An Efficient Active Camera Model for Video Surveillance[*]

Karthik Sankaranarayanan   James W. Davis

Dept. of Computer Science and Engineering

Ohio State University

Columbus, OH 43210 USA

{sankaran,jwdavis}@cse.ohio-state.edu

Abstract

We propose an efficient active camera model to map image coordinates to the camera's pan-tilt orientations in constant time. The model is based on the elliptical locus of the projections of a fixed point on the original image plane of a moving camera. The parametric location of this point along the ellipse defines the change in camera orientation. This model does not require any knowledge of camera parameters other than the focal length. Using synthetic and real data, we show the accuracy of the model by generating seamless spherical panoramas from a set of images and demonstrate the applicability of the model with a real-time active tracking application.

1. Introduction

Pan-tilt (and sometimes zoom) cameras are extensively used for wide-area surveillance. Tracking and activity analysis in such environments requires continual coverage of the entire area by the cameras. While these cameras cannot adequately cover the entire space in a single view, they can be directed to look at changing areas of interest using pan and tilt controls. For example, an automatic system could be developed to continually track a person while keeping the person centered in the image. Therefore, when processing images captured by these cameras, there is a need to map the image pixel coordinates from tracking to their corresponding pan-tilt orientations in world space (to move the camera to a new location corresponding to the centroid of the target in the current image). Therefore, understanding this mapping is a fundamental tool which aids active camera applications.

Complex camera models have been suggested to address this problem [4, 6]. However, we propose a simple, yet reliable, model which given the current pan-tilt value of the camera, computes in constant time, the pan-tilt location of any other arbitrary pixel in the given image. All that is required for this model is an estimate of the focal length (in pixels) of the camera. There is no need to know any other camera or scene-specific parameters. The principal idea behind this model is that the locus of a point on a fixed image plane, as the camera rotates, forms an ellipse from which the pan and tilt can be computed. In this paper, we address cameras mounted on pan-tilt heads (zoom will be addressed in future work).

In Sect. 2, we discuss previous approaches to active camera modeling. Section 3 describes the proposed model in detail. Section 4 covers the experiments which involve panorama generation and an application of the model to active tracking. We conclude with a summary in Sect. 5.

2. Related Work

Most existing camera models assume idealized mechanics where the optical center and geometric center of the camera are collocated, and the rotation axes are aligned [1, 3, 5, 9]. However, this is not true with most commercially available cameras where the pan-tilt motor center does not coincide with optical center of the camera. Extending the ideal model, [4] accounts for rotations around arbitrary axis in space. This results in a great deal of additional data having to be collected to robustly calibrate all camera parameters, which they do using LED marker data. Moreover, this technique employs an iterative minimization technique that is computationally expensive. In [6], they propose a more complex model where they employ a highly non-linear relationship between image coordinates and world coordinates which is then linearized. They solve for a closed-form solution by iterating until convergence. In [2], a "correction matrix" is used to continuously update the camera parameters with the latest pan and tilt information as the camera moves. However, this information is not precise enough for pixel-level mapping.

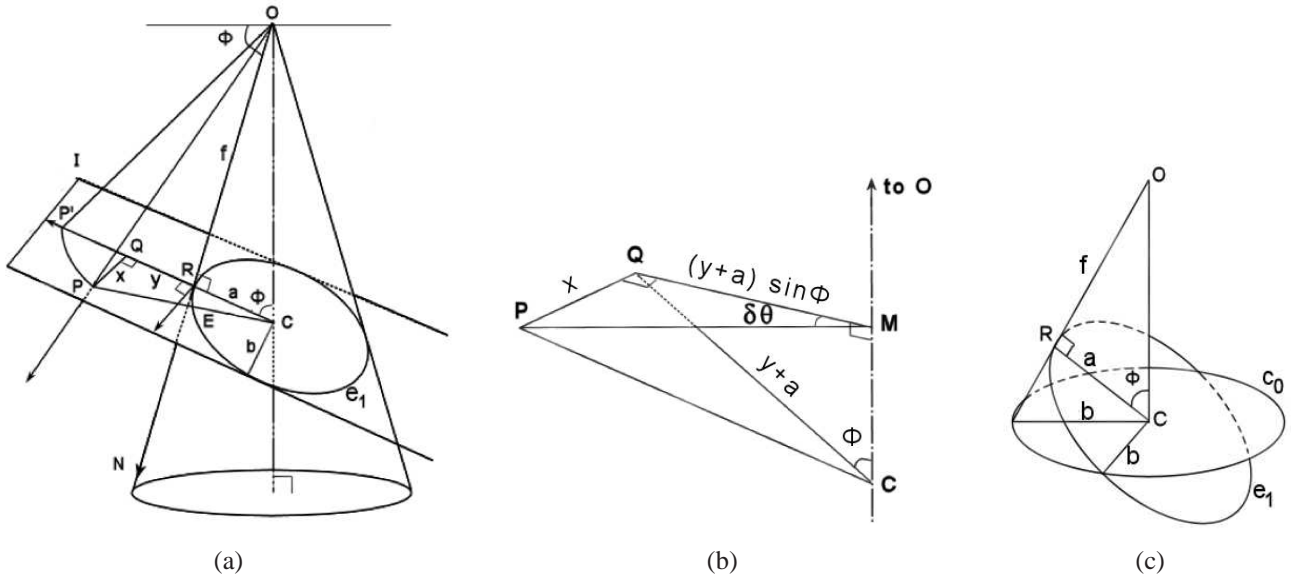[*]Appears in *IEEE Workshop on Applications of Computer Vision (WACV) 2008.*

Figure 1. Camera Geometry. (a) Overall geometry of the camera model. (b) Calculation of change in pan $\delta\theta$. (c) Calculation of minor axis b (c_0 is a circle of radius b centered at C and parallel to the ground plane).

3. Framework

We present a simple model to map the image coordinates (x, y) to the world camera orientations $(pan, tilt)$. The main idea behind this mapping is that the locus of a point on a *fixed* (the original) image plane, as the camera rotates, is an ellipse. The goal is to find the change in pan and tilt along this ellipse to center the point (x, y) in the new image (after applying the new pan-tilt rotations). The change in pan is essentially a function of coordinates (x, y), the focal length, and the major axis of the ellipse. This change is pan is further used to compute the location of the point along the ellipse, in terms of a parametric equation. This parameter value, together with the focal length, is then employed in the calculation of the change in tilt. We describe the geometry of this model in the following section.

3.1. Geometry

Figure 1 provides the main geometric relationships described in this section. The point O represents the optical center (nodal point) of the camera, which is currently tilted down at angle ϕ from the horizon. The ray ON is the principal axis and when the camera pans a complete 360 degrees at tilt ϕ, a cone carved out. The image plane I is perpendicular to the principal axis ON and intersects ON at point R, which is the principal point, at a distance equal to the focal length f. For any arbitrary point P on the image plane I, we wish to understand its locus on *this* image plane as the camera pans through some arbitrary angle.

Consider the principal point R. As the camera pans, the projections of successive principal foci $R_1, R_2, R_3, ...$ onto

the *original* image plane I form an ellipse. This is evident from the fact that the intersection of a plane with a cone is an ellipse. Therefore, the locus of projections of the R_i onto I is the ellipse (e_1) centered at C. Similarly, the locus of an arbitrary point P on I as the camera pans is also an ellipse (e_2). This ellipse intersects the Y axis on the image plane at point P'. Knowing the camera's current $(pan, tilt)$ location, our goal is to map P's image coordinates (x, y) to a change in pan ($\delta\theta$) and a change in tilt ($\delta\phi$) that would center the point P in the new image (after panning and tilting), thus making $P = R$ in the new image.

Change in Pan: First we compute the change in pan $\delta\theta$. Consider a plane parallel to the ground plane and passing through the point P (see Fig. 1(b)). Suppose this plane intersects line OC at M. Therefore, $\triangle PMQ$ is parallel to the ground plane. The desired change in pan $\delta\theta$ is the angle between planes MPC and MQC. The angle between two planes is the measured along a plane which is perpendicular to both planes. In our case, this plane is $\triangle PMQ$ and hence, the desired angle $\delta\theta$ is $\angle PMQ$.

This angle is computed as follows

$$\delta\theta = \tan^{-1}\left(\frac{x}{(y+a) \cdot \sin\phi}\right) \tag{1}$$

where a is the length of major axis of the inner ellipse e_1 and is given as (see Fig. 1(a))

$$a = \frac{f}{\tan\phi} \tag{2}$$

Substituting the value of a from Eqn. (2) in Eqn. (1), we

get

$$\delta\theta = \tan^{-1}\left(\frac{x}{y \cdot \sin\phi + f \cdot \cos\phi}\right) \tag{3}$$

Additionally, the length of the minor axis (b) of e_1 is given as (see Fig. 1(c))

$$b = \frac{a}{\sin\phi} \tag{4}$$

Change in Tilt: Next, to compute the change in tilt $\delta\phi$, we need to find $\angle P'OR$ (see Fig. 1(a)). Let line CP intersect the inner ellipse e_1 at point E. In terms of the parametric equation of an ellipse, we can represent point E as $(a\cos(t), b\sin(t))$. Since the two ellipses e_1 and e_2 are concentric, the values of this parameter t at E (for e_1) and at P (for e_2) are equal. If A and B are the major and minor axes of the outer ellipse e_2 (of which the arc PP' is a part), we can represent point P as $(A\cos(t), B\sin(t))$. From Fig. 1(a) we see that

$$A \cdot \cos(t) = y + a \tag{5}$$
$$B \cdot \sin(t) = x \tag{6}$$

Also since $\frac{A}{B} = \frac{a}{b}$, we can now compute the value of t as

$$t = \tan^{-1}\left(\frac{a}{b} \cdot \frac{x}{y+a}\right) \tag{7}$$

Using the parametric equation for P, along with Eqn. (7), we can find the length (A) of the major axis of e_2 as

$$A = \frac{y+a}{\cos\left(\tan^{-1}\left(\frac{a}{b} \cdot \frac{x}{y+a}\right)\right)} \tag{8}$$

We can now find $\angle P'OR$ (i.e., $\delta\phi$) as

$$\delta\phi = \tan^{-1}\left(\frac{A-a}{f}\right) \tag{9}$$

Using these equations, we get the desired changes in pan and tilt angles necessary to center point P in the image plane.

3.2. Focal Length

The focal length f (in pixels) is the only camera parameter needed by the model. We learn the focal length f using several points whose pan-tilt locations are known in the scene. We start by pointing the camera to an area rich in features/structure and then select several key point locations in the image. Next, we move/center the camera to those locations and store the corresponding pan and tilt changes from the initial orientation. Using the changes ($\delta\theta_i$, $\delta\phi_i$) for each of the locations (x_i, y_i) and the initial/base tilt (ϕ) value,

we employ a least-squares formulation based on Eqn. (3) to learn the value of f

$$\begin{pmatrix} \frac{\tan\delta\theta_1 \cos\phi}{x_1 - y_1 \sin\phi \tan\delta\theta_1} \\ \vdots \\ \frac{\tan\delta\theta_n \cos\phi}{x_n - y_n \sin\phi \tan\delta\theta_n} \end{pmatrix} f = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \tag{10}$$

We note that this model assumes perspective projections, and hence radial distortion correction can be employed if needed.

3.3. Advantages

An important advantage of the proposed model is that it requires no knowledge of the field-of-view or camera parameters other than the focal length. Also, previous models [4, 6] have proposed approaches for the problem of misalignment of the pan and tilt axes and the issue of the optical and geometric center of the camera not being collocated. These problems are observed in most commercially available cameras. However, our model is independent of these problems since it "simulates" a virtual camera located at the optical center O. As we do not model the actual camera field-of-view, the images could be considered to have been taken from this virtual camera centered at O. Hence the complete pan-tilt space is oriented with respect to this virtual camera at O, not the actual camera geometric center. Even though the geometric center is displaced, the variation of pan and tilt of this virtual camera at any instant is always consistent with the pan and tilt of the real camera. Also, our model is much simpler as compared to existing approaches which use either an expensive table-lookup process [10] or iterative convergence methods to get closed-form solutions [4, 6]. Our model is also faster since the algorithm computes the necessary values in constant time.

4. Experiments

The proposed model enables the mapping of the (x, y) coordinates of any image pixel to its corresponding $(pan, tilt)$ camera orientation in world space. To examine and test this model, we performed multiple experiments using synthetic and real data. The first experiment tests the accuracy of the model over a wide field-of-view by capturing images across an entire scene, mapping the individual pixels to their corresponding location in pan-tilt, and lastly generating panoramic images. If the model is sufficient, the individual images should register correctly and the resulting panoramas should be seamless. We also demonstrate a real-time active camera system, using the proposed model and an established tracking algorithm to automatically move the camera such that the subject being tracked is always centered in the image.
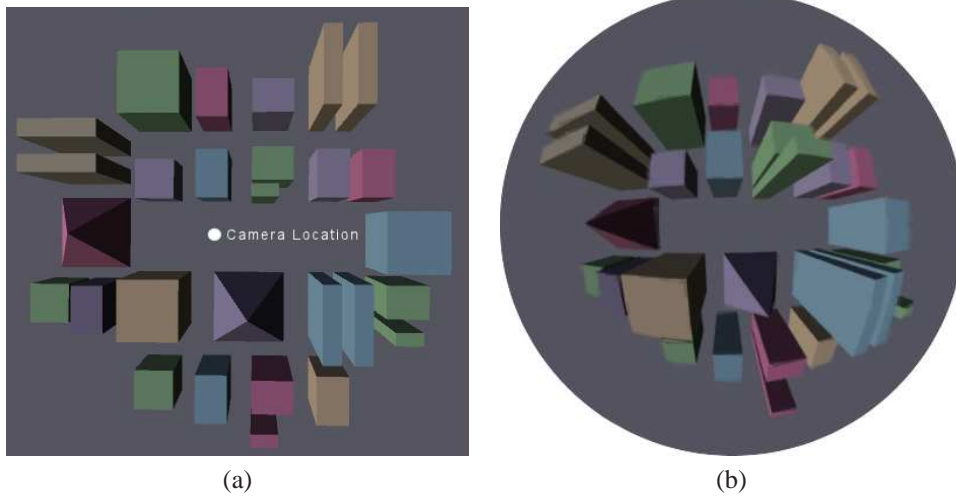
<div align="center">(a)   (b)</div>

Figure 2. Synthetic data experiment. (a) Top-down view of the virtual environment showing location of the camera. (b) Resulting panorama.

4.1. Panorama generation

Initially, we tested our model using a virtual environment synthetically generated in Maya. The environment consisted of multiple 3D geometric objects placed at random locations in the scene (see Fig. 2(a)). A virtual pan-tilt camera was positioned near the center of these objects, above the scene. By orienting the camera to a set of pre-configured pan-tilt locations, we collected images such that they cover the complete pan-tilt space. For the zoom level employed, a total of 76 images (of 640×480 resolution) were required to cover the scene with minimal overlap. After estimating the focal length (Sect. 3.3), we mapped each pixel of every image to its corresponding pan-tilt orientation in the 360×90 space (at a resolution of 0.1 degree). The pan-tilt RGB data were plotted on a polar coordinate system where the radius varies linearly with tilt and the angle represents the pan. No smoothing or blending of data was employed other than using linear interpolation to fill any missing pixels in the final panorama. The result is a spherical panorama as shown in Fig. 2(b)[1]. As shown, the 3D objects register correctly and the alignments are proper across changes in pan and tilt.

Next, we used real pan-tilt cameras to test the proposed model. We employed Pelco Spectra III SE surveillance cameras which pan a complete 360 degrees and have a tilt range of 0 to 90 degrees[2]. Similar to the experiment with the synthetic data, the same number of images were captured such that they cover the entire scene. For each image, the pixel locations were mapped to their corresponding $(pan, tilt)$ values and the panorama was generated.

---

[1]A "true" spherical panorama could be created by varying the radius with the cosine of the tilt.

[2]We observed that the camera's tilt encoder was off by +2 degrees, so the tilt values were adjusted in the code.

Since the automatic gain control (AGC) could not be disabled as the camera moved to new locations, there were illumination differences in the scene and consequently these differences are reflected in the panorama (see Fig. 3(a)). To overcome this brightness change problem, we used the technique of deriving intrinsic images [8] to compute a reflectance-only panoramic image. By shifting the pre-configured pan and tilt coverage locations in 5 and 1 degree intervals respectively, we generated multiple panoramas of the same scene at different times of the day (thus having different illuminations). While generating the intrinsic images, we separated the luminance and chrominance channels from the RGB panoramas by changing the color space from RGB to YIQ. Next, we ran the intrinsic algorithm on only the Y channel, then combined the result with the mean I and Q channels of the image set, and lastly converted back from YIQ to the RGB color space. The result provides a more uniform panorama, as shown in Fig. 3(b).

The above experiment was tested on multiple cameras mounted at varying heights (on 2, 4, and 8 story buildings). The results after intrinsic image generation are shown in Fig. 4. These experiments demonstrate the applicability of the model to real cameras at different heights.

4.2. Active Camera Tracking

Using the proposed model, we next developed an active camera application to test the ability to track people in real-time using wide-area surveillance cameras. The active camera system moves the camera such that, at all stages during tracking, the target being tracked is centered in the image.

The tracking algorithm used in this system is the Covariance Tracker proposed by [7]. This algorithm models the target using a covariance of spatial and color features and performs template-based matching in successive frames to
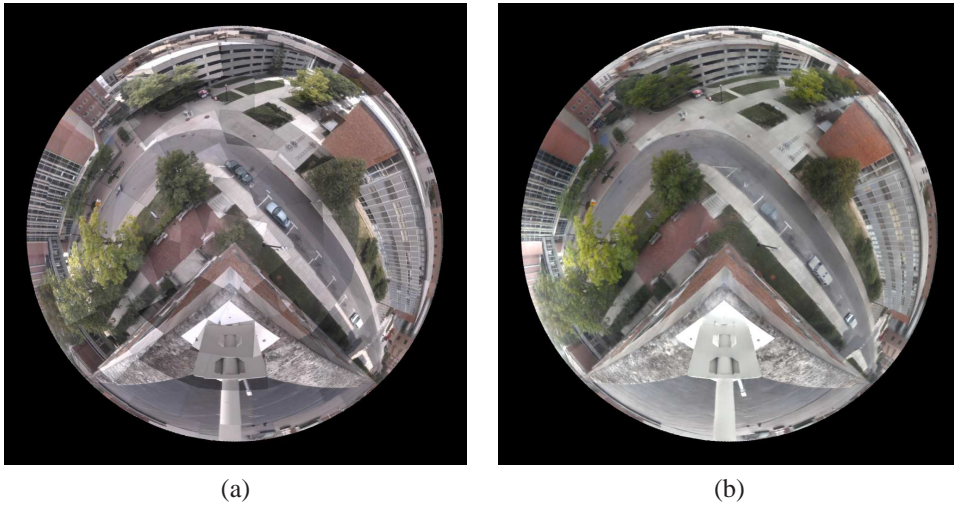
Figure 3. Brightness differences removal. (a) Panorama showing illumination differences before intrinsic image generation. (b) Final panorama.

track the target. To model the target, we selected the feature vector $f_k = [x \ y \ I(x,y) \ I_x(x,y) \ I_y(x,y)]$. The target window was represented as a covariance matrix of these features. This captures not only their spatial and statistical properties, but also their correlation within the same representation. The advantage of the covariance matrix is that its dimensionality is small and it adapts well to temporal changes in the model. The distance metric used to compute dissimilarity between matrices was the sum of the squared logarithms of their generalized eigenvalues. A model update method was employed by keeping a set of recent covariance matrices and computing their mean over the Riemannian manifold. In our system, we restricted our search for the best match to a local search window (40×40).

The tracker is initialized manually and as the target moves, the best match is found by the tracker in successive frames. Using the proposed model, the (x, y) center coordinates of the best matching patch are converted to the required change in pan and tilt angles required to center the target. While moving the camera, the system polls the motor to see if it has reached the desired pan-tilt location. Once it has, the next frame is grabbed and the best match is again found, and the process repeats.

Figure 5 shows the results of active camera tracking from two different cameras. The image chips of the targets tracked at different times are shown on the left. The target's raw (not smoothed) track is overlaid on the panorama and the chips shown are marked on the tracks. In Fig. 5(a), track 1 starts off with the camera at a low tilt angle, moves towards increasing tilt values reaching almost the maximum tilt by coming directly under the camera, moves left and then enters the building. Track 2 shows a large variation in pan angles with the tilt angles varying slightly. Figure 5(b)

shows a camera at a low height. Track 1 shows a person walk across the scene where the pan varies continuously across a large range. Track 2 shows a person start at a low tilt angle, walk randomly around the sidewalk, such that there are large variations in both pan and tilt.

The tracking results are obtained from two cameras at different heights, and the results demonstrate the robustness of the proposed model and its applicability to real-time active tracking.

5. Summary

We proposed a novel camera model to map image coordinates to pan-tilt orientations in constant time. This fast model requires no knowledge of the scene structure or camera parameters other than the focal length. The model is based on the observation that the locus of a point on a fixed image plane, as the camera rotates, is an ellipse. Therefore, using the parametric location of the point along the ellipse, we solve for the desired change in pan and tilt angles needed to center the point. We tested the model by mapping images to their pan-tilt orientations and generating accurate panoramas. We also demonstrated an active camera tracking application which used the proposed model to maintain a centered target at all times. In future work, we plan to incorporate the variation of focal length with changes in zoom so that the camera can choose to zoom in or out on a target as needed.
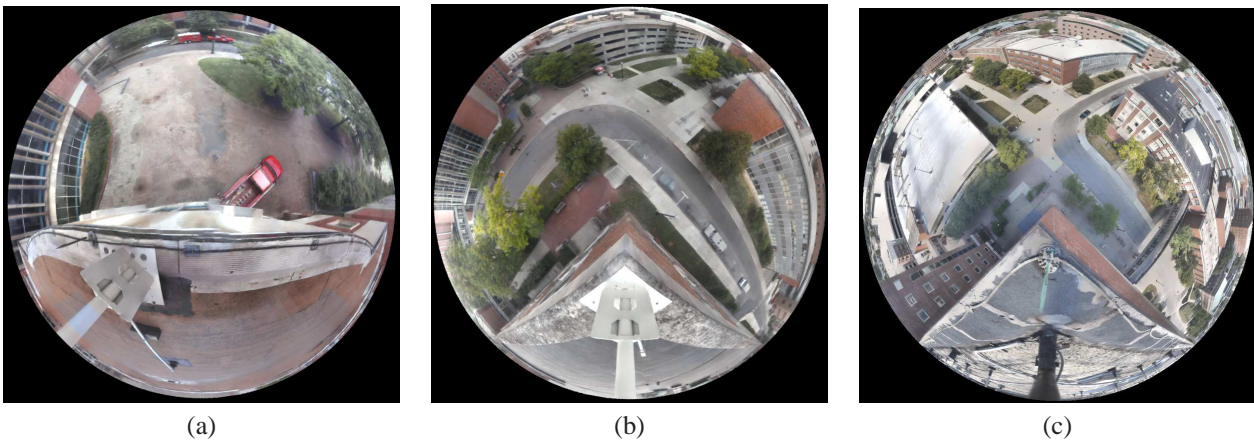
6. Acknowledgements

Figure 4. Resulting panoramas from three different cameras located on (a) 2, (b) 4, and (c) 8 story buildings.

their help with the experiments.

References

[1] A. Basu and K. Ravi. Active camera calibration using pan, tilt and roll. *IEEE Trans. Sys., Man and Cyber.*, 27(3):559–566, June 1997.

[2] K. Bernardin, F. van de Camp, and R. Stiefelhagen. Automatic person detection and tracking using fuzzy controlled active cameras. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[3] R. Collins and Y. Tsin. Calibration of an outdoor active camera system. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages I: 528–534, 1999.

[4] J. Davis and X. Chen. Calibrating pan-tilt cameras in wide-area surveillance networks. In *Proc. IEEE Intl. Conf. on Computer Vision*, Nice, France, 2003.

[5] S. Fry, M. Bichsel, P. Muller, and D. Robert. Tracking of flying insects using pan-tilt cameras. *Journal of Neuroscience Methods*, 101:59–67, 2000.

[6] A. Jain, D. Kopell, K. Kakligian, and Y. Wang. Using stationary-dynamic camera assemblies for wide-area video surveillance and selective attention. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[7] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on means on Riemannian manifolds. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[8] Y. Weiss. Deriving intrinsic images from image sequences. In *Proc. IEEE Intl. Conf. on Computer Vision*, 2001.

[9] D. Woo and D. Capson. 3D visual tracking using a network of low-cost pan/tilt cameras. *Canadian Conference on Electrical and Computer Engineering*, 2:884–889, 2000.

[10] X. Zhou, R. T. Collins, T. Kanade, and P. Metes. A master-slave system to acquire biometric imagery of humans at a distance. In *ACM SIGMM 2003 Workshop on Video Surveillance*, pages 113–120, 2003.

track 1 (red)

track 2 (blue)

(a)

track 1 (red)

track 2 (blue)
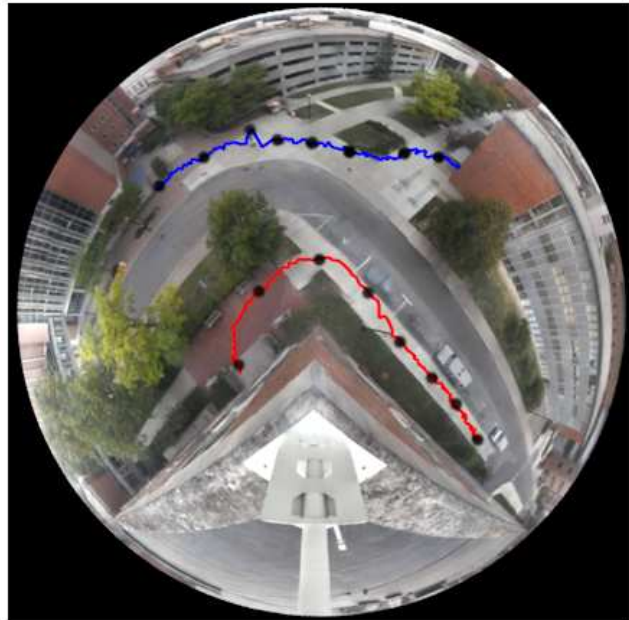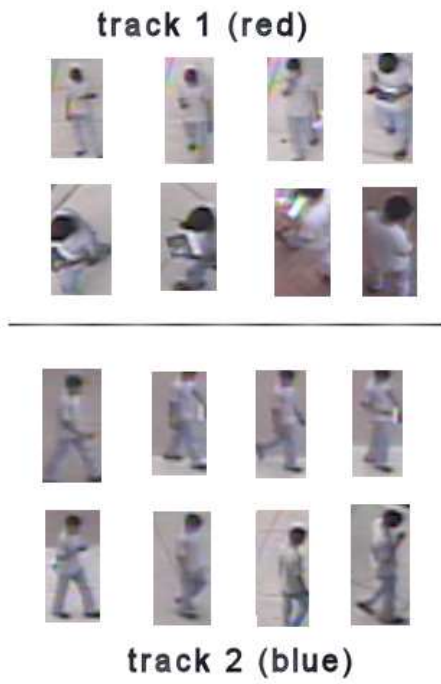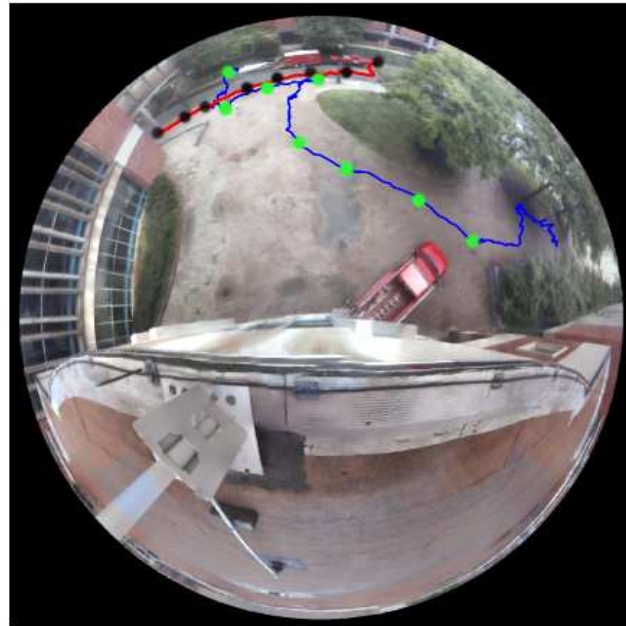
(b)

Figure 5. Active camera tracking. (a) Camera located 4 stories high. (b) Camera located 2 stories high. Dots along the track are locations shown in the image chips.