Toward Featureless Visual Navigation: Simultaneous Localization and Planar Surface Extraction Using Motion Vectors in Video Streams

Wen Li and Dezhen Song

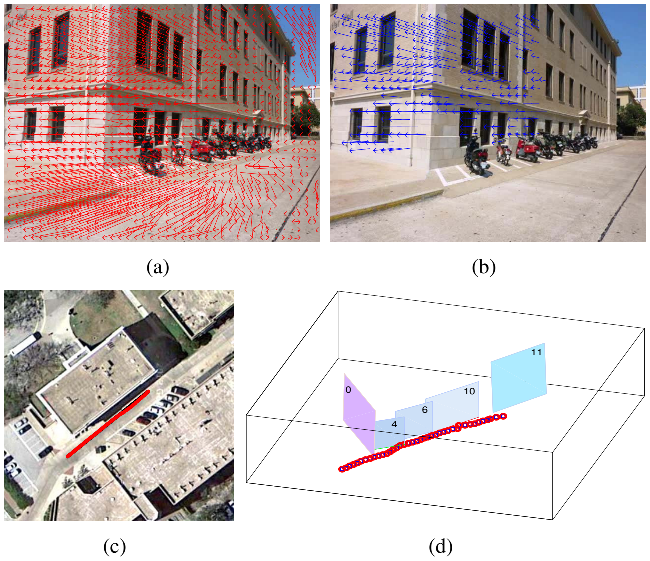
***Abstract*— Unlike the traditional feature-based methods, we propose using motion vectors (MVs) from video streams as inputs for visual navigation. Although MVs are very noisy and with low spatial resolution, MVs do possess high temporal resolution which means it is possible to merge MVs from different frames to improve signal quality. Homography filtering and MV thresholding are proposed to further improve MV quality so that we can establish plane observations from MVs. We propose an extended Kalman filter (EKF) based approach to simultaneously track robot motion and planes. We formally model error propagation of MVs and derive variance of the merged MVs, which provide the necessary observation error model for the EKF. We have implemented the proposed method and tested it in physical experiments. Results show that the system is capable of performing robot localization and plane mapping with a relative trajectory error of less than 5.1%.**

I. Introduction

Many visual navigation approaches rely on correspondence of features between individual images to establish geometric understandings of image data. To do that, image data are often first reduced to a feature set such as points. Then extensive statistical approaches such as random sample consensus (RANSAC) are employed to search for feature matches that satisfy the expected geometry relationships. Such geometric relationships enable us to derive robot/camera ego-motion estimation or scene understandings in different applications such as visual odometry or simultaneous localization and mapping (SLAM) [1]. The inherent drawback of these approaches is the expensive computation load and robustness of feature extraction, which is often hindered by varying lighting conditions and occlusions.

On the other hand, recent streaming videos are transmitted after complex compression. These algorithms exploit similarities between blocks of pixels in adjacent frame sets, which are characterized as motion vectors (MVs), to reduce bandwidth needs (Fig. 1(a)). Compared with optical flows, MVs have lower spatial resolution (per block vs. per pixel) but higher temporal resolution because MVs are extracted from multiple frames instead of mere two adjacent frames. MVs carry the correspondence information and are readily available from the encoded video data.

Despite all the aforementioned advantages, MVs are not easy to use because of their low spatial resolution and relatively high noise. Here we explore how to use MVs for simultaneous localization and planar surface extraction (SLAPSE) for a mobile robot equipped with a single camera. We establish the MV noise models to capture the observation error. We formulate the SLAPSE problem and study how to extract planes from MVs using planar homography filtering. We then develop an extended Kalman filter (EKF) based approach with planes and robot motion as state variables. We have implemented our algorithm using C/C++ on a PC platform and tested the algorithm in physical experiments. The results show that the system is capable of performing robot localization and plane mapping with a relative trajectory error of less than 5.1%.



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

Fig. 1. (a) Original MVs represented by red arrows. (b) Filtered MVs represented by blue arrows. (c) Satellite image of an experiment site. Black line is manually measured ground truth camera trajectory, red line is the estimated trajectory. (d) Estimated plane positions and camera trajectory.

II. Related Work

SLAPSE relates to recent progress in visual navigation for mobile robots, MPEG compression, and dense 3D reconstruction.

SLAPSE can be viewed as visual SLAM with special observation inputs. In a regular SLAM framework, the physical world is represented by a collection of landmarks which are primarily features observed from images, such as key points [2]–[5], line segments [6]–[11], curves [12], and surfaces [13]. In these feature-based approaches, SLAM performance is largely dependent of feature distributions and correspondences. Building on these approaches, our SLAPSE takes advantage of the fact that MVs encode correspondences of segmented scene by overcoming the noise in the MV data.

Many efforts have been made to improve the accuracy and

W. Li and D. Song are with CSE Department, Texas A&M University, College Station, TX 77843, USA. Email: {*wli, dzsong*}@*cse.tamu.edu*.

speed of MV computation in MPEG encoding. However, few studies have been conducted on utilizing MVs in complex vision problems. The main reason is because MVs are very noisy and have spatially low resolution. MVs have been applied in fast image-based camera rotation estimation [14], 2D object tracking [15], and image stabilization [16]. All of these approaches employ voting or averaging like strategies with region-based smoothing to obtain either foreground or background information separately. SLAPSE problems need to recover both the scene structure and the robot motion which require MVs with much less errors. We merge MVs across multiple adjacent frames to improve the signal to noise ratio, analyze errors on merged MVs, and utilize geometry relationship for better noise filtering.

MVs directly provide correspondences between pixel blocks. Once planes are identified through MVs in the SLAPSE problem, their corresponding pixel blocks are subsequently reconstructed in 3D. This is close to feature-based dense reconstruction, which usually requires precise dense correspondence between images. Recent dense reconstruction approaches start with a sparse set of salient points, and construct dense surfaces using photoconsistency and geometrical constraints [17]. More relevant works [18] utilize variational optical flow [19] to establish dense surface meshes from point clouds. These works inspire us to use MVs in scene mapping.

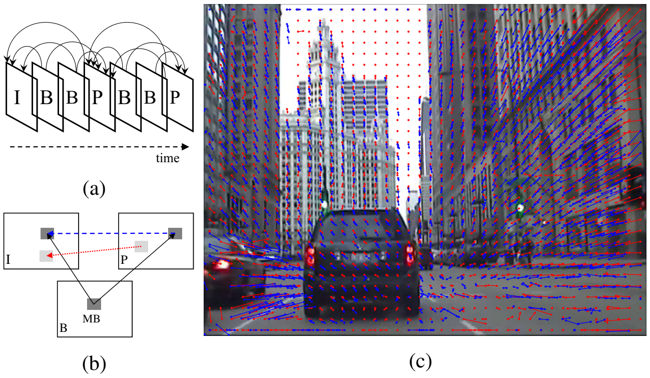
Our group focuses on developing monocular visual navigation techniques for energy and computation constrained robots. Using a vector-field approach [20], we develop a lightweight visual navigation algorithm for an autonomous motorcycle. We also address depth ambiguity problem through planning for small robot systems [21]. We have attempted different features for visual odometry such as vertical line segments [22], [23] and high level features [24], [25] to improve robustness. Through the process, we have learned shortcomings of feature-based approaches, which has motivated this work.

III. Background and Problem Definition

A. A Brief Introduction to Motion Vectors

Video encoders such as MPEG 1/2/4 often utilize block motion compensation (BMC) to achieve better data compression. BMC partitions each frame into small macroblocks (MB) (e.g. each MB is 16×16 pixels for MPEG 2). During encoding, block matching is employed to search for similar MBs in anchor frames. If a matching block is found, an MV is established. Note that each MV only represents a 2D shift in the image frame.

We use MPEG 2 as an example, and our analysis can be easily extended to other BMC-based encoding formats. There are often three types of frames (or slices of a frame): intra coded, predictive coded, and bidirectionally predictive coded, namely, I, P, and B frames, respectively. P and B frames consist of MBs defined by MVs pointing to their anchor frames. I and P frames are used as anchor frames for block matching. As illustrated in Fig. 2(a), a P frame is always predicted from the closest previous P or I frame and each MB has only one MV referring to the past. To achieve more compression, B frames utilize the closest P or I frames from both the past and the future as anchor frames. Each MB in B frame has up to two MVs point to both future and past anchor frames. The frame sequencing structure is referred to as group of pictures (GOP) in the MPEG protocols. In more advanced video format (e.g. MPEG 4), an MB can have as many as 16 MVs pointing to many reference frames.



(a)

(b)

(c)

Fig. 2. (a) GOP structure for an MPEG 2 video stream. Note that the arrows on top of the frames refer to reference relationship in computing MVs. (b) MVs between adjacent I and P frames can be obtained either directly (e.g. red dotted lines) or indirectly through B frames (e.g. blue dashed lines). (c) Sample MVs overlaid on top of their video frame. Line segments and circles represent MVs and their pointing direction.

B. Modeling Noise in Motion Vectors

If an MB centered at (u_i, v_i) in frame i finds the corresponding position (u_j, v_j) in the anchor frame j through block matching algorithm (BMA), then the resulting l-th MV can be defined as

$$m_l^{i\to j}(u_i, v_i) = \begin{bmatrix} \Delta_u \\ \Delta_v \end{bmatrix} = \begin{bmatrix} u_j - u_i \\ v_j - v_i \end{bmatrix}, \tag{1}$$

where u and v are frame coordinates. For simplicity, we sometimes use $m_l^{i\to j}$ to represent an MV between the two frames. An MB may contain many MVs. Some of them originate from the center of the MB and others may not (e.g. the reverse MV of $m_l^{i\to j}(u_i, v_i)$ is not necessarily located at the center of an MB in frame j).

Although containing image correspondence information, MVs are difficult to use due to noise introduced by BMA, which searches the most similar block in a given range. When video frames contain repetitive patterns, false matches can be generated. This is not a problem for video compression but presents a huge challenge to scene understandings. Sometimes, occlusions and scene changes may cause BMA to fail to find a matching. Say that BMA finds the correct matching with probability p, which is defined as event E_M. It is worth noting that p is also often affected by robot/camera moving speed. To avoid that, we can set frame rate proportional to the moving speed to reduce the variation in p. As observed from data, a regular street driving in urban area often has $p > 0.6$.

Even when a correct matching is found, BMA still has limited accuracy. MPEG 2 and 4 warrant 0.5 and 0.25 pixel accuracy, respectively. When the correct matching is found, this error $\boldsymbol{e}_l^{i\to j} = m_l^{i\to j} - \bar{m}_l^{i\to j}$ can be modeled as a 2D zero mean Gaussian

$$\boldsymbol{e}_l^{i\to j}|E_M \sim N(\mathbf{0}_{2\times 1}, \boldsymbol{\Sigma}), \tag{2}$$

where term $\cdot|E_M$ indicates that this is a conditional distribution, $\bar{m}_l^{i\to j}$ is the true mean of the MV, and covariance matrix $\boldsymbol{\Sigma} = diag\{\sigma^2, \sigma^2\}$ is a diagonal matrix. We set $\sigma = 0.25$ to conservatively capture the 0.5 pixel accuracy for MPEG 2. This accuracy level is sufficient for video presentation. However, due to the small time difference in adjacent frames, the motion parallax can be as small as 2-4 pixels, which leads to large relative error. Compounded with false matches, MVs are too noisy to be directly used for scene understanding.

C. Problem Formulation

To formulate SLAPSE problem, we assume that the intrinsic matrix of the camera is known as K through pre-calibration and the scene is dominated by planes, such as building facade and paved roads. Thus, the understanding of scene structure relies on estimating 3D planes.

Here all the 3D coordinate systems are right hand systems. Let us define

- $\{C_k\}$ as the 3D camera coordinate system (CCS) in 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,
- R_k and $\boldsymbol{t}_k$ as the rotation and translation of $\{C_k\}$ w.r.t. frame $\{C_{k-1}\}$,
- $\boldsymbol{\pi}_{i,k} = [\boldsymbol{n}_{i,k}^{\mathsf{T}}, d_{i,k}]^{\mathsf{T}}$ is the i-th 3D plane in $\{C_k\}$, where $\boldsymbol{n}_{i,k}$ is the plane normal and $d_{i,k}$ is the plane depth, and
- $\tilde{\boldsymbol{\pi}}_{i,k} = \boldsymbol{n}_{i,k}/d_{i,k}$ as the inhomogeneous form of a plane.

Therefore, the problem is defined as below:

Definition 1: Given the set of MVs up to time/frame k, $\{m_l^{i\to j}|i, j \leq k\}$, extract planes, estimate plane equations and camera pose R_k and $\boldsymbol{t}_k$ in each frame.

IV. System Architecture

The SLAPSE problem can be solved using an EKF-based filtering approach as shown in Fig. 3(a). The system takes MVs as the input, and tracks the 3D configuration of planes and camera poses. A key issue of the procedure is how to extract planes from MVs, which is detailed in Fig. 3(b). Let us start with the planar surface extraction.

V. Planar Surface Extraction

Planes are identified through MVs. Given that MVs may have multiple reference frames, we need to merge them to facilitate the plane extraction. Moreover, it is necessary to understand how errors in MVs are accumulated and propagated in the MV merging process.

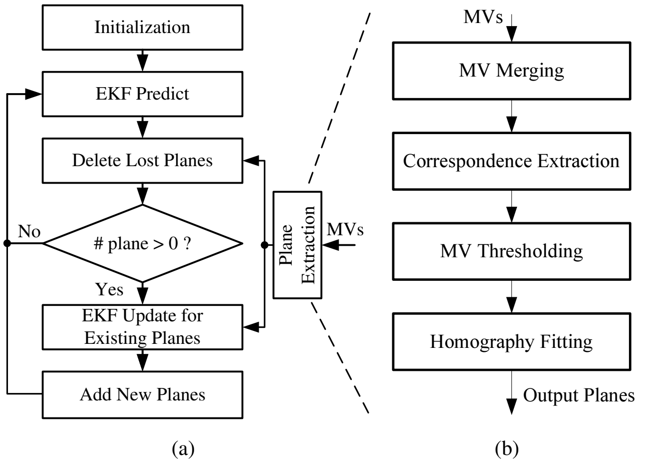


Fig. 3. System diagrams: (a) Overall SLAPSE diagram based on EKF. (b) A blowup view of plane extraction.

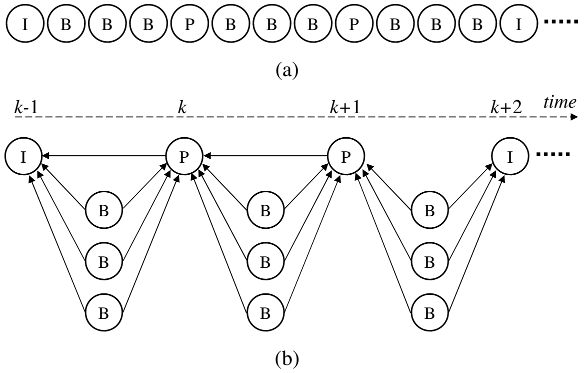


Fig. 4. MVs in B frames are merged into the nearest P and I frames. Arrows indicate the MV referencing directions. (a) A sample GOP. (b) The GOP can be decomposed into IP, PP and PI types.

A. Motion Vector Merging

According to the noise model in Section III-B, an MV represents correct MB correspondence between the current B or P frame and its reference frame with probability p. We name MVs with correct correspondence as in-line MVs (IMVs). From scene understanding point of view, IMVs have limited spatial resolution and relatively high noise. However, IMV set is actually temporally abundant. The adjacent frames differ by 1/30 or 1/25 seconds. If done properly, we can utilize IMV's temporal abundance to further reduce noise level. Since IMV accuracy determines the accuracy of scene structure, it is important to monitor the IMV variance level. Therefore, the subsequent questions are 1) what is the probability that the IMVs exist across multiple frames and 2) how accurate are these IMVs.

We begin with question 1). For a sample GOP in Fig. 4(a), we can draw the MV reference relationship in Fig. 4(b). Interestingly, the continuous frame sequence can be broken into segments with each segment beginning with an I/P frame and ending with the nearest subsequent I/P frame. Segments overlap by sharing common I or P frames. Let n_{B} be the number of B frames in each segment. $n_{\text{B}} = 3$ in Fig. 4. Utilizing these natural segments, we check IMV existence every $n_{\text{B}}+1$ frames as defined by each segment. There are three types of segments according to beginning/ending frame types: IP, PP, and PI. IP and PP share a similar structure: a direct reference between the two and n_{B} indirect references from B frames. PI pairs do not have the direct reference because I frames are not constructed from MBs. Define events E_{IP}, E_{PP}, and E_{PI} for the existence of IMV for an MB across the nearest IP, PP, and PI frames, respectively. We have the following lemma.

Lemma 1: For an MB, the probability of existing at least one IMV across the nearest I/P frame pair is,

$$P(E_{\text{IP}}) = P(E_{\text{PP}}) = 1-(1-p)(1-p^2)^{n_{\text{B}}}, \tag{3}$$

$$P(E_{\text{PI}}) = 1-(1-p^2)^{n_{\text{B}}}. \tag{4}$$

Proof: We can view the MV reference relationship in Fig. 4(b) as a probability graph where each edge has a probability of p that the MV is a correct correspondence. Therefore, for each path passing B frames, the probability that both left and right edges are correct is p^2. Subsequently, the probability that the path is incorrect is $1-p^2$. $\overline{E}_{\text{PI}}$ happens if all paths passing B frames are incorrect. Hence $P(\overline{E}_{\text{PI}}) = (1-p^2)^{n_{\text{B}}}$. Eq. (4) holds. Similarly, we can obtain $P(E_{\text{IP}})$ and $P(E_{\text{PP}})$. ■

Lemma 1 indicates that using B frames can increase the probability of IMV existence. In fact, we often have more than one IMV for each MB. Let us define frame index (also used as time index) variable k and $k+1$ corresponding to an adjacent P/I pair in a segment (see Fig. 4(b)). Define set L_{IMV} as the set of IMVs for the MB. We know that IMVs are from two sources: the direct reference between I or P frames and indirect references from B frames. The error in the former follows $N(\mathbf{0}_{2\times1}, \mathbf{\Sigma})$ in (2) whereas the error in the latter is the summation of two independent 2D Gaussian in (2) and hence follows $N(\mathbf{0}_{2\times1}, 2\mathbf{\Sigma})$. We define event E_D if there exists a correct direct reference and d as the index for the MV. For each MB, we aggregate MVs at I or P frames by minimizing the Mahalanobis distance,

$$m_l^{k+1\rightarrow k}|E_D = \frac{\sqrt{2}m_d^{k+1\rightarrow k} + \sum_{\eta\in L_{\text{IMV}},\eta\neq d} m_\eta^{k+1\rightarrow k}}{\sqrt{2}+|L_{\text{IMV}}|-1}, \tag{5}$$

$$m_l^{k+1\rightarrow k}|\overline{E}_D = \frac{1}{|L_{\text{IMV}}|}\sum_{\eta\in L_{\text{IMV}}} m_\eta^{k+1\rightarrow k}. \tag{6}$$

The aggregation results in the following error distribution:

Lemma 2: The error $\boldsymbol{e}_l^{k+1\rightarrow k} = m_l^{k+1\rightarrow k} - \bar{m}_l^{k+1\rightarrow k}$ of the resulting MV is distributed with zero mean:

$$\boldsymbol{e}_l^{k+1\rightarrow k}|E_* \sim N(\mathbf{0}_{2\times1}, \mathbf{\Sigma}_*|E_*), \tag{7}$$

where condition '$*$' represents IP, PP, and PI pairs, and three conditional covariance matrices are:

$$\mathbf{\Sigma}_{\text{PI}}|E_{\text{PI}} = \left[\sum_{i=1}^{n_{\text{B}}} \frac{2}{i}\begin{pmatrix} n_{\text{B}} \\ i \end{pmatrix}\frac{p^{2i}(1-p^2)^{n_{\text{B}}-i-1}}{1-(1-p^2)^{n_{\text{B}}}}\right]\mathbf{\Sigma}, \tag{8}$$

$$\begin{aligned}\mathbf{\Sigma}_{\text{IP}}|E_{\text{IP}} = \mathbf{\Sigma}_{\text{PP}}|E_{\text{PP}} &= (1-p)\mathbf{\Sigma}_{\text{PI}}|E_{\text{PI}} \\ &+p\left[\sum_{i=0}^{n_{\text{B}}}\frac{2+i}{(i+\sqrt{2})^2}\begin{pmatrix} n_{\text{B}} \\ i \end{pmatrix}p^{2i}(1-p^2)^{n_{\text{B}}-i}\right]\mathbf{\Sigma}. \end{aligned}\tag{9}$$

Proof: See Appendix.A. ■

Remark 1: Actually, both (8) and (9) are decreasing functions of n_{B}. This means that merging MVs from B frames into the nearest I/P frames reduces error variance. This process allows us to exchange the redundant temporary resolution to better spatial resolution.

This allows us to obtain a set of merged MVs which are denoted as $\mathcal{M}^{k+1\rightarrow k} = \{m^{k+1\rightarrow k}\}$ for each adjacent frames $k+1$ and k. Lemmas 1 and 2 ensure IMV existence and derive the corresponding error. A merged MV $m^{k+1\rightarrow k}$ provides a correspondence relationship between an MB in $k+1$ and an MB in k which naturally leads to correspondence extraction step.

B. Correspondence Extraction and MV Thresholding

Define $\boldsymbol{x}_k$ to be the homogeneous form of a point in image k. We represent the motion correspondence by a point pair:

$$\boldsymbol{x}_k = \boldsymbol{x}_{k+1}^c + \begin{bmatrix} m^{k+1\rightarrow k} \\ 0 \end{bmatrix}, \tag{10}$$

where $\boldsymbol{x}_{k+1}^c$ is the center of $m^{k+1\rightarrow k}$'s MB in $k+1$, and $\boldsymbol{x}_k$ is its corresponding position in frame k. Therefore, a set of correspondences between frame k and $k+1$ is obtained:

$$\mathcal{C}^{k+1\rightarrow k} = \{\boldsymbol{x}_k \leftrightarrow \boldsymbol{x}_{k+1}^c : m^{k+1\rightarrow k} \in \mathcal{M}^{k+1\rightarrow k}\}. \tag{11}$$

To reduce the influence of MV noise in plane estimation, we only consider planes with sufficient motion parallax. This is handled by eliminating MVs belonging to the plane at infinity which is defined as $\boldsymbol{\pi}_\infty$.

According to [26], points in $\boldsymbol{\pi}_\infty$ remain still during camera translation, therefore, they can be detected if the camera rotation is eliminated from the images.

For a pair of adjacent frames k and $k+1$, their fundamental matrix is first estimated using correspondence $\mathcal{C}^{k+1\rightarrow k}$. Camera rotation and translation are then decomposed using [27]. We re-project all $\boldsymbol{x}_k$'s to frame $k+1$ using only the rotation matrix, which results in a set of points $\boldsymbol{x}'_{k+1}$.

$$\boldsymbol{x}'_{k+1} = sK({}^{k}_{k+1}R)^{-1}K^{-1}\boldsymbol{x}_k, \tag{12}$$

where s is a scalar, and $({}^{k}_{k+1}R)$ is the matrix that rotates $\{C_k\}$ to $\{C_{k+1}\}$ according to the convention used in [28].

The distance between $\boldsymbol{x}'_{k+1}$ and $\boldsymbol{x}^c_{k+1}$ is calculated, and the MV is considered in $\boldsymbol{\pi}_\infty$ if the distance is below a threshold ϵ_m. Denote the correspondence set for $\boldsymbol{\pi}_\infty$ as

$$\mathcal{C}_\infty^{k+1\rightarrow k} = \{\boldsymbol{x}_k \leftrightarrow \boldsymbol{x}_{k+1}^c : \|\boldsymbol{x}'_{k+1} - \boldsymbol{x}^c_{k+1}\| < \epsilon_m\}, \tag{13}$$

where subscript ∞ means it corresponds to the plane at infinity and $\|\cdot\|$ represents the L_2 norm. Hence the set of correspondences is further reduced to

$$\mathcal{C}_m^{k+1\to k} = \mathcal{C}^{k+1\to k} \setminus \mathcal{C}_\infty^{k+1\to k}, \tag{14}$$

where subscript m means the thresholded correspondence set with sufficient motion parallax.

C. Homography Fitting

With the correspondence set extracted, plane extraction can be performed by verifying the homography relationship. The extraction of planes also helps filter IMVs from the correspondence set.

Consider two adjacent frames (IP, PP or PI) after MV merging and thresholding (Fig. 3(b)). We have the correspondence set $\mathcal{C}_m^{k+1\to k}$. We apply RANSAC framework to extract 2D planes and IMVs. RANSAC first samples a minimum set of correspondences to obtain a homography that represents the coplanar relationship

$$\boldsymbol{x}_k = \lambda H \boldsymbol{x}_{k+1}^c, \tag{15}$$

where H is a 3×3 matrix and λ is a scalar.

Each correspondence provides two equations to (15). Since a homography H has at most 8 degrees of freedom (DoFs), only four correspondences are needed to determine a minimal solution. A normalized direct linear transformation (DLT) can be applied to obtain an initial H (page. 109 of [26]). Then, a correspondence resulting in an error below a given threshold:

$$\|\boldsymbol{x}_k - \lambda H \boldsymbol{x}_{k+1}^c\| < \epsilon_h, \tag{16}$$

is labeled as an inlier to the plane.

To extract multiple planes, RANSAC is applied iteratively until it reaches a given maximum iteration number or there are not enough unlabeled correspondences to form a minimum solution. Denote the correspondence set $\mathcal{C}_{\pi,i}^{k+1\to k}$ for plane π_i (defined by homography H_i) as

$$\mathcal{C}_{\pi,i}^{k+1\to k} = \{\boldsymbol{x}_k \leftrightarrow \boldsymbol{x}_{k+1}^c : \|\boldsymbol{x}_k - \lambda H_i \boldsymbol{x}_{k+1}^c\| < \epsilon_h\}. \tag{17}$$

Hence we obtain a set of N_{k+1} planes with correspondences $\{\mathcal{C}_{\pi,1}^{k+1\to k}, ..., \mathcal{C}_{\pi,N_{k+1}}^{k+1\to k}\}$ from frame k and $k+1$.

Note, if a set of planes with correspondences $\{\mathcal{C}_{\pi,1}^{k\to k-1}, ..., \mathcal{C}_{\pi,N_k}^{k\to k-1}\}$ have been extracted between frames $k-1$ and k, we first run RANSAC to sample the minimum solutions only from MBs of existing planes. Thus every existing plane π_i has a chance to find its corresponding plane correspondence set $\mathcal{C}_{\pi,i}^{k+1\to k}$ in frame $k+1$. Then a regular RANSAC is applied to the remaining correspondences to discover new planes between frames k and $k+1$.

VI. Plane Tracking with EKF

With planes extracted, we can feed them as observations to an EKF framework to estimate the global plane equations and camera poses. An EKF filtering approach usually consists of prediction and update steps.

A. EKF Prediction

In the state space description, we define state vector $\boldsymbol{\mu}_k$ to be consisted of plane equations in inhomogeneous form, camera rotation angles and angular velocity, and camera translation and its velocity in frame k,

$$\boldsymbol{\mu}_k = [\tilde{\boldsymbol{\pi}}_{1,k}^\mathsf{T}, ..., \tilde{\boldsymbol{\pi}}_{N_k,k}^\mathsf{T}, \boldsymbol{r}_k^\mathsf{T}, \boldsymbol{t}_k^\mathsf{T}, \dot{\boldsymbol{r}}_k^\mathsf{T}, \dot{\boldsymbol{t}}_k^\mathsf{T}]^\mathsf{T}, \tag{18}$$

where $\boldsymbol{r} = [\alpha, \beta, \gamma]^\mathsf{T}$ defines the Euler rotation angles in $X'Y'Z'$ order, $\boldsymbol{t} = [t_x, t_y, t_z]^\mathsf{T}$ defines the camera translation w.r.t. previous frame, and $\dot{\boldsymbol{t}}$ defines translation velocity in current frame.

Denote Euler rotation matrix $\bar{R}_k = R(\tau \dot{\boldsymbol{r}}_k)$ in $Y'X'Z'$ order. The state transition of the $i-$th plane equation is

$$\tilde{\boldsymbol{\pi}}_{i,k+1} = \frac{\bar{R}_k^\mathsf{T} \tilde{\boldsymbol{\pi}}_{i,k}}{\tau \dot{\boldsymbol{t}}_k^\mathsf{T} \bar{R}_k^\mathsf{T} \tilde{\boldsymbol{\pi}}_{i,k} + 1}. \tag{19}$$

We assume the camera follows constant angular velocity and linear translation velocity. Hence the state transition is,

$$\begin{cases} \boldsymbol{r}_{k+1} & = \tau \dot{\boldsymbol{r}}_k \\ \boldsymbol{t}_{k+1} & = \tau \dot{\boldsymbol{t}}_k \\ \dot{\boldsymbol{r}}_{k+1} & = \dot{\boldsymbol{r}}_k \\ \dot{\boldsymbol{t}}_{k+1} & = \bar{R}_k \dot{\boldsymbol{t}}_k \end{cases}. \tag{20}$$

B. EKF Update

To utilize rich information from MVs, we do not consider simply making a direct observation of the plane equations. Instead, we use the correspondence sets $\mathcal{C}_{\pi,i}^{k\to k-1}$'s to update the state vectors.

For frame k, the observation of a plane $\boldsymbol{\pi}_{i,k}$ is a set of points $\{\boldsymbol{x}_{k-1}\}$ from $\mathcal{C}_{\pi,i}^{k\to k-1}$. Define rotation matrix $R_k = R(\boldsymbol{r}_k)$ following the $Y'X'Z'$ Euler form. The observation model for plane $\boldsymbol{\pi}_{i,k}$ takes the state vector $\boldsymbol{\mu}_k$ and an additional variable $\boldsymbol{x}_k^c$ as input:

$$\boldsymbol{x}_{k-1} = h(\boldsymbol{\mu}_k, \boldsymbol{x}_k^c) = K[R_k - \boldsymbol{t}_k \tilde{\boldsymbol{\pi}}_{i,k}^\mathsf{T}] K^{-1} \boldsymbol{x}_k^c, \tag{21}$$

where K is the intrinsic matrix of the camera. The Jacobian matrix is computed by taking partial derivatives on $\boldsymbol{\mu}_k$.

Lem. 2 in Sec. V-A provides the error model for the merged MVs, and is applied in setting the noise covariance for the EKF observation.

Note that, since the camera rotation and translation are involved in the observation model for each plane, $\boldsymbol{r}_k$ and $\boldsymbol{t}_k$ are also updated with observations.

C. Deleting and Adding Planes

Similar to landmark management in SLAM, planes have finite lifespan in the continuous video stream. We need to handle the appearance and disappearance of planes in camera views (see Fig. 3(a)).

When transiting from frame k to $k+1$, if $\tilde{\boldsymbol{\pi}}_{i,k}$ has corresponding set $\mathcal{C}_{\pi,i}^{k+1\to k} = \emptyset$ in frame $k+1$, then $\tilde{\boldsymbol{\pi}}_{i,k+1}$ in the state vector and its corresponding dimensions in the state covariance matrix are deleted, before EKF update.

After EKF update in frame k, if a new plane is discovered in frame k, its initialized plane equation and variance are added to the state vector and state covariance matrix.

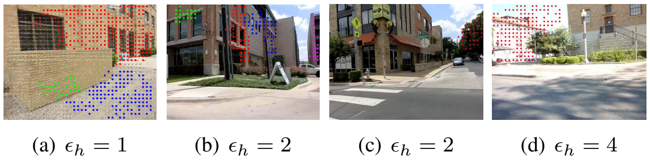
Moreover, since the filter formulation relies purely on planes in EKF updating step, the update is skipped if there are no planes in current state vector. This is not an issue as long as building facades are in the field of view.

VII. Experiments

The proposed method is implemented in C/C++ on a desktop PC. Videos and images are acquired with Casio Ex-ZR200 and Panasonic DMC-ZS3 cameras, with a resolution of 640×480 pixel captured at 30 frames per second. Cameras travel in an urban area at a speed between 25 and 50 kph.

A. Plane Extraction

Fig. 5. Sample images in plane extraction dataset.



(a) $\epsilon_h = 1$ (b) $\epsilon_h = 2$ (c) $\epsilon_h = 2$ (d) $\epsilon_h = 4$

Fig. 6. Example of extracted planes. Dots with different colors indicate different extracted planes. (a-b) show all planes extracted in the frame. (c-d) show two incorrect extractions.

To evaluate the performance of plane extraction, 7 videos of different scenes in MPEG-2 format have been acquired. We sample 50 pairs of adjacent frames from the videos, and manually label planes in images as ground truth. Fig. 5 shows sample thumbnails from the dataset. In this experiment, MVs in $\boldsymbol{\pi}_\infty$ have not been filtered out.

As the error threshold of RANSAC changes, the number of extracted planes and the true positive (TP) rates vary. Tab. I shows how the plane extraction result is influenced by ϵ_h. Note that we restrict the minimum size of an extracted plane to be 20 MBs.

TABLE I

Plane Extract Results w.r.t. ϵ_h

ϵ_h (pixel)	1	2	3	4
# extracted planes	101	183	174	215
TP rate (%)	91.09	83.61	73.56	72.09

Fig. 6 shows four example frames. Dots in the same color indicate an extracted plane. It is clear that the algorithm is able to extract primary planes. However, it may miss some reflective glass/mirror surfaces, such as the leftmost wall in Fig. 6(b), and texture-less surfaces such as the ground. Some false extractions, such as Fig. 6(c), claim trees as a plane due to far depth. In fact, Fig. 6(d) shows the necessity of MV thresholding with $\boldsymbol{\pi}_\infty$ (Sec. V-B), because far field objects tend to mix together when ϵ_h is not tight enough.

B. SLAPSE Results

To evaluate overall system performance, we perform field tests in two sites. Ground truth is manually acquired with meters and Bosch ZLR225 laser distance measurer with an accuracy of ± 1.5 mm.

The 3D estimation is up to scale of the initial camera translation. Sample results from the first site are shown in Fig. 1. It is clear that the system is able to extract dominant planes in the scene.

We project the camera trajectories to $\{C_0\}$ and scale the results by the camera translation in the first step. Comparison with manually measured ground truth is showed in Tab. II. We denote D as the total traveled distance in each site, and a $\hat{}$ on a variable stands for the ground truth value. Denote $\boldsymbol{t}^{0\rightarrow k}$ as the estimated camera translation from frame 0 to k. The mean relative error of camera location is defined as:

$$\epsilon_D = \frac{1}{N}\Sigma_k \frac{\|\boldsymbol{t}^{0\rightarrow k} - \hat{\boldsymbol{t}}^{0\rightarrow k}\|}{\|\hat{\boldsymbol{t}}^{0\rightarrow k}\|}, \tag{22}$$

where N is the total number of tracked frames.

We evaluate the estimated building facades and road segments which appear in the camera scene for at least half a second. The number of evaluated planes in each site are shown in Tab. II. Define the mean absolute error of plane depth ϵ_d and plane orientation ϵ_n as:

$$\epsilon_d = \frac{1}{\Sigma_i N_i}\Sigma_i\Sigma_k|d_{i,k} - \hat{d}_{i,k}|, \tag{23}$$

$$\epsilon_n = \frac{1}{\Sigma_i N_i}\Sigma_i\Sigma_k|\arccos(\boldsymbol{n}_{i,k}^{\mathsf{T}} \cdot \hat{\boldsymbol{n}}_{i,k})|, \tag{24}$$

where N_i is the number of frames plane i appears. Tab. II shows the mean errors for each site, where the depth errors are less than 0.65 meters and orientation errors are less than 7.07 degrees.

TABLE II

SLAPSE results

Site	D (m)	ϵ_D(%)	# planes	ϵ_d (m)	ϵ_n (degs.)
1	42.1	2.9	5	0.61	7.07
2	37.5	5.1	4	0.65	3.26

VIII. Conclusions and Future Work

We explored how to use MVs from video streams for SLAPSE for a mobile robot equipped with a single camera. Using MVs in the MPEG-2 protocol as an example, we established the MV noise models to capture the observation error. We formulated the SLAPSE problem and studied how to extract planes from MVs using planar homography filtering. We then developed an extended Kalman filter (EKF) based approach with planes and robot motion as state variables. We implemented our algorithm using C/C++ on a PC platform, and tested the algorithm in physical experiments in two sites. The results showed that the system is capable of performing robot localization and plane mapping with a relative trajectory error of less than 5.1%.

In the future, we plan to utilize the MVs in the plane at infinity for rotation estimation. We can also detect moving obstacles by group MVs with similar motion. The entire system can be merged under an interactive multi-model (IMM) EKF to improve results and provide a comprehensive navigation solution. Local bundle adjustment can be embedded as a post-processing step to improve plane estimation accuracy. We will also include the plane segmentation through the reprojection of MBs. Also, the MVs can be combined with feature-based approaches and/or other sensors to form hybrid methods.

Appendix

A. Proof of Lemma 2

Let us begin with $\mathbf{\Sigma}_{\text{PI}}$. Denote $\xi = |L_{\text{IMV}}|$ as the number of IMV for the MB. ξ is conformal to binomial distribution $B(n_{\text{B}}, p^2)$,

$$P(\xi = i) = \left(\begin{array}{c} n_{\text{B}} \\ i \end{array} \right) p^{2i}(1-p^2)^{n_{\text{B}}-i}. \tag{25}$$

Event E_{PI} means $\xi \geq 1$. Therefore, we have,

$$P(\xi = i|\xi \geq 1) = \frac{P(\xi = i, \xi \geq 1)}{1 - P(\xi = 0)} = \left(\begin{array}{c} n_{\text{B}} \\ i \end{array} \right) \frac{p^{2i}(1-p^2)^{n_{\text{B}}-i}}{1-(1-p^2)^{n_{\text{B}}}}, \text{ for } i = 1, ..., n_{\text{B}}. \tag{26}$$

Recall that $\mathbf{\Sigma}_{\text{PI}}|E_{\text{PI}} = \text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}})$ where $\text{Var}(\cdot)$ means the covariance of the random rector. Conditioning on the value of ξ, from the property of conditional variance, we know

$$\begin{aligned} &\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}) = \\ &\quad E(\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, \xi)) + \text{Var}(E(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, \xi)) \qquad (27) \\ &\quad = E(\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, \xi)), \qquad (28) \end{aligned}$$

where $E(\cdot)$ means expectation of the random vector. Eq. (28) is true because $\text{Var}(E(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, \xi)) = \mathbf{0}_{2\times 2}$ due to the fact that each $\boldsymbol{e}_l^{k+1\to k}$ is zero mean. From the property of conditional expectation, we have,

$$\begin{aligned} &E(\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, \xi)) = \\ &\quad \sum_{i=1}^{n_{\text{B}}} \text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, \xi) P(\xi = i|\xi \geq 1). \end{aligned} \tag{29}$$

According to (6), $\boldsymbol{e}_l^{k+1\to k}|(E_{\text{PI}}, \xi)$ is an average of i independent Gaussian $N(\mathbf{0}_{2\times 1}, 2\mathbf{\Sigma})$. Hence the resulting vector is still Gaussian with

$$\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, \xi) = \frac{2\mathbf{\Sigma}}{i}. \tag{30}$$

Combining (26-30), we obtain (8).

It is clear that $\mathbf{\Sigma}_{\text{IP}}|E_{\text{IP}}$ and $\mathbf{\Sigma}_{\text{PP}}|E_{\text{PP}}$ share the same value due to the same structure shown in Fig. 4(b). We use $\mathbf{\Sigma}_{\text{IP}}|E_{\text{IP}}$ to show the proof process. Conditioning on the event E_D, we have

$$\begin{aligned} \mathbf{\Sigma}_{\text{IP}}|E_{\text{IP}} &= E(\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, E_D)) \qquad (31) \\ &= \text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, E_D)P(E_D) \\ &\quad + \text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, \bar{E}_D)P(\bar{E}_D) \\ &= \text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, E_D)p + \mathbf{\Sigma}_{\text{PI}}|E_{\text{PI}}(1-p). \qquad (32) \end{aligned}$$

Note that (31) is true because the zero mean property is applied to conditional variance computation (similar to (28)). Also, when $\bar{E}_D$ occurs, $\mathbf{\Sigma}_{\text{IP}}|E_{\text{IP}}$ is reduced to $\mathbf{\Sigma}_{\text{PI}}|E_{\text{PI}}$ according to Fig. 4(b).

To compute $\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, E_D)$, we can further condition on ξ, which is similar to how (8) has been derived. However, there are two different scenarios: the first is that (25) becomes

$$P(\xi - 1 = i) = \left(\begin{array}{c} n_{\text{B}} \\ i \end{array} \right) p^{2i}(1-p^2)^{n_{\text{B}}-i} \tag{33}$$

and we do not need to use the conditional binomial defined in (26) because E_D means $\xi - 1 \geq 0$ is always true. Consequently, (29) is modified as

$$\begin{aligned} &E(\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{IP}}, E_D, \xi)) = \\ &\quad \sum_{i=0}^{n_{\text{B}}} \text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{IP}}, E_D, \xi) P(\xi - 1 = i). \end{aligned} \tag{34}$$

The second difference is the fact that we employ (5) to aggregate heterogeneous Gaussian distributions which include one error vector in $N(\mathbf{0}_{2\times 1}, \mathbf{\Sigma})$ and $\xi - 1$ error vectors in $N(\mathbf{0}_{2\times 1}, 2\mathbf{\Sigma})$. Therefore, (30) is changed to the following,

$$\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{IP}}, E_D, \xi) = \frac{2+i}{(\sqrt{2}+i)^2}\mathbf{\Sigma} \tag{35}$$

because (5) is just a linear combination of independent Gaussian distributions. Combining these equations, we obtain (9).

Acknowledgement

Thanks for Y. Lu, J. Lee, M. Hielsberg, Z. Gui, S.-H. Mun, S. Jacob, M. Midori Hirami, P. Peelen, and A. Tuan for their inputs and contributions to the NetBot Laboratory, Texas A&M University.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[2] A. Davison, L. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[3] P. Jensfelt, D. Kragic, J. Folkesson, and M. Bjorkman, "A framework for vision based bearing only 3d slam," in *IEEE International Conference on Robotics and Automation*, Orlando, Florida, May 2006.

[4] E. Ead and T. Drummond, "Scalable monocular slam," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, New York, NY, Jun. 2006, pp. 469–476.

[5] A. Gil, O. Mozos, M. Ballesta, and O. Reinoso, "A comparative evaluation of interest point detectors and local descriptors for visual slam," vol. 21, no. 6, pp. 905–920, 2010.

[6] P. Smith, I. Reid, and A. Davison, "Real-time monocular slam with straight lines," in *British Machine Vision Conference (BMVC)*, Sep. 2006, pp. 17–26.

[7] T. Lemaire and S. Lacroix, "Monocular-vision based slam using line segments," in *IEEE International Conference of Robotics and Automation*, Roma, Italy, April 2007, pp. 2791–2796.

[8] E. Eade and T. Drummond, “Edge landmarks in monocular slam,” in *British Machine Vision Conference (BMVC)*, Sep. 2006, pp. 7–16.

[9] G. Zhang and I. Suh, “Sof-slam: Segments-on-floor-based monocular slam,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 2010.

[10] ——, “Building a partial 3d line-based map using a monocular slam,” in *IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011.

[11] A. Flint, C. Mei, I. Reid, and D. Murray, “Growing semantically meaningful models for visual slam,” in *IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, Jun. 2010, pp. 467–474.

[12] D. Rao, S. Chung, and S. Hutchinson, “Curveslam: An approach for vision-based navigation without point features,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal, Oct. 2012.

[13] A. Gee, D. Chekhlov, A. Calway, and W. Mayol-Cuevas, “Discovering higher level structure in visual slam,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 980–990, Oct. 2008.

[14] R. Wang and T. Huang, “Fast camera motion analysis in mpeg domain,” in *International Conference on Image Processing*, vol. 3, pp. 691–694.

[15] L. Favalli, A. Mecocci, and F. Moschetti, “Object tracking for retrieval applications in mpeg-2.”

[16] F. Vella, A. Castorina, M. Mancuso, and G. Messina, “Digital image stabilization by adaptive block motion vectors filtering,” *IEEE Transactions on Consumer Electronics*, vol. 48, no. 3, pp. 796–801, 2002.

[17] A. Argiles, J. Civera, and L. Montesano, “Dense multi-planar scene estimation from a sparse set of images,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, Sep. 2011, pp. 4448–4454.

[18] R. Newcombe and A. Davison, “Live dense reconstruction with a single moving camera,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, Jun. 2010, pp. 1498–1505.

[19] A. Bruhn, J.Weickert, and C. Schnörr, “Lucas/kanade meets horn/schunk: combining local and global optical flow methods,” *International Journal of Computer Vision*, vol. 61, no. 3, p. 211231, 2005.

[20] D. Song, H. Lee, J. Yi, and A. Levandowski, “Vision-based motion planning for an autonomous motorcycle on ill-structured roads,” *Autonomous Robots*, vol. 23, no. 3, pp. 197–212, Oct. 2007.

[21] D. Song, H. Lee, and J. Yi, “On the analysis of the depth error on the road plane for monocular vision-based robot navigation,” in *The Eighth International Workshop on the Algorithmic Foundations of Robotics (WAFR), Dec. 7-9, 2008, Guanajuato, Mexico*, 2008.

[22] J. Zhang and D. Song, “On the error analysis of vertical line pair-based monocular visual odometry in urban area,” in *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, USA, Oct. 11-15*, 2009.

[23] ——, “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.

[24] H. Li, D. Song, Y. Lu, and J. Liu, “A two-view based multilayer feature graph for robot navigation,” in *IEEE International Conference on Robotics and Automation (ICRA), St. Paul, Minnesota*, May 2012.

[25] Y. Lu, D. Song, Y. Xu, A. Perera, and S. Oh, “Automatic building exterior mapping using multilayer feature graphs,” in *IEEE International Conference on Automation Science and Engineering Madison, Wisconsin, USA*, Aug. 2013.

[26] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[27] V. Rabaud, “Vincent’s Structure from Motion Toolbox,” http://vision.ucsd.edu/ vrabaud/toolbox/.

[28] J. Craig, *Introduction to Robotics Mechanics and Control (Third Edition)*. Pearson Education, Upper Saddle River, New Jersey, 2005.