Activity Based Matching in Distributed Camera Networks

Erhan Baki Ermis, Pierre Clarot, *Student Member, IEEE*, Pierre-Marc Jodoin, *Member, IEEE*, and Venkatesh Saligrama, *Senior Member, IEEE*

***Abstract*—In this paper, we consider the problem of finding correspondences between distributed cameras that have partially overlapping field of views. When multiple cameras with adaptable orientations and zooms are deployed, as in many wide area surveillance applications, identifying correspondence between different activities becomes a fundamental issue. We propose a correspondence method based upon activity features that, unlike photometric features, have certain geometry independence properties. The proposed method is *robust* to pose, illumination and geometric effects, *unsupervised* (does not require any calibration objects). In addition, these features are amenable to low communication bandwidth and distributed network applications. We present quantitative and qualitative results with synthetic and real life examples, and compare the proposed method with scale invariant feature transform (SIFT) based method. We show that our method significantly outperforms the SIFT method when cameras have significantly different orientations. We then describe extensions of our method in a number of directions including topology reconstruction, camera calibration, and distributed anomaly detection.**

***Index Terms*—Activity pattern matching, anomaly detection, compressive sensing, distributed video processing, multicamera networks.**

I. Introduction

THIS paper is motivated by a problem that arises in heterogeneous networks that are becoming ubiquitous in urban surveillance scenarios. The purpose of these networks is to collect information from a region of interest, fuse the collected information, and perform a number of high level tasks based upon the fused information, e.g., multicamera anomaly detection, multicamera tracking, behavior modeling, 3-D scene reconstruction, etc. A fundamental issue that arises in this context is the problem of efficiently identifying correspondences between the different fields-of-view in the multicamera network. This issue is prevalent in heterogeneous networks because the cameras have different locations (wide-baseline scenarios), orientations and zoom with respect to the scene, and the inherent topology of the network is dynamic, e.g., the cameras can change their orientations and zoom levels and, hence, the correspondences may change several times a day. Furthermore, generally these correspondences have to be established over a low bandwidth communication network. Motivated by these reasons in this paper we focus on identifying correspondences between multiple cameras with partially overlapping field of views.

Since the camera parameters and topology are often unknown and time varying in heterogeneous networks, information fusion poses a significant challenge, which in turn makes it difficult to perform high level tasks. However, given a number of pixel-level correspondences across different cameras, one can infer the camera parameters and topology, and significantly reduce the complexity of problems associated with information fusion and high level tasks. Therefore efficient and robust methods are needed to find pixel level correspondences.

In addition to its utility for camera calibration multicamera matching is of independent interest. For instance, consider *abnormal activity detection* where we wish to fuse activity from multiple cameras. Conventionally, multisensor fusion [1] requires a measurement model for the multisensor measurements (a joint distribution of measurements), but this is generally unavailable in a camera setting. As we describe in the upcoming sections, geometry independence property of the proposed activity features provides the necessary surrogate for multicamera fusion for detecting abnormal activity in some interesting cases.

The multicamera matching method developed in this paper is based upon the notion of *geometry independence of activity*. Unlike methods, described in the following section, that rely on matching geometric/photometric features across cameras, our method relies on matching the activity patterns observed at a given location. While matching geometric/photometric features is inherently dependent upon the existence of a favorable camera topology, our method is essentially independent of the camera topology. The overall contribution of our paper is described in the flow chart presented in Fig. 1.

The principle features of our scheme is summarized in the following.

1) Robust: We demonstrate that the proposed method is robust to pose, illumination & geometric effects, and arbitrary orientations and zoom levels (preview, for example, Fig. 18–Fig. 21 where we compare our method with the SIFT method). Unlike some of the

Manuscript received October 06, 2009; revised March 17, 2010; accepted May 20, 2010. Date of current version September 17, 2010. This work was supported in part by the Department of Homeland Security under Award Number 2008-ST-061-ED0001, in part by the Presidential Early Career Award (PECASE) under N00014-02-100362, in part by the NSF Career Award ECS 0449194 and NSF Award CCF-0905541, in part by the NSERC Discovery Grant 371951, and in part by ONR Award N000141010477. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Rama Chellapa.

E. B. Ermis and V. Saligrama are with Boston University, Boston, MA 02215 USA (e-mail: erhanermis@gmail.com; srv@bu.edu).

P. Clarot and P.-M. Jodoin are with the Université de Sherbrooke Sherbrooke, QC J1K 2R1, Canada (e-mail: pierre-marc.jodoin@usherbrooke.ca; pierre.clarot@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2010.2052824

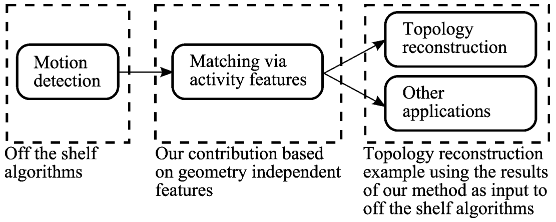

Motion detection
Matching via activity features
Topology reconstruction
Other applications
Off the shelf algorithms
Our contribution based on geometry independent features
Topology reconstruction example using the results of our method as input to off the shelf algorithms

Fig. 1. Flow chart describing the overall contribution of activity matching.

other methods described in the next section, it does not require high level tasks such as object tracking and data association across cameras. It only depends upon sufficiently good motion detection, which is now a necessary ingredient for many video applications [2]–[6].

2) Unsupervised: It is unsupervised in that it does not require the knowledge of camera locations, orientations, epipolar geometry, resolutions, and can be used in heterogeneous or dynamic camera networks. Furthermore, the method does not require the placement of any calibration object into the scene and, hence, can be used in scenes where the system user may not have control over the activities taking place in the surveilled area, e.g., a highway, sidewalk, etc.
3) Communication Efficiency: Our method is particularly well-suited for low bandwidth and distributed situations since correspondence basically involves finding nearest neighbor distances.

In devising the method presented in this paper we make certain assumptions that are listed in the following:

1) motion detected in the cameras is reliable and contains overall a small number of false detections;
2) cameras are synchronized and/or their video contains time stamps to allow the server to realign the time series;
3) the cameras are mounted relatively high above the ground level (analysis presented in the respective section);
4) cameras have overlapping field of views.

Note that we do not develop algorithms for motion detection and camera calibration here. We view motion detection as an input for our method and our method serves as an input to any camera calibration technique.

A. Related Work

Correspondence issues arise in classical stereo matching problems. However, here the cameras are implicitly assumed to have similar orientations and zoom levels. In classical stereo matching problems, most matching algorithms (see [7], [8] and references therein) use stringent assumptions on camera calibration, epipolar geometry, camera placement, and zoom levels [9], and these assumptions cannot be satisfied in heterogeneous or dynamic camera networks.

Although the correspondences can be hand selected, such a procedure is hardly conceivable as the number of cameras increases or when the camera configuration changes frequently, as in a network of pan-tilt-zoom cameras. Other methods for finding correspondences across cameras [10] have been developed through a feature detection method such as the Harris corner detection method [11] or scale invariant feature transform (SIFT) [12]. Such color-based matching methods have also been used to track moving objects across cameras [13], [14]. However, it is well known that under severe illumination variations across cameras due to atmospheric degradation, different camera settings, or when the viewing angle between cameras is too large, such feature-based matching procedures fail. In order to answer these limitations, researchers have investigated other matching methods. In [15], the authors aim at recovering the convex hull of a 3-D object through a shape from silhouette procedure. In order to calibrate the cameras, the method matches points on the objects' surface which project to points on the silhouette in two (or more) views. In [16], the authors developed a method in which the user is required to wave a laser pointer throughout the working volume. As the laser is moved, each camera films the scene and, assuming time-synchronization, the laser point in each image is matched across cameras. These methods, although valuable for the scenarios in which they were conceived, are not practical for the heterogeneous network setup that we consider. In particular, they cannot be applied effectively in large and cluttered urban environments.

In [17], Lee *et al.* describe a matching procedure wherein motion trajectories of objects tracked in different cameras are matched so that the overall ground plane can be aligned across cameras following a homography transformation. A similar approach has been proposed by Khan and Shah [18], Wang *et al.* [19], and Makris *et al.* [20] in which again motion tracks are matched together. However, although [17]–[20] use scene dynamics to find matches, unlike our method, these methods first need to solve the problems of single camera tracking and data association across cameras, which is difficult in highly cluttered scenes or when moving objects occlude each other.

Our results indicate that in urban surveillance scenarios with heterogeneous networks the proposed method can perform pixel level matching with small errors. We demonstrate quantitative and qualitative results, and the conclusion we draw from these results is that when the camera orientations with respect to the scene are nearly parallel, the proposed method and SIFT method perform comparably. In contrast, when the camera orientations are significantly different, the proposed method still performs well while the SIFT method fails to produce satisfactory results.

The paper is organized as follows: in Section II we develop the concept of geometry independence of activity. Next, in Section III we present matching algorithm for heterogeneous networks. We then describe how matching can be applied to reconstruct network topology. We present our experiments and results in Section IV. Section V describes approaches for bandwidth efficient correspondence. Section VI presents a multicamera abnormal activity detection method.

II. Geometry Independence of Activity

In this section, we present the core ideas of this paper. We propose a novel feature that is based upon the activity observed in the video footage. Unlike features that describe the physical properties of a scene (e.g., size, direction, velocity, orientation, etc.) the activity feature has geometry independence properties, i.e., under certain conditions a particular region generates the

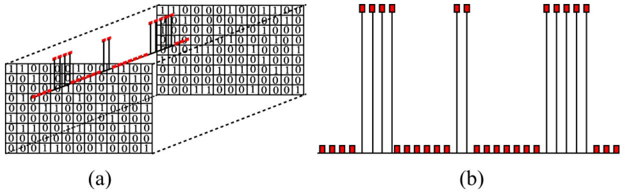


Fig. 2. Binary time series: the proposed activity features for each pixel. (a) Binary motion video; (b) time series for pixel $\mathbf{p}_i$.

same activity feature across a network of cameras irrespective of their locations, orientations, and zoom levels. In the following chapters, we demonstrate the utility of the proposed feature for information processing in distributed camera networks by presenting its applications to the problems of multicamera correspondence and multicamera anomaly detection.

Throughout this paper we assume that we are working with a binary motion video. Researchers have investigated a number of methods to perform motion detection. While advanced methods will improve the performance of the proposed methods, in this work we use temporal median filtering based background subtraction in our experiments, a commonly used method [21]–[23] to obtain the binary motion video.

A. Activity Features

Briefly, the proposed activity features are the occupancy durations of pixels by foreground objects, which are obtained by motion detection. Once the binary motion video is obtained through temporal median filtering based background subtraction, a pixel $\mathbf{p}_i$ has a time series of ones and zeros, as depicted in Fig. 2(b). We use the time series as activity features for a given location. Precisely, for pixel $\mathbf{p}_i$ in a binary video of length T, we use the following sequence as the activity features:

$$V(i,\cdot) = (V(i,1), V(i,2), \ldots, V(i,T)) \tag{1}$$

where $V(i,\tau) \in \{0,1\}$ for $\tau = 1,2,\ldots,T$. These features have a geometry independence property in the sense that under an idealized setup, a location that undergoes some activity generates the same binary time series across cameras irrespective of their locations and zoom levels. This is discussed in more detail in the sequel.

We begin with the geometry independence properties under a simplified scenario where objects are 2-D and live on the plane of motion. We then discuss how the 3-D scenarios can be dealt with.

B. Geometry Independence: Idealized Case

Consider an object moving through a point $\mathbf{x}_0$, which is observed by two cameras of infinite resolution as depicted in Fig. 3. Let $\mathbf{p}_1$ and $\mathbf{q}_1$ be the points corresponding to $\mathbf{x}_0$ in the projection plane of Camera 1 and Camera 2, respectively. Assume that the object moves on a plane and that the moving object is only 2-D, lying on the plane of motion. The cameras can be placed at any direction and distance with the constraint that they must not lie on the plane of motion. Then the object occupies $\mathbf{p}_1$ and $\mathbf{q}_1$ for the same duration. We state this as Lemma 2.1.

Lemma 2.1: Let $\mathbf{x}_0$ be any point on the horizontal plane through which a 2-D object on the plane moves. Denote the location of the two cameras observing $\mathbf{x}_0$ by $\mathbf{c}_1$ and $\mathbf{c}_2$, and the points corresponding to $\mathbf{x}_0$ by $\mathbf{p}_1$ (in the projection plane of Camera 1) and $\mathbf{q}_1$ (in the projection plane of Camera 2). For infinite resolution cameras the object occupies pixels $\mathbf{p}_1$ and $\mathbf{q}_1$ for the same duration.

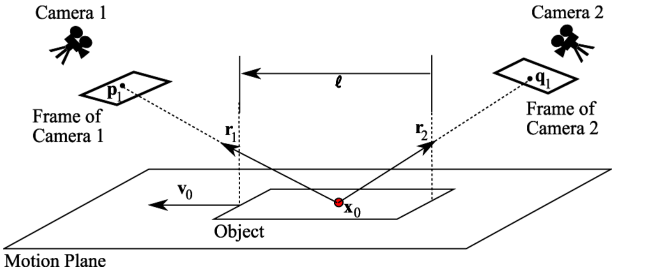


Fig. 3. Setup for geometry independence: the 2-D object moves through $\mathbf{x}_0$ with a velocity $\mathbf{v}_0$. $\mathbf{r}_1$ and $\mathbf{r}_2$ are unit vectors that indicate the direction of the cameras (more precisely, the direction of $\mathbf{p}_1$ and $\mathbf{q}_1$) with respect to $\mathbf{x}_0$. The object has length $|\ell|$. No matter where the cameras are placed, the projection of ℓ and $\mathbf{v}_0$ scale with the same factor for each camera. This cancels the effect of observing from various orientations.

Proof: To prove this result, let $\mathbf{v}_0$ be the vector denoting the velocity of the object. Define ℓ to be a vector that has the same direction as $\mathbf{v}_0$ and whose magnitude equals the magnitude of the sum of the lengths of the line segment of the object that has crossed or that is to cross $\mathbf{x}_0$. Define $\mathbf{r}_1 = \mathbf{c}_1 - \mathbf{x}_0/|\mathbf{c}_1 - \mathbf{x}_0|$, the unit vector that indicates the direction of Camera 1 with respect to the observation point $\mathbf{x}_0$. Similarly define $\mathbf{r}_2 = \mathbf{c}_2 - \mathbf{x}_0/|\mathbf{c}_2 - \mathbf{x}_0|$ for Camera 2. Then, the effective length of the line segment is $|\ell \times \mathbf{r}_1|$ and the effective speed of the object is $|\mathbf{v}_0 \times \mathbf{r}_1|$ with respect to Camera 1. For Camera 2 we get similar expressions where $\mathbf{r}_1$ is replaced by $\mathbf{r}_2$. Now, notice that no matter what $\mathbf{r}_1$ and $\mathbf{r}_2$ are, their scaling effects vanish when calculating the time it takes to go through $\mathbf{p}_1$ and $\mathbf{q}_1$, i.e., $t_1 = |\ell \times \mathbf{r}_1|/|\mathbf{v}_0 \times \mathbf{r}_1| = |\ell|/|\mathbf{v}_0|$ and $t_2 = |\ell \times \mathbf{r}_2|/|\mathbf{v}_0 \times \mathbf{r}_2| = |\ell|/|\mathbf{v}_0|$. Hence, the object occupies $\mathbf{p}_1$ and $\mathbf{q}_1$ for the same duration. ■

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 Corollary 2.2. Intuitively, while the zoom level scales the apparent length of the object with some constant factor c, the velocity is also scaled with c and, hence, duration of occupancy remains the same for all zoom levels.

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

Uniqueness Property: The previous argument only shows that two pixels corresponding to the same location has identical time series. Nevertheless, to find good matches it is important that two pixels corresponding to different locations have different time series. This property follows if we assume that the time series is uncorrelated in space and time. These properties follow from a markov chain model for the time series. Basically, the busy-idle time periods can be described by a two state markov chain at each location, where the states correspond to a busy state and an idle state. This model can be justified from empirical as well as theoretical considerations [24]. Basically the different busy periods at a pixel are independent since they correspond to different objects, while the independence of idle periods corresponds to the fact that it is related to interobject distances. Now the reason for uncorrelatedness in time follows from strong mixing property of markov chains. This implies that even the time series of two pixels that are on the same track of objects are uncorrelated. Two time series corresponding to pixels that are not on different tracks are generally independent since they correspond to tracks of different objects.

1) Extension to Finite Resolution Cameras: This lemma can be extended to cameras of finite resolution with minor modifications. However, before we proceed with this extension, we need to define precisely what we mean by a pixel to be occupied.

Definition 2.3: A pixel is considered to be occupied if at least half of the area of its back projection onto the scene is covered by a moving object. Specifically, let R be the back projection of a pixel $\mathbf{p}$ that has an area $|R|$. Then $\mathbf{p}$ is occupied if $R' \subset R$ is covered by a moving object such that $|R'| \geq |R|/2$.

With this definition of occupancy we now formally state the extension to the finite resolution cameras.

Lemma 2.4: Let $\mathbf{p}_1$ and $\mathbf{q}_1$ be the corresponding pixels in two cameras such that there is a point $\mathbf{x}_0$ on the observation plane that falls into the back projection of the pixels. Let R_1 and R_2 be the back projection of $\mathbf{p}_1$ and $\mathbf{q}_1$, respectively, such that $R_2 \subset R_1$. If O is a rectangular strip, passing through $\mathbf{x}_0$, that is at least as large as half the area of the larger of R_1 and R_2, then $\mathbf{p}_1$ and $\mathbf{q}_1$ are occupied for the same duration by O.

The assumption on the object size with respect to R_1 and R_2, which is reasonable for practical applications, is necessary in order to avoid situations where the object is detected by one camera and missed by another.

Proof: (Follow the exposition through Fig. 4.) Consider the region R_1 that is the back-projection of $\mathbf{p}_1$. There exists a point $x_1 \in R_1$, such that if the rectangular strip touches $\mathbf{x}_1$ it occupies at least half of R_1. $\mathbf{p}_1$ will be occupied as long as the object occludes $\mathbf{x}_1$, and the duration of this occlusion will be $|\ell|/|\mathbf{v}_0|$ where $|\ell|$ is the length of the rectangular strip that crosses $\mathbf{x}_0$. However, a similar analysis will give the same result for pixel $\mathbf{q}_1$. Specifically, $\mathbf{q}_1$ will be occupied as long as the object occludes $\mathbf{x}_2$, the half occupancy point of the back-projection region of $\mathbf{q}_1$ (R_2). The duration of this occlusion is again $|\ell|/|\mathbf{v}_0|$, because a rectangular strip is going through R_1 and R_2, hence, the line-segments that go through $\mathbf{x}_1$ and $\mathbf{x}_2$ have the same length. Therefore, the duration of occupancy will be the same for both $\mathbf{p}_1$ and $\mathbf{q}_1$. ■

Fig. 4 depicts this setup. Note that the figure is drawn deliberately simplistically in order to avoid confusion. In fact, no matter what the shape of the back-projection of each pixel is, as long as they are included in the rectangular strip that goes through $\mathbf{x}_0$, the extension holds true.

2) Examples: We now give some real life examples that demonstrate the geometry independence of activity. We recorded the videos simultaneously from two cameras overlooking a street from the top of a 9-floor building, and performed background subtraction on both sequences. We then looked at the time series data of two pixels ($\mathbf{p}_1$ from Camera 1 and $\mathbf{q}_1$ from Camera 2) each corresponding to the same point on the road (carefully chosen to be so). The results, presented in Fig. 5, demonstrate that the time series can be nearly identical even when we do not have flat objects.

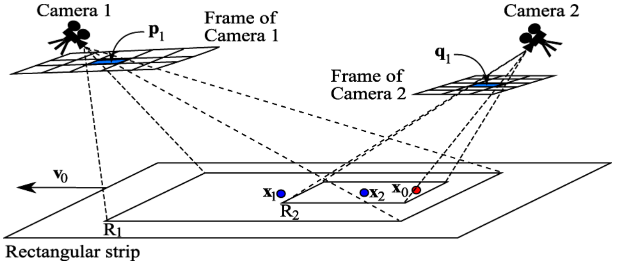


Fig. 4. 2-D setup for extension of geometry independence to finite resolution cameras: Region R_1 is the back-projection of $\mathbf{p}_1$ and region R_2 is the back-projection of $\mathbf{q}_1$. $\mathbf{x}_1$ is a point in R_1 such that if the rectangular strip touches $\mathbf{x}_1$, it occupies at least half of R_1 (similarly for $\mathbf{x}_2$ and R_2). The object is a rectangular strip that moves over the regions, and is larger than half the larger of R_1 and R_2. The object covers the back-projection of $\mathbf{p}_1$ and $\mathbf{q}_1$, and it traverses the half-occupancy points $\mathbf{x}_1$ and $\mathbf{x}_2$ for the same amount of time. This causes $\mathbf{p}_1$ and $\mathbf{q}_1$ to be occupied for the same amount of time by the object.

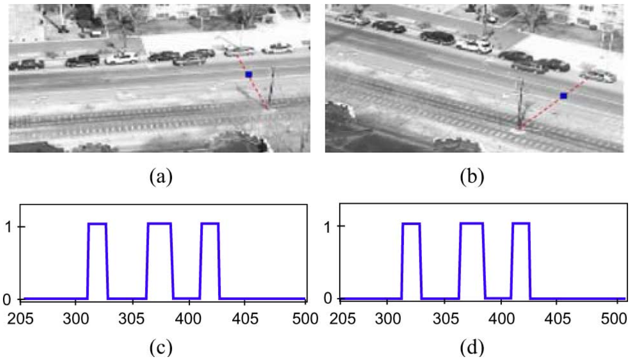
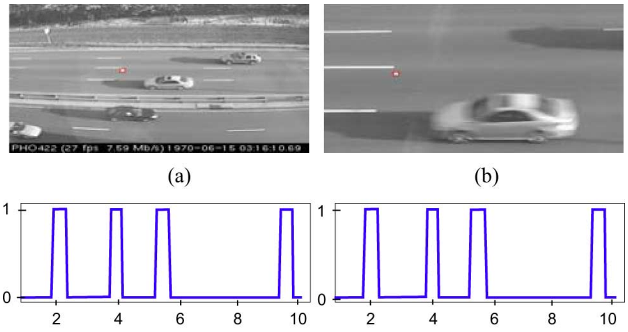


Fig. 5. Geometry independence example on real life videos. In (a) and (b), we selected $\mathbf{p}_1$ from Camera 1 and $\mathbf{q}_1$ from Camera 2 such that they both correspond to the same point on the road. These pixels are shown in blue (on the red dashed lines). In (c) and (d), we present the time series of $\mathbf{p}_1$ and $\mathbf{q}_1$, which are nearly identical despite the fact that objects are not flat. The camera orientations with respect to the scene are also different. (a) $\mathbf{p}_1$ in Camera 1 (blue pixel over the dashed red line); (b) $\mathbf{q}_1$ in Camera 1 (blue pixel over the dashed red line); (c) time series of $\mathbf{p}_1$; and (d) time series of $\mathbf{q}_1$.

We next demonstrate the claim that the time series are invariant to different zoom levels. For this demonstration we used two highway videos, which were obtained simultaneously using two cameras with different zoom levels. We again performed background subtraction and carefully selected two pixels ($\mathbf{p}_1$ from Camera 1 and $\mathbf{q}_1$ from Camera 2) that corresponds to the same point on the road. We then plot the binary time series of these pixels, which are presented in Fig. 6. We observe that the time series generated by a series of cars passing by are the same in both pixels, irrespective of the zoom levels of the cameras.

Remark: Although the geometry independence result is formally developed for a 2-D object, it has strong implications for a certain class of 3-D objects. It asserts that if the third dimension (height) of an object is negligible with respect to the other two dimensions (length and width), then no matter where the cameras are located, the object will generate similar activity patterns in both cameras. Similarly, if the cameras are mounted high above the ground, then the height of objects can be neglected, and the geometry independence principle holds again. We analyze 3-D scenarios in the next subsection and present an analysis of the effect of the height of objects on the geometry independence principle.



(a) (b)

(c) (d)

Fig. 6. Geometry independence example on real life videos. In (a) and (b), we selected $\mathbf{p}_1$ from Camera 1 and $\mathbf{q}_1$ from Camera 2 such that they both correspond to the same point on the road. These pixels are shown in white with red borders. In (c) and (d), we present the time series of $\mathbf{p}_1$ and $\mathbf{q}_1$, which are identical despite the fact that objects are not flat. The cameras have significantly different zoom levels with respect to the scene. (a) $\mathbf{p}_1$ in Camera 1 (white pixel red borders); (b) $\mathbf{p}_1$ in Camera 1 (white pixel red borders); (c) time series of $\mathbf{p}_1$ for 0–10 s, and (d) time series of $\mathbf{q}_1$ for 0–10 s.

C. Geometry Independence: 3-D Scenarios

We will now discuss two important issues that arise when we consider the nonideal case of 3-D objects. The first issue is related to the varying occupancy durations of matching pixels in each camera. In particular, consider $\mathbf{p}_1$ and $\mathbf{q}_1$ observing $\mathbf{x}_0$. In the previous subsections, we showed that in the idealized scenario where the objects are flat, $\mathbf{p}_1$ and $\mathbf{q}_1$ are occupied for the same duration irrespective of the cameras' orientations and zoom levels. In the 3-D cases, the occupancy durations of $\mathbf{p}_1$ and $\mathbf{q}_1$ are not exactly the same.

The second issue is related to the generation of what is called the *spurious activity* at pixels that do not correspond to the actual location of the three dimensional object. In the sequel we present an analysis of these two issues and propose a method to resolve them.

1) Issue 1: Varying Occupancy Durations: Consider a cuboid object that moves over a point $\mathbf{x}_0$ 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 $\mathbf{v}$, and its length in the direction of motion is l. Assume that two infinite resolution cameras observe the object from different views with different zoom levels. Let α be the ratio of the object's height h to its length l, e.g., 1/3 for a car, 1/6 for a truck, and about 6 for people. Let θ_i and ϕ_i be the observation angles defined as in Fig. 7 for Camera i, $i = 1, 2$. In each camera's frame (projection plane) there is a point that corresponds to $\mathbf{x}_0$. Call these points $\mathbf{p}_1$ in Camera 1, and $\mathbf{q}_1$ in Camera 2.

Let $|\mathbf{v}| = v$, and $t = l/v$ be the actual occupancy duration of $\mathbf{x}_0$ by the object. Define t_{p_1} and t_{q_1} as the occupancy duration of pixels $\mathbf{p}_1$ and $\mathbf{q}_1$ 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. This scenario is depicted in Fig. 8. In Fig. 8(a), we present the moment when $\mathbf{p}_1$ being its occupancy and in Fig. 8(b) we present the moment when $\mathbf{p}_1$ finishes its occupancy.

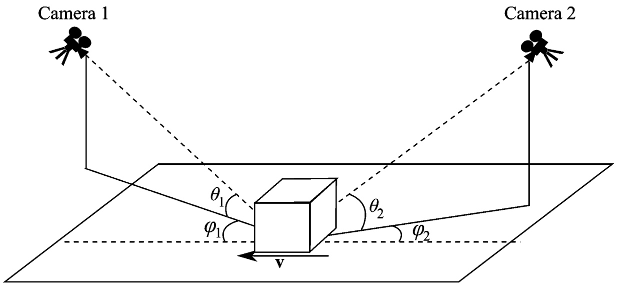
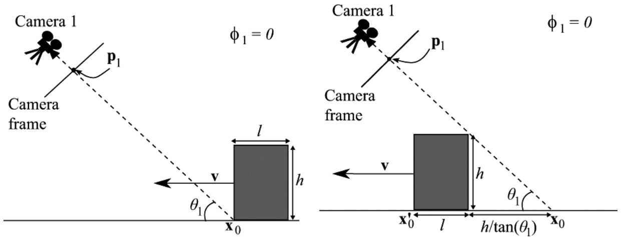


Fig. 7. Multicamera observation angles.



(a) (b)

Fig. 8. Side view for the scenario when Camera 1 is placed at some height along the direction of motion, in front of the object. In this setup $\phi_1 = 0$. $\mathbf{p}_1$ will become occupied when the front end of the object reaches $\mathbf{x}_0$, and it will remain occupied until the front end of the object leaves $\mathbf{x}_0'$, which is precisely when the top right corner of the object leaves the line of sight from $\mathbf{p}_1$ to $\mathbf{x}_0$. Consequently, $\mathbf{p}_1$ will remain occupied for the period of time during which the front end of the object moves from $\mathbf{x}_0$ to $\mathbf{x}_0'$. (a) Moment when $\mathbf{p}_1$ begins occupancy (b) Moment when $\mathbf{p}_1$ ends occupancy.

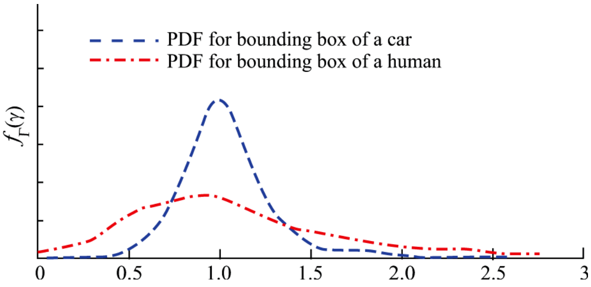


Fig. 9. Probability density functions of $\Gamma = t_{p_1}/t_{q_1}$ for the bounding box of a car and a human. The probability of large deviations from the ratio of 1 decays more rapidly for the cars.

In this setup, $\mathbf{p}_1$ will become occupied when the front end of the object reaches $\mathbf{x}_0$, and it will remain occupied until the front end of the object leaves $\mathbf{x}_0'$, which is precisely when the top right corner of the object leaves the line of sight from $\mathbf{p}_1$ to $\mathbf{x}_0$. Consequently, $\mathbf{p}_1$ will remain occupied for the period of time during which the front end of the object travels the length $l_{C_1} = l + h/\tan(\theta_1)$, where h is the height of the object. Rewriting this in terms of the aspect ratio, we have $l_{C_1} = l + \alpha l/\tan(\theta_1) = l(1 + \alpha/\tan(\theta_1))$. Then the occupancy duration of $\mathbf{p}_1$ will be $t_{p_1} = l_{C_1}/v = l/v(1 + \alpha/\tan(\theta_1)) = t(1 + \alpha/\tan(\theta_1))$, where $t = l/v$ is the actual occupancy duration of $\mathbf{x}_0$ by the object.

Now, as we increase the angle ϕ_1 from zero to 90°, the extension term $(\alpha l/\tan(\theta_1))$ in l_{C_1} begins to shrink. At 90°, it becomes precisely zero, and at 180° 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 [1], and takes a value of 0 at $\phi_i = 90$ and 1 at $\phi_i = 0$ and $\phi_i = 180$. This leads to

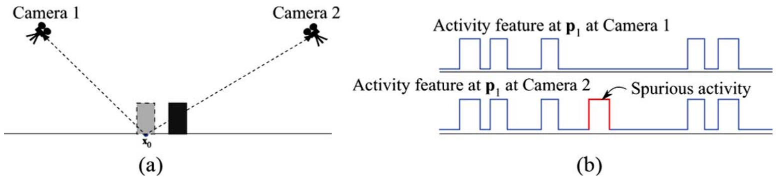

Camera 1
Camera 2
$\mathbf{x}_0$
(a)
Activity feature at $\mathbf{p}_1$ at Camera 1
Activity feature at $\mathbf{p}_1$ at Camera 2
Spurious activity
(b)

Fig. 10. Light object that actually occupies $\mathbf{x}_0$ generates the regular activity at $\mathbf{p}_1$ and $\mathbf{q}_1$, however the darker object that occludes $\mathbf{x}_0$ in the view of Camera 2 without actually going through $\mathbf{x}_0$ generates a spurious activity in $\mathbf{q}_1$. (a) Back view of setup; (b) activity features.

the following approximations: $t_{p_1} = t(1 + \alpha/\tan(\theta_1)\chi(\phi_1))$, $t_{q_1} = t(1 + \alpha/\tan(\theta_2)\chi(\phi_2))$ and the ratio

$$\frac{t_{p_1}}{t_{q_1}} = \frac{\left(1 + \frac{\alpha}{\tan(\theta_1)}\chi(\phi_1)\right)}{\left(1 + \frac{\alpha}{\tan(\theta_2)}\chi(\phi_2)\right)}. \qquad (2)$$

Note that we need only care about how t_{p_1}/t_{q_1} behaves for multicamera correspondence. In the idealized case $t_{p_1}/t_{q_1} = 1$, however according to the previosuly shown analysis in the general 3-D cases t_{p_1}/t_{q_1} deviates from 1. While analytic expressions on the occupancy durations can be obtained for the simple setup where $\phi_i = 0$, obtaining such results for the more general setting where ϕ_i and θ_i take arbitrary values is difficult. Consequently, we will characterize the distribution of their ratio (over the randomness of observation angles) through empirical studies in the following subsection.

Remark: Throughout our discussion in this section we have assumed that the cameras are time synchronized in the sense that they obtain new frames at exactly the same time and they have exactly the same frame rate. Such assumptions have been made in the literature [15], [16], [18].

2) Discrepancy Characterization: We now present a characterization of the discrepancy of occupancy durations across the cameras, i.e., $\Gamma = t_{p_1}/t_{q_1}$. It is difficult to obtain a closed form expression of Γ for the more general setup where $\phi_i > 0, \quad i = 1, 2$ (see (2)). Consequently, in order to characterize the discrepancy between the occupancy durations across cameras for the general setup, we set up an experiment using 3-Ds Max®, where we discretized the possible observation angles $\theta_i \in \Theta = \{15,\ 30, \ldots, 90\}\ ^\circ$ and $\phi_i \in \Phi = \{0,\ 15, \ldots, 90\}^\circ$.

Next we made a cuboid object to simulate the bounding box of a car $(\text{height} = 1 \text{ unit}, \text{ width} = 1.5 \text{ units}, \text{ length} = 3 \text{ units})$, and created an animation where the object moved through a point $\mathbf{x}_0$. Then, we placed two cameras at randomly selected angles $\theta_i \in \Theta$, $\phi_i \in \Phi$ (each value in Θ and Φ had an equal chance of occurring), where the cameras recorded the movement of the cuboid over the point of interest. To relate the occupancy duration of one camera to another we extracted the binary time series of the pixels $\mathbf{p}_1$ and $\mathbf{q}_1$ that observed $\mathbf{x}_0$ and calculated the ratio of the occupancy duration of these pixels $\Gamma = t_{p_1}/t_{q_1}$. We then repeated this experiment 5000 times, each time placing cameras at randomly selected angles, and using the same animation of object motion, obtaining 5000 values for Γ. Finally, using kernel density estimation with Gaussian kernels, we obtained a density estimate of Γ over the randomness in camera placement for a cuboid that represents a car.

We then repeated this experiment with a cuboid object that simulated a human $(\text{height} = 1 \text{ unit}, \text{ width} = 1 \text{ unit}, \text{ length} = 6 \text{ units})$, and obtained a density estimate of Γ over the randomness in camera placement for a cuboid that represents a human. These distributions are presented in Fig. 9. For the object that has the smaller aspect ratio, the distribution of $\Gamma = t_{p_1}/t_{q_1}$ is more concentrated around 1.

These results conforms with our expectation from the theoretical analysis of the distribution of Γ that objects with large aspect ratios generate larger discrepancies with higher probabilities. In the following sections we will describe how this effect can be mitigated.

3) Issue 2: Spurious Activity: To understand the notion of spurious activity consider two cameras that observe a scene from opposite directions, and tall objects that move in the scene (into and out of the page), as depicted in Fig. 10. Since the objects are tall, some (e.g., object on the right) occlude $\mathbf{q}_1$ in Camera 2 but not $\mathbf{p}_1$ in Camera 1. Therefore, although the time series of $\mathbf{p}_1$ and $\mathbf{q}_1$ would have been the same if the objects were flat, due to their heights the time series of $\mathbf{q}_1$ carries an activity that the time series of $\mathbf{p}_1$ does not, and we call this activity *spurious activity*. This is because it is not generated by the object moving through $\mathbf{x}_0$ but rather because the object occludes $\mathbf{x}_0$ due to its height.

4) Aspect Ratio Normalization: While geometry invariance does not hold in general for these cases it follows that with the mild assumptions we can recover this invariance. We assume that the objects in 3-D world are approximately vertical with respect to the ground and that the cameras are vertically oriented with respect to the observation plane. Specifically, we introduce a postprocessing step after background subtraction to replace the foreground (moving) objects with rectangles that occupy the region where the objects meet the ground. This is similar to the approach taken in [17], where the objects are replaced by points at their center of mass.

5) C.6 Remark: While elaborate methods for object identification can be used to place the rectangles at precise locations, we use a simple pseudo-algorithm (Algorithm 1). This algorithm is motivated by the analysis of Section II-C-IV, where we concluded that objects with small aspect ratios lead to small discrepancies between the activity features at different cameras, and by the observation presented previously, where the height of objects were the main source of generation of spurious activity at irrelevant pixels. The effect of Algorithm 1 can be seen as reducing the height of the actual object in the 3-D world, which brings us closer to the idealized scenario, as depicted in Fig. 11.

The constant k in Algorithm 1 can be seen as the parameter that governs by what factor the height is reduced. In our studies we used $k = 5$ when the aspect ratio of objects needed to be reduced. While the parameters k and α' can be chosen by the

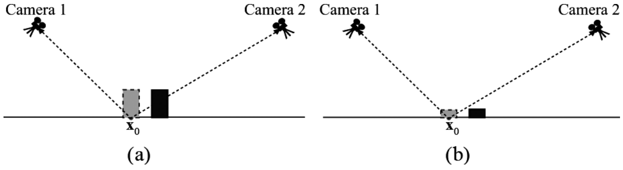


Fig. 11. Aspect ratio normalization can be seen as reducing the height of the actual object in the 3-D world, which brings us closer to the ideal setup. (a) Back view of the original scene; (b) implied back view after aspect ratio normalization.

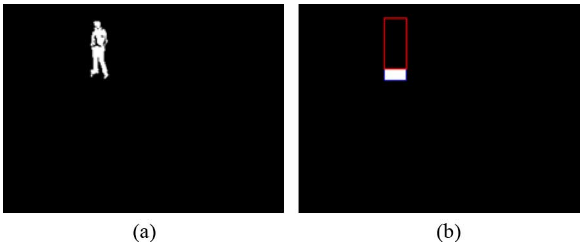


Fig. 12. Aspect ratio normalization example: Here the frame in (a) is used as input to Algorithm 1, and the frame in (b) is obtained as output with a factor $k = 5$. The red box has been manually drawn around the pixels from which spurious activity has been removed. The white box in (b) replaces the object in (a). (a) Binary motion fame after temporal median filter based background subtraction; (b) binary motion fame after Algorithm 1.

user we can fix these parameters *a priori* for each camera based upon the worst-case aspect ratios.

Fig. 12 presents two binary frames from one camera, where in the first frame we present a binary motion video frame obtained through temporal median filter based background subtraction and in the second frame we present the binary motion video frame obtained after the spurious activity has been removed from a set of active pixels using Algorithm 1. To obtain the second frame, the first frame was used as input to Algorithm 1. In the second frame, a red box has been manually drawn around the pixels from which the spurious activity has been removed.

Algorithm 1 Aspect Ratio Normalization

```
Input: Binary video, scaling factor $k$
Output: Binary video (spurious activity removed)
Identify foreground objects
Find the bounding box of each foreground object, let $w$ and $h$ be the width and height of the bounding box, respectively
If Aspect ratio $h/w$ is larger than a threshold $\alpha'$ then
    Replace the foreground objects with a rectangle of dimensions $w \times h/k$, place the rectangle at the bottom of the bounding box.
else
    Keep the object
end If
```

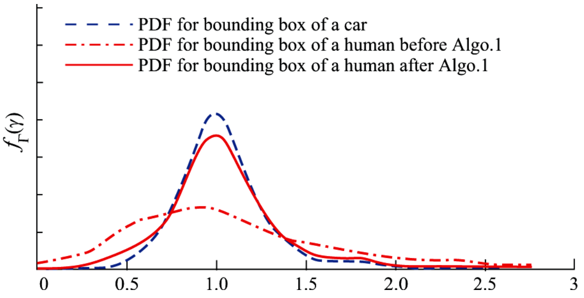


Fig. 13. PDF of $\Gamma = t_{p_1}/t_{q_1}$ before and after aspect ratio normalization. After the normalization the PDF for the bounding box of a human has a clear concentration around 1. This can be interpreted as the objects having a smaller apparent aspect ratio after normalization.

As we presented in the theoretical analysis of the distribution of the scaling parameter Γ as well as demonstrated in Fig. 9, for objects with small aspect ratios the distribution of $\Gamma = t_{p_1}/t_{q_1}$ is concentrated around 1. Consequently, if the effect of Algorithm 1 is indeed equivalent to reducing the apparent aspect ratio of the object, we should observe this in the distribution of Γ as well. To test this hypothesis we repeated the 3-Ds Max® experiment for the bounding box of a human. For this experiment, we took the very same videos as used in the experiment described earlier and obtained binary motion videos through temporal median based background subtraction. We then input these videos to Algorithm 1 and obtained new videos in which the aspect ratio of objects have been corrected with a factor of $k = 5$. Then, we plotted the probability distribution of the ratio Γ with the same method used for the previous 3-Ds Max® experiment. The result is presented in Fig. 13, which demonstrates that with use of Algorithm 1 there is a significant concentration of the density function around 1. This result supports the claim that the apparent aspect ratio of the bounding box of a human is indeed smaller after Algorithm 1.

Finally, in obtaining these figures, we have considered a general setup wherein observation angles down to 15° were allowed. When we consider the urban surveillance scenarios where the cameras are high above the ground and constrain ourself to observation angles that are more than 45°, we observe a significant increase in the concentration of the distribution of $\Gamma = t_{p_1}/t_{q_1}$ around 1. Specifically, more than 90% of the probability mass falls on the interval [0.9, 1, 1] for both the bounding box of a human and the bounding box of a car. This is presented in Fig. 14. The experiment performed to obtain this figure is the same as the one performed for Fig. 9 with the modification that here $\theta_i > 45,\ i = 1, 2$.

Remark: We have observed that, in our multicamera correspondence application, the spurious activity was not a point of concern for scenarios involving vehicles, however it had an effect for scenarios involving pedestrians, such as a sidewalk. In Section IV we present realistic examples and demonstrate how the correspondence results are impacted by this issue and how aspect ratio normalization improves results. From here on we assume that the aspect ratios of objects are small or that Algorithm 1 has been utilized to mitigate the effects of spurious activity as well as the discrepancies in the occupancy durations.

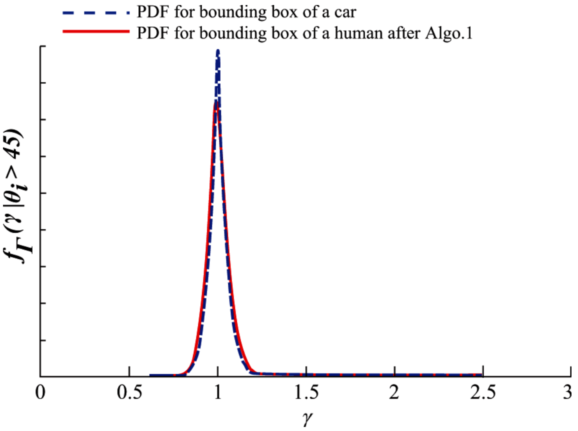


Fig. 14. PDF of $\Gamma = t_{p1}/t_{q1}$ for $\theta_i > 45^o$, $i = 1, 2$. After aspect ratio normalization, the PDFs have a significant increase in their concentration around 1 (more than 90% of the probability mass is in the interval $[0.9,\ 1,\ 1]$ for both distributions). This suggests that beyond elevation angles of 45°, the realistic scenarios are very close to the idealized scenario.

III. Multicamera Correspondence in Heterogeneous Networks

In this section, we present the activity-based matching algorithm. Later, we will present results on topology reconstruction where we use the matching results obtained by our method as an input to the topology reconstruction algorithms. We begin by describing the matching algorithm.

A. *Multicamera Correspondence as a Hypothesis Test*

Let $\mathbf{p}_i$ be a pixel in Camera 1, and $V_1(i,\cdot)$ be its binary time series. Given a pixel $\mathbf{q}_j$ in Camera 2 and its time series $V_2(j,\cdot)$, the correspondence problem can be cast as a hypothesis testing problem. To understand this connection, assume for the moment that $\mathbf{p}_i$ and $\mathbf{q}_j$ observe the same location. Then, by the principle of geometry independence they observe the same binary sequences corrupted with some binary noise. This binary noise arises due to several factors (e.g., errors in motion detection algorithms) and can be modeled as a binary symmetric channel.

This binary symmetric channel formulation leads directly to the following signal level model. Basically, we note from Section II that if $\mathbf{p}_i$ and $\mathbf{q}_j$ observe different locations then their time series are essentially independent, while if they observe the same location their time series is identical modulo few false alarms and misses arising from imperfect background subtraction. This implies that we can view the correspondence problem as a hypothesis testing problem as follows:

$$H = H_0 : V_2(j,\cdot) = V_1(i,\cdot) \oplus n_1(\cdot)$$
$$H = H_1 : V_2(j,\cdot) = n_2(\cdot)$$

where n_1 and n_2 are binary sequences of Bernoulli random variables with probability of success $\alpha_1 < 0.5$ and $\alpha_2 < 0.5$, respectively, and $\oplus$ denotes a bit-by-bit modulo 2 addition, e.g., if $V_1(i,\tau) = 1,\ n_1(\tau) = 1$ then $V_1(i,\tau) \oplus n_1(\tau) = 0$. This problem setup leads to the well known likelihood ratio test, and in particular the minimum probability of error (MPE) rule leads to

$$\frac{\alpha_1^{\|n_1\|_1}(1-\alpha_1)^{T-\|n_1\|_1}}{\alpha_2^{\|V_2(i,\cdot)\|_1}(1-\alpha_2)^{T-\|V_2(i,\cdot)\|_1}} \underset{H_1}{\overset{H_0}{\gtrless}} 1$$

where T denotes the length of the sequences and $\|\cdot\|_1$ denotes the ℓ_1 norm. Taking logarithms and simplifying the expressions we get

$$dist(V_2(j,\cdot), V_1(i,\cdot)) \ln\left(\frac{\alpha_1}{1-\alpha_1}\right)$$
$$-\|V_2(i,\cdot)\|_1 \ln\left(\frac{\alpha_2}{1-\alpha_2}\right) \underset{H_1}{\overset{H_0}{\gtrless}} 0$$

where $dist(\cdot,\cdot)$ denotes the Hamming distance between two binary sequences, which suggests an algorithm that compares the hamming distance normalized by the amount of activity observed at the pixel to a threshold. Now, this is the test obtained when we compare the signature for pixel, p_j at camera j with pixels, q_j from camera 2. On the other hand if we were to compare pixel q_j at camera 2 with pixels p_j at camera 1 we would get comparison of the same Hamming distance with activity at pixel p_j in camera 1. Since for a good match the Hamming distance must satisfy both these inequalities, we get

$$d(\mathbf{p}_i, \mathbf{q}_j) \doteq \frac{1}{\eta}\text{dist}(V_2(j,\cdot), V_1(i,\cdot)) \underset{H_1}{\overset{H_0}{\gtrless}} \mathcal{L} \qquad (3)$$

where, $\mathcal{L}$ is a suitable threshold. Note that the variable η can either be chosen as $\eta = \max(\|V_2(j,\cdot)\|_1, \|V_1(i,\cdot)\|_1)$ corresponding to the maximum activity or as $\eta = \min(\|V_2(j,\cdot)\|_1, \|V_1(i,\cdot)\|_1)$ corresponding to the minimum activity. The first choice is optimistic since it leads to accepting a larger number of pixels as potential matches, while the latter choice is restrictive. In particular if one pixel has lower activity, the distance has to be relatively small for a potential match. The function $d(\mathbf{p}_i,\ \mathbf{q}_j)$ will serve as our similarity metric between two pixels, $\mathbf{p}_i,\ \mathbf{q}_j$ from now on with η equal to the maximum activity from now on. Note that the similarity function is not a metric due to the normalization.

Remark: While for simplicity of exposition we assumed a Bernoulli model here, it is possible to perform similar analysis with Markovian models and get similar results. Such models for the time series in binary videos have been used in [24] to perform abnormal behavior detection.

Observe that the binary channel models' parameters also play a role in correspondence. In particular, when the binary model for n_2 has $\alpha_2 \longrightarrow 1/2$, the sufficient statistic reduces to the distance between the two time series. Similarly, if the channel model under H_0 has a small probability of flipping the bits, i.e., $\alpha_1 \longrightarrow 0$, then the distance becomes the dominant term and even small distances between the time series can lead to the rejection of the null hypothesis (presence of a match). Finally, when there is little activity, then the distance is required to be very small in order for the null hypothesis to be selected.

Remark: Other formulations can be considered, e.g., models where under the null hypothesis $V_1(i,\cdot)$ passes through a

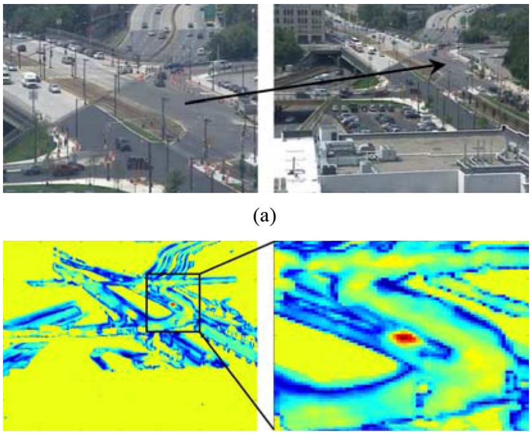


Fig. 15. Similarity map for a scene. The heat-map indicates the similarity between the selected pixel in Camera 1 and all the pixels in Camera 2. Red indicates high similarity, blue medium similarity, and yellow low similarity. (a) Matching pixels (Camera 1 on the left, Camera 2 on the right); (b) similarity heat-map in Camera 2.

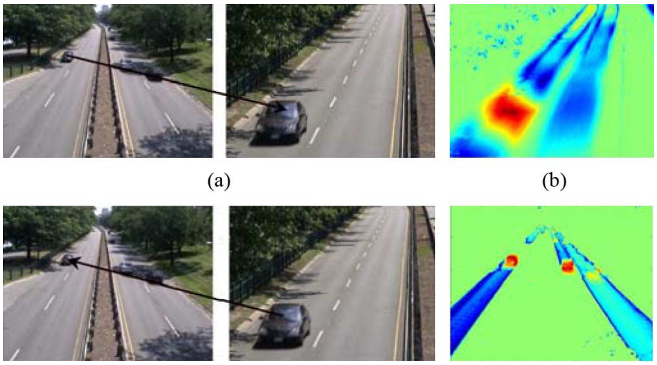


Fig. 16. Similarity map for a scene in two different cameras. In (b), the heat-map indicates the similarity between the selected pixel in Camera 1 and all the pixels in Camera 2. In (d), the heat-map indicates the similarity between the selected pixel in Camera 2 and all the pixels in Camera 1. Red indicates high similarity, blue medium similarity, and green low similarity. (a) Matching pixels (Camera 1 on the left, Camera 2 on the right); (b) similarity heat-map in Camera 2; (c) matching pixels (Camera 1 on the left, Camera 2 on the right); and (d) similarity heat-map in Camera 1.

nonsymmetric binary channel to obtain $V_2(j,\cdot)$. These types of formulations are amenable to a probabilistic analysis through Chebyshev's inequality, however this direction will be considered as part of future work.

1) Correspondence Algorithm: Based upon the derivation of the previous section, for each pixel $\mathbf{q}_j$ in Camera 2, we compute the dissimilarity measure $d(p_i, q_j)$ as defined in (3), with $\eta = \max\{\sum_{\tau=1}^{T} V_1(i,\tau), \sum_{\tau=1}^{T} V_2(j,\tau), 1\}$. This corresponds to the maximum activity at pixels $\mathbf{p}_i$ and $\mathbf{q}_j$. Let $d_i^{\max} = \max_{\mathbf{q}_j} d(\mathbf{p}_i, \mathbf{q}_j)$ and $d_i^{\min} = \min_{\mathbf{q}_j} d(\mathbf{p}_i, \mathbf{q}_j)$. We then define a similarity function between the time series of $\mathbf{p}_i$ and $\mathbf{q}_j$ as $s(\mathbf{p}_i, \mathbf{q}_j) = d_i^{\max} - d(\mathbf{p}_i, \mathbf{q}_j)/d_i^{\max} - d_i^{\min}$, where $s(\mathbf{p}_i, \mathbf{q}_j) \in [0,1]$.

Remark: Our goal here is to demonstrate that using the activity based features we can find correspondences in urban monitoring scenarios without the need for calibration. Therefore we will proceed with the similarity measure defined previously. However, other similarity functions can be considered such as the maximum normalized cross-correlation or the dynamic time warping distance(see [25], [26] and references therein) between the time series of two pixels that are candidate matches.

Before we formally describe how we obtain the matching pixel in Camera 2, we present some similarity heat-maps to provide the reader with an intuitive understanding of our method. These similarity heat-maps correspond to the similarity function between a pixel in one camera and all the other pixels in the other, and a high similarity is indicated with a dark red where lower similarity levels are indicated with yellow or green. Figs. 15 and 16 present previews of matched pixels and associated similarity heat maps.

When there is little activity in the region of observation, as is the case where only a short video is available, pixels in different regions may exhibit high similarities as predicted by the hypothesis test of (3), and as shown in Fig. 16(d). In order to handle these situations, we assign the corresponding pixel in Camera 2 to be $\mathbf{q}_{j^*}$, where we obtain the index j^* by 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 [27]. See Algorithm 2. Briefly, this algorithm operates as follows: Given two binary videos V_1, V_2 and a pixel $\mathbf{p}$ in V_1, we first calculate the similarity between $\mathbf{p}$ and all the pixels $\mathbf{q}$ in V_2. Then, there will be a high similarity between $\mathbf{p}$ and in a subset Q of the pixels in V_2, i.e., $Q = \{\mathbf{q} : s(\mathbf{p},\mathbf{q}) > \hat{s}\}$, where $\hat{s}$ is the threshold of similarity above which we say two pixels have a high similarity. When we eliminate the outliers from Q, we can essentially take the center of mass (mean of the coordinates of the pixels in Q) and declare that this center of mass is the matching pixel to $\mathbf{p}$. This method is formalized in the Algorithm 2, where we used $\hat{s} = 0.9$ to obtain the presented results.

Algorithm 2 Activity Matching

```
Input: $V_1,\ V_2,\ \mathbf{p}$ in $V_1$,
Output: $\tilde{\mathbf{q}}$ in $V_2$
1:Estimate the similarity $s(\mathbf{p},\mathbf{q})$ following (3)
2:$t = 0$, $Q^t = \{\tilde{\mathbf{q}} : s(\mathbf{p},\tilde{\mathbf{q}}) \geq \hat{s}\}$
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:    $d_E(C^t,\tilde{\mathbf{q}}) = \sqrt{(C_x^t - \tilde{\mathbf{q}}_x)^2 + (C_y^t - \tilde{\mathbf{q}}_y)^2}$
6:end for
7:Let $d_Q^t = \{d_E(C^t,\tilde{\mathbf{q}})\}$ and $\mathrm{med}_Q^t \doteq \mathrm{median}\{d_Q^t\}$
8:Set $Q^{t+1} = \{\tilde{\mathbf{q}} : \tilde{\mathbf{q}} \in Q^t,\ d_E(C^t,\tilde{\mathbf{q}}) \leq \mathrm{med}_Q^t\}$
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:$\tilde{\mathbf{q}} = C^t$
```

The method described previously maps the activity regions, however it requires motion information to be present in the time series data of each pixel that is to be matched. For certain cases it is possible to use the well-known homography mapping method (see [28], [29] and references therein), where the homography transformation matrix can be estimated using the matching results based upon activity regions, and the estimated homography matrices can be used to find the matches in the inactive regions.

2) Identifying Good Matches: Once we obtain a set of matching pixels in each camera, we next move on to identify which matches are correct. For this purpose we employ a two step process:

Left-Right Check: First, a left-right check is performed to validate that $(\mathbf{p} \leftrightarrow \mathbf{q})$ is a true match (see Algorithm 3). Briefly, the left-right check method operates as follows: We begin with a pixel $\mathbf{p}$ in Camera 1 and find its corresponding pixel $\mathbf{q}$ in Camera 2 using the proposed matching method in Algorithm 2. Then, we input $\mathbf{q}$ in Camera 2 into the matching algorithm and find its corresponding match $\mathbf{p}'$ in Camera 1. If $\mathbf{p}$ and $\mathbf{p}'$ are within a desired range of each other, then $(\mathbf{p} \leftrightarrow \mathbf{q})$ is kept as a correct match. If $\mathbf{p}$ and $\mathbf{p}'$ are separated more than the desired amount then the pair $(\mathbf{p} \leftrightarrow \mathbf{q})$ is labeled as an incorrect match and dropped.

Algorithm 3 Left right check

```
Input: $V_1,\ V_2,\ \mathbf{p},\ \epsilon$
Output: Decision
$\mathbf{q} = \text{best match from Algorithm 2 } (V_1, V_2, \mathbf{p})$
$\mathbf{p}' = \text{best match from Algorithm 2 } (V_2, V_1, \mathbf{q})$
if $||\mathbf{p} - \mathbf{p}'|| < \epsilon$ then
    return $\mathbf{q}$
else
return NULL
end if
```

Homography and Outlier Rejection: Although the left-right check procedure is well geared to locate good pairs of matching pixels, it can nevertheless return outliers (i.e., erroneously matched pairs $(\mathbf{p}_j \leftrightarrow \mathbf{q}_j)$ that show similar behavior). In our experiments, up to 10% of the pairs of matches returned by the left-right check were outliers. In order to reduce the effect of these outliers, we use what are called the homography matrices, estimated with Ransac (RANdom SAmple Consensus) algorithm [30], [31]. A homography matrix H relates each pixel $\mathbf{p}_j$ in Camera 1 to its associated pixel $\mathbf{q}_j$ in Camera 2, and can be expressed as: $\mathbf{q}_j = H\mathbf{p}_j$ where H is a 3×3 matrix and $\mathbf{p}_j = (p_{xj}, p_{yj}, 1)^{\mathrm{T}}$ and $\mathbf{q}_j = (q_{xj}, q_{yj}, 1)^{\mathrm{T}}$. This is true, for example, when the observed scene is planar.

Estimation of homography matrices with Ransac not only returns a more precisely estimated matrix, but also identifies the outliers. The outliers are then removed and the remaining matches are used to perform topology reconstruction, e.g., estimate the projection, fundamental, and essential matrices as well as extrinsic camera parameters. See [32] for an analysis of the Ransac algorithm as used in computer vision applications.

B. K-Camera Matching

Up until now, we described how matching can be performed between two cameras with overlapping field of views. Now lets consider how this generalizes to a K-camera setup. One reason for matching pixels in K cameras is to retrieve the topology of a system, i.e., recover the projection and the calibration matrices of each camera up to an overall projective transformation [31]. In that case, the goal is to find a pixel $\mathbf{q}_i$ in each camera C_i which map to the same 3-D point in the scene. This happens when the K-cameras have overlapping field of views as is the case, for example, when filming a sport field from different position and orientation. Assuming that all cameras look at a scene containing sufficient activity to allow for 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. This procedure is illustrated in Algorithm4 : given a pixel $\mathbf{p}$ in camera C_1, the algorithm returns its projection in the $K - 1$ other cameras. The algorithm returns NULL when $\mathbf{p}$ does not map to a pixel in one (or more) camera. Note that the homography matrix between each pair of camera can be estimated using Ransac to allow for more precise solutions. Note that line 3, 4 and 5 can be removed from Algorithm 4. In that case, the algorithm can be used to recover which cameras have overlapping FOVs and which does not. This is useful in wide camera networks. Later in Section IV we present experimental results for camera structure estimation for 4 cameras.

Algorithm 4 K-Camera Matching

```
Input: $\{V_1, V_2, \ldots, V_K\}, \mathbf{p}$
Output: $Q = \{\mathbf{q}_2, \mathbf{q}_3, \ldots, q_K\}$
1:for $j = 2$ up to $K$ do
2:$q_j$ = Left_right_check $(V_1, V_j, \mathbf{p})$ // Algo 3.
3:if $q_j == \text{NULL}$ then
4:    return NULL
5:end if
6:end for
7:return $Q$;
```

IV. Results

A. Activity-Based Matching

1) Performance Evaluation: In order to quantify the performance of our matching algorithm, we devised an indoor example where we placed two cameras with opposite orientations to a plane of observation. We used the snapshot of a grid on the floor to calculate a ground truth map. We next used a remote controlled toy car to generate activity in the scene, randomly selected a number of pixels in Camera 1, and found their matches in Camera 2 using the method described in Section III. We then calculated the Euclidean distance between our match

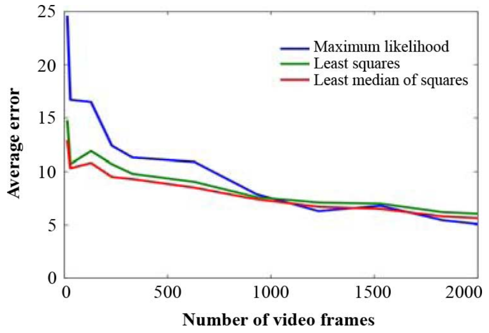


Fig. 17. Average Euclidean distance between the ground truth matches and the matches found using the proposed method. As the length of the video sequence increases, the error decreases significantly. When a small number of frames are available the least median of squares is the most robust method.

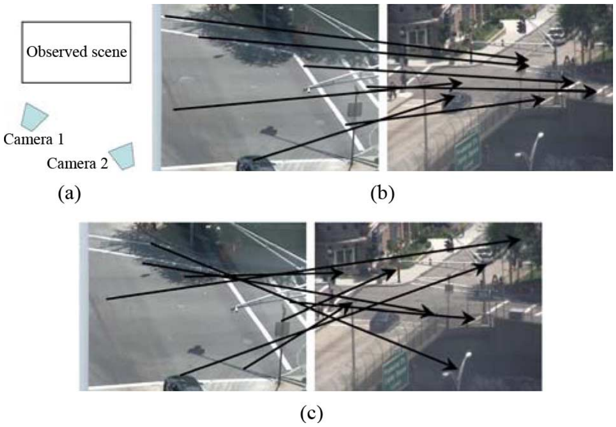


Fig. 18. (a) Camera setup, (b) matching results using 90 s of video using the proposed method, and (c) matching results using SIFT. The cameras are not calibrated to have the same zoom levels. The proposed method is able to match the corresponding pixels with a small error, whereas the SIFT method fails to find correct matches.

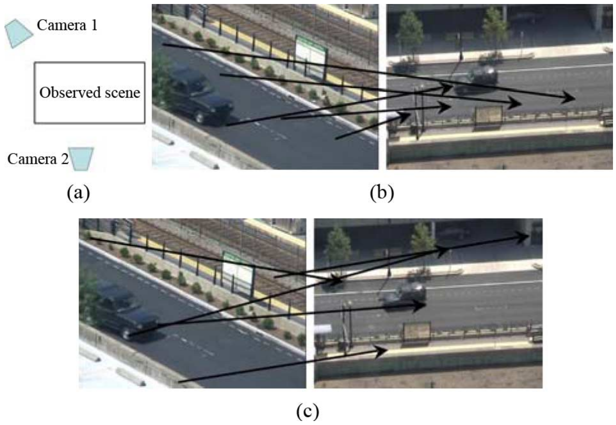


Fig. 19. (a) Camera setup, (b) matching results using 90 s of video using the proposed method, and (c) matching results using SIFT. The proposed method is able to match the corresponding pixels with a small error, whereas the SIFT method fails to find correct matches.

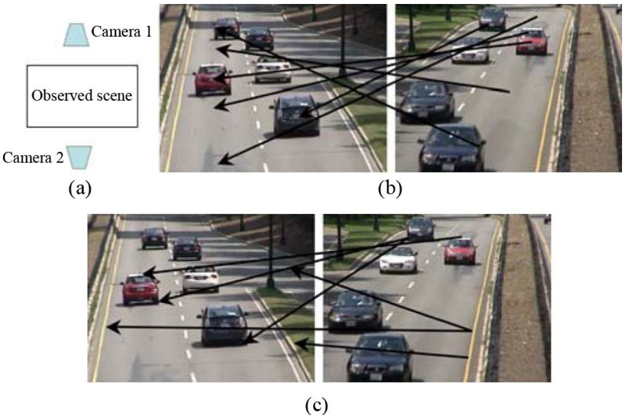


Fig. 20. (a) Camera setup, (b) matching results using 90 s of video using the proposed method, and (c) matching results using SIFT. The proposed method is able to match the corresponding pixels with a small error, whereas the SIFT method fails to find correct matches.

and the ground truth match, and defined average error as the average Euclidean distance between our estimate and ground truth across selected pixels. The results, demonstrated in Fig. 17, indicate that the least median of squares method is more robust than the maximum likelihood and least squares methods when the video sequence is short.

2) Matching in Real Life Scenarios: Next 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. For comparison purposes, we also present the results we obtained with the SIFT method [12]. The results are presented in Fig. 18 through Fig. 20.

Remark: In scenarios involving vehicles, neither the ghost activity nor the varying occupancy durations affected the performance of the proposed method significantly, and therefore in these experiments we did not utilize the intermediate step described by Algorithm 1. However, in the upcoming subsection we will give an example of how ghost activity affects the performance of the proposed method when pedestrians are involved.

3) Effects of Ghost Activity: We now present an example where the ghost activity challenges the premise of geometry independence, and how Algorithm 1 proposed in Section II-C resolves the issue. For this experiment we took videos of pedestrians walking. We first used the matching algorithm on the original binary video sequences. Next, we used Algorithm 1 described in Section II-C and replaced the foreground objects with rectangles. We then applied the matching algorithm on the new video, where the objects are replaced by the rectangles. The removal of ghost activity has drastically improved the results of the matching algorithm. Fig. 21 presents these results. Again, for comparison purposes we present the results obtained by the SIFT method as well. To give a quantitative example, mean matching error with respect to the ground truth was 68 pixels before removal of ghost activity and 12 pixels after, which amounts to more than 80% reduction in average matching error. In this setup the SIFT method failed to find good matches.

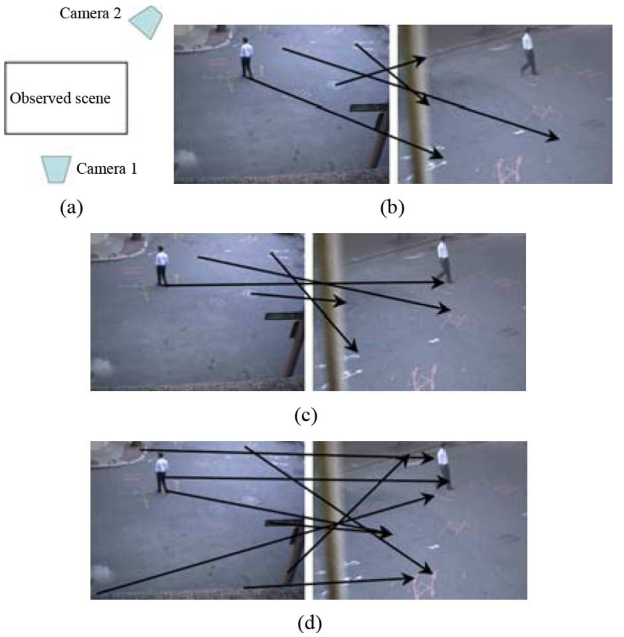


Fig. 21. (a) Camera setup, (b) matching results without removal of ghost activity, (c) matching results when the ghost activity has been removed (Algorithm 1), and (d) matching results using SIFT. While the proposed method has a high error rate (68 pixels) with the ghost activity present in the pedestrian case, it performs matching to within several (12) pixels when the ghost activity is removed. The SIFT method fails to find good matches.

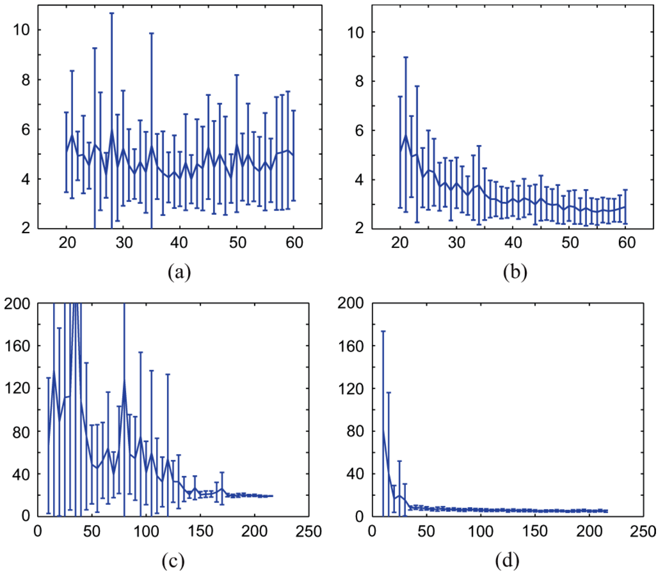


Fig. 22. Robustness of our method with (and without) Ransac with respect to the number of pairs of points N used to estimate the fundamental and homography matrices for the scenario presented in the first row of Fig. 26. (a) $E(F)$ versus number of matched pairs with SVD without using Ransac, (b) $E(F)$ versus number of matched pairs with SVD using Ransac, (c) $E(H)$ versus number of matched pairs with SVD without using Ransac, (d) $E(H)$ versus number of matched pairs with SVD using Ransac.

B. *Applications to Topology Reconstruction*

1) 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 detecting motion (c.f (3)) we wish to evaluate the robustness of our method to noisy motion masks. 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 by carefully hand selecting 20 pairs of points. Once H_{gt} is known, a residual error function for H and F is implemented

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

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

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

In Fig. 22, error curves have been obtained with and without Ransac for the scenario presented in the first row of Fig. 26. For each value N on the X-Axis, we randomly selected N pairs of points based upon 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. This result is noteworthy considering that the precision of the manually estimated ground truth is about 1 pixel.

Fig. 23 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 (3). For each noise percentage, a number of $N = 100$ pairs of points are randomly obtained, based upon which F and H are estimated. This procedure is repeated 10 times to get an average error and a variance for each noise value. In a similar way, Fig. 24 shows the effect of the video length on the performance of the proposed method. In both case, the results are clearly in favor or Ransac, especially with a low noise level and a small number of frames.

In order to compare our method to a competing method, we estimated F and H using Ransac and matches obtained with SIFT [31]. Of course, since SIFT is unable to find good matches when cameras' orientation are too different, we took a "SIFT-friendly" sequence, namely the one presented in the first row of Fig. 26. The error curves in Fig. 25 show the precision for the fundamental and homography matrices when different numbers of matching points are being used. From these curves, we noticed that our method is at most 3 pixels away from SIFT, which is less than 1% deviation.

We mentioned in Section III-B that the two-camera matching procedure generalizes well to a K-camera setup. To validate this assertion, we recovered the 3-D position and orientation of a system made of four cameras (see Fig. 28). Here, the projection and calibration matrices of each camera have been recovered following Strum and Triggs' method [33]. The four cameras are located on the fifth floor of a building, and the height effect is successfully recovered in the 3-D reconstruction of the camera locations. We also computed the average difference between the matches obtained by our method and those obtained by hand

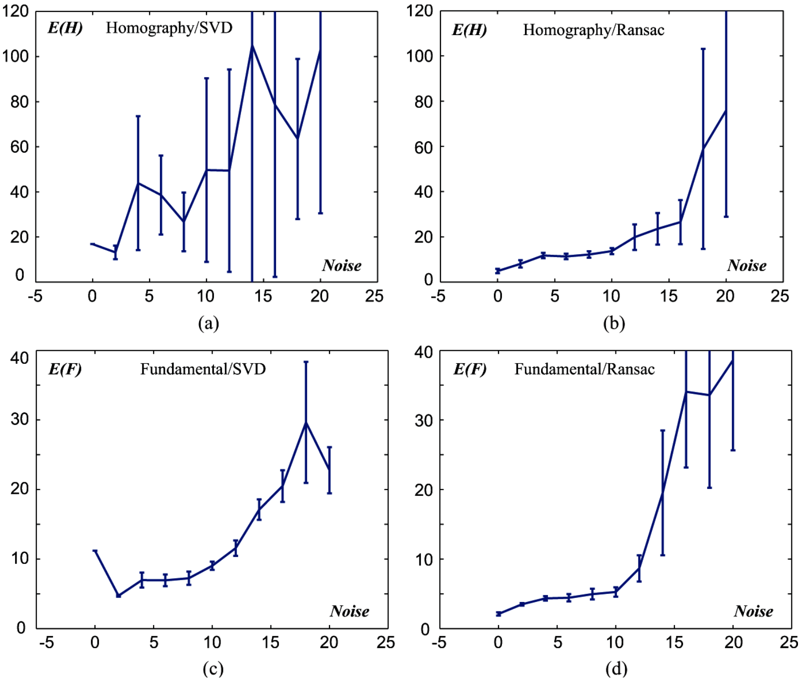


Fig. 23. Robustness of our method with (and without) Ransac with respect to noise for the scenario presented in the first row of Fig. 26. (a) $E(F)$ versus noise level without Ransac; (b) $E(F)$ versus noise level with Ransac; (c) $E(H)$ versus noise level without Ransac; and (d) $E(H)$ versus noise level with Ransac.

selection for each pair of Camera C_i and C_j . This led to an average error of only 2.6 pixels which is less than 1% error considering the size of the images.

2) Qualitative Results: In Fig. 26, the homography of five different pairs of videos are presented. In order to illustrate the homographic transformation from Camera 1 to Camera 2, a 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 the example presented on the fourth row [(m) through (p)] in which cameras are placed on opposite sides of the boulevard and, thus, the transformation results into a flip of the corners. In the first three rows [(a) through (l)] the camera orientations are similar, and both SIFT and the proposed method perform well. However, in the last two rows [(m) through (t)] the SIFT method fails to find a homography whereas our method still performs well. Here SIFT fails 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 $\mathbf{q}F^{\mathrm{T}}$. These are presented in Fig. 27.

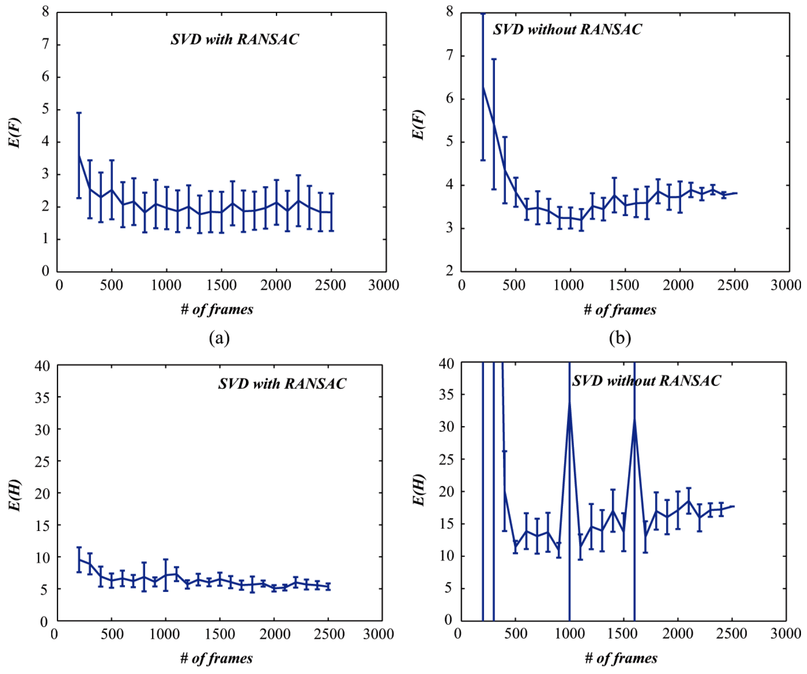
Note that using the proposed method we can also generate occlusion maps by simply using a left-right check once the matching is performed. In the following, we present several occlusion maps where we identify three regions in the images: the blue colored regions are present in both cameras, the red colored regions exist in only one camera and not the other, and the green colored regions are the no-motion regions. Fig. 29 presents the results of this segmentation.

3) Effects of Ghost Activity: In order to understand the impact of Algorithm 1 on topology reconstruction we estimated the homography transformation for the pedestrian scenario that was presented in Fig. 21. The results are presented in Fig. 32. The average matching error with respect to the ground truth was 68 pixels before removal of ghost activity and 12 pixels after the removal. This amounts to more than 80% reduction in average matching error. As in the previous cases where the camera orientations were significantly different, the SIFT method did not produce meaningful results.

4) Limitations: Our method is not void of drawbacks. We mentioned previously that our method is robust to spurious false positives and false negatives in V_1 and V_2 (Fig. 23). However, we noticed that the method is likely to break down when the number of false detections gets too large. We empirically observed that false positives caused by environmental changes are the most common source of error. One typical example is shown in Fig. 30. In this example, the method detects pedestrians and replace it by a small rectangle following the aspect ratio normalization algorithm. The method worked well up until when the illumination changed drastically. This lead to a large number of false positives that went on for approximately 500 frames. These false positives lead the method to detect pedestrians almost everywhere in the video frames.

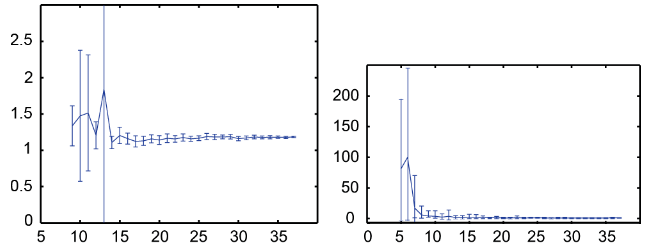
V. Communication Efficiency Through Compressive Sensing

The emerging surveillance scenario of distributed camera network of smart cameras motivates us to consider distributed processing and communication efficiency aspects. In general, a small number of pixel level correspondences are sufficient for topology reconstruction and aid in other higher level tasks. For example, we show that with about 50 correspondences we can estimate the homography and fundamental matrices reliably. In order to obtain these matches in a distributed setting, we need not transmit the whole video or even entire frames across the network. Instead to find a match between two cameras we



(a) (b) (c) (d)

Fig. 24. Robustness of our method with (and without) Ransac with respect to the length of video for the scenario presented in the first row of Fig. 26. (a) $E(F)$ versus video length with SVD without using Ransac; (b) $E(F)$ versus video length with SVD using Ransac; (c) $E(H)$ versus video length with SVD without using Ransac; and (d) $E(H)$ versus video length with SVD using Ransac.



(a) (b)

Fig. 25. Robustness of Sift with Ransac with respect to the number of points used for matching. (a) $E(F)$ (b) $E(H)$.

merely need to ensure that the transmitted waveform has the distance preservation property. This is because the correspondence algorithm is based upon the squared-norm distance given by (3). It turns out that random projections satisfy such a property, namely, the squared norm distance in the projected space is essentially equal to the squared norm distance in the original space. This provides a bound on the size of the message set.

Concretely, let $G \in \mathbb{R}^{M\times T}$, be a random matrix consisting of IID Bernoulli $\{-1,\ 1\}$ components with equal probability. The transmitted waveform corresponding to any pixel, p_i, is given by

$$Y_i(t) = \sum_{\tau=1}^{T} G(t,\tau)V_i(\tau),\ t = 1,\,2,\,\ldots,M.$$

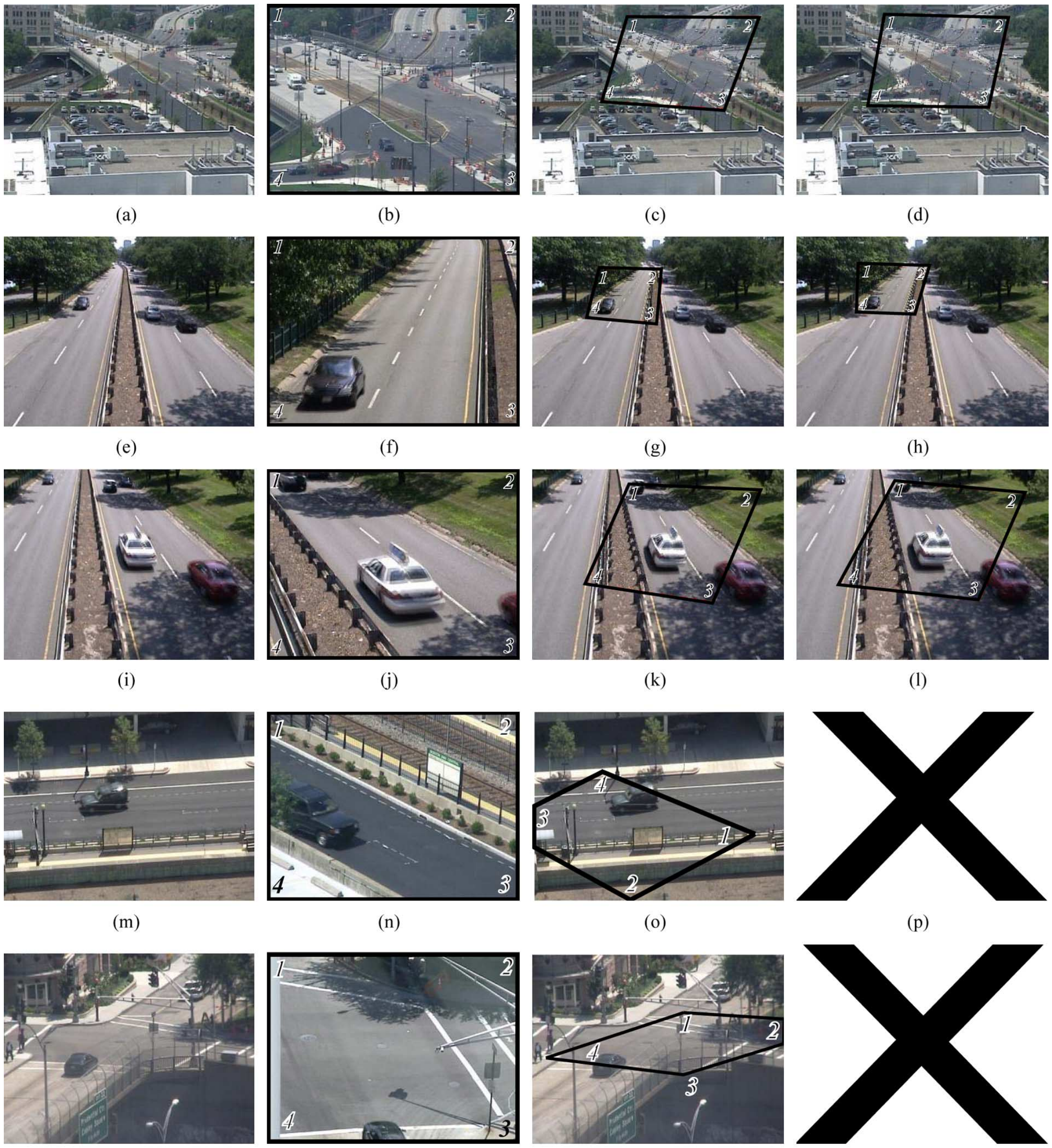
The celebrated Johnson-Lindenstrauss lemma [34] states that

$$\|V_i(\cdot) - V_j(\cdot)\|_2 \approx \frac{1}{M}\|GV_i(\cdot) - GV_j(\cdot)\|^2$$

for all pixels p_i as long as $M = \Omega(\log(N))$, where N is the number of pixels. Thus, if there are J cameras with each camera having N pixels the communication message complexity scales as $O(J\log(N))$ for establishing correspondence. Note that the dimension T of the time series is irrelevant for ensuring distance preservation. Also note that each component of the waveform, namely, $Y_i(t)$, is an integer with its maximum value typically less than the number of moving objects passing through that pixel. Consequently, if S is the maximum number of objects then the communication bit complexity scales as $O(J\log(N)\log(S))$.

To understand the inherent tradeoffs we conducted experiments on the high-way data. We took two video sequences and multiplied their binary signatures with a $M \times T$ random matrix. Then, the compressed signatures $V_1'(i,\tau)$ and $V_2'(i,\tau)$ containing only M values are used directly in our matching procedure (Algorithm 2). As predicted by the theory M corresponds to only 5% of T while V_1 and V_2 contain an average amount of activity of 20%, the matching did not seem to suffer from the random projection. These are illustrated in Fig. 31 (a) and (b).

In certain cases one may wish to reconstruct the entire binary time series for each camera at a remote destination. This situation requires more communication resources. However, the inherent sparsity of the binary time series can be exploited in these cases. Note that the binary time series $V_i(t)$ for each pixel p_i is sparse if busy periods are sufficiently small and the number of objects are relatively small. When these assumptions are violated, namely, when the busy periods are relatively large, we can obtain a sparse time series, $D_i(t)$, through differentiation of the binary series. Indeed in this alternative representation the number of nonzero entries is proportional



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

Fig. 26. Homography results. First column: view from Camera 1; second column: view from Camera 2; third column: homography obtained with our method, Fourth column: Homography obtained with SIFT (with Ransac). In the first three rows the camera orientations are similar, and both SIFT and the proposed method perform well. However, in the last two rows the camera orientations are significantly different, and the SIFT method fails to find a homography and generate a homography whereas the proposed method still performs well.

to the number of moving objects. The distributed algorithm is again through random projections and the reconstruction is an application of compressive sensing [35]. Each camera randomly projects the derivative measurements, $D_i(t)$ or the original time series $V_i(t)$ (whichever is sparser) for all of the pixels onto an M dimensional space. In this case it turns out from well-known results in compressive sensing that if there are S moving objects in time duration, T, then for J cameras with each camera having N pixels the communication complexity scales as $M = O(S \log(T/S))$ messages per pixel per camera. To understand the implication consider 1000 frames of video. Suppose the total amount of activity is 20% corresponding to 10 busy periods (i.e., 10 moving objects), then there are only 2×10 nonzero entries in $D_i(t)$.

VI. Multicamera Abnormal Activity Detection

We adopt the implicit assumption of conventional statistical anomaly detection, namely that the nominal data occurs in high likelihood regions of the nominal model, while the anomaly occurs in the low likelihood regions. One approach to anomaly detection is to learn a nominal distribution, $g_0(\cdot)$ from training data (see [36]) and declare as anomalies those test points, x, which are least likely under the nominal distribution.

One of the difficulties of anomaly detection in video is what features would appropriately capture the nominal activity. While we do not have an answer to this question, we attempt to capture nominal activity by means of busy-idle time periods. Such features have been extensively employed by [24] resulting

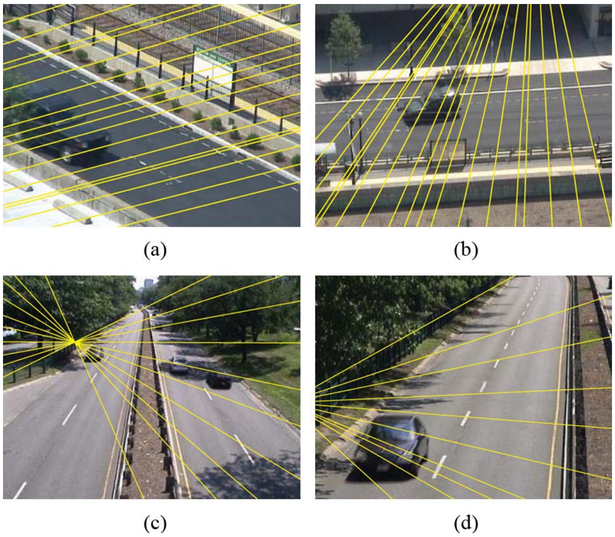


Fig. 27. Epipoles and epipolar lines for two scenes. As can be seen, the epipolar lines in (a) and (c) correspond well to those in figure (b) and (d).

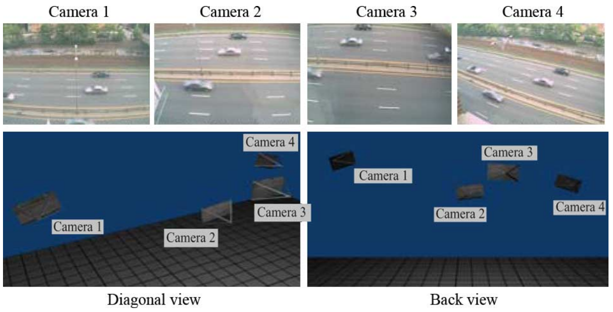


Fig. 28. Structure estimated from a 4-view system made of cameras located on the fifth floor of a building. The recovered position and orientation of the cameras correspond well to the actual configuration of the network.

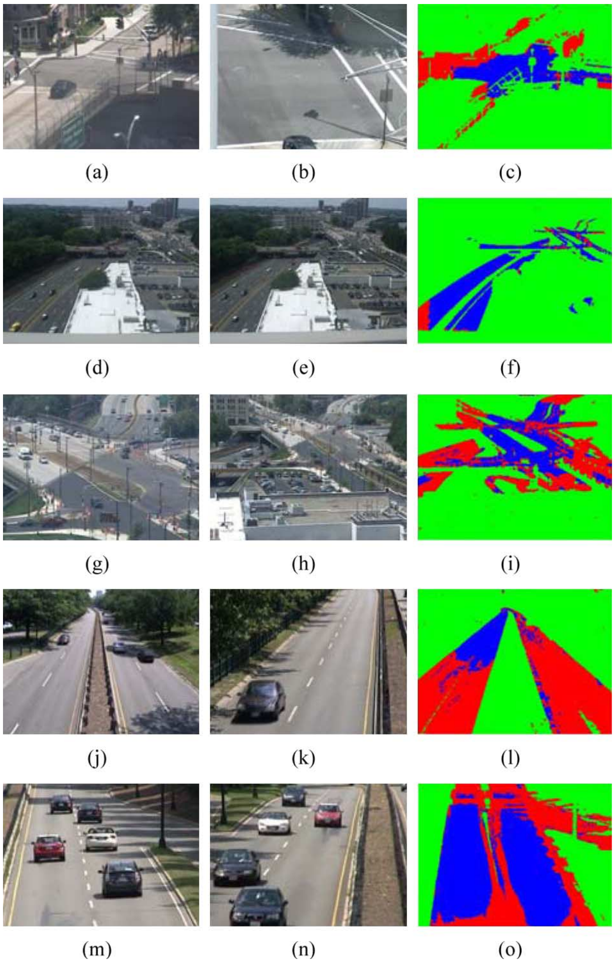


Fig. 29. Occlusion map using left-right check and the proposed method: (a,d,g,j,m) Camera 1 frames, (b,e,h,k,n) Camera 2 frames, (c,f,i,l,o) Segmentation results for Camera 1 frames: Red regions appear in Camera 1 frame but not in Camera 2 frame, blue regions are common in Camera 1 and Camera 2, green regions carry no motion.

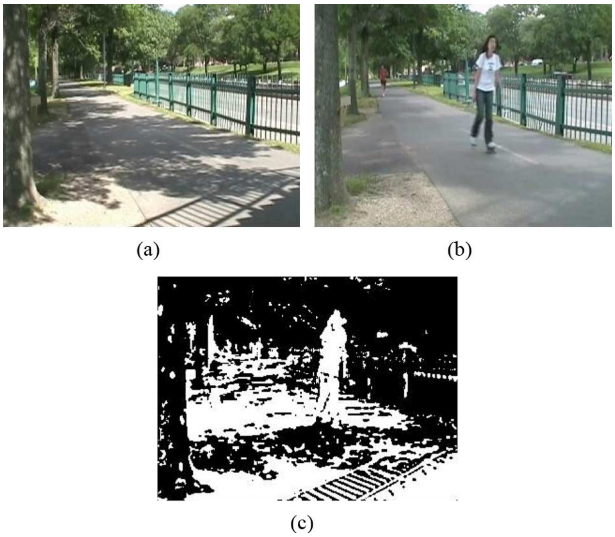


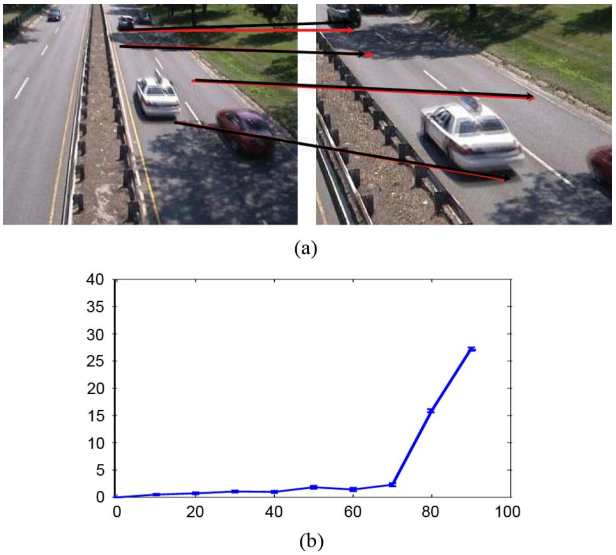
Fig. 30. Typical situation for which our method fails. In this case, the global illumination changed rapidly [image (a) was taken 5 s before image (b)] leading to severe distortions in the motion detection label field (c).

in successful detection of anomaly in many interesting and general unsupervised situations. To this end, consider the proposed activity features for pixel $\mathbf{p}_i$, a sequence of ones and zeros

$$V_1^i = (V_1(i,1), V_1(i,2), \ldots, V_1(i,\tau)). \tag{4}$$

Let B_n^i (busy rate) denote the length of nth set of consecutive ones in V^i and I_n^i (idle rate) denote the length of the nth set of consecutive zeros in V^i, as depicted in Fig. 33.

For each pixel $\mathbf{p}_i$, (B_n^i, I_n^i) tuples are samples of a 2-D distribution, where the unknown distribution characterizes the behavior at $\mathbf{p}_i$. The (B_n^i, I_n^i) samples are independent (as we described before in Section II and identically distributed, and we define the behavior observed at a pixel $\mathbf{p}_i$ as the underlying distribution that generates the (B_n^i, I_n^i) tuples. Also, since these behavior features are derived from the activity features, they also possess the geometry independence property. Let $\{(B_1^i, I_1^i),\ (B_2^i, I_2^i), \ldots, (B_{s_i}^i, I_{s_i}^i)\}$ be the set of s_i busy-idle samples for pixel $\mathbf{p}_i$ in the training video, and $\hat{f}^i$ the corresponding density estimate, which we now call the *learned behavior model*. We now have reduced the video anomaly detection problem to a simple statistical anomaly detection problem. Consider the time series for pixel $\mathbf{p}_i$ in the test sequence, a sequence of ones and zeros: $\hat{V}_1{}^i = (\hat{V}_1(i,1), \hat{V}_1(i,2), \ldots, \hat{V}_1(i,\tau'))$. The test video at a pixel is reduced to a sequence of busy-idle periods (or a windowed average of the busy idle periods). The likelihood of the


(a)

(b)

Fig. 31. (a) In red, matches obtained with Algorithm 2 (zero compression) and in black, matches obtained in compressed space (85% compression). (b) Mean squared error for matches obtained with Algorithm 2 (zero compression) and different compression rates. The error scale is in pixels. (a) Matching in compressed space; (b) MSE versus compression ratio.

test sample is compared against a threshold, $\mathcal{L}$. The threshold $\mathcal{L}$ is chosen to control the false alarm below a desired threshold γ.

A. Multicamera Fusion and Anomaly Detection

The anomaly detection method described previously can be readily generalized to multicamera setup since all of the different cameras observe identical behavior distributions. In particular, let $\mathbf{p}_i$ in Camera 1 and $\mathbf{q}_j$ in Camera 2 be the corresponding pixels that observe the same location. Then under the idealized scenario ($\Gamma = 1$) their behavior models f^i and g^j are the same. As a consequence, the behavior model learned from the samples of $\mathbf{p}_i$ can be imported to $\mathbf{q}_j$, and this model can be used to detect anomalies in $\mathbf{q}_j$. By importing the behavior models generated at pixel $\mathbf{p}_i$, we can achieve the receiver operating characteristics (ROC) curve at $\mathbf{q}_j$ with insignificant training for pixel q_j.

To demonstrate this idea, we consider a scenario where two pixels $\mathbf{p}_i$ and $\mathbf{q}_j$ observe a sidewalk with a bimodal behavior. In Mode 1 the scene is crowded, consequently the idle periods are short (about 1.5 s). In Mode 2, the scene is not crowded and the idle periods are longer (about 12 s). When people go through the regions observed by the pixels, they take around 1 s to pass. We assumed that the ground truth scaling ratio $\Gamma = 1.2$. We found that this multimodal situation to be realistic from our experiments and used this to generate a ground truth model for pixel q_j(using a 30 frame per second recording rate)

$$H_0^{\mathbf{q}_j} : (B, I) \sim 0.8\mathcal{N}\left(\begin{bmatrix} 30 \\ 45 \end{bmatrix}, \begin{bmatrix} 100 & 0 \\ 0 & 225 \end{bmatrix}\right) + 0.2\mathcal{N}\left(\begin{bmatrix} 30 \\ 350 \end{bmatrix}, \begin{bmatrix} 100 & 0 \\ 0 & 10000 \end{bmatrix}\right).$$

We now populated this dataset with anomalies drawn uniformly in the pixel space $[0,\ 80] \times [0,\ 600]$ of Camera 2. Now Camera 1 collects samples from its ground truth nominal behavior model for pixel, p_i, learns a behavior, and sends this model to $\mathbf{q}_j$ (call this model $\hat{f}^i$). Then, we generate ROC curves at $\mathbf{q}_j$ for the following scenarios:

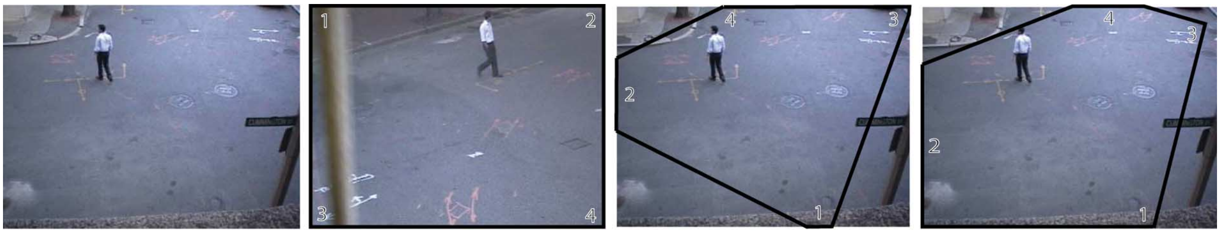
1) Directly using the received model: $\mathbf{q}_j$ uses $\hat{f}^i$ directly to perform anomaly detection,
2) Model correction (40 s, 2.8 min): $\mathbf{q}_j$ collects about 40 s (resp. 2.8 min) of training data from its ground truth normal behavior model, obtains an ML estimate of the scaling parameter Γ, and transforms $\hat{f}^i$ back to its observation space. Then it performs anomaly detection with this model,
3) Learning (2.8, 5.6, 8.5 min): $\mathbf{q}_j$ collects about 2.8 min (resp. 5.6, 8.5 min) of training data from its ground truth normal behavior model to learn its own behavior model. It then performs anomaly detection with this model.

The resulting ROC curves are presented in Fig. 34. The results demonstrate that for the modeled scenario, when $\mathbf{q}_j$ receives the model from $\mathbf{p}_i$ and performs model correction through an ML estimate of Γ with 40 sec of training data, it performs better than it would have if it learned the model from scratch with more than 8.5 min of data. In the scenarios where the cameras of a heterogeneous network change views frequently (once every hour or so), 8 min is a very significant improvement. Interestingly, in the scenario presented where $\Gamma = 1.2$, even without any model correction the detection performance is better if the received model is used as opposed to learning the model from scratch with less than 5.6 min of data.

We have also performed real life experiments on information fusion where two cameras observe a scene from significantly different orientations with different zoom levels. We learn a behavior model with one camera, send the model to the second camera, and use this model as a surrogate behavior model in the second camera to detect anomalies. The results are presented in Fig. 35. This is a powerful result in that it demonstrates that we no longer need to learn the behavior models from scratch when the topology changes. Instead, we can import the corresponding models learned from the previously active topologies to detect anomalies in the current topology.

VII. Conclusion

In this paper we considered the problem of matching in a heterogeneous camera network with overlapping fields of view. Motivated by wide-area surveillance applications, we considered situations where cameras have arbitrary orientations and zoom levels. This led us to propose geometry independent features. We then developed a new algorithm for matching (Algorithm 1) camera regions based upon activity. We demonstrated that the algorithm works well without the knowledge of any camera parameters such as location, orientation, epipolar geometry, etc. Based upon our development we presented real-world examples as well as quantitative performance evaluation. We also presented the results obtained with SIFT method, and the conclusion of this study was that if the cameras have similar orientations the SIFT and our method both work comparatively



(a) (b) (c) (d)

Fig. 32. Homography results. (a) View from Camera 1; (b) view from Camera 2; (c) homography obtained before removal of ghost activity; (d) homography obtained after removal of ghost activity. The homography estimate after removal of ghost activity is significantly better. The average matching error with respect to the ground truth was 68 pixels before removal of ghost activity and 12 pixels after, which amounts to more than 80% reduction in average matching error.

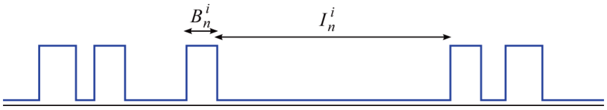
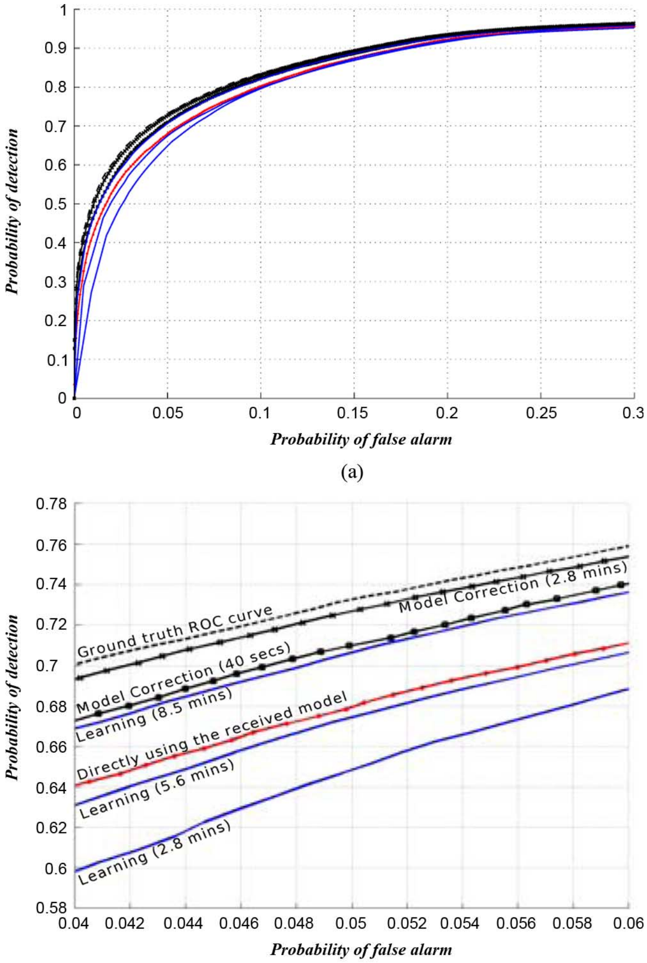


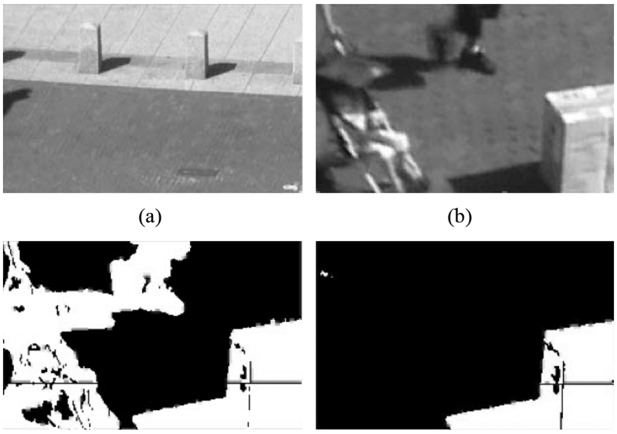
Fig. 33. Time series and sample (B, I) rates for a pixel: B_n^i (busy rate) denote the length of nth set of consecutive ones in the time series and I_n^i (idle rate) denote the length of the nth set of consecutive zeros.



(a)

(b)

Fig. 34. Resulting ROC curves for various scenarios. When $\mathbf{q}_j$ receives the model from $\mathbf{p}_i$ and performs model correction through an ML estimate of Γ with 40 s of training data, it performs better than it would have if it learned the model from scratch with more than 8.5 min of data. (b) presents a labeled, zoomed version of (a). (a) ROC curve; (b) ROC curve zoomed.



(a) (b)

(c) (d)

Fig. 35. Anomaly detection in Camera 2 using the behavior models from Camera 1: In the training video we observe a sidewalk with pedestrians walking, and this is assumed to be normal behavior. Using the training video obtained by Camera 1 we generate a behavior model for each behavior cluster, and share this information with Camera 2. In the test video we abandon an object on the sidewalk, which generates anomalous behavior. Then we detect this anomaly in Camera 2 using the experience of Camera 1. Observe that the cameras have significantly different zoom levels as well as different views with respect to the scene and that the pedestrians are absent in the anomaly detection results presented in (d). (a) Camera 1: the view in training video; (b) Camera 2: the view in test video; (c) Camera 2: binary motion frame; (d) Camera 2: detected anomaly.

well. However, if the camera orientations are significantly different, then our method still performs well whereas the SIFT method fails. We also presented examples of topology reconstruction based upon the matching results of our method. These results demonstrated that even for significantly different camera orientations we are able to recover the topology of the camera network with satisfactory results. We then exploited the inherent sparsity of activity sequences to develop a compressed sensing formulation to realize significant reductions in communication cost without a major degradation in performance. Finally, we described multicamera anomaly detection using activity features. We showed that activity features enables essentially multicamera fusion analogous to multisensor fusion.

References

[1] P. K. Varshney, *Distributed Detection and Data Fusion*. New York: Springer-Verlag, 1997.

[2] D. Zhang and G. Lu, "Segmentation of moving objects in image sequence: A review," *Circuits Syst. Signal Process.*, vol. 20, no. 2, pp. 143–183, 2001.

[3] C. Stauffer and E. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[4] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proc. IEEE*, vol. 90, no. 7, pp. 1151–1163, Jul. 2002.

[5] A. Neri, S. Colonnese, G. Russo, and P. Talone, "Automatic moving object and background separation," *Signal Process.*, vol. 66, no. 2, pp. 219–232, 1998.

[6] J. Konrad, "Motion detection and estimation," in *Handbook of Image and Video Processing*, A. Bovik, Ed., 2nd ed. New York: Academic, 2005, ch. 3.10, pp. 253–274.

[7] O. Veksler, "Stereo correspondence with compact windows via minimum ratio cycle," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1654–1660, Dec. 2002.

[8] A. Ogale and Y. Aloimonos, "Shape and the stereo correspondence problem," *Int. J. Comput. Vis.*, vol. 65, no. 3, pp. 147–162, 2005.

[9] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, no. 1-3, pp. 7–42, 2002.

[10] D. Devarajan, Z. Cheng, and R. Radke, "Calibrating distributed camera networks," *Proc. IEEE*, vol. 96, no. 10, pp. 1625–1639, Oct. 2008.

[11] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc Alvey Vision Conf.*, 1988, pp. 147–151.

[12] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[13] B. Song and A. R. Chowdhury, "Stochastic adaptive tracking in a camera network," in *Proc. IEEE Int. Conf. Computer Vision*, 2007, pp. 1–8.

[14] S. Khan and M. Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 505–519, Mar. 2009.

[15] S. Sinha, M. Pollefeys, and L. McMillan, "Camera network calibration from dynamic silhouettes," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 2004, pp. 195–202.

[16] T. Svoboda, D. Martinec, and T. Pajdla, "A convenientmulti-camera self-calibration for virtual environments," *PRESENCE: Teleoperators and Virtual Environments*, vol. 14, no. 4, pp. 407–422, 2005.

[17] L. Lee, R. Romano, and G. Stein, "Monitoring activities from multiple video streams: Establishing a common coordinate frame," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 758–767, Aug. 2000.

[18] S. Khan and M. Shah, "Consistent labeling of tracked objects in multiple cameras with overlapping fields of view," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1355–1360, Oct. 2003.

[19] X. Wang, K. Tieu, and E. Grimson, "Correspondence-free activity analysis and scene modeling in multiple camera views," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 56–71, Jan. 2010.

[20] D. Makris, T. Ellis, and J. Black, "Bridging the gap between cameras," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 2004, vol. II, pp. 205–210.

[21] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[22] R. Cutler and L. Davis, "View-based detection and analysis of periodic motion," in *Proc. 14th Int. Conf. Pattern Recognition*, Aug. 1998, vol. 1, pp. 495–500.

[23] B. Gloyer, H. K. Aghajan, K.-Y Siu, and T. Kailath, "Video-based freeway-monitoring system using recursive vehicle tracking," in *Image and Video Processing III*, R. L. Stevenson and S. A. Rajala, Eds. Bellingham, WA: SPIE, 1995, vol. 2421, pp. 173–180.

[24] P.-M. Jodon, V. Saligrama, and J. Konrad, Behavior Subtraction 2009 [Online]. Available: http://arXiv.org

[25] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowl. Inf. Syst.*, vol. 7, no. 3, pp. 358–386, 2005.

[26] R. Agrawal, K. i. Lin, H. S. Sawhney, and K. Shim, "Fast similarity search in the presence of noise, scaling, and translation in time-series databases," *Proc. Very Large Data Bases*, pp. 490–501, 1995.

[27] P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection, ser. Probability and Mathematical Statistics*. Hoboken, NJ: Willey, 1987.

[28] J. Su, R. Chung, and L. Jin, "Homography-based partitioning of curved surface for stereo correspondence establishment," *Pattern Recognit. Lett.*, vol. 28, no. 12, pp. 1459–1471, 2007.

[29] R. Chung and A. Arengo, "Polyhedral environment in stereo views: Representation and extraction," *Proc. Vision Interface*, pp. 97–102, 1999.

[30] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[31] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[32] O. Chum, "Two-View Geometry Estimation by Random Sample and Consensus," Ph.D. dissertation, Czech Tech. Univ., Prague, 2005.

[33] P. Sturm and B. Triggs, "A factorization based algorithm for multi-image projective structure and motion," in *Proc. Eur. Conf. Computer Vision*, 1996, pp. 709–720.

[34] W. Johnson and J. Lindenstrauss, "Extensions of lipschitz maps into a hilbert space," *Contemp. Math.*, vol. 26, p. 189206, 1984.

[35] E. Candès, J. Romberg, and T. Tao, "Robust uncertainity principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[36] D. W. Scott and S. R. Sain, *Multi-Dimensional Density Estimation*, C. R.Rao and E. J. Wegman, Eds. Amsterdam, The Netherlands: Elsevier, 2004, pp. 229–263.

Erhan Baki Ermis received the B.S. degree in electrical engineering from Purdue University, Lafayette, IN, in 2003, and the M.S. and Ph.D. degrees from Boston University, Boston, MA, in 2005 and 2010, respectively.

His research interests focus on signal processing and statistical methods for decision making.

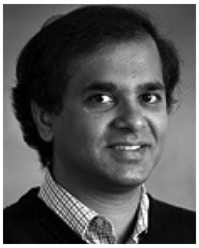
Dr. Ermis is the recipient of Boston University Electrical and Computer Engineering Award.

Pierre Clarot (S'10) is currently pursuing the M.S. degree in computer science from University of Sherbrooke, QC, Canada. He is also working toward a concurrent degree in conjunction with the ESEO, an engineering school in Angers, France.

His work focuses on computer vision and video analysis.

Pierre-Marc Jodoin (M'08) received the B.Sc. degree in computer science from the Ecole Polytechnique de Montreal, QC, Canada, in 2000, the M.Sc. degree in computer graphics, and the Ph.D. degree in computer vision and video analysis, both from the University of Montreal, QC, Canada, in 2002 and 2007, respectively.

He is currently an Assistant Professor at the University of Sherbrooke, QC, Canada. His research interests are in video surveillance, video analysis/processing, medical imaging, and computer vision.



Venkatesh Saligrama (M'01–SM'07) received the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, in 1997.

He is a faculty member in the Electrical and Computer Engineering Department at Boston University, Boston, MA. His research interests are in statistical signal processing, information and control theory, and statistical learning theory. He edited the book Networked Sensing, Information and Control.

Dr. Saligrama has been an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He is the recipient of numerous awards, including the Presidential Early Career Award, ONR Young Investigator Award, and the NSF Career Award. More information about his work is available at http://iss.bu.edu/srv.