Robust Extrinsic Camera Calibration from Trajectories in Human-Populated Environments

Guillermo Baqueiro and Jean-Bernard Hayet

Computer Science Group,
Centro de Investigación en Matemáticas (CIMAT, A.C.),
Guanajuato, Gto. 36240 México.
baqueiro,jbhayet@cimat.mx *

Abstract. This paper proposes a novel robust approach to perform inter-camera and ground-camera calibration in the context of visual monitoring of human-populated areas. By supposing that the monitored agents evolve on a single plane and that the cameras intrinsic parameters are known, we use the image trajectories of moving objects as tracked by standard trackers in a RANSAC paradigm to estimate the extrinsic parameters of the different cameras. We illustrate the performance of our algorithm on several challenging experimental setups.

1 Introduction

In spite of its spectacular development, video-surveillance is yet largely depending on human agents in charge of monitoring up to dozens of TV screens, which may be responsible for negative detections. Recent years have seen the emergence of automatic, computer-aided video-surveillance systems in the computer vision community. Typically these systems use state-of-the-art tracking algorithms in each camera in the network, and fusion techniques to recover 3D trajectories of the moving objects in the scene [1]. Then, this information feeds pre-defined or unusual event detection and may trigger alarms. An important element for a widespread use of such systems is an *automatic* calibration algorithm that would not require the costly intervention of an expert.

This article presents such an algorithm, that estimates the extrinsic parameters of different cameras involved in a surveillance network, i.e. the different 3D transformations between pairs of cameras and between each camera and the reference plane. The assumptions we make are that (1) the targets are moving on a planar scene, which is a common setup in surveillance systems, (2) we have an estimation of the intrinsic parameters of the cameras, and (3) the cameras are static. An important characteristic of such camera networks is that the viewpoints may be dramatically different from one camera to another, e.g., in the frames from two sequences of cameras at Fig. 1. In particular, this prevents us from using traditional feature-based matching techniques based on local descriptors around interest points [2] for estimating the underlying geometric transforms. Instead, in the vein of the seminal work of [3], we rely on the output of motion detection and motion tracking to guess correspondences at the level of motion blobs or motion tracks and to infer the corresponding geometry.

* This work was partially funded by CONCyTEG through its grant 09-02-K662-073.

Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 3-10

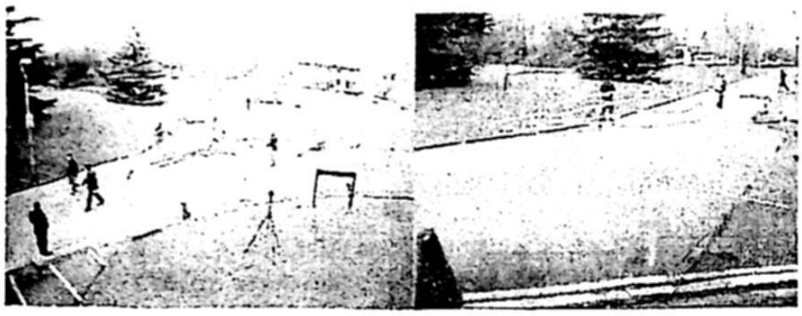

Fig. 1. The typical input/output of our algorithm: we form correspondences (one color, one correspondence) among trajectories (from standard trackers) to compute robustly the feet-to-feet homography H_{ij} between the two views. These inlier *trajlets* (see Section 3) correspond to the computations of homographies between camera 2 and 7 of the PETS 2009 data (i.e., subfigures of the right column of Fig.3).

The organization of this paper is as follows: In Section 2, we highlight noticeable related work in the literature; in Section 3, we describe our algorithm for robust inter-camera homography estimation; in Section 4, we recover all the extrinsic parameters from homographies and in Section 5 we comment results on different setups; finally, Section 6 draws conclusions and introduces future work.

2 Related work and contributions

The seminal work of Lee and al. [3] uses the centroids of blobs extracted with standard background subtraction techniques to perform homography fitting with a least median square (LMS) approach, that is further refined in a second step. Its main drawback is that the number of putative correspondences grows very fast with the number of targets, so that the number of inliers for the LMS optimization drops dramatically in proportion, making the algorithm unsuitable for regularly crowded scenes. Obviously the dimension of the search space is reduced drastically when, instead of *motion detection blobs*, one form the correspondences from *tracking sequences* [4, 5]. In [5], the authors present a RANSAC-like approach that performs non-uniform sampling in the set of putative sequences. It sequentially tests homographies from two pairs of sequences (two pairs in each video) and keeps the best homography but the likelihood functions that ponder each sample are not clearly defined. The work in [4] is more general in a sense, as it is extended to fundamental matrix estimation, and is also based on RANSAC, but does not make particular distinction between samples to guide the consensus to the most promising pairs of sequences. In another paradigm, the work of [6] uses perspective invariants, namely the cross ratio of five points, to select corresponding trajectories between video sequences. The algorithm also allows to calibrate the time offset between cameras. However, in most situations, it is quite difficult to isolate non-degenerate trajectories – i.e, sufficiently far from straight lines – to compute stable cross-ratios, so that the possible applications of this work are limited. A common inconvenient of the previous approaches is that it uses tracking trajectories as they come from the tracking algorithm, which causes problems of robustness in the case that the tracking fails – and that the system is not aware of it. Among the most recent works in the area, the one

of [7] is interesting as it also takes radial distortion into account. However, the correspondences are determined on the base of control points manually selected onto trajectories, which may make it more adapted for expert users.

In many situations, e.g. because of occlusions in crowded scenes, tracking algorithms may be confounded and may assign a wrong identity to some tracked object. This may be catastrophic for the estimation of scene geometry. Our approach brings several contributions among which (1) robustness with respect to the possible failures of the tracking algorithms and (2) more reliable guidance of the optimization process to the correct geometry.

3 Robust inter-video homography estimation

3.1 Problem formulation

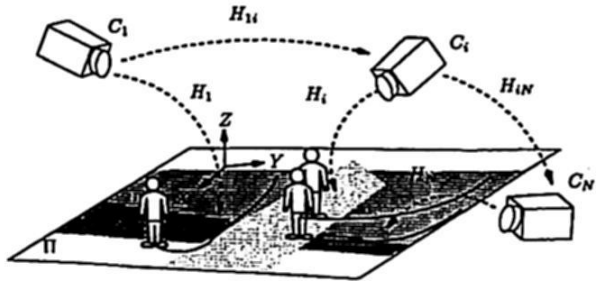

Fig. 2. Setup: Several cameras C_i for $1 \leq i \leq N$ observe a *planar* scene laid on a plane Π. The inter-camera homographies between two cameras i and j are denoted by H_{ij}, while the camera-to-reference plane homographies from camera i to Π are denoted by H_i. Each camera has it own (shaded) scene coverage.

The problem setup and notations are detailed in Fig. 2: several cameras C_i, $1 \leq i \leq N$, with different degrees of overlap, monitor a scene where people or other mobile objects move. We suppose that this scene is laid on a reference plane Π, which induces an *homography* between any pair of cameras (i, j) monitoring the scene, i.e. if $p_i = (u_i, v_i)^T$ is an image point in camera C_i, projection of a point P of the plane Π, and $p_j = (u_j, v_j)^T$ the projection of this same point on camera C_j, then we have the classical relationship [8],

$$p_i = \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} \sim H_{ij} p_j = \begin{pmatrix} h_{ij}^{11} & h_{ij}^{12} & h_{ij}^{13} \\ h_{ij}^{21} & h_{ij}^{22} & h_{ij}^{23} \\ h_{ij}^{31} & h_{ij}^{32} & h_{ij}^{33} \end{pmatrix} \begin{pmatrix} u_j \\ v_j \\ 1 \end{pmatrix}, \tag{1}$$

where $\sim$ means that a relation of equality holds for any multiplication factor $\lambda > 0$, so that H_{ij} has in fact only 8 degrees of freedom. The problem consists in estimating these transforms, first, and, on a second step, the homographies H_i that maps points p_i to the points they are the projection of, i.e. points P,

$$p_i = \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} \sim H_i P = \begin{pmatrix} h_i^{11} & h_i^{12} & h_i^{13} \\ h_i^{21} & h_i^{22} & h_i^{23} \\ h_i^{31} & h_i^{32} & h_i^{33} \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}, \tag{2}$$

where $P = (X, Y, 0)^T$ are the coordinates of point P in a frame (X, Y, Z) (depicted in Fig. 2) such that $Z = 0$ is the equation of Π. Traditional methods estimate homographies H_{ij} by searching for point correspondences (p_i, p_j) and using them to solve the linear system directly induced by Eq. 1. As these correspondences are extremely difficult to find with points and appearance whenever the viewpoint changes strongly, we rely on tracks from video trackers.

Our algorithm can be summarized in: (1) **collect trajectories** in each stream V_i with a tracking algorithm, (2) **pre-process** the trajectories to eliminate potential ambiguities at occlusion points, we will refer to the trajectory parts built in this way as *trajlets*, and (3) apply the RANSAC-like robust **optimization process** with a likekihood-guided sampling process.

3.2 Collecting and pre-processing trajectories

As we want the algorithm to be *robust* w.r.t. the properties of the 2D tracker, we apply any state-of-the art 2D tracker on our video streams and collect the resulting tracks. Practically, we used different algorithms implemented in the OpenCV library. The result, for a video stream i of camera C_i, is a set of trajectories $L_i = \{l_i^{(m)}, m \geq 0\}$, encoding the position of one target **centroid** along the time. The centroids are chosen because they are not as sensitive to noise as the feet position. However, a consequence is that the computed homography will correspond to a *plane passing through target centroids* (i.e., not Π). We denote it as $\hat{H}_{ij}$ and will see that H_{ij} can be deduced from $\hat{H}_{ij}$.

The second step forms what we call *trajlets*, i.e., pieces of trajectories that are a priori not susceptible to be contaminated by occlusion errors from the tracking algorithm. For all pairs of trajectories $(l_i^{(m)}, l_j^{(m)})$ for which some of the points $p_{i,t}, p_{j,t}$ are close (at some timestamp t), we simply cut off the ambiguous parts on a given time radius δ. This creates four sub-trajectories (*trajlets*) $(l_i^{(m)+}, l_t^{(m)-}, l_j^{(n)+}, l_j^{(n)-})$, such that,

$$\begin{cases} l_i^{(m)} = l_i^{(m)-} \cup \{p_{i,t-\delta} .. p_{i,t+\delta}\} \cup l_i^{(m)+}, \\ l_j^{(n)} = l_j^{(n)-} \cup \{p_{j,t-\delta} .. p_{j,t+\delta}\} \cup l_j^{(n)+}. \end{cases}$$

Also, we smooth these trajectories by using local filtering based on Bezier curves. The result of this processing is, again, for each video stream i, a – a priori larger – set $L_i' = \{l_i^{(m)}, m \geq 0\}$, as illustrated by Fig. 1.

3.3 Robust homography estimation

The estimation of $\hat{H}_{ij}$ is done in a RANSAC-like scheme described in this section. A priori, a candidate homography for explaining the two images from the same scene can be derived from just one correspondence between a trajectory in i and a trajectory in j, since it is entirely defined by 4 point correspondences [4, 6]. However, most of the *trajlets* appearing in usual video-surveillance contexts are close to degenerate, i.e. linear. This is why we generate here the candidate homographies from *two trajlets* correspondences instead of one. This, in turn, has an inconvenient, since, if we have an order of magnitude of τ *trajlets* appearing at intersecting windows of time, then the probability for a sampled pair (i, j) to

match is $\frac{1}{\tau}$, and the one for two consecutively sampled trajlets $\frac{1}{\tau^2}$. Hence, the number of samples needed in RANSAC to ensure (in expectation) that a correct pair is sampled is **quadratic** in τ, which can be problematic with crowded scenes.

A solution is to avoid an uniform sampling process by assigning likelihood values to all possible pair of trajectories, and by sampling the trajectories pairs according to these values of likelihoods. We define them in the following way:

$$p(l_i^{(m)}, l_j^{(n)}) \propto \frac{1}{\max(N_j(l_i^{(m)}), N_i(l_j^{(n)}))} \Delta(l_i^{(m)}, l_j^{(n)}),$$

where $N_j(l_i^{(m)})$ stands for the number of *trajlets* from L'_j that have a time overlap with trajectory $l_i^{(m)}$ (the value being defined as one if there is no time overlap) and $\Delta(l_i^{(m)}, l_j^{(n)})$ measure the time overlap. These two terms (1) penalize the sampling of trajectories that could result **ambiguous to match** (large N_j or N_i) and (2) favor those trajectories with large overlap, which should improve the homography computation by preferring larger sequences. Both criteria make the required number of samples much lower than the aforementioned quadratic term above. In practice, we need a few dozens iterations to get a pair of correctly matched *trajlets*. The whole optimization process is described in Algorithm 1.

Algorithm 1 Homography computation between cameras C_i and C_j

$\hat{S} \leftarrow 0$
repeat
1. Sample a pair of *trajlets* $(l_i^{(m)}, l_j^{(n)})$ according to the likelihoods $p(l_i^{(m)}, l_j^{(n)})$.
2. Compute the candidate homography $\hat{H}_{ij}^{mn}$ from the correspondences between all points of $l_i^{(m)}$ and $l_j^{(n)}$), by using the classical DLT [8], and its inverse $(\hat{H}_{ij}^{mn})^{-1}$.
3. For all trajlets pairs $(l_i^{(r)}, l_j^{(s)})$ compute the residual symmetric error $\epsilon^2(r,s)$:

$$\epsilon^2(r,s) = \frac{1}{2|l_i^{(r)} \times l_j^{(p)}|} \sum_{(p_r,p_s)\in l_i^{(r)}\times l_j^{(p)}} d^2(\hat{H}_{ij}^{mn} p_r, p_s) + d^2(p_r, (\hat{H}_{ij}^{mn})^{-1} p_s).$$

4. Identify in the residual matrix $\epsilon^2(r,s)$ elements that are (1) below a given threshold and (2) minima on the line r and column s.
5. Sum in S the lengths of the trajectories corresponding to the identified elements.
6. If $S > \hat{S}$, $\hat{S} \leftarrow S$; $\hat{H}_{ij} \leftarrow \hat{H}_{ij}^{mn}$.

until a given proportion of trajectories from video streams V_i and V_j have been explained by $\hat{H}_{ij}$ or a given number of iterations have been done.
if $\hat{H}_{ij}$ explains a sufficient proportion of trajectories in V_i and V_j **then**
  consider $\hat{H}_{ij}$ as recovered
else
  consider the two views as un-registered.
end if

4 Extrinsic parameters estimation

Homography decomposition. Once the homographies $\hat{H}_{ij}$ have been recovered, we estimate extrinsic parameters, i.e. the parameters of the rigid 3D transform between the two cameras acquiring video streams i and j. To this purpose, we use the following decomposition of H_{ij} [8],

$$\hat{H}_{ij} \sim K_i[dR + tn^T]K_j^{-1} \tag{3}$$

where the matrices K_i are the intrinsic parameters of cameras i, supposed known here, and where d, n give the equation of the plane (here, the centroids plane) in camera i frame, i.e. its equation is $n^T Q = d$, where Q are the coordinates of 3D points in the camera i frame. Note that Eq. 3 is given only *up to a scale factor*, that we will determine in a second time. We use Triggs' algorithm to determine the decomposition values [9]. It gives two possible pairs (R, t), one of them being easily discardable.

Image plane to ground plane homography. From the image-to-image homography and its decomposition, one recovers[1] an homography to the centroids plane, by deriving from the projection equation on camera C_i,

$$(u_i, v_i, 1)^T \sim K_i(dn + Q_{n^\perp}),$$

where $Q_{n^\perp}$ is the component of Q onto the centroids plane. By choosing a base e_1, e_2 of vectors generating the centroids plane, for example $e_1 = (1, 0, 0)^T \wedge n$, and $e_2 = e_1 \wedge n$, one derives in terms of the spatial coordinates on the – real – centroids plane, (α, β),

$$(u_i, v_i, 1)^T \sim K_i \left(e_1, e_2, dn\right) (\alpha, \beta, 1)^T,$$

i.e. $\hat{H}_i = K_i \left(e_1, e_2, dn\right)$ acts as an homography from the centroids plane to the image plane in camera C_i.

Scale recovery. As t and d are computed only up to a scale, we used some knowledge about the scene to compute the scale factor, one option is to assume a constant, fixed velocity for the object in the scene with median velocity, another one to assume a half-height of one meter between people's centroids and feet. In our experiments, the second option gave better and much stabler results[2]. Once the scale is recovered, one finally gets either a ground plane-to-image H_i (as in Eq. 2) or an image-to-image H_{ij} induced by the ground plane Π (as in Eq. 1). Most of the results below illustrate this second form. Note that for the moment all the homographies are expressed in a different frame relative to Π.

5 Results

As mentioned before, this work has been implemented in C++ with the OpenCV library. We tested our algorithm on the PETS [10] benchmark data, which provides different case studies, with eight video streams for each case, corresponding

[1] Up to a scale factor for d !

[2] It seems that one problem in using velocity information is that the framerates of most videos of the benchmark data are not constant.

to different level of people density. For calibration purposes, we used the medium density crowd dataset (S0), and give some results in Fig. 3. We depict three of the computed homographies H_{ij} (left) and their inverse (right) by warping the image j onto the image plane i. It can be observed that most of the spottable elements in Π (roads, lines) that appear in both views are mostly correctly warped in the other view. The Fig. 1 gives the inliers *trajlets* that served to the estimation of the homography of the last column (i.e. between views 2 and 7). In addition of the three examples shown in Fig. 3, the algorithm can calibrate 15 of the possible 28 camera pairs, and we are still improving the algorithm to get more homographies (in fact some of the cameras have too little overlap).

6 Conclusion

The external calibration algorithm we present in this paper exhibit several advantages over comparable algorithms in the literature: (1) by using pieces of trajectories adequately sub-divided and by assigning them likelihood values reflecting a priori values for being matched unambiguously, it keeps the computational complexity (driven by the number of iterations to perform in the RANSAC-like process) reasonable; (2) the fact of not relying on the entire trajectories, but instead in smaller parts, makes it much more robust to occlusions that occur with 2D tracking algorithms. We presented rectification results in some challenging situations, where the viewpoint changes make nearly impossible to recover the geometrical transform by traditional point correspondences techniques.

Our ongoing and future work focus in determining uncertainties on the different homographies we compute, and by using these uncertainties to (1) re-optimize all homographies by using relationships of the form $H_{ij} = H_{ik}H_{kj}$ and (2) track moving in objects in 3D, both in a probabilistic manner.

References

1. Du, W., Hayet, J.B., Verly, J., Piater, J.: Ground-target tracking in multiple cameras using collaborative particle filters and principal axis-based integration. IPSJ Transactions on Computer Vision and Applications **1** (2009) 58–71
2. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2003) 257–263
3. Lee, L., Romano, R., Stein, G.: Monitoring activities from multiple video streams: Establishing a common frame. IEEE Trans. on Pattern Analysis and Machine Intelligence **22**(8) (August 2000) 1–10
4. Caspi, Y., Simakov, D., Irani, M.: Feature-based sequence-to-sequence matching. Int. J. Comput. Vision **68**(1) (2006) 53–64
5. Stauffer, C., Tieu, K.: Automated multi-camera planar tracking correspondence modeling. IEEE Conf. in Computer Vision and Pattern Recognition (CVPR'03) **1** (2003) 259
6. Nunziati, W., Sclaroff, S., Del Bimbo, A.: Matching trajectories between video sequences by exploiting a sparse projective invariant representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **PP**(99) (2003) 1
7. Kayumbi, G., Cavallaro, A.: Multi-view trajectory mapping using homography with lens distortion correction. EURASIP J. on Image and Video Processing (2008)
8. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
9. Triggs, B.: Autocalibration from planar scenes. In: In Proc. ECCV. (1998) 89–105

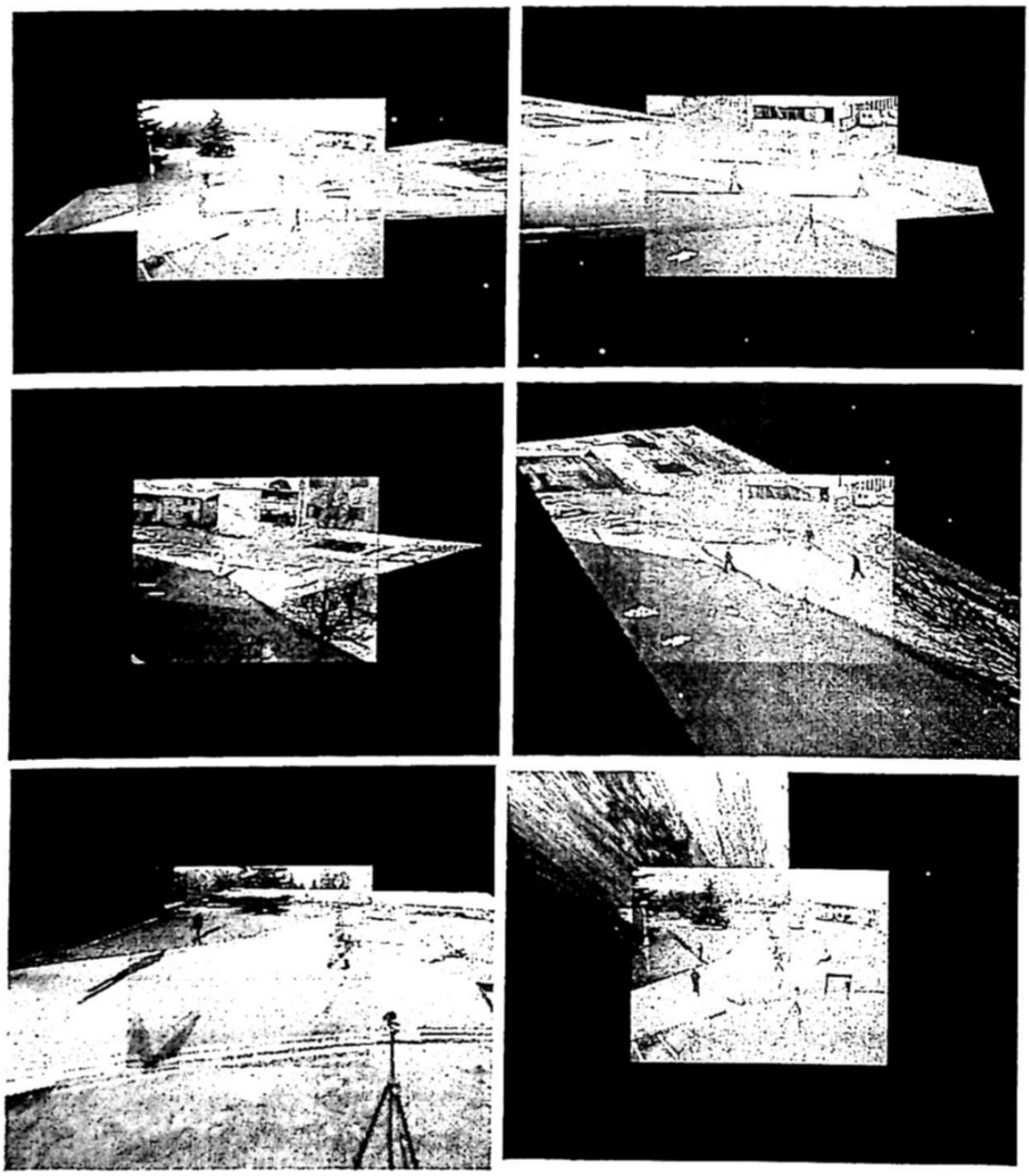

Fig. 3. Video registrations on the PETS2009 [10] sequences. Each column depicts one pair (i, j) and the corresponding homography H_{ij} (up) and its inverse (down). The processed pairs are, from left to right, $(1, 2)$, $(1, 3)$ and $(2, 7)$.

10. : Ieee int. workshop on performance evaluation of tracking and surveillance. http://www.cvg.rdg.ac.uk/PETS2009/ (2009)