AN ACTIVE CAMERA SYSTEM FOR ACQUIRING MULTI-VIEW VIDEO

Robert T. Collins, Omead Amidi, and Takeo Kanade

Robotics Institute, Carnegie Mellon University

ABSTRACT

A system is described for acquiring multi-view video of a person moving through the environment. A real-time tracking algorithm adjusts the pan, tilt, zoom and focus parameters of multiple active cameras to keep the moving person centered in each view. The output of the system is a set of synchronized, time-stamped video streams, showing the person simultaneously from several viewpoints.

1. INTRODUCTION

For applications in human identification, activity recognition, 3D reconstruction, entertainment and sports, it is often desirable to capture a set of synchronized video sequences of a person from multiple camera viewpoints (see Figure 1). One way to achieve this is to set up a ring of cameras all statically aimed at a single point in space, and to have an actor perform at this fixation point while the video footage is shot. This is the method used to create spectacular special effects in the movie The Matrix, where playing back frames from a single time step, across all cameras, yielded the appearance of freezing the action in time while a virtual camera flew around the scene. However, in surveillance or sports applications it is not possible to predict beforehand the precise location where an interesting activity will occur, and therefore it is necessary to dynamically adjust the fixation point of multiple camera views. We have developed a system that tracks a person in real-time and adjusts the pan, tilt, zoom and focus of each camera to acquire synchronized multi-view video of a person moving through the scene.

2. SYSTEM DESIGN

The system design emphasizes modularity of each individual camera by minimizing the amount of information each camera has about the other cameras. Such modularity is useful in practical situations since all cameras may not necessarily be installed at the same time – in fact, new cameras may be added, and some may malfunction and be removed, even while the rest of the system is running. To achieve this level of modularity, each camera is calibrated independently with respect to a 3D scene coordinate system, and intercamera relationships are only implicitly represented through their joint individual relationships with the 3D scene. The cameras communicate by passing geometric "messages" through the shared 3D scene geometry. For example, there is no explicit representation of the epipolar geometry between any pair of cameras. A viewing direction from camera A is transformed into a 3D oriented ray in scene coordinates, which when projected into camera B essentially defines the appropriate epipolar ray. In this way, each camera can be treated independently, leading to a system in which truly distributed multi-camera processing can occur.

This work is supported by ONR contract N00014-00-1-0915.

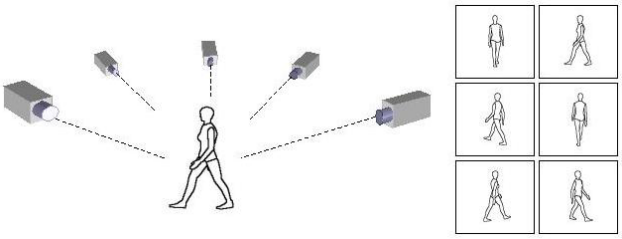
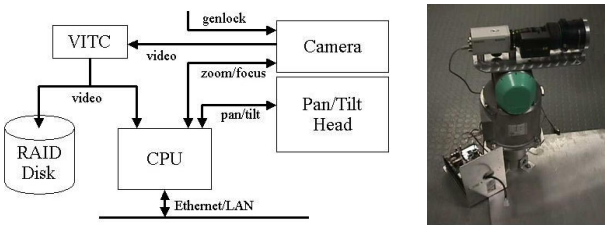


Fig. 1. The goal is to use multiple active cameras to acquire synchronized views of a moving person from multiple viewpoints.



(a) (b)

Fig. 2. (a) Block diagram of a single camera module for actively acquiring time-stamped video of a moving person. (b) Prototype hardware implementation of this module.

Figure 2a shows a block diagram of one active camera module. A computer CPU is connected to a camera mounted on a pan/tilt device. This CPU is responsible for

processing video from the camera, and based on the results it adjusts pan, tilt, zoom and focus parameters to maintain tracking, e.g. to keep the tracked person centered in the image. The camera is synchronized to a common system-wide genlock signal, so that the shutter for each camera fires at precisely the same time, resulting in video frames taken at the same time instant. Video from the camera is time-stamped using a VITC time code generator that inserts a system-wide time stamp directly into the vertical blanking interval of the video signal. This video branches both to the tracking CPU, and to a hard drive for recording. The tracking CPU communicates with other camera modules through a local area network.

(Figure 2b) shows a hardware prototype of this camera subsystem. The camera body is a Sony DXC-950, 3 CCD color camera that produces interlaced NTSC video output. To that is mounted a Canon YH18x6.5 motorized zoom lens. At high zoom, this lens has approximately a four degree field of view. The pan/tilt head consists of the first two joints of a Mitsubishi Heavy Industries industrial robot arm. This head was chosen for its accuracy, repeatability, and ability to carry a moderately heavy payload. A small industrial computer running the VxWorks operating system provides real-time image processing and camera control. Not shown in the picture is a Linux PC containing the RAID disk to which full-frame color video is streamed at 30 frames per second.

3. ACTIVE TRACKING ALGORITHMS

Each camera module actively adjusts its pan, tilt, zoom and focus to track a target person in real-time, as described below. Since single-camera tracking is sensitive to occlusions and clutter, the set of camera modules communicate to fuse their individual location estimates into a 3D estimate of the person's location in the scene. Cameras that lose track of the person can thus recover, as long as some subset of cameras has continued to correctly track the person.

3.1. Single-camera tracking

Much previous work in people detection and tracking for surveillance uses adaptive background subtraction [1, 2, 3, 4]. However, in the current system it is necessary to track a moving person while the camera is panning, tilting and zooming. Although theoretically video from a rotating and zooming camera can be registered and subtracted from a panoramic mosaic [5], adaptation to lighting changes is difficult since the whole panoramic scene is not being viewed continuously, and there are issues in representing panoramas at multiple scales when variable camera zoom is present.

To track objects from a continuously moving camera, we use the mean-shift algorithm [6, 7]. Each pixel in a window of interest within the incoming video frame is assigned a likelihood of belonging to the person being tracked, using an appearance model learned when the person was first sighted. This likelihood map represents an implicit probability distribution on the location of the person in the 2D video frame. The mean-shift algorithm is a non-parametric method for rapidly finding the nearest local mode of this distribution (Figure 3). The 2D location found is used to control the camera pan and tilt parameters to keep the person in the center of the image.

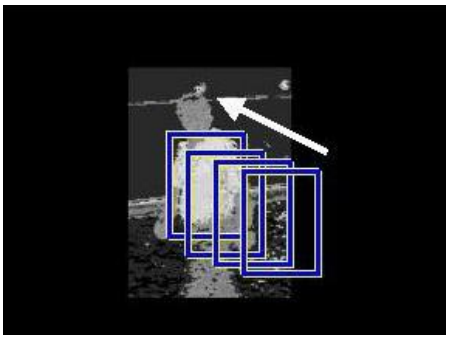


Fig. 3. Two-dimensional tracking is achieved by applying the mean-shift algorithm to an image where pixel values represent likelihood of belonging to the tracked person. The mean-shift algorithm is a non-parametric method for climbing to the nearest local mode of this likelihood map.

We currently use a simple appearance model based on a histogram in normalized color space of the pixel colors falling within a rectangle centered on the person. Future work will improve the appearance model to incorporate multiple cues including texture, shape and predicted motion.

3.2. Multi-camera location updates

Camera modules communicate to achieve a consensus on the person's 3D location. This allows cameras that have a good view of the person to aid other cameras with poor or occluded views. Once per second, each camera i broadcasts a time-stamped UDP packet containing its 3D focal point location c_i, its principal viewing ray orentation u_i computed from current pan and tilt angles, and a weight w_i that specifies how confident the camera is in its current tracking results. The message from each camera i thus constrains the person to lie along a 3D ray $c_i + ku_i$, with k being a positive distance along the ray. When two or more cameras view the same person, an estimate of the person's 3D location can be computed via triangulation of these viewing rays (Figure 4). Due to inaccuracies, these rays will not intersect exactly at a single point, but we can compute a *pseudo-intersection* point P that minimizes the sum of squared distance to each pointing ray. Point P is found as the solution to the linear

system of equations

$$[\sum_{i}^{n} w_i(I - u_i u_i')]\, P \;=\; \sum_{i}^{n} w_i(I - u_i u_i')c_i \quad . \qquad (1)$$

This 3D estimated location P is used to adjust the pan/tilt angles of each camera, to an extent that depends on the camera's tracking confidence w_i. Cameras from which the person is occluded can therefore continue to track the virtual position of the person from their viewpoint. The distance from estimated 3D location P to each camera's location c_i can be used to control camera zoom and focus to keep the person the same size and in sharp focus in all the images, even though the cameras are different distances away from the person.

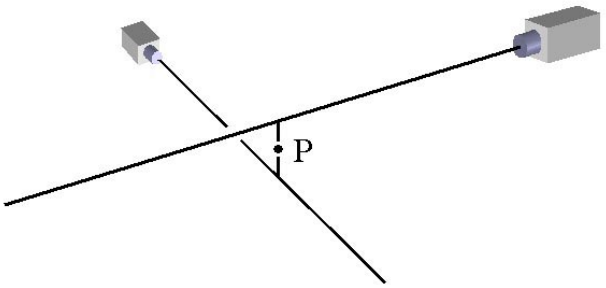


Fig. 4. Multiple camera viewing rays are fused into a single object location estimate by finding the pseudo-intersection point P that minimizes sum of squared distance to each pointing ray.

4. CAMERA CALIBRATION

Before operation of the system, each camera is calibrated so that its relationship to the scene is explicitly known. This requires determining the pose (location and orientation) of the camera with respect to a scene coordinate system, determining the relationship of the zoom control parameter to angular field of view, and determining the relationship of the focus control parameter to the distance of objects in the scene.

Camera pose is determined by measuring pan/tilt angles towards a set of distinguished points or "landmarks" with known 3D coordinates. The 3D landmark points are determined prior to calibration by surveying with a GPS unit or theodolite. Sighting each landmark involves rotating the pan/tilt device from a user interface, until the landmark point is centered within the field of view of the camera. The pan/tilt parameters at this position are then stored with the X,Y,Z coordinates of the landmark, to form one pose calibration measurement. Camera orientation R and location c are determined by an optimization procedure that minimizes the angle between pan-tilt viewing rays rotated by R and direction vectors from the camera origin c to the 3D landmark points. The basic pose solution method is presented in [8].

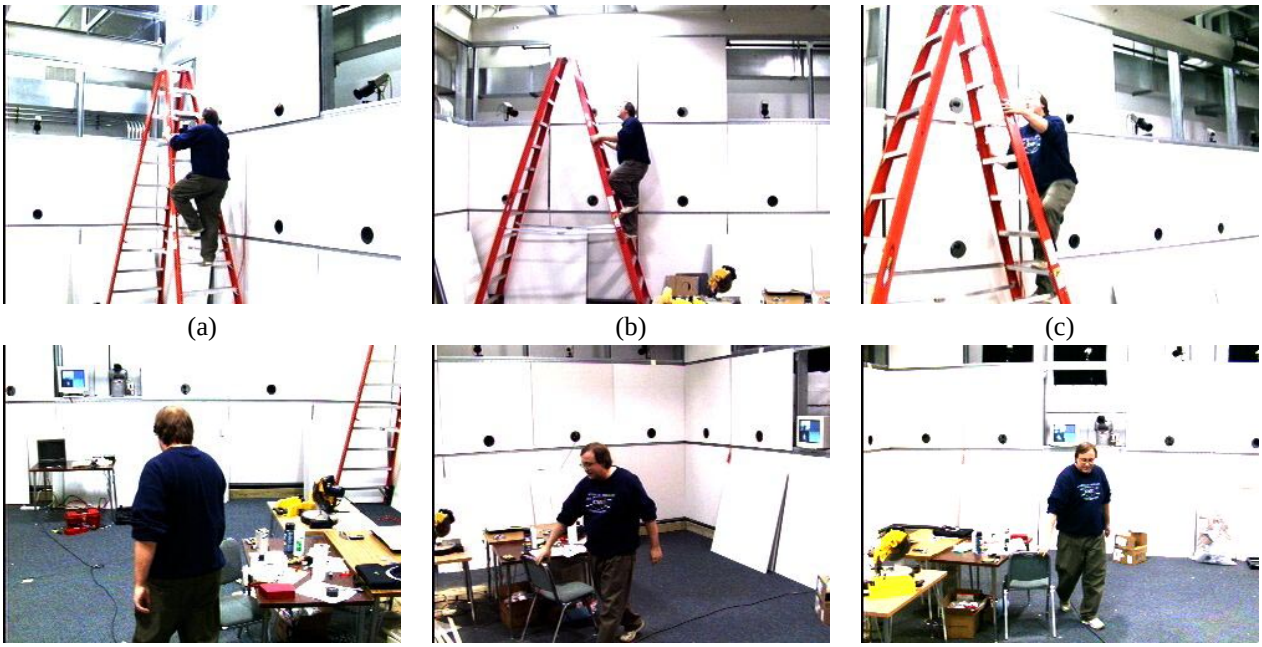
Computer control of motorized zoom lenses involves sending the desired zoom and focus as a command to the camera/lens system. The effect of the value of these parameters on physical lens settings must be determined through calibration. The zoom parameter is calibrated by stepping through the allowable values and measuring the field of view after the motorized zoom is complete. User control of the pan/tilt head is used to actively and directly measure the field of view at each setting. Some visible landmark is chosen in the scene, roughly level to the pan/tilt device. The head is then directed by hand to find the left and right pan angles that bring the landmark to the far right and left edges of the image. Alternatively, an automated method based on self-calibration via active camera rotation can be used [8].

The relationship between focus parameter value and object distance is calibrated by focusing on objects at different distances from the camera, and deriving an implicit relationship between focus value and distance. This implicit relationship is represented as a lookup table of focus parameter settings, indexed by inverse distance to the desired focal distance in the scene. Focus to points at intermediate distances is determined by interpolation of these stored table values. A table indexed by inverse object distance is preferrable to one indexed by distance, since good results can be achieved using only linear interpolation on a sparse set of distance/focus measurements.

5. SAMPLE RESULTS

A three-camera prototype system has been built in the Virtualized Reality lab at Carnegie Mellon University. This small demo system is used as follows. Initially, the room is empty, and a background model is acquired from each camera. A person then enters the room, and is detected by background subtraction and thresholding. Pixels that are determined to be part of the person's silhouette are used to estimate a normalized color histogram, which forms the appearance model for the mean-shift tracking algorithm. After the appearance model is acquired, the cameras begin active tracking and recording as the person moves throughout the space.

Figure 5 shows sample results from one recording session. Each row shows corresponding frames from each of the three cameras for a specific time sample. During recording, the demo tracking system automatically adjusted the pan and tilt parameters of each camera in real-time to keep the person's torso centered in each image.



(a) (b) (c)

(d) (e) (f)

Fig. 5. (a)-(c) One time sample from synchronized video taken by a three-camera system that actively acquires multi-view video by controlling pan and tilt of each camera in real-time. (d)-(e) Another time sample from the same recording session.

6. SUMMARY

A system has been developed for acquiring multi-view video of a person moving through the scene. The approach is to use a real-time appearance-based tracking algorithm to control the pan, tilt, zoom and focus parameters of multiple active cameras. The output of the system is a set of synchronized, time-stamped video streams of the person, seen simultaneously from several viewpoints. The system design emphasizes the modularity of each individual camera subsystem by minimizing the amount of information that each camera has about the other cameras. The cameras communicate by passing geometric "messages" through the shared 3D scene geometry, enabling a distributed approach to multi-camera active tracking.

7. REFERENCES

[1] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *International Conference on Computer Vision*, 1999, pp. 255–261.

[2] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *European Conference on Computer Vision*, 2000.

[3] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multi-sensor surveillance," *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1456–1477, October 2001.

[4] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Computer Vision and Pattern Recognition*, 1999, pp. II:246–252.

[5] F. Dellaert and R. Collins, "Fast image-based tracking by selective pixel integration," in *ICCV 99 Workshop on Frame-Rate Vision*, September 1999.

[6] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *IEEE Computer Vision and Pattern Recognition*, 2000, pp. II:142–149.

[7] G.R. Bradski, "Computer vision face tracking for use in a perceptual user interface," in *IEEE Workshop on Applications of Computer Vision*, 1998, pp. 214–219.

[8] R. Collins and Y. Tsin, "Calibration of an outdoor active camera system," in *IEEE Computer Vision and Pattern Recognition (CVPR '99)*, June 1999, pp. 528 – 534.