Journal of Imaging Science and Technology® 57(2): 020507-1–020507-10, 2013.

Spatial Calibration of a Dual PTZ–Fixed Camera System for Tracking Moving Objects in Video

Grzegorz Szwoch, Piotr Dalka, and Andrzej Czyżewski
Gdansk University of Technology, Multimedia Systems Department, Narutowicza 11/12, 80-233 Gdansk, Poland
E-mail: greg@sound.eti.pg.gda.pl

Abstract. *A dual camera setup is proposed, consisting of a fixed (stationary) camera and a pan–tilt–zoom (PTZ) camera, employed in an automatic video surveillance system. The PTZ camera is zoomed in on a selected point in the fixed camera view and it may automatically track a moving object. For this purpose, two camera spatial calibration procedures are proposed. The PTZ camera is calibrated in relation to the fixed camera image, using interpolated look-up tables for pan and tilt values. For the calibration of the fixed camera, an extension of the Tsai algorithm is proposed, based only on measurements of distances between calibration points. This procedure reduces the time needed to obtain the calibration set and improves calibration accuracy. An algorithm for calculating PTZ values required for tracking of a moving object with the PTZ camera is also presented. The performance of the proposed algorithms is evaluated using the measured data.*

[DOI: 10.2352/J.ImagingSci.Technol.2013.57.2.020507]

INTRODUCTION

Video cameras in modern surveillance systems are used not only for operator-controlled, visual monitoring of the protected areas, but also for automatic video content analysis and detection of important security threats. Such systems serve as automatic assistants to surveillance operators, notifying them about events that may represent security threats.[1] Although the majority of such systems are focused on analysis of video recordings, modern solutions performing event detection in real time are currently engineered by scientists and produced by manufacturers. Automatic event detection requires performing several video analysis operations in a chain; all of these operations need to be performed in the online mode. Fixed or stationary cameras, with a constant field of view, are typically used for video content analysis. They are often high-resolution cameras, in order to enable detailed video analysis and event detection.

Pan–tilt–zoom (PTZ) cameras, with adjustable field of view, form a second type of video acquisition device used in surveillance systems. Because of the changing view, these cameras are not usually employed for automatic video content analysis. However, the adjustable field of view and great optical zoom capabilities make this camera suitable for providing a detailed, zoomed-in view of a selected part of the monitored space, e.g. an object causing an event. In the majority of monitoring systems, PTZ cameras are operator-controlled only, so if the system detects an event, the operator has to manually set the PTZ camera on the area of interest.

In order to make automated video surveillance systems more efficient, a dual camera setup is proposed in this article. One of the cameras is a fixed one, providing data for automated video content analysis with event detection. If an important event is detected, the second camera, of PTZ type, is automatically pointed at the area of the event. Moreover, the proposed system is able to track movements of the selected object on a frame-by-frame basis, by directing the PTZ camera at the detected position of the moving object. In order to realize these goals, a relationship between the point position in the real world and the position of the same point observed in both cameras has to be established. Therefore, a spatial calibration procedure has to be performed. Based on the measured coordinates of the calibration points, transformations between different coordinate systems (real world and two separate cameras) are computed.

Various methods of calibrating cameras with constant field of view have been proposed in the literature. One of the most popular approaches is the Tsai method,[2] which uses pairs of coordinates of calibration points measured in both the real world and the camera image as an input to the procedure that estimates the conversion parameters. This method is based on a pinhole perspective model and uses two transformations: one for the perspective effect and another for camera lens distortions. This method is applicable to both indoor and outdoor cameras, but its main drawback is that the position of each calibration point has to be accurately measured in the real world coordinates. An alternative and often used method was proposed by Zhang.[3] It uses a simpler approach, requiring the camera to observe a planar pattern, e.g. a regular checkerboard, in several different orientations. This is a quick and accurate method, in which the coordinates of the calibration points may be extracted automatically. However, this method is mostly suitable for indoor cameras, for calibration in a close field of view. Calibration of outdoor cameras mounted several meters high above the ground using this method is impractical. Some other approaches may also be found in the literature. For example, Heikkil and Silven proposed a four-step procedure for camera calibration with implicit image correction, with an easier method of estimating camera intrinsic parameters.[4] Clarke and Fryer published a survey of some older calibration methods.[5]

Received Apr. 27, 2012; accepted for publication Apr. 22, 2013; published online Mar. 1, 2013. Associate Editor: Susan Farnand.
1062-3701/2013/57(2)/020507/10/$20.00

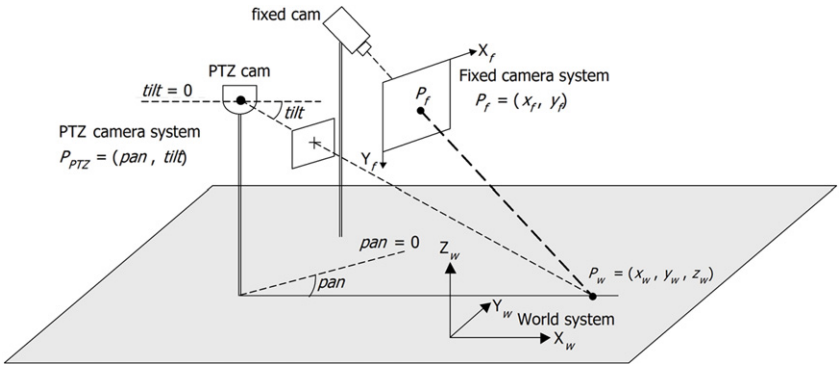

Figure 1. The dual camera setup and the three coordinate systems.

Several spatial calibration methods dedicated to PTZ cameras have also been published. The first important work on this subject is by Agapito et al.[6] They proposed several self-calibration methods, based on infinite homography constraints. Their work was extended by Junejo and Foroosh, who solved the general rotation problem using a series of Givens rotations.[7] Obukhov et al. developed a quick and fully automatic method for PTZ camera calibration, which assumes that intrinsic camera parameters are known *a priori*.[8] Other approaches perform PTZ camera calibration using multiple views, by analyzing inter-image homographies.[9]

So far, no work on calibrating a specific dual camera system, in which the PTZ and the fixed camera are not treated independently, has been published. In previous work by the authors, the dual camera system was calibrated using geolocalization data, mainly for tracking an object with a GPS receiver.[10,11] Other experiments were conducted with spatial calibration of a dual camera system with known (accurately measured) height and position of the camera in the real world coordinate system.[12] In this article, a novel method of dual camera setup calibration for a system designed for automatic tracking of a moving object using the PTZ camera is proposed. Two separate spatial calibrations are performed. The first one relates the PTZ camera position with pixel coordinates of the fixed camera image and allows for directing the PTZ camera to a specific position in the fixed camera view. A method for efficient calibration using only the fixed camera image is proposed. The second calibration allows for coordinate conversion between the fixed camera image and the real world system. For this purpose, the Tsai method is extended with a robust algorithm for estimating coordinates of calibration points using only direct measurements of distances between them.

SPATIAL CALIBRATION OF A DUAL CAMERA SYSTEM

The aim of the study is to calibrate a dual PTZ–fixed camera system in order to facilitate video content analysis using fixed camera images, track movement of an object using the PTZ camera and position this object in a real world space. Therefore, coordinate systems are defined separately for each camera and for the real world, then separate calibration methods for both cameras are proposed and, finally, an algorithm for object tracking with the calibrated camera system is presented.

The dual camera system setup

The setup consists of two cameras. One of them is a fixed camera, with a constant field of view. The images obtained from this camera are used for video content analysis (detection and tracking of moving objects, automatic event detection). The second camera is a PTZ one, with the field of view (FOV) adjustable with pan, tilt and zoom parameters. This camera is used to provide a detailed view of a selected area in fixed camera view and to track moving objects. The PTZ camera is required to allow absolute positioning, i.e., setting the camera to given PTZ values has to result in the same FOV. Additionally, the camera has to provide a means for accurate reading of the current PTZ parameters. During the experiments carried out, IP cameras controlled by HTTP commands for setting and reading camera position were used. It is not required that both cameras are mounted close to each other, although the FOV of the fixed camera should be covered by a reasonable range of pan and tilt values in the PTZ camera. For the purpose of conversion of point coordinates between these two cameras and the real world space, the following three coordinate systems are defined (Figure 1).

The PTZ camera coordinates are related to the camera positioning parameters. The pan is a horizontal camera angle (azimuth), ranging from −180 to 180 degrees. Increasing pan by a positive value results in turning the camera clockwise. It is required that the camera allows continuous pan changes in the whole range of horizontal angles, therefore angle wrapping has to be taken into account. The tilt is a vertical angle (elevation), with increasing value when the camera is tilted up; the range is defined by the camera capabilities. Therefore, $P_{\mathrm{PTZ}} = (p, t)$ is a pair of pan and tilt settings that define a 'ray' cast from the camera through the center pixel of the PTZ camera image. The zoom value only sets the area of the camera view and it may be neglected, provided that changing the zoom value does not shift the center point of the view.

The fixed camera system is related to the view of the fixed camera. The position of any point visible by the fixed camera

is defined by the pixel coordinates

$$P_f = (x_f, y_f) | x_f \in \langle 0, w_f - 1 \rangle, \quad y_f \in \langle 0, h_f - 1 \rangle, \quad (1)$$

where w_f and h_f are the width and the height of the fixed camera image in pixels, respectively. The point (0, 0) is situated in the top-left image corner, point $(w_f - 1, h_f - 1)$ in the bottom-right corner.

The real world coordinate system defines the position $P_w = (x_w, y_w, z_w)$ of any point in the three-dimensional Cartesian system, using physical units, e.g., meters. It is assumed that the origin and axis directions of this system may be selected arbitrarily.

Because of the three different coordinate systems used at the same time, a method of coordinate conversion between these systems (conversion to or from P_{PTZ}, P_f, P_w) has to be developed. Establishing relations between different coordinate systems requires calibration of both camera types. Conversion methods are described in the following sections.

Spatial calibration of the PTZ camera

Transformation of point coordinates between two different camera types is required for aiming the PTZ camera at a specific point in the fixed camera view. Most of the published work on PTZ camera calibration relates PTZ camera parameters to the real world coordinate system. However, because a dual camera setup is utilized in the work described here, the PTZ camera system will be related to the fixed camera system. The problem may be defined as follows. Given any pixel position $P_f = (x_f, y_f)$ in the fixed camera view, it is required to find $P_{\mathrm{PTZ}} = (p, t)$ so that the center point of the PTZ camera view represents the same point in the real world system as P_f. Using this conversion, it is possible to point the PTZ camera at any object whose position is specified in fixed camera coordinates. If this object moves, the pan and tilt values may be modified according to the current object position provided by the object tracker, so that the PTZ camera follows the moving object.

The relationship between the coordinate systems of two different camera types is, in general, non-linear and dependent on camera orientation. Therefore, finding a mathematical relation between these two systems is problematic. Instead, the authors have chosen an approach based on interpolation of measurement results. The spatial calibration procedure is performed as follows. First, a number of distinct points in the fixed camera image are selected. These points should be easy to identify in the PTZ camera view, remain constant in time and be positioned on the ground plane. Additionally, these points should cover the whole camera view (including both near and far fields) and should not be clustered in some parts of the view (a grid of evenly spaced calibration points would be an optimal case). Various landmarks such as posts, traffic lane markings, small trees, etc. are good candidates for selection, provided that a point situated on the ground plane (e.g., the base of a post) is clearly visible in the camera view. If the number of such landmarks is not sufficient, custom calibration markers have to be placed in the camera view. For each calibration point, its pixel position P_f is measured. In the next stage, the PTZ camera is set so that the selected point is situated exactly in the center of the image (the zoom value is set so that the point and its surroundings are clearly visible). Pan and tilt values are read from the camera. As a result, a set of N_c calibration points is defined:

$$\mathbf{c} = \{(x_{fi}, y_{fi}, p_i, t_i) | (p_i, t_i) \sim (x_{fi}, y_{fi})\}, \quad i = 1 \ldots N_c. \quad (2)$$

This set defines the conversion between the systems of the fixed and PTZ cameras, for points contained in the set. Pan and tilt values for pixel positions that are not contained in the set are obtained by interpolation. Additionally, extrapolation is needed for points situated outside the range of calibration points. Linear interpolation is the simplest method, and it may be performed in real time, but its accuracy may be too low for exact PTZ camera control, especially if the number of calibration points is small. More accurate interpolation methods, e.g., the cubic one, are too computationally expensive for real time implementation. Therefore, an approach based on offline computation of interpolated and extrapolated PTZ values was chosen. Two matrices are obtained for estimated pan and tilt values. The size of each matrix is equal to the size of the fixed camera image in pixels. These matrices are stored in memory and used as look-up tables, allowing for quick conversion between the pixel and the pan–tilt values.

For calculation of the look-up tables, a method that allows for accurate interpolation and extrapolation of a non-linear surface, given a small number of node points (less than 0.01% of the total pixel count), is required. Because of this, a method based on biharmonic splines, proposed by Sandwell,[13] was chosen. This method corresponds to multiquadric interpolation, and it both interpolates and extrapolates points non-uniformly spaced on the grid. If the input data is a vector

$$\mathbf{x} = [x_f, y_f]^{\mathrm{T}}, \quad (3)$$

then the pan value at position $\mathbf{x}$ is given by the equation

$$p(\mathbf{x}) = \sum_{j=1}^{N_c} \alpha_j \phi_2(\mathbf{x} - \mathbf{x}_j), \quad (4)$$

where α_j is a coefficient found by solving a linear equations system

$$p_i(\mathbf{x}) = \sum_{j=1}^{N_c} \alpha_j \phi_2(\mathbf{x}_i - \mathbf{x}_j) \quad (5)$$

for $i = 1 \ldots N_c$, and ϕ_2 is a biharmonic Green's function for two-dimensional interpolation given by[13]

$$\phi_2(\mathbf{x}) = |\mathbf{x}|^{2(\ln|\mathbf{x}|-1)}. \quad (6)$$

Solving Eqs. (3)–(6) for all pixels in the fixed camera image (both inside and outside of the calibration points range)

results in the complete look-up table for pan values. The table for tilt values is computed separately using the same method.

A reverse conversion from PTZ values to a fixed camera pixel position is more complex, as it requires finding the position (x_f, y_f) within the look-up tables $\mathbf{T}_P$, $\mathbf{T}_T$ for which the distance between the input pan–tilt (p, t) and the values stored in tables the is the smallest:

$$P(p,t) = (x_f, y_f) | \min \times \left(\sqrt{(\mathbf{T}_{\mathbf{P}(x_f,y_f)} - p)^2 + (\mathbf{T}_{\mathbf{T}(x_f,y_f)} - t)^2} \right). \quad (7)$$

With the proposed approach, transformations in both directions between the coordinates of the two camera types are defined. It should be noted that in the case of IP cameras remotely accessible through a network, the whole calibration procedure can be executed remotely, without the need to perform any measurements *in situ* (provided that no custom markers have to be placed in the camera view).

Spatial calibration of the fixed camera

The problem of coordinate conversion between the fixed camera and the real world systems is solved by finding a relation between the pixel coordinates (x_f, y_f) in the fixed camera view and the point (x_w, y_w, z_w) in the real world, represented by this pixel. For a fixed type and orientation of the camera relative to the ground plane, the conversion is determined by intrinsic parameters, related to the camera optical system, and extrinsic parameters, depending on the camera positioning and orientation. The calibration procedure of the fixed camera is usually performed by collecting data on selected calibration points and measuring their coordinates in both systems. Pairs of coordinates (P_f, P_w) for all calibration points are then input into the algorithm that finds values of both intrinsic and extrinsic parameters, usually by means of non-linear optimization. With these parameters calculated, coordinates of any point may be converted to the other system. This conversion is needed, e.g., in order to find positions of moving objects in the real world, as well as for object size estimation (for object classification purposes) and velocity estimation (for automatic event detection).

In the described work, a method for calculation of conversion parameters proposed by Tsai[2] is used. Eleven parameters need to be estimated: five intrinsic parameters (focal length, radial lens distortion, position of the center of radial lens distortion and the uncertainty scale factor) and six extrinsic ones, related to translation and rotation of the camera relative to the real world coordinate system in three dimensions. In order to estimate all parameters, a reasonable number of calibration points have to be provided and their coordinates in both systems have to be measured with high accuracy. A non-planar calibration, including points of different height z_w, requires more calibration points than a planar one, in which all points are assumed to be positioned on the ground plane ($z_w = 0$). It should be noted that Tsai's algorithm requires six more so-called 'fixed intrinsic' parameters, related to the camera's frame grabber. These parameters are expected to be constant for a given camera model and their values should be provided by the manufacturer. However, in most modern camera models these are not available, and estimation of these fixed parameters is problematic.

Tsai's calibration method imposes some important restrictions on the calibration data. The real world coordinate system has to be right-handed, with the origin inside the image view and not close to either the center of the camera view or the Y axis in the camera system. This may be handled by rotation and translation of the real world coordinate system and inversion of the direction of axes. Moreover, the calibration points have to span the whole usable range in the camera view and the perspective distortion effect has to be evident. However, the most important restriction is that direct real world coordinates of the calibration points have to be provided. It is not sufficient to measure distances between the selected calibration points, as the distance and orientation of each point relative to the real world system origin must be specified. This fact is a main source of measurement errors that result in an inaccurate calibration. The measurements are usually easier in indoor environments, where it is possible to use distinct elements, e.g., floor tiles, as a reference. However, in the case of outdoor environments where calibration points cannot be arbitrarily selected, e.g., on busy crossroads in the city, this problem is difficult to overcome.

In order to simplify the calibration procedure in the described cases, a method of estimation of real world coordinates based only on measured distances between calibration points is proposed. This method assumes that all calibration points are coplanar, e.g., they are placed on the ground level, and it is based on constructing a triangular mesh from the calibration points. Measured distances must be selected carefully in order to guarantee that a convex hull of calibration points can be divided into triangles in such a way that any two triangles share a common side, a common vertex (calibration point) or do not have a common point at all. There are no restrictions on the relative placement of calibration points, i.e., they do not have to be collinear or orthogonal, and no angular data are required. However, it is advised to avoid singular paths of triangles having common edges that connect different parts of the mesh in order to make the calibration results more robust against measurement inaccuracy. In the example shown in Figure 2(a), there is only one path connecting triangles A and B, and hence inaccuracy in measuring edge E would result in a skew of the whole mesh, so errors would increase with the distance from the edge E. This sample mesh should be supplemented with one more calibration point P and four edges (measuring distances) presented in Fig. 2(b) in order to increase calibration robustness.

The input of the procedure for estimation of real world coordinates is a set of edge lengths (i.e., distances between calibration points). The edges form a unidirectional graph. Triangles in the graph are found using a deep-first search algorithm in order to recognize all back edges, i.e., the edges

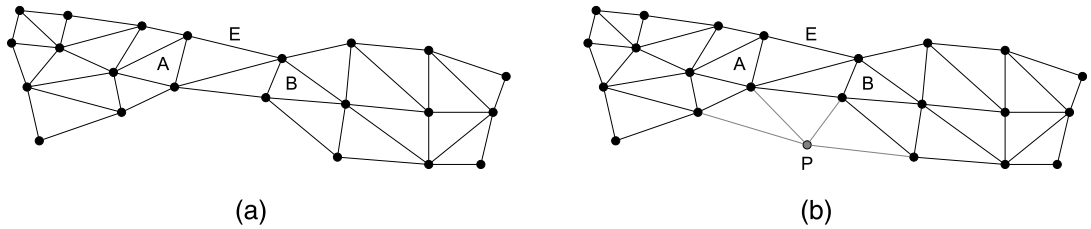

Figure 2. Sample triangular mesh of calibration points with a weak path between triangles *A* and *B* (a) that is very sensitive to edge *E* measurement inaccuracy, and the corrected mesh (b) with one point *P* and four edges added.

that point back to an ancestor during the graph traversal. A back edge denotes a cycle in the graph, and all cycles induced by back edges form a basic set of cycles; each basic cycle corresponds with one triangle. Therefore, each back edge defines two vertices of a triangle; the third one is found as a graph vertex being adjacent to both back edge vertices.

In the next step, the coordinates of all graph vertices are found. The basic procedure finds coordinates of the third vertex C of the triangle T, knowing the coordinates of the other two vertices $A = (A_x, A_y)$ and $B = (B_x, B_y)$ and the distances d_{AC} and d_{BC} between the known vertices and the vertex $C = (C_x, C_y)$. This may be accomplished by calculating the two intersection points of two circles anchored at A and B, having radii d_{AC} and d_{BC}, respectively, according to the formulas

$$\begin{aligned} C_x &= \left(\frac{B_x + A_x}{2} + \frac{(B_x - A_x) \cdot (d_{AC}^2 - d_{BC}^2)}{2 \cdot d_{AB}^2} \right. \\ &\quad \left. \pm \frac{2 \cdot (B_y - A_y) \cdot K}{d_{AB}^2} \right), \\ C_y &= \left(\frac{B_y + A_y}{2} + \frac{(B_y - A_y) \cdot (d_{AC}^2 - d_{BC}^2)}{2 \cdot d_{AB}^2} \right. \\ &\quad \left. \mp \frac{2 \cdot (B_x - A_x) \cdot K}{d_{AB}^2} \right), \end{aligned} \tag{8}$$

where d_{AB} denotes the distance between vertices A and B. K is the area of the triangle T calculated using Heron's formula:

$$K = \frac{1}{4} \cdot \sqrt{((d_{AC} + d_{BC})^2 - d_{AB}^2) \cdot (d_{AB}^2 - (d_{AC} - d_{BC})^2)}. \tag{9}$$

The valid intersection point is selected using the fact that the C vertex has to extend the boundary of the current triangulated area; therefore, it has to be on the opposite side of the line connecting points A and B to the third, already-known vertex of the other triangle containing edge AB; in the case of the first triangle of the graph, any solution may be selected.

Using the calibration point A as the starting vertex of the graph, the coordinates of all other calibration points are calculated. Point A forms the origin (0, 0) of the instantaneous coordinate system. The first triangle T containing vertex A is found. The positive X axis of the coordinate system is directed along the vector connecting the point A with an arbitrarily chosen second vertex B of the triangle T; therefore, its coordinates are $(0, d_{AB})$. The coordinates of the third vertex of triangle T are found according to Eq. (8). Next, another triangle having only one vertex with unknown coordinates is located and the procedure is invoked again until the coordinates for all calibration points are known.

The calibration procedure is repeated for each graph vertex as the starting point. In the result, a set of N coordinates for each calibration point is acquired, where N is the number of calibration points. The results are merged in order to reduce the influence of measurement inaccuracies on the calibration results. For this purpose, an affine transformation matrix M_i having size 2×3 is found for each set of coordinates S_i that converts the coordinates to the common coordinate system defined by the user as the most convenient one; usually one axis of the system is collinear with one of the graph edges. The final real world coordinates of the point $P = (P_x, P_y, 0)$ are calculated using the equation

$$\begin{bmatrix} P_x \\ P_x \end{bmatrix} = \frac{1}{N} \sum_{i=1}^{N} M_i \cdot \begin{bmatrix} P_x^i \\ P_y^i \\ 1 \end{bmatrix}, \tag{10}$$

where (P_x^i, P_y^i) are the coordinates of the point P in the i-th set S_i.

With the fixed camera calibrated, the conversion of the coordinates is a two stage process. The coordinates of the point in the real world system are first converted to the undistorted camera plane using extrinsic parameters, and then to the distorted camera plane, with intrinsic parameters, resulting in calculated pixel coordinates in the fixed camera view. The inverse conversion is possible using the same algorithm, but since it is performed from 2D to 3D space, the height value z_w needs to be provided. Therefore, either a height map is needed, with the z_w value stored for each pixel, or the height must be directly specified (for example, it may be assumed that the point is on the ground, i.e., $z_w = 0$).

TRACKING MOVING OBJECTS WITH THE PTZ CAMERA

The calibration procedures for the fixed and PTZ cameras presented in the previous section are implemented in the framework for automatic tracking of objects moving within a fixed camera FOV, with PTZ cameras. A graph of the framework is presented in Figure 3. The framework receives

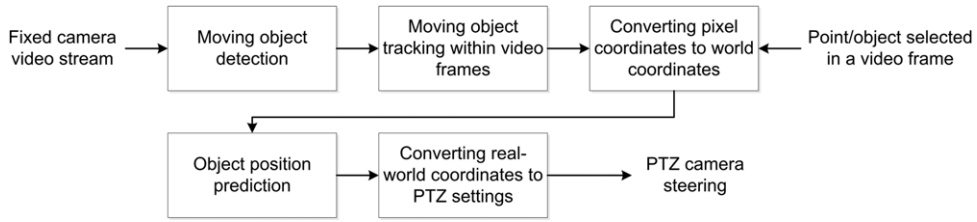

Figure 3. Block diagram of the framework for moving object tracking with PTZ cameras.

a video stream from a fixed camera. Moving objects are automatically detected using the algorithm based on background modeling with Gaussian mixture models as these proved to be effective in the authors' earlier experiments.[14] The results of background modeling are post-processed by detecting and removing shadow pixels (based on the color, luminance and texture of shaded regions). Next, movements of the detected objects (blobs) are tracked in successive image frames using a method based on Kalman filters that allows prediction of object positions in the current frame based on past observations.[15] By comparing results of background subtraction with predicted object positions, it is possible to correlate each tracker with the detected movement, so that the movement of each object is tracked continuously.[16] The standard method has been improved by the authors in order to handle conflict situations such as passing by, partial occlusions, merging, splitting, etc.[17] For this purpose, an iterative, appearance-based blobs to trackers matching procedure has been developed, which has proved to perform well in the case of high traffic.[18]

With the graphical user interface application, a user of the framework is able to select an arbitrary point or a moving object within the fixed camera FOV. The selection is sent to the framework and point/object real world coordinates are calculated using the procedure described above, assuming that the selected point/object is located on the ground level (the z coordinate of the real world system is zero). In the case of moving objects, the delay of the system must be taken into account in order to successfully track persons/vehicles in motion, i.e., objects of interest have to be always present near the center of a video frame from the PTZ camera. The delay is caused by video processing, data transmission and time required for executing the PTZ command. The delay compensation is performed by setting the PTZ camera to a predicted, real world position of the object instead of the one converted directly from the pixel coordinates. The prediction time should be equal to the total delay in the framework. In the current implementation, a linear predictor is used that estimates object position based on its instantaneous velocity and heading (direction of movement). The delay introduced by video processing is roughly estimated by comparing the current time with a timestamp generated when a video frame, used for obtaining the object location, was captured by a fixed camera. An additional constant delay equal to 0.1 s is added to compensate for other delay factors.

The predicted object position $(\hat{x}, \hat{y}, \hat{z})$ is given by the equation

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} + d \cdot \begin{bmatrix} v \cdot \sin(\Theta) \\ v \cdot \cos(\Theta) \\ 0 \end{bmatrix}, \tag{11}$$

where $(x, y, 0)$ is the real world position of the object derived directly from video frame pixel coordinates, d is the framework delay in seconds, and v and Θ are the object's current speed and heading, calculated as follows:

$$\begin{aligned} v &= \sqrt{v_x^2 + v_y^2}, \\ \Theta &= a\tan 2(v_y, v_x), \end{aligned} \tag{12}$$

where v_x and v_y are estimates of object velocity (in physical units) in the horizontal and vertical directions of the real world coordinate system, respectively.

The estimated object position (or direct point position) is used to calculate pan, tilt and zoom parameters according to the description above and to aim the PTZ camera at the required location.

EXPERIMENTS AND RESULTS

PTZ camera calibration

In order to verify the proposed procedure for conversion between the coordinates of PTZ and fixed cameras, a test system consisting of one PTZ and one fixed camera, mounted on the same lamp post, was calibrated. The fixed camera was an IQeye 702, with 1600 × 1200 pixels resolution. The PTZ camera was an Axis 233D, with a resolution of 704 × 576, continuous pan control and tilt controlled in the range from −90 to 0. The spatial calibration procedure was performed and verified remotely, through the network. The PTZ camera was controlled using HTTP commands through the VAPIX protocol.

In the fixed camera view, 181 calibration points were selected. This constitutes less than 0.01% of all pixel coordinates in the image. Distinctive points, such as edge of traffic lane markings, posts, etc. were used. These points covered the whole FOV of the fixed camera (Figure 4). The pixel coordinates of the calibration points were measured. In the next stage, the PTZ camera was directed at each individual calibration point, using the crosshair cursor in the center of the PTZ camera image. The zoom value was set so that the calibration point could be easily identified. Pan

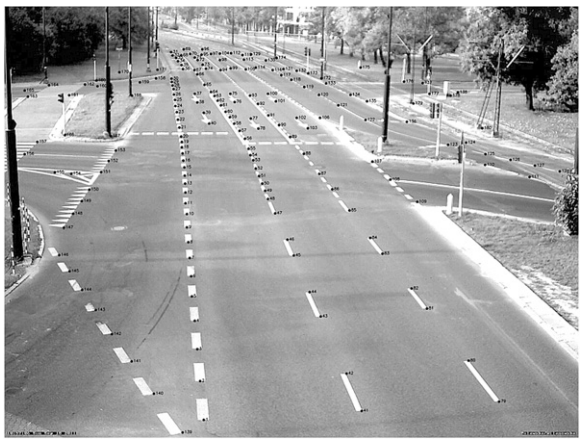

Figure 4. Calibration points selected in the fixed camera image.

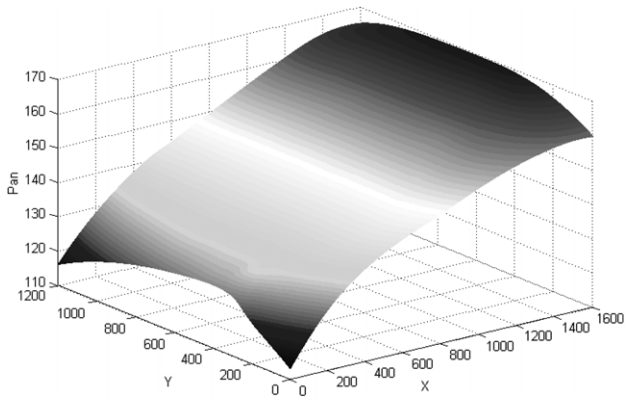

Figure 5. Surface plot of pan values estimated for all pixel positions (x_f, y_f), obtained using the proposed method.

and tilt values were noted for each calibration point. As a result, each calibration point was described using four values: (x_f, y_f, p, t).

In the next stage, look-up tables were computed for pan and tilt, separately. The approach presented in this article, based on biharmonic splines, was used for interpolation of data between calibration points used as mesh nodes, as well as for extrapolation outside of the calibration points range. Computed look-up tables may be visualized as surfaces (Figures 5 and 6). It can be noted that in the case of the test system, the tilt surface is almost flat and tilted in relation to the fixed camera image plane. The surface for the pan look-up table is almost flat in the central image area and bent on the edges.

In order to test how many calibration points are required for satisfactory accuracy of pan/tilt estimation, the calibration procedure was repeated for smaller sets of calibration points, uniformly selected from the original set. Next, the obtained look-up tables were used for estimating pan and tilt values for all calibration points. The results, expressed as a mean square error and a mean absolute difference between the measured and the estimated pan/tilt values, are presented in Table I. It can be observed that in the case of the test system, as few as 20 calibration points are enough to obtain satisfactory accuracy, with a mean absolute

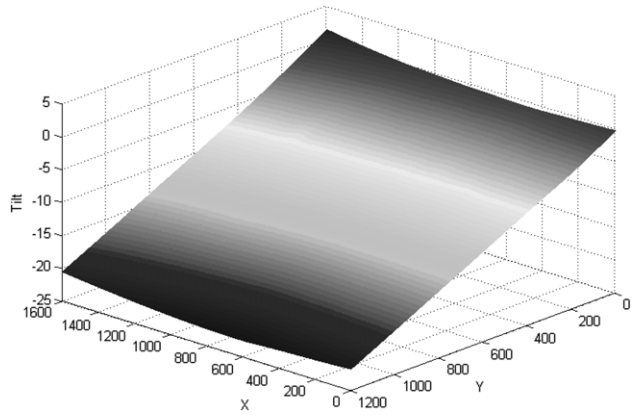

Figure 6. Surface plot of tilt values estimated for all pixel positions (x_f, y_f), obtained using the proposed method.

difference smaller than 1° for pan and 0.1° for tilt. Further reduction of the number of calibration points leads to rapid increase in error values. Using 50 to 75 points should be enough for good estimation accuracy with fixed cameras of similar resolution; fewer points will be required for lower image resolutions. For more than 100 points, no significant improvement in estimation accuracy was observed, absolute differences were low and mean square error was fluctuating.

Fixed camera calibration

For assessment of the accuracy of the proposed extension of Tsai's calibration method which calculates coordinates of calibration points based only on measurements of distances between the points, the following test was performed. Several synthetic sets of a different number of calibration points, covering a virtual field of 18 × 15 m, were generated. In each set, points were first spaced uniformly on the grid, then they were shifted randomly in the range of 1 m. Thus, reference sets of points were obtained. Next, triangular meshes for each set were constructed and the side lengths in each triangle were computed as the measurement data. In order to simulate measurement errors, the calculated distances were 'polluted' with a zero-mean Gaussian noise. The test procedure calculated the coordinates of points using the provided data. The results were compared with the reference data and values of root mean square error (RMSE) were computed. These tests were repeated for varying standard deviation of the Gaussian noise, simulating the range of measurement errors.

In Figure 7, results of the simulation for three point sets, consisting of 30 points (6 × 5 grid), 20 points (5 × 4) and 12 points (4 × 3), are presented. Calculated RMSE values are plotted against the standard deviation of the noise. In the case of a zero noise, the estimated coordinates are consistent with the reference data. With increasing standard deviation of the noise, RMSE rises almost linearly. For larger calibration point sets, the error is higher, and the difference increases when the noise deviation is larger. This effect was expected, because for larger data sets, a larger number of triangles has to be processed and errors introduced with the processing of each triangle are accumulated. However,

Table I. Results of pan and tilt estimation using the proposed method, expressed as mean square error (MSE) and mean absolute difference (MAD), for varying number of points used for estimation.

Num. of points		10	20	25	50	75	100	125	150
Pan	MSE	7.948	2.338	1.851	0.388	0.036	0.015	0.031	0.048
	MAD	2.023	0.608	0.482	0.162	0.064	0.042	0.035	0.032
Tilt	MSE	0.053	0.025	0.009	0.0019	0.0016	0.0013	0.001	< 0.001
	MAD	0.193	0.089	0.054	0.028	0.023	0.017	0.011	0.006

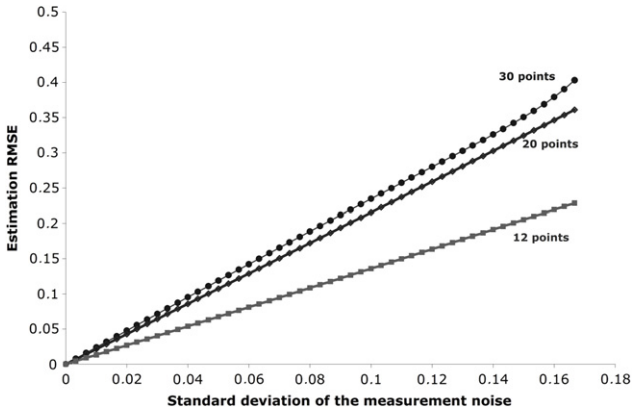

Figure 7. Results of simulation in which coordinates of calibration points were estimated with the proposed method, using synthetic data with added Gaussian noise. The plot presents the root mean square error of the estimation versus the standard deviation of the noise, for three synthetic data sets.

assuming that the accuracy of real measurements is up to 10 cm, which is roughly simulated by a Gaussian noise with standard deviation equal to 0.03, the mean error of point coordinate estimation is about 2 cm for the 12-point data set and about 8 cm for the 30-point data set. Therefore, the accuracy of the proposed algorithm is satisfactory. Moreover, elimination of the need to maintain the correct orientation of each calibrated point relative to the reference point (the origin of the coordinate system) reduces measurement errors in the original approach. Note that although smaller data sets result in lower RMSE in the estimation, higher errors may be obtained during the actual calibration due to small coverage of the calibrated area.

In order to assess the performance of the proposed algorithm in a real world scenario, calibration of two cameras was performed using two approaches. First, the original Tsai calibration method was used for a set of calibration points carefully positioned on a rectangular grid, so that the exact real world coordinates of each point could be obtained. Next, distances between points selected arbitrarily in the camera view were measured, forming a triangular mesh covering the observed space. The coordinates of these points were then calculated using the proposed triangle algorithm, then the Tsai model was applied for calibration. The results of both calibrations were compared using three error metrics. An image plane error is calculated by converting measured real world coordinates of each calibration point to image coordinates using the calibrated model, and measuring the pixel distance between the actual and the converted pixel position. An object space error is calculated by performing the reverse conversion, from image to real world system, and calculating the distances between the converted and the measured points. Both measures are expressed as a mean and standard deviation of distance error, as well as a mean squared error. Additionally, a normalized calibration error (NCE) is calculated for assessment of the overall accuracy of the calibration model, using the formula proposed by Weng[19]:

$$\mathrm{NCE} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2}{\hat{z}_i^2(f_u^{-2} + f_v^{-2})/12}\right]^{1/2}, \qquad (13)$$

where (x_i, y_i, z_i) are the true coordinates of the i-th calibration point in the camera-centered system, $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$ are the coordinates of the i-th point back-projected to plane $z = z_i$ using the calibrated model, f_u and f_v are row and column focal lengths, respectively, and N is the number of calibration points. Ideally, the NCE metric should be equal to one.

Test results are presented in Table II. It can be seen that the proposed method improves calibration accuracy, as all error metrics are lower than for the original method. Errors in the original approach resulted from inaccurate measurements of calibration point coordinates, as each point position needed to be related to the real world system origin, which was difficult to achieve. This problem was not present in the proposed approach, as only distances between point pairs were measured. Errors introduced by the algorithm converting measurements to point coordinates were lower than those related to measurements in the original method. As a result, the MSE for the object space, representing the accuracy of estimating point positions in the real world system, was approximately 3.3 times lower for the first camera and 6 times lower for the second one. The NCE values were also lower for the proposed method. It should also be noted that the proposed approach is easier to perform than the original one, as only distances between selected points are measured, without need to maintain the relation to the real world system origin.

Additionally, several distances between pairs of points selected in the camera view (other than the calibration points) were measured. The pixel coordinates of these points, measured in the camera image, were then converted to the real world system using the calibrated camera model. Distances between the converted points were calculated and compared with the measurement results. The calculated values of the MSE for this test set were 11,630 mm^2 for the

Table II. Error metrics for two cameras calibrated with the Tsai algorithm, using the original and the proposed methods. Errors are expressed as means with standard deviations and mean square errors for both the image plane and the object space, and also as normalized calibration errors (NCEs).

Camera	Error metric	Original method	Proposed method
Camera 1 (704 × 576)	Image plane error (mean ± std dev)	1.01 ± 0.48 px	0.72 ± 0.41 px
	Image plane error (MSE)	27.91 px^2	8.16 px^2
	Object space error (mean ± std dev)	36.82 ± 19.63 mm	26.89 ± 18.38 mm
	Object space error (MSE)	41,399 mm^2	12,394 mm^2
	NCE	2.465	1.781
Camera 2 (640 × 480)	Image plane error (mean ± std dev)	2.17 ± 1.12 px	1.28 ± 0.73 px
	Image plane error (MSE)	160.55 px^2	32.12 px^2
	Object space error (mean ± std dev)	67.76 ± 36.03 mm	37.44 ± 19.12 mm
	Object space error (MSE)	157,712 mm^2	26,154 mm^2
	NCE	5.287	3.143

original method and 4450 mm^2 for the proposed method, which, after calculating a square root, yields values of 107.8 mm and 66.7 mm, respectively. This confirms that the proposed method produces more accurate calibration results.

Object tracking with the PTZ camera

The novel calibration methods proposed in this article have been applied to the practical task of tracking moving objects with a PTZ camera. The setup consisted of one fixed camera, an Axis Q1755, providing images with 720 × 1280 resolution (90° rotation) and one PTZ camera, an Axis 233D, working in 704 × 576 resolution. The automatic video processing framework presented above has been used to detect moving objects in the fixed camera FOV and to calculate their real world coordinates. On a user request, the PTZ camera settings were calculated for a chosen object. Values of pan and tilt were obtained from the look-up tables calculated during the PTZ camera calibration, with the method proposed in this article. Additionally, a look-up table was constructed for zoom values as a function of camera tilt, by setting the camera view for several tilt values in such a way that the object sizes in the zoomed-in images are approximately constant, acquiring zoom values from the camera and applying cubic interpolation. With these PTZ settings, the camera was aimed at the object and was tracking it continuously (Figure 8). The experiments carried out proved that regardless of the object type (fast moving, larger vehicles or slow, smaller persons) the framework is able to track moving objects with high accuracy.

CONCLUSIONS

We presented two solutions for spatial calibration of a dual camera setup, consisting of a fixed and a PTZ camera. The procedure for calibration of the PTZ camera allows for pointing this camera at any point in the real world visible in the fixed camera view. No data on point coordinates in the real world system are required and the whole calibration procedure may be performed remotely, provided that the cameras are accessible through a network and a sufficient number of calibration points may be selected. The interpolation and extrapolation method used in the algorithm allows for accurate estimation of pan and tilt parameters. The interpolation is performed offline while the actual procedure works in online mode, at the cost of storing precomputed look-up tables in the memory. For the fixed camera calibration, the algorithm proposed by Tsai was extended with a procedure for estimating coordinates of calibration points. With this approach, it is no longer necessary to measure these coordinates directly, which was a main source of errors in the calibration procedure. Instead, only distances between calibration points forming a triangular mesh are required. With this procedure, it is possible to reduce the time needed to obtain the calibration measurements data and to improve calibration accuracy.

An algorithm that uses PTZ cameras for automatic tracking of movement of objects whose positions in each frame are obtained from the video content analysis system using fixed camera images as the source was proposed. With this approach it is possible to use the fixed camera for automatic video analysis, including object detection, tracking and event detection, while the PTZ camera is used to provide a zoomed-in view of an event area or to track movement of a selected object.

Application of the dual camera setup in the automated video surveillance system may help in improving the efficiency of such solutions. The automatic video analysis system may notify the operator of important events, such as traffic violations and security threats. The PTZ camera is automatically directed on a moving object or an event site, providing a more detailed view of the event. For example, the PTZ camera may show a restricted area after a person enters it, or it may track a vehicle that violated a traffic rule. Future research on this topic will be focused on using multiple PTZ cameras that allow for continuous object tracking, providing another increase of efficiency of automated video surveillance systems.

ACKNOWLEDGMENTS

This research is subsidized by the European Commission within FP7 project INDECT, Grant Agreement No. 218086.

Figure 8. Sample video frames from a fixed camera (top, moving objects marked with rectangles) and a PTZ camera (bottom) while tracking moving objects on a rainy winter day.

REFERENCES

[1] A. Czyżewski, G. Szwoch, P. Dalka, P. Szczuko, A. Ciarkowski, D. Ellwart, T. Merta, K. Łopatka, Ł. Kulasek, and J. Wolski, "Multi-stage video analysis framework," *Video Surveillance*, W. Lin, Ed. (Intech, Rijeka, 2011), pp. 147–172.

[2] R. Y. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, New York City, New York, 1986), pp. 364–372.

[3] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," *Proc. 7th IEEE Int. Conf. on Computer Vision* (IEEE, New York City, New York, 1999), pp. 666–673.

[4] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition* (IEEE, New York City, New York, 1997), pp. 1106–1112.

[5] T. A. Clarke and J. G. Fryer, "The development of camera calibration methods and models," Photogrammetric Record **16**, 51 (1998).

[6] L. D. Agapito, E. Hayman, and I. Reid, "Self-calibration of rotating and zooming cameras," Int. J. Comput. Vision **45**, 107 (2001).

[7] I. N. Junejo and H. Foroosh, "Practical PTZ camera calibration using Givens rotations," *Proc. Int. Conf. on Image Processing ICIP'08* (IEEE, New York City, New York, 2008), pp. 1936–1939.

[8] A. Obukhov, K. Strelnikov, and D. Vatolin, "Fully automatic PTZ camera calibration method," *Proc. of GraphiCon 2008* (GraphiCon, Moscow, Russia, 2008), pp. 122–127.

[9] Y. Seo, M. H. Ahn, and K. S. Hong, "A multiple view approach for auto-calibration of a rotating and zooming camera," IEICE Trans. Inform. Syst. **E83-D**, 1375 (2000).

[10] P. Dalka, A. Ciarkowski, P. Szczuko, G. Szwoch, and A. Czyzewski, "Surveillance camera tracking of geo-positioned objects," *Studies in Computational Intelligence* **Vol. 226**, (Springer, Berlin, 2009), p. 21.

[11] G. Szwoch, P. Dalka, A. Ciarkowski, P. Szczuko, and A. Czyzewski, "Visual object tracking system employing fixed and PTZ cameras," J. Intell. Decis. Technol. **5**, 177 (2011).

[12] G. Szwoch and P. Dalka, "Automatic detection of abandoned luggage employing a dual camera system," *Proc. IEEE Int. Conf. on Multimedia Communications, Services and Security* (IEEE, New York City, New York, 2010), pp. 56–61.

[13] D. T. Sandwell, "Biharmonic spline interpolation of GEOS-3 and SEASAT altimeter data," Geophys. Res. Lett. **14**, 139 (1987).

[14] P. Dalka, "Detection and segmentation of moving vehicles and trains using Gaussian Mixtures, shadow detection and morphological processing," Mach. Graphics Vision **15**, 339 (2006).

[15] N. Funk, "A study of the Kalman filter applied to visual tracking", University of Alberta, Project for CMPUT 652, Dec. 7, (2003).

[16] A. Czyzewski and P. Dalka, "Examining Kalman filters applied to tracking objects in motion", *Proc. 9th Int. Workshop on Image Analysis for Multimedia Interactive Services WIAMIS'09*, (2008) pp. 175–178.

[17] G. Szwoch, P. Dalka, and A. Czyzewski, "Resolving conflicts in object tracking for automatic detection of events in video," Elektronika **52**, 52–55 (2011).

[18] M. Szczodrak, P. Dalka, and A. Czyzewski, "Performance evaluation of video object tracking algorithm in autonomous surveillance system", *2nd Int. Conf. on Information Technology (ICIT)*, (2010) pp. 31–34.

[19] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," IEEE Trans. Pattern Anal. Mach. Intell. **14**, 965 (1992).