17^{th} *Computer Vision Winter Workshop*
Matej Kristan, Rok Mandeljc, Luka Čehovin (eds.)
Mala Nedelja, Slovenia, February 1-3, 2012

Unsupervised Calibration of Camera Networks and Virtual PTZ Cameras

Horst Possegger Matthias Rüther Sabine Sternig Thomas Mauthner
Manfred Klopschitz Peter M. Roth Horst Bischof
Institute for Computer Graphics and Vision
Graz University of Technology
{possegger,ruether,sternig,mauthner,klopschitz,pmroth,bischof}@icg.tugraz.at

Abstract. *Pan-Tilt-Zoom (PTZ) cameras are widely used in video surveillance tasks. In particular, they can be used in combination with static cameras to provide high resolution imagery of interesting events in a scene on demand. Nevertheless, PTZ cameras only provide a single trajectory at a time. Hence, engineering algorithms for common computer vision tasks, such as automatic calibration or tracking, for camera networks including PTZ cameras is difficult. Therefore, we propose a virtual PTZ (vPTZ) camera to simplify the algorithm development for such camera networks. The vPTZ camera is built on a cylindrical panoramic view of the scene and allows to reposition its field of view arbitrarily to provide several trajectories. Further, we propose an unsupervised extrinsic self-calibration method for a network of static cameras and PTZ cameras solely based on correspondences between tracks of a walking human. Our experimental results show that we can obtain accurate estimates of the extrinsic camera parameters in both, outdoor and indoor scenarios.*

1. Introduction

Video surveillance systems are used to simplify several tasks, such as security critical surveillance or sports game analysis. Although single static cameras can provide a wide field of view (FOV), the captured information is often insufficient to analyze dynamic crowded scenes, such as team sports, where the objects of interest will frequently occlude each other. To handle such scenarios, typically a network of several static cameras with overlapping fields of view is necessary. Furthermore, one often needs high quality imagery of particular events within the scene. This can be accomplished by combining the static cameras with PTZ cameras. This combination allows to simultaneously capture wide views of a scene with static cameras and additionally provides high resolution video feeds of specific parts of the scene with PTZ cameras.

PTZ cameras are very flexible, as they can observe an almost 360° horizontal field of view (HFOV) by repositioning the camera head. Nevertheless, while developing algorithms for common computer vision tasks, such as multiple object tracking, the use of a real PTZ camera is complex. In order to accurately test a vision algorithm which uses a PTZ camera, several trajectories of the PTZ camera need to be evaluated. However, the camera head cannot be repositioned off-line, which poses a problem for application developers. Thus, we propose a virtual PTZ (vPTZ) camera to replace a real dynamic camera during development. The vPTZ builds on a panoramic view of the scene created from the video stream of a spherical camera. This allows to simulate arbitrary PTZ trajectories on a single data set off-line.

Furthermore, we implement a method for self-calibration of the extrinsic parameters of cameras in a network of static cameras with overlapping FOVs and PTZ cameras. Calibrating a camera network requires point correspondences between several FOVs. In scenarios where the baseline between the fields of view is rather small, these correspondences may be found by extracting robust features, such as SIFT [12]. However, these approaches deteriorate if there are many repetitive structures or only weak key points, which are both common conditions in a realistic man-made environment. Moreover, cameras are typically placed at distant locations of a scene instead of ensuring a small baseline, which again cannot be handled by a local feature approach.

One possible solution to overcome these problems is to observe motion in the scene, *e.g.*, walking hu-

mans or driving cars. Point correspondences required for the self-calibration process can be obtained by extracting feature points from moving objects. In this way, even non-professional users are able to easily calibrate a network of multiple cameras.

Given synchronized video streams of static cameras and PTZ cameras, we propose a self-calibration method to obtain the extrinsic camera parameters based on detected head and foot locations of a walking human. We demonstrate the self-calibration method using the proposed vPTZ on both outdoor and indoor setups, consisting of three to four static cameras and one spherical camera capturing the panoramas used for the vPTZ. As image measurements often contain noise, we introduce an outlier removal method to obtain consistent point correspondences for the calibration process. Our experimental results show that after outlier removal, we obtain a robust estimate of the cameras' extrinsic parameters. We provide the evaluated data sets for further academic use, as well as an implementation of the vPTZ, which can be used to simulate trajectories of a real PTZ camera on the provided data sets.

2. Related Work

To the best of our knowledge, simulating a real PTZ camera using panoramic imagery of a scene has not been addressed before. A closely related concept is creating virtual camera planes which has been demonstrated in [14]. The authors extract perspective views from an omnidirectional vision system in order to remove distortions for interest region matching. Another related concept for simplifying the development of algorithms for multiple camera networks has been addressed in [20]. There, a synthetic camera network is placed inside a virtual scene. However, since modelling realistic human behavior within a virtual environment is a complex task, our approach benefits from using real video footage.

Self-calibration of both, static cameras (*e.g.*, [5, 17]) and PTZ cameras (*e.g.*, [3, 2, 9]) has been of significant interest in the past. Basically, when combining static and dynamic cameras, one faces the problem of controlling the PTZ camera within the camera network. This can be realized either by computing lookup tables between the positions of the PTZ and feature points in the FOVs of the static cameras, *e.g.*, [25, 4], or by calibrating the camera network in order to establish a geometric relation between pixel positions in the static views and the position of the PTZ camera, *e.g.*, [7]. Our system adopts to the latter approach.

Establishing geometric relations between the cameras in a surveillance network is often based on extracting local features, *e.g.*, [16]. However, given scenarios where the angles between the FOVs of the cameras are large or the static scene contains a lot of repetitive features, a static key point based approach is not applicable. One possible solution for such scenarios is to obtain correspondences by observing moving objects in the scene. In [22, 11], centroids of moving objects, tracked in at most three camera views, are used to calibrate the cameras and establish a common ground plane, respectively. As the centroids of the objects are above the ground plane, additional refinement steps are needed to estimate the common ground plane.

Another motion based self-calibration method has been demonstrated in [13], where head and foot locations of detected pedestrians are used to estimate vanishing points in the camera views. Since this approach is rather sensitive to noise, [10] shows that additionally incorporating a statistical model of human motion can improve the calibration results.

Very recently, Puwein *et al.* proposed a self-calibration method of a PTZ camera network based on foot point correspondences of players of a soccer game [19]. The extracted foot point trajectories are used to establish geometric constraints for the calibration process. Additionally, they include correspondences from detected field lines to improve the calibration results.

The use of foot locations of tracked humans has also been proposed in [8]. However, this approach is just semi-automated since the user has to define two bundles of coplanar parallel lines for each camera in order to recover the relative orientation of the camera to the ground.

Another method based on human detections has been demonstrated in [15], where the authors extract the head and foot locations by matching the silhouette of a standing human with silhouette imagery of a synthetic 3D model. This approach requires the human calibration target to stand still at a few positions within a camera's FOV.

If there is only little overlap between the cameras' fields of view, one has to compensate the lack of correspondences. One solution to this problem is to apply prior knowledge about the target's motion, *e.g.*, [21, 18].

3. Virtual PTZ

In order to simulate a PTZ camera off-line, we use a panoramic camera to capture the 360° horizontal view of the scene. In particular, we use a Point Grey Ladybug3 camera, which uses six 2 MP lenses to capture the required panoramas. We obtain a virtual PTZ camera by creating a virtual pinhole view and resampling its image plane using the panoramic imagery.

The concept of vPTZ from a panorama view can be considered as the inverse problem of image stitching. In the image stitching workflow, multiple images are projected onto a suitable panorama model, *e.g.*, a cylinder. Afterwards, the projected images have to be mapped from the model's surface onto a flat region in order to obtain a panorama image. On the other hand, given panoramic images of a scene, we can simulate a PTZ camera by combining a standard pinhole camera model and the panorama projection.

For a better understanding of the vPTZ camera, we first briefly summarize the creation of a cylindrical panorama according to [23, 24]. If the rotation of the camera which captures one part of the panorama is the identity, *i.e.*, the camera is in its canonical representation, the optical axis is aligned with the z-axis and the y-axis is aligned vertically. We denote the camera in the canonical representation as c. The 3D ray from the optical center to the pixel $(x_c, y_c)^\top$ on the panorama tile is thus $(x_c, y_c, f_c)^\top$, where f_c is the focal length of the camera. Points on the cylinder are given by an angle θ and a height h. The corresponding 3D cylindrical coordinates are thus $(\sin\theta, h, \cos\theta)^\top \propto (x_c, y_c, f_c)^\top$, as shown in Figure 1(a). The point $(x_c, y_c)^\top$ on a panorama tile can be mapped to the pixel position $(x', y')^\top$ of the panorama image by

$$x' = s\theta = s \operatorname{atan} \frac{x_c}{f_c} \tag{1}$$

$$y' = sh = s \frac{y_c}{\sqrt{x_c^2 + f_c^2}}, \tag{2}$$

where s is a scaling factor depending on the desired resolution of the panorama. Moreover, s can also be used to minimize the distortion near the center of the image by choosing $s = f_c$.

Given a cylindrical panorama as above, we can build a virtual PTZ camera by resampling a virtual pinhole view. The pinhole camera is placed such that its optical center coincides with the center of the cylinder and thus renders the view of the virtual PTZ. To obtain the imagery of the vPTZ, we have to compute a ray to each pixel on the image plane of the virtual PTZ and intersect it with the cylindrical surface. A ray r from the optical center of the vPTZ to a pixel $(u, v)^\top$ on the virtual image plane is given as $(u, v, f_v)^\top$, where f_v is the focal length of the virtual PTZ, which controls zooming. To account for the pan and tilt angles (λ and φ) of the vPTZ, r has to be transformed such that the vPTZ is in its canonical representation, *i.e.*, its axes are aligned with the cylinder of the panorama projection. This is done by using the rotation matrix R, such that $\hat{r} = Rr$. The rotation matrix is computed as $R = R_y R_x$, where R_x, R_y are rotation matrices around the x and y axes, respectively, given as

$$R_x(\varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sin(\varphi) & \cos(\varphi) \end{pmatrix} \tag{3}$$

$$R_y(\lambda) = \begin{pmatrix} \cos(\lambda) & 0 & \sin(\lambda) \\ 0 & 1 & 0 \\ -\sin(\lambda) & 0 & \cos(\lambda) \end{pmatrix}. \tag{4}$$

The pixel value of the virtual image plane can be obtained by mapping the intersection of the ray $\hat{r}$ and the cylindrical surface onto the panorama image. This is realized by substituting $\hat{r}$ into Eq. (1) and (2). Figure 1 illustrates the scheme of the virtual PTZ and shows a sample view.

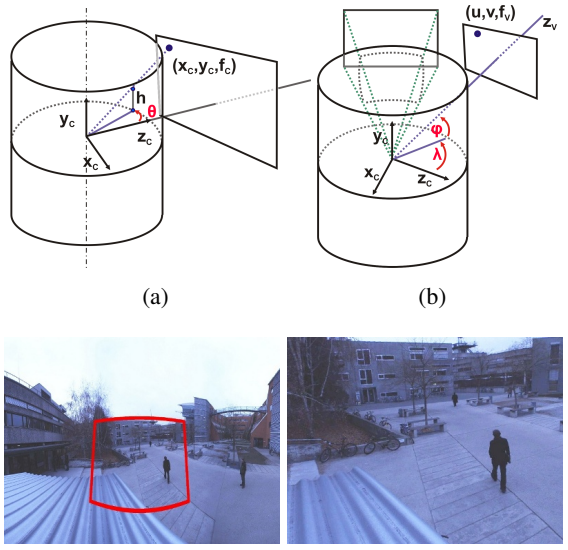
The zooming capability of the vPTZ is limited by the quality of the panoramic imagery. Thus, choosing a too large focal length results in undesirable artifacts. However, our experimental results show that even medium quality panoramas are sufficient to use the vPTZ as a simulation of a real PTZ camera.

4. Unsupervised Calibration Method

Our self-calibration method for a network of multiple static and vPTZ cameras consists of the following steps. First, we track a walking human throughout the scene and compute its foot and head locations for every camera in the network. Next, we remove outliers in the detected foot and head measurements to obtain clean measurements for the calibration step. Finally, we perform a modified bundle adjustment to estimate the extrinsic parameters of the cameras.

4.1. Head and Foot Point Localization

To obtain head and foot point measurements of a walking human over multiple frames of a video,



(a) (b)

(c) (d)

Figure 1. Concept of the virtual PTZ. Image stitching projects a panorama tile onto the cylinder (a). Aligning a virtual pinhole camera with the cylinder then allows to simulate a PTZ camera. Two virtual pinhole views are illustrated in Figure (b), as well as the optical axis z_v of one view with the corresponding pan and tilt angles. Figures (c) and (d) show a part of the cylindrical panorama with the superimposed FOV of the vPTZ and the corresponding view of the vPTZ camera, respectively.

we first segment and track the object of interest. In scenes with sparse motion, moving objects may be easily segmented by performing background subtraction. Considering simple environments, *i.e.*, scenarios where there are almost no shadows nor other noise which degrades the quality of the background model, the detected foreground blobs correspond perfectly to the human calibration target. To obtain a robust segmentation, we estimate two adaptive background models in parallel. One using the intensity values and one on the saturation channel of the imagery converted to the HSV color space. By combining these two models, we overcome the problems caused by penumbras, *i.e.*, soft shadows, and weak reflections of the pedestrian.

As simple background modelling would fail for more complex scenarios, *e.g.*, scenes with dense motion, hard shadows, heavy reflections, or occlusions of the object of interest, we validate the results of the segmentation by performing template matching. Therefore, we compare patches extracted around the detected foreground blobs with several templates showing a pedestrian in various poses. In order to follow the trajectory of the pedestrian, we use a blob

(a) (b) (c)

Figure 2. Pedestrian segmentation - background patches and corresponding real imagery with superimposed head and foot locations and axes of the first eigenvectors. (a), (b) Static camera views. (c) Segmented human in the vPTZ view.

tracking method based on spatial proximity and size similarity.

The head and foot locations of the segmented pedestrian can be determined by fitting an ellipse over the corresponding foreground blob. This is done by analyzing the eigenvectors of the covariance matrix of the blob. As we assume a standing or walking human calibration target, the first eigenvector will be aligned almost vertically. Thus, we consider the head position to be the topmost point of the foreground blob along this axis. We compute the head location by intersecting the first eigenvector axis with the top border of the minimum bounding rectangle of the detected blob. Similarly, we obtain the foot point by an intersection of the first eigenvector axis with the bottom border of the minimum bounding rectangle. Figure 2 illustrates head and foot locations calculated in this way.

It may be noted that real feet positions of a pedestrian are only aligned along the axis of the first eigenvector if the human is standing still, or its feet cross each other while walking. Thus, our denoted foot positions are in fact the projections of the human's center of mass along the first eigenvector axis, which provide consistent correspondences across all camera views.

4.2. Outlier Removal

Outliers in the measured head and foot locations can be removed by estimating pairwise homographies, *i.e.*, planar projective transformations, between the camera views based on the detected head and foot locations. A robust homography estimation can be obtained by using the well-known RANSAC algorithm and the normalized Direct Linear Transform [6]. Additionally to the homography, RANSAC returns a consensus set, containing all measurements which support the estimated transformation, *i.e.*, the so called inliers. Computing the intersection of the

consensus sets of the pairwise homographies, we obtain cleaned sets of image measurements which can be used for the self-calibration process.

To obtain clean measurements, we group the N cameras of the network into pairs $p_i = \{\text{camera}_i, \text{camera}_j\}_{i=1\ldots N, j\neq i}$. Next, we compute the pairwise homographies, which gives N sets of inliers. By intersecting these sets, we obtain the consensus set over all cameras, *i.e.*, the cleaned head and foot locations.

The maximum set of cleaned measurements can be obtained by computing the homographies between spatially neighboring cameras. Given additional information on the sequential arrangement of the cameras in the network, we can group the cameras clockwise, *i.e.*, assuming N cameras and a clockwise increasing labelling, we group the cameras into the pairs $p_i = \{\text{camera}_i, \text{camera}_{i+1 \bmod N}\}_{i=1\ldots N}$. Estimating the homographies and the resulting intersection set, we obtain the maximum number of cleaned head and foot locations, which slightly improves the results when used as input to our self-calibration method.

4.3. Camera Calibration

According to [6], the camera projection matrix P_j projects a 3D point X to a homogeneous point on the image plane of camera j. This projection matrix can be decomposed into the camera's intrinsic parameters, described by the upper triangular calibration matrix K_j, and its extrinsic parameters, described by a rotation matrix R_j and a translation vector t_j. Thus, the projection matrix is given as

$$P_j = K_j \left[R_j | t_j\right], \quad K_j = \begin{pmatrix} f_{j_x} & \gamma_j & p_{j_x} \\ 0 & f_{j_y} & p_{j_y} \\ 0 & 0 & 1 \end{pmatrix}, \quad (5)$$

where f_{j_x}, f_{j_y} are the focal length parameters in the x and y directions, γ_j denotes the skew, and $p_j = (p_{j_x}, p_{j_y})^\top$ is the principal point offset of camera j.

The projection of a 3D point by a normalized camera, *i.e.*, a camera where the intrinsic calibration is the identity, depends solely on the camera's extrinsic parameters. Thus, we can estimate the rotation and translation of a camera w.r.t. a global coordinate system from point correspondences if its intrinsic calibration is known. As the projection matrix is homogeneous, we need the homogeneous representations of the 3D foot and head points. These are given as $X = (a, b, 0, 1)^\top$ for 3D foot points, and $X = (a, b, h, 1)^\top$ for 3D head points, respectively, where (a, b) is the unknown 2D position at the corresponding plane, and h is the height of the detected human. Transforming a homogeneous 3D point X with the projection matrix $\hat{P}$ of a normalized camera gives the normalized coordinates $x = \hat{P}X$. Given the image measurements m_j of the head and foot points in camera j, as well as its intrinsic calibration, we obtain the normalized coordinates as $x_j = K_j^{-1} m_j$. These normalized measurements can be used to evaluate the estimated extrinsic calibration by computing the reprojection error $E^s = \|x_j - P_j X\|^2$, which vanishes if the correct solution has been found.

Although this pixel-based reprojection error can be easily computed for a static camera in the scene, it cannot be applied to a vPTZ camera directly, due to its variable parameters. However, as the vPTZ additionally provides the pan/tilt/zoom parameters (λ_n, φ_n, and f_n) at the time of capturing frame n, we can define an alternative reprojection error based on angular distances. Therefore, we assume that the optical center of the camera and the rotational center are identical, which holds for the vPTZ.

Given a 3D ray from the optical center of the vPTZ to the measured pixel location $m_n = (u_n, v_n)^\top$, we obtain the corresponding pan and tilt angles (ω and ψ) w.r.t. the virtual camera's coordinate system as

$$\omega = \lambda_n + \operatorname{atan} \frac{u_n}{f_n}, \quad \psi = \varphi_n + \operatorname{atan} \frac{v_n}{f_n}. \quad (6)$$

Additionally, we need to compute the pan and tilt angles (Ω and Ψ) of the corresponding 3D point $M_n = (a_n, b_n, z_n)^\top$, w.r.t. the camera's coordinate system. Using the currently estimated extrinsic parameters of the vPTZ, *i.e.*, the rotation R and the translation t, these can be computed as

$$(r_1, r_2, r_3)^\top = [R|t]\,(a_n, b_n, z_n, 1)^\top \quad (7)$$

$$\Omega = \operatorname{atan} \frac{r_1}{r_3}, \qquad \Psi = \operatorname{atan} \frac{r_2}{r_3}. \quad (8)$$

Thus, we can define the reprojection error for the vPTZ as $E^d = \|(\omega, \psi)^\top - (\Omega, \Psi)^\top\|^2$.

For a network with static cameras N_s and vPTZ cameras N_d, we can compute the extrinsic calibration of each camera by minimizing the reprojection errors as

$$\underset{R_j, t_j, a_i, b_i}{\arg\min} \sum_{j \in N_s, i} E^s_{i,j} + \sum_{j \in N_d, i} E^d_{i,j} \quad (9)$$

$$E^s_{i,j} = \|x_{i,j} - P_j X_i\|^2 \quad (10)$$

$$E^d_{i,j} = \|(\omega_{i,j}, \psi_{i,j})^\top - (\Omega_{i,j}, \Psi_{i,j})^\top\|^2. \quad (11)$$

We solve this non-linear least squares optimization problem using the iterative Levenberg-Marquardt (LM) algorithm. Similar to [1], our experimental results show that the LM algorithm provides sufficiently accurate results, even if initialized without any knowledge of the cameras' real positions. In our experiments, we always initialize the cameras 10 meters above the global 3D coordinate center such that the optical axes face in positive z-direction.

The calibration method for the proposed vPTZ camera can also be applied to real PTZ cameras, as long as they provide sufficiently accurate measurements of the current pan/tilt/zoom parameters. It may be noted, that our proposed method adopts the common geometric model, where the orthogonal rotation axes are aligned with the camera's imaging optics. The vPTZ camera follows this ideal model, whereas in general, the optical center of a real PTZ camera and its rotation axes are not aligned exactly. However, in typical surveillance scenarios, the resultant deviations are minimal w.r.t. to the scene's dimension and thus do not notably affect the accuracy of computer vision applications, such as tracking.

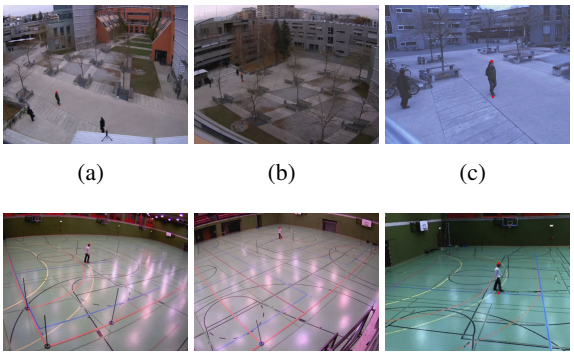
If calibrating a real PTZ camera as proposed, we have to consider the varying lens distortion, which depends on the current focal length. One solution is to estimate the lens distortion for several zoom levels, such that the distortion at arbitrary zoom levels can be interpolated from the available estimates. Thus, for an arbitrary focal length of the PTZ camera, we can compute the undistorted image measurements, which can be used to accurately estimate the extrinsic calibration using the proposed method.

It may be noted that the reprojection error for PTZ cameras assumes that all point correspondences lie on the same half-sphere w.r.t. the camera's optical center. However, by initially placing the camera far above the coordinate center for the LM algorithm, we ensure that all projected measurements lie on the same half-sphere throughout optimization.

5. Experimental Results

To evaluate the proposed self-calibration method, we use two data sets, one outdoor and one indoor. The camera networks in our experiments contain three (outdoor) to four (indoor) static Axis P1347 cameras, as well as one spherical Point Grey Ladybug3 camera. Both camera setups cover a large area of interest, as can be seen in Figure 3.

We estimate the cameras' intrinsic parameters for each camera setup using a publicly available toolbox[1]. Next, we use the intrinsic parameters to obtain the normalized coordinates of the measured head and foot locations. After outlier removal, the normalized measurements are used to estimate the extrinsic parameters of the cameras by solving Eq. (9). Since we have no prior knowledge of the real camera positions, we initially place all cameras 10 meters above the (unknown) global coordinate center, with the optical axes facing in positive z-direction. Thus, the initial position of each camera before LM optimization is $(0, 0, -10)$. After convergence, we obtain both, the positions and orientations of the cameras, as well as the estimated 2D coordinates $(a, b)^\top$ of the head and foot points in the corresponding plane. Figure 4 illustrates the computed setup of both data sets.



(a) (b) (c)

(d) (e) (f)

Figure 3. Sample views of the outdoor (top row) and indoor (bottom row) data sets with superimposed head and foot locations of the human calibration target (red points). (a),(b),(d),(e) Static camera views. (c),(f) Views of the vPTZ.

To evaluate our method, we project known points onto the ground plane using the estimated projection matrices. According to [6], the homography between the world plane at $z = 0$ and the image plane of camera j is given as $H = [r_1, r_2, t_j]$, where r_i is the i-th column of the camera rotation matrix R_j, and t_j is its translation. Thus, the pixel locations $(u, v)^\top$ of the imaged ground plane points can be projected onto the world plane at $z = 0$ as $(\hat{x}, \hat{y}, \hat{w})^\top = H^{-1}(u, v, 1)^\top$. The projected 2D point on the ground plane is given as $(\hat{x}/\hat{w}, \hat{y}/\hat{w})^\top$. Next, we compute distances between the known points and compare these with measurements of the real scenes. Table 1 lists the resultant deviations. As can be seen,

[1] J.-Y. Bouguet. Camera Calibration Toolbox for Matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/index.html`, accessed November 24, 2011.

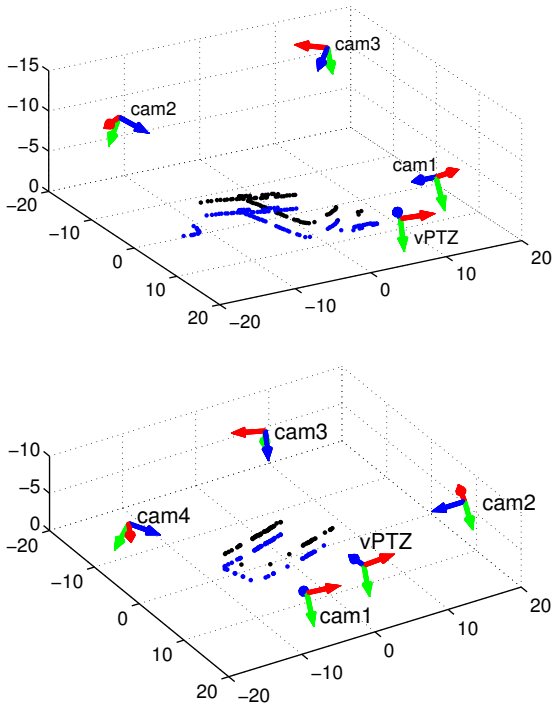


Figure 4. Estimated camera positions of the outdoor (top) and indoor (bottom) scene. Axes units are meters. Dots show sample head (black) and foot (blue) points used for calibration. As the optical axes initially face in positive direction, the resultant heights (z-axis) are negative.

Data set	Camera				
	1	2	3	4	vPTZ
Outdoor	0.015	0.019	0.072	n.a.	0.082
Indoor	0.139	0.107	0.159	0.083	0.210

Table 1. Mean reprojection errors in meters.

our self-calibration method provides sufficient accuracy for typical surveillance applications, such as tracking.

We provide both data sets[2] for further academic use. The package contains the video streams and extrinsic parameters of all cameras within the scene, the intrinsic calibration of the static cameras, and sample code to utilize the vPTZ. The implementation can be used to generate sample PTZ trajectories for the proposed data sets, such as illustrated in Figure 5.

6. Conclusions

We presented a virtual PTZ camera based on panoramic imagery of a real scene. The vPTZ can be used to evaluate different pan/tilt/zoom trajectories on the same data set on demand, whereas real PTZ cameras can only provide a single trajectory. Considering the task of developing applications for video surveillance systems with integrated PTZ cameras, the vPTZ can be used to simulate a real PTZ camera off-line.

Furthermore, we presented a method for unsupervised calibration of a network of static cameras and PTZ cameras with overlapping fields of view. Our self-calibration method is based on correspondences from tracks of a walking human. We presented how to extract head and foot locations of the pedestrian from videos, remove outliers within these measurements, and obtain the extrinsic parameters of the cameras by solving a non-linear optimization problem on the reprojection error. We demonstrated our method on both, indoor and outdoor data sets, using the proposed vPTZ camera. As our experimental results show, the detected feature points provide sufficient accuracy for calibrating a camera network which covers a large region of interest.

In order to obtain the complete PTZ calibration, we plan to investigate the effect of varying zoom levels and adopt the presented calibration method accordingly. We will focus our research on robust tracking algorithms of scenes with multiple objects of interest, such as sports games, where additional fields of view provided by PTZ cameras can be used to resolve ambiguities.

[2]Available online at `http://lrs.icg.tugraz.at/download.php#vptz`.

Acknowledgments

This work was supported by the Austrian Science Fund (FWF) under the project MASA (P22299) and by the Austrian Research Promotion Agency (FFG) under the project MobiTrick (8258408).

References

[1] M. Brown and D. Lowe. Unsupervised 3D Object Recognition and Reconstruction in Unordered Datasets. In *Proc. 3DIM*, 2005. 6

[2] J. Davis and X. Chen. Calibrating pan-tilt cameras in wide-area surveillance networks. In *Proc. ICCV*, 2003. 2

[3] L. de Agapito, R. Hartley, and E. Haymen. Linear Self-Calibration of a Rotating and Zooming Camera. In *Proc. CVPR*, 1999. 2

[4] A. Del Bimbo, F. Dini, A. Grifoni, and F. Pernici. Uncalibrated Framework for On-line Camera Cooperation to Acquire Human Head Imagery in Wide Areas. In *Proc. AVSS*, 2008. 2

[5] R. Hartley. Self-Calibration of Stationary Cameras. *IJCV*, 22(1), 1997. 2

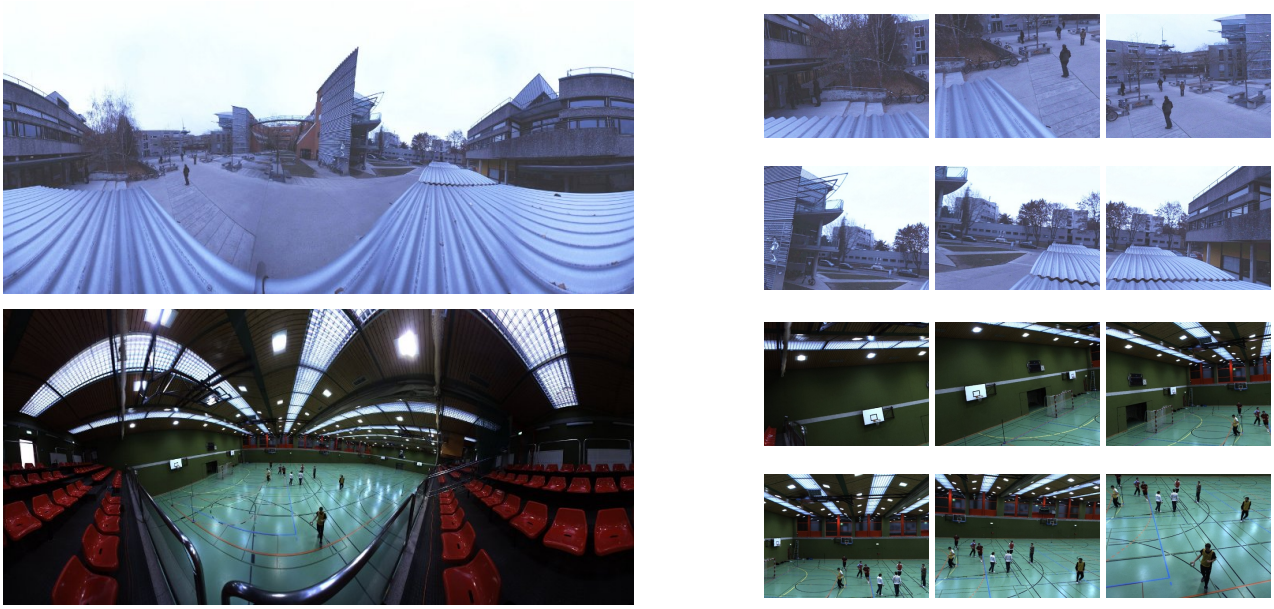


Figure 5. Sample panoramas (left) with different views of the vPTZ (right).

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. 4, 5, 6

[7] R. Horaud, D. Knossow, and M. Michaelis. Camera cooperation for achieving visual attention. *MVA*, 16(6), 2006. 2

[8] C. Jaynes. Multi-View Calibration from Planar Motion for Video Surveillance. In *Proc. VS*, 1999. 2

[9] I. Junejo and H. Foroosh. Refining PTZ camera calibration. In *Proc. ICPR*, 2008. 2

[10] N. Krahnstoever and P. Mendonça. Autocalibration from Tracks of Walking People. In *Proc. BMVC*, 2006. 2

[11] L. Lee, R. Romano, and G. Stein. Monitoring Activities from Multiple Video Streams: Establishing a Common Coordinate Frame. *PAMI*, 22(8), 2000. 2

[12] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2), 2004. 1

[13] F. Lv, T. Zhao, and R. Nevatia. Self-Calibration of a camera from video of a walking human. In *Proc. ICPR*, 2002. 2

[14] T. Mauthner, F. Fraundorfer, and H. Bischof. Region matching for omnidirectional images using virtual camera planes. In *Proc. CVWW*, 2006. 2

[15] B. Micusik and T. Pajdla. Simultaneous surveillance camera calibration and foot-head homology estimation from human detections. In *Proc. CVPR*, 2010. 2

[16] R. Mörzinger and M. Thaler. Establishing correspondence in distributed cameras by observing humans. In *Proc. ICDSC*, 2010. 2

[17] G. Nebehay and R. Pflugfelder. A self-calibration method for smart video cameras. In *Proc. ECVW*, 2009. 2

[18] R. Pflugfelder and H. Bischof. Localization and trajectory reconstruction in surveillance cameras with non-overlapping views. *PAMI*, 32(4), 2010. 2

[19] J. Puwein, R. Ziegler, L. Ballan, and M. Pollefeys. PTZ Camera Network Calibration from Moving People in Sports Broadcasts. In *Proc. WACV*, 2012. 2

[20] F. Qureshi and D. Terzopoulos. Surveillance Camera Scheduling: A Virtual Vision Approach. *Multimedia Systems*, 12(3), 2006. 2

[21] A. Rahimi, B. Dunagan, and T. Darrell. Simultaneous Calibration and Tracking with a Network of Non-Overlapping Sensors. In *Proc. CVPR*, 2004. 2

[22] G. Stein. Tracking from Multiple View Points: Self-calibration of Space and Time. In *Proc. DARPA IU Workshop*, 1998. 2

[23] R. Szeliski. Image Alignment and Stitching: A Tutorial. *FTCGV*, 2(1), 2006. 3

[24] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, first edition, 2010. 3

[25] X. Zhou, R. Collins, T. Kanade, and P. Metes. A Master-Slave System to Acquire Biometric Imagery of Humans at Distance. In *Proc. IWVS*, 2003. 2