

# Unsupervised Camera Network Structure Estimation Based on Activity

†Pierre Clarot ††Erhan B. Ermis †Pierre-Marc Jodoin ††Venkatesh Saligrama

† Université de Sherbrooke, Dépt. d'Informatique  
2500 Boul. de l'Université  
Sherbrooke, Qc, Canada, J1K 2R1  
[pierre-marc.jodoin, pierre.clarot]@usherbrooke.ca

†† Boston University, Electrical and Computer Eng.  
8 Saint Mary's Street  
Boston MA 02215, USA  
[ermis,srv]@bu.edu

**Abstract**—In this paper we consider the problem of unsupervised topology reconstruction in uncalibrated visual sensor networks. We assume that a number of video cameras observe a common scene from arbitrary and unknown locations, orientations and zoom levels, and show that the extrinsic and calibration matrices, fundamental and essential matrices, the homography matrix, and the physical configuration of the cameras with respect to each other can be estimated in an unsupervised manner. Our method relies on the similarity of activity patterns observed at various locations, and an unsupervised matching method based on these activity patterns. The proposed method works in cases with cameras having significantly different orientations and zoom levels, where many of the existing methods cannot be applied. We explain how to extend the method to a multicamera case where more than two cameras are involved. We present both qualitative and quantitative results of our estimates, and conclude that this method can be applied in wide area surveillance applications in which the deployed systems need to be flexible and scalable, and where calibration can be a major challenge.

## I. INTRODUCTION

A common goal when dealing with camera networks in automated systems is to recover the topology of the network, by which we mean the extrinsic and calibration matrices, fundamental and essential matrices, the homography matrix, and the physical configuration of the cameras with respect to each other. Such topology information is fundamentally important in order to relate events observed by the camera network as well as to estimate the position and distance of the objects in the physical three dimensional world.

In general, the topology can be reconstructed through a calibration procedure in which two dimensional points in each image, together with their three dimensional positions in the scene, are used in a regression [1]. Since the exact size of objects (cars, buildings, trees, etc.) in the scene are rarely known, usually a calibration object (such as a cube with a checkerboard texture on it) with known properties is used. However, when dealing with real-world scenarios or with a dynamic camera network, which can require frequent updates (due to cameras being added, removed, moved on a regular basis), it is difficult to include a calibration object to the scene (e.g., a highway or a crowded mall) every time calibration is required.

Autocalibration of a network of cameras has long been studied ([2] contains an exhaustive survey on this topic). The topology can be estimated up to a projective transformation [2]

based on the projection  $\mathbf{p}_{ij} = (x, y)$ , in each camera  $C_i$ , of a series of points  $\mathbf{P}_j = (X, Y, Z)$  in the actual three dimensional scene.  $\mathbf{p}_{ij}$  can be estimated by matching feature points (typically corners) with similar characteristics across cameras [2].

Interestingly, Devarajan *et al.* [3] proposed a decentralized auto-calibration method based on feature matching. This method is useful for wireless devices with power and/or broadband limitations. This is different from our method which requires centralized processing of video. This however is not a major issue for our method since, as will be shown, our method is fairly robust to frame drops as video frames come with a time stamp which allows to keep each video temporally aligned.

Due to different camera parameters (white balance, opening of the shutter, etc.), atmospheric degradation caused by large distances between the objects and the cameras, or simply different points of views, matching feature points across cameras is not always a reliable approach. In particular, when the cameras observe the scene from significantly different orientations and with different zoom levels, many of the existing methods do not perform well.

In this work we consider a network of uncalibrated cameras, i.e., with unknown location, orientation, as well as zoom level, and propose a method to reconstruct the topology in the sense described above, i.e, estimate the extrinsic and calibration matrices, fundamental and essential matrices, the homography matrix, and the physical configuration of the cameras with respect to each other. Our method does not require the placement of any particular calibration object. Instead, it uses the activity patterns observed in a given location, which turn out to be invariant to the observation geometry, i.e., the camera locations, zoom levels, orientations, etc., an idea first proposed in [4].

Using the geometry independence of activity patterns, we obtain a set of matching pairs of points between the frames of each camera through a novel method. Once we obtain the matching results, we reverse engineer the configuration of the cameras up to a rotation and translation of the coordinate systems. In the sequel, we describe the proposed method and give examples from real life scenarios. We present both qualitative and quantitative results of our estimates, and conclude that this method can be applied in wide area surveillance applications,

in which the deployed systems need to be flexible and scalable, and where manual calibration can be a major challenge.

## II. MATCHING USING ACTIVITY PATTERNS

### A. Geometry Independence of Activity

The idea of geometry independence of activity patterns observed at a given location was introduced in [4]. The premise there is that irrespective of the location, orientation, and zoom level of the cameras, the occupancy duration of the pixels corresponding to a given location remains fixed. The authors described this principle over a two dimensional setup, and although the two dimensional setup captures the essence of their idea, it falls short of addressing the scenarios encountered in real life scenarios. Here we extend their geometry independence principle to more general cases and investigate the discrepancy between the ideal two dimensional and non-ideal three dimensional setups.

Consider a cuboid object that moves over a point  $x_0$  on a surface. We describe the idea over a cuboid object, since a cuboid bounding box around any object can be drawn, and a majority of the objects can be approximated by this bounding box. Assume that the object moves with velocity  $v$ , and its length in the direction of motion is  $l$ . Also assume that two infinite resolution cameras observe the object from different views with different zoom levels. Let  $\alpha$  be the ratio of the object's height  $h$  to its length  $l$ , e.g., 1/4 for a car, 1/10 for a truck, and about 6 for people. Let  $\theta_i$  and  $\phi_i$  be the observation angles defined as in Fig. 1 for Camera  $i$ ,  $i = 1, 2$ . In each camera's frame (projection plane) there is a point that corresponds to  $x_0$ . Call these points  $\mathbf{p}$  in Camera 1, and  $\mathbf{q}$  in Camera 2.

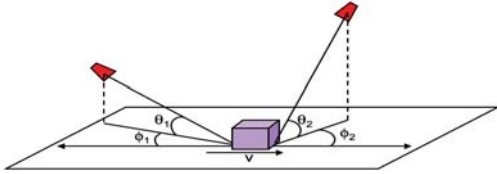


Fig. 1. Multicamera observation angles.

Let  $t = l/v$  be the actual occupancy duration of  $x_0$  by the object. Define  $t_p$  and  $t_q$  as the occupancy duration of pixels  $\mathbf{p}$  and  $\mathbf{q}$  by the projection of the object in each camera. First consider the case where  $\phi_1 = 0$ , i.e., Camera 1 is placed at some height along the direction of motion. Assume for simplicity that the camera is placed in front of the object. In this case, the back-projection of the object's image onto the surface will have length

$$l_{C_1} = l + h/\tan(\theta_1) \quad (1)$$

where  $h$  is the height of the object. Rewriting this in terms of the aspect ratio, we have

$$l_{C_1} = l + \alpha l/\tan(\theta_1) = l\left(1 + \frac{\alpha}{\tan(\theta_1)}\right). \quad (2)$$

Observe that the occupancy duration of  $\mathbf{p}$  will be the same as the amount of time it takes for the back-projection to cross over  $x_0$ . Then, we have

$$t_p = l_{C_1}/v = \frac{l}{v}\left(1 + \frac{\alpha}{\tan(\theta_1)}\right) \quad (3)$$

$$= t\left(1 + \frac{\alpha}{\tan(\theta_1)}\right). \quad (4)$$

Now, as we increase the angle  $\phi_1$  from zero to 180 degrees, the extension term ( $\alpha l/\tan(\theta_1)$ ) in  $l_{C_1}$  begins to shrink. At 90 degrees, it becomes precisely zero, and at 180 degrees it again becomes  $\alpha l/\tan(\theta_1)$ . Therefore, to model this effect, we multiply the extension term with a function  $\chi(\phi_i)$ , which is bounded to  $[0,1]$ , and takes a value of 0 at  $\phi_i = 90$  and 1 at  $\phi_i = 0$  and  $\phi_i = 180$ . This leads to the following approximations for  $t_p$  and  $t_q$ :

$$t_p = t\left(1 + \frac{\alpha}{\tan(\theta_1)}\chi(\phi_1)\right), \quad (5)$$

$$t_q = t\left(1 + \frac{\alpha}{\tan(\theta_2)}\chi(\phi_2)\right). \quad (6)$$

Here we are only interested in accounting for the scaling effect for a given  $\alpha$ . Therefore, we can define  $\gamma_i = \chi(\phi_i)/\tan(\theta_i)$ , and treat the scaling factor as a single parameter. With this setup the observations belong to a family of signals parameterized by a single parameter and can be accounted for without the knowledge of  $\theta_i$  and  $\phi_i$  through statistical methods. Furthermore, the uniqueness property presented below guarantees reliable disambiguation among different pixels, and this information can be used for topology reconstruction.

**Remark:** For many real-world objects, the expressions  $t\left(1 + \frac{\alpha}{\tan(\theta_i)}\chi(\phi_i)\right)$  turn out to be upper-bounds on the occupancy durations.

*Lemma 2.1:* Let  $\mathbf{p}$  be a pixel in Camera 1, and  $\mathbf{q}$  be a pixel in Camera 2. If  $\mathbf{p}$  and  $\mathbf{q}$  do not observe the same location, and if the events occur randomly in the observed region, then the Hamming distance between the time series of  $\mathbf{p}$  and  $\mathbf{q}$  is positive with high probability for sufficiently long video sequences.

Notice that nowhere in our development of geometry independence did we use any assumptions about the zoom levels of cameras. Hence, the zoom levels are irrelevant features in our setup. We state this as a lemma below.

*Lemma 2.2:* The time series of a particular location is invariant to different zoom levels with which it is observed.

*Discrepancy Characterization:* In order to characterize the discrepancy between the observed and actual occupancy rates, we set up an experiment using 3ds Max, where we placed cameras for all combinations of  $\theta = \{15, 30, \dots, 90\}$  degrees and  $\phi = \{0, 15, \dots, 90\}$  degrees. In total we had 36 cameras covering the first octant, all observing the same point in the middle of their field of view. Note that once  $\theta = 90$  there is no need to vary  $\phi$  hence 36 cameras and not 42.

Next we made two cuboids to simulate the bounding box of a car and the bounding box of a human, both with appropriate dimensions. These objects moved through the point and we recorded the videos with each of the cameras. Then we

		Phi							
		0	15	30	45	60	75	90	
Theta	15	2.05	1.87	1.44	1.26	1.15	1.05	1	
	30	1.49	1.49	1.38	1.23	1.13	1.05	1	
	45	1.28	1.28	1.26	1.21	1.13	1.05	1	
	60	1.15	1.15	1.13	1.10	1.08	1.05	1	
	75	1.08	1.08	1.05	1.05	1.03	1.03	1	
	90	-	-	-	-	-	-	1	

(a) Cuboid for cars

		Phi							
		0	15	30	45	60	75	90	
Theta	15	20.8	3	2.2	1.4	1.6	1.2	1	
	30	10.4	2.8	2.2	1.6	1.4	1.2	1	
	45	6.4	2.8	1.8	1.6	1.4	1.2	1	
	60	4.2	2.8	1.8	1.6	1.4	1.2	1	
	75	2.6	2.4	1.8	1.6	1.4	1.2	1	
	90	-	-	-	-	-	-	1	

(b) Cuboid for humans

Fig. 2. Table of discrepancy ratios ( $\tau$ ) for various angles of observation. The cuboid for cars has aspect ratio  $\alpha = 1/4$  whereas the cuboid for humans has aspect ratio  $\alpha = 6$ .

looked at the occupancy duration of the point of interest in each camera, and finally calculated the ratio of observed occupancy duration to that of actual, i.e.,  $\tau = t_p/t$ . The results are presented in Fig. 2 for both cuboids. Once the singular observation angles are ignored  $\tau$  has a narrow range, which also narrows the parameter search space.

---

#### Algorithm 1 Activity Matching

---

**Input:**  $V_1, V_2, \mathbf{p}$   
**Output:**  $\mathbf{q}$

- 1: Estimate  $s(\mathbf{p}, \mathbf{q})$  following Eq. (8)
- 2:  $t=0, Q^t = \{\tilde{\mathbf{q}} : s(\mathbf{p}, \mathbf{q}) \geq 0.9\}$
- 3: Find center of mass  $C^t = (C_x^t, C_y^t)$  of  $Q^t$  with LS
- 4: **for each**  $\tilde{\mathbf{q}} \in Q^t$  **do**
- 5:  $d_E(C^t, \tilde{\mathbf{q}}) = \sqrt{(C_x^t - \tilde{q}_x)^2 + (C_y^t - \tilde{q}_y)^2}$
- 6: **end for**
- 7: Let  $d_Q^t = \{d_E(C^t, \tilde{\mathbf{q}})\}$  and  $\text{med}_Q^t \doteq \text{median}\{d_Q^t\}$
- 8: Set  $Q^{t+1} = \{\tilde{\mathbf{q}} : \tilde{\mathbf{q}} \in Q^t, d_E(C^t, \tilde{\mathbf{q}}) \leq \text{med}_Q^t\}$
- 9: Calculate the new center of mass  $C^{t+1}$  with LS
- 10: **if**  $d_E(C^t, C^{t+1}) > \gamma$  **then**
- 11:  $t=t+1$  and go to step 2
- 12: **end if**
- 13:  $\mathbf{q} = C^t$

---

#### B. Multicamera matching

In order to perform a multicamera point-to-point matching, we first subtract the background from the video. Researchers have investigated a number of methods to perform motion detection and background subtraction (see [5], [6], [7] and references therein). In this paper, a simple background subtraction technique is being used. Once the background is subtracted we obtain a binary motion video  $V(\cdot, \cdot)$ , where  $V(\mathbf{p}, \tau)$  denotes the binary value of a pixel  $\mathbf{p}$ , in frame number  $\tau$ . Throughout the paper  $V_1$  will denote the binary video obtained from Camera 1,  $V_2$  will denote the binary video obtained from Camera 2. Similarly,  $\mathbf{p}$  will denote a pixel in Camera 1 and  $\mathbf{q}$  will denote a pixel in Camera 2. For simplicity we assume that both  $V_1$  and  $V_2$  are composed of  $m$ -pixel frames.

The multicamera matching problem is then described as follows: given a number of cameras,  $C_1, C_2, \dots, C_k$  and their observation matrices (frames)  $f_1, f_2, \dots, f_k$ , the matching between any two cameras is a function (look-up table)  $\mathcal{T}_{i,j}$  that maps a pixel  $\mathbf{p}$  in  $f_i$  to its corresponding pixel  $\mathbf{q}$  in  $f_j$ . Without loss of generality we will present our results for a two-

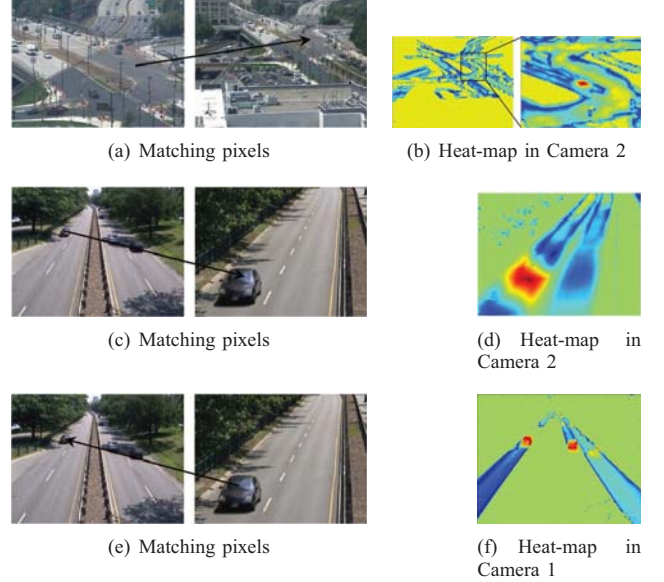


Fig. 3. The similarity map corresponds to  $s(\mathbf{p}, \mathbf{q})$  where red indicates  $s(\mathbf{p}, \mathbf{q}) = 1$ , green indicates  $s(\mathbf{p}, \mathbf{q}) = 0.5$  and blue indicates  $s(\mathbf{p}, \mathbf{q}) = 0$ .

camera case, and note that the extension to multiple cameras is straight forward.

#### C. Mapping Activity Regions

Let  $\mathbf{p}$  be a pixel in Camera 1, and  $V_1(\mathbf{p}, \cdot)$  be its binary time series. For each pixel  $\mathbf{q}$  in Camera 2, we compute a variation of Euclidean distance between  $\mathbf{p}$  and  $\mathbf{q}$ , i.e.,

$$d(\mathbf{p}, \mathbf{q}) = \frac{1}{\eta} \sqrt{\sum_{\tau=1:T} (V_1(\mathbf{p}, \tau) - V_2(\mathbf{q}, \tau))^2}, \quad (7)$$

where  $\eta = \max\{\sum_{\tau=1:T} V_1(\mathbf{p}, \tau), \sum_{\tau=1:T} V_2(\mathbf{q}, \tau), 1\}$  is a normalization factor, and  $T$  is the length of the video sequences. The normalizing constant  $\eta$  diminishes the effect of small discrepancies in sequences where there is a large amount of activity, yet retains the importance of errors when there is little activity.

Let  $d_p^{max} = \max_{\mathbf{q}} d(\mathbf{p}, \mathbf{q})$  and  $d_p^{min} = \min_{\mathbf{q}} d(\mathbf{p}, \mathbf{q})$ . We then find a similarity measure between the time series of  $\mathbf{p}$  and  $\mathbf{q}$  as

$$s(\mathbf{p}, \mathbf{q}) = \frac{d_p^{max} - d(\mathbf{p}, \mathbf{q})}{d_p^{max} - d_p^{min}}, \quad (8)$$

where  $s(\mathbf{p}, \mathbf{q}) \in [0, 1]$ .

Before we formally describe how we obtain the matching pixel in Camera 2, we present some heat-maps to provide the reader with an intuitive understanding of our method. These heat-maps correspond to the similarity function between a pixel in one camera and all the other pixels in the other. Fig. 3 presents a preview of matched pixels and associated heat-maps. When there is very little activity in the region of observation, as in the cases where a very short video

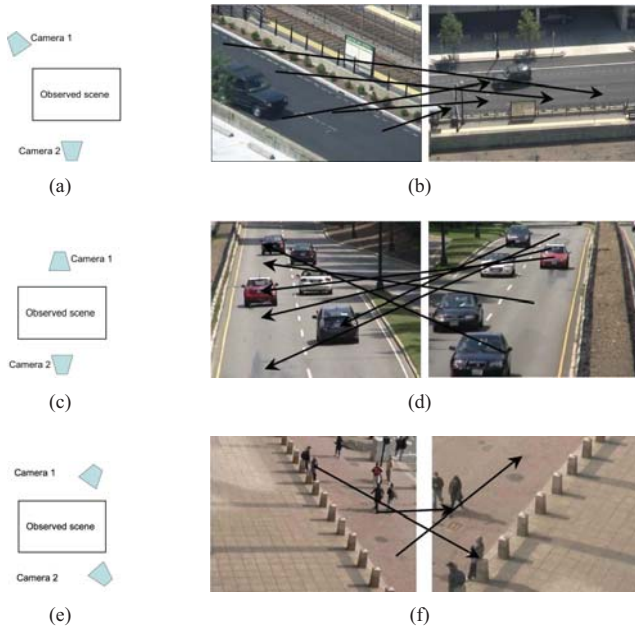


Fig. 4. (a,c,e) Camera setup, (b,d,f) Matching results using 90 seconds of video. The cameras are not calibrated to have the same zoom levels. The pixels can be matched across cameras in low or high-activity scenarios.

is available, pixels in different regions may exhibit high similarities, as presented in Fig. 3 (f).

In order to be able to handle these situations, we assign the corresponding pixel in Camera 2 to be  $\mathbf{q}$  using the least median of squares (LMS) algorithm. As opposed to least-squares (LS), LMS is robust to as much as 50% outliers in the data [8].

#### D. Matching Results

Here we present several outdoor examples, where two cameras observe a scene. The cameras have different orientations with respect to the observed scene as well as different zoom levels. Unlike the standard stereo-matching problems, presenting a visual disparity map here can hardly provide the reader with an intuitive understanding of the mapping function. Hence, in order to present the results of this section, we picked several pixels in one camera and drew arrows to their corresponding pixels in the other camera. The results are presented in Fig. 4. Quantitative evaluation of our method will be presented in the journal paper under preparation.

#### E. Occlusion Estimation with Left-Right Check

Note that using the proposed method we can also generate occlusion maps, *i.e.* maps which differentiates regions seen by every camera from those seen by a subset of cameras. This can be done by simply using a left-right check as described in [9] once the matching is performed. Below we present an occlusion map where we identify three regions in the images: the blue colored regions are present in both cameras, the red colored regions exist in only one camera and not the other, and

the green colored regions are the no-motion regions. Fig. 5 presents the results of this segmentation.

### III. STRUCTURE FROM MATCHES

Unsupervised calibration techniques aim at estimating, up to a projective transformation, the structure of the system based on the two dimensional position  $\mathbf{p}_i$  in each camera of a set of  $N$  three dimensional points  $\mathbf{P}_i$ . In this section, we show how these points  $\mathbf{p}$  can be selected based on the multicamera matching technique and the left-right check. We also show how these points are used to estimate the homography matrix, the essential and fundamental matrices as well as the extrinsic parameters. We will start with the two-camera case, and will later generalize to the  $K$ -camera case.

#### Selection of $N$ pairs of points

Here the goal is to find a set of  $N$  pairs of points  $\Gamma = \{(\mathbf{p}_1 \leftrightarrow \mathbf{q}_1), (\mathbf{p}_2 \leftrightarrow \mathbf{q}_2), \dots, (\mathbf{p}_N \leftrightarrow \mathbf{q}_N)\}$  in two cameras with our matching procedure. Since the matching function (Eq.(8)) relies on the co-occurrence of activity,  $\mathbf{p}_i$  and  $\mathbf{q}_i$  can only be selected with non-zero activity. We thus randomly select in Camera 1 pixels  $\mathbf{p}_j$  in areas with non-zero activity. Then, following Algo. 1, the best match  $\mathbf{q}_j$  in Camera 2 is estimated. A left-right check is then performed to validate that  $(\mathbf{p}_j \leftrightarrow \mathbf{q}_j)$  is a true match, *i.e.* that  $\mathbf{p}_j$  and  $\mathbf{q}_j$  are not occluded (see Algo. 2). If the test fails then the pair  $(\mathbf{p}_j \leftrightarrow \mathbf{q}_j)$  is rejected. This procedure is repeated until  $N$  pairs of points  $(\mathbf{p}_j \leftrightarrow \mathbf{q}_j)$  are obtained. While ideally one would use all the matching points, in this work we use only a subset in order to speed up the processing time. We present the effect of this in Section IV.

#### Homography

Given a set of  $N \geq 4$  pairs of points  $\Gamma$ , the goal is to estimate a homography matrix  $H$  which relates each pixel  $\mathbf{p}_j$  in Camera 1 to its associated pixel  $\mathbf{q}_j$  in Camera 2. Note that such global relation is only valid for planar scenes and cannot deal with arbitrary parallax effects.

---

#### Algorithm 2 Left right check

---

**Input:**  $V_1, V_2, \mathbf{p}, \epsilon$

**Output:** Decision

- 1:  $\mathbf{q} = \text{best match from algo.1 } (V_1, V_2, \mathbf{p})$
  - 2:  $\mathbf{p}' = \text{best match from algo.1 } (V_2, V_1, \mathbf{q})$
  - 3: **if**  $\|\mathbf{p} - \mathbf{p}'\| > \epsilon$  **then**
  - 4:     **return**  $\mathbf{q}$
  - 5: **else**
  - 6:     **return** NULL
  - 7: **end if**
- 

However, since in this paper we consider surveillance cameras looking down at cars and pedestrians moving on streets, highways, and sidewalks, the planar assumption holds true for most scenes we deal with. The homography matching is usually expressed as  $\mathbf{q}_j = H\mathbf{p}_j$  where  $H$  is a  $3 \times 3$  matrix and  $\mathbf{p}_j = (p_{xj}, p_{yj}, 1)^T$  and  $\mathbf{q}_j = (q_{xj}, q_{yj}, 1)^T$ . Without lost

of generality, this equation can be expressed as a cross product :  $\mathbf{q}_j \times H\mathbf{p}_j = 0$  or, equivalently, by the following equation

$$\begin{pmatrix} \mathbf{0} & -\mathbf{p}_j^\top & q_{yj}\mathbf{p}_j^\top \\ \mathbf{p}_j^\top & \mathbf{0} & -q_{xj}\mathbf{p}_j^\top \\ -q_{yj}\mathbf{p}_j^\top & q_{xj}\mathbf{p}_j^\top & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{h}^{1^\top} \\ \mathbf{h}^{2^\top} \\ \mathbf{h}^{3^\top} \end{pmatrix} = 0$$

where  $\mathbf{0} = (0, 0, 0)$  and  $\mathbf{h}^i$  is the  $i^{\text{th}}$  row of  $H$  expressed as a line vector. Since the left-hand side  $3 \times 9$  matrix has rank 2, we can write

$$\begin{pmatrix} \mathbf{0} & -\mathbf{p}_j^\top & q_{yj}\mathbf{p}_j^\top \\ \mathbf{p}_j^\top & \mathbf{0} & -q_{xj}\mathbf{p}_j^\top \end{pmatrix} \begin{pmatrix} \mathbf{h}^{1^\top} \\ \mathbf{h}^{2^\top} \\ \mathbf{h}^{3^\top} \end{pmatrix} = 0$$

$$A_j h = 0, \quad (9)$$

where  $A_j$  is a  $2 \times 9$  matrix associated to the  $(\mathbf{p}_j \leftrightarrow \mathbf{q}_j)$  pair. Since each matrix  $A_j$  can be bundled together into a single  $2N \times 9$  matrix  $A$ , the solution  $h$  of this system is obtained with a Singular Value Decomposition (SVD) of  $A$ . Please refer to Hartley and Zisserman's book for more details on this procedure [2].

It is well documented that such method suffers from numerical instabilities due to the fact that  $p_{xj} \gg 1, p_{yj} \gg 1, q_{xj} \gg 1$  and  $q_{yj} \gg 1$ [1], [2]. Consequently, we translate and normalize the two dimensional coordinates of each point  $\mathbf{p}_j$  and  $\mathbf{q}_j$  such that their average Euclidean distance to the origin is  $\sqrt{2}$ . This normalization is also performed for the fundamental matrix and in the  $K$ -camera case.



Fig. 5. From left to right : image from Camera 1, image from Camera 2 and the occlusion map obtained with our method. Green signifies no activity, red signifies region of activity as seen by Camera 1 only and blue signifies regions of activity seen by both cameras.

### Fundamental Matrix

The fundamental matrix is used to relate a two dimensional point in a camera to its epipolar line in the other camera. Mathematically, this boils down to :  $\mathbf{q}_j^\top F \mathbf{p}_j = 0$ , where  $F$  is a  $3 \times 3$  matrix and  $\mathbf{p}_j, \mathbf{q}_j$  are two dimensional points with homogeneous coordinates. By simply reorganizing the terms, one can see that

$$\begin{pmatrix} q_{xj}p_{xj}, q_{xj}p_{yj}, q_{xj}, q_{yj}p_{xj}, q_{yj}p_{yj}, q_{yj}, p_{xj}, p_{yj}, 1 \end{pmatrix} \vec{f} = 0$$

$$B_j \vec{f} = 0$$

where  $f$  is a vector made of the entries of  $F$  in row-major order. Since we have  $N$  pairs of points and that each is associated to a  $B_j$  vector, these vectors can be stacked together to get an equation of the form :  $B\vec{f} = 0$ , where  $B$  is a  $N \times 9$  matrix. In an ideal noiseless situation, only  $N = 8$

pairs of points are needed to estimate  $F$  up to a scaling factor. However, usually there is not a unique solution and thus, one needs to find  $F$  with a least-squares solution. Similarly to the homography case, we resort to an SVD of  $B$  to estimate  $F$ .

### Essential Matrix

The essential matrix is very similar to the fundamental matrix as it relates a point  $\hat{\mathbf{p}}_j$  in one camera to a line in the other camera following the equation :  $\hat{\mathbf{q}}_j^\top E \hat{\mathbf{p}}_j = 0$ . The only difference being that  $\hat{\mathbf{p}}_j$  and  $\hat{\mathbf{q}}_j$  are expressed in *camera coordinates*. By the very nature of  $E$  and  $F$ , one can show that  $E = K_1^\top F K_2$  where  $K_i$  is the calibration matrix of camera  $C_i$  [1]

$$K_i = \begin{pmatrix} \alpha_x f_i & s_i & x_0 \\ 0 & \alpha_y f_i & y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (10)$$

where  $s_i$  is the skew parameter,  $f_i$  the focal length,  $\alpha_y/\alpha_x$  the aspect ratio of each pixel, and  $(x_0, y_0)$  the coordinates of the optical axis. In this paper, we assume that both cameras have the same settings ( $K_1 = K_2$ ) and that  $s = 0$ .

Due to the specific nature of  $E$ , one can show that it has rank 2 with two equal non-zero eigenvalues. Thus, as mentioned by Whitehead and Roth [10], the goal is to find a calibration matrix  $K$  (i.e. a value for " $\alpha_x f$ " and " $\alpha_y f$ ") such that  $E = K^\top F K$  has two eigenvalues  $\sigma_1, \sigma_2$  as close as possible. One cost function which encapsulates this constraint is  $1 - \frac{\sigma_2}{\sigma_1}$  which we minimize with a downhill simplex method. For more details (and code) on the simplex optimizer, please refer to [11].

### Extrinsic parameters

The extrinsic parameters are the rotation and translation matrices which defines the transformation between the world coordinate frame to the camera coordinate frame. Those parameters are contained into a  $3 \times 4$  matrix  $M = [R|T]$  where  $R$  is a  $3 \times 3$  rotation matrix and  $T$  is a three dimensional column vector. It has long been established [1], [2], [12] that since  $E = U \text{diag}(\sigma_1, \sigma_2, 0) V^\top$  (with  $\sigma_1 = \sigma_2$ ) following an SVD, if Camera 1 is located at the origin looking towards the Z-Axis ( $M_1 = [I|0]$ ), then there are four possible choices for the extrinsic parameters of the second camera:

$$M_2 = [UWV^\top | \mathbf{u}_3] \text{ or } [UWV^\top | -\mathbf{u}_3]$$

$$\text{or } [UW^\top V^\top | \mathbf{u}_3] \text{ or } [UW^\top V^\top | -\mathbf{u}_3] \quad (11)$$

where  $\mathbf{u}_3$  is the third column of  $U$  and  $W$  is a skew-symmetric matrix

$$\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (12)$$

Consequently, the transformation  $M_2$  between 2 cameras can be directly obtained based on the essential matrix. Note that the best choice among the four being the one for which a three dimensional point in front of  $C_1$  also falls in front of  $C_2$ .

### Dealing with outliers

Although the left-right check procedure is well geared to locate good pairs of points, it can nevertheless return outliers (*i.e.* erroneously matched pairs of points  $(\mathbf{p}_j \leftrightarrow \mathbf{q}_j)$  who show similar behavior). Actually, based on our experiments, up to 10% of those pair of points returned by the left-right check are outliers. In order to reduce the effect of these outliers, the homography and fundamental matrices are estimated with Ransac [13] which not only returns a more precisely estimated matrix, but also separates the inliers from outliers. Those inliers are then used to estimate the projection matrices in the K-Camera case.

### K-Camera case

Here we aim at estimating the structure of a  $K$  camera system where  $K > 2$ . Similar to the 2-camera case, we assume no camera calibration or additional three dimensional information, and recover the structure up to an overall projective transformation of the three dimensional space. Assuming that all cameras look at a scene containing enough activity to allow a multi camera matching, the left-right check procedure can be easily extended to find a set of  $N$  matching pixels in each camera  $C_i : \{\mathbf{p}_{ij} | \forall j \in [1, N]\}$ . Then, in order to eliminate outliers, the homography matrix between each pair of camera is estimated using Ransac.

Given the fact that  $\mathbf{p}_{ij} = M_i \mathbf{P}_j$  or, without loss of generality,  $\lambda_{ij} \mathbf{p}_{ij} = M_i \mathbf{P}_j$  where  $\lambda_{ij}$  is a constant<sup>1</sup>, one can easily show that [14]

$$W = \begin{pmatrix} \lambda_{11} \mathbf{p}_{11} & \dots & \lambda_{1n} \mathbf{p}_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{K1} \mathbf{p}_{K1} & \dots & \lambda_{Kn} \mathbf{p}_{Kn} \end{pmatrix} = \begin{pmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_K \end{pmatrix} (\mathbf{P}_1 \dots \mathbf{P}_n).$$

As mentioned in [14], [2] the left-hand side  $3K \times n$  rescaled measurement matrix  $W$  has rank at most 4 when the projective depths  $\lambda_{ij}$  are correctly set (hence these depths are being considered).

The projection matrices  $M_i$  as well as the three dimensional points  $\mathbf{P}_j$  are estimated following a six-step procedure similar to the one proposed in [2], [14]. (1) To minimize the effect of noise, we first translate and normalize the  $K \times n$  two dimensional points  $\mathbf{p}_{ij}$  such that the average Euclidean distance to the origin is  $\sqrt{2}$ . (2) Initialize each depth  $\lambda_{ij}$  following Eq.(3) in [14]. (3) The depths  $\lambda_{ij}$  are normalized following the 2-pass procedure suggested by Hartley and Zisserman [2], *i.e.*

$$\lambda'_{ij} = \frac{\lambda_{ij}}{\sqrt{\sum_I \lambda_{Ij}^2}}, \quad \lambda_{ij} = \frac{\lambda'_{ij}}{\sqrt{\sum_J \lambda_{ij}^2}}.$$

(4) Matrix  $W$  is built following an SVD procedure :  $W = UDV^T$  where  $D$  is a diagonal matrix made of singular values. Since  $W$  is of rank 4, all but the first 4 entries of  $D$  are set to zero :  $\hat{D} = \text{diag}(D(1,1), D(2,2), D(3,3), D(4,4), 0, 0, \dots)$ .

<sup>1</sup>Since  $\mathbf{p}_{ij}$  is in homogeneous coordinates,  $(x_{ij}, y_{ij}, 1) = (\lambda_{ij} x_{ij}, \lambda_{ij} y_{ij}, \lambda_{ij})$

(5) Based on  $\hat{D}$ , a new matrix  $\hat{W} = U\hat{D}V^T$  is built. (6) The camera matrices as well as the three dimensional points are then computed

$$(M_1^T, M_2^T, \dots, M_K^T) = U\hat{D}$$

$$(\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n) = V^T$$

(7) With the newly computed  $M_i$  and  $\mathbf{P}_j$ , re-estimate the  $\lambda_{ij}$  values [2]. Repeat step (3)-(7) up until every point  $\mathbf{P}_j$  stabilizes.

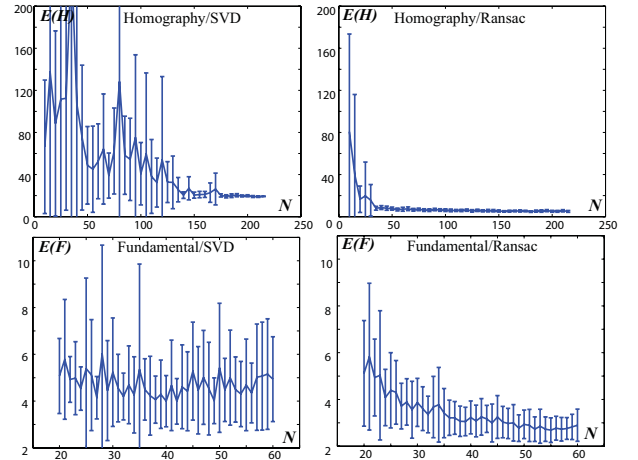


Fig. 6. Robustness of our method with (and without) Ransac with respect to the number of pairs of points  $N$  used to estimate the homography and fundamental matrices.

## IV. EXPERIMENTAL RESULTS

### Performance evaluation

Here we measure the accuracy of the homography and fundamental matrices estimated by our method. The objective is two-fold. First, since our method relies on its ability of accurately detect motion (*c.f.* Eq.(8)) we wish to evaluate how robust to noisy motion masks our method is. Second, since our method depends on a set of  $N$  pairs of points, we examine how this parameter affects the results. To do so, we estimate the ground truth homography matrix  $H_{gt}$  for a sequence (the one in Fig. 9 (d)) by carefully hand selecting 20 pairs of points. Once  $H_{gt}$  is known, a residual error function for  $H$  and  $F$  is implemented

$$E(H) = \frac{1}{2N} \sum_{\mathbf{p}} (\|H\mathbf{p} - H_{gt}\mathbf{p}\| + \|\mathbf{q}H - \mathbf{q}H_{gt}\|)$$

$$E(F) = \frac{1}{2N} \sum_{\mathbf{p}} (\text{dist}(F\mathbf{p}, \mathbf{q}) + \text{dist}(\mathbf{q}F^T, \mathbf{q}))$$

where  $N$  is the total number of pixels,  $F\mathbf{p}/\mathbf{q}F^T$  are epipolar lines,  $\mathbf{q}$  is the matching point of  $\mathbf{p}$  ( $\mathbf{q} = H_{gt}\mathbf{p}$ ), and  $\text{dist}(\cdot)$  is the point-line distance in pixels.

In Fig. 6, error curves have been obtained with and without Ransac. For each value  $N$  on the X-Axis, we randomly selected  $N$  pairs of points (following the method in Section

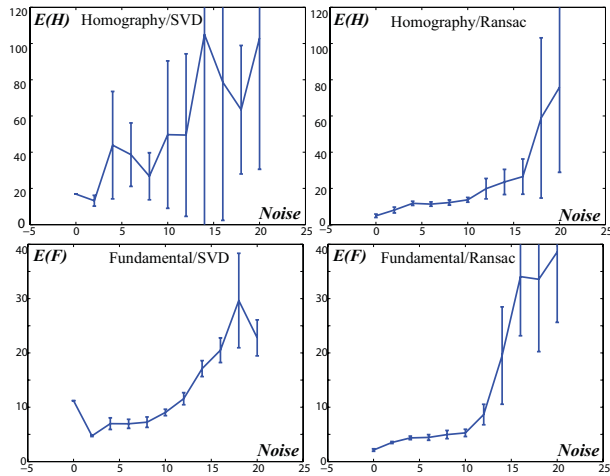


Fig. 7. Robustness of our method with (and without) Ransac with respect to noise.

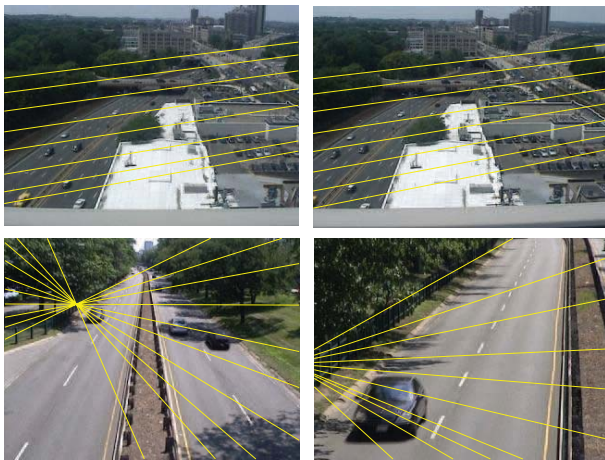


Fig. 8. Epipoles and epipolar lines for two scenes.

III) based on which  $F$  and  $H$  are estimated. This procedure is repeated 10 times to get an average error and a variance for each  $N$  value between 10 and 200. As can be seen, the results are clearly in favor of Ransac which adds robustness to the basic SVD solutions. Also, these curves underline the fact that more than  $N = 50$  pairs of points do not significantly improve the results as the average curves plateau around an error of 1 to 3 pixels.

Fig. 7 shows the effect of noise on our results. Here, a percentage of noise (between 0% and 20%) is added to each binary motion masks  $V_1$  and  $V_2$  used by our matching function (Eq. (8)). For each noise value, a number of  $N = 100$  pairs of points are randomly obtained, based on which  $F$  and  $H$  are estimated. This procedure is repeated 10 times to get an average error and a variance for each noise value. Again, the results are clearly in favor of Ransac, especially with a low noise level.

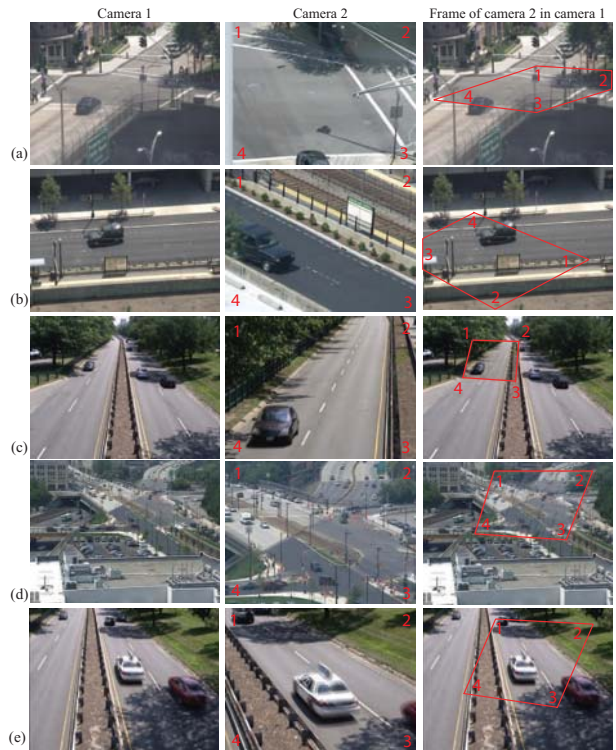


Fig. 9. Homography transformation for five different scenes.

### Qualitative results

In Fig. 9, the homography of five different pairs of videos are presented. In order to illustrate the homographic transformation from Camera 1 to Camera 2, a red box illustrating the frame of Camera 1 projected onto Camera 2 is placed over the image of Camera 2. The numbers from 1 to 4 are used to identify each corner of the box. This is especially useful in example (b) in which cameras are placed on opposite sides of the boulevard and thus the transformation results into a flip of the corners. Notice that in (a), (b) and (d), traditional feature matching methods (such as the ones based on SIFT or on corner detection methods) would fail due to the significant difference in the position/orientation of each camera.

In order to illustrate the fundamental matrix, we took two pairs of videos on which we put epipolar lines  $F\mathbf{p}$  and  $\mathbf{q}F^T$ . Those are presented in Fig. 8. In the first example, both cameras are located on the ninth floor of a building looking down at the scene. Since their optical axis are parallel, the epipoles are far from the image center and their epipolar lines are, for all practical purposes, parallel. In the second example, the cameras have different position and orientation and, as can be seen, by varying  $\mathbf{p}$  and  $\mathbf{q}$ , the epipolar lines meet at a common epipole located on the left of the car.

In Fig. 10, the three dimensional position and orientation of a system made of two and four cameras are presented. The four cameras are located on the fifth floor of a building, and the height effect is successfully recovered in the three dimensional

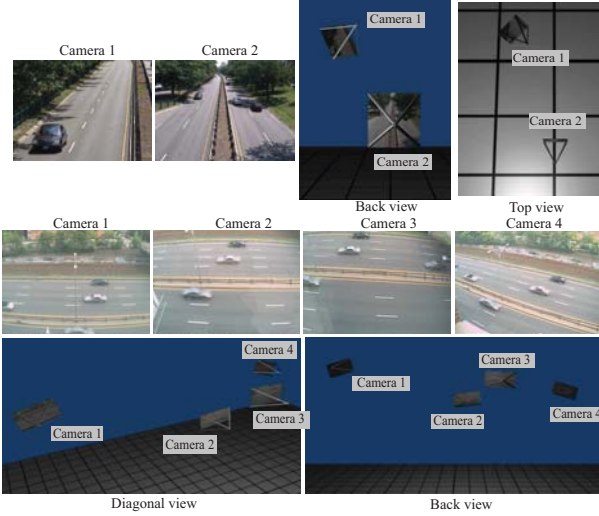


Fig. 10. Structure estimated from a 2-view and a 4-view system. The bottom four sequences were obtained with cameras located on the fifth floor of a building.

reconstruction of the camera locations. Note that error plots showing relationship between the number of cameras and performance are yet to be obtained on synthetic 3D scenes with ground truth values.

## V. CONCLUSIONS

In this paper, we presented an activity-based multicamera matching procedure which allows to find for any point  $\mathbf{p}$  in Camera 1 its corresponding point  $\mathbf{q}$  in Camera 2. As shown in Section 2, to find a good match, our method needs objects with a fairly small aspect ratio and cameras located above the moving objects. However, we showed that our method is effective on various traffic scenes with cameras having different elevation angles. Also, since not every point  $\mathbf{p}$  are in areas common to both cameras, a left-right check procedure is implemented to help find non-occluded pixels. We showed that based on those matches, one can reliably estimate the structure (up to a perspective projection) of the system which is the fundamental matrix, the homography matrix, the essential matrix, and the physical configuration of the cameras with respect to each other. We showed that this could be done for a two-camera case as well as for a  $K$ -camera case. Also, since the left-right check procedure may sometimes return outliers (that is, a bad  $(\mathbf{p} \leftrightarrow \mathbf{q})$  match whose points  $\mathbf{p}$ ,  $\mathbf{q}$  nonetheless show similar behavior), Ransac was used to estimate the fundamental matrix and the homography matrix. Experiments show that our method is fairly robust to noise and that good results can be obtained with only 50 pair of points.

## ACKNOWLEDGMENT

This research was supported by the Presidential Early Career Award (PECASE) N00014-02-100362, NSF CAREER award ECS 0449194, the Department of Homeland Security, ALERT Program, and the NSERC Discovery Grant 371951.

## REFERENCES

- [1] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, 1998.
- [2] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [3] D. Devarajan, Z. Cheng, and R.J. Radke, "Calibrating distributed camera networks," *Proceedings of the IEEE (Special Issue on Distributed Smart Cameras)*, vol. 96, no. 10, pp. 1625–1639, 2008.
- [4] E. B. Ermis, V. Saligrama, P-M Jodoin, and J. Konrad, "Abnormal behavior detection and behavior matching for networked cameras," in *proc. of ICDCS 2008*, pp. 1–10, Sept. 2008.
- [5] D. Zhang and G. Lu, "Segmentation of moving objects in image sequence: A review," *CSSP*, vol. 20, no. 2, pp. 143–183, 2001.
- [6] A. Elgammal, R. Duraiswami, D. Harwood, and L.S. Davis, "Background and foreground modeling using nonparametric kernel density for visual surveillance," *Proc of the IEEE*, vol. 90, no. 7, pp. 1151–1163, 2002.
- [7] J. Konrad, "Motion detection and estimation," in *Handbook of Image and Video Processing, 2nd Edition*, A. Bovik, Ed., chapter 3.10, pp. 253–274. Academic Press, 2005.
- [8] Rousseeuw P. and Leroy A., *Robust Regression and Outlier Detection*, Probability and Mathematical Statistics. John Wiley & Son, 1987.
- [9] G. Egnal and R. P. Wildes, "Detecting binocular half-occlusions: Empirical comparisons of five approaches," *IEEE PAMI*, vol. 24, no. 8, pp. 1127–1133, 2002.
- [10] A. Whitehead and G. Roth, "Estimating intrinsic camera parameters from the fundamental matrix using an evolutionary approach," *EURASIP J. Appl. Signal Process*, vol. 8, pp. 1113–1124, 2004.
- [11] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, Cambridge University Press, New York, NY, USA, 2007.
- [12] R. Hartley, "Estimation of relative camera positions for uncalibrated cameras," in *in proc. of ECCV*, 1992, pp. 579–587.
- [13] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [14] P. Sturm and B. Triggs, "A factorization based algorithm for multi-image projective structure and motion," in *procs. of ECCV*, 1996, pp. 709–720.