Robust Extrinsic Camera Calibration from Trajectories in Human-Populated Environments

Guillermo Baqueiro Victorín[1] and Jean Bernard Hayet[2]

[1] Digipro, Distrito Federal,
Mexico
[2] Centro de Investigación en Matemáticas (CIMAT A.C.), Guanajuato,
Mexico

{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 and compare it to existing approaches.

Keywords. Calibration, computer vision, tracking, video-surveillance and multiple camera systems.

Calibración extrínseca robusta de un sistema de cámaras a partir de trayectorias en ambientes humanos

Resumen. Este artículo propone un nuevo método robusto para realizar las calibraciones inter-cámaras y suelo-cámara en el contexto de vídeo-vigilancia sobre escenas pobladas por humanos. Suponemos que los agentes transitan en un simple plano y que los parámetros intrínsecos de las cámaras son conocidos. Usamos las trayectorias de objetos en movimiento en las imágenes, como por ejemplo las generadas por algoritmos de rastreo del estado del arte, para estimar los parámetros extrínsecos de las diferentes cámaras. Ilustramos el desempeño de nuestro algoritmo sobre diferentes configuraciones experimentales desafiantes, y lo comparamos con diferentes métodos existentes.

Palabras clave. Calibración de cámaras, visión por computadora, rastreo, vídeo-vigilancia y sistemas de cámaras múltiples.

1 Introduction

In spite of its spectacular development, video surveillance is yet largely dependent on human agents in charge of monitoring up to dozens of TV screens, which may be a source of 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 of the network and fusion techniques to recover the 3D trajectories of the moving objects in the scene [3]. Then, this information feeds pre-defined or unusual event detection and may trigger alarms for the agents. 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 and that would allow the data collected throughout all the video streams to be fused properly.

This article presents an algorithm that estimates the extrinsic parameters of a set of different cameras involved in a surveillance network, i.e. the 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 each camera, 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 the two sequences of cameras depicted in Fig. 1. In particular, this prevents us from using traditional feature-based matching techniques based on local descriptors around interest points [7] for estimating the underlying geometric transforms.

Instead, in the vein of the seminal work of [6], 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. In other words, we use the dynamic part of the scene for registering views instead of the static part.

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 see how to recover all the extrinsic parameters from homographies and show how to calibrate the whole camera network; in Section 5, we comment results obtained on different setups and compare our algorithm to other existing works in the literature; finally, Section 6 draws conclusions and introduces future work.

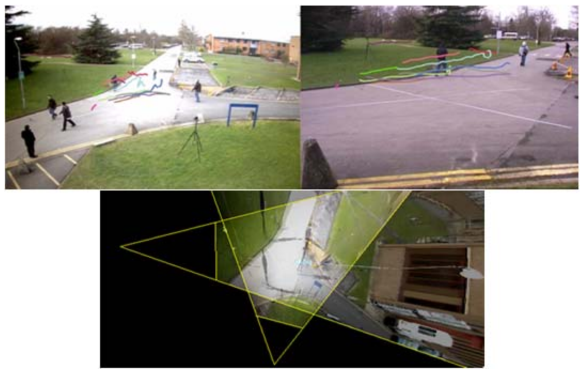
Fig. 1. The typical input/output of our algorithm. Above, we form correspondences (one color, one correspondence) among trajectories (from standard trackers) to compute robustly the feet-to-feet homography H_{ij}, between the two views. These inlier *trajlets* (see Section 3) correspond to the computation of homographies between camera 2 and 7 of the PETS 2009 data (i.e., subfigures of the last line in Fig. 9). Below, the geometry of the scene can be recovered through homographies H_i and H_j from each camera to the reference plane, which is suitable for multi-view tracking. Each field of the view is re-projected with yellow lines

2 Related Work and Contributions

The seminal work of Lee et al. [6] 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 simultaneously detected at a given time-stamp, 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 forms the correspondences from *tracking sequences* [1, 9]. In [9], the authors present a RANSAC-like approach that performs, as we do here, 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. However, the likelihood functions that ponder each sample are not clearly defined.

The work in [1] is more general in a sense, as it is extended to fundamental matrix estimation. It 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 [8] uses perspective invariants, namely the cross ratio of five points, to match trajectories between video sequences. The algorithm also allows calibrating the time offset between video streams. 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. Among the most recent works in the area, the one of [4] is interesting as it also takes radial distortion into account. However, the correspondences are determined on the base of control points manually selected on trajectories, which may make it more adapted for expert users. A common inconvenience of these previous approaches is that they use tracking trajectories directly 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. In many situations, e.g., because of occlusions in crowded scenes, tracking algorithms may be unable to distinguish one object from another and may assign a wrong identity to some tracked object. This may be catastrophic for the estimation of scene geometry.

Our approach takes some of the ideas developed in [1, 9] to cut down the algorithmic complexity of the correspondence problem and brings several contributions including (1) robustness with respect to possible failures of the tracking algorithms, (2) more reliable guidance of the optimization process to the correct geometry, and (3) an optimization process to find the calibration of a set of N cameras. As far as notations are concerned, we will use bold capital letters for matrices, regular letters for scalars and vectors. The indices will generally refer to the camera(s) to which the variable is related.

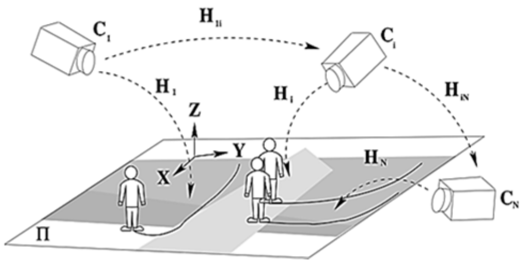

Fig. 2. Setup: cameras C_i for $1 \leq i \leq N$ observe a *planar* scene laid on a plane Π. Inter-camera homographies between cameras i and j are denoted by $\mathbf{H}_{ij}$, while camera-to-reference plane homographies from camera i to Π are denoted by $\mathbf{H}_i$. Each camera has its own (shaded) scene coverage

3 Inter-Video Homography Estimation

3.1 Problem Formulation

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 a *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, the projection of a point P of the plane Π, and $p_j=(u_j,v_j)^T$ is the projection of this same point on camera C_j, then we have the classical relationship in homogeneous coordinates [4],

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

where ~ means that a relation of equality holds for any multiplication factor λ>0, so that $\mathbf{H_{ij}}$ has in fact only 8 degrees of freedom. The problem consists in estimating (1) these transforms and (2) the homographies $\mathbf{H}_i$ that map points P of the reference plane to their projection p_i,

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

where $(X,Y,0)^T$ 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 $\mathbf{H_{ij}}$ by searching for point correspondences (p_i,p_j) and by using them to solve the linear system directly induced by all the instances of Eq. 1. As these correspondences are difficult to find with static scene points and appearance whenever the viewpoint changes strongly, we rely on tracks from video trackers.

Our algorithm can be summarized as follows: (1) **collect trajectories** in each stream V_i with a tracking algorithm, (2) **pre-process** trajectories to eliminate ambiguities at occlusion points (we will refer to the trajectory parts built in this way as *trajlets*), (3) apply the RANSAC-like robust **optimization process** with a likelihood-guided sampling process between all pairs of cameras, and (4) **refine** the whole camera set calibration by non-linear optimization techniques.

3.2 Collect and Pre-Process Trajectories

In the first step of our algorithm, we collect tracking trajectories from all available video streams. We conceived our algorithm to be *robust* w.r.t. the properties of the 2D tracker, so that which tracker to use is not very relevant here. Practically, we used some of the 2D tracker algorithms implemented in the OpenCV library. The result, for a video stream i of camera C_i, is an initial 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 here instead of feet positions because most of the authors seem to agree that they are not as sensitive to noise as the feet position [1, 6, 10]. However, a consequence is that the computed homography will correspond to a *plane* Π^c *passing through target centroids* (i.e., not Π). We denote it as $\hat{\mathbf{H}}_{ij}$ and will see that the feet-to-feet homography $\mathbf{H}_{ij}$ can be estimated from $\hat{\mathbf{H}}_{ij}$.

In the second step, we form what we call *trajlets*, i.e., pieces of trajectories smaller than the initially collected ones in L_i. The idea is to get a second set of trajectories that are not too short, in order for the optimization to remain tractable computationally, but at the same time cut in such a way that they would not be susceptible to be contaminated by errors from the tracking algorithm, e.g. occlusion errors. The latter case is quite common for most tracking algorithms: two tracks that intersect at some point may exchange their respective target identity. In that case, the result is that both tracks are unusable for establishing correspondences. To avoid this, for all the pairs of collected trajectories $(l_i^{(m)}, l_j^{(n)})$ for which some of the points $p_{i,t}, p_{j,t}$ are close in the image (at some timestamp t), we simply cut off the ambiguous parts within a given time radius δ. For each cut on an initial pair $(l_i^{(m)}, l_j^{(n)})$, this process creates four sub-trajectories (*trajlets*) $(l_i^{(m)+}, l_i^{(m)-}, l_j^{(n)+}, l_j^{(n)-})$, such that,

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

Then, we smooth these trajectories by using local filtering based on Bezier curves so as to reduce the impact of noise in the objects' position (this smoothing is evaluated in Section 5). The result of this processing is, again, for each video stream i, an - a priori larger - set Li' = {li(m), m≥0}. Some of these trajlets are drawn in the upper part of Fig. 1.

3.3 Robust Homography Estimation

The estimation of $\hat{\mathbf{H}}_{ij}$ is done in a RANSAC-like scheme described in this section. A priori, a homography candidate for explaining the two images of 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 [1, 8]. 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 inconvenience, since, if we have an order of magnitude of τ *trajlets* appearing at intersecting windows of time, then the probability for a sampled pair to match is roughly 1/τ. Then, the probability for two consecutively sampled *trajlets* to match is $1/\tau^2$. Hence, the number of sampling iterations needed in RANSAC to ensure (in stochastic expectation) that at least one correct pair is sampled is **quadratic** in τ, which can be problematic with crowded scenes.

A solution is to avoid a uniform sampling process by assigning likelihood values to all possible pairs of trajectories and by sampling the trajectories according to these values of likelihoods. We define them through

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

where $N_j(l_i^{(m)})$ stands for the number of *trajlets* in L_j' that have a time overlap with trajectory $l_i^{(m)}$ (the value being defined as if there were no time overlap) and $\Delta(l_i^{(m)}, l_j^{(n)})$ measures the time overlap between trajectories $l_i^{(m)}$ and $l_j^{(n)}$. These two terms (1) penalize the sampling of trajectories that could result **ambiguous to match** (large N_j or N_i) and (2) favor the trajectories with large overlap which improves homography estimation by using larger sequences. Another important point in the sampling process is a geometrical consistency check made on *pairs* of trajectories, according to which the polygon formed by the extremities of two trajectories should just keep or inverse the order of its vertices in other views.

Both criteria make the required number of samples much lower than the aforementioned quadratic term above. In practice, we need a few dozen iterations to get a pair of correctly matched *trajlets*. The remaining part of the process is based on the classical RANSAC scheme and described in Algorithm 1:

Algorithm 1. Computation of homography $\hat{\mathbf{H}}_{\mathbf{ij}}$ 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 $\hat{\mathbf{H}}_{\mathbf{ij}}^{\mathbf{mn}}$ from
   the correspondences between all points of $l_i^{(m)}$
   and $l_j^{(n)}$, by using the classical DLT [5], and
   compute its inverse $(\hat{\mathbf{H}}_{\mathbf{ij}}^{\mathbf{mn}})^{-1}$.
3. For all *trajlets* pairs ($l_i^{(r)}$,$l_j^{(s)}$), compute the
   residual symmetric error $\varepsilon^2(r,s)$:
   $\epsilon^2(r,s) = \frac{1}{2|l_i^{(r)} \times l_j^{(p)}|} \sum_{(p_r,p_s)\in l_i^{(r)} \times l_j^{(p)}} d^2(\hat{\mathbf{H}}_{\mathbf{ij}}^{\mathbf{mn}} p_r, p_s) + d^2(p_r, (\hat{\mathbf{H}}_{\mathbf{ij}}^{\mathbf{mn}})^{-1} p_s).$
4. In the residual matrix $\varepsilon^2(r,s)$, identify 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** $S>\hat{S}$,
          $\hat{S} \leftarrow S$;
          $\hat{\mathbf{H}}_{\mathbf{ij}} \leftarrow \hat{\mathbf{H}}_{\mathbf{ij}}^{\mathbf{mn}}$.
       **end**
**until** a given proportion of trajectories from video
streams $V_i$ and $V_j$ have been explained by $\hat{\mathbf{H}}_{\mathbf{ij}}$ or a
given number of iterations have been done.
**if** $\hat{\mathbf{H}}_{\mathbf{ij}}$ explains enough trajectories in $V_i$ and $V_j$,
   1. Consider $\hat{\mathbf{H}}_{\mathbf{ij}}$ as recovered;
   2. Refine $\hat{\mathbf{H}}_{\mathbf{ij}}$ by a few Levenberg-Marquardt
      iterations on the residual symmetric error
      minimization.
**else**
      Consider the two views as unregistered.
**end**
```

4 Extrinsic Parameters Estimation

In this section, we describe how to recover a geometry of the scene (i.e. the relative position of two cameras), from an inter-image homography. Then, we propose an optimization scheme to calibrate a set of N cameras.

Homography decomposition. Once the homographies $\hat{\mathbf{H}}_{\mathbf{ij}}$ have been recovered as we saw in the previous section, we estimate the extrinsic parameters, i.e. the parameters $\mathbf{R}_{\mathbf{ij}}$, t_{ij} of the rigid 3D transform between the two cameras acquiring video streams i and j. For this purpose, we use the following decomposition of matrix $\hat{\mathbf{H}}_{\mathbf{ij}}$ into intrinsic and extrinsic parameters [4],

$$\hat{\mathbf{H}}_{\mathbf{ij}} \sim \mathbf{K_j R_{ij}}[t_{ij} n_{ij}^T - d_{ij}\mathbf{I_{3\times 3}}]\mathbf{K_i}^{-1} \tag{3}$$

where the matrices $\mathbf{K}_i$ are the intrinsic parameters of cameras i, supposed known here, and where d_{ij},n_{ij} give the equation of the plane (here, the *centroids* plane) in camera i frame, i.e. its equation is $n_{ij}^T\ Q = d_{ij}$, where Q are the coordinates of 3D points in the camera i frame. The indices ij may seem superfluous in n_{ij} and d_{ij} as the plane equation is expressed just in the frame of C_i. However, we will use it to distinguish these estimates from other estimates of the same quantities. For example, vectors n_{ik} resulting from the decomposition of the computed homographies $\hat{\mathbf{H}}_{\mathbf{ik}}$ are also estimates of the normal to Π^c expressed in the frame of C_i.

Note that Eq. 3 is given *only up to a scale factor* that we will determine in a second time. We use Triggs' algorithm [10] to determine the decomposition values of $\hat{\mathbf{H}}_{\mathbf{ij}}$. Note that this algorithm gives two possible pairs for $\mathbf{R}_{\mathbf{ij}}$, t_{ij}, but one of them can be easily discarded. Here, we just select the one that corresponds to the most horizontal configuration of the camera.

Determining a frame on the reference plane.Once the normal n_{ij} to the plane Π^c (and Π, which is parallel) has been computed, a base (e_1',e_2') of vectors generating the centroid plane can be chosen, for example $e_1'=(e_1 \wedge n_{ij}) /||e_1 \wedge n_{ij}||$ and $e_2'=(e_1' \wedge n_{ij})$, where the e_k form the canonical base ($e_1=(1,0,0)^T$). This allows defining a frame associated to Π^c as described in Fig. 3, centered on D_i, the orthogonal projection of the center of projection of C_i onto Π^c. As a consequence, the vector of coordinates of any point Q given in the camera i frame can be written as

$$Q = d_{ij}n_{ij} + Q_{n^{\perp}} = d_{ij}n_{ij} + \alpha e'_1 + \beta e'_2,$$

where (α,β) are the coordinates of Q in the defined frame.

Image plane to ground plane homography. From the computed image-to-image homography $\hat{\mathbf{H}}_{ij}$ and its decomposition, one recovers (up to a scale factor for d_{ij}) an homography to the centroids plane by deriving from the projection equation on C_i of point Q onto point $q = (u_q,v_q)^T$

$$(u_q, v_q, 1)^T \sim \mathbf{K_i} \left(\; e'_1, e'_2, d_{ij}n_{ij} \; \right) (\alpha, \beta, 1)^T, \quad (\qquad (4)$$

from which one derives in terms of the spatial coordinates on the (real) centroid plane, (α,β),

$$(u_q, v_q, 1)^T \sim \mathbf{K_i}(d_{ij}n_{ij} + \alpha e'_1 + \beta e'_2),$$

i.e. $\hat{\mathbf{H}}_i = \mathbf{K_i}$ (e_1' | e_2' | d_{ij} n_{ij}) acts as a homography from the centroid plane Π^c (coordinates α,β) to the image plane in camera C_i. However, an ambiguity remains in this definition as d_{ij} is computed only up to a scale.

Scale recovery. As t_{ij} and d_{ij} are computed from Eq. 3 only up to a scale, we use 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 that has the median velocity. Another one is to assume a known half-height between people's centroids and feet. In both cases, the scale recovery is straightforward.

We will explain it hereafter for the second case, and it can be proven in a similar way for the first one. Let us suppose that we have the knowledge of the half-height of a person, as the quantity L (e.g., 80 cm).

We will denote the different intrinsic parameters of camera C_i as $\alpha_{u,i}$, $\alpha_{v,i}$, $u_{0,i}$ and $v_{0,i}$,

$$\mathbf{K_i} = \begin{pmatrix} \alpha_{u,i} & 0 & u_{0,i} \\ 0 & \alpha_{v,i} & v_{0,i} \\ 0 & 0 & 1 \end{pmatrix}$$

If F and Q are the 3D points corresponding respectively to the feet and centroid of a tracked target (see Fig. 3), then we may write

$$F = \begin{pmatrix} \alpha \\ \beta \\ L \end{pmatrix} \text{ and } Q = \begin{pmatrix} \alpha \\ \beta \\ 0 \end{pmatrix} \qquad (5)$$

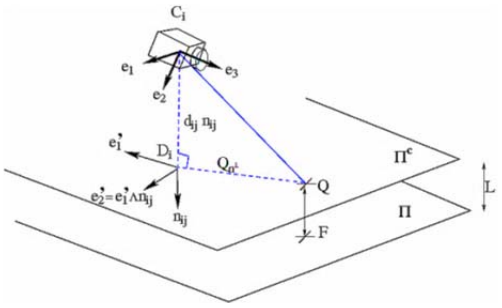

Fig. 3. Definition of a reference frame associated to camera C_i. The origin is set at point D_i which is the orthogonal projection of the center of projection of C_i onto Π. Its axis are defined by vectors e_1' and e_2'

The projections of these two points will be denoted by *f* and *q*, and the half-height as measured in the image of camera C_i is l=||*q-f*||. By using the projection equations, Q is projected onto *q* through

$$q \sim \mathbf{K_i}[\alpha e'_1 + \beta e'_2 + d_{ij}n_{ij}] \sim \begin{pmatrix} \alpha_{u,i}\frac{\alpha {e'_1}^x + \beta {e'_2}^x + d_{ij}n_{ij}^x}{\alpha {e'_1}^z + \beta {e'_2}^z + d_{ij}n_{ij}^z} + u_{0,i} \\ \alpha_{v,i}\frac{\alpha {e'_1}^y + \beta {e'_2}^y + d_{ij}n_{ij}^y}{\alpha {e'_1}^z + \beta {e'_2}^z + d_{ij}n_{ij}^z} + v_{0,i} \\ 1 \end{pmatrix} \qquad (6)$$

and, similarly for F,

$$f \sim \mathbf{K_i}[\alpha e'_1 + \beta e'_2 + (L + d_{ij})n_{ij}] \sim \begin{pmatrix} \alpha_{u,i}\frac{\alpha {e'_1}^x + \beta {e'_2}^x + (L+d_{ij})n_{ij}^x}{\alpha {e'_1}^z + \beta {e'_2}^z + (L+d_{ij})n_{ij}^z} + u_{0,i} \\ \alpha_{v,i}\frac{\alpha {e'_1}^y + \beta {e'_2}^y + (L+d_{ij})n_{ij}^y}{\alpha {e'_1}^z + \beta {e'_2}^z + (L+d_{ij})n_{ij}^z} + v_{0,i} \\ 1 \end{pmatrix}.$$

We can express the observed distance in the image by taking the norm of q-f, l, which leads, by denoting $q=(u_q,v_q)^T$, to

$$l = \frac{L\sqrt{\left[\left((u_q - u_{0,i})n_{ij}^z - \alpha_{u,i}n_{ij}^x\right)^2 + \left((v_q - v_{0,i})n_{ij}^z - \alpha_{v,i}n_{ij}^y\right)^2\right]}}{\alpha {e'_1}^z + \beta {e'_2}^z + (L + d_{ij})n_{ij}^z} \qquad (7)$$

Then, by using Eq. 6, we get two equations in terms of $x_q=(u_q-u_{0,i})/\alpha_{u.i}$ and $y_q=(v_q-v_{0,i})/\alpha_{v,i}$,

$$\begin{cases} x_q(\alpha {e'_1}^z + \beta {e'_2}^z + d_{ij}n_{ij}^z) = \alpha {e'_1}^x + \beta {e'_2}^x + d_{ij}n_{ij}^x \\ y_q(\alpha {e'_1}^z + \beta {e'_2}^z + d_{ij}n_{ij}^z) = \alpha {e'_1}^y + \beta {e'_2}^y + d_{ij}n_{ij}^y, \end{cases}$$

from which we express $\alpha = K_\alpha d_{ij}$ and $\beta = K_\beta d_{ij}$ where K_α and K_β are deduced from the known quantities x_q, y_q, e_1', e_2', and n_{ij}. Now, by plugging these expressions for α and β into Eq. 7, one can deduce an expression for d_{ij},

$$d_{ij} = \frac{L\left[-n_{ij}^{z} + \frac{\sqrt{\left[\left((u_q-u_{0,i})n_{ij}^{z}-\alpha_{u,i}n_{ij}^{x}\right)^2+\left((v_q-v_{0,i})n_{ij}^{z}-\alpha_{v,i}n_{ij}^{y}\right)^2\right]}}{l}\right]}{K_\alpha {e'_1}^{z} + K_\beta {e'_2}^{z} + n_{ij}^{z}} \tag{8}$$

Each blob (q,f) giving an estimate of d_{ij} with the previous development and adult people forming the vast majority of the tracked blobs but not the entirety, we take the median among the different computed estimates for all the estimated inliers of $\mathbf{\hat{H}_{ij}}$. In our experiments, this option assuming the people's heights constant (as opposed to assuming the velocities constant) gave better and much more stable results[1].

Once the scale is recovered, one finally gets either a ground plane-to-image $\mathbf{H_i}$ (as in Eq. 2) or an image-to-image $\mathbf{H_{ij}}$ induced by Π (as in Eq. 1). Most results below illustrate the second form. Note that for any homography we estimate, we will get a different definition of a frame relative to Π. Also note that once the centroid-to-reference plane $\mathbf{\hat{H}_i}$ is estimated, we get immediately the feet-to-plane $\mathbf{H_i}$ by

$$\mathbf{H_i} = \mathbf{K_i}[e'_1, e'_2, (d_{ij}+L)n_{ij}].$$

Calibration of a network of cameras. As described in the previous paragraphs, each pair of cameras (C_i,C_j) gives an estimate of the relative scene geometry for these cameras. Now, to get an estimate of the geometry of the set of N cameras, we proceed as follows. For each possible pair of videos, a calibration process following the previous method is done. The result is a set of external calibration data: $\{(n_{ij},d_{ij},\mathbf{R_{ij}},t_{ij})\}_{ij}$ for all pairs of cameras (i,j) that we could calibrate. For all these pairs, we also get a set of inlier points pairs ($p_i^{[k]}$,$p_j^{[k]}$) from matched trajectories, i.e., $\mathbf{I_{ij}}=\{(p_i^{[k]},p_j^{[k]})\}_k$.s mentioned before, we may get several estimates of the same plane parameters, from different pairs of video streams. For example, the estimation of $\hat{H}_{12}$ and $\hat{H}_{13}$ both lead to estimates of Π^c in the frame of camera 1.

Also, among all the estimates, some constraints should apply. For example, if the equation of Π is estimated in both frames i and j respectively by homographies $\mathbf{H_{ij}}$ and $\mathbf{H_{jk}}$, then we should be able to get the plane parameters expressed in C_i from the ones expressed in C_j and the 3D transformation ($\mathbf{R_{ij}}$, t_{ij})

$$\begin{cases} n_i &= \mathbf{R_{ij}} n_j, \\ d_i &= d_j + n_j^T \mathbf{R_{ij}} t_{ij} \end{cases} \tag{9}$$

To use efficiently all this redundancy, we propose an optimization scheme that cycles over all video streams i, for which it alternates the following steps:

- **Step 'P'** refines the plane parameters (n_i,d_i) expressed in camera i, with transformations ($\mathbf{R_{ij}}$, t_{ij}) fixed, over all cameras j that have been registered with i ($\mathbf{I_{ij}} \neq \emptyset$);
- **Step 'R',** allows to enforce Eq. 9,

$$\begin{aligned} n_i &= \gamma_i n_i + \textstyle\sum_j \gamma_j \mathbf{R_{ij}} n_j, \\ d_i &= \gamma_i d_i + \textstyle\sum_j \gamma_j (d_j + n_j^T \mathbf{R_{ij}} t_{ij}) \end{aligned} \tag{10}$$

where the sums are taken on all cameras j such that an homography has been computed between i and j, and the γj are weights summing to one that are proportional to the total number of inlier points contributing to the estimation of the plane parameters through video j (i.e., a rough level of confidence over the estimation of plane parameters);

- **Step 'T',** for each camera j registered with camera i in the first steps, refine the transformations (Rij, tij) with the plane parameters (ni,di) remaining fixed.

The two refinements (steps 'P' and 'T') are made respectively over the union of all sets of inliers $\mathbf{I_{ij}}$ and over individual sets $\mathbf{I_{ij}}$. The step 'P' tries to minimize a function F_i

$$F_i(n,d) = \frac{1}{2}\sum_{i,j} \sum_{(p_i^{[k]},p_j^{[k]})\in\mathcal{I}_{ij}} \|h(\mathbf{K_j R_{ij}}[I_{3\times3} - \frac{t_{ij}n^T}{d-L}]K_i^{-1}p_i^{[k]}) - p_j^{[k]}\|^2 \tag{11}$$

under the constraint ||n||=1, where h(a,b,c)=(a/c,b/c). The step 'T' tries to minimize, for each registered pair (i,j), a function F_{ij}:

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

$$F_{ij}(\gamma,t) = \frac{1}{2} \sum_{(p_i^{[k]},p_j^{[k]}) \in \mathcal{I}_{ij}} \|h(\mathbf{K_j}\mathbf{R}(\gamma)[\mathbf{I_{3\times3}} - \frac{tn_i^T}{d_i - L}]\mathbf{K_i^{-1}} p_i^{[k]}) - p_j^{[k]}\|^2 \tag{12}$$

where γ is a triple of angles, and t is the translation vector. The outputs of the steps 'T' are then used again in the following 'P' steps (as new values for $\mathbf{R}_{ij}$ and t_{ij}) in the next iterations of this cycle. For each of these steps, we used a classical Levenberg-Marquardt (LM) approach.

Algorithm 2. Scene calibration.

```
$\mathbf{I}_{ij}$ ← Apply algorithm 1 between all pairs (i,j).
Decompose all $\hat{\mathbf{H}}_{ij}$'s into $\mathbf{R}_{ij}$, $t_{ij}$, $n_{ij}$, $d_{ij}$ using [10].
Determine the scale of $t_{ij}$, $d_{ij}$ by using Eq. 8 on each blob in $\mathbf{I}_{ij}$ and taking the median.
Initialize $n_i$, $d_i$ by averaging all $n_{ij}$,$d_{ij}$
repeat
    for camera $C_i$ do
        Refinement LM steps on $n_i$,$d_i$ on the objective function given by Eq. 11.
        Relaxation scheme on $n_i$ (Eq. 10).
        Refinement LM steps on $\mathbf{R}_{ij}$, $t_{ij}$ on the objective function given by Eq. 12.
    end
until convergence
Choose reference camera ı with the highest $\sum_j |\mathbf{I}_{ıj}|$, compute a reference frame on Π and a homography $\mathbf{H}_ı$ from this frame to $C_ı$, (Eq. 4).
For all cameras j connected to ı by a path of homographies, compute the reference plane-to-image $\mathbf{H}_{ıj}$.
```

The scene calibration is summed up in algorithm 2, which makes use of the previously described non-linear optimization steps. Once these steps are done, we simply select the most promising camera ı in terms of inliers contributing to the estimations of $\mathbf{H}_{ıj}$ and for all cameras j that can be connected to ı by a path of homographies (e.g., $\mathbf{H}_{ık}$, $\mathbf{H}_{kl}$, $\mathbf{H}_{lj}$), we deduce $\mathbf{H}_j$ from Eq. 4 and the estimated relative transforms.

5 Results

We tested and compared our algorithm on the PETS [11] benchmark data. This database provides several datasets, with increasing levels of difficulty for the tracking algorithms, i.e., different levels of people density. Each dataset gives eight video streams which makes it suitable for our evaluation needs. For our calibration purposes, we used the medium density crowd dataset (labelled as S0).

As mentioned before, this work has been implemented entirely in C++ with the OpenCV library. It uses a classical tracking algorithm from OpenCV (color-based particle filter). Moreover, we use a somewhat more efficient variant of the RANSAC algorithm, LO-RANSAC [2] that locally optimizes the estimation of the model after each improvement of the current winner model.

We first give some qualitative results of camera-to-camera homography registrations $\mathbf{H}_{ij}$ in Fig. 9. We depict the registrations obtained with three of the computed homographies $\mathbf{H}_{ij}$ (left) and their inverse $\mathbf{H}_{ij}^{-1}$ (right) by warping the image j onto the image plane i, all of these without applying yet the optimization from Section 4. Ideally, if the homographies were correct, all the elements of the scene belonging to the plane Π should coincide. One can note that most of the recognizable roads, lines, spots are correctly warped in the other view. Fig. 1 depicts the inliers *trajlets* (i.e., the matched data in the RANSAC process of Alg. 1) that allowed the estimation of the third homography in Fig. 9 (i.e. between views 2 and 7). In addition to these three examples of Fig. 9, the algorithm calibrates 15 of the possible 28 camera pairs. When pairs are not calibrated, it is mainly because of a too little overlap between the cameras or due to some large off-the-plane obstacles (e.g., trees) that make the track correspondences difficult to find.

Another qualitative result is depicted in Fig. 4: the output of Algorithm 2 for cameras 1, 2, 3, 4, 5, 6, and 7, i.e. computed homographies $\mathbf{H}_1$, $\mathbf{H}_2$, $\mathbf{H}_3$...

The images at some timestamp t from all these video streams i are projected with the homographies $\mathbf{H}_i$ onto the bird view over the real scene.

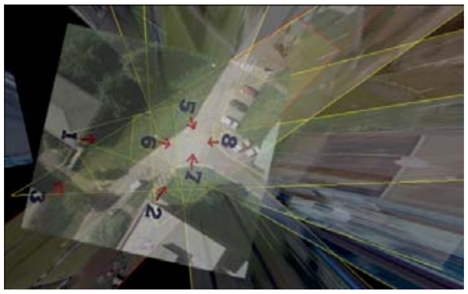

Fig. 4. This mosaic results superposes (1) the rectification of the images of cameras 1, 2, 3, 4, 5, 6, 7 on the ground plane, with fields of view depicted in yellow and (2) an aerial image given on the PETS2009 [11] website that indicates each camera position by the blue numbers

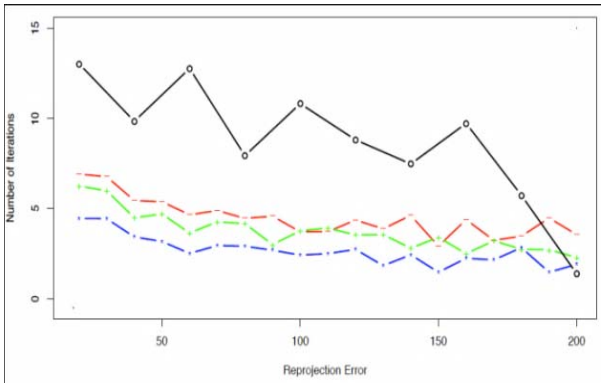

Fig. 5. Number of iterations (y-axis, in log/scale) necessary for the calibration process to reach a given precision level (x-axis). The more iterations are, the slower the algorithm is. We depict the required number of iterations for [6] in black, for the standard RANSAC algorithm [1], in red, for [9], in green, and for our algorithm, in blue

The field of view of each camera is materialized through yellow angular sectors (or red angular sectors if the registration is not done directly by one homography to the reference cameras, but by several ones). To compare it with the ground truth data given with the dataset, we manually superimposed an aerial view of the zone where all camera positions are indicated by blue numbers. Ideally, the vertices of the field of views should indicate the positions of the cameras. As it can be observed, the position errors are in the order of dozens of centimeters.

Quantitatively speaking, in Fig. 5 we compare our approach with different algorithms in the literature, as far as computational efficiency is concerned. To evaluate this efficiency, we plot the average numbers of iterations the RANSAC loop has to perform (y-axis) before reaching a given level of precision (x-axis), i.e., satisfying decreasing levels of quality. These numbers are averaged over 11 runs and presented in log scale for better visualization. In black, we plot the number of iterations necessary for the blob-based algorithm of [6], which is far above all the others, as it has been pointed out already, because of its intrinsic higher complexity. In red, we plot the standard RANSAC algorithm such as [1]. In green, we plot the result of [9] and in blue the one of our algorithm: the last three have a rather similar behavior, but ours gives systematically lower time requirements to reach a given precision bound because of (1) the non-uniform sampling scheme and (2) the pre-processing of trajectories into *trajlets*.

Conversely, in order to evaluate our algorithm in terms of precision and robustness, we compared the image-to-image homographies estimated by Alg. 1 to the ones computed through (a) a naive implementation of RANSAC (similar to [1]) and (b) a better implementation with non-uniform sampling (similar to [9]). These three schemes are evaluated for fixed numbers of iterations (x-axis) in Fig. 6, for the pairs 1-2 (left) and 2-7 (right). The results we obtained for the other pairs are similar. The comparison is made on per-pixel average re-projection errors, measured by the Euclidean distance between the projections by $\mathbf{H}_{ij}$ of points in images i onto images j and the projections of these same points with ground-truth homographies $\mathbf{G}_{ij}$. We took the mean error over 15 different runs. One can notice that the mean error for the first two schemes are quite high and unstable because of the presence of outlier trajectories in the RANSAC scheme that may be inserted in the estimation, whereas ours converges much more stably to its best value.

In Fig. 7, we depict two outputs of the non-linear optimization scheme of Section 4. In the left part, we plot the error residuals of functions F_{ij} (y-axis) that, as expected, tends to drop first and then remains stable with the number of steps (x-axis). In the right part, we plot the norm n_1-$\mathbf{R}_{12}n_2$,

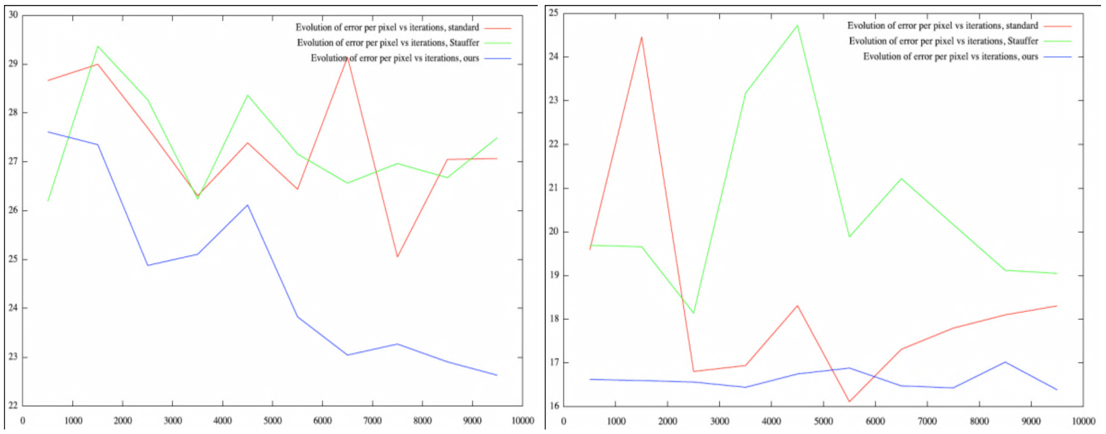

Fig. 6. Comparison of the reprojection errors (i.e., registration quality) obtained as a function of the number of iterations in the RANSAC scheme. The plots are mean values over 15 registration intents and are relative to two pairs of videos (1-2 on the left, 2-7 on the right). In red, results with the standard RANSAC ([1]), in green, with the algorithm of [9], and in blue, with our approach

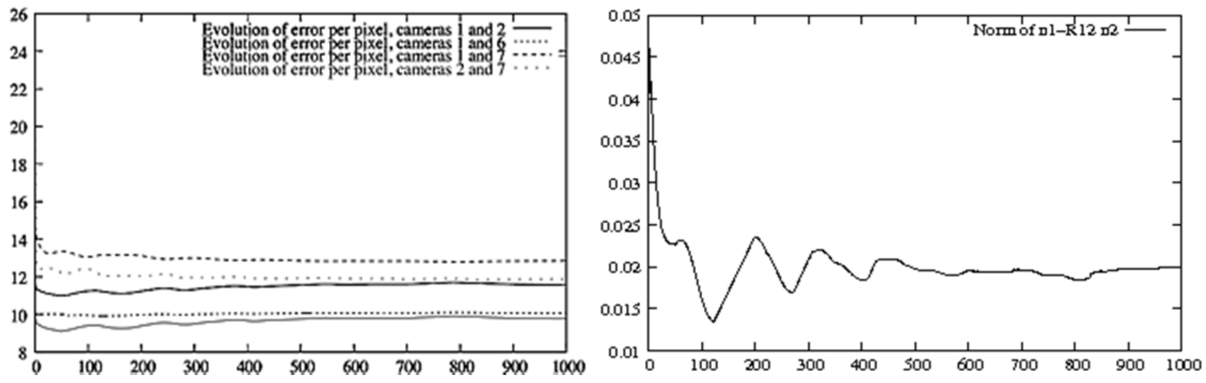

Fig. 7. Two outputs of the non-linear optimization scheme applied to scene calibration, based on camera pairs 1-2, 1-6, 1-7, and 2-7, that makes the set of estimates of pairwise geometries evolve towards consistent values. Left, evolution of the value of the objective functions F_{ij} on a per pixel basis. Right, evolution of the norm of the geometric coherence term n_1-$\mathbf{R}_{12}n_2$.

which, as we explained it previously, is extracted from two separate estimations, but theoretically should vanish. It can be observed that because of our relaxation step in Alg. 2, this geometrical consistency term n_1-$\mathbf{R}_{12}n_2$ tends to be minimized with the number of steps.

Finally, in Fig. 8 we present the distributions of the same reprojection error of Fig. 6, but for two versions of our algorithm, i.e., with (right) and without (left) smoothing of the trajectories. We which, as we explained it previously, is extracted show the results for pair (1,6) on the left, while on the right for pair (2,7). What can be shown is that the median (bold horizontal line) error is not necessarily better with smoothing, but there are much less outlier situations (unfilled dots), that is why we have chosen to smooth trajectories.

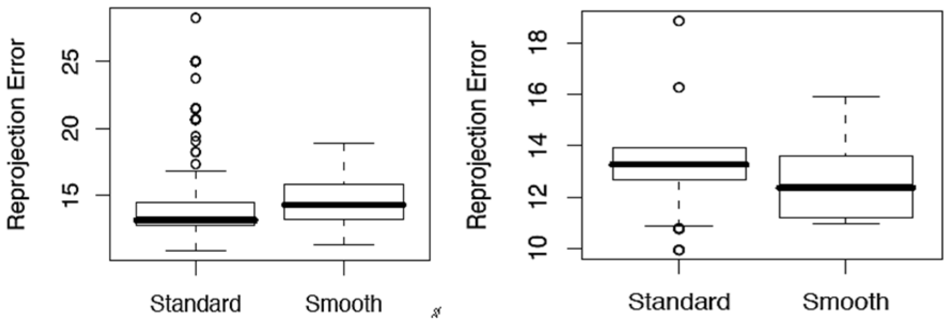

Fig. 8. Distributions of reprojection errors for non-smoothed (left) and smoothed (right) trajectories. The left data is relative to pair (1,6), the right data is relative to pair (2,7)

6 Conclusions

We presented an algorithm for the calibration of a set of video surveillance cameras. It has several advantages over comparable algorithms in the literature as follows:

(1) by pre-processing trajectories into *trajlets* and by assigning likelihood values to pairs of them so that unambiguous matches are favored, it keeps the computational complexity reasonable;

(2) as we do not rely on entire trajectories, but instead on smaller parts less susceptible to be erroneous, it is much more robust to the occlusion problems accompanying standard 2D tracking algorithms;

(3) its non-linear optimization step incorporates all geometrical consistency constraints between the cameras, so that the final estimates between the pairs are not contradicting each other.

We presented rectification results in challenging situations where the viewpoint changes makes it nearly impossible to register the views by traditional point correspondences techniques, and we showed that our own approach outperforms existing ones when efficiency is concerned, i.e., the number of iterations needed for the calibration process to reach a given level of precision is always lower with our algorithm. Moreover, to our knowledge, there is no other work in the literature that goes further the pairwise camera calibration to perform the calibration of a whole network of cameras while taking the consistency of the computed estimates into account.

As ongoing and future work, we aim to model and compute probabilistic uncertainties on the d estimated homographies based on the uncertainties on measured trajectories. We plan to use them

(1) in optimization, to favour less uncertain homographies and

(2) in a 3D people tracking scheme over the camera network.

References

1. **Caspi, Y., Simakov, D., & Irani, M. (2006).** Feature-Based Sequence-to-Sequence Matching. *International Journal of Computer Vision,* 68(1), 53–64.
2. **Chum, O., Matas, J., & Kittler, J. (2003).** Locally Optimized RANSAC. *25th DAGM Symposium, Lecture notes in Computer Science, 2781,* 236–243.
3. **Du, W., Hayet, J.B., Verly, J. & Piater, J. (2009).** Ground-Target Tracking in Multiple Cameras Using Collaborative Particle Filters and Principal Axis-Based Integration. *IPSJ Transactions on Computer Vision and Applications,* 1, 58–71.
4. *Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance.* (PETS 2009). Retrieved from: http://www.cvg.rdg.ac.uk/PETS2009/

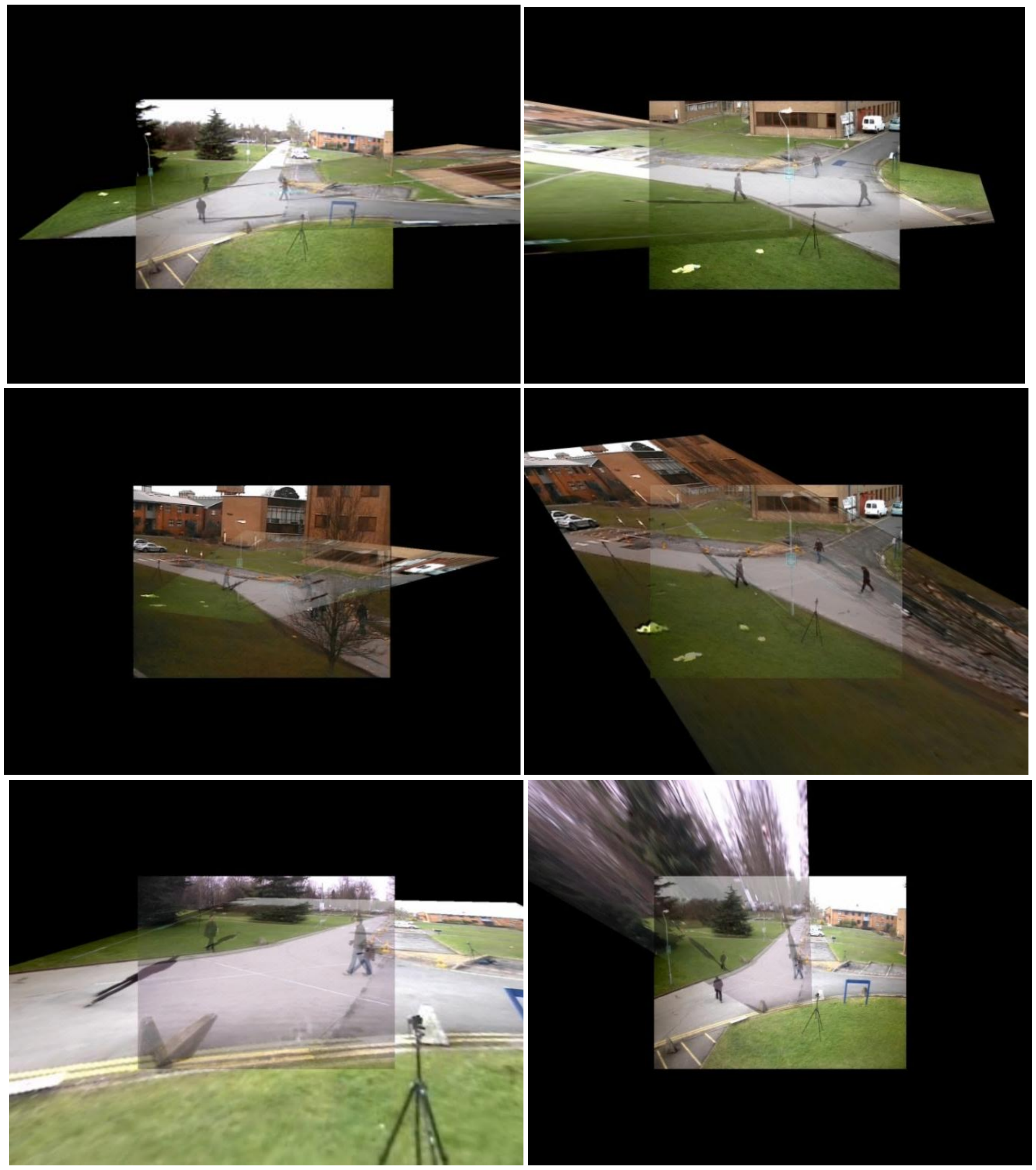

Fig. 9. Video registrations on the PETS2009 [11] sequences. Each row depicts one pair (i,j) and the corresponding homography $\mathbf{H}_{ij}$ (left) and its inverse (right). The processed pairs are, from up to down, (1,2), (1,3), and (2,7)

5. **Hartley, R. & Zisserman, A. (2000).** *Multiple View Geometry in Computer Vision,* Cambridge, UK: Cambridge University Press.
6. **Kayumbi, G. & Cavallaro, A. (2008).** Multi-view trajectory mapping using homography with lens distortion correction. *EURASIP Journal on Image and Video Processing, 2008,* Article ID 145715.
7. **Lee, L., Romano, R., & Stein, G. (2000).** Monitoring Activities from Multiple Video Streams: Establishing a Common coordinate Frame. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 22(8), 758–767.
8. **Mikolajczyk, K. & Schmid, C. (2005).** A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 27(10)*,* 1615–1630.
9. **Nunziati, W., Sclaroff, S., & Del Bimbo, A. (2010).** Matching Trajectories between Video Sequences by Exploiting a Sparse Projective Invariant Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 32(3), 517–529.
10. **Stauffer, C. & Thieu, K. (2003).** Automated multi-camera planar tracking correspondence modelling. *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* Madison, WI, USA, 259–266.
11. **Triggs, B. (1998).** Autocalibration from Planar Scenes. *5th European Conference in Computer Vision (ECCV'98)*, Freiburg, Germany, 1, 89–105.

Guillermo Baqueiro graduated in Mathematics from the Autonomous University of Yucatan in 2007. He got his Master degree in Computer Science at CIMAT in 2010 and joined Digipro SA de CV in the same year.

Jean-Bernard Hayet graduated from ENSTA (Paris), University Paris VI and got his Ph.D. from U. of Toulouse (2003). He has been post-doc fellow at U. of Liege. Since 2007, he has worked at CIMAT in Guanajuato, Mexico. He teaches computer science and does research in robotics and computer vision.

Article received on 11/03/2010; accepted on 04/02/2011.