Image matching based on co-motion statistics

Zoltan Szlavik, Laszlo Havasi, and Tamas Sziranyi

Abstract— **This paper presents a method for matching partially overlapping image-pairs where the object of interest is in motion, even if the motion is discontinuous and in an unstructured environment. In a typical outdoor multi-camera surveillance system, an observed object as seen by separate cameras may appear very different, due to the variable influence of factors such as lighting conditions and camera angles. Thus static features such as object color, shape, and contours cannot be used for image matching. In this paper a different method is proposed for matching partially overlapping images captured by such cameras. The matching is achieved by calculation of co-motion statistics, followed by detection and rejection of points outside the overlap area and a nonlinear optimization process. The robust algorithm we describe finds point correspondences in two images without searching for any structures and without the need for tracking continuous motion. Trials using statistical motion-based image cross-registration, a robust rejection algorithm, and automatic 3D image-transformation and camera calibration on real-life outdoor images have demonstrated the feasibility of this approach.**

Index Terms—**camera calibration, co-motion statistics, image matching.**

I. Introduction

Computer-assisted observation of human or vehicular traffic movements using multiple cameras is now a subject of great interest for many applications; examples are semi-mobile traffic control using automatic calibration, or tracking of humans in a surveillance system. In case of scenes including several objects in random motion, successful 3D registration of images from separate cameras conventionally requires some *a priori* object definition or some human interaction. In a typical outdoor scene multiple objects, such as people and cars, move independently on a common ground plane. Transforming the activity captured by separate individual video cameras from the respective local image coordinates to a common spatial frame of reference is a prerequisite for global analysis of the activity in the scene.

To estimate the object location in a scene we must know or estimate the calibration matrices for each camera in the system. Estimating camera parameters requires a set of matching points in two or several overlapping views [1]. Matching different images of a single scene may be difficult, because of occlusion, aspect changes and lighting changes that occur in different views. Over the years numerous algorithms for static and video image matching have been proposed. Still-image matching algorithms can be classified into two categories. In *"template matching"* the algorithms attempt to correlate the gray levels of image patches, assuming that they are similar for a given object-element in the two images [2][3]. This approach appears to be usable for image pairs with small differences; however it may fail at occlusion boundaries and within featureless regions. In the other category, *"feature matching"*, the algorithms first extract salient primitives (edges or contours) from images, and match them in two or more views [4][5][6]. An image can then be described by a graph with primitives as nodes and geometric relations defining the links between nodes. Image registration is then performed by the mapping of the two graphs (subgraph isomorphism). These methods are fast in execution because the subset of the image pixels that needs to be processed is small; but they may fail if the chosen primitives cannot be reliably detected. The views of the scene from the various cameras may be very different, so we cannot base the decision solely on the color or shape of objects in the scene.

In a multi-camera observation system the video sequences recorded by cameras can be used for estimating matching correspondences between different views. Video sequences in fact contain much more information than does the static scene structure of any individual frame; in particular, in the time-dimension information is also captured about the scene dynamics. The scene dynamics are an inherent property of the scene; they are common to all video sequences recorded from the same scene, even when taken from cameras in different positions, or with different zoom-lens settings. References [7] and [8] present approaches in which motion-tracks of the observed objects are aligned. Stein et al. in [7] describe a method for estimating a time-shift and a homography between two image-sequences based on alignment of the centroids of tracked moving objects. Caspi et al. in [8] rely on detecting correspondences between object trajectories. But in these cases a robust capability for object tracking is assumed; and this is the weak point of both methods. The algorithm must assume that the speed does not change by more than a predefined amount, and that the objects in the scene are moving in a smooth fashion without discontinuities.

In our practical investigations we used standard PAL digital

This work was supported by the TeleSense Project of the Hungarian National R&D Program.

Zoltán Szlávik is with the Analogical and Neural Computing Laboratory, Computer and Automation Research Institute of Hungarian Academy of Sciences, Budapest, Hungary (corresponding author; phone: +36-1-2796181; e-mail: szlavik@sztaki.hu).

László Havasi is with the Peter Pazmany Catholic University, Budapest, Hungary (e-mail: havasi@digitus.itk.ppke.hu).

Tamás Szirányi is with the Analogical and Neural Computing Laboratory, Hungarian Academy of Sciences, Budapest, Hungary (e-mail: sziranyi@sztaki.hu).

cameras equipped with wide-angle lenses. Since the field of view is large, the size of features is correspondingly small, and the images were blurred and noisy. The common field of view of two neighboring cameras was about 30%. We tested several correlation-based algorithms intended to achieve object matching between the separate images, but they gave poor results. In case of several randomly-moving objects in the field of view, with conventional methods the 3D spatial registration of images from separate cameras usually requires some assistance, in the form of *a priori* object definition or human interaction, in order to achieve correct registration.

The approach we propose in the paper is an extension, albeit a considerable one, of the previously mentioned sequence-based image matching methods for non-structured estimation [7][8]. As a previous work, in [14] we have introduced the use of co-motion statistics for the alignment of the common groundplane of two overlapping views. In the paper we present a different, to that applied in [14], approach for outlier- rejection and show that relative camera parameters can be calculated from the estimated point correspondences. In our approach, instead of the trajectories of moving objects, we aim to use the statistics of concurrent motions – the so-called co-motion statistics – to locate matching points in pairs of images. The inputs of the system are video sequences derived from cameras located in fixed positions; however, the actual camera positions, orientations, and zoom settings are unknown. After matching of images the system aligns the multiple camera views into a single spatial frame; this makes it possible to track moving objects across different views, applying appropriate corrections for the 3D positions and orientations of the respective cameras. In our approach no *a priori* information is needed, and furthermore our method also works well for images containing random motion. Other methods fail in case of random motion, particularly with missing fixed background information.

II. MATCHING IMAGES

The main steps of our algorithm for image matching are as follows:

1. Detect motion: record the point coordinates where motion is detected (see *Section A*, below).
2. Update local and remote statistical maps (the concept of statistical maps is explained in *Section B*).
3. Extract candidate point-pairs from the statistical maps.
4. Reject points that are not relevant because they lie outside the common field-of-view (see *Section C*).
5. Fine-tune point correspondences by minimizing the reprojection error between the candidate point-pairs (see *Section D*).
6. Align the two images from the separate cameras.

The major initial assumption we make is that the image-sequences from the cameras are time-synchronized. Provided this is true, the motion information can be transformed into motion-statistics.

A. Motion detection

The application under consideration typically has several particular characteristics that are relevant to the motion-extraction task, and we aimed to reproduce these in our practical tests. Our videos of open-air road traffic scenes were created with wide-angle digital cameras of general-purpose type, so the images are blurred and noisy. The moving objects have a great variety of sizes; the images contain small moving blobs (walking people) as well as very large ones (e.g. buses). The static background cannot be extracted perfectly, because we do not want to assume any *a priori* knowledge about the scene.

In the first step, we define the pixels that are to be considered in the statistical calculations. The "motion blobs" are extracted by using simple running-average background subtraction with large β, deleting the irrelevant parts by using the preceding image I_{k-1} as a reference:

$$I_k(x,y) = \beta I_k(x,y) + (1-\beta) I_{k-1}(x,y), \quad 0 < \beta < 1$$

This method is fast and very sensitive with low threshold value; a disadvantage however is that in many cases image noise and background flashes are extracted. In the preprocessing algorithm the detected motion blobs are dilated until these reach the local maximums of the edge maps; we determine these local maximums by using a similar algorithm to that proposed by Canny [9]. This approach appears to be a usable solution for the detection of the significant moving objects in the scene. In our method we do not need precise motion detection and object extraction, because thanks to the later statistical processing such minor errors are irrelevant. The binarized image containing the detected objects is the motion map, which is used for updating the statistical maps. Typical results can be seen in Fig. 1.

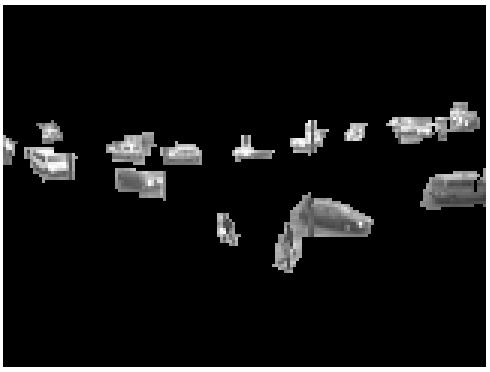


Fig. 1. The detected objects are visible in image (a); the corresponding motion map is shown in (b).

B. Co-motion statistics

In order to find point correspondences between two images in the case of wide-baseline stereoscopic video sequences we decided to analyze the dynamics of the scene. To do this, co-motion statistics (statistics of concurrent motions) were employed. In the case of a single video sequence, a motion statistics map for a given pixel can be generated as follows. When motion is detected at a pixel position, the coordinates are recorded of all other pixels where motion is also detected at that instant. In the motion statistics map, the motion values for the pixels at these recorded coordinates are updated. As a final step, this statistics map is normalized so that it has a global maximum equal to unity (see Fig. 2).

In the case of stereoscopic video sequences, the procedure is extended as follows. For each point in each image, two motion-statistics maps are generated: one local, and one remote. The "local" map means the motion-statistics map in respect of the image from which the given pixel is selected; the "remote" map refers to the motions in the other image. After motion is detected on the local side, for the points defined by the local motion map the local motion-statistics map is updated using the local motion map. Similarly, for each point where motion is detected on the local side, the corresponding remote motion-statistics map is updated using the motion map of the remote side.

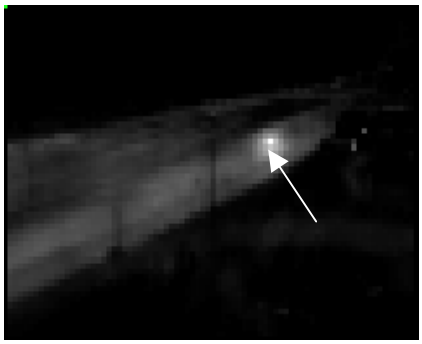


Fig. 2. Motion statistics map for the marked pixel.

The flowchart of the statistics-recording algorithm is given in Fig. 3; two examples for different cases are given in Fig. 5.

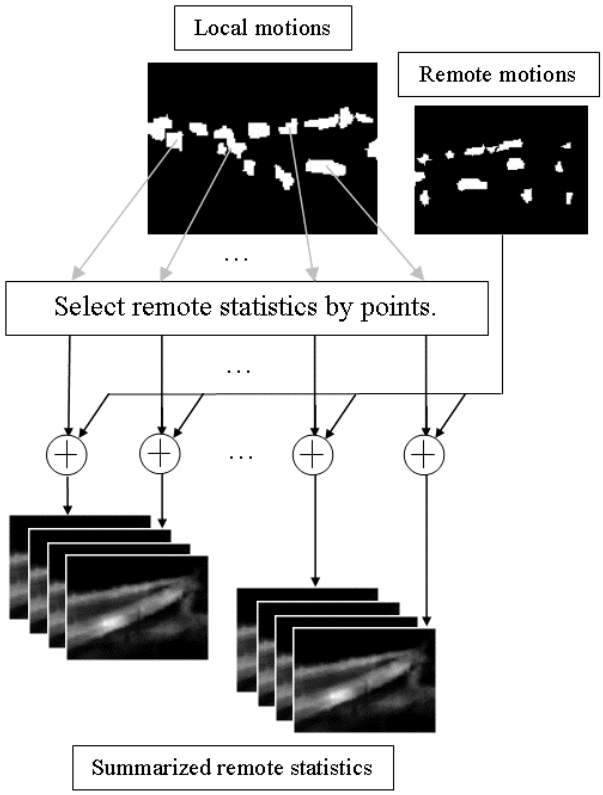


Fig. 3. Flowchart for recording of co-motion statistics.

C. Rejection of outliers

As candidate matches we consider global maximums on the local and remote statistical images. However, two problems may occur:

1. In the global statistical map (see Fig. 4), there may be image regions where many more moving objects are detected than in other regions. If we have a point in the first image located in a part of the scene that is not in the field of view of the second camera, then the corresponding maximum in the remote statistical map will be at an incorrect position, at a point where the value of the motion statistics in the global statistical map is high (see Fig. 6).
2. Because of the size of the moving objects the global maximums may be shifted, and may occur somewhere in the neighborhood of the desired corresponding point. This "shifting" can result in several distinct points from the local statistical image being mapped onto one and the same point in the remote statistical image.

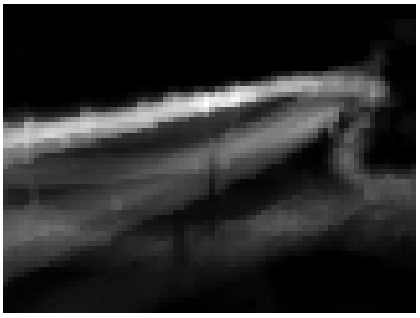


Fig. 4. Global statistics for one of the images are shown in the picture. Lighter areas represent regions where more motion was detected during the time period for which the statistics were calculated.

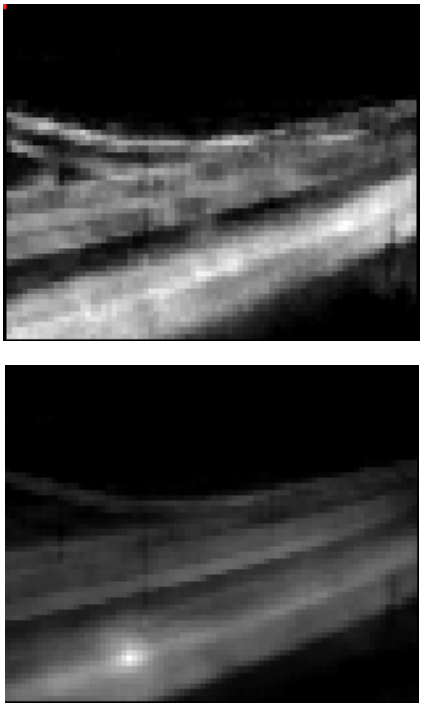


Fig. 5. Remote statistical maps for two cases are shown. (a) Map for a point not in the common field of view of the two cameras; (b) for a point that is in the common field of view.

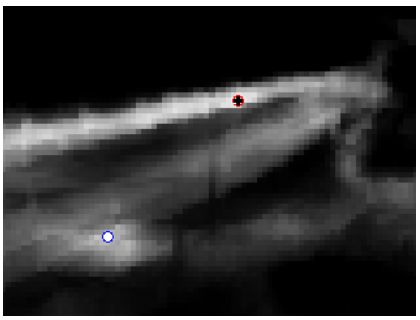


Fig. 6. Example of error caused by data from points outside the common field of view. The global maximum is marked by the black cross; while the correct position is shown by the white dot.

Point-pairs, in which both of points are from the overlapping views are assumed to be the inliers while any other point-pairs are the outliers.

Rejecting out the outliers from the set of candidate point-pairs is a classification of a set of points into two subsets, which is a binary classification problem. To perform it we have implemented a Bayes-decision algorithm. For this purpose we must define prior- and conditional probabilities and cost functions.

If we have a look on statistical maps we could see that in case of inlier point-pairs, see Fig. 5, the statistical maps can be described by a suitable three-dimensional normal distribution. But in case of outlier point-pairs the remote statistical map is very different from normal distribution, see Fig. 5. The difference between these two types of surfaces can be described by their standard deviations.

The definition of the prior probabilities is as follows. We have selected a number of statistical maps for inlier point-pairs and outlier point-pairs as well. Then we calculated the standard deviations for them and built up two clusters, one for inliers' standard deviations and one for outliers' standard deviations. The calculation of prior probabilities for inliers and outliers is as follows:

- Calculate the standard deviation for a given statistical map;
- Decide to which of the clusters the calculated standard deviation belongs to – by calculating the distance to the centroid of the clusters and selecting the smaller one;
- Normalize the calculated distance to get a probability.

Based on prior probabilities we can divide the whole set of point-pairs into inliers or outliers.

For the calculation of conditional probabilities the number of neighbors of a given point-pair is calculated. The only difference between calculation of conditional probabilities for inliers and outliers is that different mappings are applied to the same characteristics.

For the case of inliers the calculation of conditional probability is as follows:

- Calculate the number of neighbors within neighborhood of radius R. In case of an inlier point-pair we want to find approximately R^2 neighbors in the neighborhood. Finding a few or more than R^2 neighbors shows that it is also wrong. This have been done by calculating the conditional probability of being inlier with the function:

$$P_1^i(x) = \exp\left(-\frac{(x_n - N_0)^2}{2\sigma^2}\right),$$

where $\sigma=3$, $N_0=R^2$ in our examples; x_n is the number of neighbors in the R neighborhood of x. This mapping punishes as too much as a few neighbors within radius R and it rewards the case if the number of inliers is about R^2.

- For the estimation of the transformation between the images unique point pairs are needed. The number of pairs in the local statistical map of a point from a remote statistical map easily can be calculated. In this case the probability of being inlier is calculated by the mapping:

$$P_2^i(x) = \begin{cases} 1, x_u = 1 \\ 0, x_u \neq 1 \end{cases},$$

where x_u is the number of pairs in the remote statistical image of x. The overall conditional probability of being inlier is calculated as:

$$P((x_l, x_r) \mid inlier) = P_1^i(x_n) P_2^i(x_u),$$

where (x_l, x_r) is a point-pair from local and remote statistical maps; x_n is the number of inliers in the R neighborhood of x; x_u is the number of pairs in the remote statistical image of x_l.

For the case of outliers the calculation of the conditional probability of being outlier is very similar to the calculation in the case of inliers. The differences are only that, the number of outliers and being not unique are rewarded, the conditional probability is calculated by the formula:

$$P((x_l, x_r) \mid outlier) = P^o(x_n) P^o(x_u),$$

where (x_l, x_r) is a point-pair from local and remote statistical maps; x_n is the number of outliers in the R neighborhood of x; x_u is the number of pairs in the remote statistical image of x_l, P^o performs the following mapping:

$$P^o(x) = -\exp(-x) + 1$$

D. Fine-tuning of point correspondences

The above-described algorithm for rejection of points lying outside the overlapping parts of the images gives a set of point correspondences; but these results must be fine-tuned to enable correct alignment of the two views.

For the alignment of the two images a transformation is estimated from the extracted point correspondences. Consider a set of matches (m_1^i, m_2^i). It is required to find a two-dimensional projective transformation P that maps each m_1^i to m_2^i. The projective matrix is computed from the following linear equation:

$$KP = M_2$$

where

$$M_2 = (m^1_{11}, m^2_{11}, ..., m^n_{11}, m^1_{12}, m^2_{12}, ..., m^n_{12})$$

and

$$K = \begin{pmatrix} m^1_{21} & m^1_{22} & 1 & 0 & 0 & 0 & -m^1_{11}m^1_{21} & -m^1_{11}m^1_{22} \\ m^2_{21} & m^2_{22} & 1 & 0 & 0 & 0 & -m^2_{11}m^2_{21} & -m^2_{11}m^2_{22} \\ m^n_{21} & m^n_{22} & 1 & 0 & 0 & 0 & -m^n_{11}m^n_{21} & -m^n_{11}m^n_{22} \\ 0 & 0 & 0 & m^1_{21} & m^1_{22} & 1 & -m^1_{12}m^1_{21} & -m^1_{12}m^1_{22} \\ 0 & 0 & 0 & m^2_{21} & m^2_{22} & 1 & -m^2_{12}m^2_{21} & -m^2_{12}m^2_{22} \\ 0 & 0 & 0 & m^n_{21} & m^n_{22} & 1 & -m^n_{12}m^n_{21} & -m^n_{12}m^n_{22} \end{pmatrix}.$$

If $n \geq 4$, then we have an overdetermined set of equations, which can be solved using Singular Value Decomposition. The results of such a transformation are shown in Fig. 7. But from a glance at the Figure it is clear that the resulting transformation is not the desired one: edges which are continuous appear broken if we try to assemble a composite view from the transformed images. The point coordinates can contain errors; they may be shifted by several pixels, due to the nature of the co-motion statistics recording. If we have an error

Fig. 7. Image (a) is the view seen by the left-hand camera; (b) is the transformed view seen by the right-hand camera, without "fine-tuning". The matching is poor: edges that are continuous appear broken if a composite view is generated from the two images.

of even 1 pixel in the point coordinates, the fine alignment of the images cannot be performed. This simple point-rejection algorithm must be followed by a robust optimization to "fine-tune" point correspondences and achieve sub-pixel accuracy.

An iterative technique is used to refine both the point placements and the transformation. The method used is the Levenberg-Marquardt iteration [10] to minimize the sum-of-squares difference between the obtained coordinates and the transformed values. The entries of the transformation matrix as well as the coordinates of points in the image from the right-hand camera are treated as variable parameters in the optimization, while the point coordinates of the image from the left-hand camera are kept constant. The initial condition for this iteration consists of the entries of the transformation matrix and the point coordinates estimated by using the above-described rejection algorithm. The results of the optimization can be seen in Fig. 8 and Fig. 11.

Fig. 8. Point correspondences after "fine-tuning" of point locations. Circles denote correspondences before "fine-tuning"; while crosses denote correspondences after optimization.

III. Time-synchronization

Until now, we have assumed that the clocks of the two cameras are synchronized. For time-synchronization many algorithms have been developed, e.g. the Berkeley algorithm. In [14] we have applied a robust algorithm for the synchronization of videos. The procedure is as follows.

We have implemented a two-step procedure. Firstly, if the cameras were not synchronized then the generated co-motion statistics would no longer refer to concurrent motions detected in the two elements of the stereoscopic sequence. So, when we apply our algorithm for rejection of outliers, we will not get a "large" set of point correspondences; more point correspondences can be extracted in the case of synchronized sequences. This obvious prediction is indeed found to be true in practice. Therefore to synchronize the sequences we can calculate point correspondences for different time-offset values, and then perform a one-dimensional search for the value giving the largest set of point correspondences. Unfortunately, even in the case of unsynchronized sequences the algorithm produces point correspondences.

Secondly, if we analyze the sum-of-square differences score (the reprojection error in this case), we find that the global minimum is at the same offset value as the global maximum for the cardinalities of sets of point correspondences.

IV. Estimation of the epipolar geometry

The geometry of multiple views is well-understood [11]. For two views there exist geometric constraints that relate corresponding points to the 3D camera geometry.

In our experiments we have implemented the normalized 8 point algorithm, the extended version of the Longuet-Higgins' 8 point algorithm, for the estimation of the epipolar geometry of two views.

Point set reconstruction

The points are reconstructed in the object space that projects on to u_i and u'_i in the two images, under the respective transforms P_1 and P_2. An iterative technique is used to minimize the sum-of-square differences between the image coordinates u_i and u'_i and the predicted values $u_i = P_1 x_i$ and $u'_i = P_2 x_i$. The method used is the Levenberg-Marquardt iteration [13/10]. The entries of the matrix P_2 as well as the objects coordinates x_i are treated as variable parameters in the optimization, but the matrix P_1 is held equal to (I|0). Convergence of this algorithm is rapid and reliable, given initial estimates for the camera matrices derived from the factorization.

V. Results

The above-described approach was tested on videos captured by two cameras, having partially overlapping views, at Gellert (GELLERT videos) and Ferenciek squares (FERENCIEK videos) in Budapest. The GELLERT videos are captured at resolution 160×120, at same zoom level and with same cameras while the FERENCIEK videos are captured at resolution 320×240, at different zoom levels and with different cameras.

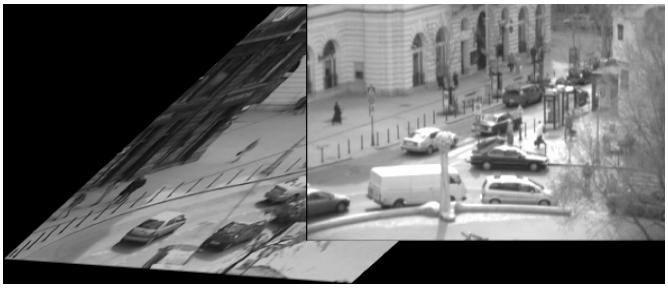
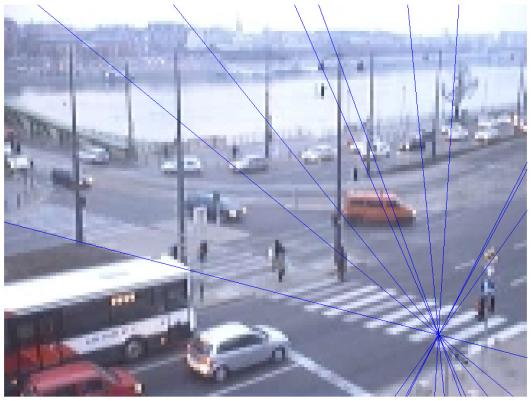


Fig. 11. The constructed composite views are in the pictures. The upper one is generated for the GELLERT videos, the lower one is for the FERENCIEK videos.Based on the extracted point-correspondences the epipolar geometry easily can be estimated by using standard algorithms [11]. The self-calibration of cameras can be done.

The common field of view of the two cameras in both cases is about 30%. The proposed outlier rejection algorithm rejects most (98%) of the candidate point pairs. For the GELLERT videos it results in 49 point-correspondences and in 23 for FERENCIEK videos, which are still enough to estimate common groundplanes.

The computation time of the whole statistical procedure was about 10 minutes for 10 minutes of video presented in the figures. For longer sequences and higher resolution we apply a two-step procedure: the generated statistical maps are of resolution 80×60, then, based on them, the fine-tuning of point-correspondences was done at the video's native resolution.

Having the corresponding points computed the epipolar geometry is estimated. The epipolar pencils for both of image pairs can be seen in Fig. 12 and Fig. 13. It can be seen from Fig. 12 and Fig 13. that the epipolar geometry is estimated correctly. However, only the "relative" camera parameters can be estimated from two views [11].



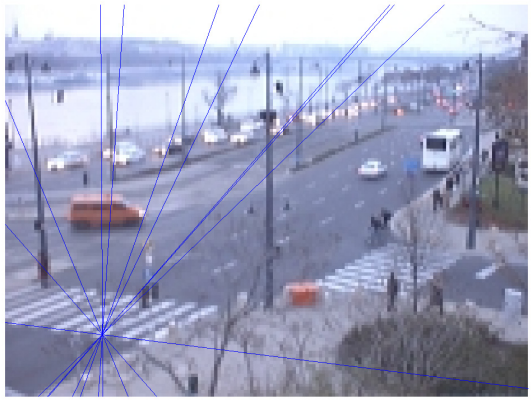
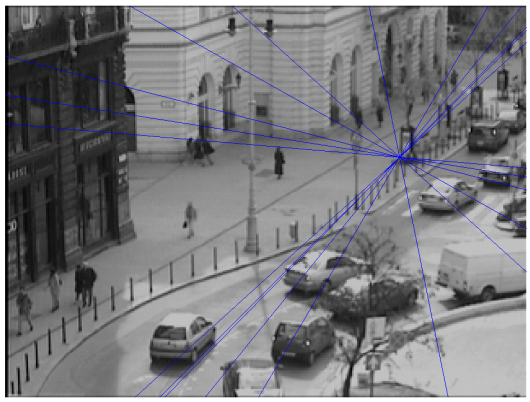


Fig.12. The epipolar pencils for the GELLERT test videos.



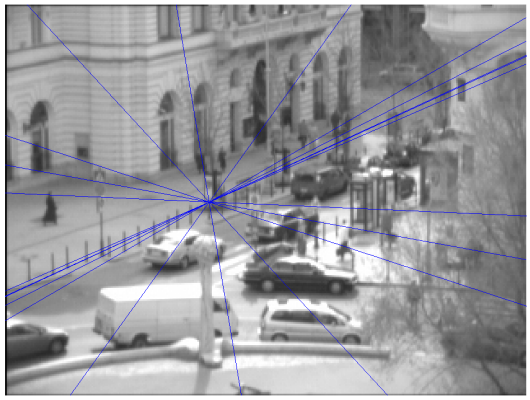


Fig. 13. The epipolar pencils for the FERENCIEK test videos.

VI. Conclusions

The paper has shown that outdoor cameras placed in freely-chosen positions, viewing arbitrary scenes where motion is present, can be calibrated automatically without human interaction. In our approach no *a priori* information on camera positions or characteristics is needed, and the method also works well for images containing randomly moving objects;

in this situation other methods fail, especially if fixed structures are lacking.

In our approach we introduce co-motion statistics to find matching points in image pairs. We first derive motion statistics, and then choose global maximums as candidate matches. This step is followed by the elimination of points lying outside the overlap area from the set of candidate matches, and an optimization process based on the minimization of the reprojection error between images, to "fine-tune" the locations of candidate point-pairs. In practical tests good image-matching is achievable, with acceptable processing-time.

Our next experimental set-up will consist of at least three cameras so the absolute camera parameters would be estimated [7][11]. The estimation of the epipolar geometry, correctly, shows the feasibility of our approach.

REFERENCES

[1] O. D. Faugeras, Q.-T. Luong, S. J. Maybank, "Camera self-calibration: Theory and experiments," in *Proc. ECCV '92, Lecture Notes in Computer Science*, vol. 588, Berlin Heidelberg New York, Springer-Verlag, 1992, pp. 321-334.

[2] D. H. Ballard, C. M. Brown: *Computer Vision*, Prentice-Hall, Englewood Cliffs NJ, pp. **, 1982.

[3] 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.

[4] S. T. Barnard, W. B. Thompson, "Disparity analysis of images," *IEEE Trans. PAMI*, vol. 2, pp. 333-340, 1980.

[5] J. K. Cheng, T. S. Huang, "Image registration by matching relational structures," *Pattern Recog.*, vol. 17, pp. 149-159, 1984.

[6] J. Weng, N. Ahuja, T. S. Huang, "Matching two perspective views," *IEEE Trans. PAMI*, vol. 14, pp. 806-825, 1992.

[7] L. Lee, R. Romano, G. Stein, "Monitoring activities from multiple video streams: establishing a common coordinate frame," *IEEE Trans. PAMI*, vol. 22, 2000.

[8] Y. Caspi, D. Simakov, and M. Irani, "Feature-based sequence-to-sequence matching," in *Proc. VAMODS (Vision and Modelling of Dynamic Scenes) workshop, with ECCV'02*, Copenhagen, 2002.

[9] J. Canny, "A computational approach to edge detection," *IEEE Trans. on Pattern An. and Mach. Intell.*, vol. 8, pp. 679-698, 1986.

[10] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*. Cambridge, Cambridge University Press, 1986.

[11] R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, Cambridge University Press, 2003.

[12] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, 1981.

[13] R. Hartley, "Estimation of relative camera positions for uncalibrated cameras," in *Proc. of ECCV'92, Lecture Notes in Computer Science*, vol. 588, Berlin Heidelberg New York, Springer-Verlag, 1992.

[14] Zoltán Szlávik, László Havasi, Tamás Szirányi, "Estimation of common groundplane based on co-motion statistics", in *Proc. of ICIAR'04*, *Lecture Notes in Computer Science,* 2004.