Postprint of: Szwoch G., Dalka P., Ciarkowski A., Szczuko P., Czyżewski A., Visual Object Tracking System Employing Fixed and PTZ Cameras, Intelligent Decision Technologies-Netherlands, Vol. 5, Iss. 2 (2011), pp. 177-188, DOI: 10.3233/IDT-2011-0105

Visual Object Tracking System Employing Fixed and PTZ Cameras

Grzegorz Szwoch, Piotr Dalka, Andrzej Ciarkowski, Piotr Szczuko, Andrzej Czyżewski

Multimedia Systems Department, Gdańsk University of Technology, Naturowicza 11/12 Str., 80-233 Gdańsk, Poland. Phone: (+48 58) 347-13-01, Fax: (+48 58) 347-11-14

greg@sound.eti.pg.gda.pl

Abstract:

The paper presents a video monitoring system utilizing fixed and PTZ cameras for tracking of moving objects. First type of camera provides image for background modelling, being employed for foreground objects localization. Estimated objects locations are then utilised for steering of PTZ cameras when observing targeted objects with high close-ups. Objects are classified into several classes, then basic event detection is being performed. Event type, object localisation and images acquired by the cameras are presented visually in a "live map" system. In the paper details related to detection of moving objects are presented. Next, camera calibration procedure and geopositioning coordinates way of processing are discussed. Event detection is described. Finally an experiment is presented, organised in order to verify the camera tracking system accuracy.

Keywords:

computer vision, image analysis, background subtraction, monitoring, event detection.

1. Introduction

Nowadays for efficient image analysis it is assumed that analysed stream is acquired by fixed camera. In that case moving objects are detected as a set of pixels obscuring actual background. The background modelling is discussed in Sec. 3. The view area of fixed camera is narrow and of low resolution, therefore limiting possibility of detailed analysis of detected objects. Contrary a PTZ cameras, with capability of panning, tilting and zooming the image, provide technical means for following moving objects with high close-ups. For that purpose a precise localization of the object is needed. Object location in the image from calibrated fixed camera can be transformed into real word coordinates, next serving as a steering data for PTZ camera. That procedure is described in Sec. 3.2, 3.3. In Sec. 4. the event detection algorithm is discussed, allowing for automatic selection of crucial objects for tracking in the scene. Results of tracking and event detection are presented in a "live map" system, described in Sec. 5.

Sec. 6. summarises the work, by presenting verification of efficiency of tracking algorithm.

2. System architecture

Developed and discussed tracking algorithm assumes that there is a monitoring system comprising fixed cameras and PTZ cameras connected to central server. The video streams are analysed, moving objects location is being estimated, and that information is being handled by the server and employed for steering of moving cameras. For storing and processing locations of objects and cameras a geo-positioning coordinates are used. Manipulating the GPS data to transfer object location between fixed and PTZ cameras facilities creation of a *distributed* surveillance system because each camera may be placed practically anywhere and calibrated independent of other cameras. During exploitation of the system the same object can be observed by various cameras from different angles, and effectively identified as a single object, occupying the same coordinates in space.

3. Object tracking with PTZ cameras

For the purpose of tracking moving objects with a PTZ camera, information regarding object localization needs to be obtained. It can be accomplished in a few ways. The easiest way is to equip a moving object (e.g. a vehicle) with a GPS receiver and to send its localization through a cellular network or other wireless connection available. However, this approach has very limited applicability. In more convenient solutions, localization of objects has to be acquired by the means of remote measuring.

In the system presented in the paper, all moving objects are automatically localized in a fixed camera field of view. For this purpose, video processing algorithms are used to detect and track all objects present in video frames acquired from the camera. In the next step, translation of pixel-based coordinates into local, real-world coordinates is performed. This stage requires proper camera calibration. In the last step, local coordinates are translated into GPS coordinates. Based on the universal GPS data, any PTZ camera in a surveillance system may be aimed directly and instantly at any object detected in a video stream coming from the fixed camera. However calculating proper values for pan, tilt and zoom parameters of the PTZ camera requires additional processing that involves PTZ camera calibration.

3.1 Video Object Detection and Tracking

Moving object detection and tracking is performed to find all moving objects in each video frame and to trace their movements in the camera field of view. There are many methods for video object detection. Most of them usually employ spatial and/or temporal information to generate binary masks of objects [8][9][10]. The solution presented here utilizes spatial segmentation to detect moving objects in video sequences. The most popular region-based approach is background subtraction [17]. However simple time-averaging of video frames is insufficient for a surveillance system because of limited adapting capabilities.

In our surveillance system, the adaptive background subtraction method models pixels as mixtures of Gaussians and uses an on-line approximation to update the model [2][4]. This method proved to be very useful in many applications, as it is able to cope with illumination changes and to adapt the background model accordingly to the changes in the scene, e.g. when motionless foreground objects eventually become a part of the background. Furthermore, the background model can be multi-modal, allowing regular changes in the pixel colour. This makes it possible to model such events as trees swinging in the wind or traffic light sequences.

Object detection is supplemented with shadow detection and removal. This step is required for every outdoor video processing application, especially in the field of video surveillance. The shadow of a moving object moves together with an object and as such is detected as a part of the foreground object by the background removal algorithm. The shadow detection method is based on the idea that while the chromatic component of a shadowed background part is generally unchanged, its brightness is significantly lower [4][7].

In the result of the background modelling, a binary mask denoting pixels recognized as belonging to foreground objects in the current frame is obtained. It needs to be refined by the means of morphological processing in order to allow object segmentation [4]. This process includes finding connecting components, removing objects being too small, morphological closing and filling holes in regions. Additionally, an algorithm for shadow removing from the mask using morphological reconstruction is implemented [16]. It combines results of objects detection with and without shadows removed in order to reconstruct masks of all object damaged by the aggressive shadow removal procedure.

Moving object tracking has to assure the integrity of the traced object in case of faulty background subtraction or when an object temporarily disappears behind an obstacle (e.g. pillar) or to achieve continuous, valid tracks in situations when objects temporary occlude each other. The Kalman filtering provides a useful approach to this task [6][11]. In the process of tracking, each of the detected moving objects is assigned its own Kalman filter (so-called tracker). A tracker may be illustrated as a rectangle surrounding an object and provides information about the object location, size and current changes in the location and size. The most valuable feature of Kalman filters is the ability to predict the object location and size in the following image frame. The results of the prediction are compared with the real results of object detection in order to establish proper relations between regions denoting moving objects and trackers.

There are 6 basic types of relations between trackers and regions, each of them requiring some different actions to be taken [3]. These relations and actions are summarized in Tab. 1. In order to relate each tracker to the valid region successfully in case of the many trackers to many regions relation, a 2D colour histogram using a chromatic space of R_cG_c colours was applied for each object. Sample results of object tracking in case of serious occlusions are shown in Fig. 1.

3.2. Determining GPS coordinates of the object

Once the positions of all moving objects in the current frame of the fixed camera image are determined, the PTZ camera may be used to obtain a zoomed-in view of the selected object. However, the object tracking procedure provides positions of the objects in the image pixels coordinates. In order to set the field of view of the PTZ camera to the area containing the selected object, a procedure for conversion between the fixed camera image coordinates (in pixels) and PTZ camera settings (pan angle, tilt angle and zooming coefficient) is needed. It is convenient to divide this problem into three separate conversions. First, image coordinates (in pixels) are converted to real world coordinates (in meters). In the next step, real world coordinates are converted to GPS position. The last step is conversion of the GPS position to the PTZ camera settings. The first two steps of this procedure will be described below and the details of the last step will be presented in Sec. 3.3.

3.2.1 Conversion of image coordinates to local coordinates

The local coordinates system is used to establish a position of any object in the real world. As an origin of the coordinates system any point in the real world may be set (although it is convenient to select a point inside the camera's field of view) and usually *x* and *y* axes are positioned on the ground plane, while the *z* axis is perpendicular to the ground. A conversion between the two-dimensional image coordinates and the three-dimensional world coordinates is possible once a set of conversion parameters is found. A typical approach used

for this task is performing the camera calibration procedure which calculates the needed parameters using pairs of image coordinates and real coordinates for each calibration point.

Various camera calibration methods may be found in the literature, but the method proposed by Tsai [14], based on the pinhole perspective projection model, is probably the most frequently used one. This approach was also used in the experiments described in this paper. Tsai's calibration model defines a total of 17 conversion parameters that may be divided into three groups. The first group contains five intrinsic (internal) parameters, related to camera's lens: an effective focal length, a radial lens distortion coefficient, x and y coordinates of the center point of the radial lens distortion and the scale factor. These parameters are constant for a given fixed lens camera and they do not change if the camera's field of view is changed (unless the lens zoom is also changed). The second set of extrinsic (external) parameters describes an orientation of the image coordinates system relative to the world coordinates system and consists of three rotation angles and three translation coefficients. These values need to be recalculated if the camera is repositioned. The third group contains six fixed intrinsic parameters related to the camera sensor and they are constant for a given camera.

Tsai's calibration procedure computes values of the parameters described above using a set of calibration points provided in both image coordinates (in pixels) and local coordinates (in meters), numerically solving the conversion equations. Tsai's method requires that the local coordinates system is right-handed, i.e. x values increase in right-to-left direction in the image frame and y values increase in top-to-bottom direction. In order to obtain proper calibration accuracy, the origin of the world coordinates system should be positioned neither near the center of the image nor near the image's left border. Additionally, two calibration modes are available: a coplanar mode (if all the calibration points are situated on the ground plane, this mode was used in the experiments) or a non-coplanar mode (if the calibration points differ in height). The calibration points should ideally span the whole calibrated area. A minimum number of calibration points is 11, a greater number of points increases the calibration accuracy, provided that points coordinates are also accurate.

Once all the conversion parameters are found, it is possible to convert the world coordinates of any point in the camera's field of view to the image coordinates. The conversion procedure uses the parameters obtained during the calibration and it may be divided into several conversion steps: first the world coordinates system is rotated and translated in order to align it with the image coordinates system, then it is converted to undistorted plane coordinates (3D to 2D conversion), the result is converted to distorted plane coordinates (taking lens distortion into account) and finally image coordinates are computed. In order to obtain a position of a moving object in the world (local) coordinates, a conversion between the 2D image coordinates and the 3D world coordinates has to be performed using the steps listed above in the reversed order. However, in order to convert 2D coordinates into tree dimensions, a value of z world coordinate has to be provided. Assuming that all the object move on the ground plane, $z = 0$ may be used.

Fig. 2 shows an example image frame from the fixed camera with calibration points marked. This figure illustrates several problems that occur when the calibration procedure is performed in practice. The accuracy of the camera calibration depends on the accuracy of measurements of calibration points positions, both in camera

image and in the real world. Ideally, calibration points should form a rectangular mesh. This is hard to achieve in the practical situation, especially if the camera's field of view is large (like in Fig. 2), and placement of the calibration points is difficult because of heavy traffic. Usually, some visible landmarks are selected as calibration points. However, in this case finding an exact position of each point relative to the origin of the world coordinates system is problematic. During the experiments, the image shown in Fig. 2 was used for camera calibration and it was assumed that the posts situated on the sidewalks and posts in the fence form three parallel lines. However, this simplification introduces some errors in measurements of the world positions of calibration points. Also, measurements of distances between the calibration points (with the measure tape) introduces some errors that may accumulate and decrease the calibration accuracy. It is not possible to avoid these errors unless a sophisticated (and expensive) equipment is used in the measurements. The authors tried to use satellite images or GPS receivers for the measurement of exact calibration points positions. However, technical limitations (limited resolution of available satellite images and limited positional accuracy of GPS receivers) made these approaches unsuitable for calibration purposes.

The accuracy of camera calibration is further limited by errors in determining the position of calibration points in the image frame. Ideally, coordinates of the calibration points should be provided with the subpixel accuracy. This condition is hard to fulfill in practical situations, because the calibration markers have to be large enough to be visible in the camera image, so they usually span several pixels and determining the pixel that exactly represents the measured point is problematic. The problem is more prominent if the contrast between the calibration marker and the background is not sufficient, so it is hard to determine the contours of the marker in the blurred zoomed-in image (Fig. 3). As a consequence, the pixel selected as a calibration point position in the image is usually not the same point that was used in the real world measurements which results in errors in conversion between the image and world coordinates.

To conclude this section, it is possible to convert image coordinates to real world (local) coordinates using the camera calibration parameters. However, one has to remember that several factors (discussed before) decrease the accuracy of the calculated real world position, which may influence the accuracy of the PTZ camera steering.

3.2.2 Local coordinates to GPS coordinates conversion

Conversion of local, real-world coordinates to GPS coordinates is the last stage in obtaining GPS localization of moving objects detected in a video stream. This stage is an affine transformation that includes translation and linear transformations like scaling and rotation.

The conversion assumes that in the small scale longitudes and latitudes form the Cartesian coordinate system and requires defining two parameters. An orientation of the local coordinate system is the first one. It is characterized by the angle θ between the vertical axis of the local coordinate system and the North. Finding this parameter with enough accuracy usually do not pose any problems as the axes are typically placed along significant, linear features in the scene (e.g. roads) so it may be measured directly (e.g. with a compass).

The second parameter is formed by one fixed, reliable relation between local and GPS coordinates. It contains one reference point in the local system (x_{ref}, y_{ref}), for which the longitude and the latitude (lon_{ref}, lat_{ref}) is known. Unless there is an access to the very precise, cartographic data, this relation is prone to inaccuracy because typical GPS receivers do not provide required precision. However, this inaccuracy does not have a major impact on the results of PTZ camera tracking and may be ultimately eliminated by manual tuning based on the tracking result observation.

Conversion of any point (x, y) in the local, real-world coordinates to GPS coordinates consists of four steps. First the point is translated in order to move the origin of the coordinate system to the reference point (x_{ref}, y_{ref}). It is accomplished with the following equation:

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_{ref} \\ y_{ref} \end{bmatrix} \tag{1}$$

Next, the point (x_t, y_t) is rotated around the new origin to orientate the vertical axis of the local system along the North-South direction and the horizontal axis – along the East-West direction:

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \end{bmatrix} \tag{2}$$

The third operation involves scaling coordinates of the point (x_t, y_t) according to the equation:

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} \frac{1}{D_{lon}} & 0 \\ 0 & \frac{1}{D_{lat}} \end{bmatrix} \cdot \begin{bmatrix} x_r \\ y_r \end{bmatrix} \tag{3}$$

where D_{lon} and D_{lat} denote surface distance (in meters) per 1° change in longitude and latitude for a given latitude. Calculation of these parameters is described in section 3.3.1.

The last step allows obtaining GPS coordinates (lon, lat) of the point (x, y) in the local system by the means of the following translation:

$$\begin{bmatrix} lon \\ lat \end{bmatrix} = \begin{bmatrix} lon_{ref} \\ lat_{ref} \end{bmatrix} + \begin{bmatrix} x_s \\ y_s \end{bmatrix} \tag{4}$$

The GPS coordinates (lon, lat) of a moving object may now be used to fix any PTZ camera (providing the object is in its area of coverage) on the object. This process is facilitated by providing current speed and bearing of the object.

The speed of the object v is easily calculated based on the difference of the current object position (x, y) in the local coordinate system and the position in the previous video frame (x_{-1}, y_{-1}):

$$v = \sqrt{(x - x_{-1})^2 + (y - y_{-1})^2} \cdot fps \tag{5}$$

where *fps* is the number of video frames processed in a second.

Bearing of the object *br* is obtained with the following formula:

$$br = \phi - (90 - \operatorname{atan2}(y - y_{-1}, x - x_{-1})) \tag{6}$$

If an object is not moving (its speed $v = 0$), bearing *br* is unknown.

3.3. PTZ camera steering

Efficient object tracking with PTZ cameras requires two steps to be performed. First of all, an algorithm for conversion of object's GPS position to appropriate settings of a PTZ camera needs to be implemented. Furthermore, each camera must be calibrated in order to be able to lock on a moving object with high enough accuracy.

3.3.1 Conversion of object's GPS data to PTZ camera settings

This section describes the algorithm used to fix a PTZ camera on a nearby object based on its known GPS data [5]. The aim of the algorithm is to calculate pan, tilt and zoom parameters for a dome camera to guarantee that an object of known longitude and latitude will be present in video frames. Because of a very low accuracy of altitude data from a GPS receiver, it is assumed that all objects move on the same altitude equal to the ground level. Another assumption is that in a small area around a PTZ camera, longitude and latitude form a Cartesian coordinate system.

The algorithm has four settings regarding the camera: its longitude lon_C, latitude lat_C (both in degrees), height above the ground level h_C and pan offset p_{off}, which is defined as a clockwise distance (in degrees) between the north and the camera zero pan position. These settings are defined during calibration procedure described in section 3.3.2.

Input data for the algorithm include 4 parameters: longitude *lon* and latitude *lat* of the object (in degrees), its speed *v* and bearing *br* (in degrees). They are calculated by the means of video processing of object images captured by a fixed camera.

In the first stage of the algorithm, local, Cartesian 3D coordinate system is defined with the camera in its center. *Y*-axis of this system is headed north and *x*-axis is headed east. Camera position is therefore denoted as $(0, 0, h_C)$ and object position is described as $(x, y, 0)$. Object's *lat* and *lon* coordinates are translated to x, y coordinates as follows:

$$x = (lon - lon_C) \cdot D_{lon}(lat_C) , \qquad y = (lat - lat_C) \cdot D_{lat}(lat_C) \tag{7}$$

where D_{lon} and D_{lat} denote surface distance (in meters) per 1° change in longitude and latitude for a given latitude. Because of a relatively small distance between the camera and the object, it is assumed that D_{lon} and D_{lat} parameters for latitudes of the camera and the object are the same and are equal to the ones of the camera. They are calculated according to the WGS84 ellipsoid used by all GPS devices [15]. D_{lat} value changes from 110574 m at 0° latitude to 111694 m at 90° latitude and is approximated with the following polynomial:

$$\begin{aligned} D_{lat}(lat) = & -1.70706 \cdot 10^{-9} \cdot lat^6 + 7.02332 \cdot 10^{-7} \cdot lat^5 - 1.01783 \cdot 10^{-4} \cdot lat^4 \\ & + 3.20988 \cdot 10^{-3} \cdot lat^3 + 0.267432 \cdot lat^2 + 0.575556 \cdot lat + 110574 \end{aligned} \tag{8}$$

D_{lon} value for a given latitude is obtained from the following equation:

$$D_{lon}(lat) = \frac{\pi}{180} \cdot \cos(lat) \cdot \sqrt{\frac{a^4 \cos(lat)^2 + b^4 \sin(lat)^2}{(a\cos(lat))^2 + (b\sin(lat))^2}} \tag{9}$$

where a = 6378137 m and b = 6356752.3 m are Earth's equatorial and polar radii, respectively.

There is a significant delay in the system of tracking video objects with PTZ cameras, caused by data transmission, computing time and executing PTZ command by the camera. This delay must be compensated for in order to assure that a fast-moving object is always present in a video frame center.

System delay compensation is performed by setting the PTZ camera to the predicted position of the object. Prediction time should be equal to the total delay in the system. In the current implementation, a linear predictor is used that estimates object position based on its instantaneous speed v and bearing br:

$$\hat{x} = x + d \cdot v \cdot \sin(br) , \qquad \hat{y} = y + d \cdot v \cdot \cos(br) \tag{10}$$

where d is the delay of the system and was set to 1 second based on experimental results.

The pan parameter p for the camera is calculated as follows:

$$p = 90 - \operatorname{atan2}(\hat{y}, \hat{x}) - p_{off} \tag{11}$$

and translated to the range required by the PTZ camera.

The tilt parameter t is given with the equation:

$$t = \begin{cases} -\arctan\left(\dfrac{h_C}{\sqrt{\hat{x}^2 + \hat{y}^2}}\right), & \text{if } \hat{x} \neq 0 \vee \hat{y} \neq 0 \\ -90, & \text{otherwise} \end{cases} \tag{12}$$

The last camera parameter, the zoom, depends on the object's distance and speed. The closer the object is to the camera and the faster it moves, the smaller is the zoom value. This approach assures that object dimensions

in video remain more or less constant and the object is always present in video frames although the camera position is changed once per second.

3.3.2 PTZ camera calibration

In order to achieve satisfactory accuracy of moving object tracking, 4 camera parameters (lon_C, lat_C, h_C, p_{off}) must be defined with a very high precision. Initial experiments proved that simple measurement of camera position with a GPS receiver does not provide the required precision. Also, determining camera's altitude above the ground level might be problematic. Theoretically, the easiest parameter to obtain is the camera pan offset, however the precision required (especially when tracking distant objects) makes any direct measurement very difficult.

Therefore, a one-time optimization approach was chosen to estimate camera parameters with better precision [5]. Initial approximations of four parameters are obtained with direct measurements. All parameters are then further tuned during two-stage, non-linear optimization process, which minimizes iteratively the cost functions describing the differences between pan and tilt values calculated by the algorithm and the ones measured directly. Localizations of N points scattered equally in camera vicinity were measured with a GPS receiver for this purpose. In the same time, pan and tilt values were obtained from the camera pointing at each spot. The camera zoom does not influence pointing accuracy, thus this parameter is omitted from the optimization process.

The number of calibration points should not be too small because reference data gathered with GPS receiver are often inaccurate and their precision is low. During experiments, 13 calibration points were used.

In the first stage of the optimization process, 3 parameters responsible for object tracking in the horizontal plane (lon_C, lat_C, p_{off}) are estimated. The cost function for this stage is defined as follows:

$$E_1\left(lon_C, lat_C, p_{off}\right) = \sum_{i=1}^{N} w_i \cdot \left(p_i^c - p_i^r\right)^2 \tag{13}$$

where p_i^c is a pan value calculated according to Eq. (11), p_i^r is a real pan value obtained from the camera for the ith calibration point and w_i is the weight of the calibration point. The weight of a point is directly proportional to its distance from the camera. This assures that the farther the point is from the camera, the greater is its influence on the results of optimization.

During the optimization procedure utilizing conjugate gradient method [1], lon_C, lat_C and p_{off} values are tuned in order to minimize the value of E_1 function. At the second stage of calibration process, the h_C value responsible for object tracking in the vertical plane is tuned based on calibration data and on 3 parameters found during the first stage. The cost function for this stage is given as:

$$E_2\left(h_c\right) = \sum_{i=1}^{N} w_i \cdot \left(t_i^c - t_i^r\right)^2 \tag{14}$$

where t_i^c is a tilt value calculated according to the Eq. (12) and t_i^r is a real tilt value obtained from the camera for the *i*th calibration point. The optimization procedure alters h_C value in order to minimize value of E_2 function.

4. Automatic selection of tracked objects using event detection

The system described in this paper allows for tracking of a single moving object with the PTZ camera. The question that arises here is: how is the tracked object selected? A simplest possibility is the manual selection of the object by the system operator, e.g. by pointing the selected object with the cursor. However, in practical system, several sets of cameras may be used at the same time, so the unattended camera sets should be able to select the tracked objects automatically. In this section we describe the system that automatically analyses the events occurring in the fixed camera image and selects the object to be tracked.

The low-level image processing stages (described in Section 3.1) lead to determining the position and size of each moving object in the consecutive fixed camera frames. Additionally, the object classification module assigns each object to the class (e.g. a vehicle, a person, an object) and the subclass (e.g. a passenger car, a truck, a motorcycle, etc.). The task of the event detector is to analyse the current states of the moving objects and to detect predefined events by testing the respective detection rules. Examples of typical events include an object entering a defined area, an object crossing the barrier, an object stopping or resuming its movement, etc. [13]. If the predefined event is detected in the current image frame, the PTZ camera may be automatically set to track the object that caused the event.

The structure of the event detector is organized as follows. The current state of each tracked moving object (its position, size, velocity, etc.) is stored in a buffer, together with a number of previous states. In the current implementation, a total of 5 latest states (including the current one) is stored in the circular buffer. Each event detection rule works by analysing all the states stored in the buffer and checking whether a defined rule is fulfilled. Using the history buffer in event detection improves its accuracy, because the 'momentary' events, occurring in single frames only (e.g. due to inaccuracy of object tracking) are filtered out. A set of simple event rules, implemented in the current version of the system, is presented below:

- *An object entering (or leaving) an area*. This event may be useful e.g. for the detection of intrusion. An area is defined using a polygon. The detector checks whether the 'hot spot' of any object (the middle of the bottom border of the tracker's bounding box) is inside any of the defined areas during all the states stored in the buffer. The point is inside the area if it is situated on the 'inside' of all the lines that form the area polygon. If the object is inside the area and it was previously outside of it, an event is generated. An object leaving an area is detected in a similar way. Additionally, identifiers of the objects that are inside each area are stored in buffers in order to avoid multiple detections of the same event.
- *An object crossing a barrier*. The barrier is defined as a straight line. Additionally, some directions of movement through the barrier may be defined as allowed and some as forbidden. The detector checks whether the line of object's movement, connecting the oldest position stored in the buffer of states with

the current position, crosses the border and if the direction of movement is forbidden. If this is the case, an event is detected.

- *An object stopping or starting moving*. The event rule checks whether the averaged speed (calculated as a moving average of momentary object's speed during the last 10 image frames) is below the threshold (for the stopping object) or above the threshold (for the object that started moving) for all states stored in the buffer.
- *Abandoned or taken object*. This rule detects objects that are left in the observed area or that were removed from the background. It is intended to detect e.g. unattended luggage or stolen objects. The detector checks if there is a new tracker which position and size did not change during all the states stored in the buffer. Additionally, this object needs to have a 'parent' – another object that caused its appearance. If these conditions are fulfilled, the detector checks whether the tracker represents an actual (abandoned) object or a part of the background remaining after the object was taken. This is done by performing edge detection on the relevant part of the image. If the contour of the tracked object's mask contains the edges, it is an abandoned object, otherwise it is a space left by a taken object.
- *Object entering and leaving the observed area* – a simple detector that finds objects that appeared on the scene or disappeared from it, this rule may be combined with the others.

These rules may be combined in order to form more complex, higher level rules. For example, a rule may detect objects of class 'passenger car' that stopped inside a defined area. The system for event detection is flexible and new event rules may be easily added.

One main problem that needs explanation here is what happens if a new event is detected and the PTZ camera already tracks another object (or what happens if more than one event is detected in the current frame). In order to avoid conflicts and constant switching of the PTZ camera between different objects, a weight is assigned to each event detection rule, indicating its priority. For complex rules, the weights of their components are summed. If the PTZ camera is already tracking an object and the new event is detected, the camera switches to the newly detected event only if its the rule has higher priority than the previously used one. All the detected events are stored in a list, sorted by their priority, and the rule at the top of the list determines which object is tracked. If the tracked object disappears from the camera's field of view, its rule is removed from the list and the rule that becomes the new 'top one' indicates the object that the PTZ camera should switch to. This simple procedure allows the system to automatically select the part of the fixed camera's field of view in which the most important event occurred and use the PTZ camera to provide a detailed view of this event.

5. Presentation of tracked object on map

Having selected the tracked object and knowing its estimated GPS position allows for presenting it on a map updated in quasi real-time. This functionality is called a "live map" and is an important feature of the described

system. In a practical, complex, surveillance solution it constitutes a base for visual user interface, allowing for direct interaction with tracked objects and clear indication on their relative positions.

For the purpose of visualisation and practical evaluation of aforementioned object tracking algorithms, an example "live map" has been developed as a Flash object, which may be embedded either as a part of web page, or directly within host application, possibly adding more advanced features to the map. Both modes of map operation have their specific uses. Embedding in a web page is more suitable method for presentation of information which is regarded "public", while the interaction with host application enables the implementation of more advanced security and management mechanisms and allows the reuse of common communications modules providing tight integration with the rest of system. This mode has been specifically designed for creating a surveillance central station application, which aims at coupling geolocation data to multimedia streaming, thus making it more expressive and easier to comprehend.

The choice of Flash technology for development of presentation layer seems to be justified by its inherent design towards processing of vector graphics that significantly reduces amount of work needed to display digital maps and manage layers thereof. However, this design affects communication abilities of the application. Flash objects are subject to security policies affecting their ability to maintain network connections with arbitrary services, therefore they require careful configuration. In order to make it possible for the Flash application to receive notifications on update of presented data it was necessary to implement system module translating event description data to so called *XmlSocket* protocol and featuring Flash-specific mechanism for provisioning of security policy data.

Figure 4 shows sample view of the "live map" presenting locations of tracked objects gathered from multiple node stations. The objects are denoted by encircled car symbol (three of them near the center, one on bottom-right).

6. Experiments

For the purpose of the experiments, a test bed of the surveillance system was build. It consists of the central server, a system operator desk and one node station. The station is equipped with a megapixel, wide angle, fixed camera and with a PTZ camera. The system operator sees live streams from the both video cameras. All moving objects detected in the fixed camera video stream are marked with rectangles. The system operator is able to select any moving object with a computer mouse and the PTZ camera is automatically and instantly aimed at the object and tracks it as long, as the object stays in the fixed camera field of view.

The main purpose of the experiments was to estimate accuracy of tracking, i.e. whether the object being tracked is positioned near the center of video frames from the PTZ camera. Sample results of experiments are shown in Fig. 5.

There is a noticeable time-delay in images from the fixed camera comparing with the PTZ camera. It is caused by the buffering mechanism in the video processing algorithms. However, the PTZ camera is controlled

by the object position acquired from the latest image frame, therefore the buffering has no impact on the accuracy of PTZ camera tracking.

The results of experiments show that an object being tracked is always present in the PTZ camera frames. However the accuracy of tracking is not perfect (i.e. the camera is seldom pointed exactly at the object). This inaccuracy is caused by the imperfect calibration data and limits the maximum value of the zoom parameter. An impact of each calibration procedure on the results of object tracking needs to be further studied. Despite of the inaccuracy, PTZ object tracking is fully functional and meets the expectations.

Object position predicting turned out to be the crucial element of the system, especially when higher zoom levels are used. If this functionality is turned off, tracking of moving vehicle is not possible, because when the PTZ camera is being set to a new position, a vehicle is just leaving its new field of view.

7. Conclusions

The system for moving objects tracking was presented. It enables automatic steering of PTZ cameras for tracking of objects that are simultaneously observed by at least one fixed camera. It can be easily extended with higher number of cameras, and integrated with other image content analysis algorithms. Currently a face detection algorithm is being incorporated, for automatically obtaining tracked person en-face close-up image for database storage.

Acknowledgements

Research is subsidized by the Polish Ministry of Science and Higher Education within Grant No. R00 O0005/3 and by the European Commission within FP7 project “INDECT” (Grant Agreement No. 218086).

References

[1] M. Avriel, *Nonlinear Programming: Analysis and Methods*, Dover Publishing, 2003.

[2] A. Czyzewski and P. Dalka, Visual Traffic Noise Monitoring in Urban Areas, *International Journal of Multimedia and Ubiquitous Engineering* **2** (2007), 91–101.

[3] A. Czyzewski and P. Dalka, Examining Kalman filters applied to tracking objects in motion, in: *Proc. of 9th International Workshop on Image Analysis for Multimedia Interactive Services*, Klagenfurt, Austria, 2008, pp. 175–178.

[4] P. Dalka, Detection and Segmentation of Moving Vehicles and Trains Using Gaussian Mixtures, Shadow Detection and Morphological Processing, *Machine Graphics and Vision* **15** (2006), 339–348.

[5] P. Dalka, A. Ciarkowski, P. Szczuko, G. Szwoch and A. Czyżewski, Surveillance Camera Tracking of Geo positioned Objects, in: *2nd International Symposium on Intelligent Interactive Multimedia Systems and Services*, Mogliano Veneto, Italy, 2009.

[6] N. Funk, *A Study of the Kalman Filter applied to Visual Tracking University of Alberta*, Project for CMPUT 652, 2003.

[7] T. Horprasert, D. Harwood, L. Davis, A statistical approach for real-time robust background subtraction and shadow detection, in: *Proc. of IEEE Frame Rate Workshop*, Kerkyra, Greece, 1999, pp. 1–19.

[8] J. Konrad, Videopsy: Dissecting Visual Data in Space-Time, *IEEE Communication Magazine* **45** (2007), 34–42.

[9] H. Li, K. Ngan, Automatic Video Segmentation and Tracking for Content-Based Applications, *IEEE Communication Magazine* **45** (2007), 27–33.

[10] Y. Liu, Y. Zheng, Video Object Segmentation and Tracking Using y-Learning Classification, in: *IEEE Trans. Circuits and Syst. For Video Tech.* **7** (2005) 885–899.

[11] J. Martínez-del-Rincón, J.E. Herrero-Jaraba, J.R. Gómez, C. Orrite-Uruñuela, Automatic left luggage detection and tracking using multi-camera UKF, in: *Proc. 9th IEEE Internat. Workshop on Performance Evaluation in Tracking and Surveillance* (PETS '06), New York, USA, 59–66, 2006.

[12] PETS 2006 – a collection of test recordings from the Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, New York, USA, 2006.

[13] G. Szwoch, P. Dalka, A. Czyzewski, Objects classification based on their physical sizes for detection of events in camera images, *NTAV/SPA 2008*, Poznan, pp. 15-20, 2008.

[14] R.A Tsai, A Versatile Camera Calibration Technique for High Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses, *IEEE Journal of Robotics and Automation* **RA-3** (1987), 323-343.

[15] G. Tyler, G. Blewitt, *Intelligent Positioning: GIS-GPS Unification*, Wiley, Chichester, UK, 2006.

[16] L. Xiu, J. Landabasso, M. Pardas, Shadow removal with blob-based morphological reconstruction for error correction, in: *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, II–729–732, 2005.

[17] T. Yang, S. Li, Q. Pan, J. Li, Real-Time and Accurate Segmentation of Moving Objects in Dynamic Scene, in: *ACM Multimedia-2nd International Workshop on Video Surveillance and Sensor Networks*, New York, 2004, pp. 10–16.

Tab. 1. Possible relations between trackers and regions denoting moving objects and actions taken in each case

Type of relation	Action
Region without a tracker	New tracker created based on the region properties
Tracker without a region	Only prediction phase carried out; if the tracker fails to relate to a region within several subsequent frames, it is deleted
One tracker – one region	Tracker is updated with the parameters of the region
One tracker – many regions	Tracker is updated with the parameters of the rectangle covering all regions
Many trackers – one region	Each tracker is updated with the parameters of the same region.
Many trackers – many regions	Iterative analysis of region-tracker clusters and updating a tracker with the most suitable region based on visual similarity.

Fig. 1. Frames illustrating continuous tracking of many persons passing by each other. Frames number: a) 1055, b) 1071, c) 1091, d) 1013, taken from benchmark video footage S1-T1-C3 recording from PET 2006 set [12]

Fig. 2. Example of calibration of the fixed camera performed during the experiments, using landmarks (marked as dots on the image) as calibration points

Fig. 3. An illustration of the problem of determining the position of the calibration point in the low-contrast camera image: 1:1 crop of the camera image (left) and its part (marked by the rectangle) in a 3:1 zoom view (right); finding the point where the post touches the ground is problematic

Fig. 4. Application displaying tracked objects on a city map

Fig. 5. Two sample results of PTZ camera tracking (in columns); top row: segments of video frames from a fixed camera with moving objects detected (colour rectangles) and an object selected by an operator (white circle); bottom row: video frames from a PTZ camera automatically aimed at the object selected by the operator


a)


b)


c)


d)