

# PTZ Camera Modeling and Panoramic View Generation via Focal Plane Mapping<sup>\*</sup>

Karthik Sankaranarayanan and James W. Davis

Dept. of Computer Science and Engineering  
Ohio State University  
Columbus, OH, USA  
{sankaran,jwdavis}@cse.ohio-state.edu

**Abstract.** We present a novel technique to accurately map the complete field-of-coverage of a camera to its pan-tilt space in an efficient manner. This camera model enables mapping the coordinates of any  $(x, y)$  point in the camera's current image to that point's corresponding orientation in the camera's pan-tilt space. The model is based on the elliptical locus of the projections of a fixed point on the original focal plane of a moving camera. The parametric location of this point along the ellipse defines the change in camera orientation. The efficiency of the model lies in the fast and automatic mapping technique. We employ the proposed model to generate panoramas and evaluate the mapping procedure with multiple PTZ surveillance cameras.

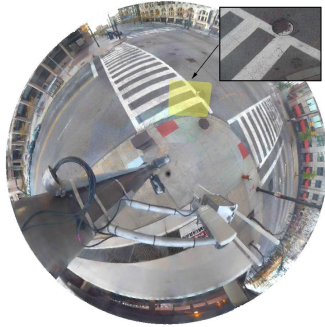
## 1 Introduction

Pan-tilt-zoom (PTZ) cameras are extensively used in wide-area surveillance applications. Tracking, monitoring, and activity analysis in such environments require continual coverage of the entire area by these cameras. While most computer vision algorithms such as object detection, tracking, etc. work on the Cartesian x-y space of images captured by these cameras, the movement and control of these cameras take place in an angular pan-tilt space. Therefore, in order to relay information across these spaces to comprehensively utilize the pan-tilt capabilities, a fast and accurate pixel to pan-tilt mapping model is required. For example, an active tracking system would require such a real-time mapping in order to continually track a target with a PTZ camera to keep the target centered in its view as it moves across the scene. Such a mapping would also need to be employed when constructing models for pedestrian activity patterns across an entire scene and not just in local views. In these applications, when processing images captured by the camera, there is a need to map the local image pixel coordinates to the corresponding pan-tilt orientations in world space.

Although various complex camera models have been proposed for these tasks [1–5], we expand on the preliminary work of [6] and propose a real-time, reliable

---

<sup>\*</sup> Appears in *Tenth Asian Conference on Computer Vision*, November 2010



**Fig. 1.** High resolution spherical panorama generated using the proposed model showing complete field-of-coverage of a PTZ camera. Inset: A single view from the camera.

and completely automatic PTZ mapping model. We theoretically derive the minimum formulation necessary to map any arbitrary point on the focal plane of a PTZ camera to its corresponding pan-tilt orientation, based on the current pan-tilt value of the camera. The principal idea that we exploit here is that the locus of a point on a fixed image plane, as the camera pans, forms an ellipse from which the pan and tilt can be recovered. Our formulation shows that an alternative (and simpler) linear field-of-view based mapping approach is not sufficient while at the same time demonstrates that more complex methods are not required. Therefore, while the proposed model we derive is computationally simple, it is not simplistic in its functionality (as established in the experiments).

Important contributions of the work here (beyond [6]) include 1) new extensions to the previous PT-only model to propose a complete pan-tilt-*zoom* mapping model, 2) studies and experiments analyzing the variations between focal length and zoom in the mapping, 3) quantitative experiments and discussion on the model (previous work was only qualitative), 4) new experiments demonstrating robustness with different brands of cameras (previous work was with one type of camera), and 5) new qualitative and quantitative experiments within a full pan-tilt-zoom active tracking application. Due to the nature of the proposed approach, extensive experiments are required and presented. Our resulting new system is also completely automatic (unlike the previous) which is important for scalability in large distributed camera networks.

The rest of the paper is organized as follows. In Sect. 2, we discuss previous approaches to the problem addressed. Section 3 describes the proposed model in detail and Sect. 4 presents the experiments. We discuss the contributions of this work in Sect. 5 and conclude with a summary in Sect. 6.

## 2 Related Work

Even though in commercially available cameras the pan-tilt motor center does not coincide with optical center of the camera, most existing camera models as-

sume idealized mechanics where the optical center and geometric center of the camera are collocated, and the rotation axes are aligned [7–11]. A simple model is also assumed in [12] and requires bundle adjustment to obtain global image alignment in the generated panoramas. In [2], they propose a more complex model where they employ a highly non-linear relationship between image coordinates and world coordinates which is then solved for a solution by iterating until convergence. While [1] accounts for rotations around arbitrary axis in space, in order to robustly calibrate all camera parameters it requires a controlled environment using LED marker data. Moreover, this technique employs an iterative minimization technique that is computationally expensive. Similarly [4] employs a search mechanism to minimize an SSD matching score function, which again is computationally expensive. In [3], 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.

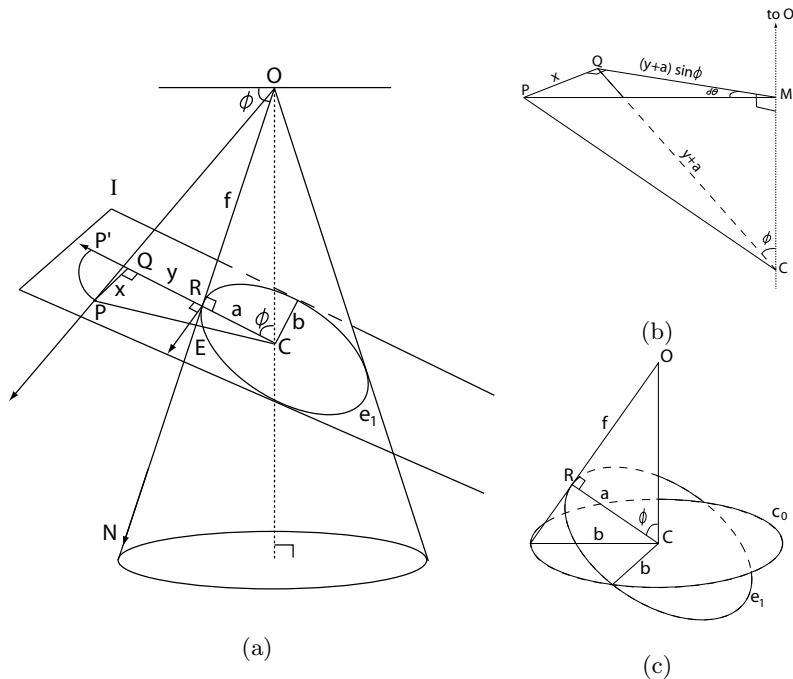
This paper extends the initial work introduced by [6] by proposing techniques for a fully automatic and *complete* pan-tilt-zoom mapping model. The main advantage and novelty of the proposed model is that it is independent of the aforementioned problems since it “simulates” a virtual camera located at the optical center. The pan-tilt space is oriented with respect to this virtual camera at the optical center, not the actual camera geometric center. Even though the geometric center is displaced, the variation/change of pan and tilt of this virtual camera at any instant is consistent with the pan and tilt of the actual camera. *See supplemental slides.* Since our model depends on changes in pan and tilt angles, the images could be considered to have been taken from this virtual camera. By presenting this compact model with thorough and robust results, we demonstrate that the excess computations in the previous work are unnecessary.

### 3 Proposed Camera Model

In this section, we describe the proposed mapping model (initial work is presented in [6]), techniques to automate the model parameter calculations, and analysis of the model with variations in zoom. The main idea behind the mapping model is that the locus of a point along the principal axis on a *fixed* infinitely-extended image plane as the camera pans, is an ellipse. We wish to compute the changes in pan and tilt along this ellipse to map the image point  $(x, y)$  to its corresponding  $(pan, tilt)$  orientation.

#### 3.1 X-Y to Pan-Tilt Mapping

The mapping model presented here is based on the camera geometry shown in Fig. 2. In Fig. 2(a),  $O$  represents the optical center of the camera whose principal axis (ray  $ON$ ) intersects the image plane  $I$  at its principal point  $R$ , at distance  $f$  (focal length). Notice that as the camera pans, the projections of successive principal points onto the *original* extended image plane  $I$  form the ellipse ( $e_1$ )



**Fig. 2.** Camera Geometry [6]. (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).

centered at  $C$ . This can also be seen as the bounded case of intersection of a conic by an oblique plane  $I$ . Similarly, the locus of an arbitrary point  $P$  on the image plane  $I$  as the camera pans is also an ellipse ( $e_2$ ). We now wish to compute the change in pan ( $\delta\theta$ ) and the change in tilt ( $\delta\phi$ ) that would center the point  $P$  (after panning and tilting the camera), given the camera's current ( $pan, tilt$ ) location and  $P$ 's image coordinates  $(x, y)$ .

From Fig. 2(b), the desired change in pan  $\delta\theta$  is  $\angle PMQ$ , which is the angle between planes  $MPC$  and  $MQC$  and is computed using

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

Therefore, the change in pan is essentially a function of the image coordinates  $(x, y)$ , the focal length ( $f$ ), and the current tilt ( $\phi$ ) of the camera.

Further from Fig. 2(a), in order to calculate the change in tilt  $\delta\phi$ , we need to compute  $\angle P'OR$ . In terms of the parametric equation of an ellipse, we can represent point  $E$  as  $(a \cos(t), b \sin(t))$ . Also note that 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  $e_2$ ,

from Fig. 2(a) we see that

$$A \cdot \cos(t) = y + a \quad (2)$$

$$B \cdot \sin(t) = x \quad (3)$$

Using the relation  $\frac{A}{B} = \frac{a}{b}$ , we compute the value of  $t$  as

$$t = \tan^{-1} \left( \frac{a}{b} \cdot \frac{x}{y+a} \right) \quad (4)$$

Therefore, using Eqn.(2) and Eqn.(4), we get

$$A = \frac{y+a}{\cos \left( \tan^{-1} \left( \frac{a}{b} \cdot \frac{x}{y+a} \right) \right)} \quad (5)$$

This gives the required angle  $\angle P'OR$  ( $\delta\phi$ ) as

$$\delta\phi = \tan^{-1} \left( \frac{A-a}{f} \right) \quad (6)$$

Again, we need only the  $(x, y)$ , the tilt ( $\phi$ ), and the focal length  $f$  to compute the required change in tilt  $\delta\phi$ .

As seen from Eqn.(1) and Eqn.(6), the changes in pan and tilt depend on the current tilt ( $\phi$ ) and the pixel location  $(x, y)$ . Therefore, a simple linear mapping of  $(x, y)$  to  $(\theta, \phi)$  based only on the proportional change in the sensor field-of-view is insufficient.

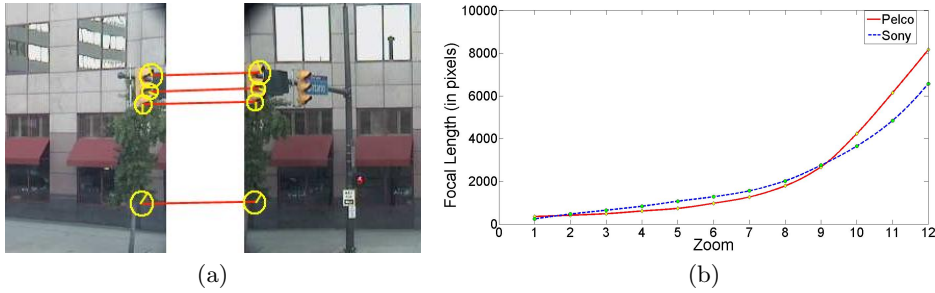
### 3.2 Automatic Calculation of Focal Length

For this model, the focal length of the PTZ camera  $f$  (in pixels) is needed. Moving beyond the manual method of [6], we propose an automatic technique to calculate  $f$  based on SIFT keypoint matching across adjacent overlapping images and using RANSAC for elimination of outliers.

We begin by pointing the camera to two locations which have overlapping coverage and whose pan-tilt locations are  $[(\theta_1, \phi_1)$  and  $(\theta_2, \phi_2)]$ . We then match SIFT keypoints [13] in the image overlap area (see Fig. 3(a)). For each such matched keypoint in the first image, we have Eqn.(1) to compute its change in pan ( $\delta\theta_1$ ) from  $\theta_1$  (in terms of  $f$ ). For each corresponding matched keypoint in the second image, we again have Eqn.(1) to compute its change in pan ( $\delta\theta_2$ ) from  $\theta_2$  (also in terms of  $f$ ). Since we know the actual pan locations of the center of the two images ( $\theta_1$  and  $\theta_2$ ), we can enforce the constraint (with  $\theta_2 > \theta_1$ ) that

$$\delta\theta_1 + |\delta\theta_2| = \theta_2 - \theta_1 \quad (7)$$

This gives us an equation from which  $f$  can be solved. We repeat the above technique for other such pairs of images obtained by moving the camera to different overlapping pan-tilt locations. However, since the SIFT keypoint finding and matching may not be perfect, we use the well known RANSAC algorithm [14] to perform a robust estimation of the focal length and eliminate outliers using multiple matched keypoints in the images. This process allows us to automatically calculate a robust estimate of the focal length.



**Fig. 3.** (a) SIFT keypoints matched across overlapping regions of an adjacent pair of images. (b) Variation of focal length with zoom for Sony and Pelco cameras.

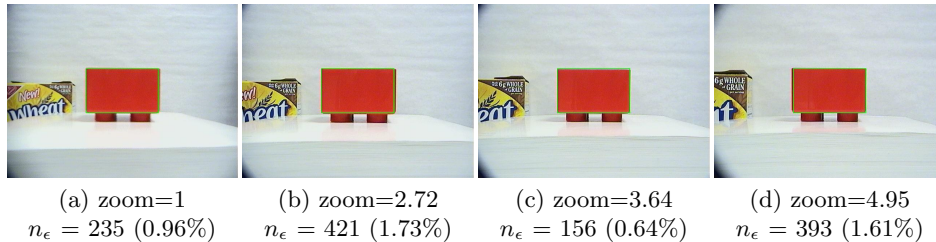
### 3.3 Variation with Zoom

A key component of the mapping model is the PTZ camera’s focal length and in order to generalize the model to all zoom levels, we need to know the interrelationship between focal length ( $f$ ) and zoom ( $z$ ). Therefore, we next model the effect of zoom and analyze the variation of  $f$  with  $z$ .

We perform the automatic focal length calculation described in Sect. 3.2 at different zoom levels of the cameras. Here, we choose the range of zoom=1 (wide field-of-view) to zoom=12 as the upper limit of zoom=12 has been observed to be more than sufficient for standard surveillance applications. Once the focal length values for all the zoom levels are calculated, we compute a piecewise cubic fit to this data. Figure 3(b) shows the variation of focal length with zoom for two different brands of PTZ cameras. This function permits the use of an arbitrary zoom level in our model by mapping any zoom value to its corresponding focal length so that it can then be utilized in the  $(x, y)$  to  $(pan, tilt)$  mapping.

To determine the smoothness and accuracy of the learned  $f$ - $z$  function, we employ an object size preservation metric. A uniform colored rectangular object is placed in the center of the camera’s view and the object’s motion is restricted to be along the line-of-sight of the camera. Keeping the camera’s pan-tilt  $(\theta, \phi)$  fixed, the camera’s zoom is automatically adjusted so that the size of the object at any instant during its motion remains the same as in the first frame ( $w \times h$  pixels). To do this, the object’s upper-right corner point  $(x, y)$  is detected at every frame (using for example, Canny edge detection) and the point’s difference in pan ( $\delta\theta$ ) from the home pan ( $\theta$ ) is calculated using Eqn.(1). Since the size of the object is to remain the same and the object is always centered, the locations of all of its corners should also remain constant  $(\pm w/2, \pm h/2)$ . Therefore, in order to scale the location of the upper-right corner point from  $(x, y)$  back to  $(w/2, h/2)$  by adjusting the zoom, the new focal length needed for the camera is calculated using the values  $\delta\theta$ ,  $\phi$ ,  $w/2$ , and  $h/2$  in Eqn.(1) and then reformulated to solve for  $f$ . This focal length value is now mapped to a corresponding zoom level using the  $f$ - $z$  function, thereby adjusting the zoom to this new value.

To demonstrate this concept, Fig. 4 shows a red block remaining fairly constant in size as the block is moved forward and the zoom level is adjusted automatically for the PTZ camera. At each of these levels the number of error pixels ( $n_\epsilon$ ) is obtained by finding the number of red pixels outside the target box and the number of non-red pixels inside the target box using color segmentation. Figure 4 also presents the low number of error pixels (and percentage error) at different zoom levels. Note that since the block is moved manually, there may be some minimal translation involved. The values obtained for the error pixels can be used to evaluate the precision of the mapping with variations in zoom for particular application needs.



**Fig. 4.** Size of the red block in the image (24,400 pixels) remains almost constant due to the automatic zoom update. Note the increase in size of the background object.

## 4 Experiments

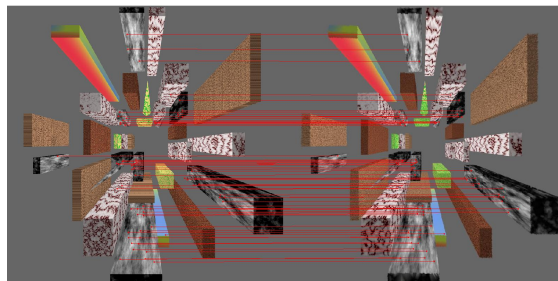
In order to evaluate the complete PTZ mapping model proposed in this work, we performed multiple experiments using synthetic data and real data captured from commercial off-the-shelf PTZ cameras. All the experiments covered in this section are important contributions of this work (beyond [6]) because they comprehensively tested the model both qualitatively and quantitatively in the following way. The first set of experiments examined the accuracy of the model by mapping individual pixels from images across the entire scene to their corresponding pan-tilt locations to generate spherical panoramas. We then performed experiments using synthetic data to quantify the accuracy of the mapping model. We next studied the pan-tilt deviations of the cameras and their effects on the mapping model. We then demonstrate the validity of the model across various zoom levels. The final experiment demonstrates the complete pan-tilt-zoom capabilities of the model within a real-time active tracking system.

All the experiments involving real data were performed using two models of commercial security PTZ cameras: 1) Pelco Spectra III SE surveillance camera (analog) and 2) Sony IPELA SNC-RX550N camera (digital). The accuracy of pan-tilt recall for these cameras were evaluated by moving the camera from arbitrary locations to a fixed home pan-tilt location and then matching SIFT feature

points across the collected home location images. The mean ( $\mu$ ) and standard deviation ( $\sigma$ ) of the pan-tilt recall error (in pixels) measured for the Sony and Pelco cameras were (2.231, 0.325) and (2.615, 0.458) respectively. Similarly, we also calculated the mean and variance of the zoom recall error for the Pelco and Sony cameras as (0.0954, 0.0143) and (0.0711, 0.0082) respectively, by changing the zoom from arbitrary zoom levels to a fixed home zoom level and then polling the camera’s zoom value (repeated for multiple home zoom levels). The radial distortion parameter ( $\kappa$ ) values obtained for the Sony and Pelco cameras were 0.017 and 0.043 respectively, determined using the technique from [15] [16]. These values can be used by the reader to evaluate the quality of the cameras used in the experiments.

#### 4.1 Panorama generation

PTZ cameras generally have a large pan-tilt range (e.g.,  $360 \times 90$  degrees). By orienting the camera to an automatically calculated set of overlapping pan-tilt locations, we collected a set of images such that they cover the complete view space. The number of images required for complete coverage vary with the zoom level employed, a study of which follows later. We then took images which are pairwise adjacent from this set and used them to automatically calculate a robust estimate of the focal length  $f$  (using the technique described in Sect. 3.2). Next, we mapped each pixel of every image to its corresponding pan-tilt orientation in the  $360 \times 90$  degree 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 sweep angle represents the pan. Bilinear interpolation was used to fill any missing pixels in the final panorama. The result is a spherical (linear fisheye) panorama which displays the entire pan-tilt coverage of the camera in a single image.



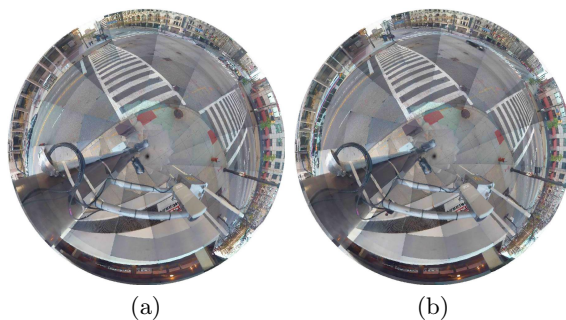
(a) Ground truth (b) Constructed panorama

**Fig. 5.** SIFT keypoint matching quantifying the accuracy of the mapping model.

To quantitatively evaluate the accuracy of the above technique, we performed an experiment using a virtual environment synthetically generated in Maya.



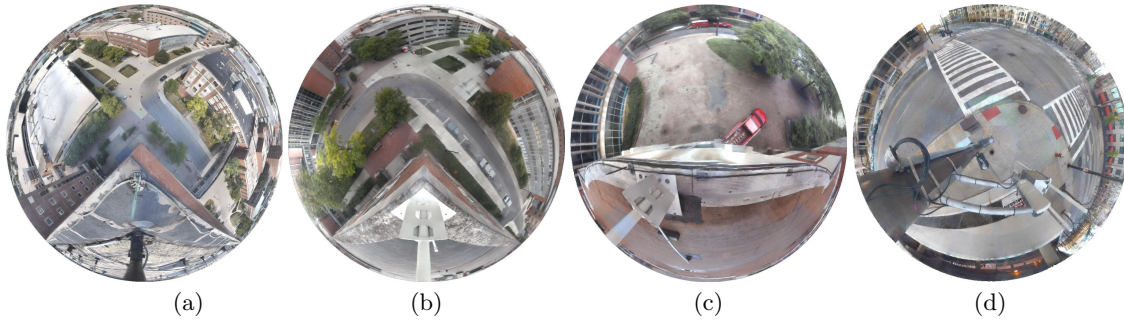
The ground truth data (Fig. 5(a)) was obtained by setting up a camera with a wide field-of-view<sup>1</sup> directly above the environment so that it captures the entire field-of-coverage. We then generated a panorama (Fig. 5(b)) using the synthetic data captured from different pan-tilt locations using our technique. By using SIFT to match keypoints between the ground truth image and the generated panorama, we calculated the distance between the matched locations for the top 100 keypoints. The mean and standard deviation values were obtained as 7.573 pixels and 0.862 pixels respectively. For images of size  $900 \times 900$ , we believe these values are reasonably small and within the error range of SIFT matching since the interpolation does not reproduce the fine textures very accurately in the scale space. These low values for the matching distance statistics quantitatively demonstrate the accuracy of the panorama generation technique.



**Fig. 6.** Panoramas for different shifts show illumination differences before intrinsic image generation.

While performing the above experiments using real cameras, the automatic gain control (AGC) on these cameras could not be fully disabled as the camera moved to new locations. Consequently there were illumination gain differences among the different component images captured across the scene and these differences were reflected in the panorama. Therefore, we used a technique of deriving intrinsic images [17] and computed a reflectance-only panoramic image to overcome this problem. We generated multiple panoramas of the same scene (see Fig. 6) by shifting the pre-configured pan and tilt coverage locations in multiples of 5 and 1 degree intervals respectively. These small shifts in pan and tilt displace the location of the illumination seams in the panoramas across multiple passes, thus simulating the appearance of varying shadows. We then separated the luminance and chrominance channels from the RGB panoramas by changing the color space from RGB to YIQ. After running the intrinsic image algorithm on the Y (luminance) channel alone, we combined the result with the mean I

<sup>1</sup> Maya camera parameters: Film Gate-Imax, Film aspect ratio-1.33, Lens squeeze ratio-1, camera scale-1, focal length-2.5.



**Fig. 7.** Panoramas from Pelco (a-c) and Sony (d) cameras mounted on buildings at different heights.

and Q channels of the panorama set, and converted back from YIQ to the RGB color space for display. As shown in Fig. 7(d), the result provides a more uniform panorama as compared to Fig. 6.

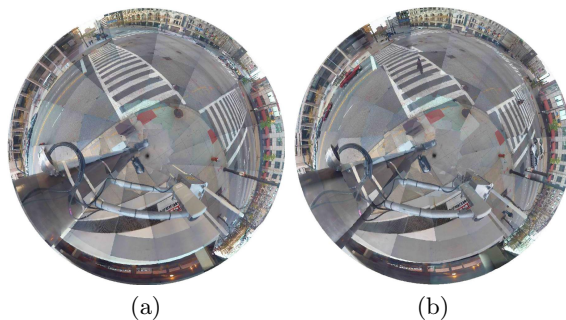
The above process was tested on multiple Pelco cameras mounted at varying heights (on 2, 4, and 8 story buildings) and a Sony camera. The final results after intrinsic image generation are shown in Fig. 7, demonstrating the applicability of the model to real surveillance cameras at different heights. *See high resolution version of Fig. 7(a) in supplementary material.*

## 4.2 Pan-Tilt Recall

In order to study the effect of inaccuracies in the pan-tilt motor on the proposed model, we simulated varying degrees of error in the data capturing process and then compared the resulting panoramas. This was performed by introducing either a positive or negative (randomly determined) fixed shift/error  $\epsilon$  in the pan and tilt values at each of the pre-configured pan-tilt locations and then capturing the images. This data was then used to generate a spherical panorama using the above mentioned technique. We repeated this process to generate panoramas for different values of simulated pan-tilt shift/error  $\epsilon$  in the range 0.1 to 5 in steps of 0.1 degrees. Figure 8 shows the results of this experiment for  $\epsilon$  values of 1.5 and 2.5 (before intrinsic analysis). The alignment errors for panoramas with  $\epsilon$  values less than 1.5 degrees were visually negligible, but were exacerbated for  $\epsilon$  greater than this value.

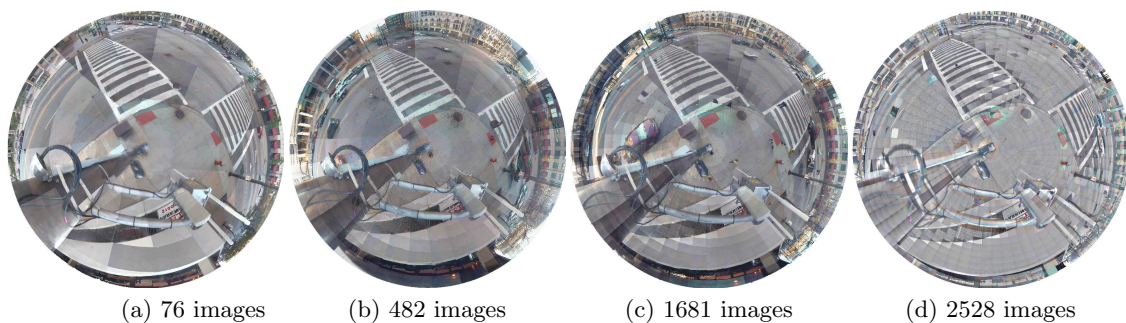
## 4.3 Validity Across Zoom levels

The next set of experiments were aimed towards verifying that the model works at varying zoom levels. We performed the panorama generation process at different zoom levels in the range of zoom=1 to zoom=12. At each zoom level, the appropriate focal length was chosen by employing the  $f$ - $z$  mapping function



**Fig. 8.** Panoramas with simulated pan-tilt error  $\epsilon$  values of (a) 1.5 and (b) 2.5. Note the alignment problems in the panorama with  $\epsilon=2.5$ .

learned using the technique described in Sect. 3.3. As the zoom level increases, the field-of-view of each component image reduces and consequently the number of component images required to cover the complete pan-tilt space increases. Figure 9 shows representative panoramas (before intrinsic analysis) generated at zoom levels 1, 5, 10, and 12 for the Sony camera (and presents the number of component images needed to construct the panoramas). In each of the panoramas, it was observed that the small component images registered correctly and no noticeable misalignments were seen with any change in zoom. This demonstrates that the  $(x, y)$  to  $(pan, tilt)$  mapping model works irrespective of the zoom level employed.



**Fig. 9.** (a)-(d) Panoramas (pre-intrinsic) generated at zoom levels 1, 5, 10, and 12 respectively with their corresponding number of component images.

#### 4.4 Use in Tracking with Active PTZ Cameras

Combining the pan, tilt, and zoom mapping capabilities of the proposed model, we developed a real-time active camera tracking application with a wide-area PTZ surveillance camera. The active camera system continually follows the target as it moves throughout the scene by controlling the pan-tilt of the camera to keep the target centered in its view. In addition, the camera’s zoom is continually adjusted so that the target being tracked is always of constant size irrespective of target’s distance from the camera.

We used the appearance-based tracking algorithm from [18] to demonstrate the applicability of our PTZ model. Note that our model is indifferent to the tracker itself and various other frame-to-frame tracking algorithms could be employed to handle strong clutter and occlusion situations if needed.

In the system, we model the target using a covariance of features and matched it in successive frames to track the target. To build the appearance-based model of the target, we selected a vector of position, color, and gradient features  $f_k = [x \ y \ R \ G \ B \ I_x \ I_y]$ . The target window was represented as a covariance matrix of these features. Since in our application, we desire to track targets over long distances across the scene, the appearance of targets undergo considerable change. To adapt to this and to overcome noise, a model update method from [18] was used.

The tracker is initialized manually by placing a box on the torso of the target. As the target moves, the best match  $(x_{match}, y_{match})$  is found by the tracker in successive frames and its distance  $d_{match}$  is stored. In this system, we restricted our search for the best match to a local search window of  $40 \times 40$  (in frames of size  $320 \times 240$ ). Using the proposed camera model, the  $(x_{match}, y_{match})$  coordinates of the best matching patch are converted to the change in pan and tilt angles required to re-center the camera to the target. The tracker then checks to determine if an update in zoom is necessary. To do so, it checks the distances of a larger patch and a smaller patch at  $(x_{match}, y_{match})$ . If either of these distances are found to be better than  $d_{match}$ , the zoom is adjusted to the new size using the same technique as described in the red block experiment in Sect. 4.4. While moving the camera, the system polls the motor to see if it has reached the desired pan-tilt location. Once it has reached this position, the next frame is grabbed and the best match is again found. This process is repeated to continually track the target.

Figure 10 shows a few frames from two sequences with a target being automatically tracked using the proposed technique. In the first sequence, the target walks a total distance of approximately 400 feet away from the camera and the zoom level varies from 2 (in the first frame) to 5.12 (in the last frame). In the second sequence, the target covers a distance of around 650 feet along a curved path and the camera’s zoom varies in the range from 3 to 6.77 during this period. *See supplemental tracking movie.* The ground truth locations of the target were obtained by manually marking the left, right, top, and bottom extents of the target’s torso and calculating its center in each frame. By comparing these with the center of the tracking box (of size  $25 \times 50$ ), the tracking error statistics (mean,



**Fig. 10.** Active camera tracking results from two sequences.

standard deviation) in pixels were obtained as  $(2.77, 1.50)$  and  $(3.23, 1.63)$  respectively for the two sequences. In addition, the accuracy of the zoom update function was evaluated by comparing the ground truth height of the target’s torso with the height of the tracking box. The error in this case was obtained to be 6.23% and 8.29% for the two sequences. This application demonstrates a comprehensive exploitation of the pan-tilt-zoom capabilities of the camera using the proposed model. Again, other trackers could also be employed with our PTZ model for different tracking scenarios/requirements.

## 5 Key Contributions and Advantages

To summarize the key contributions of this work, we presented 1) a theoretical derivation of a complete pan-tilt-zoom mapping model for real-time mapping of image  $x-y$  to pan-tilt orientations, 2) a SIFT-based technique to automatically learn the focal length for commercial-grade PTZ cameras, 3) an analysis of the variations between focal length and zoom in the model along with its evaluation using an object size preservation metric, 4) quantitative experiments and discussion on the PTZ mapping model with panorama generation, 5) experiments demonstrating robustness with different brands of cameras, and 6) qualitative and quantitative experiments within an example PTZ active tracking application.

The proposed camera model is a straight-forward and closed-form technique. Since previous work in this area have been based on more complex methods (explained in Sect. 2), by showing our technique with thorough and robust results, we demonstrate that any additional modeling and computation (as shown in previous work) is in fact not necessary for the demonstrated capabilities. We believe our simplification compared to other over-complicated methods is a strength, as this makes our technique more practically applicable to a wide variety of real-time problems.

## 6 Conclusion and Future Work

We proposed a novel camera model to map the  $x$ - $y$  focal plane of a PTZ camera to its pan-tilt space. The model is based on the observation that the locus of a point on a fixed image plane, as the camera pans, is an ellipse using which we solve for the desired change in pan and tilt angles needed to center the point. We proposed techniques to automatically calculate the focal length, analyzed the variation of focal length with zoom, tested the model by generating accurate panoramas, and validated its usability at varying zoom levels. In future work, we plan to utilize the model to map the camera's pan-tilt-zoom space to ground-plane information for registration of multiple cameras.

**Acknowledgement:** This research was supported in part by the National Science Foundation under grant No. 0236653.

## References

1. Davis, J., Chen, X.: Calibrating pan-tilt cameras in wide-area surveillance networks. In: Proc. ICCV. (2003)
2. Jain, A., Kopell, D., Wang, Y.: Using stationary-dynamic camera assemblies for wide-area video surveillance and selective attention. In: Proc. IEEE CVPR. (2006)
3. Bernardin, K., van de Camp, F., Stiefelwagen, R.: Automatic person detection and tracking using fuzzy controlled active cameras. In: Proc. IEEE CVPR. (2007)
4. Jethwa, M., Zisserman, A., Fitzgibbon, A.: Real-time panoramic mosaics and augmented reality. In: Proc. BMVC. (1998)
5. Tordoff, B., Murray, D.: Reactive control of zoom while fixating using perspective and affine cameras. IEEE TPAMI **26** (2004) 98–112
6. Sankaranarayanan, K., Davis, J.: An efficient active camera model for video surveillance. In: Proc. WACV. (2008)
7. Basu, A., Ravi, K.: Active camera calibration using pan, tilt and roll. IEEE Trans. Sys., Man and Cyber. **27** (1997) 559–566
8. Collins, R., Tsing, Y.: Calibration of an outdoor active camera system. In: Proc. IEEE CVPR. (1999) I: 528–534
9. Fry, S., Bichsel, M., Muller, P., Robert, D.: Tracking of flying insects using pan-tilt cameras. Journal of Neuroscience Methods **101** (2000) 59–67
10. Woo, D., Capson, D.: 3D visual tracking using a network of low-cost pan/tilt cameras. Canadian Conf. on Elec. and Comp. Engineering **2** (2000) 884–889
11. Barreto, J., Araujo, H.: A general framework for the selection of world coordinate systems in perspective and catadioptric imaging applications. IJCV **57** (2004)
12. Sinha, S., Pollefeys, M.: Pan-tilt-zoom camera calibration and high-resolution mosaic generation. Comp. Vis. and Image Understanding **103** (2006) 170 – 183
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
14. Fischler, M., Bolles, R.: Random sample consensus: a paradigm for model fitting. Communications of the ACM archive **24** (1981) 381–395
15. Tordoff, B., Murray, D.: The impact of radial distortion on the self-calibration of rotating cameras. (Comp. Vis. and Image Understanding)
16. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521623049 (2000)
17. Weiss, Y.: Deriving intrinsic images from image sequences. In: Proc. ICCV. (2001)
18. Porikli, F., Tuzel, O., Meer, P.: Covariance tracking using model update based on means on Riemannian manifolds. In: Proc. IEEE CVPR. (2006)