Computer Vision and Image Understanding 127 (2014) 73–85

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Background subtraction for the moving camera: A geometric approach ☆

Daniya Zamalieva *, Alper Yilmaz

Photogrammetric Computer Vision Laboratory, The Ohio State University, Columbus, OH 43210, United States

ARTICLE INFO

Article history:
Received 10 July 2013
Accepted 13 June 2014
Available online 23 June 2014

Keywords:
Background subtraction
Object detection
Camera motion
View geometry

ABSTRACT

Background subtraction is a commonly used technique in computer vision for detecting objects. While there is an extensive literature regarding background subtraction, most of the existing methods assume that the camera is stationary. This assumption limits their applicability to moving camera scenarios. In this paper, we approach the background subtraction problem from a geometric perspective to overcome this limitation. In particular, we introduce a 2.5D background model that describes the scene in terms of both its appearance and geometry. Unlike previous methods, the proposed algorithm does not rely on certain camera motions or assumptions about the scene geometry. The scene is represented as a stack of parallel hypothetical planes each of which is associated with a homography transform. A pixel that belongs to a background scene consistently maps between the consecutive frames based on its transformation with respect to the "hypothetical plane" it lies on. This observation disambiguates moving objects from the background. Experiments show that the proposed method, when compared to the recent literature, can successfully detect moving objects in complex scenes and with significant camera motion.

1. Introduction

Background subtraction is the process of detecting objects (foreground) residing in the static scene (background). Generally speaking, background subtraction involves building a scene representation referred to as the *background model*, which is compared against incoming video frames to detect the objects. In the case when camera moves, it is important to compensate the camera motion, such that the background model and a new frame are registered. This requirement imposes significant constraints on the types of motions the camera can undergo.

A typical setting commonly adopted by many in the field is when the camera is stationary. In this setting, the static background pixels maintain their locations, such that the transformation from the incoming frame to the background model is an identity transformation, which geometrically implies that the static scene can be considered a single 3D plane. There are too many papers to cite in the literature that assume a stationary camera [1–3]. These papers and many others employ similar statistical models to represent the color distributions of pixels which include parametric [1,2] and non-parametric models [3]. There are also a few studies that perform background subtraction for stationary cameras but with *dynamic* backgrounds [4–10]. For more detailed discussion on background subtraction for stationary cameras, we refer the reader to comprehensive surveys in [11–13].

Another typical but less studied camera setting is when the camera is restricted to only pan, tilt and zoom, such that the optical center of the camera does not move [14–17]. This limited motion imposes a *rotational homography*, which is also known as the *plane projective transformation*, between the background model and the incoming frame. Similar to the stationary camera, this setting lets construction of a panoramic background model that has a wide scene coverage. The moving object detection proceeds by first registering the video frame to the panoramic background model and employing the background subtraction with this registered model.

When the optical center of the camera moves, the one-to-one mappings between the background model and the video frame can no longer be computed. In fact, the image regions that belong to different planes in the 3D scene create parallax and require different transformations and background models. This requirement is emphasized when the 3D scene contains significant depth variations. In order to address this limitation, several methods assume existence of a *dominant scene plane* with wide spatial coverage [18–22]. These so-called *plane + parallax* methods register images to the dominant plane and use the epipolar constraint [18,19], shape constancy constraint [20] or the structure consistency constraint [21] to resolve incorrect registrations due to parallax and object motion. Authors of [22] handle parallax by applying robust statistics [23] to motion estimation [24].

☆ This paper has been recommended for acceptance by Radu Horaud.

* Corresponding author. Address: 2015 Neil Ave. DL395, Columbus, OH 43210, United States.

E-mail addresses: zamalieva.1@osu.edu (D. Zamalieva), yilmaz.15@osu.edu (A. Yilmaz).

http://dx.doi.org/10.1016/j.cviu.2014.06.007
1077-3142/© 2014 Elsevier Inc. All rights reserved.

While plane + parallax methods perform well for scenes that can be approximated by a single plane, such as in aerial imagery, difficulties arise in cases when the scene contains more than one dominant plane. For such cases, researchers resort to estimating a set of physical planes in the scene [25–29]. In context of background subtraction, Jin et al. [30] adopt the multi-plane representation and generate different background models for each scene plane. Given a sequence of images, the scene planes and respective homography transformations are estimated by applying a cascade of RANSAC steps. This process assumes that the scene planes contain a high number of matching salient points[1] between consecutive video frames. All the planes selected through this process, however, may not necessarily provide full spatial coverage of the scene. In addition, selecting multiple scene planes through a cascaded RANSAC process practically results in non-existing scene planes leading to incorrect background models. In order to address this shortcoming, Monnet et al. [31] formulate the problem of background subtraction as the complement of saliency detection, such that, background model becomes insensitive to the camera motion due to its locally discriminant nature. A recent study by Lim et al. [32] proposes an online iterative algorithm where the geometry of the scene is approximated by dividing the image into multiple blocks and estimating the background/foreground motion of each block using optical flow. As observed in experimental results, the background labeling in [32], however, is prone to complex scene geometry and small objects that inhibit the planar scene assumption for each the block. Similarly, Kwak et al. [33] use the same block-wise approach but adopt a non-parametric belief propagation with Gaussian mixtures for motion estimation.

Alternative to multi-layer representations, Sheikh et al. [34] applied factorization based shape from motion to a set of tracked points. They labeled the trajectories as background or foreground based on the assumption that background trajectories form a 3D subspace. Their approach, however, requires offline processing and generates a sparse model due to disjoint trajectories. A recent method proposed by Elqursh et al. [35] represents trajectories in a low-dimensional space and groups them by relearning the Gaussian Mixture Model at each frame. The decision of which trajectory groups belong to background or foreground is given by a set of heuristics such as compactness, surroundedness, and spatial closeness, which may not always hold. Another method for moving object detection with hand held cameras was proposed in [36]. The authors combine color and locality cues to detect moving objects when the object motion is sparse and insufficient. This method assumes that the global background motion can be modeled by a single homography transformation. Existence of a single homography transform, however, does not hold when the camera translation is significant for a complex scene. Zhang et al. [37] improved the single homography assumption by introducing a bilayer segmentation approach for handheld cameras. Their approach, however, requires a manual preprocessing step to indicate the foreground layer each time a new foreground object enters the scene. Background from moving cameras can also be estimated by video grounding that removes the moving objects and reconstructs the occluded parts of the background in a sequence of images. In the case of video grounding using moving cameras, Evangelidis et al. [38] exploit information obtained from multiple image sequences capturing the same scene and use a set of background/foreground classifiers with temporal coherence constraints to estimate the background.

In this paper, we adopt the idea of multi-layer representation for detecting moving objects when the camera moves freely in the space. Unlike [30], we do not select a number of scene planes. Instead, we select a single plane with the highest number of salient points as the *reference plane*, and generate a set of *hypothetical planes* that *slice* the 3D scene creating object cross-sections (see Fig. 1). This treatment overcomes the aforementioned problems of plane selection for multi-layer approaches. In plane-sweep stereo, researchers have used such hypothetical planes for projective 3D recovery [39] and surface visibility tests [40] in context of volumetric scene recovery. For the background subtraction problem, we conjecture that all pixels in an image are projections of 3D points lying on respective hypothetical planes. This conjecture suggests that given sufficient number of planes, pixels that are members of the static scene only register to the images of the hypothetical planes they belong. These memberships generate a 2.5D background model which contains color statistics for each hypothetical plane (see Fig. 2). The *residual pixels* that do not satisfy the statistics of any hypothetical plane belong to the moving objects. In our approach, a set of points lying on the same physical plane may be associated with a number of hypothetical planes. This property provides the flexibility to generate background models for nonplanar static scene structures, which is otherwise impossible for typical multi-layer based scene models. In summary, the main contributions of the paper include: (1) ability to detect moving objects for freely moving cameras; (2) generating 2.5D background model for a generic scene containing nonplanar structures; and (3) implicit disambiguation of scene structures with high parallax in consecutive video frames based on the scene geometry.

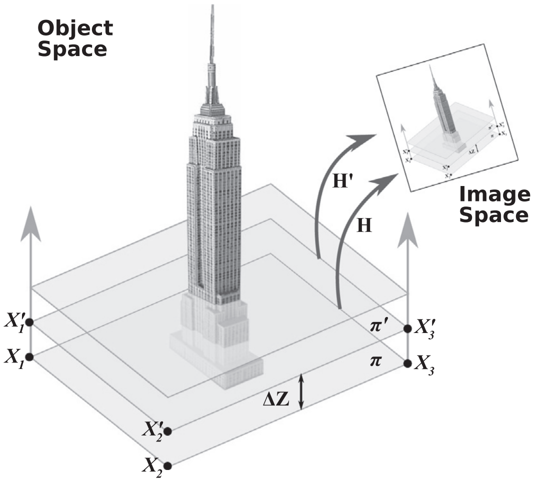


Fig. 1. A set of parallel hypothetical planes cross-sectioning the scene and 3D object. The images of points $\mathbf{X}_i$ are used to model 2.5D scene by estimating homography transformations across the images.

The remainder of the paper is organized as follows. In the next section, we elaborate on image transformations induced by different types of camera motion and introduce the plane topology used to model the scene. Based on this topology, the details of proposed background subtraction method are sketched in Section 3. Section 4 provides the experimental evaluations and discussions. Finally, we conclude the paper and provide future directions in Section 5.

2. Camera motion and scene geometry

Pinhole camera provides a simple yet powerful model that projects the 3D object space to 2D image plane. The transformation is algebraically governed by the perspective projection in the homogenous coordinates and is given by:

$$s\mathbf{x} = \mathtt{P}\mathbf{X} = \mathtt{KR}[\mathtt{I}\,|-\mathbf{C}]\mathbf{X}, \tag{1}$$

[1] We refer to images of static scene points as salient points.

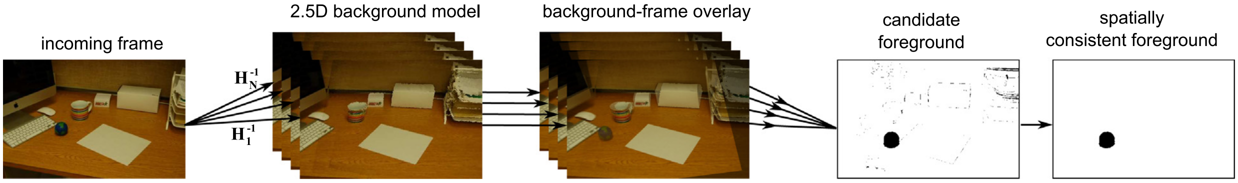


Fig. 2. Overview of the proposed method. Incoming frame is projected to each layer of a 2.5D background model by using the corresponding transformation $\mathtt{H}_k^{-1}$. The layers contain color statistics of the corresponding hypothetical plane and are illustrated by the mean values of the highest weighted component in the Gaussian mixture. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where a 3D point $\mathbf{X} = (X, Y, Z, 1)^\top$ maps to a 2D point $\mathbf{x} = (x, y, 1)^\top$ with scale s. Based on central projection, the 3×4 projection matrix $\mathtt{P}$ in (1) can be decomposed into the camera calibration matrix $\mathtt{K}$, camera rotation matrix $\mathtt{R}$ and the optical center vector $\mathbf{C}$. Let two images $\mathbf{x}$ and $\mathbf{x}'$ of a 3D point $\mathbf{X}$ be acquired at consecutive time instants, as the camera rotates, translates and zooms in on objects. For relative camera motion, the initial camera center can be assumed to reside at the origin and have its principle axis aligned with the z-axis of the coordinate system, such that $s\mathbf{x} = \mathtt{K}[\mathtt{I}|\mathbf{0}]\mathbf{X}$ and $s'\mathbf{x}' = \mathtt{K}'[\mathtt{R}|\mathbf{t}]\mathbf{X}$ are the projections for two consecutive cameras. In the following discussion, we will analyze different camera motions to understand its effect on the static scene in context of generating background models.

(1) **Stationary camera:** When camera is stationary, $\mathtt{R} = \mathtt{I}$ and $\mathbf{t} = \mathbf{0}$, with a fixed focal length, the projection matrices between consecutive frames become identical: $\mathtt{P} = \mathtt{P}'$. In this setting, a 3D point projects to the same pixel in both images, $s\mathbf{x} = \mathtt{K}\mathtt{K}^{-1}s'\mathbf{x}'$, such that the transformation between the two images simplifies to identity matrix $\mathbf{x}' = \mathtt{I}\mathbf{x}$ due to the constant scale. A desired outcome of the identity transformation is that despite the complexity of the scene geometry, one can directly use the color observations at a particular pixel for generating/updating the background model without image registration.

(2) **Pan-tilt-zoom camera:** Pan-tilt-zoom (PTZ) camera is commonly used in moving object detection. This camera undergoes only rotational changes in its external configuration due to the pan/tilt motion, and focal length changes due to zooming. Since the camera does not translate, projections from 3D to images become $s\mathbf{x} = \mathtt{K}\mathbf{X}$ and $s'\mathbf{x}' = \mathtt{K}'\mathtt{R}\mathbf{X}$. These projections result in direct image registration $s\mathbf{x} = s'\mathtt{K}\mathtt{R}^\top \mathtt{K}'^{-1}\mathbf{x}'$ by means of rotational homography [41, p. 326], such that $s\mathbf{x} = \mathtt{H}_\mathtt{R}\mathbf{x}'$. Disregarding the complexity of the scene structure, rotational homography provides a one-to-one mapping between the images which can be used to generate panoramic background models. The added complexity of this configuration compared to the stationary camera is the additional step for detecting and matching/tracking a minimum of four salient points across frames.

(3) **Free camera motion + planar scene:** When the scene is assumed to be planar, such as in aerial imagery [15], 3D points can be conjectured to lie on a ground plane π. Without loss of generality, selecting π as $Z = 0$ plane, using the projection matrix $\mathtt{P} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4]$ in (1), points $\mathbf{X}^\pi = (X, Y, 1)^\top$ on π are projected to image by $s\mathbf{x} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4]\ \mathbf{X}^\pi = \mathtt{H}(\mathbf{X}^\pi)$, where s is the scale factor and $\mathtt{H}$ is the homography transform between the scene plane and the image. The registration between two frames with respect to the plane π becomes $s\mathbf{x} = \mathtt{H}\mathbf{X}^\pi = \mathtt{H}\mathtt{H}'^{-1}\mathbf{x}' = (\mathtt{H}_\pi)\mathbf{x}'$, where $\mathtt{H}_\pi$, similar to $\mathtt{H}_\mathtt{R}$, is a homography transform. The complexity of this configuration is the same as in PTZ camera case and it requires detection and matching of salient points for registration of images to detect moving objects. This approach, however, will not work for images where the scene parallax causes registration errors.

(4) **Free camera motion + complex scene geometry (2.5D Representation):** In this configuration, we conjecture that the 3D scene can be decomposed into a stack of N "hypothetical planes" slicing the scene into a set of disjoint 2D subspaces. Considering that the choice of the coordinate system in Euclidean geometry is arbitrary, we let a reference plane π_0 generating the hypothetical planes coincide with the $Z = 0$ plane: $\pi_0 = [0, 0, 1, 0]^\top$. Note that the reference plane does not need to be a physical scene plane. Translating the reference plane by $k\Delta Z$ in Z direction generates a parallel hypothetical plane $\pi_k = [0, 0, 1, -k\Delta Z]^\top : k \in \mathbb{N}$ with a pencil at infinity (see Fig. 3). The points $\mathbf{X}_{i,0}$ lying on the reference plane π_0 can be transferred to the hypothetical plane π_k by a 4×4 homography matrix:

$$\mathbf{X}_{i,k} = \begin{bmatrix} \mathtt{I} & (k\Delta Z)\vec{\mathbf{z}} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{X}_{i,0} = \mathtt{H}_\mathtt{T}\mathbf{X}_{i,0}, \tag{2}$$

where $\mathtt{I}$ is 3×3 identity matrix, $\mathtt{H}_\mathtt{T}$ is the 3D translation, $\vec{\mathbf{z}}$ is the z-axis direction vector and $\mathbf{X}_{i,k}^\top \pi_k = 0$. The value of ΔZ defines the granularity of the 2.5D representation, and varying k identifies a hypothetical plane with the corresponding set of points $\{\mathbf{X}_{i,k}\}$. These transformed points do not necessarily correspond to physical scene features, however, they project to pixels $\{\mathbf{x}_{i,k}\}$ with scale factors $\{s_{i,k}\}$ in the image by the following projection:

$$s_{i,k}\mathbf{x}_{i,k} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4] \begin{pmatrix} X_{i,0} \\ Y_{i,0} \\ 1 \end{pmatrix} + (k\Delta Z)\mathbf{p}_3, \tag{3}$$

where $\mathbf{p}_3$ when homogenized becomes the vanishing point in the normal direction of π_0 [41, p. 159], and will be referred to as $\mathbf{v}_z$ in the remainder of the paper. The relation in (3) can be reorganized into:

$$s_{i,k}\mathbf{x}_{i,k} = s_{i,0}\mathbf{x}_{i,0} + \gamma(k\Delta Z)\mathbf{v}_z, \tag{4}$$

where $\mathbf{x}_{i,0}$ is the point on π_0, and $\mathbf{x}_{i,k}$ is the point on π_k in the same image, $s_{i,0}$ and $s_{i,k}$ are the respective scale factors, and γ is the scale factor of $\mathbf{v}_z$. Since the canonical forms of homogenous pixel

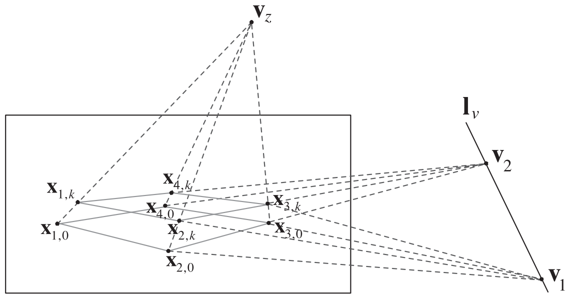


Fig. 3. The vanishing line $\mathbf{l}_v$ is uniquely defined by vanishing points $\mathbf{v}_1$ and $\mathbf{v}_2$. In this image, scene points corresponding to pixels $\mathbf{x}_{1,0}$ to $\mathbf{x}_{4,0}$ and $\mathbf{x}_{1,k}$ to $\mathbf{x}_{4,k}$ belong to plane π_0 and π_k respectively. The intersection of vertical parallel lines specify a vertical vanishing point $\mathbf{v}_z$.

coordinates have their third component set to 1, the scale factor $s_{i,k}$ can be computed as $s_{i,k} = s_{i,0} + \gamma(k\Delta Z)$. Note that γ is constant for all points, and can be combined with ΔZ, i.e. $\Delta\widetilde{Z} = \gamma\Delta Z$. Introducing this relation in (4) results in:

$$s_{i,0} = \frac{k\Delta\widetilde{Z}(\mathbf{x}_{i,k} - \mathbf{x}_{i,0})^\top(\mathbf{v}_z - \mathbf{x}_{i,k})}{(\mathbf{x}_{i,k} - \mathbf{x}_{i,0})^\top(\mathbf{x}_{i,k} - \mathbf{x}_{i,0})}, \tag{5}$$

which implies that the scale $s_{i,0}$ of a point $\mathbf{x}_{i,0}$ can be estimated from a point $\mathbf{x}_{i,k}$ corresponding to an identifiable physical feature. The scale factors for the remaining points coplanar with $\mathbf{x}_i$ can be computed using earlier result in [39], which suggests that the ratio between their scales is equal to the inverse ratio of their distances to the vanishing line $\mathbf{l}_v$ of the plane:

$$s_i = s_j \frac{\mathbf{x}_j^\top \mathbf{l}_v}{\mathbf{x}_i^\top \mathbf{l}_v}. \tag{6}$$

Considering that there exists a single vanishing line for the hypothetical planes generated by translating π_0, estimating the images of points for these non-existing planes becomes the problem of estimating pixel scales using (5) and (6). Using these points, the image registration with respect to any hypothetical plane is achieved by computing its respective homography transform:

$$s_{i,k}\mathbf{x}'_{i,k} = \mathtt{H}_k\mathbf{x}_{i,k}, \tag{7}$$

where $\mathtt{H}_0, \mathtt{H}_1, \ldots, \mathtt{H}_{N-1}$ are transformations providing one-to-one mappings between consecutive images for each plane $\pi_0, \pi_1, \ldots, \pi_{N-1}$.

The multi-layer scene model given above requires the vanishing line $\mathbf{l}_v$ and the vertical vanishing point $\mathbf{v}_z$. Vanishing lines can be estimated using different approaches. One of these approaches is to use cascaded Hough transform to first estimate vanishing points such that their join becomes the vanishing line [42] (see Fig. 3). Alternatively, vanishing line can be estimated by computing the relative camera motion between consecutive frames. In our case, since we use the ground plane as the reference plane, the camera matrices of the first few frames can be estimated using [43]. Note that the vanishing line needs to be estimated only for the first frame. Using the dual of the homography in (7), the vanishing line estimated initially simply transfers to the following frame by

$$\mathbf{l}'_v = \mathtt{H}_0^{-\top}\mathbf{l}_v, \tag{8}$$

where $\mathtt{H}_0$ is the homography transform in (7) for plane π_0.

Given the vanishing line $\mathbf{l}_v$ of the plane, the vanishing point $\mathbf{v}_z$ is computed in closed form using the topology illustrated in Fig. 4, where **CC** and **PP** are the camera center and the principal point, respectively. This geometry provides the vertical point $\mathbf{x}_{l_v}$ on $\mathbf{l}_v$, such that $\mathbf{x}_{l_v} = \mathbf{l}_v \times \mathbf{l}_v^\perp$, where $\mathbf{l}_v^\perp$ is the line passing through **PP** and perpendicular to $\mathbf{l}_v$. Using this relation $\mathbf{v}_z$ is calculated as:

$$\mathbf{v}_z = \mathbf{PP} + (\mathbf{PP} - \mathbf{x}_{l_v})b/a, \tag{9}$$

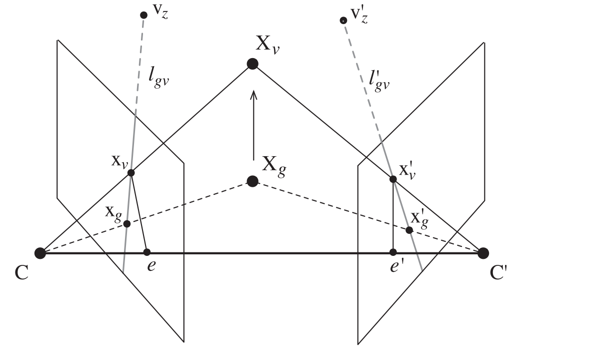


Fig. 5. The geometry of the vertical points $\mathbf{x}_v$ and $\mathbf{x}'_v$. The point $\mathbf{x}'_v$ is estimated as the intersection of line $\mathbf{l}_{gv}$ and the epipolar line $\mathtt{F}\mathbf{x}_v$.

where

$$\mathbf{a} = ||\mathbf{x}_{l_v} - \mathbf{PP}||, \quad \text{and} \quad \mathbf{b} = \mathbf{f}^2/\mathbf{a}. \tag{10}$$

In order to generate hypothetical planes, we use $\mathbf{x}_g$, which is the image of an arbitrary salient point $\mathbf{X}_{i,0}$, and $\mathbf{x}_v$, which is the image of the corresponding point $\mathbf{X}_{i,k}$. The plane π_k is selected by setting an arbitrary projective scale of a point $\mathbf{x}_v$ in the first frame lying on the line connecting the vanishing point $\mathbf{v}_z$ to any point $\mathbf{x}_g$ on the reference plane π_0. In our experiments, we used 100 as the reference projective scale. In the next frame, the vanishing line $\mathbf{l}'_v$ is estimated using (8) and $\mathbf{v}'_z$ is calculated from $\mathbf{l}'_v$ using (9). The ground point $\mathbf{x}_g$ is transformed using the ground homography $\mathtt{H}_0$ to find $\mathbf{x}'_g$. Based on the geometry illustrated in Fig. 5, the vertical point $\mathbf{x}'_v$ can be estimated as the intersection of the of line $\mathbf{l}'_{gv}$ and the epipolar line $\mathtt{F}\mathbf{x}_v$, such that

$$\mathbf{x}'_v = (\mathbf{v}'_z \times \mathbf{x}'_g) \times \mathtt{F}\mathbf{x}_v, \tag{11}$$

where $\mathtt{F}$ is the fundamental matrix between consecutive frames.

3. View-geometric background model

Our objective is to estimate a binary labeling $\mathcal{L}^* = \{l_1, l_2, \ldots, l_n\}$, which denotes if the pixel belongs to the background or the foreground. Motivated by the MAP-MRF formulation outlined in [44,45], the labeling can be achieved by finding $\mathcal{L}^* = \text{argmin}_{\mathcal{L}} E(\mathcal{L}, \mathbf{x})$, which minimizes the energy function given by:

$$E(\mathcal{L}, \mathbf{x}) = \sum_{\mathbf{x}_i \in I} D(\mathbf{x}_i) + \sum_{\mathbf{x}_i, \mathbf{x}_j \in N(I)} V(\mathbf{x}_i, \mathbf{x}_j). \tag{12}$$

In this equation, $D(\mathbf{x}_i)$ and $V(\mathbf{x}_i, \mathbf{x}_j)$ respectively represent the data and smoothness terms in local neighborhood $N(\cdot)$. The data term can be written as:

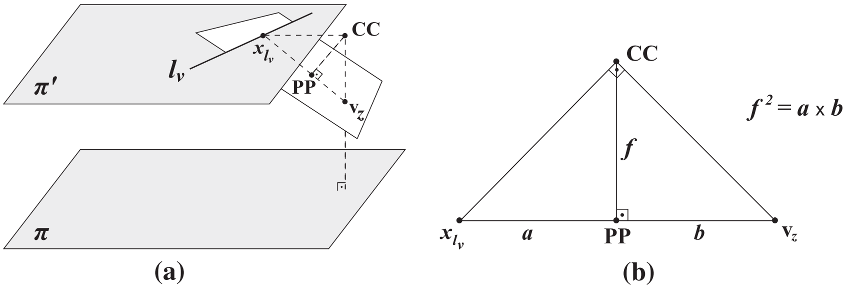


Fig. 4. (a) Geometry of the vanishing line and vanishing point and (b) detailed geometry formed from the principal point projective center and a point on vanishing line.

$$D(\mathbf{x}_i) = -\ln p(l_i|\mathbf{x}_i) = \begin{cases} -\ln p(b|\mathbf{x}_i) & \text{if } l_i = 0 \\ \eta_f & \text{if } l_i = 1 \end{cases},$$

where $p(b|\mathbf{x}_i)$ is the probability of $\mathbf{x}_i$ being a background pixel, η_f is the parameter determining the cost of labeling the pixel $\mathbf{x}_i$ as foreground and $l_i \in \{0, 1\}$ represents the label for background (0) and foreground (1).

We compute the probability $p(b|\mathbf{x}_i^t)$ by representing the background scene as a stack of hypothetical planes as described in Section 2. With this representation, points in the scene belong to one of the hypothetical planes. Moreover, points belonging to the static scene structures register between consecutive frames with one of the homography transforms $\mathtt{H}_0, \mathtt{H}_1, \ldots, \mathtt{H}_{N-1}$. Let $\mathbf{X}_i$ be a 3D point and $\mathbf{x}_i^t, \mathbf{x}_i^{t-1}$ be its images in frames I^t and I^{t-1}, respectively. We compute N possible registrations $\mathbf{x}_{i,0}^{t-1}, \mathbf{x}_{i,1}^{t-1}, \ldots, \mathbf{x}_{i,N-1}^{t-1}$ of $\mathbf{x}_i^t$ with respect to each plane:

$$s_{i,k}\mathbf{x}_{i,k}^{t-1} = (\mathtt{H}_k^t)^{-1}\mathbf{x}_i^t, \qquad (13)$$

where $\mathtt{H}_k$ is the homography transform in (7), and the subscript k in $\mathbf{x}_{i,k}^{t-1}$ indicates that $\mathbf{x}_i^t$ is transformed with $\mathtt{H}_k^t$. If $\mathbf{X}_i$ belongs to a plane π_m, we expect $\mathbf{x}_{i,m}^{t-1} = \mathbf{x}_i^{t-1}$ if $\mathbf{X}_i$ belongs to a static scene, and $\mathbf{x}_{i,m}^{t-1} \neq \mathbf{x}_i^{t-1}$ if $\mathbf{X}_i$ belongs to a foreground region. As a result, background points will result in *consistent* mappings across the frames with respect to one of the hypothetical planes (see Fig. 6), while the foreground pixels will result in *inconsistent* mappings. We exploit this fact to distinguish background and foreground.

Finding the plane on which $\mathbf{X}_i$ lies is not trivial. We eliminate the requirement of implicit estimation of the plane on which a point resides by maintaining a background model for each hypothetical plane π_k as if all points in the image belongs to that plane. With this setting, a pixel $\mathbf{x}_i^t$ can be labeled as background by applying maximum a posteriori probability (MAP) estimate, such that the probability of $\mathbf{x}_i^t$ being a background pixel, $p(b|\mathbf{x}_i^t)$ is computed by:

$$p(b|\mathbf{x}_i^t) = \max_{0\leqslant k<N} p(b|\mathbf{x}_{i,k}^{t-1}, I^t(\mathbf{x}_i^t)), \qquad (14)$$

where $p(b|\mathbf{x}_{i,k}^{t-1}, I^t(\mathbf{x}_i^t))$ is the conditional probability of $\mathbf{x}_i^t$ being a background pixel with respect to plane π_k given the corresponding position $\mathbf{x}_{i,k}^{t-1}$ computed using (13) and color value $I^t(\mathbf{x}_i^t)$ of the pixel. Recall that the background model for each hypothetical plane π_k models all pixels assuming that they belong to this plane. When a physical static scene point $\mathbf{X}_i$ resides on a plane π_m, this assumption, while not valid for most of the planes $\pi_{k\neq m}$, is guaranteed to hold for at least one plane $\pi_{k=m}$. The use of MAP estimate requires the presence of at least one distribution modeling $\mathbf{X}_i$ correctly. Applying the Bayes' rule, the expression $p(b|\mathbf{x}_{i,k}^{t-1}, I^t(\mathbf{x}_i^t))$ in (14) becomes

$$p(b|\mathbf{x}_{i,k}^{t-1}, I^t(\mathbf{x}_i^t)) = c \cdot \underbrace{p(I^t(\mathbf{x}_i^t)|\mathbf{x}_{i,k}^{t-1}, b)}_{\text{appearance}} \underbrace{p(\mathbf{x}_{i,k}^{t-1}|b)}_{\text{geometry}}, \qquad (15)$$

where $c = p(b)p(\mathbf{x}_{i,k}^{t-1}, I^t(\mathbf{x}_i^t))^{-1}$ is constant due to the fact that $p(b)$ and $p(\mathbf{x}_{i,k}^{t-1}, I^t(\mathbf{x}_i^t))$ are drawn from uniform distributions. The first probability terms in (15) are computed from the *appearance* and *geometry* distribution introduced in Sections 3.1 and 3.2, respectively, and provide the likelihood of the observation to be a member of that distribution.

In order to enforce the spatial smoothness during the labeling process, the smoothness term can be formulated as [46]:

$$V(\mathbf{x}_i, \mathbf{x}_j) = \lambda(1 - \delta(l_i - l_j)) \exp\left(\frac{-||I(\mathbf{x}_i) - I(\mathbf{x}_j)||t^2}{2\beta}\right), \qquad (16)$$

where λ controls the effect of the smoothness term, $\delta(\cdot)$ is a Kronecker delta function. As suggested by [32], the constant β represents the average of intensity variations in the neighborhood of each point and is defined as:

$$\beta = \frac{1}{n}\sum_{i=1}^{n} \sum_{\mathbf{x}_j \in G(\mathbf{x}_i)} ||I(\mathbf{x}_i) - I(\mathbf{x}_j)||^2, \qquad (17)$$

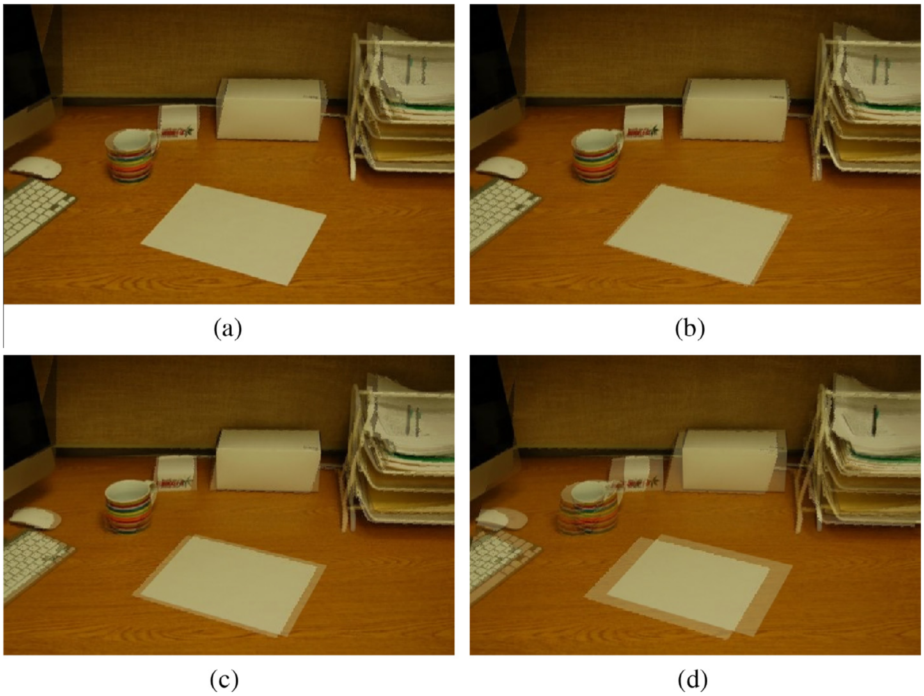


Fig. 6. Mean values of the most probable Gaussian of background mixture model distribution with respect to (a) reference plane π_0, (b)π_{17}, (c) π_{33} and (d) π_{72} overlaid with the actual frame.

where $G(\mathbf{x}_i)$ is a set of neighboring pixels around $\mathbf{x}_i$. The solution of the energy minimization can be efficiently computed using the graph-cut algorithm [47–49].

3.1. Modeling the appearance

Let $\mathbf{X}_i$ be a static scene point and $\mathbf{x}_i^t$ be its image at frame I^t, where $I^t(\mathbf{x}_i^t)$ is the appearance function at time t. In order to learn the underlying appearance distribution for $\mathbf{X}_i$, and compare an incoming observation to see whether it is drawn from this distribution, we need to know where $\mathbf{X}_i$ projects across images. This operation is not explicitly possible in case when the camera parameters are unknown. However, with our 2.5D formulation, the appearance of the background pixels can be modeled without explicit knowledge of which plane a point belongs to. We start with the assumption that $\mathbf{X}_i$ lies on all N hypothetical planes, such that at frame t, we have N different mappings $\mathbf{x}_{i,0}^{t-1}, \mathbf{x}_{i,1}^{t-1}, \ldots, \mathbf{x}_{i,N-1}^{t-1}$ of a pixel $\mathbf{x}_i^t$ to pixels in frame $t-1$, computed using the homography transforms $\mathtt{H}_k^t$ from image I^t to image I^{t-1} with respect to the plane $\pi_k : k = 0, \ldots, N-1$. Considering that our goal is to generate a scene model for moving object detection, this assumption is exercised by creating N appearance distributions $\{\Psi_{i,0}^t, \Psi_{i,2}^t, \ldots, \Psi_{i,N-1}^t\}$ using the mapping given in (13) (see Fig. 2). An appearance distribution $\Psi_{i,k}$ for a pixel $\mathbf{x}_i$ is maintained with the assumption that $\mathbf{x}_i$ lies on a plane π_k. In other words, by adding the new observations from incoming frames, $\Psi_{i,k}^t$ at frame t models distribution of values,

$$\{I^1(\mathbf{x}_{i,k}^1), I^2(\mathbf{x}_{i,k}^2), I^3(\mathbf{x}_{i,k}^3), \ldots, I^t(\mathbf{x}_{i,k}^t)\}, \tag{18}$$

where $\mathbf{x}_i$ is transformed with a corresponding homography transformation $\mathtt{H}_k^1, \mathtt{H}_k^2, \ldots, \mathtt{H}_k^t$.

In our implementation, we adopt the Gaussian mixture model of [2] as the distribution $\Psi_{i,k}$ for kth hypothetical plane, and estimate the posterior probability by:

$$p(I^t(\mathbf{x}_i^t)|\mathbf{x}_{i,k}^{t-1}, b) = \sum_{j=1}^{N_g} w_j \mathcal{N}(I^t(\mathbf{x}_i^t), \mu_j, \Sigma_j), \tag{19}$$

where b denotes background label, w_j, μ_j and $\Sigma_j = \sigma_j^2 \mathtt{I}$ are respectively the weight, mean and covariance of the jth Gaussian distribution in $\Psi_{i,k}^{t-1}$, and $\mathtt{I}$ is 3×3 identity matrix. A pixel value $I(\mathbf{x})$ is checked against each of the Gaussian components in Ψ until a match is found. A pixel matches a component if its value is within 2.5 standard deviations [2]. The parameters of a distribution that matches the current pixel are updated as:

$$\mu \leftarrow (1-\alpha)\mu + \alpha I(\mathbf{x}), \tag{20}$$

$$\sigma^2 \leftarrow (1-\alpha)\sigma^2 + \alpha(I(\mathbf{x}) - \mu)^\top (I(\mathbf{x}) - \mu), \tag{21}$$

where α is the learning rate. The mean and covariance of the unmatched distributions remain unchanged. The weights of the distributions are updated as follows:

$$w \leftarrow \begin{cases} (1-\alpha)w + \alpha & \text{if the distribution matches,} \\ (1-\alpha)w & \text{otherwise.} \end{cases} \tag{22}$$

If none of the N_g distributions match $I(\mathbf{x})$, the distribution with lowest weight is replaced by a new distribution with $\mu = I(\mathbf{x})$, low prior weight and initial high variance. As discussed in [2], the mixture model makes background appearance distribution multimodal and allows accounting for illumination changes, camera flicker, and repetitive motion.

The marginal updates performed by introducing observations to the model result in evolution of the background model as illustrated in Fig. 7 for the reference plane of the desk sequence. In this

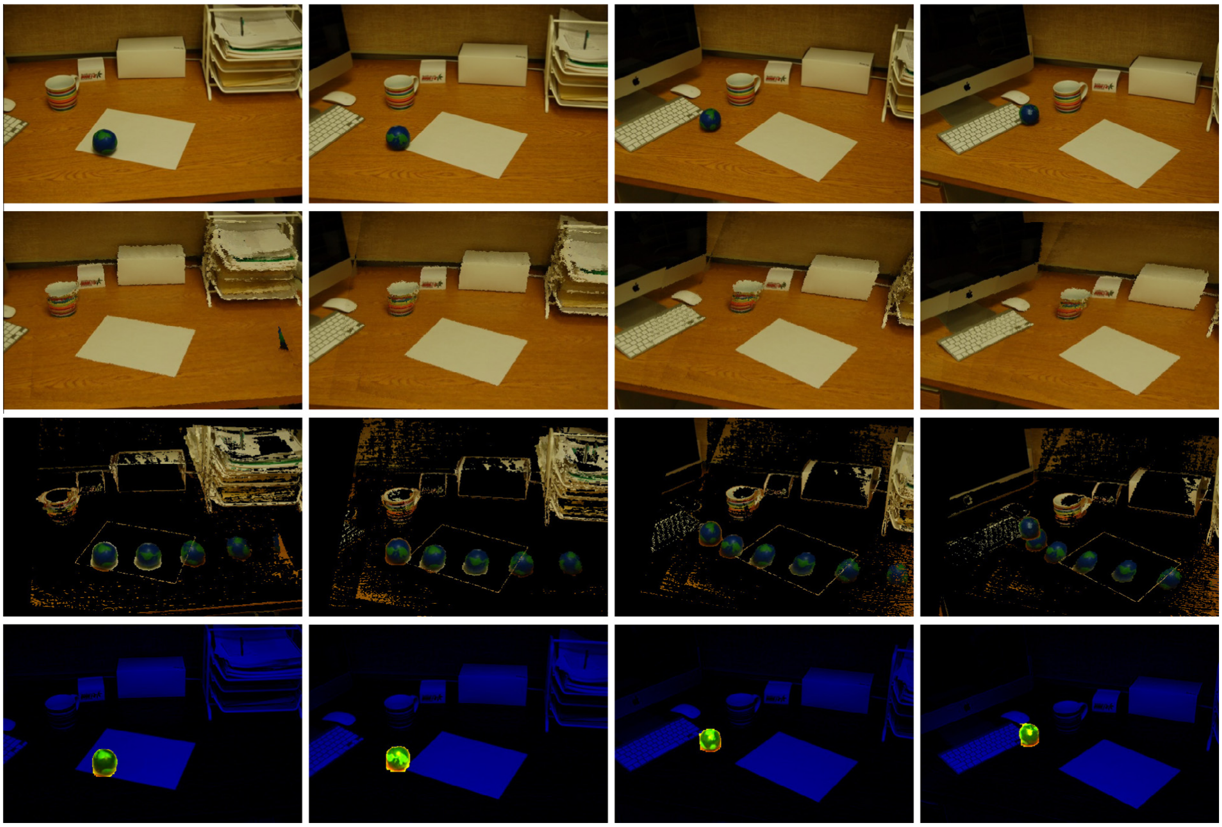


Fig. 7. Background model for consecutive frames in a sequence. The images are shown in the first row. The mean values of most probable and least probable Gaussian in the background model for the reference plane are shown in row 2 and 3, respectively. The extracted background (blue) and foreground (yellow) regions are shown in row 4. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

figure, the top of the desk and the bottom parts of all objects placed on it belong to the reference plane and are successfully modeled in the background distribution with highest weighted Gaussian component (Fig. 7, row 2). On the other hand, the other pixels that do not belong to the reference plane, e.g., the wall and the white box, are modeled with the lowest weighted Gaussian component.

For the stationary camera, since the transformations become identity matrices, the proposed background model reduces to the traditional background model with the added complexity of retaining the appearance models for each π_k. For the moving camera case, $\Psi_{i,k}$ uniquely models the appearance of the scene feature $\mathbf{X}_i$ with respect to the hypothetical plane π_k that it lies on. On the other hand, the appearance distributions for $\mathbf{X}_i$ with respect to the other hypothetical planes $\pi_{m\neq k}$ will model observations obtained due to wrong mappings between consecutive frames. In our model, moving objects change their positions from one frame to the next and their registration with respect to any of the hypothetical planes will cause misalignments and will not conform to learned appearance models. Fig. 6 presents the background model of the desk sequence for selected planes aligned with one of the frames. It can be noted that π_0 represents the top of the desk and bottom parts of the objects residing on it. The top of the small box is a part of plane π_{17}. Plane π_{33} correctly models the hypothetical plane containing top of the large box and the top part of the mug. Plane π_{72} corresponds to the plane passing through the pen and papers on the top shelf of the paper rack.

3.2. Geometric cues for labeling pixels

The second probability term in (15) relates to the proposed 2.5D background model composed of multiple background distributions due to projections onto multiple planes as shown in Fig. 8(a). Pixel $\mathbf{x}_i$ can be drawn from a distribution of hypothetical planes defining the background, which will be referred to as the *geometry distribution*.

If a physical scene point $\mathbf{X}_i$ resides on a plane π_m, its projection in images will be mapped correctly with the homography $\mathtt{H}_m$. This mapping consistently results in the same appearance observation, leading to a high value of $p(I(\mathbf{x}_i^t)|\mathbf{x}_{i,k=m}^{t-1}, b)$ and low values of $p(I(\mathbf{x}_i^t)|\mathbf{x}_{i,k\neq m}^{t-1}, b)$ in all the frames where the point $\mathbf{X}_i$ is not occluded by a foreground object. One can keep track of consistent mappings for the hypothetical planes by maintaining a $1 \times N$ vector Γ_i for each pixel $\mathbf{x}_i^t$. We start with a vector of $\Gamma^{t=0}$ of all zeros. Then, for each k, we calculate the probability $p(I^t(\mathbf{x}_i^t)|\mathbf{x}_{i,k}^{t-1}, b)$ using (19). If the probability is sufficiently high, Γ_i^t is updated as following:

$$\Gamma_i^t(x) \leftarrow \Gamma_i^t(x) + K_h(x-k), \tag{23}$$

where $x = 1, \ldots, N$ and $K_h(x) = \exp\left(-\frac{\|x\|^2}{2h^2}\right)$ denotes a kernel function with bandwidth parameter h. Intuitively, h should decrease as N decreases, since with fewer planes we expect less confusion in plane association. Motivated by the model update in [2], the construction of Γ_i^t is finalized by an exponential update based on the existing model Γ_i^{t-1}:

$$\Gamma_i^t \leftarrow (1-\alpha)\Gamma_i^{t-1} + \alpha\Gamma_i^t, \tag{24}$$

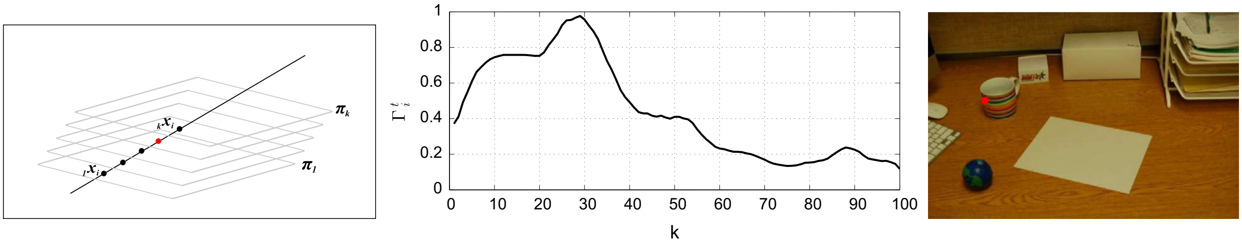
where α is the learning rate used for appearance model update in (20)–(22). The values of Γ_i^t are further normalized to be in the range $[0, 1]$. With this design, the second probability term $p(\mathbf{x}_{i,k}^{t-1}|b)$ in (15) is proportional to $\Gamma_i^t(k)$. In Fig. 8(b), we plot the geometric distribution as a function of the plane index for the pixel marked in red color in part (c). It can be observed that the probability is maximum for the plane that the pixel belongs to.

3.3. Algorithmic details

Algorithm 1. Background subtraction process.

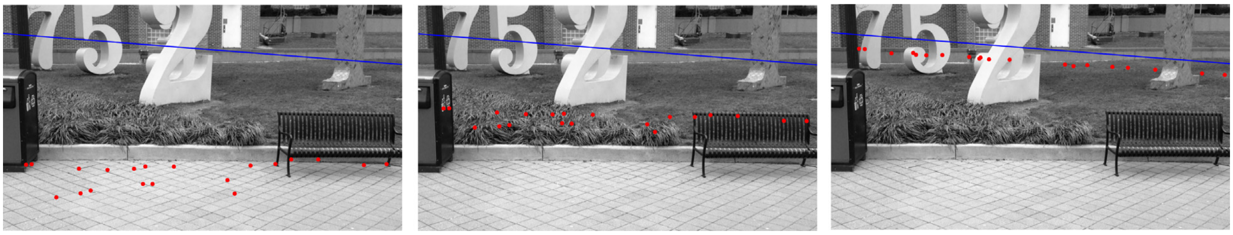
```
Data: frame sequence
Result: detection of moving objects
N: number of hypothetical planes;
n: number of points in $S_k^t$;
identify a set of points $S_0^1 = \{\mathbf{x}_{i,0}^1\}$, $|S_0^1| = n$ in $I^1$;
estimate $\mathbf{l}_v^1$ and $\mathbf{v}_z^1$;
find $S_k^1$ for each plane $\pi_k$ using (4);
for each new frame $I^t$ do
    track $S_0^{t-1}$ to construct $S_0^t$;
    estimate $\mathbf{l}_v^t$ and $\mathbf{v}_z^t$;
    for $k \leftarrow 0$ to $N-1$ do
        find $S_k^t$ using (4);
        estimate $\mathrm{H}_k^t$ using $S_k^{t-1}$ and $S_k^t$;
        for $i \leftarrow 1$ to $n$ do
            compute $\mathbf{x}_{i,k}^{t-1}$ using (13);
            compute $p(I^t(\mathbf{x}_i^t)|\mathbf{x}_{i,k}^{t-1}, b)$ using (19);
            $\Psi_{i,k}^t \leftarrow \Psi_{i,k}^{t-1}$ updated using $I^t(\mathbf{x}_i^t)$;
            update $\Gamma_i^t(k)$ using (23);
        end
        finalize $\Gamma_i^t$ using (24);
    end
    compute $p(b|\mathbf{x}_i^t)$ using (14);
end
```

The steps of the proposed method are detailed in Algorithm 1. First, a set of coplanar pixels S_0^1 is detected by robust homography estimation using RANSAC. In our experiments, we use dense optical flow [24] to establish point correspondences. For very low frame rates, point detection or tracking (for example, KLT tracking [50]) can also be used. To construct the 2.5D representation, we estimate the vanishing line and the vertical vanishing point for the selected plane using the procedure described in Section 2. With each new frame I^t, the tracked set of points is augmented by



(a) (b) (c)

Fig. 8. (a) Projections of a pixel onto background models corresponding to respective hypothetical planes. The projected pixel is only a member of one of these planes denoted by a red circle. (b) Geometry distribution as a function of hypothetical plane index for the pixel marked red in (c). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) π_0 (b) π_{20} (c) π_{40}

Fig. 9. (a) Tracked points on the ground plane. (b–c) The points computed using Eq. (4) with (b) $k = 20$ and (c) $k = 40$. The vanishing line $\mathbf{l}_v$ is shown with blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

introducing new points to the inlier set S_0^{t-1} which satisfy the homography transform H_0^t, allowing gradual change of the scene content. The points in S_0^t are used to obtain the sets S_k^t for each hypothetical plane π_k using (4). The corresponding points in S_k^{t-1} and S_k^t are used to estimate the homography H_k^t for the plane π_k.

Pixels $\mathbf{x}_i^t$ in a novel image are projected to each layer of the background model governed by N homography transforms as $\mathbf{x}_{i,0}^{t-1}, \mathbf{x}_{i,1}^{t-1}, \ldots, \mathbf{x}_{i,N-1}^{t-1}$ using (13). To eliminate occasional small errors in homography estimation, we perform a local search in a small window around each pixel, selecting the position which induces the highest probability for each plane π_k:

$$\tilde{\mathbf{x}}_{i,k} = \underset{\mathbf{x}_{j,k} \in N(\mathbf{x}_{i,k})}{\operatorname{argmax}} p(I(\mathbf{x}_i)|\mathbf{x}_{j,k}, b), \qquad (25)$$

where $\tilde{\mathbf{x}}_{i,k}$ is the corrected position of $\mathbf{x}_{i,k}$, and $N(\mathbf{x}_{i,k})$ is a set of pixels in the small window centered around $\mathbf{x}_{i,k}$. In our experiments, we use a 3 × 3 window. This formulation allows us to avoid erroneous updates of the background models, while the spatial smoothing using MRF affects the labeling output only.

The labeling likelihood is computed using (14). Some of the pixels $\mathbf{x}_{i,k}^{t-1}$ may not have corresponding appearance distributions $\Psi_{i,k}^{t-1}$. This usually happens because of new regions appearing on the boundaries of the image as the camera views unseen parts of the scene. For these regions, we start a new distribution $\Psi_{i,k}^t$ and initialize it with $\mu = I^t(\mathbf{x}_i^t)$, low prior weight and high initial variance values.

In order to ensure the temporal consistency of estimated probabilities, we compute the weighted average of the probabilities $p(b|\mathbf{x}_i^t), p(b|\mathbf{x}_i^{t-1})$ and $p(b|\mathbf{x}_i^{t-2})$ from the last three frames [51] using:

$$p(b|\mathbf{x}_i^t) = g_1 p(b|\mathbf{x}_i^t) + g_2 p(b|\mathbf{x}_i^{t-1}) + g_3 p(b|\mathbf{x}_i^{t-2}), \qquad (26)$$

where $(g_1,\ g_2,\ g_3)$ denote the weights. In our implementation, we used $(0.7, 0.2, 0.1)$ as probability weights which emphasizes the current frame more than the previous frames. Note that application of the temporal consistency on probabilities may not guarantee consistency of labels. This is primarily due to the fact that MRF, which is applied after temporal consistency, suppresses small objects.

For hypothetical planes that are close to the plane which passes through camera center, the points converge to the horizon line (see Fig. 9). As a result, the induced homographies become degenerate and do not provide mappings between the frames. Therefore, when the vanishing line $\mathbf{l}_v$ can be seen in the image, we ignore the image regions that are spatially close to $\mathbf{l}_v$, and assume that they belong to background. This assumption is not considered to be a drawback of the proposed algorithm since objects close to horizon are infinitesimally small.

4. Experiments

We test the performance of the proposed approach using sequences acquired both indoors and outdoors from moving cameras. To initialize the appearance distributions $\Psi_{i,k}^1$, we assume that all pixels in the first frame belong to background and set the parameters of one of the Gaussians $\Psi_{i,k}^1$ as $\mu_1 = I^1(\mathbf{x}_i^1)$, $\sigma_1^2 = 50, w_1 = 1.0$ and the rest as $w_j = 0.0$, where $j = 2, \ldots, N_g$ and $k = 0, 1, \ldots, N-1$. We set $N_g = 3$, which is considered a practical minimum [53]. The learning rate for updating the distributions is set to $\alpha = 0.05$. The cost of labeling a pixel as foreground is set to $\eta_f = -\ln(0.4)$, and $\lambda = 5$ is used for spatial smoothing. The neighborhood $N(\cdot)$ in (12) is a 4-neigborhood. All experiments are conducted on a PC with dual-core Intel i5 Ivy bridge 1.8 GHz CPU. The proposed method is implemented using a combination of unoptimized C++ and MATLAB code, and the execution time for a 400 × 700 video frame with 1, 5, 10, 25 and 50 hypothetical planes is measured as 0.49, 2.42, 4.05, 10.81 and 21.67 s, respectively. For comparison, the execution time of partial MATLAB and C++ implementations of state-of-the-art methods [34,32,30] is measured as 479.08, 124.02, and 36.99 s, respectively. The reported timings for all methods do not include the optical flow estimation and graph-cut processes, which are measured to take 50.07 s (MATLAB implementation [24]) and 1.54 s (C++ implementation [47]) per frame, respectively.

In contrast to the stationary camera case, there is no benchmark dataset for evaluating performances of background subtraction methods for moving cameras. Due to this unavailability, some studies do not provide quantitative comparisons [37,36]. In this paper, we use a set of sequences from the Hopkins dataset [54] (*people1–2*) and from [52] (*cars, person*) which have been used by recent quantitative papers on the topic [32–35]. We additionally include five challenging sequences (*bolz1, bolz2, oxley, garage, dreese, skating*) that reflect a real world setting acquired with a smartphone camera. The sequences include both camera rotation and translation, multiple moving objects, and gradual changes in the scene content. To account for the cases where the camera motion is very fast, we use *skating1* and *football* sequences [32]. In order to quantitatively judge the performance, we manually generate ground truth data by extracting the moving objects in all frames and measure the precision and recall for detected pixels versus ground truth pixels. The number of frames and short description for each sequence is presented in Table 1. Note that only first half of the frames for *garage* sequence is annotated, due to its length. The frames are extracted at 10 fps.

In order to analyze the effect of the number of hypothetical planes on detection performance, we generate the 2.5D background model with 2, 3, 4, 5, 10, 25 and 50 hypothetical planes. We also used different variations of the proposed approach (with $\lambda = 0$ and $\lambda > 0$) to give insight of how different steps change the performance. The results are shown in Table 2 for each variation. The first two rows in the table, for which $N = 1$, reflect the results for the baseline approach. We chose the baseline approach as the traditional background subtraction, where the consecutive frames are registered prior to background subtraction based on the ground plane. Different variations of our approach include (i) direct thres-

Table 1
Description of the image sequences.

Sequence	# Frames	Description
Bolz1	135	Camera rotation and translation complete change of the scene
Bolz2	141	Camera rotation and translation complete change of the scene multiple moving objects complex scene geometry
Oxley	101	Camera rotation and translation complex scene geometry
Garage	444	Camera rotation and translation complete change of the scene multiple moving objects complex scene geometry
Dreese	98	Camera rotation and translation multiple moving objects complex scene geometry
Cars [52]	81	Camera rotation and translation multiple moving objects
Person [52]	51	Camera rotation and translation
Skating1 [32]	103	Camera rotation and translation fast camera motion zooming
Football [32]	116	Camera rotation and translation fast camera motion zooming
People1 [54]	41	Camera rotation and translation
People2 [54]	30	Camera rotation and translation

Table 2
Performance of the baseline approach against different variations of the proposed approach. The table shows the precision (P) and recall (R) values for each sequence.

Sequence method	Bolz1		Bolz2		Oxley		Garage		Dreese		Cars		Person		Skating1		Football		People1		People2	
	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R
$N = 1$ w/o MRF	47	86	41	88	26	90	15	93	23	77	36	72	62	77	26	95	58	92	13	88	34	82
$N = 1$	69	85	59	91	78	92	29	92	48	67	69	74	78	85	46	88	77	81	39	90	58	88
$N = 2$ w/o MRF	53	85	43	88	30	90	18	92	26	77	46	72	63	77	34	94	58	92	19	86	43	80
$N = 2$	70	85	63	91	80	93	40	91	63	67	76	74	79	85	82	85	77	81	62	84	75	83
$N = 3$ w/o MRF	53	85	49	88	30	90	21	89	27	77	47	72	63	77	37	94	58	92	22	84	49	77
$N = 3$	70	85	71	91	81	92	48	86	64	67	76	74	79	85	88	83	77	81	75	82	80	78
$N = 4$ w/o MRF	57	84	62	80	36	87	22	89	27	77	48	72	64	77	39	94	58	92	22	84	49	77
$N = 4$	70	84	87	83	86	88	48	87	65	67	76	74	79	85	88	83	77	81	78	82	80	78
$N = 5$ w/o MRF	57	83	65	80	39	87	25	89	27	77	48	72	64	77	40	94	58	92	23	83	52	77
$N = 5$	70	82	87	84	86	88	77	87	65	67	76	74	79	85	89	83	77	81	79	81	81	78
$N = 10$ w/o MRF	59	83	77	77	39	87	27	89	38	73	50	72	67	74	40	94	58	92	25	83	55	76
$N = 10$	70	82	89	80	86	88	80	84	77	63	76	74	80	82	90	82	77	81	81	81	82	77
$N = 25$ w/o MRF	61	83	78	77	40	87	29	88	41	73	55	71	69	73	44	93	58	92	27	83	57	76
$N = 25$	70	82	89	80	86	88	80	82	82	63	81	73	81	80	91	81	77	81	83	80	82	76
$N = 50$ w/o MRF	61	83	80	77	41	86	30	87	47	69	56	71	69	73	45	93	58	92	28	82	59	75
$N = 50$	70	82	89	80	87	87	81	80	86	59	81	73	81	80	91	80	77	81	85	79	83	75

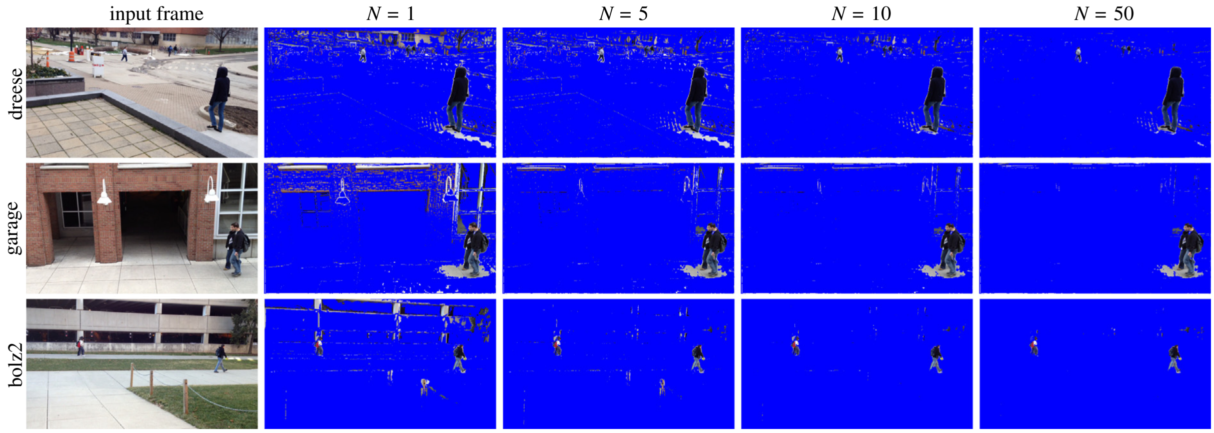


Fig. 10. Results of background subtraction for the proposed method without MRF for (row 1) *dreese*, (row 2) *garage* and (row 3) *bolz2* sequences. The columns correspond to (1) original image, (2) the baseline $N = 1$, (3) $N = 5$, (4) $N = 10$, (5) $N = 50$.

holding of the probabilities computed using (14) in Section 3.2 (or, the case where $\lambda = 0$) and (ii) our complete method with $\lambda = 5$.

The results in Table 2 reflect that when the scene can be well approximated by a single plane (*bolz1*, *person* and *football* sequences), the performance of the method is very similar for different numbers of hypothetical planes as well as for the baseline. However, for the scenes with complex geometry containing multiple physical planes (*bolz2*, *dreese* and *garage* sequences), the proposed method significantly outperforms the baseline approach. Note that with complex scene geometry, there is a constant increase in precision with larger number of hypothetical planes. This increase is especially apparent when MRF is not employed. Fig. 10 illustrates intermediate results for different number of hypothetical planes prior to applying MRF. Evidently, the results become more precise as the number of hypothetical planes increase. However, with the use of MRF, in most of the cases the performance for $N = 10$ and $N = 25$ is very close to that of $N = 50$. Therefore, we suggest that $N = 10$ is sufficient to model even complex scenes. Example qualitative results for the sequences with $N = 50$ with MRF are displayed in Fig. 12.

Another parameter that affects the performance of the proposed method is λ, which controls the spatial smoothing. Fig. 11 presents

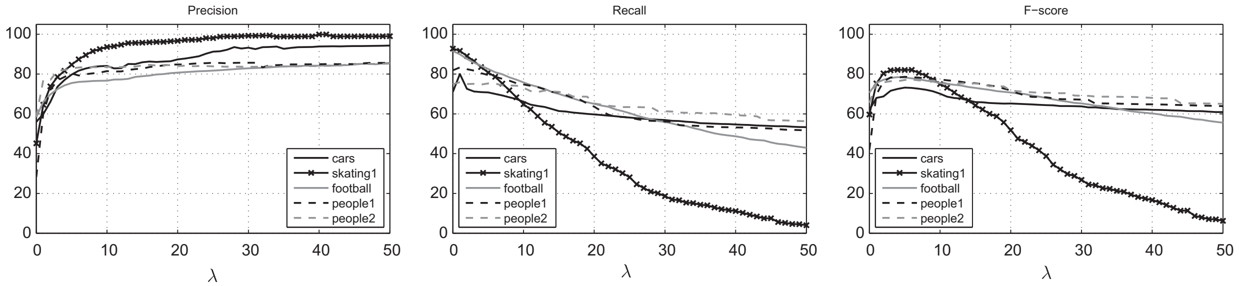


Fig. 11. The effect of the parameter λ on precision, recall and F-score values averaged over all frames.

the precision, recall and F-score values for varying λ for *cars, skating1, football, people1* and *people2* sequences. A similar trend is observed for the rest of the sequences. As expected, as λ increases, the precision increases and recall decreases (note that there may be slight increase in recall for the small values of λ, as the foreground may be filled in as a result of spatial smoothing). In our experiments, we set $\lambda = 5$, which results in relatively high precision without sacrificing recall. After MRF, we remove very small regions (less than 100 pixels), which further increases precision values (as a result, the values reported in Table 2 and Fig. 14 can differ from the ones in Fig. 11 for $\lambda = 5$).

We also compare the proposed method with the methods introduced in [34,32,30], as well as a simple motion segmentation approach with the epipolar constraint, which is a segmentation based on fundamental matrix estimated with RANSAC. For fair comparison, our implementation of [34] is based on dense optical flow estimation instead of using particle video approach introduced in [52]. The qualitative and quantitative results are presented in Figs. 13 and 14, respectively. In most of the cases, the proposed approach outperforms other methods. We observe that [34] is susceptible to the noise in trajectories around the moving objects and incomplete trajectories around image boundaries. This method often results in inconsistent detections due to the lack of appearance models and temporal constraints. The approach introduced in [32] employs fundamental matrix to infer camera and object motion, which further guides the propagation of the corresponding appearance models. As a result, their models become corrupted when the fundamental matrix estimation is unsuccessful for a few consecutive frames. We also observed that this method is highly dependent on the correct background/foreground initialization in the first frame. Note

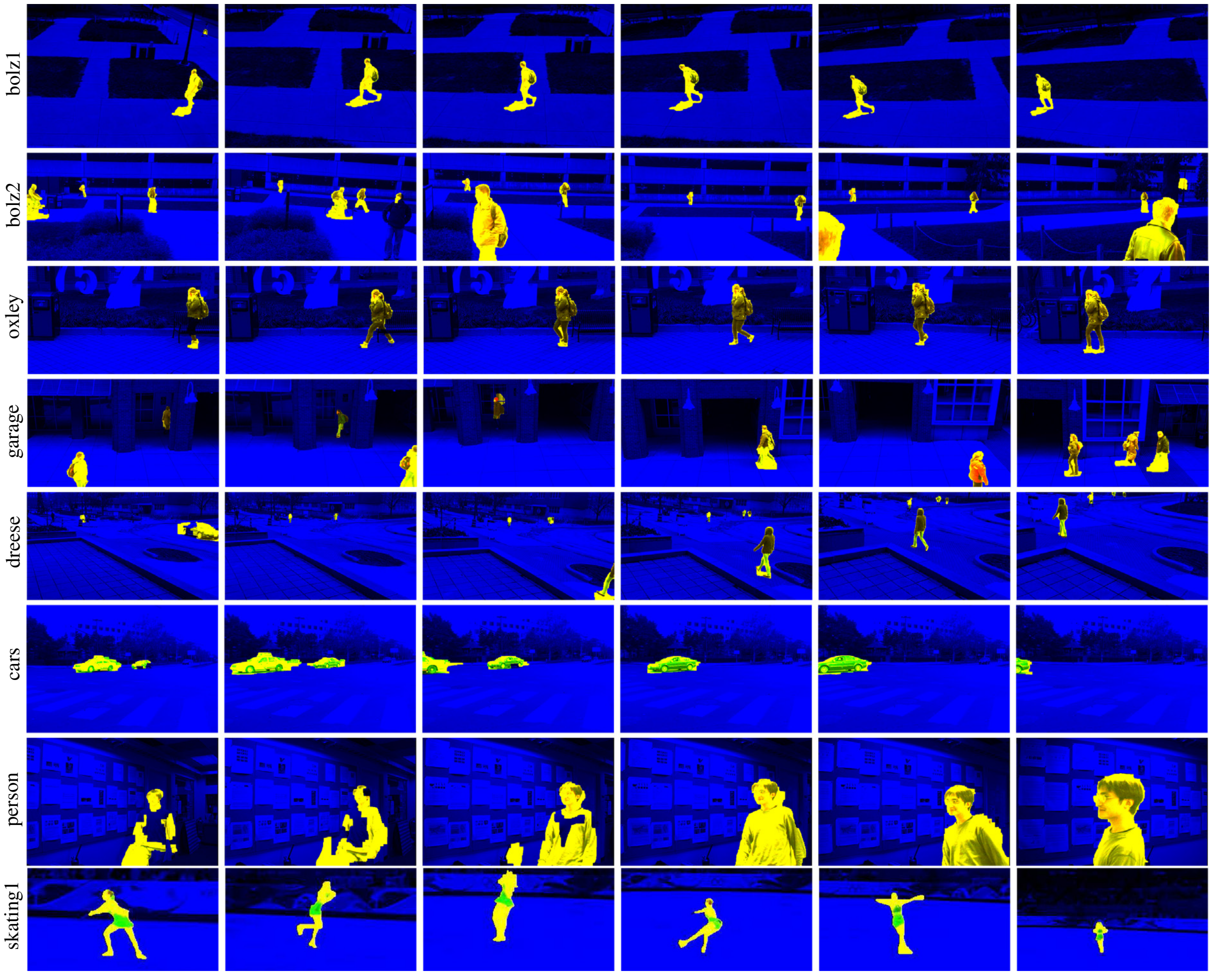


Fig. 12. Results of background subtraction with the proposed method for $N = 50$ with the application of MRF for (row 1) the *bolz1*, (row 2) *bolz2*, (row 3) *oxley*, (row 4) *garage*, (row 5) *dreese*, (row 6) *cars*, (row 7) *person* and (row 8) *skating1* sequences.

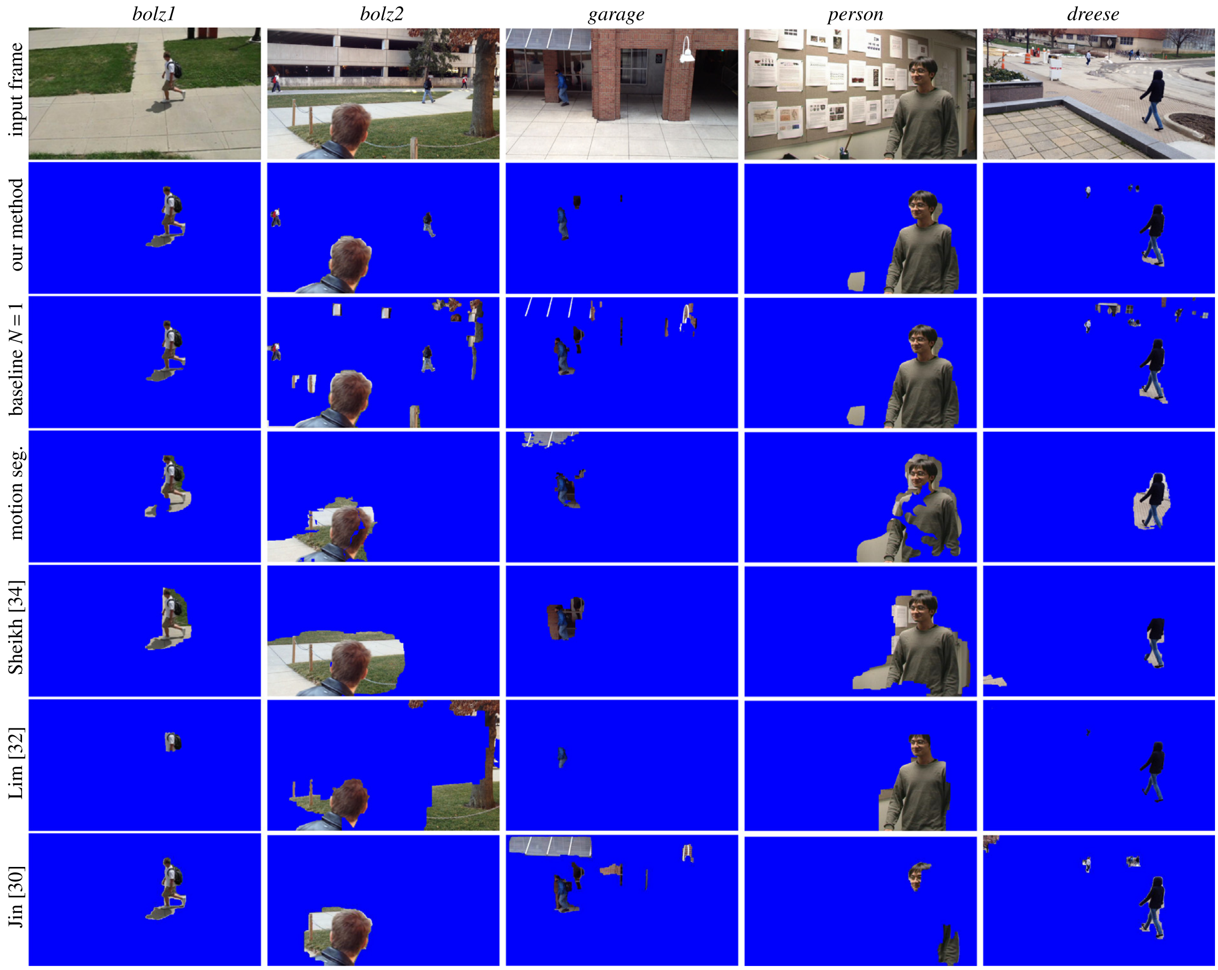


Fig. 13. Results of background subtraction for (row 1) the proposed method for $N = 50$ with MRF, (row 2) the baseline $N = 1$ with MRF, (row 3) the motion segmentation with epipolar constraint, (row 4) [34], (row 5) [32], and (row 6) [30]. The columns correspond to the *bolz1*, *bolz2*, *garage*, *person* and *dreese* sequences.

that authors of [32] also report that small objects are generally difficult to detect by their algorithm, which explains the low recall values for the *dreese* and *bolz2* sequences. The method introduced in [30] relies on cascaded homography estimations to detect multiple planes in the scene, and transforms the background model accordingly. The authors suggest to stop the algorithm when the remaining feature points used for homography estimation are fewer than 16. We notice that, unless the moving object is very small, it usually contains more than 16 interest points. With this formulation, a homography transform is estimated for the foreground as well, and it inevitably becomes part of background due to the nature of the mixture of Gaussians model, resulting in very low recall values. Instead of using 16 points as a threshold, we use 10% of total number of points. While it allows the detection of the moving objects, it also prevents the algorithm from learning a homography transform for small planes. Also note that, this method is still susceptible to large moving objects (for example, *person* sequence).

On the other hand, as it is typical in background subtraction papers, our method may fail to detect moving objects for a number of frames (depending on the speed of the object and camera) when the video first starts with a moving object in the field of view. An example of this issue can be observed in the *person* sequence shown in Fig. 12. Another limitation in our current implementation is that the unseen region, which is revealed as the camera moves, is introduced as part of the background to the model. This initialization results in labeling a first-time seen object on the boundary of the image as part of the background. Once the object moves, however, our algorithm detects it correctly as the foreground region. An example of this limitation is illustrated for the *bolz2* sequence in Fig. 12. While our approach uses a mixture of Gaussians [2], dynamic background regions may also be labeled as foreground. For this reason, the reflections, shadows and strong illumination changes are also labeled as foreground (see sequence *bolz1*, where the shadow is detected and *garage*, where the reflections are detected, in Fig. 12).

5. Conclusions and future directions

This paper introduces a 2.5D background model for detecting moving regions for moving cameras. The proposed approach introduces two distributions: one for modeling the appearance found in many background subtraction methods, and the other for modeling the geometry of the scene. The geometric model depends on the 2.5D representation which is generated from a stack of hypothetical 3D planes slicing the 3D scene and induce a set of homography transformations between the images. The pixels projected from 3D points lying on these planes consistently map across consecutive frames. This mapping provides distributions which are later used to classify pixels as background or foreground based on their deviation from the model. Experiments on a set of images show that the proposed algorithm can successfully detect moving objects in complex scenes and with significant camera motion when compared to recent literature.

The proposed approach can be further improved by maintaining a foreground model for each pixel and developing a suitable transformation model for frame-to-frame foreground model propaga-

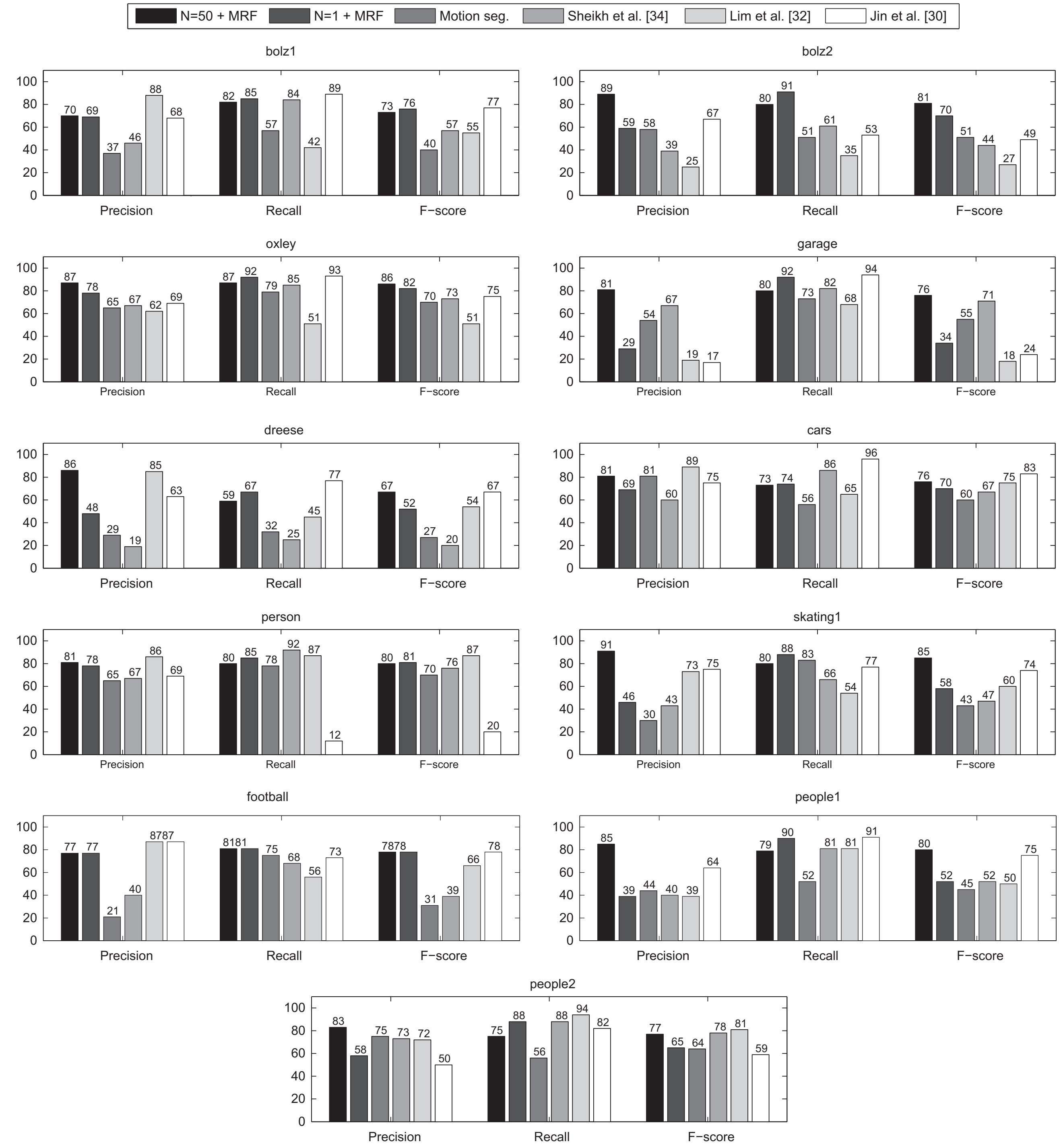


Fig. 14. Quantitative results. The precision, recall and *F*-score values are averaged over all frames.

tion. The foreground model can be further improved by preprocessing images for shadow removal as stated in [55,56]. In addition, motion clues can be used to decide whether new regions appearing in images due to camera motion belong to the foreground.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.cviu.2014.06.007.

References

[1] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland, Pfinder: real-time tracking of the human body, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1997) 780–785.

[2] C. Stauffer, W. Eric, W.E.L. Grimson, Learning patterns of activity using real-time tracking, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 747–757.

[3] A.M. Elgammal, D. Harwood, L.S. Davis, Non-parametric model for background subtraction, in: Proc. 6th European Conference on Computer Vision, 2000.

[4] Y. Sheikh, M. Shah, Bayesian modeling of dynamic scenes for object detection, IEEE Trans. Pattern Anal. Mach. Intell. 27 (11) (2005) 1778–1792.

[5] A. Mittal, N. Paragios, Motion-based background subtraction using adaptive kernel density estimation, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2004, pp. 302–309.

[6] R. Pless, J. Larson, S. Siebers, B. Westover, Evaluation of local models of dynamic backgrounds, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2003.

[7] T. Ko, D. Estrin, S. Soatto, Warping background subtraction, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2010.

[8] V. Mahadevan, N. Vasconcelos, Background subtraction in highly dynamic scenes, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[9] L. Maddalena, A. Petrosino, The 3dSOBS+ algorithm for moving object detection, Comput. Vis. Image Underst. 122 (2014) 65–73.

[10] S. Yoshinaga, A. Shimada, H. Nagahara, R. Taniguchi, Object detection based on spatiotemporal background models, Comput. Vis. Image Underst. 122 (2014) 84–91.

[11] M. Piccardi, Background subtraction techniques: a review, in: Proc. IEEE International Conference on Systems, Man and Cybernetics, vol. 4, 2004, pp. 3099–3104.

[12] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Comput. Surveys 38 (4) (2006).

[13] A. Sobral, A. Vacavant, A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos, Comput. Vis. Image Underst. 122 (2014) 4–21.

[14] Y. Ren, C.-S. Chua, Y.-K. Ho, Statistical background modeling for non-stationary camera, Pattern Recogn. Lett. 24 (1–3) (2003) 183–196.

[15] A. Mittal, D. Huttenlocher, Scene modeling for wide area surveillance and image synthesis, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2000.

[16] E. Hayman, J.O. Eklundh, Statistical background subtraction for a mobile observer, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2003.

[17] S. Rowe, A. Blake, Statistical mosaics for tracking, Image Vis. Comput. 14 (8) (1996) 549–564.

[18] M. Irani, B. Rousso, S. Peleg, Recovery of ego-motion using region alignment, IEEE Trans. Pattern Anal. Mach. Intell. 19 (3) (1997) 268–272.

[19] M. Irani, P. Anandan, A unified approach to moving object detection in 2d and 3d scenes, IEEE Trans. Pattern Anal. Mach. Intell. 20 (6) (1998) 577–589.

[20] H. Sawhney, Y. Guo, R. Kumar, Independent motion detection in 3d scenes, IEEE Trans. Pattern Anal. Mach. Intell. 22 (10) (2000) 1191–1199.

[21] C. Yuan, G. Medioni, J. Kang, I. Cohen, Detecting motion regions in presence of strong parallax from a moving camera by multi-view geometric constraints, IEEE Trans. Pattern Anal. Mach. Intell. 29 (9) (2007) 1627–1641.

[22] A. Strehl, J.K. Aggarwal, MODEEP: a motion-based object detection and pose estimation method for airborne FLIR sequences, Mach. Vis. Appl. 11 (6) (2000) 267–276.

[23] P.J. Huber, Robust Statistics, Wiley, New York, NY, 1981.

[24] M.J. Black, P. Anandan, The robust estimation of multiple motions: parametric and piecewise-smooth flow fields, Comput. Vis. Image Underst. 63 (1) (1996) 75–104.

[25] J. Wang, E. Adelson, Representing moving images with layers, IEEE Trans. Image Process. 3 (5) (1994) 625–638.

[26] Q. Ke, T. Kanade, A robust subspace approach to layer extraction, in: Proc. Workshop on Motion and Video Computing, 2002.

[27] J. Xiao, M. Shah, Motion layer extraction in the presence of occlusion using graph cut, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2004.

[28] S. Sinha, D. Steedly, R. Szeliski, Piecewise planar stereo for image-based rendering, in: Proc. IEEE Conference on Computer Vision, 2009.

[29] G. Zhang, Z. Dong, J. Jia, T.-T. Wong, H. Bao, Efficient non-consecutive feature tracking for structure-from-motion, in: Proc. European Conference on Computer Vision, 2010.

[30] Y. Jin, L. Tao, H. Di, N. Rao, G. Xu, Background modeling from a free-moving camera by multi-layer homography algorithm, in: Proc. 15th IEEE International Conference on Image Processing, 2008.

[31] A. Monnet, A. Mittal, N. Paragios, V. Ramesh, Background modeling and subtraction of dynamic scenes, in: Proc. IEEE International Conference on Computer Vision, 2003.

[32] T. Lim, B. Han, J.H. Han, Modeling and segmentation of floating foreground and background in videos, Pattern Recogn. 45 (4) (2012) 1696–1706.

[33] S. Kwak, T. Lim, W. Nam, B. Han, J.H. Han, Generalized background subtraction based on hybrid inference by belief propagation and Bayesian filtering, in: Proc. IEEE Conference on Computer Vision, 2011.

[34] Y. Sheikh, O. Javed, T. Kanade, Background subtraction for freely moving cameras, in: Proc. IEEE 12th International Conference on Computer Vision, 2009.

[35] A. Elqursh, A.M. Elgammal, Online moving camera background subtraction, in: Proc. European Conference on Computer Vision, 2012.

[36] F. Liu, M. Gleicher, Learning color and locality cues for moving object detection and segmentation, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[37] G. Zhang, J. Jia, W. Hua, H. Bao, Robust bilayer segmentation and motion/depth estimation with a handheld camera, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 603–617.

[38] G. Evangelidis, F. Diego, R. Horaud, From video matching to video grounding, in: Proc. ICCV Workshop on Computer Vision in Vehicle Technology, 2013.

[39] P. Lai, A. Yilmaz, Efficient object shape recovery via slicing planes, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[40] R. Collins, A space-sweep approach to true multi-image matching, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1996.

[41] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2004.

[42] T. Tuytelaars, M. Proesmans, L. Van Gool, The cascaded Hough transform, in: Proc. International Conference on Image Processing, 1997, pp. 736–739.

[43] B. Triggs, Autocalibration from planar scenes, in: Proc. 5th European Conference on Computer Vision, 1998, pp. 89–105.

[44] Y. Boykov, O. Veksler, R. Zabih, Markov random fields with efficient approximations, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1998, pp. 648–655.

[45] D.M. Greig, B.T. Porteous, A.H. Seheult, Exact maximum a posteriori estimation for binary images, J. Roy. Stat. Soc. Ser. B 51 (2) (1989) 271–279.

[46] Y. Boykov, M.-P. Jolly, Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images, in: Proc. IEEE International Conference on Computer Vision, 2001.

[47] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Trans. Pattern Anal. Mach. Intell. 23 (11) (2001) 1222–1239.

[48] V. Kolmogorov, R. Zabin, What energy functions can be minimized via graph cuts?, IEEE Trans Pattern Anal. Mach. Intell. 26 (2) (2004) 147–159.

[49] Y. Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, IEEE Trans. Pattern Anal. Mach. Intell. 26 (9) (2004) 1124–1137.

[50] J. Shi, C. Tomasi, Good features to track, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 593–600.

[51] J. Yang, A.G. Hauptmann, Exploring temporal consistency for video analysis and retrieval, in: Proc. ACM International Workshop on Multimedia Information Retrieval, 2006, pp. 33–42.

[52] P. Sand, S. Teller, Particle video: long-range motion estimation using point trajectories, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2195–2202.

[53] P.W. Power, J.A. Schooness, Understanding background mixture models for foreground segmentation, in: Proc. Image and Vision Computing, 2002.

[54] R. Tron, R. Vidal, A benchmark for the comparison of 3-d motion segmentation algorithms, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[55] A. Prati, I. Mikic, M.M. Trivedi, R. Cucchiara, Detecting moving shadows: algorithms and evaluation, IEEE Trans. Pattern Anal. Mach. Intell. 25 (7) (2003) 918–923.

[56] S. Nadimi, B. Bhanu, Physical models for moving shadow and object detection in video, IEEE Trans. Pattern Anal. Mach. Intell. 26 (8) (2004) 1079–1087.