Featureless Motion Vector-based Simultaneous Localization, Planar Surface Extraction, and Moving Obstacle Tracking *

Wen Li and Dezhen Song

Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, USA. Emails: {wli,dzsong}@cse.tamu.edu

Abstract. Motion vectors (MVs) characterize the movement of pixel blocks in video streams and are readily available. MVs not only allow us to avoid expensive feature transform and correspondence computations but also provide the motion information for both the environment and moving obstacles. This enables us to develop a new framework that is capable of simultaneous localization, scene mapping, and moving obstacle tracking. This method first extracts planes from MVs and their corresponding pixel macro blocks (MBs) using properties of plane-induced homographies. We then classify MBs as stationary or moving using geometric constraints on MVs. Planes are labeled as part of the stationary scene or moving obstacles using MB voting. Therefore, we can establish planes as observations for extended Kalman filters (EKFs) for both the stationary scene and moving objects. We have implemented the proposed method. The results show that the proposed method can establish plane-based rectilinear scene structure and detect moving objects while achieving similar localization accuracy of 1-Point EKF. More specifically, the system detects moving obstacles at a true positive rate of 96.6% with a relative absolution trajectory error of no more than 2.53%.

1 Introduction

For most mobile robots in GPS-challenged environments, simultaneous localization and mapping (SLAM) and obstacle avoidance are two critical navigation functionalities. They are often handled separately because SLAM usually views moving obstacles as noises in the environment whereas obstacle avoidance only concerns the relative motion between the robot and obstacles. This artificial separation was mostly due to the limitation of existing methods. Both SLAM results and obstacle motion information should be considered together when planning robot trajectories in real applications. In fact, the artificial separation can lead to problems such as synchronization or redundant processing of information, which are not desirable for time, power, and computation constrained mobile robots.

* This work was supported in part by the National Science Foundation under IIS-1318638.

Motion vectors (MVs) characterize the movement of pixel blocks in video streams, which are readily available. With a monocular camera as the only sensor, we have employed MVs from video streams to create a new featureless SLAM method for visual navigation [15]. However, the method assumes a stationary environment despite that MVs encode motion information for both the environment and moving objects.

Here we show that MVs allow us to develop a new algorithm that is capable of performing the SLAM task and obstacle tracking in a single framework by simultaneous localization, planar surface extraction, and tracking of moving objects. Assuming a quasi-rectilinear urban environment, this method first extracts planes from MVs and their corresponding pixel macro blocks (MBs). We classify MBs as stationary or moving. These steps are based on geometric constraints and properties of plane-induced homographies under random sample consensus (RANSAC) framework. Planes are labeled as part of the stationary scene or moving obstacles using an MB voting process. This allows us to establish planes as observations for extended Kalman filters (EKFs) for both the stationary scene and moving objects. We have implemented the proposed method and compared it with the state-of-the-art 1-Point EKF [4]. The results show that the proposed method achieves similar localization accuracy. The relative absolute error is less than 2.53%. At the same time, our method can directly provide plane-based rectilinear scene structure, which is a higher level of scene understanding, and is capable of detecting moving obstacles at a true positive rate of 96.6%.

2 Related Work

Our work relates to vision-based SLAM (vSLAM) with a monocular camera. The general goal of vSLAM is to estimate the robot pose and reconstruct the 3D environment, while the robot travels in the environment. In a regular vSLAM approach, the environment is represented by a collection of landmarks, and cameras are used as the only sensors to provide observations for landmarks.

Depending upon landmarks/features, existing works for monocular vSLAM can be classified into different categories. *Feature points* have been well studied and are the most commonly used landmarks. A comprehensive study of different point detectors is provided in [11], where features like Harris corner, smallest univalue segment assimilating nucleus (SUSAN), scale invariant feature transform (SIFT), and speeded up robust features (SURF) are compared in aspects of stability and discover rates. Low level features like *edgelets* [6] and *lines* [13] are also studied, and combined for better performance. Recently, *high-level features like 3D lines and planes* [9, 10, 14, 16, 17, 20, 25] are introduced to vSLAM works to construct hierarchical environment representations, and *semantic features* such as vertical and horizontal lines [8] also attract attentions. All of these works require feature transform, which is often computationally expensive.

For many vSLAM works, a common assumption is that the environment is stationary. This assumption becomes invalid when a robot navigates in an urban environment with moving vehicles and pedestrians. In recent years, vSLAM in

dynamic environments receives increasing research attention. In existing methods, this problem is separated as a vSLAM in a stationary environment and a 3D visual tracking problem for each moving object [22, 23]. Our work is similar to these works in that we use multiple filters to track stationary and moving objects separately. However, existing methods do not perform motion separation and only work when the stationary landmarks are fixed or the moving objects' templates are given. To integrate motion separation with vSLAM, Zhou et al. [26] propose a multi-camera based approach using multiple views to triangulate points and compare the reprojection error between frames to differentiate stationary and moving points. For a monocular camera, the triangulation approach is not applicable within a single frame. Therefore, our work relies on an MV-based motion segmentation method using adjacent frames.

The motion separation in our work relates to motion-based object detection in monocular vision. Many existing MV-based object detection approaches require a stationary background [1,7,19,24]. Assuming that MBs on an object have the same motion, different clustering methods, such as expectation-maximization (EM) [1] and mean-shift [19], are used to classify foreground MVs into different regions. With the given object regions, the tracking can be performed by searching along all MVs in the object region [7]. However, these methods do not apply to our problem because the background is not stationary in our videos, and the object motion on images cannot be approximated by affine motion. Similar to MVs, optical flows (OFs) enable many motion-based object-detection work [3, 5, 18]. When a camera moves, OFs are used to detect a single dominant plane with the homography constraint [18]. When the dominating plane is the ground plane in [3], an OF model for the ground plane movement is estimated according to the camera motion where all mis-matchings to the model are detected as obstacles. Considering the low accuracy of MVs, we also use planes as landmarks in our work. However, the camera motion is unknown in our model.

3 Problem Formulation

3.1 System Overview and Introduction to Motion Vectors

Fig. 1 shows that the proposed system consists of three parts: the plane extraction and camera motion estimation (top), the stationary scene filter (middle), and the moving object filter (bottom). The plane extraction and camera motion estimation takes MVs as input and outputs labeled stationary/moving planes and the estimated camera motion between the adjacent frames. The extracted stationary planes and camera motion information are fed into the stationary scene filter to perform localization and mapping tasks. The extracted moving planes are entered to the moving object filter for tracking. Since moving and stationary planes are not permanent in applications (e.g. a moving car may come to a stop), a plane management module is introduced to allow us to add, remove, verify, and/or re-label them according to EKF outputs.

Filtered MVs are the input to the entire system. Let us briefly introduce MVs here. Detailed description and the filtering process can be found in [15].

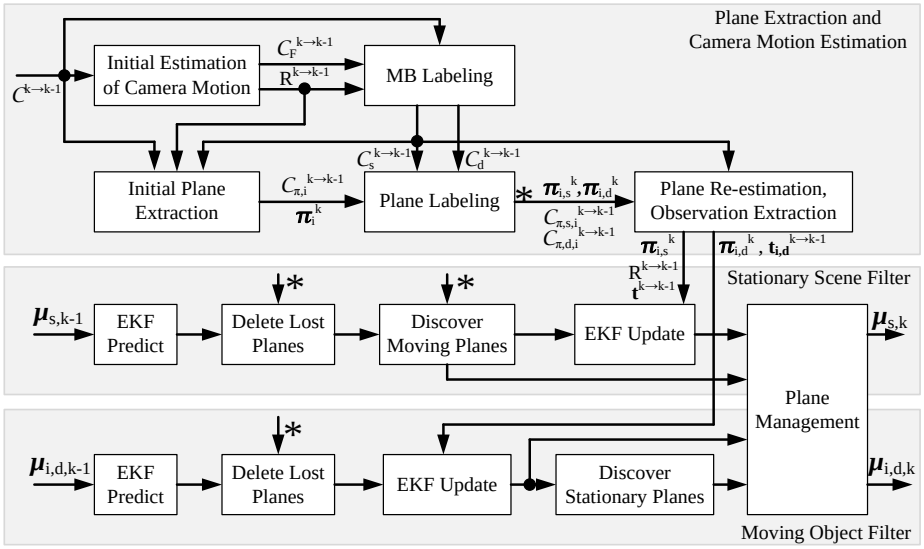


Fig. 1. System diagram. The $*$ represents the output of plane labeling, which is also the input to three sub-blocks below.

Moving Picture Experts Group (MPEG) stands for a class of video compression algorithms that are the most popular in use today. To achieve compression, each frame is partitioned into MBs in MPEG-1/2/4 standards (e.g. MPEG-2 codec uses 16×16-pixel MB). During encoding, block matching is performed to find similar MBs in reference frames. An MV is then established to represent a 2D shift of an MB with respect to (w.r.t) the reference frame. Depending on group of picture structure in different MPEG protocols, raw MVs may point to multiple future or past reference frames. It is worth noting that MVs are often noisy or missing due to the fact that MVs are computed purely based on the similarity of MBs. The similarity could be corrupted by occlusion, lighting, and large perspective changes or tricked by repetitive patterns.

Comparing to optical flows, MVs are readily available. However, MVs are sparser in spatial resolution but denser in temporal dimension. In [15], we have showed how to exploit this characteristic to reduce noise in MVs, which results in the filtered MVs. Actually, filtered MVs represent the set of corresponding MBs between key frames k and $k-1$, and are denoted by

$$\mathcal{C}^{k \to k-1} := \{\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c\}, \tag{1}$$

where $\boldsymbol{x}_k^c$ indicates the center of the MB in reference frame k and $\boldsymbol{x}_{k-1}$ shows its corresponding position in reference frame $k-1$.

3.2 Problem Definition

To formulate the problem, we assume the urban scene can be approximated using planes: stationary or moving. A set of stationary planes is a good representation of quasi-rectilinear urban environments and always exists in sight. Moving planes can approximate vehicle exteriors. We assume that there are more stationary planes than moving objects. We also assume that moving planes follow pure translation in the short duration of observation. The intrinsic camera matrix K is constant and known through pre-calibration. All 3D coordinate systems are right-handed coordinates, and common notations are defined as follows:

- *Coordinate systems*: $\{\Phi_k\}$ is a camera coordinate system (CCS) at frame k. For each CCS, its origin locates at the camera optical center, z-axis coincides with the optical axis and points to the forward direction of the camera, its x-axis and y-axis are parallel to the horizontal and vertical directions of the CCD sensor plane, respectively. The world coordinate system (WCS) $\{W\}$ coincides with $\{\Phi_0\}$. To differentiate variables in CCS and WCS, a superscription k means the variable is in $\{\Phi_k\}$ or its corresponding image coordinate system, while no superscription is default for $\{W\}$. In addition, a superscription $k \to k-1$ means from $\{\Phi_k\}$ to $\{\Phi_{k-1}\}$
- *Image coordinate system*: $\boldsymbol{x} \in \mathbb{P}^2$ is the homogeneous representation of an image coordinate where $\mathbb{P}^2$ is 2D projective space.
- *3D planes*: $\boldsymbol{\pi} = [\boldsymbol{n}^\mathsf{T}, d]^\mathsf{T}$ represents a 3D plane, where $\boldsymbol{n} \in \mathbb{R}^3$ is the plane normal vector and d is the plane depth. $\tilde{\boldsymbol{\pi}} = \boldsymbol{n}/d$ is the inhomogeneous form.
- *Subscripts*: k is the time/frame index. To distinguish stationary scene and moving objects, a subscript s stands for stationary and d represents dynamically moving. For example, $\boldsymbol{\pi}_{s,k}$ is a stationary plane at frame k.
- $\varepsilon_F(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k, F)$ denotes the Sampson's error (p. 287 in [12]) for fundamental matrix F, where $\boldsymbol{x}_k^\mathsf{T} F \boldsymbol{x}_{k-1} = 0$. $\varepsilon_H(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k, H)$ denotes the Sampson's error (p. 99 in [12]) for homography matrix H, where $\boldsymbol{x}_{k-1} = H\boldsymbol{x}_k$.

With the notations defined, we formulate the problem as below:

Problem 1 *Given the set of MVs, $\mathcal{C}^{k\to k-1}$, up to time/frame k, estimate camera rotation R_k from $\{W\}$ to $\{\Phi_k\}$ and camera location $\boldsymbol{t}_k$ in $\{W\}$ for each frame k, identify/label MBs for each plane, and reconstruct stationary and moving planes.*

To solve this problem, we begin with planar surface extraction and camera motion estimation (top box in Fig. 1).

4 Planar Surface Extraction and Camera Motion Estimation

Since MVs are often too noisy to be used directly, we exploit the coplanar property of MBs in each adjacent key frame pair to filter MVs. We estimate camera motion first and then use the motion information to label MBs by identifying whether they belong to stationary scene or moving objects. This allows us to establish planes as observations for the later EKF-based approach.

4.1 Initial Estimation of Camera Motion

With the input MVs $\mathcal{C}^{k\to k-1}$ defined in (1), let us estimate camera motion between two adjacent frames. The correct MV for the stationary scene across adjacent frames should conform the relation

$$(\boldsymbol{x}_k^c)^\mathsf{T} F^{k\to k-1}\boldsymbol{x}_{k-1} = 0, \tag{2}$$

where $F^{k\to k-1}$ is the fundamental matrix between the two frames. We first obtain an initial $F^{k\to k-1}$ using normalized 8-point algorithm under RANSAC framework (p. 281 in [12]). This gives the inlier correspondence set for $F^{k\to k-1}$:

$$\mathcal{C}_F^{k\to k-1} := \{\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c : \|(\boldsymbol{x}_k^c)^\mathsf{T} F^{k\to k-1}\boldsymbol{x}_{k-1}\| < \epsilon_f, \boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c \in \mathcal{C}^{k\to k-1}\}, \tag{3}$$

where ϵ_f is an error threshold and $\|\cdot\|$ represents the l^2 norm. This verification filters out many non-static MBs and noisy MVs that do not move along the epipolar line, such as the black arrows in Fig. 2(a).

The fundamental matrix can be parameterized by camera rotation and translation as follows:

$$F^{k\to k-1} = K^{-\mathsf{T}}[\boldsymbol{t}^{k\to k-1}]_\times R^{k\to k-1} K^{-1} \tag{4}$$

where $R^{k\to k-1}$ is the camera rotation matrix from $\{\Phi_k\}$ to $\{\Phi_{k-1}\}$, $\boldsymbol{t}^{k\to k-1}$ is the camera translation from $\{\Phi_k\}$ to $\{\Phi_{k-1}\}$ measured in $\{\Phi_k\}$, and $[\cdot]_\times$ stands for the skew-symmetric matrix representation of the cross product.

Therefore, by minimizing Sampson's error on set $\mathcal{C}_F^{k\to k-1}$ using Levenberg-Marquardt algorithm:

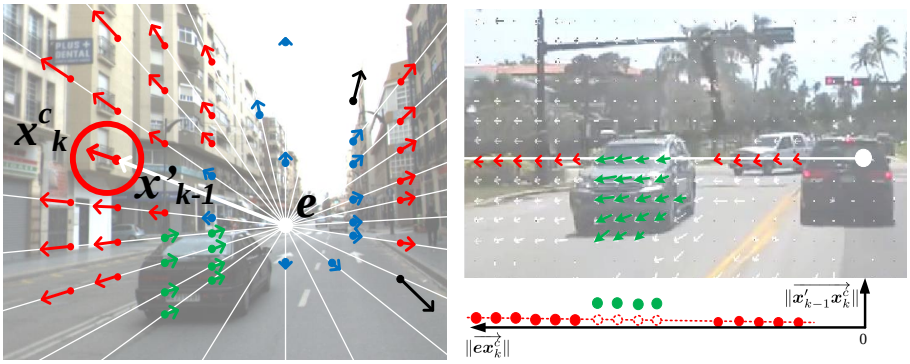
$$\min_{R^{k\to k-1}, \boldsymbol{t}^{k\to k-1}} \sum_{\boldsymbol{x}_{k-1}\leftrightarrow\boldsymbol{x}_k^c \in \mathcal{C}_F^{k\to k-1}} \varepsilon_F(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k^c, F^{k\to k-1}), \tag{5}$$

we obtain an initial estimation of camera motion between adjacent frames.

4.2 MB Labeling for Stationary and Moving Objects

Before estimating planes, we need to properly classify MBs that belong to moving objects or the stationary scene. The simple verification in (3) cannot filter out all MBs on moving objects from the stationary background. If a vehicle moves along the epipolar line, then the corresponding MBs also satisfy (3). This happens frequently when a vehicle is in front of the camera and moves in the same direction with the camera on a straight road. The green arrows on the vehicle in Fig. 2(a) show a sample case. Since there are two cases: passing vehicles from the same direction of camera motion and approaching vehicles in the opposite direction, we verify the direction and magnitude of the MVs to identify them, respectively.

MV direction constraint: For a passing vehicle on a straight road, the MVs of the vehicle move along the epipolar line in an opposite direction with the



(a) (b)

Fig. 2. Illustration of the MB labeling process (best viewed in color). The white dot and lines are the epipole and epipolar lines, respectively. Arrows indicate the movement of MBs between two adjacent frames. (a) MV direction constraint illustration: The camera motion is voted to be "forward", and red MBs are labeled stationary MBs, green and black MBs are moving MBs, and blue MBs are detected to be on the plane at infinity. (b) MV magnitude constraint illustration. Red arrows are labeled stationary, and the green arrows are moving. The red dashed line illustrates the fitted relationship between $\|\boldsymbol{x}'_{k-1}\boldsymbol{x}^c_k\|$ and $\|\boldsymbol{e}\boldsymbol{x}^c_k\|$ along the white epipolar line.

background (e.g. the green arrows in Fig. 2(a)). If we know the camera moving direction, these MVs can be detected by checking direction consistency. Therefore, we start with detecting the camera moving direction. Since we know camera rotation from (5) and are only interested in camera translation, we can remove the effect of camera rotation first. This is done by projecting $\boldsymbol{x}_{k-1}$ to $\boldsymbol{x}'_{k-1}$

$$\boldsymbol{x}'_{k-1} = sKR^{k\to k-1}K^{-1}\boldsymbol{x}_{k-1} \tag{6}$$

where s is a scalar. After the projection, the displacement between $\boldsymbol{x}'_{k-1}$ and $\boldsymbol{x}^c_k$ is caused by pure camera translation for stationary MBs. According to epiploar geometry (p. 247 in [12]), when the camera performs a pure translation, the epipole $\boldsymbol{e}$ should be a fixed point, and all stationary MBs should appear to move along lines radiating from the epipole (see Fig. 2(a)). The colored dots in the figure are $\boldsymbol{x}'_{k-1}$ and the arrows point to $\boldsymbol{x}^c_k$, an illustration of MVs.

If the camera moves forward along its optical axis, vectors $\overrightarrow{\boldsymbol{e}\boldsymbol{x}'_{k-1}}$ and $\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}^c_k}$ should be in the same direction, as the red arrows in the highlighted circle shown in Fig. 2(a). If the camera moves backward, $\overrightarrow{\boldsymbol{e}\boldsymbol{x}'_{k-1}}$ and $\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}^c_k}$ should be in the opposite direction. Denote the absolute angle between $\overrightarrow{\boldsymbol{e}\boldsymbol{x}'_{k-1}}$ and $\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}^c_k}$ as α. Of course, the perfect collinear relationship may not hold due to noises in the system. α is always somewhere between 0° and 180°. We examine each MV $\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}^c_k \in \mathcal{C}_F^{k\to k-1}$. If its α is less than 90°, a vote of "forward" is assigned, otherwise a "backward" vote is assigned. Then the camera moving direction is obtained as the majority direction from all inlier correspondences. Fig. 2(a)

shows the camera moving direction is voted as "forward" because most of the MBs move away from the epipole. With the detected camera moving direction, we can identify MBs belonging to passing vehicles easily. However, this would not work for vehicles approaching the camera along the direction parallel to camera motion vector. The MVs on the approaching vehicles also move along the epipolar line and share the same direction as the background motion. For such cases, we need to verify the magnitude of MVs.

MV magnitude constraint: The additional motion introduced by the object results in sudden changes of MV magnitude along the epipolar line. To detect this type of moving objects, we start with computing the magnitude of MVs after removing camera rotation. Denote the MV magnitude of $\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c$ as $\|\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}_k^c}\|$, and the Euclidean distance between the MB and the epipole as $\|\overrightarrow{\boldsymbol{e}\boldsymbol{x}_k^c}\|$. From projective geometry we know that closer objects have larger displacements under the same camera motion. Therefore, along one epipolar line, $\|\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}_k^c}\|$ should gradually increase as $\|\overrightarrow{\boldsymbol{e}\boldsymbol{x}_k^c}\|$ increases. For each epipolar line, we approximate the 2D relationship between $\|\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}_k^c}\|$ and $\|\overrightarrow{\boldsymbol{e}\boldsymbol{x}_k^c}\|$ using RANSAC-based line fitting. An example of the fitted relationship is shown by the dashed line at the bottom of Fig. 2(b). Therefore, for a given $\|\overrightarrow{\boldsymbol{e}\boldsymbol{x}_k^c}\|$ on the epipolar line, an predicted MV magnitude $\|\widetilde{\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}_k^c}}\|$ can be obtained from the fitted relationship (dashed circles in Fig. 2(b)). If the difference between $\|\widetilde{\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}_k^c}}\|$ and $\|\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}_k^c}\|$ is greater than a threshold ϵ_e, we consider the corresponding MB is potentially moving. In the example shown in Fig. 2(b), the green MBs have magnitudes much greater than the expected red dashed line, and thus labeled as moving MBs.

With the above constraints, we can label every MB and partition the set $\mathcal{C}^{k\to k-1}$ into a stationary correspondence set $\mathcal{C}_s^{k\to k-1}$ and a moving correspondence set $\mathcal{C}_d^{k\to k-1}$, where $\mathcal{C}_s^{k\to k-1} \bigcup \mathcal{C}_d^{k\to k-1} = \mathcal{C}^{k\to k-1}$:

Definition 1 (MB Labeling) *An MV $\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c \in \mathcal{C}^{k\to k-1}$ and its corresponding MBs are labeled as stationary $\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c \in \mathcal{C}_s^{k\to k-1}$, if the following three conditions are all satisfied:*

1) $\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c \in \mathcal{C}_F^{k\to k-1}$,

2) $\alpha < 90^\circ$ if camera moves forward or $\alpha \geq 90^\circ$ if camera moves backward,

3) $|\|\widetilde{\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}_k^c}}\| - \|\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}_k^c}\|| < \epsilon_e$.

Otherwise, the MB belongs to moving objects: $\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c \in \mathcal{C}_d^{k\to k-1}$.

In Fig. 2(a), the MBs on building facades are labeled as stationary with red arrows whereas the MBs on the vehicle are labeled as moving.

4.3 Initial Plane Extraction and Labeling

With the labeled MB correspondences, we are able to extract planar regions. Since MBs in the plane at infinity π_∞ have very low signal-to-noise ratio for

camera translation estimation, they should be removed before plane extraction for better accuracy. Denote the set of correspondences in π_∞ as $\mathcal{C}_\infty$,

$$\mathcal{C}_\infty^{k\to k-1} := \{\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c : \|\boldsymbol{x}'_{k-1} - \boldsymbol{x}_k^c\| < \epsilon_m, \boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c \in \mathcal{C}_s^{k\to k-1}\} \quad (7)$$

where ϵ_m is the motion threshold. Fig. 2(a) shows the detected π_∞ in blue arrows.

On the rest of correspondences $\mathcal{C}^{k\to k-1} \setminus \mathcal{C}_\infty^{k\to k-1}$, RANSAC is applied iteratively to extract all possible planes. To extract one plane, four correspondences are sampled, and an homography H is obtained using normalized direct linear transformation (p. 109 in [12]). Then, all correspondence resulting in an error below a given threshold: $\|\boldsymbol{x}_{k-1} - \lambda H \boldsymbol{x}_k^c\| < \epsilon_h$, is labeled as an inlier to the plane. In each RANSAC iteration, one largest plane is extracted, and its inliers are removed before next RANSAC iteration. This iterative RANSAC procedure can be replaced by J-linkage [21] if needed.

Then a set of planes, $\mathit{\Pi}^{k\to k-1} = \{\tilde{\boldsymbol{\pi}}_i^k, i \in \mathcal{I}\}$ is initially constructed from $\{\mathit{\Phi}_k\}$. We use $\mathcal{I}$ to denote the index set for planes, and $i \in \mathcal{I}$ is the i-th plane. For each extracted plane $\tilde{\boldsymbol{\pi}}_i^k$, we denote its corresponding MV set as $\mathcal{C}_{\pi,i}^{k\to k-1}$. Thus, $\bigcup_{i\in\mathcal{I}} \mathcal{C}_{\pi,i}^{k\to k-1} \subseteq \mathcal{C}^{k\to k-1} \setminus \mathcal{C}_\infty^{k\to k-1}$. To perform tracking and improve plane estimation, all planes need to be labeled as either stationary or moving. With the MB labeling result $\mathcal{C}_s^{k\to k-1}$ and $\mathcal{C}_d^{k\to k-1}$, the plane labeling is determined by the result of a majority voting of labeled MBs:

Definition 2 (Plane Labeling) *A plane $\tilde{\boldsymbol{\pi}}_i^k \in \mathit{\Pi}^{k\to k-1}$ and its corresponding MV set $\mathcal{C}_{\pi,i}^{k\to k-1}$ are labeled as stationary $\tilde{\boldsymbol{\pi}}_{i,s}^k$ and $\mathcal{C}_{\pi,i,s}^{k\to k-1}$, respectively, if $|\mathcal{C}_{\pi,i}^{k\to k-1} \bigcap \mathcal{C}_s^{k\to k-1}| > |\mathcal{C}_{\pi,i}^{k\to k-1} \bigcap \mathcal{C}_d^{k\to k-1}|$. Otherwise, they are labeled as moving objects, $\tilde{\boldsymbol{\pi}}_{i,d}^k$ and $\mathcal{C}_{\pi,i,d}^{k\to k-1}$, respectively.*

After the labeling step, the set of all planes $\mathit{\Pi}^{k\to k-1}$ is partitioned into

$$\mathit{\Pi}^{k\to k-1} = \mathit{\Pi}_s^{k\to k-1} \bigcup \mathit{\Pi}_d^{k\to k-1}, \quad (8)$$

where $\mathit{\Pi}_s^{k\to k-1} = \{\tilde{\boldsymbol{\pi}}_{i,s}^k\}$ is the set of stationary planes and $\mathit{\Pi}_d^{k\to k-1} = \{\tilde{\boldsymbol{\pi}}_{i,d}^k\}$ denotes the set of moving planes.

4.4 Plane Re-estimation and Observation Extraction

With the labeled planes, we can refine all estimations and prepare observations for EKFs. We start with the stationary scene and the camera motion. For a stationary plane $\tilde{\boldsymbol{\pi}}_{i,s}^k$, the correspondences $\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c \in \mathcal{C}_{\pi,i,s}^{k\to k-1}$ conform to homography relation:

$$\boldsymbol{x}_{k-1} = H_i^{k\to k-1}\boldsymbol{x}_k^c = K(R^{k\to k-1})^{-1}[I_{3\times 3} + \boldsymbol{t}^{k\to k-1}(\tilde{\boldsymbol{\pi}}_{i,s}^k)^\mathsf{T}]K^{-1}\boldsymbol{x}_k^c, \quad (9)$$

where $H_i^{k\to k-1}$ is the homography matrix introduced by the plane, $I_{3\times 3}$ is a 3-dimensional identity matrix. Therefore, for the stationary scene, the observations of relative camera motion and stationary plane equations can be estimated

by minimizing the total errors of fundamental relationship in all stationary correspondences and homography relationship in all planar correspondences:

$$\min_{R^{k\to k-1},\boldsymbol{t}^{k\to k-1},\tilde{\boldsymbol{\pi}}^k_{i,s}\in\Pi_s^{k\to k-1}} \sum_{\boldsymbol{x}_{k-1}\leftrightarrow\boldsymbol{x}_k^c\in\mathcal{C}_s^{k\to k-1}} \varepsilon_F(\boldsymbol{x}_{k-1},\boldsymbol{x}_k^c,F^{k\to k-1}) + \sum_i \sum_{\boldsymbol{x}_{k-1}\leftrightarrow\boldsymbol{x}_k^c\in\mathcal{C}_{\pi,i,s}^{k\to k-1}} \varepsilon_H(\boldsymbol{x}_{k-1},\boldsymbol{x}_k^c,H_i^{k\to k-1}) \quad (10)$$

where $F^{k\to k-1}$ and $H_i^{k\to k-1}$ are from (4) and (9), respectively. The resulting optimal $R^{k\to k-1}$, $\boldsymbol{t}^{k\to k-1}$ and $\tilde{\boldsymbol{\pi}}^k_{i,s}$'s are inputs to the stationary EKF in the next section.

For a moving plane $\tilde{\boldsymbol{\pi}}^k_{i,d}$, denote its translation as $\boldsymbol{t}_d$. If we back shift the plane by $-\boldsymbol{t}_d$, then a homography relationship can be established for $\boldsymbol{x}_{k-1}\leftrightarrow\boldsymbol{x}_k^c\in\mathcal{C}_{\pi,i,d}^{k\to k-1}$,

$$H_i^{k\to k-1}=K(R^{k\to k-1})^{-1}[I_{3\times3}+(\boldsymbol{t}^{k\to k-1}-\boldsymbol{t}_{i,d}^{k\to k-1})(\tilde{\boldsymbol{\pi}}^k_{i,d})^\mathsf{T}]K^{-1}, \quad (11)$$

Therefore, a moving plane is estimated by minimizing the following,

$$\min_{\tilde{\boldsymbol{\pi}}^k_{i,d},\boldsymbol{t}_{i,d}^{k\to k-1}} \sum_{\boldsymbol{x}_{k-1}\leftrightarrow\boldsymbol{x}_k^c\in\mathcal{C}_{\pi,i,d}^{k\to k-1}} \varepsilon_H(\boldsymbol{x}_{k-1},\boldsymbol{x}_k^c,H_i^{k\to k-1}) \quad (12)$$

where $H_i^{k\to k-1}$ is from (11) with the estimated camera motion from (10). The resulting optimal plane equations and translations are inputs to the individual moving object filters later.

5 EKF-based Localization and Tracking

With the planes and camera motions extracted for adjacent key frame pairs, we can feed them as observations to EKFs for global robot localization, stationary plane mapping, and moving object tracking. As Fig. 1 shows, the robot localization and stationary plane mapping are handled by one single EKF below.

Camera Localization and Static Scene Mapping: Based on stationary planes, this part is similar to the traditional visual SLAM problem. Following an EKF framework, we define the state vector $\boldsymbol{\mu}_k$ for the EKF filter as follows:

$$\boldsymbol{\mu}_{s,k}=[...,\tilde{\boldsymbol{\pi}}^\mathsf{T}_{i,s,k},...,\boldsymbol{r}_k^\mathsf{T},\boldsymbol{t}_k^\mathsf{T},\dot{\boldsymbol{r}}_k^\mathsf{T},\dot{\boldsymbol{t}}_k^\mathsf{T}]^\mathsf{T}, \quad (13)$$

which includes the plane equations in $\{W\}$, the y-x-z Euler angles $\boldsymbol{r}_k$ for camera rotation from $\{W\}$ to $\{\Phi_k\}$, the camera location $\boldsymbol{t}_k$ in $\{W\}$, camera motion velocity $\dot{\boldsymbol{t}}_k$ in $\{W\}$, and the angular velocity of the camera $\dot{\boldsymbol{r}}_k$ in $\{\Phi_k\}$. Since stationary planes are segmented as observations, the problem is reduced to the same problem in [15]. We can employ the same EKF design in [15].

Moving Object Tracking: Similarly, this step is also handled using EKF (the bottom part of Fig. 1). Moving objects are considered to move independently w.r.t to the camera and each other. We employ one EKF to track each

moving object individually. In each EKF, one global plane equation and one velocity vector are tracked. Here, we assume the motion of moving plane follows a constant linear velocity in $\{W\}$ without rotation, which is usually true for pedestrians or vehicles appearing in the camera view for a short period of time. The state vector for a single moving plane filter becomes

$$\boldsymbol{\mu}_{i,d,k} = [\tilde{\boldsymbol{\pi}}_{i,d,k}^{\mathsf{T}}, \boldsymbol{v}_{i,d,k}^{\mathsf{T}}]^{\mathsf{T}}, \tag{14}$$

where $\boldsymbol{v}_{i,d,k}$ is the velocity of the i-th object in $\{W\}$. The state transition for the moving object i is straightforward:

$$\begin{cases} \tilde{\boldsymbol{\pi}}_{i,d,k} = \tilde{\boldsymbol{\pi}}_{i,d,k-1}/(1 - \tilde{\boldsymbol{\pi}}_{i,d,k-1}^{\mathsf{T}} \boldsymbol{v}_{i,d,k-1}\tau) \\ \boldsymbol{v}_{i,d,k} = \boldsymbol{v}_{i,d,k-1} \end{cases}, \tag{15}$$

where τ is the time interval. The observations for the moving object filters are the estimated plane equations in $\{\Phi_k\}$, and the observation function is the transform between coordinate systems given the camera rotation and translation:

$$\boldsymbol{z}_{i,d,k} = [(\tilde{\boldsymbol{\pi}}_{i,d}^{k})^{\mathsf{T}}, (\boldsymbol{t}_{i,d}^{k\to k-1})^{\mathsf{T}}]^{\mathsf{T}} = \begin{bmatrix} R(\boldsymbol{r}_k)^{-1}\tilde{\boldsymbol{\pi}}_{i,d,k}/(1 + \tilde{\boldsymbol{\pi}}_{i,d,k}^{\mathsf{T}}\boldsymbol{t}_k) \\ -\tau R(\boldsymbol{r}_k)^{-1}\boldsymbol{v}_{i,d,k} \end{bmatrix}. \tag{16}$$

Plane Management: Apart from removal of planes that are no longer in the sight from the corresponding EKFs, plane labels are not permanent as a moving object may come to a stop or a parked vehicle may start moving. Since each plane has a stationary/moving label, plane label exchange happens when the label of an existing plane is not consistent with the outcome of the EKF. A moving plane's label will also be changed to stationary if its velocity is close to zero. When a plane changes its label, the corresponding state variables are moved from previous EKF filter to the EKF corresponding to the new label, with an initialized velocity if necessary. For each newly discovered plane, its parameters are added into the corresponding EKF according to its label.

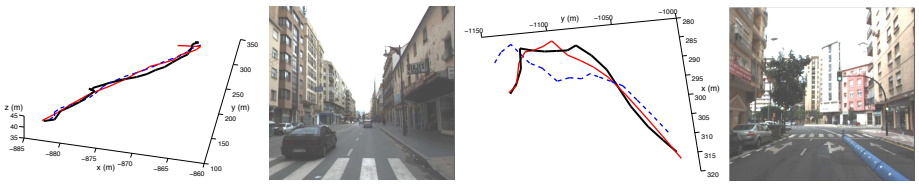
6 Experiments

We have implemented the proposed system using C/C++ in Cygwin environment under Microsoft Windows 7. To test the performance of the method, evaluation is conducted in the following three aspects: the localization error, the stationary plane estimation error, and the detection of moving planes.

6.1 Localization Evaluation

Dataset: We perform the evaluation using the Màlaga urban dataset [2] which provides stereo videos from vehicle driving in a dense urban area. The video frame rate is 20 fps. Images with a resolution of 1024×768 are rectified and the intrinsic camera matrix after rectification is provided. Ground truth data are collected using multiple sensors including GPS, IMU, and laser range

finder. Since we assume the scene is quasi-rectilinear with many static planes, two typical urban scenes from the data set are used in the experiment. Since our method is monocular, we only use the images from the left camera in the dataset. Sample thumbnails of frames in the experiment are shown in Fig. 3. The lengths (i.e. travel distance) of the two sequences are provided in Tab. 1.



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

Fig. 3. Trajectories and sample frame thumbnails. (a) and (c) are the camera trajectories in the two sequences, measured in meters. Black lines are the GPS ground truth, red solid lines are the estimated trajectories using our method and the blue dashed lines are trajectories estimated using [4]. (b) and (d) are the sample image frames in the two sequences.

Metric: The localization result is compared with GPS data. The GPS data is sampled once per second, and the image time stamps are aligned according to the GPS clock. The errors are measured using the absolute trajectory error (ATE) [4]. We define the GPS coordinate system by $\{G\}$ and the camera position in $\{G\}$ as $\hat{\boldsymbol{t}}_k^G$. For the estimated camera position $\boldsymbol{t}_k$ in $\{W\}$, a similarity transformation (rotation $R^{W\to G}$, translation $\boldsymbol{t}^{W\to G}$ and scale s) is applied to transform the position to the GPS coordinate $\boldsymbol{t}_k^G = sR^{W\to G}\boldsymbol{t}_k + \boldsymbol{t}^{W\to G}$. The rotation, translation and scale are obtained via a non-linear optimization that minimizes the total error between the GPS data $\hat{\boldsymbol{t}}_k^G$ and the transformed estimation result $\boldsymbol{t}_k^G$. Therefore, the ATE for a frame k is defined as $e_k = \|\boldsymbol{t}_k^G - \hat{\boldsymbol{t}}_k^G\|$.

Comparison: We compare our result with the popular 1-Point EKF [4] since both methods are EKF-based. The 1-point EKF [4] approach uses feature points as landmarks. Their system is tested under long distance trajectories with robust performance. The code for 1-Point EKF is acquired from the authors' website and is directly run in Matlab on our testing dataset. Tab. 1 shows the mean and maximum ATE for each sequence for both methods. The results show that the mean ATEs of our method are below 3.5 meters for both sequences and are below 3% of the overall trajectory length, which is comparable to [4]. In the first sequence, the vehicle travels on a mostly straight road, with occasional lane changes. In this case, our method and [4] perform similar, with [4] slightly better. In the second sequence, the vehicle starts from straight driving and experiences curved road later. In this case, our method outperforms [4] over 5 meters in average. This experiment confirms that MV-based featureless navigation method is feasible.

Table 1. Localization Results using the Màlaga Dataset

	length (m)	#frames	method	mean ATE	max ATE	% over distance
seq 1	201.08	497	Our method	2.87m	6.33m	1.43%
			1-Point EKF	1.99m	3.67m	0.99%
	length (m)	#frames	method	mean ATE	max ATE	% over distance
seq 2	133.76	318	Our method	3.38m	4.99m	2.53%
			1-Point EKF	9.08m	12.30m	6.80%

6.2 Stationary Plane Estimation

To evaluate plane mapping accuracy, we compare our method with our previous work [15] which is referred as SLAPSE method since it only performs localization and plane mapping without ability of tracking moving objects. We use the dataset from [15] for comparison where ground truth is computed by points measured using a laser distance measurer with ±1 mm accuracy. The reason that we do not use the Màlaga urban dataset here is because there is no ground truth data for planes. Similar to [15], we only consider the planes that appear in more than 3 continuous frames. The same error functions in [15] for plane depth and angles are used:

$$\epsilon_d = \frac{1}{\sum_i N_i} \sum_i \sum_k |d^k_{i,k} - \hat{d}^k_{i,k}|, \text{ and } \epsilon_n = \frac{1}{\sum_i N_i} \sum_i \sum_k |\arccos((\boldsymbol{n}^k_{i,k})^\mathsf{T} \cdot \hat{\boldsymbol{n}}^k_{i,k})|, \tag{17}$$

where N_i is the number of frames plane i appears, and $\hat{}$ stands for the ground truth. The number of planes extracted in the site and the estimation errors are shown in Tab. 2. The comparison results show our method improves the estimation of scene planes in both depth and orientation accuracy.

Table 2. Static Plane Estimation Results

method	# planes	ϵ_d (m)	ϵ_n (degs.)
Our method	5	0.55	6.80
SLAPSE	5	0.61	7.07

6.3 Moving Object Detection

To evaluate the performance of moving object detection, the test is focused on the plane labeling algorithm as the EKF-based tracking performance is determined by the labeling correctness. A dataset of 64 video clips are manually collected from the Internet, such as YouTube. All video clips are recorded by cameras mounted on vehicles driving in urban environments. The frame rates vary between 23 and 30 fps, and the image resolution is between 640×360 and

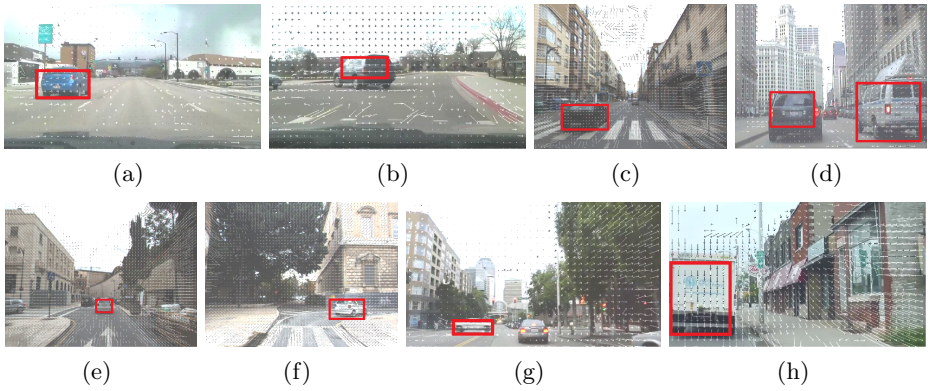


Fig. 4. Detected moving objects are highlighted with red rectangles.

1024×768. From all videos, there are a total of 88 moving vehicles that are manually identified, and their bounding box in each frame in annotated as ground truth. Note that the vehicles parking at red light or curbside are not labeled as moving objects, and the vehicles that are very far are not labeled because they are not objects of interest for collision avoidance.

Then the plane extraction and labeling method in Sec. 4 is applied to extract stationary and moving planes. Among 88 labeled moving objects, 85 are detected and labeled as moving planes, and the detection rate is 96.6%. Among the 3 failure cases, 2 cases are caused by lack of correct MVs on the vehicles. This situation happens when the vehicle is too texture-less and has a color either similar to the ground or with large saturation. Another 1 case happens because the vehicle is relatively stationary to the camera, thus the MVs on it are not distinguishable from those on the infinite plane. The right most vehicle in Fig. 2(b) shows an example of this situation. Actually, due to the zero relative speed, that vehicle is not a concern for collision avoidance purpose.

Fig. 4 shows some examples of the detected moving planes in a bounding box. The detection of moving object helps to separate outliers and wrong MVs that influence the static localization and mapping results.

7 Conclusion and Future Work

We presented a new algorithm that is capable of performing SLAM task and obstacle tracking using MVs as inputs. This algorithm simultaneously localizes the robot, establishes scene understanding through planar surface extraction, and tracks moving objects. To achieve this, we first extracted planes from MVs and their corresponding pixel MBs. We labeled MBs as either stationary or moving using geometric constraints and properties of plane-induced homographies. Similarly, planes were also labeled as either stationary or moving using an MB voting process. This allows us to establish planes as observations for extended Kalman

filters (EKFs) for both stationary scene mapping and moving object tracking. We implemented the proposed method and compared it with the state-of-the-art 1-point EKF. The results showed that the proposed method achieved similar localization accuracy. However, our method can directly provide plane-based rectilinear scene structure, which is a higher level of scene understanding, and is capable of detection moving obstacles at a true positive rate of 96.6%.

In the future, we plan to adopt a local bundle adjustment approach to further improve localization accuracy. We will combine MVs with appearance data to establish higher level scene mapping. Fusing with other sensors such as depth or inertial sensors is also under consideration.

Acknowledgement

Thanks for Y. Lu, J. Lee, M. Hielsberg, X. Wang, Y. Liu, S. Jacob, P. Peelen, Z. Gui, and M. Jiang for their inputs and contributions to the NetBot Laboratory, Texas A&M University.

References

1. Babu, R., Ramakrishnan, K.: Compressed domain motion segmentation for video object extraction. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). pp. IV–3788–IV–3791 (2002)
2. Blanco-Claraco, J., nas, F.M.D., Gozalez-Jimenez, J.: The màlaga urban dataset: High-rate stereo and lidars in a realistic urban scenario. International Journal of Robotics Research (IJRR) (2013, DOI: 101177/0278364913507326)
3. Braillon, C., Pradalier, C., Crowley, J., Laugier, C.: Real-time moving obstacle detection using optical flow methods. In: IEEE Intelligent Vehicles Symposium. pp. 466–471. Tokyo, Japan (2006)
4. Civera, J., Grasa, O., Davison, A., Montiel, J.: 1-point ransac for extended kalman filtering: Application to real-time structure from motion and visual odometry. Journal of Field Robotics 27(5), 609–631 (2010)
5. Denman, S., Fookes, C., Sridharan, S.: Improved simultaneous computation of motion detection and optical flow for object tracking. In: Digital Image Computing: Techniques and Applications. pp. 175 – 182 (2009)
6. Eade, E., Drummond, T.: Edge landmarks in monocular slam. In: British Machine Vision Conference (BMVC). pp. 7–16 (Sep 2006)
7. Favalli, L., Mecocci, A., Moschetti, F.: Object tracking for retrieval applications in mpeg-2. IEEE Transactions on Circuits and Systems for Video Technology 10(3), 427–432 (2000)
8. Flint, A., Mei, C., Reid, I., Murray, D.: Growing semantically meaningful models for visual slam. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 467–474. San Francisco, CA (Jun 2010)
9. Gee, A., Chekhlov, D., Calway, A., Mayol-Cuevas, W.: Discovering higher level structure in visual slam. IEEE Transactions on Robotics 24(5), 980–990 (Oct 2008)
10. Gee, A., Chekhlov, D., Mayol, W., Calway, A.: Discovering planes and collapsing the state space in visual slam. In: BMVC. pp. 1–10 (2007)

11. Gil, A., Mozos, O., Ballesta, M., Reinoso, O.: A comparative evaluation of interest point detectors and local descriptors for visual slam. Machine Vision and Applications 21(6), 905–920 (2010)
12. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press. (2003)
13. Jeong, W., Lee, K.: Visual slam with line and corner features. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. Beijing, China (Oct 2006)
14. Li, H., Song, D., Lu, Y., Liu, J.: A two-view based multilayer feature graph for robot navigation. In: IEEE International Conference on Robotics and Automation (ICRA), St. Paul, Minnesota (May 2012)
15. Li, W., Song, D.: Toward featureless visual navigation: Simultaneous localization and planar surface extraction using motion vectors in video streams. In: IEEE International Conference on Robotics and Automation. Hong Kong, China (May 2014)
16. Lu, Y., Song, D., Xu, Y., Perera, A., Oh, S.: Automatic building exterior mapping using multilayer feature graphs. In: IEEE International Conference on Automation Scienece and Engineering. Madison, Wisconsin (Aug 2013)
17. Lu, Y., Song, D., Yi, J.: High level landmark-based visual navigation using unsupervised geometric constraints in local bundle adjustment. In: IEEE International Conference on Robotics and Automation. Hong Kong, China (May 2014)
18. Ohnishi, N., Imiya, A.: Dominant plane detection from optical flow for robot navigation. Pattern Recognition Letters 27, 1009–1021 (2006)
19. Park, S., Lee, J.: Object tracking in mpeg compressed video using mean-shift algorithm. In: Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. vol. 2, pp. 748–752 (2003)
20. Pietzsch, T.: Planar features for visual slam. In: KI 2008: Advances in Artificial Intelligence. pp. 119–126. Kaiserslautern, Germany (Sep 2008)
21. Toldo, R., Fusiello, A.: Robust multiple structures estimation with j-linkage. In: European Conference on Computer Vision. pp. 537–547 (2008)
22. Wang, Y., Lin, M., Ju, R.: Visual slam and moving-object detection for a small-size humanoid robot. International Journal of Advanced Robotic Systems 7(2), 133–138 (2010)
23. Wangsiripitak, S., Murray, D.: Avoiding moving outliers in visual slam by tracking moving objects. In: IEEE International Conference on Robotics and Automation. Kobe, Japan (May 2009)
24. Yokoyama, T., Iwasaki, T., Watanabe, T.: Motion vector based moving object detection and tracking in the mpeg compressed domain. In: Seventh International Workshop on Content-based Multimedia Indexing. pp. 201–206 (2009)
25. Zhang, J., Song, D.: Error aware monocular visual odometry using vertical line pairs for small robots in urban areas. In: Special Track on Physically Grounded AI (PGAI), the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10). Atlanta, Georgia, USA (July 2010)
26. Zou, D., Tan, P.: Coslam: Collaborative visual slam in dynamic environments. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(2), 354–366 (2013)