Optimizing of Searching Co-Motion Point-Pairs for Statistical Camera Calibration

Zoltán Szlávik, Tamás Szirányi
Analogic and Neural Computing Laboratory
Hungarian Academy of Sciences
H-1111 Budapest, Kende u. 13-17, Hungary

László Havasi, Csaba Benedek
Péter Pázmány Catholic University
H-1052 Budapest Práter u. 50/a., Hungary

Abstract— **In the paper we introduce an algorithm 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 our previous work [10] we have shown that by using co-motion statistics matching of overlapping views can be done and then the projective geometry can be estimated. Here we will show how to optimize searching for concurrently moving pixels. The robust algorithm we describe here finds point correspondences in two images by using entropy-based thresholding and without searching for any structures and without the need for tracking continuous motion. Our method makes it possible to (re)calibrate multicamera systems without human assistance.**

Keywords-camera calibration, image registration, co-motions

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 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. Usually an algorithm for the alignment of different views and calibration of cameras has the following steps:

1. Feature detection;
2. Extraction of candidate point-pairs;
3. Rejection of outliers and estimation of the model that does the alignment;
4. Alignment of different views;
5. Estimation of epipolar geometry.

In the paper we will present an algorithm for the first four steps of the above general schema. Having the point correspondences extracted the estimation of epipolar geometry can be done by well-known algorithms [1][8][10].

Matching different images of a single scene may be difficult, because of occlusion, aspect changes and lighting changes that occur in different views. Still-image matching algorithms [2][3][4][5] search for still features in images such as: edges, corners, contours, color, shape etc. They are usable for image pairs with small differences; however they may fail at occlusion boundaries and within featureless regions. They may fail if the chosen primitives or features 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 also contain information about the scene dynamics besides the static frame data. Scene dynamics is an inherent property of the scene independently of the camera positions, the different zoom-lens settings and lighting conditions. References [6] and [7] present approaches in which motion-tracks of the observed objects are aligned. However, in these cases a robust capability for object tracking is assumed; and this is the weak point of both methods.

As a previous work the use of co-motion statistics for the estimation of projective geometry was introduced in [9][10]. The approach proposed in [10] is an extension, albeit a considerable one, of the previously mentioned sequence-based image matching methods for non-structured estimation [6][7]. In [9][10] we have introduced the use of co-motion statistics for the matching and alignment of two overlapping views and estimation of the common groundplane. In that approach, instead of the trajectories of moving objects, the statistics of concurrent motions – the so-called co-motion statistics – were used 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. The main advantage of the use of co-motion statistics that no *a priori* information about motion, objects or structures is needed. The disadvantage of co-motion statistics is that the system needs huge memory for storing it.

The purpose of our paper is to present an algorithm for the efficient estimation of co-motion point-pairs and a robust feature extraction method. In the presented algorithm less memory is needed for coding scene dynamics, the calculations

0-7803-9134-9/05/$20.00 ©2005 IEEE

have been done on-line. We also have tested new entropy-based methods for the definition of changes of importance to extract features.

II. MATCHING OF IMAGES

The algorithm described here is based on the use of co-motion statistics for matching images [10]. The steps of the algorithm are the following:

1. Detect changes.
2. Store changes that the dynamics of the scene can be reconstructed later.
3. Extract point-correspondences from the stored scene dynamics – detection of features, extraction of candidates, rejection of outliers.

Do the alignment of the cameras' views.

A. Co-motion statistics

Scene dynamics is encoded in co-motion statistics, so if static features (corners, edges etc.) cannot be reliably detected the information for matching can be extracted from co-motion statistics [9][10].

In case of single video sequence a motion statistical map for a given pixel can be recorded as follows: when motion is detected in a pixel, the coordinates are recorded of all pixels where motion is also detected at that moment. In the motion statistical map the values of the pixels at the recorded coordinates are updated. After all, this statistical map is normalized to have global maximum equal to 1.

In case of stereo video sequences to each point in the images, two motion-statistic maps are assigned: a local and a remote. Local map means the motion-statistical map in the image from the pixel is selected, the remote motion-statistical map is refer to the motions in the other image. After the motion detected on the local side, for the points defined by the local motion map the local statistical map updated by the local motion map. For each point where motion is detected on the local side, the local motion map of the remote side updates the corresponding remote statistical map. An example of co-motion statistics for inlier point-pairs can be seen in Figure 1.

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

The advantage of this interpretation of the scene dynamics is that point correspondences in the above case were interpreted as maximums of statistical maps and their extraction is very simple. The main disadvantage of co-motion statistics is that the system must keep two statistical maps (grayscale pictures) for each pixel of input image, which means that the algorithm needs huge memory, in case of 160*120 statistical map resolution it means 1,4 GBs!

B. Coding scene dynamics

Having the result of change detection the scene dynamics can be coded and stored. To overcome the problem that huge memory is needed for storing co-motion statistics we propose to store the motion history in a vector for each pixel instead of storing an image-size map for each of them as in [9][10]. This motion history vector has as many entries as long is the video sequence and in each of its entries is 1 if change was detected at the given frame or zero if not. This coding reduces the memory needs while the scene dynamics is also coded in the vectors. The disadvantage is that for the extraction of point correspondences all the motion histories of cameras must be compared. Because of information loss after thresholding in change detection we compared it in "binary" series to "real-valued" series in which the value of two frames' absolute difference is stored instead of 1 or 0 for pixel *(x,y)*. The advantage of this method is that less memory is needed for the storage of scene dynamics than in the case of co-motion statistics: 2.3 MB for using binary series and 18 MB for using "real-valued" series in case of 500 frame long image sequence, while for the storing of co-motion statistics 1,4 GB is needed.

C Extraction of point correspondences

Usually the estimation of point correspondences in two given images consists of three steps. Firstly, features are detected then candidates of point pairs are extracted and, finally the outliers are rejected and the given model is estimated.

C.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 have compared two methods. In the first method we are integrating the motion histories. If this value is above some threshold then the corresponding pixel is selected as a feature point. This method is very sensitive to the threshold value.

In the second we have calculated the Shannon entropy of pixels' motion history vectors.

$$entropy = -\sum p(x_i)\log p(x_i) \qquad (1)$$

where $p(x_i)$ is the frequency of x_i in vector v, v – real-valued motion history vector. Our experiments with different indoor and outdoor videos showed that the entropy of motion history vectors of flashings and other random noise will very "high" – about 0.4-0.5, while the entropy of motion history of deterministic motion of real object will lower – it will be a value between 0 and 0.2.

Instead of traditional definition of entropy for vector v, we have also tested the formula for the estimation of the "entropy":

$$entropy^* = -\frac{1}{\log(N)}\sum v_i \log v_i \qquad (2)$$

where v_i are the elements of the history vector v, N – the length of history vector v. Applying this formula is not a serious restriction to our algorithm. The meaning of its output is similar to that of the output of the traditional formula. If in a given pixel the system is observing only small flashings then the value of $entropy^*$ will be high (logarithm of a small number is large number in absolute value). If object is moving through the pixel then the detected change will be much higher then in case of flashings and the value of $entropy^*$ will be low (logarithm of a large number is small). The thresholds that were used for selecting candidate points were shifted, the new interval is [0.1,0.32] for the extraction of feature points. The main advantage of formula (2) against (1) that it can be calculated on-line, from frame to frame as the implemented change detection algorithm.

It is obvious that if all 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 maximum and its entropy is within a given interval.

C.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. For the comparison of feature points, the corresponding motion history vectors in our case, we have implemented different methods for binary and real-valued motion history vectors.

In the case of real-valued motion history vectors the extraction of candidate point pairs is based on the calculation of the correlation between a given feature point and feature points of the other view.

In the case of binary motion history 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.

C.3 Robust estimation of the model and rejection of outlier

For the estimation of transformation P 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 [8][10]. In our experiments people and cars are moving on the groundplane. In this case P is a projective transformation that can be represented by a 3*3 matrix and can be calculated from at least 4 point-pairs.

III. Experimental Results

The above-described algorithms were tested and compared on videos captured by two cameras, having partially overlapping views, at Gellert (GELLERT videos) square in Budapest. The GELLERT videos are captured at resolution 160×120, at same zoom level and with same cameras.

An example of the extracted inlier points can be seen in Figure 2.

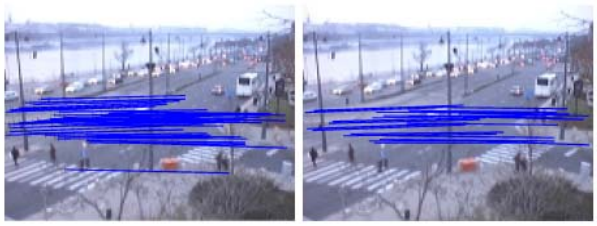

Figure 2 Inliers for the real-valued motion history vectors and on-line entropy based feature extraction (left image); and for the binary motion history vectors and on-line entropy based feature extraction (right image).

The results of alignment are compared for five transformations: T1 - the point correspondences are extracted by using binary motion history and thresholding of integrated motion history for feature extraction; T2 - the point correspondences are extracted by using binary motion history and on-line entropy for feature extraction; T3 - the point correspondences are extracted by using real-valued motion history and thresholding of integrated motion history for feature extraction; T4 - the point correspondences are extracted by using real-valued motion history and Shannon entropy for feature extraction; T5 - the point correspondences are extracted by using real-valued motion history and on-line entropy for feature extraction.

For the exact comparison of the obtained results we have estimated a reference image alignment transformation P_1 based on manual feature point selection in the input images. 100 reference points were created with reference transformation P_1 in both images. Then the symmetric transfer errors (STE) [8] of transformations T1-T5 on the set of reference points were calculated. The results of comparison are shown in Table1.

Table 1 The average of symmetric transfer errors (STE) for the obtained transformations (see the text).

Transformation	Average STE	Min STE	Max STE
T1	13,40	0,73	39,40
T2	6,54	0,44	17,27
T3	11,18	0,88	36,23
T4	9,00	0,63	52,57
T5	7,15	0,16	28,22

As it can be seen from the results of Table 1, algorithms T2, T4, T5 performed well (T2 is the best), in which the feature selection is entropy-based,. This is because in the entropy-based feature selection the entropy of motion history series measures the “quality” and not only the quantity of motion through the pixel. Pixels with low value of integrated motion history also can be extracted. The result of final alignment is shown in Figure 3.

Figure 4 shows the results of final alignment with transformation T2 for the FERENCIEK videos. FERENCIEK

videos are captured at Ferenciek square in Budapest at resolution 320×240, at different zoom levels and with different cameras.

IV. Conclusions

We have shown that partially overlapping camera views can be registered by motion history vectors of images' reference pixels of outdoor cameras placed in freely-chosen positions, viewing arbitrary scenes where motion is present, and this matching is automatic without any human interaction. The main advantage of the presented algorithm is that it does not need *a priori* information about objects or structures. We have shown that the entropy of motion history vectors unequivocally defines the threshold for the selection of feature points. To speed up the calculations we used motion-history in the comparison and this history is captured by defining an on-line entropy while the final alignment remains quite acceptable.

Acknowledgment

The authors would like to acknowledge the support received from the NoE MUSCLE project of the EU.

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, pp. 321-334, 1992.

[2] 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.

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

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

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

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

[7] 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.

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

[9] 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,* pp. 347-353, 2004.

[10] Z. Szlávik, L. Havasi, T. Szirányi, "Image matching based on co-motion statistics", Proc. of 2nd Int. Symposium on 3DPVT, Thessaloniki, 2004.

Figure 3 Final alignment of two views with transformation T2 for the GELLERT videos.

Figure 4 Final alignment of two views with transformation T2 for the FERENCIEK videos.