

# Motion-based Flexible Camera Registration

Benedek Cs, Havasi L  
Péter Pázmány Catholic University,  
H-1083 Budapest Práter u. 50/a., Hungary  
{Benedek, Havasi}@digitus.itk.ppke.hu

Szirányi T, Szlávik Z  
Computer and Automation Research Inst.  
H-1111 Budapest, Kende u. 13-17, Hungary  
{Sziranyi, Szlavik}@sztaki.hu

## Abstract

*We outline our latest methods and algorithmic solutions for detection of concurrent motions and registering cameras in real-life surveillance systems. We show that cameras can be registered in several rather unfavorable conditions, based on:*

- *Unpredictable motion without structured background or defined object shapes or*
- *Walking persons of undefined silhouettes and short detectable walking distances or*
- *Shadows of undefined structures in front of flickering background.*

*These methods are tested in real-life sequences and we show that flexible 3D camera registration is possible even in bad lighting conditions and in the lack of any known structures or motions.*

## 1. Introduction

Registration between partially overlapping views of the same scene is a fundamental task in a number of applications involving multi-camera systems, such as stereovision, three-dimensional reconstruction, or object tracking/observation in surveillance systems. Locating more and more cameras for surveillance systems there is a demand to flexibly register them each to the other without manual intervention, including continuous reregistration and position testing.

In the literature of computer vision, many examples about how the registration of different views has been achieved are described, together with the associated problems. The existing methods can be divided into two groups: those which are still-image based, and motion-based ones. Still-image based algorithms, e.g. those used in [1][2][3][4], attempt to match static features in images, such as edges, corners, contours, color, shape etc. They are often used for image pairs with small differences, when the difference between features is negligible. However, they may fail at

occlusion boundaries, within featureless regions, and if the chosen primitives or features cannot be reliably detected. Wide-baseline camera positions also call for more distinguished analysis. The other group, the motion-based methods such as [5][6], try to find matchings between different views by analyzing the dynamics of the scene as recorded by the different cameras. In [5], the tracks of moving objects are the basic features for the matching of the different views. In this case the capability of robust object tracking is assumed and this is the weak point of the method. In [6], a method is reported which finds co-motion point-pairs in the videos as recorded from the same scene by different cameras. Both these methods assume that the objects of interest are moving on the ground-plane; and also that the cameras are far distant from the scene, so that the height of the moving objects is small enough for them to appear in the recorded videos as “moving blobs on the ground”.

In practice, the existing algorithms can be used only in restricted situations. The reported methods focus on the solution of the view-registration problem in respect of outdoor scenes, and neglect the additional difficulties, which tend to arise for indoor scenes. In case of indoor cameras, the still-image based methods may fail due to the variability of conditions: occlusions, changing illumination etc. Due to the larger size of the moving objects, the cited motion-based methods will also fail; the observed motions are not necessarily on the ground-plane – while for outdoor scenes, such an assumption can safely be made.

The aim of this paper is to present algorithms for indoor and outdoor registration of cameras considering unfavorable conditions. Our methods are based on the detection of concurrently moving points in the recorded videos.

We considered several situations where earlier methods cannot work:

- Registration based on co-occurrence of motion points in case of unpredictable motion without

structured background or defined object shapes in outdoor circumstances;

- Registration of cameras by detecting the leg-ground joint in case of walking persons of undefined silhouettes and short detectable walking distances;
- Finding points for registration from shadows of undefined structures in front of flickering background.

We demonstrate our new methods in real-life indoor and outdoor video sequences and we show that flexible 3D camera registration is possible even in bad lighting conditions and in the lack of any known structures.

## 2. Registering camera views

If the observed motions are on the groundplane then the relation between images of the same scene can be modeled as a homography [11]; it takes points from one image and transforms them to the corresponding points in the other image.

Finding corresponding points is not an obvious task in a non-model assumption. When different (mostly unpredictable) objects are moving in the scene, the altitude of a given point is uncertain: background or shadow, or some object hiding the background. There are several methods to find some predefined silhouettes and estimate the altitudes (e.g. considering typical human body shape), but in a cluttered environment it is mostly a tautological question. For this reason our registering methods are based on definitions with the less preliminary definitions: statistics of co-motion, pattern of symmetry-series in walking and shadows.

The estimation of the homography between two views in the presented algorithms is based on the extraction of concurrently moving points in the two views [7] that were projected into cameras as images of the same moving object. The general scheme of the proposed algorithms is the following:

1. Background modeling.
2. Detection of features.
3. Extraction of point-correspondences – extraction of candidates, rejection of outliers.
4. Alignment of the cameras' views.

In the very general case, when there are different unpredictable motions on the screen with static features (corners, edges etc.) cannot be reliably detected and we have time to leave the camera system to register itself, then the information for matching can be extracted by the detection of concurrently moving points [6]. Calculating co-motion statistics [6] for

given points in two videos the concurrently moving points simply could be detected, see Figure 1. However, this method needs a longer time period (5-15 minutes) to collect enough statistics, so we searched for some more definite points where we do not need strict definitions of any shapes.

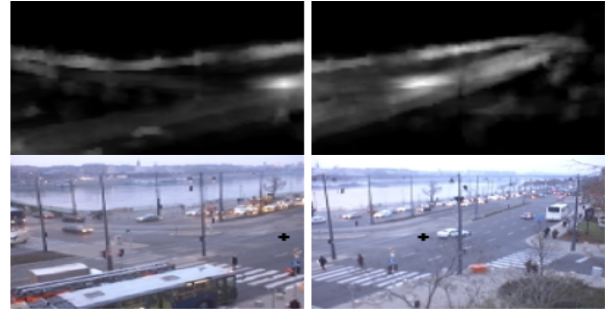


Figure 1 Top images: example of co-motion statistics for point-pairs. Below: a corresponding point-pair is shown in the images of the left and right cameras.

### 2.1. Background-modeling and shadow detection

In case of co-motion background meant something with non-motion co-occurrence. However, in other cases we need some more precise measurement of the background, to extract motion and shadow.

The background-modeling step is based on the work of Stauffer et al [12]. The algorithm collects statistics about the occurring values at each pixel position, and the recent history is modeled as a mixture of Gaussians. Each component of the mixture has the following parameters: weight, mean value and covariance matrix. The component with the greatest weight is considered the background component. The parameters are updated via online k-means algorithm. The model follows the changes in the background and the varying lightning properties adaptively, but the resulting binary silhouette image often contains the shadows as part of moving objects, like on Figure 2b.

We needed shadow detection for two reasons. For motion analysis we wanted to get the accurate shapes of the objects, and we used the collected shadow points for the camera registration (section 4.3).

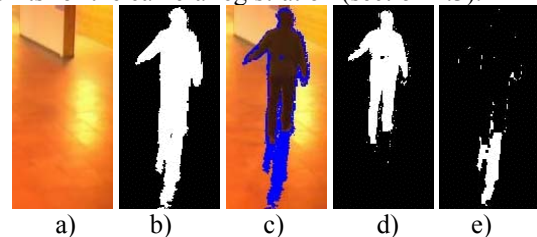


Figure 2. a) Background image and b) silhouette image created by Stauffer-Grimson algorithm c) Result of shadow detection with 'Sakbot' method (blue color)

marks shadow points) d) Object silhouette after removing the shadows e) Filtered shadow mask used for camera registration

As shadow detector we used the ‘Sakbot’ [13] method. It works in the HSV color space, which corresponds closely to the human perception of color. The method exploits that the occlusion of cast shadow darkens the background pixel and saturates its color. As it can be observed in Figures 2c and 2d, the shadow pixels were found well on the images. However there were false positive shadow occurrences especially around the body, so the shadow mask used in section 4.3 was filtered by an erosion step, see Figure 2e.

## 2.2. Co-motion based feature extraction

Due to the relative small size and large distance between cameras and moving objects in outdoor scenes we can assume that the objects are moving on the groundplane. By extracting the concurrently moving points from the two given images the point correspondences can be estimated and the registration of the views could be done [6].

Having the result of change detection the scene dynamics can be coded and stored. For each pixel the motion history is stored in a vector, which has as many entries as long is the video sequence and in each of its entry has 1 if change was detected at the given frame or 0 if not.

**2.2.1. Feature detection.** From the images of the two views we extract feature points related to pixels of real objects (cars, people etc.) moved through them. We don’t want to extract pixels in which change was detected due to flashings or random noise on the background. For the extraction of these points we integrated the motion histories. If this value is above some threshold then the corresponding pixel is selected as a feature point.

It is obvious that if all of our candidate points are from the same region of input images and close to each other then small error in point coordinates (which comes from the change detection, which is, of course, not perfect) will result in great error in final alignment of the whole images. To reduce it we forced points to be better distributed in the region by introducing some structural constraints: images are divided into blocks of  $n*n$  and for each block the algorithm searches for only one candidate point, for which the integrate of motion history is the maximal.

**2.2.2. Extraction of candidate point pairs.** Having the features points detected in both views for the extraction of candidate point-pairs the feature points of

different views must be compared. First, the time-series of the history-vectors are filtered. This morphological filter removes single peaks and groups neighbor peaks if they are within a predefined distance. After filtering the Hamming distance is calculated as correlation between two binary motion history vectors of different views.

## 2.3. Human’s motion based feature extraction

Motion based methods [5][6] focus on the solution of the view-registration problem in respect of outdoor scenes, and neglects the additional difficulties, which tend to arise for indoor scenes. Due to the larger size of the moving objects, the cited [5][6] motion-based methods will also fail; the observed motions are not necessarily on the ground-plane – while for outdoor scenes, such an assumption can safely be made. Our method for the registration of views of indoor scenes is based on the extraction of features that are surely on the groundplane.

**2.3.1. Symmetry patterns generated by human motion.** The feature extraction method proposed here is based on the detection of human motion-activity (namely, walking) in the scene. This task is a binary classification problem: the periodicity of human walking, together with the characteristic human shape of the target, provide key differences which enable us to distinguish legs of pedestrians from the motion-patterns of other objects. For the detection of human walking patterns we have previously introduced a simplified symmetry-extraction algorithm [8], and have also described an effective method for the tracking and detection of human motion [9][10].

Sample results of the image-processing steps are shown in Figure 3, which illustrates the results of the algorithm steps up to the stage of symmetry-pattern extraction from the input image. It was shown [7][10] that the extracted symmetry pattern is a very characteristic pattern and it can be used for the detection of walking humans.

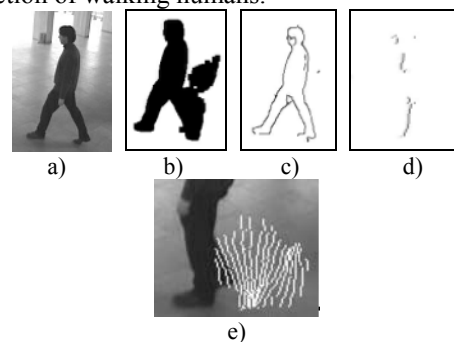


Figure 3. Overview of walk detection method: input image, change detection mask, canny edge map, third level symmetries (L3S), coherent masks in a sequence and the symmetry-pattern

There are two major advantages of this feature-extraction approach: the extracted symmetry-pattern reflects the change of structural properties of the object; and secondly, its computation is extremely fast, and hence it can be readily used in real-time applications. In our experiments, we found that the most critical factor is the image refresh rate: we found that a rate of at least 10 frames/second is required. However this requirement can be easily met, since the method can run at 20-40 FPS on a simple desktop PC (the achievable rate depends on the number of tracked objects).

**2.3.2. Classification of symmetry patterns.** A recognition process is done on the symmetry-series to classify it as a walking-pattern of human-leg or anything else. A more developed version of our previous detection method [10] operates in the eigenwalks space and utilizes the Support Vector Machine method (SVM) for pattern classification [7]. The classification process is carried out using a non-linear classification method, namely SVM with radial basis kernel function. In our tests, the correct-classification rate of walking was found to be 92%.

**2.3.3. Feature extraction – identification of the leading leg.** The 2D motion vector on the image-plane, and the walker’s gait-period, can be extracted directly from the detected patterns: we estimate the motion vector by fitting a regression line to the last half-trajectory of the lower two points of the pattern. In this section we present a method to determine, from one detected walk cycle, whether the leading leg is the right or the left leg. According to our terminology, the leading leg is the “standing” leg, which at that instant carries the person’s weight (see Figures 4b and 4c).

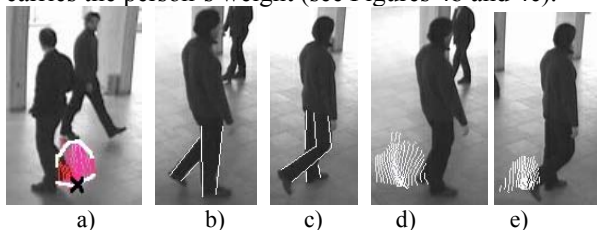


Figure 4. a) An image showing the location of the derived symmetry-pattern (marked with white border; “x” marks a feature-point, see Section 4). b), c) Illustrations of our definition of “leading leg”; the “standing” or leading leg is the right leg in b), and the left leg in c) (legs highlighted manually). d), e) The detected patterns for the same steps as shown in b)

and c); the 2D direction is bottom-left to upper-right (case 2 in Table 1).

Depending on the 3D walk-direction, and on which is currently the leading leg, one leg or the other practically obscures the visible area between the legs (Figure 4a). During a walk-cycle the ratio of the areas, together with the 2D direction on the image-plane, can be used to identify which is the leading leg. During one cycle, the left leg and right leg in turn are in the leading position. The above-described method can detect one step. To connect two successive steps as one walk-cycle, we calculate the 2D displacement vector of a detected step, and then searching for another step in the estimated 2D position and at a time-point after a forecast walk-period. Table 1 summarizes the relationship between the leading leg and the ratio of surfaces from two successive patterns defined by:

$$ratio = \frac{area_t}{area_{t-1}} \quad (1)$$

A limitation of the described method is that it cannot identify the leading leg when the motion is parallel to the camera plane, since in such cases the areas are nearly equal (cases 3, 4 and 9, 10 in Table 1).

Table 1. Surface-area dependencies on the 2D walk-direction (6 of 8 possible cases) and the leading leg (Right, Left).

Case	2D Direction	Leading leg	Ratio
1		Right	>1
2		Left	<1
3		Right	≈1
4		Left	≈1
5		Right	<<1
6		Left	>>1
7		Right	<1
8		Left	>1
9		Right	≈1
10		Left	≈1
11		Right	>>1
12		Left	<<1

**2.3.4. Extraction of candidate point pairs.** To detect corresponding points, we use our walk-detection and leading-leg identification methods. Both methods provide information, which is useful in matching points between the two views: detected walk patterns must be concurrent in both views; and, likewise, the leading leg must be the same. In both views the central lower points of the detected walk-patterns are the feature points (e.g. the one marked with a black “x” in Figure 4a). The extraction of a feature point from one of the views is followed by searching for its pair in the other view. The algorithm searches for concurrent

points by examining the timestamps of points, and for points, which were detected during walk cycles with the same leading leg.

#### 2.4. Shadow based feature extraction

The shadow based feature extractor module is the same as the co-motion based one. The only difference is that the output of the shadow-detector is the input to the co-motion based feature extractor module instead of the motion-detector. In case of shadows could be detected they are excellent features to match in a relatively short target-distance (mainly indoor), because they are mainly on the ground.

### 3. Robust estimation of scene homography

For the estimation of transformation  $H$  that maps points of one scene onto another and rejection of outliers from the set of candidate point-pairs we have implemented the RANSAC algorithm [11]. In our experiments people and cars are moving on the groundplane. In this case  $H$  is a projective transformation that can be represented by a  $3 \times 3$  matrix and can be calculated from at least 4 point-pairs.

## 4. Experimental results

All the proposed algorithms were tested on real-life video sequences. The major assumption is the time synchronization between the cameras. When it exists, the concurrently moving points could be extracted from images. We have shown before that by using some further processing this assumption can be avoided [6].

#### 4.1. Co-motion based alignment

The above-described algorithm was tested videos captured by two cameras, having partially overlapping views at Ferenciek (FERENCIEK videos) square in Budapest, which have been captured at resolution  $320 \times 240$ , at different zoom levels and with different cameras.



Figure 5. Final alignment of two views for the FERENCIEK videos.

#### 4.2. Symmetry pattern based alignment

We evaluated the registration algorithm by using surveillance cameras placed in a public area located in the university building. In this experiment the angle between the view-axes of the two cameras employed was nearly  $90^\circ$  (hence, to detect corresponding points using standard optical methods would be difficult). The videos are captured at resolution  $320 \times 240$ , at different zoom levels and with different cameras. The algorithm searches for concurrent points of symmetry patterns by the timestamp of points and for points, which were detected during walk cycles with the same leading leg.

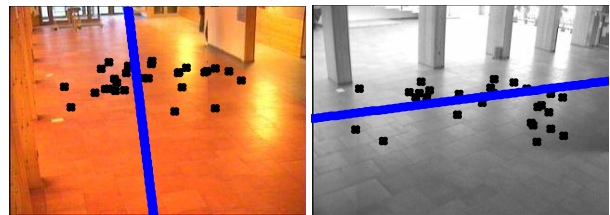


Figure 6. Transformation from the first-camera view (left) to the second (right): Detected corresponding points, and a synthetic line-trajectory.

#### 4.3. Shadow based alignment

In the next experiment we used the implemented shadow detection method and the condition of concurrency for the extraction of feature points.



Figure 7. Alignment of views based on the detection of concurrently moving shadows.

Shadows are excellent features for indoor scenes, because they are mainly on the ground. The results of the alignment of two images are shown in Figure 7. Note that tiles are matched at the boundaries of the two images.

#### 4.4. Walking pedestrian based alignment of non-overlapping views

In the last experiment we aligned images of cameras with non-overlapping field of view. The schematic map of experiment is shown in Figure 8.

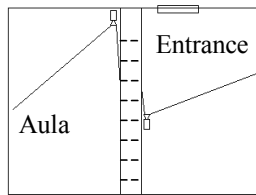


Figure 8. Schematic map of the experiment: placement of cameras and their field of views.

The images of the Aula and the Entrance cameras are shown in Figure 9.



Figure 9. Images of Entrance and Aula cameras with control lines on the ground.

It can be seen that the field of views of the cameras are not overlapped because of the wall between Aula and Entrance areas but virtually they do. The estimation of the homography is based on line correspondences and not on point correspondences as in previous experiments. Two successive walking steps were detected and a line was calculated through them. The major assumption in this experiment is that people are moving along straight lines from Aula to Entrance and vice versa. Every line from one view was paired with every line in the other view and the RANSAC algorithm was used for the estimation of the model and rejection of outliers. The results of aligned images are shown in Figure 10.

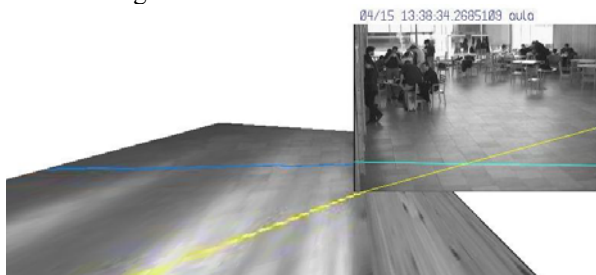


Figure 10. Result of alignment of non-overlapping views.

It can be seen from Figure 10 that the alignment is not perfect (see the direction of control lines), but this preliminary result proves that non-overlapping views could be registered by using such an algorithm.

## 5. Conclusions

We have introduced some of our latest results based on the novel methods for detection of concurrent motions for camera-registration. We have shown that

cameras can be registered in several rather unfavorable conditions, based on:

- Unpredictable motion without structured background or defined object shapes or
- Walking persons of undefined silhouettes and short detectable walking distances or
- Shadows of undefined structures in front of flickering background.

These methods are tested in real-life sequences and we show that flexible 3D camera registration is possible even in bad lighting conditions and in the lack of any known structures or motions.

## 6. References

- [1] S. T. Barnard, W. B. Thompson, "Disparity analysis of images," *IEEE Trans. PAMI*, vol. 2, pp. 333-340, 1980.
- [2] J. K. Cheng, T. S. Huang, "Image registration by matching relational structures," *Pattern Recog.*, vol. 17, pp. 149-159, 1984.
- [3] J. Weng, N. Ahuja, T. S. Huang, "Matching two perspective views," *IEEE Trans. PAMI*, vol. 14, pp. 806-825, 1992.
- [4] Z. Zhang, R. Deriche, O. Faugeras, Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial Intelligence Journal*, vol. 78, pp. 87-119, 1995.
- [5] L. Lee, R. Romano, G. Stein, "Monitoring activities from multiple video streams: establishing a common coordinate frame," *IEEE Trans. PAMI*, vol. 22, 2000.
- [6] Z. Szlávik, L. Havasi, T. Szirányi, "Estimation of common groundplane based on co-motion statistics", in *Proc. of ICIAR'04, Lecture Notes in Computer Science*, Vol.LNCS 3211, pp. 347-353, 2004.
- [7] L. Havasi, Z. Szlávik, T. Szirányi, "Eigenwalks: Walk detection and biometrics from symmetry patterns", *ICIP 2005*, accepted, available at: <http://digitus.itk.ppke.hu/~havasi/kutatas/avss>
- [8] Havasi L., Szlávik Z., "Symmetry feature extraction and understanding", in *Proc. CNNA'04*, pp. 255-260. 2004.
- [9] L. Havasi, Cs. Benedek, Z. Szlávik, and T. Szirányi, "Extracting structural fragments of overlapping pedestrians", *VIIP*, 2004.
- [10] L. Havasi, Z. Szlávik, and T. Szirányi, "Pedestrian detection using derived third-order symmetry of legs", *Kluwer, ICCVG*, 2004.
- [11] R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, Cambridge University Press, 2003.
- [12] Stauffer C., Eric W., and Grimson L., "Learning patterns of activity using real-time tracking", *IEEE Trans. on PAMI*, 22(8), 747-757, 2000.
- [13] Cucchiara R., Grana C., Neri G., Piccardi M., and Prati A., "The Sakbot System for Moving Object Detection and Tracking", *Video-Based Surveillance Systems—Computer Vision and Distributed Processing*, pp. 145-157, 2001.