Camera calibration and navigation in networks of rotating cameras

Adam Gudyś[1,2], Kamil Wereszczyński[1,2], Jakub Segen[1], Marek Kulbacki[1], and Aldona Drabik[1]

[1] Polish-Japanese Academy of Information Technology, Koszykowa 86, 02-008 Warsaw, Poland
{agudys, kw, js, mk}@pjatk.edu.pl
[2] Silesian University of Technology, Institute of Informatics, Akademicka 16, 44-100 Gliwice, Poland

Abstract. Camera calibration is one of the basic problems concerning intelligent video analysis in networks of multiple cameras with changeable pan and tilt (PT). Traditional calibration methods give satisfactory results, but are human labour intensive. In this paper we introduce a method of camera calibration and navigation based on continuous tracking, which requires minimal human involvement. After the initial precalibration, it allows the camera pose to be calculated recursively in real time on the basis of the current and previous camera images and the previous pose. The method is suitable if multiple coplanar points are shared between views from neighbouring cameras, which is often the case in the video surveillance systems.

1 Introduction

Current video surveillance systems are based on multiple cameras which can rotate and zoom (PTZ). Performing video analysis on data from multiple cameras usually requires the cameras to be calibrated with respect to a common world reference frame (WRF). Traditional camera calibration methods [10, 2] require placing 3D reference points or markers in the view of the cameras, which is labour intensive, especially in the case of exterior cameras. Therefore, there is a need for methods able to calibrate such cameras with a minimum human effort. Calibration is understood as a computation of a camera model M consisting of two components, a pinhole camera model P and a distortion model D. These two elements together make it possible to project any point from the world reference frame to the image plane as well as to reconstruct 3D objects on the basis of their images. An important limitation of many video systems is lack of the information about height of objects being observed–only planar two-dimensional map is available. This must be taken into account by the calibration algorithm.

A calibration system suitable for a network of rotating (PT) exterior cameras, that requires a minimum of human involvement, without placing markers in camera views, is introduced in this paper. Its extension to a network of PTZ

cameras will be the subject of future work. The calibration procedure in the system is divided into three stages: image preparation, i.e., undistortion and background separation (I), computation of intrinsic and extrinsic camera parameters (II), and navigation (III). The problem of distortion introduced by imperfect lens geometry is known to affect negatively accuracy of 3D reconstruction, thus different techniques for dealing with it like straight-lines [3] or radial trifocal tensors [9] were presented. The system employs the first approach which, in spite of its simplicity, turned out to render satisfactory results. The presence of background separation is caused by the fact that only fixed background points are utilised by the following stages. Therefore, moving objects (foreground) should be filtered out to prevent from disturbing the calibration. This is done with a method based on Gaussian mixture models [1, 11]. The next stage is the computation of intrinsic camera parameters with a use of a method suited for rotating cameras [6]. This is followed by the estimation of extrinsic camera parameters for a number of selected pan-tilt positions. This process is often performed with a use of correspondences between WRF and image points [5, 7], thus it requires human assistance. As only two-dimensional information about the observed scene is available, the algorithm suited for coplanar points was employed [8]. Finally the navigation stage begins which, after some preliminary steps, allows camera pose to be calculated in the real time on the basis of the current camera image.

The system has been implemented in C++ with a use of OpenCV library.

2 Image preparation

The camera model used most widely in a computer vision area is called a pinhole camera model. However, real-life cameras do not perfectly agree with this model, namely they introduce non-linear distortion. The distortion can be divided into two components: radial (along the direction from the center of the distortion to the considered point) and tangential (along the perpendicular direction), and can be written as an infinite series $x_u = x_d(1+k_1{r_d}^2+k_2{r_d}^4+\ldots)$, where (x_u, y_u) and (x_d, y_d) are undistorted and distorted image points, respectively, and r_d is a distorted radius.

It has been proven, however, that considering only first order distortion parameter k_1 gives sufficient accuracy [10]. Let us denote (c_x, c_y) as a center of distortion and s_x as a distortion aspect ratio, which may differ from the image aspect ratio. The undistorted coordinates are expressed as follows:

$$\begin{cases} x_u = x_d + (x_d - c_x)k_1{r_d}^2 \\ y_u = y_d + (y_d - c_y)k_1{r_d}^2 \end{cases}, \text{ where } r_d = \sqrt{\left(\frac{x_d - c_x}{s_x}\right)^2 + (y_d - c_y)^2}. \quad (1)$$

The algorithm used in the library [3] is based on a fact that in the perfect pinhole camera, projections of straight lines are also straight lines. Consequently, when the image is distorted, straight lines are seen as curves. The idea is to select on the image curves, which are projections of straight lines, and estimate distortion parameters in the way that after applying undistortion transformation

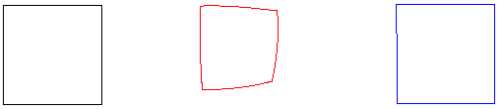

Fig. 1: Synthetic undistortion experiment. The black square on the left is used as a reference. The red square in the middle was obtained by distorting the reference with the following model: $k = 0.67$, $c_x = -0.38$, $c_y = -0.25$. The blue square on the right is a result of the undistortion procedure.

these curves become straight. For this aim a standard non-linear optimisation method can be applied [4].

The algorithm works in two stages. At first only k parameter is being estimated while c_x,c_y,s_x are fixed (c_x,c_y are set to the centre of the image while s_x is set to 1). After algorithm converges to some value of k, the minimisation procedure is executed once again with all distortion parameters being optimised. As s_x parameter models tangential distortion which has a moderate influence on the final image, it is fixed to 1 in both optimisation steps which may positively influence the algorithm convergence. As suggested by the results of synthetic (Fig. 1) and real-life (Fig. 2) experiments, the undistortion procedure renders satisfactory results.

The next step of the image preparation is background separation. The original method based on Gaussian mixture model [1] analyses consecutive frames and estimates background colour of a pixel on the basis of how frequently different colours are observed. In the time perspective, pixels form a cloud of points, each described by a time stamp and a pixel colour. These points can be gathered into K clusters, each representing a Gaussian component. If a colour not belonging to any cluster is observed, a new Gaussian component is created with a mean set to this colour, large covariance matrix and with a small weight.

To formalise this idea, let q_n be a specific pixel on frame n; $p_q(c_n)$—the probability of pixel q having colour c in frame n; w_k—counted weight of k-th distribution component; K—current number of Gaussian distribution components; T, α—algorithm parameters. In that case, $p_q(c_n) = \sum_{k=1}^{K} w_j \eta(c_n, \theta_k)$, where $\eta(c, \theta_k)$ is the normal distribution of k-th Gaussian component. It is defined as $\eta(c, \mu_k, \Sigma_k)$, with μ_k being the mean and $\Sigma_k = \sigma_k^2$ being the covariance of k-th component.

The distributions are sorted decreasingly according to their fitness defined as w_k/σ_k, and first B components are selected to be updated (B is computed on the basis of T). The update of distribution ω_k is done according to the equations:

$$\begin{cases} w_k^{n+1} &= (1-\alpha) w_k^n &+\alpha\ p(\omega_k|c_N) \\ \mu_k^{n+1} &= (1-\alpha) \mu_k^n &+\alpha\ \eta(c_{n+1}, \mu_k^n, \Sigma_k^n)\ c_{n+1} \\ \Sigma_k^{n+1} &= (1-\alpha) \Sigma_k^n &+\alpha\ \eta(c_{n+1}, \mu_k^n, \Sigma_k^n) \left(c_{n+1} - \mu_k^{n+1}\right)^2 \end{cases} \tag{2}$$

where $p(\omega_k|c_n)$ equals 1 if ω_k is the first match component to c_n and 0 otherwise.

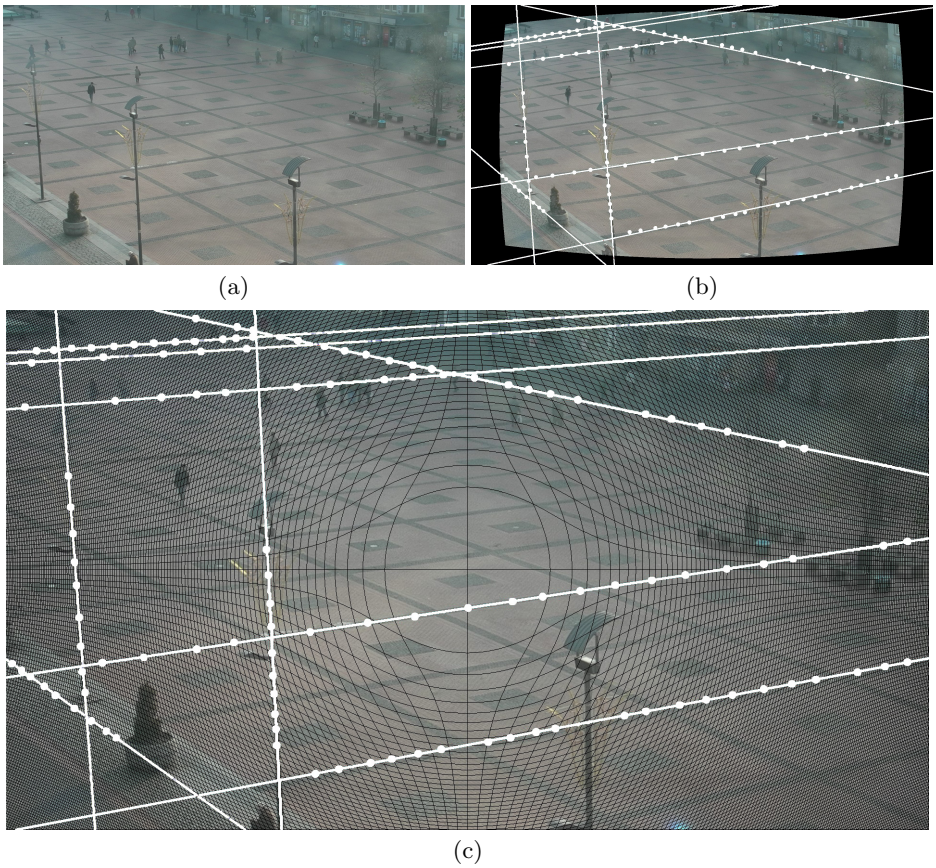

Fig. 2: Undistortion procedure on exemplary data. After taking the reference image (a), a synthetic distortion is applied on the image and several lines known to be straight in the real world are selected by the user (b). The undistortion procedure executed afterwards properly restores the original image (c). The radial grid is caused by some of output points not being covered by the undistortion transformation.

If none of the K distributions match pixel value c_n, the least probable component is replaced by a distribution mean $\mu_m^n = c_n$, an initially high variance, and a low weight parameter w_m^n. It was shown, that $\log_{1-\alpha} T$ frames are needed for satisfactory background separation, and the best values of parameters are $\alpha = 0.02$ and $T = 0.5$.

The problem concerning aforementioned approach is that the number of Gaussian components for all pixels is the same, which may render unsatisfactory results. In [11] a method of dealing with this issue known as adaptive Gaussian mixture model (AGMM) is presented. This variant adjusts the number of Gaussian components for each pixel independently and is used in the presented

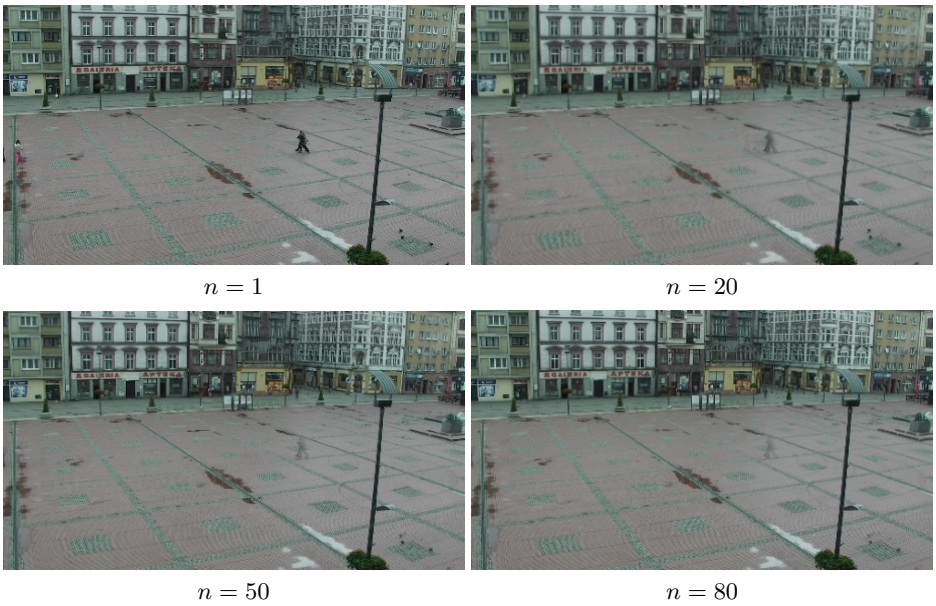

Fig. 3: Results of AGMM background separation step based on n images.

system. Results of performing background separation on the example sequence of frames are shown in Fig. 3.

3 Intrinsic and extrinsic camera parameters

Camera intrinsic parameters are: focal length f, skew s, coordinates of the principal point $P = (p_u, p_v)$, and scale factors in horizontal and vertical directions k_u, k_v. They are all gathered in K matrix.

$$K = \begin{bmatrix} fk_u & s & p_u \\ 0 & fk_v & p_v \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Extraction of these parameters is done according to [6]. The algorithm takes as an input a set of at least three overlapping images from rotating camera. No information about camera orientation during image acquisition is necessary. One of the images is selected as a reference one. The first step of the algorithm is to find transformations from the reference image to all the other ones. Let A, B and $P_{3\times3}$ be the reference image, the destination image, and the transformation from the reference to the destination, respectively. The computation of P is done as follows:

1. Find all key-points on the images A and B using feature detection algorithm, e.g. SIFT or SURF. Each key-point is described by a vector of descriptors.

2. Find correspondences between key-points on A and B using Euclidean distances in a descriptor space.
3. Pick M best pairs of corresponding key-points to calculate transformation from A to B using RANSAC [5]. Points should uniformly cover the overlap.

In order to check correctness of this step, a part of key-point pairs were used for testing. This allows pixel distances between points from A transformed by P and corresponding points from B to be compared. The intuitive way to verify whether P has been properly calculated is to transform A by P and draw it on B. As confirmed by the experiments both views fit to each other producing proper panorama. This operation is referred to as *image stitching*.

Having set of transformations from the reference image to the other images, the algorithm calculates matrix K of intrinsic camera parameters. For each transformation P, one can write the following equation based on the orthogonality of the rotation matrix:

$$(KK^T)P^{-T} = P(KK^T). \tag{4}$$

KK^T is a positive definite symmetric matrix with 6 degrees of freedom:

$$KK^T = \begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix}. \tag{5}$$

Eq. 4 can be rephrased as a homogeneous system $AX = 0$ of nine linear equations with six unknowns $X = [a\ b\ c\ d\ e\ f]^T$. Elements of A can be computed straightforwardly. However, as equations in such system are not independent, at least two different P transformations are required to calculate X, which corresponds to three overlapping images. When KK^T elements are found (this can be done to the linear factor as KK^T appears on both sides of Eq. 4), matrix is normalised in the way that f is equal to 1. Then KK^T is decomposed.

The accuracy of the procedure was confirmed by simulated experiments. A set Q of four synthetic points in a world reference frame (WRF) was created. Extrinsic parameters were set in the way that the points from Q were visible by the camera. Additionally, non-trivial intrinsic camera parameters were assumed. The points from Q were projected into the image plane resulting in a reference view. Then the camera was rotated slightly in both, vertical and horizontal directions and images of Q were taken at each pose (resulting in 8 additional views overlapping with the reference one). For each pair of overlapping views a transformation was calculated (images of points from Q were used as key-points). Set of transformations was given as an input for Hartley et al.'s algorithm. Estimated intrinsic parameters were compared to the ones used for data generation proving that algorithm works properly. The whole procedure was successfully repeated for many matrices of intrinsic parameters.

The next step after calculating K matrix is to compute camera pose in a world reference frame for several selected pan-tilt positions. The pose consists of two components: translation vector $T_{3\times1}$ and rotation matrix $R_{3\times3}$. Calculations are based on the correspondence between coordinates of scene objects and their images. An important requirement to be met is that method must work

Fig. 4: Calibration results on exemplary dataset. Blue points were directly selected by the user. White points are world points (selected on the map) projected on the image plane using calculated intrinsic and extrinsic camera parameters.

correctly for coplanar scene objects. This is motivated by the fact that no height information is available.

The algorithm employed by the system is a POSIT strategy suited for coplanar points [8]. The method approximates perspective projection by a scaled orthographic projection and iteratively refines camera extrinsic parameters in order to minimise projection error. In the case of non-coplanar points, the algorithm returns a single pose. In the coplanar case, there are two poses returned, giving user the opportunity to choose the more appropriate one.

In order to test extrinsic camera calibration the same experimental scheme was used as previously. Several synthetic points were placed in WRF and images of these points were taken using assumed intrinsic and extrinsic camera parameters. Then the points from WRF as well as their images were given to Oberkampf et al.'s algorithm as an input. Estimated pose was compared with the one used for data generation. The procedure was repeated for both non-coplanar and coplanar WRF points under wide range of observation angles. In all cases the algorithm successfully estimated real camera pose.

The computation of both intrinsic and extrinsic camera parameters was further tested by experiments on real-life camera images. At the beginning, a number of overlapping images from Market Square in Bytom were acquired and undistorted with a use of straight-line approach. Then, they were put to Hartley's algorithm to extract intrinsic parameters. Next, a set of corresponding points was selected manually on the camera images and the 2D map of Market Square from GoogleMaps which was followed by the calculation of a camera

pose. Finally, the map points were projected with a use of intrinsic and extrinsic camera parameters on the image plane and compared with the selected image points (Fig. 4).

4 Navigation: continuous tracking

Navigation is a process of real-time camera calibration which recalculates pose without human interference immediately after camera changes its pan-tilt orientation. In the presented study the navigation is performed with a use of a method referred to as a continuous tracking. It is assumed that the camera zoom during a single continuous navigation process is constant.

The navigation is preceded by the preparatory phase consisting in a generation of key-point clouds, each cloud corresponding to some camera orientation in a PT space. The key-point is defined as a pair of corresponding points in the image and in the WRF. The procedure of clouds generation is presented in Pseudocode 1.

The first key-point cloud is built upon an arbitrarily chosen reference image (Fig. 5-top-left). At the beginning a set of points in the image and corresponding points in the WRF (the map in the middle of Fig. 5) is selected manually by the user (step 2). These points are referred to as known points and are represented in Fig. 5 by green dots. Afterwards, point correspondences are used to calculate a homography between the world and the image space. This homography together with the intrinsic camera parameters allows initial pose to be computed (step 3) and is further used to increase the density of the current cloud (step 4). Namely, a set of additional feature points is found on the image with a use of SURF detector and is mapped to the WRF by the homography (yellow diamonds in Fig. 5).

The initial key-point cloud is employed for generation of consecutive clouds. The camera is rotated in a way that the significant part of a new image (Fig. 5-bottom-left) overlaps with the previous one (step 6). The SURF detector is run in order to find feature points on the new image (step 7). Since these points do not have WRF coordinates assigned, they are referred to as unknown and marked with blue squares. However, as the new image overlaps with the reference one, there are correspondences between known and unknown points. To find them, for each point the algorithm calculates its descriptor (a vector of 64 or 128 features). Then an attempt is made to match unknown points from the new image with known points from the reference cloud by calculating distances in a descriptor space (step 8). The points for which relevant correspondences were found are called adjustment points (yellow diamonds in Fig. 5-bottom-left). These points have known WRF positions, thus are further used to compute a new homography and estimate an updated camera pose (step 9). After that the algorithm goes back to step 6 changing the reference cloud if necessary.

The aim of the cloud generation procedure is to obtain full covering of the area observed by the camera keeping the number of acquired images at the minimal level to reduce computational and memory overhead. Fig. 6 shows the

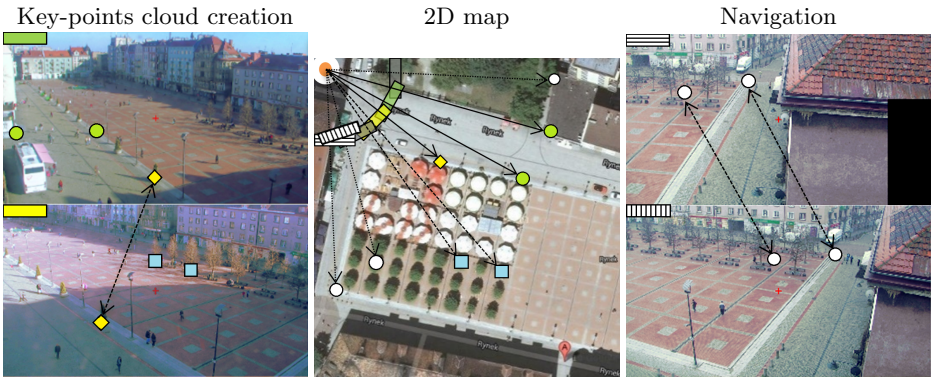

Fig. 5: Visualisation of the key-points cloud creation and navigation.

Algorithm 1 Pseudo-code of the key-point clouds creation.

```
Set initial orientation, acquire the reference image I.
Initialise cloud by manually selecting corresponding points from I and WRF.
Calculate homography H mapping I to WRF, estimate camera pose.
Increase density of the cloud using H and feature points from I detected by SURF.
repeat
    Alter camera orientation, acquire new image I, create empty cloud Ψ.
    Detect feature points on I using SURF.
    Find key-points matching to them in the closest cloud Ω.
    Compute homography H using points from I and matching key-points from Ω.
    Increase density of Ψ using points from I and H.
until entire area covered by clouds
```

mechanism of altering camera orientation in a PT space during key-point clouds generation. The blue dot in the middle is a starting position. The camera pan and tilt are modified in the way that consecutive positions produce a circle in the PT space. This circle will be referred to as a generation. When the current generation is completed, the radius of the circle is increased by a given value, and another generation of clouds is acquired. As described previously, for each processed PT position, a reference key-point cloud has to be selected in order to find matching feature points. In the first generation a starting cloud is used (the blue dot) for this purpose. For consecutive generations, the closest cloud from the previous generation is chosen to be used as a reference.

After all key-point clouds have been generated, the navigation procedure starts. It consists of continuous pose re-estimation on the basis of the current camera image and gathered key-point clouds. The navigation is done according to the following steps: (I) get an image from the camera; (II) select a reference key-point cloud C corresponding to the nearest PT to the current camera orientation; (III) using SURF find on the image a set Y of 70 feature points matching best to C; (IV) compute a homography H upon Y and corresponding WRF points from

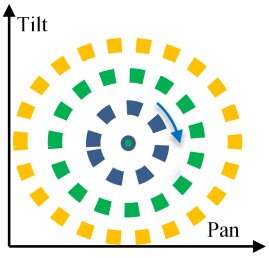

Fig. 6: Rotating camera method used in key-point creation.

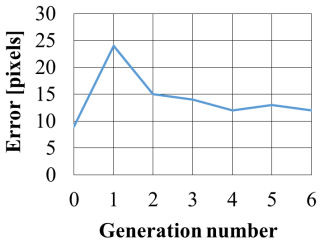

Fig. 7: Average error of navigation.

C; (V) estimate pose using H, Y, and intrinsic camera parameters. In Fig. 5 the current and the reference images are marked, respectively, with a vertically and horizontally hatched rectangles.

5 Accuracy evaluation

The aim of the experimental part was to assess the accuracy of the navigation procedure. For this purpose, approximately 30 different camera orientations in a PT space were tested. For each orientation, an image was acquired and ~ 10 characteristic points were selected manually. Those points were reprojected with a use of estimated camera parameters to the WRF (the map) and compared with their real positions. Reprojection error was measured in pixels. Note, that, according to the navigation method, for each camera orientation, the nearest key-point cloud is used for pose estimation. Thus, a generation containing this cloud was assigned for each orientation, and the reprojection errors were averaged within generations. The results can be seen in Fig. 7. As one can see, the error starts from 10 pixels at generation 0 and after the initial high peak to 25 for generation 1, it decreases and stabilises in the range 10-15 pixels.

6 Conclusions

In the paper we present a camera navigation system allowing rotating cameras to be calibrated in real time without extensive human effort. The preliminary steps include image undistortion, background separation, computation of intrinsic and extrinsic camera parameters for several pan-tilt positions, and generation of the key-points cloud. After that, the continuous tracking procedure starts which automatically recalculates camera pose immediately after changing pan or tilt.

As the method is based on key-points detection, it very sensitive to environment conditions like: weather, season, illumination and even time of a day (due to shadow changes). We plan to solve this issue basing on maximum likelihood methods leading to recognise environmental differences significant for continuous tracking process. The future work will also include addition of zoom as an adjustable parameter, i.e. the extension of the methods to PTZ cameras.

Acknowledgements

This work was supported by a project UOD-DEM-1-183/001 from the Polish National Centre for Research and Development.

References

1. Bowden, P., KaewTraKulPong, R.: An improved adaptive background mixture model for real-time tracking with shadow detection. Proc. of 2nd European Workshop on Advanced Video-Based Surveillance Systems, 135–144 (2001)
2. Davis, J., Chen, X.: Calibrating pan-tilt cameras in wide-area surveillance networks. Proc. of ICCV 2003, 144-149 (2003)
3. Devernay, F., Faugeras, O.: Automatic Calibration and Removal of Distortion From Scenes of Structured Environments. Proc. of SPIE 1995, 62–72 (1995)
4. Devernay, F.: C/C++ Minpack. http://devernay.free.fr/hacks/cminpack/ (2007)
5. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM, 24(6), 381–395 (1981)
6. Hartley, R.I.: Self-Calibration from Multiple Views with a Rotating Camera. Proc. of ECCV 1994, 471–478 (1994)
7. Horaud, R., Conio, B., Leboulleux, O., Lacolle, B.: An analytic solution for the perspective 4-point problem. Comput. Vision Graph., 48(2), 277–278 (1989)
8. Oberkampf, D., DeMenthon, D.F., Davis, L.S.: Iterative Pose Estimation Using Coplanar Feature Points. Comput. Vis. Image Und., 63(3), 495–511 (1996)
9. Thirthala, S., Pollefeys, M.: The radial trifocal tensor: A tool for calibrating the radial distortion of wide-angle cameras. Proc. of CVPR 2005, 321–328 (2005)
10. Tsai, R.Y.: A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. IEEE Robot. Autom. Mag., 3(4), 323–344 (1987).
11. Zivkovic, Z.: Improved Adaptive Gaussian Mixture Model for Background Subtraction. Proc. of ICPR 2004, 28–31 (2004)