, i.e. $\mathbf{V}(\mathbf{x}) = \mathbf{0}$, then $\sum_{\mathbf{x}} \left(I_c(\mathbf{x}) - I_r(\mathbf{x}) \right)^2$ should reach the minimum. A simple comparison shows that it achieves the same objective as (1). Therefore No_MC inter-coded macroblocks can be used to represent the background of the scene, which has no motion.

In a scene captured in panoramic video, the region where there is motion is the ROI. In the compressed domain processing, the background is first detected using the macroblock motion information. The ROI is detected by taking the complement of the background region.

Figure 7.a shows an example of a P picture in an MPEG-2 video. Figure 7.b shows the foreground detection results based on No_MC coding information, where the white macroblock is the background, and the black macroblock is the foreground. The coordinates of the detected centroid of the ROI can be found in figure 8.a. Since the body region is connected, a median filter is used to improve the detection result. Figure 7.c shows the result after median filtering of figure 7.b. The median filter is of size 3x3. In practice, the upper body is more important than the lower body as ROI output. Therefore, if the ROI size is small, the detected ROI can be shifted upward to center it on the upper body.

After the foreground object is detected, the centroid of the object can be computed easily in the sub-sampled image domain. The computed result is then scaled by 16 times to get the estimation of the ROI centroid position in the original video frame.

Frame	Frame Information
	(a) P Picture Frame Size: 720x480 ROI Size: 200x200 ROI Centroid (x,y) : (530.5, 279.5)
	(b) B Picture The ROI centroid of (a) is applied here.
	(c) B Picture The ROI centroid of (a) is applied here.
	(d) I Picture The ROI centroid of (a) is applied here.

Figure 8. Four consecutive frames in different frame types in an MPEG-2 video.

The frame of (a) is the same as the frame of figure 7.a.

4.2.2.3 Temporal Sub-sampling and Up-sampling Using the P Picture

In a typical video considered the 30 frames per second frame rate is higher than is necessary for processing. Figure 8.a-d show four consecutive frames in an MPEG-2 video in a seminar room setting. Note that the frame-to-frame motion of the speaker is quite small. If only the centroid of the ROI, which covers the speaker, is considered, it moves only several pixels in each direction. Therefore, even if the ROI centroid of figure 8.a is applied to the following frames in figures 8.b to 8.d, there is no noticeable difference (note that the original ROI is shifted 40 pixels upward in the picture to center it on the upper body).

Therefore, it is reasonable to use the P picture to sub-sample the video sequence first. After the ROI is detected in the P pictures, it is then up-sampled to obtain ROI positions of neighboring I and B pictures in the original video sequence. As shown in figure 5, the P pictures, numbered 3, 6, 9, and 12, sub-sample the displayed video sequence. Depending on the organization of the frame types, the distance between two P pictures varies. However, it was found in the experiments that sub-sampling using P pictures is very effective in video sequences coded as in figure 5.

To reduce the high frequency noise introduced by the up-sampling process, the positions of the ROI in the I and B pictures can be interpolated using the two P pictures that bound them. But in this case, the ROI position of the latter P picture has to be detected first, before the interpolation. Note that the sub-sampling is not uniform, since the P picture distances are not the same.

4.3 Tracking using a Kalman Filter

The detection of the ROI centroid coordinates is, in general, a noisy process. The noise may come from sub-sampling, lighting change, etc. If the noise is assumed to be Gaussian, then it can be handled by using an extended Kalman filter. The centroid has a trace in 2D space. The trace in the x- direction can be modeled by the second-order Taylor series expansion of the form:

$$x(k+1) = x(k) + v_x(k) + a_x(k)T^2/2 + h.o.t. \quad (4)$$

$$v_x(k) = a_x(k)T + h.o.t. \quad (5)$$

Where $x(k)$ is the centroid coordinate in the x- direction, $v_x(k)$ is the corresponding velocity, $a_x(k)$ is the corresponding acceleration, and T is the time interval. Similarly, the same model applies to the trace in the y- direction. Combining the models in two directions gives the centroid system model:

$$\mathbf{F}(k+1) = \Phi\mathbf{F}(k) + \Gamma\mathbf{w}(k) \quad (6)$$

Where $\mathbf{F}(k) = \left[x(k), y(k), v_x(k), v_y(k)\right]^t$, and $y(k)$ is the centroid coordinate in the y- direction, and $v_y(k)$ is the corresponding velocity, while $\mathbf{w}(k)$ is the system Gaussian noise, representing the acceleration of the centroid in the x- and y- directions, and

$$\Phi = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix}$$

Higher order Taylor series expansions can be applied to the centroid system model, which would lead to higher orders of the system model. However, in the whole detection process, high order terms tend to be noise in the system. Therefore a low order system model is generally more robust. Experiments have shown that this model works well. In addition, as discussed in the following section, the system variables provide enough information for virtual camera control.

Since the speaker is modeled as a simple point, the measurement of it can be modeled as:

$$\mathbf{Z}(k) = H\mathbf{F}(k) + \mathbf{n}(k) \quad (7)$$

Where $\mathbf{Z}(k)$ is the measurement, $\mathbf{n}(k)$ is the Gaussian measurement noise, and H is the measurement transfer function, in this case a scaling factor.

The covariance form of Kalman filtering is used to recursively update the prediction based on the innovation information at each step. The prediction at each update is output for further ROI virtual control purposes. The predicted or estimated variable used to control the recording process is $\hat{\mathbf{F}}(k) = [\hat{x}(k), \hat{y}(k), \hat{v}_x(k), \hat{v}_y(k)]^t$.

4.4 Virtual Camera Control

Kalman filtering reduces most of the noises inherent in the tracking estimate, and suffices for most purposes. However, if the tracking result is used to move the ROI window directly, the quality of the output video is often jittery. The resulting motion is less smooth than that of a physical camera, which has inertia. Therefore, an additional filtering step is taken to produce smooth and pleasant ROI video output.

The method proposed here for virtual camera control is based on the following observation. When an experienced camera operator records a lecture, if the speaker is motionless or moving only within a small region, the operator usually does not move the camera (stabilization control). When the speaker changes his position by a large distance, the operator must move the camera to catch up with the speaker (transition control). After the speaker has been centered, the operator follows further movement (following control). Accordingly, the virtual camera control operates in three similar regimes.

Stabilization control is based on the Kalman filter estimates of position and velocity. The initial centroid position is registered first, denoted as $\mathbf{Y}_R(k) = [x_R(k), x_R(k)]^t$, where $x_R(k), y_R(k)$ correspond to its coordinates in the x- direction and y- direction respectively. Then at each frame, the estimated speed and position are checked. They can be obtained from $\hat{\mathbf{F}}(k)$ during the Kalman filter update process. If the following two conditions are satisfied, the virtual camera is fixed and the registered position is used as a position output. Firstly, the new position must be within a specified distance of the registered position in a given direction. Secondly, the estimated speed must be below a specified threshold at a given direction. Otherwise, the virtual camera control is changed to the "transition" regime. The stabilization control conditions can be formalized as:

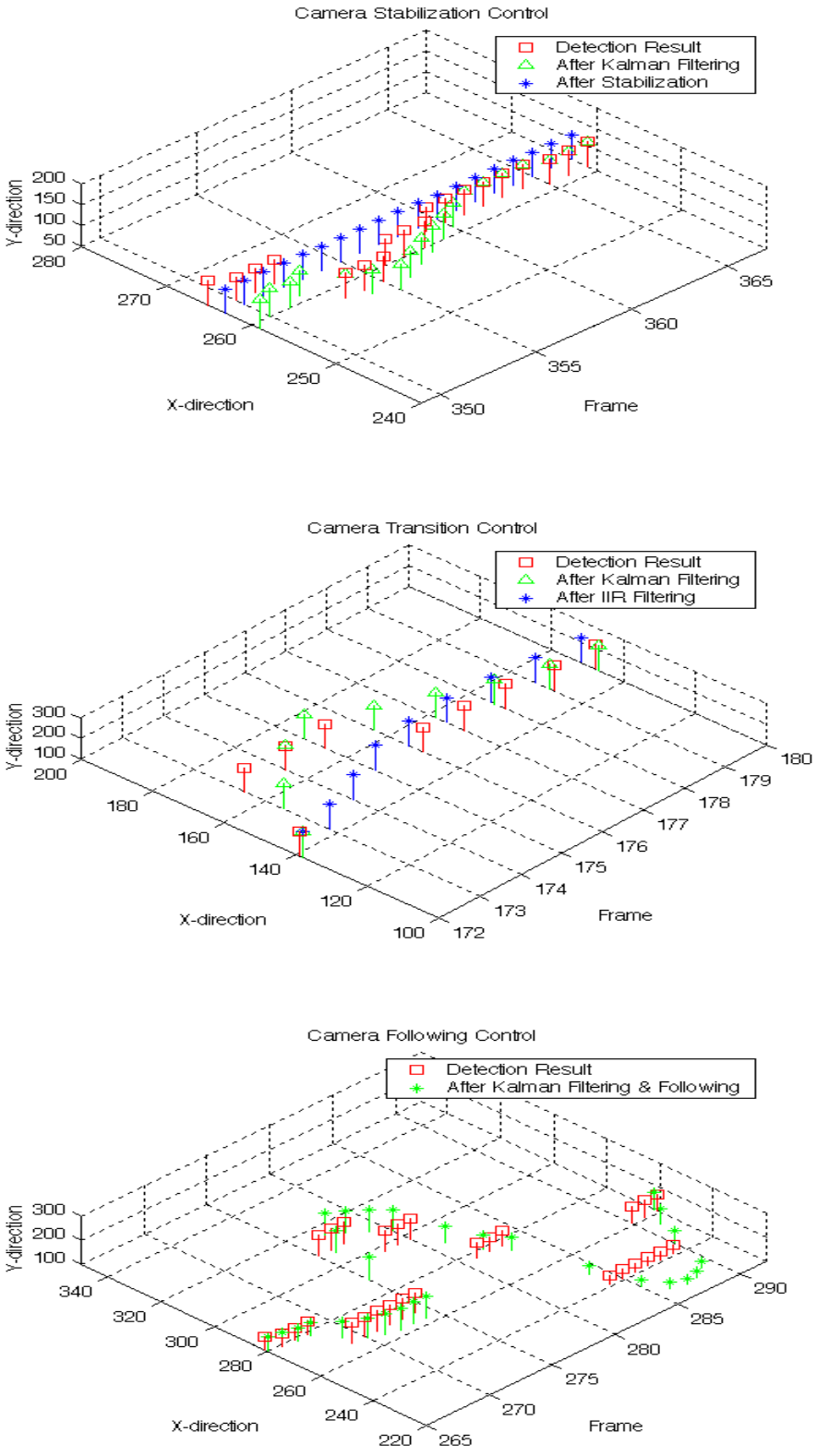


Figure 9. Simulation of three types of camera control.

$$\mathbf{Y}(k) = \mathbf{Y}_R(k) \tag{8}$$

if $|\hat{x}(k) - x_R(k)| < s_1,\ |\hat{y}(k) - y_R(k)| < s_2$

and $|\hat{v}_x(k)| < s_3,\ |\hat{v}_y(k)| < s_4$

Where s_1, s_2, s_3, *and* s_4 are thresholds, and $\mathbf{Y}(k)$ is the ROI output.

In the transition regime, a lowpass filter is used to update the virtual camera location. For this purpose, a first order lowpass IIR filter is used:

$$\mathbf{Y}(k+1) = a_1\mathbf{Y}(k) + a_2\hat{\mathbf{X}}(k) \tag{9}$$

Where $a_1 + a_2 = 1, a_1, a_2 > 0$, and $\hat{\mathbf{X}}(k) = [\hat{x}(k), \hat{y}(k)]^T$ is the estimated centroid from the Kalman filter, and serves as the input to the IIR filter. The virtual camera now follows $\mathbf{Y}(k)$, which is smoother than the Kalman filter output. Experiments show that values of $a_1 = 0.8, a_2 = 0.2$ give a reasonable simulation of human camera operation.

Since the IIR filter (9) tends to create delays in the output, the number of steps of in the virtual camera transition is limited. After a certain time in the transition regime, for example 0.5 seconds, the camera control is switched to the "following" regime. Updating the ROI position directly from the Kalman filter output realizes this objective:

$$\mathbf{Y}(k) = \hat{\mathbf{X}}(k) \tag{10}$$

Note that this is equal to setting $a_1 = 0, a_2 = 1$, in the IIR filter (9). Figure 9 shows the results for three kinds of camera control.

The Kalman filter assumes environmental noise is Gaussian. It can handle lighting change, occlusion very well. But there are some other noises which are not Gaussian. For example, the projection display and the audience both can produce constant noise in fixed regions in the background, as can be seen in figure 1. This knowledge can be incorporated into the tracking system to improve performance, especially as the panoramic video cameras are fixed with respect to the background. Configuration parameters allow some part of these regions to be ignored. By offering this kind of flexibility, the tracking technology can be easily adapted to different environments.

5. EXPERIMENTAL RESULTS

Seminar room panoramic video is used for experiments. The speaker moves in the front of the classroom during a lecture. Panoramic video is recorded and encoded using an MPEG-2 encoder. The video size is 720x480 and the frame rate is 29.97 per second. To ensure the video frame rate, the stitching process can be done offline. The size of video can be changed to fit different applications.

Figure 10. The interface of the MPEG-2 player.

A video player is developed based on the MPEG-2 player distributed by the MPEG Software Simulation Group [9]. It has been implemented on a SGI ORIGIN 200 machine. While decoding the video stream, the player also goes through the ROI detection, Kalman tracking, and virtual camera control processes. Only ROIs of size 200x200 are displayed on the screen in order to check the result. Since the upper body is more important in viewing, the ROI is shifted 40 pixels upward when playing. Five video sequences lasting around 30 minutes each are used for the experiments. Experiments show that after several frames of initialization, once the speaker starts to move, the program controls the virtual camera to follow him. After this initialization, the speaker is always included in the ROI. The ROI video output is also smooth and is similar to a human controlled video recording. The whole process is done at frame rate. Note that not all macroblocks in a frame need to be decoded. The number of such macroblocks decoded depends on the position of the ROI. The control parameters computed in the compressed domain can also be used in other applications such as for indexing object trajectories. Because the size of the P picture macroblock information is insignificant compared to the entire video stream, the delay created by virtual camera control sub-processing cannot be noticed while it is playing. Figure 10 shows an interface to the player. The image shown corresponds to the ROI.

6. CONCLUSIONS

In this paper a new method is presented for recording the region of interest in a scene. The FlyCam panoramic camera system is used to capture the scene. After the video is compressed, the proposed method integrates detection, tracking and recording processes, and simulates human camera control. This processing is done in the compressed domain. The entire process is fully automated and experiments show that it is robust and fast enough for real time applications.

Provided there is only one speaker in the scene, this method can be applied to a panoramic view of up to 360º using the system shown in figure 3.a. Note that, even though the FlyCam system is used here for panoramic view capturing, it can be applied to panaromic videos produced by other systems as well. Even

though the MPEG-2 format is discussed here, the method can be applied to other formats such as H.263 which have similar forward motion compensation schemes.

For typical lectures, the speaker remains at roughly the same distance from the camera, thus zooming is not necessary. However, zooming could be achieved by scaling the ROI for applications that need it, as discussed in [14] and [18]

Since in the system the cameras are stationary, the tracking information also provides a feature description of the video content. This feature information is useful for content-based retrieval applications. Also, since the region of interest is isolated from other objects in the scene, the recording result may be useful for object based coding, such as in MPEG-4. Other research possibilities include virtual camera control for multiple objects, synchronizing the ROI output with PowerPoint slides, analyzing speaker activity, or using the ROI image as a basis for gesture tracking or face recognition.

Acknowledgment This research is in part supported by the following grants/awards: NSF #EIA-9986057, NSF#EIA-0080134, Samsung Electronics, and ONR#N00014-01-1-0391. The first author would like to thank Dr. Yanglim Choi, Samsung electronics, for many fruitful discussions.

7. REFERENCES

[1] Bergen . J. R., Anandan, P., Hanna, K. J., and Hingorani, R., "Hierarchical Model-Based Motion Estimation," in *ECCV'92*, pp.237-252, 1992.

[2] Burt, P. J., and Adelson, E. H., "The Laplacian Pyramid as a Compact Image Code," *IEEE Trans. On Communications*, 31, pp. 532-540, 1983.

[3] L. Campbell and A. Bobick, "Recognition of Human Body Motion Using Phase Space Constraints," in *ICCV'95*, pp. 624-630, 1995

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

[5] Darrell, T., Gordon, G., Harville, M., and Woodfill, J., "Integrated person tracking using stereo, color, and pattern detection," in P*roc. CVPR'98*, pp. 601-608, 1998.

[6] Foote, J. and Kimber, D.,"FlyCam: practical panoramic video and automatic camera control," *Proc. ICME'2000*, pp. 1419-1422, 2000.

[7] Haskell ,G., Puri, A. ,and Netravali,A. N., "Digital Video: An Introduction to MPEG 2," *Chapman and Hall*, 1997.

[8] Majumder, A., Seales, W. B., Gopi, M., and Fuchs, H. , " Immersive teleconferencing: a new algorithm to generate seamless panoramic video imagery," in *Proc ACM Multimedia'99*, pp.169-178, 1999.

[9] MPEG: http://www.mpeg.org

[10] Nayar, S. "Catadioptric omnidirectional camera," in *Proc CVPR'99*, pp. 482-488,1999.

[11] Nicolescu , M., and Medioni, G. , "Electronic pan-tilt-zoom: a solution for intelligent room systems, " in *Proc. ICME'2000*, pp. 1581-1584, 2000.

[12] O'Rourke, J., and Badler., N.I., "Model-based image analysis of human motion using constraint propagation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2(6), pp. 522-536, 1980.

[13] Sony EVI-D30: www.sony.com.

[14] Stiefelhagen, R., Yang, J. , and Waibel, A.," Modeling Focus of Attention for Meeting Indexing, " in *Proc ACM multimedia'99* pp. 3-10, 1999.

[15] Sun, X., and Manjunath, B. S. "Motion Quantized Alpha Histogram as a Video Unit Descriptor," *ISO/IEC JTC1/SC29/WG11/P75," MPEG7 group*, 1999.

[16] Swaminathan, R. and Nayar, S., "Non-metric Calibration of Wide-angle Lenses and Polycameras," in *Proc CVRP'99*, pp. 413-419,1999.

[17] Teodosio, L., and Bender, W., "Salient Video Stills: Content and Context Preserved," in *Proc. ACM Multimedia 93*, pp. 39-46, 1993.

[18] Wang, C. and Brandstein , M. S. "A Hybrid Real-Time Face Tracking System," in *Proc. ICASSP'98*, pp. 3737-3740, 1998.

[19] Wang, H. and Chang , S-F., "A Highly Efficient System for Face Region Detection in MPEG Video," *IEEE Trans. Circuits and Sys. for Video Tech*, 7(4), pp. 615-628, 1997

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

[21] Zhang, H. J., Low,C. Y., and Smoliar,S. W. "Video Parsing and Browsing Using Compressed Data," *Multimedia Tools and Applications*, 1(1): pp.89-111, 1995.