Kai Ki LEE, Ying Kin YU, Kin Hong WONG,"Recovering camera motion from points and lines in stereo images: A recursive model-less approach using trifocal tensors",17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, (SNPD 2017, Kanazawa, Ishikawa, Japan, 26-29 June 2017.

Recovering camera motion from points and lines in stereo images: A recursive model-less approach using trifocal tensors

Kai Ki Lee
Department of Information Engineering
The Chinese University of Hong Kong
Hong Kong
kklee6@ie.cuhk.edu.hk

Ying Kin Yu* Kin Hong Wong
Department of Computer Science and Engineering
The Chinese University of Hong Kong
*Hong Kong
ykyu.hk@gmail.com, khwong@cse.cuhk.edu.hk

***Abstract*— Estimating the 3-D motion of a moving camera from images is a common task in robotics and augmented reality. Most existing marker-less approaches make use of either points or lines. Taking the advantages of both kinds of features in an unknown environment is more attractive due to their availability and differences in characteristics. A novel model-less method is presented in this paper to tackle the 3-D motion tracking problem. Two Bayesian filters, one for point measurements while another for line measurements, are embedded in the Interacting Probabilistic Switching (IPS) framework. They compensate for the weaknesses in one another by utilizing both kinds of features in the stereo images. The proposed method is able to obtain the 3-D motion given as little as two line or two point correspondences in consecutive images with the use of multiple trifocal tensors. Our method outperformed two recent methods in terms of accuracy and the problem of drifting was very little in real scenarios.**

Keywords—Camera motion estimation, Pose tracking, Stereo Vision, Kalman filtering, Trifocal tensor

I. INTRODUCTION

Computation of the position and orientation of a moving camera is a common task in robot navigation and augmented reality. It is particularly challenging if such a piece of pose information is required to be measured at high speed in an unknown environment only with 2-D images as input. In this work, we have made significant improvements over the latest model-less 3-D motion tracking algorithm in [11]. Instead of using only point features in stereo images, we utilize both points and straight lines in the 3-D motion recovery process. A novel way to recover 3-D pose with line features in the images and trifocal tensors is proposed. The interacting probabilistic switching (IPS) framework is devised to enable the algorithm work with a mixture of point and line features. The algorithm operates in a recursive manner with the use of multiple Bayesian filters. Our methods are classified as model-less, as the scene model is not reconstructed and the 3-D structure of the scene is not known beforehand. No 3-D information about the scene structure is known in advance.

Most previous camera motion tracking algorithms, no matter whether they require prior knowledge about the scene, are specific to either point [3, 6, 9, 10, 11, 12, 25, 36] or line [1, 5, 13, 15, 16, 17, 26] features but are unable to handle both. For instances, Persson et. al. [30] developed a stereo visual odometry system on the basis of monocular techniques using point features. Silva et. al. [32] described a probabilistic approach to estimate the egomotion from calibrated stereo cameras in vehicles. A dense probabilistic 5-D camera motion estimation algorithm is combined with a sparse keypoint-based method. Zhang et. al. [29] proposed the Bernoulli filter to address stereo visual odometry using SURF features.

To consider multiple cues in a tracking process, probabilistic techniques [24, 27] are widely used. Most of these methods are model-based. It means that they require prior knowledge about the 3-D model of the scene. For example, Comport et. al. [15] presented a virtual visual servoing framework for the estimation of camera pose that exploits various 3-D geometrical primitives including straight lines, circles, cylinders and spheres. 3-D model of the scene is needed. Pressigout and Marchand [2] devised another model-based approach that integrates texture information into traditional pose estimation algorithm using only line features. Ramalingam et. al. [18] proposed a minimal pose estimation algorithms using points and lines with a known 3-D model of a city for geo-localization. Koletschka et. al. [31] explored the combination of point and line features to compute camera motion between consecutive stereo frames. 3-D structure of the scene is reconstructed on the fly. Rother [35] propsoed a linear method to simultaneously reconstruct 3-D points, lines, planes and cameras from multiple views with the assumption that a reference plane is visible in the views. Some other researches considered both points and lines to compute the trifocal tensor and in turn reconstruct the scene structure [21] [22]. However, their focus is not on the recursive estimation of the 3-D motion.

The contributions of this paper are as follows:

Taking the advantages of both point and line features. The proposed approach takes the advantages of both point features and straight lines in stereo images. Our hybrid method is able to operate under most realistic conditions. It is not the case for existing approaches that depend solely either on points or edges. To the best of the our knowledge, state-of-the-

art techniques for recovering camera motion from images without reconstructing the 3-D scene structure are, however, either based solely on point or line features. [3, 11, 12]. Efficient trackers that exploit both points and lines require 3-D models in advance [2, 4, 18]. As lines in images can often be computed with great precision and point features are commonly available in the scene, our method can achieve a higher accuracy than either kinds of the algorithms that only makes use of points or lines alone.

Eliminating the step of reconstructing the 3-D scene structure. Our algorithm is model-less and the 3-D model is not reconstructed even with no prior information about the scene. This is achieved by integrating the trifocal tensors into the Kalman filtering and IPS framework. This characteristic improves the accuracy as the trifocal constraint can stabilize the solution of the recovered motion.

Exploiting the strengths of a stereo camera. As stereo images are utilized in our method, the problems of monocular algorithms such as scale factor, observability and instability during the initialization stage, can be avoided with the help of the trifocal constraint. The trifocal tensors are arranged in a special way to lock up the features in every four views as in Fig. 4. Trifocal constraints are applied to transfer points and lines in the measuremnt models. Our algorithm is able to operate with two line or two point correspondences in consecutive stereo frames.

Experimental results show that our approach is more stable and accurate than an existing point-based approach [11], and another recent method using straight lines [5]. We have found that our algorithm can generate impressive results at high speed from long image sequences with small drifting.

The rest of this paper is organized as follows. In Section II the problem of 3-D motion estimation is defined. In Section III, an overview of the proposed motion tracking algorithm is given. In Section IV, the details of our Bayesian filters and the interacting probabilistic switching framework are described. In Section V, experimental results of our method with real images and synthetic data are shown.

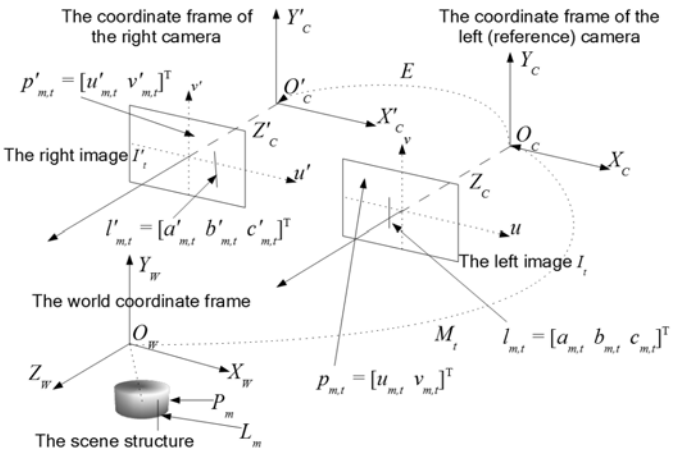

Fig. 1: The geometric model used in this article.

II. Problem Modeling

Fig. 1 illustrates the setup of the imaging system. Here $I_1, I_1^{'}$ will be used throughout to designate the reference images taken by the left camera and right camera at the first time-step t=1, respectively. Likewise, $I_t, I_t^{'}$ are the current images taken by the left and right camera at time t. Points and lines are extracted from images. Point $p_{m,t}$ and line $l_{m,t}$, where $m=1,2..$, are extracted from image I_t. Similarly, $p_{m,t}^{'}$ and $l_{m,t}^{'}$ are extracted from $I_t^{'}$. Points $p_{m,t}$ and $p_{m,t}^{'}$ are the projection of the 3-D scene point P_m on the left and right view, respectively. Lines $l_{m,t}$ and $l_{m,t}^{'}$ are the projection of the 3-D straight line L_m in the scene on the left and right image, respectively. $a_{m,t}$, $b_{m,t}$ and $c_{m,t}$ are respectively the parameters a, b and c of a straight line $ax+by+c$ in general form.

Upper triangular matrix K encodes a camera's intrinsic parameters. Matrix E represents the extrinsic parameters of the rigid transformation between the system cameras. Both matrices are found by the calibration utility in [14]. Matrix M_t, or equivalently the 6-dimensional twist vector ξ_t, encodes the pose information that transforms the 3-D structure from the world frame to the reference camera at time instance t. $\widetilde{\xi}_t$ is the matrix form of ξ_t with the upper 3×3 component as a skew-symmetric matrix.

$$\xi_t = [x_t \quad y_t \quad z_t \quad \alpha_t \quad \beta_t \quad \gamma_t]^T \tag{1}$$

$$\widetilde{\xi}_t = \begin{bmatrix} 0 & -\gamma_t & \beta_t & x_t \\ \gamma_t & 0 & -\alpha_t & y_t \\ -\beta_t & \alpha_t & 0 & z_t \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2}$$

x_t, y_t and z_t are the components that determines the amount of translation along the axes. α_t, β_t, γ_t are the rotations about the x, y and z axis respectively. $\widetilde{\xi}_t$ can be converted to M_t with the exponential map. It is given as:

$$M_t = e^{\widetilde{\xi}_t} = I + \widetilde{\xi}_t + \frac{(\widetilde{\xi}_t)^2}{2!} + \frac{(\widetilde{\xi}_t)^3}{3!} + \cdots \tag{3}$$

The geometric relationships between the 3-D point $P_m = [x_m, y_m, z_m, 1]^T$ in the scene and its projection $\widetilde{p}_{m,t}$ on the left view and $\widetilde{p}_{m,t}^{'}$ on the right view in normalized form are obtained as:

$$\widetilde{p}_{m,t} = [\widetilde{u}_{m,t}, \widetilde{v}_{m,t}, \widetilde{w}_{m,t}]^T = KP_m \tag{4}$$

$$\widetilde{p}_{m,t}^{'} = [\widetilde{u}_{m,t}^{'}, \widetilde{v}_{m,t}^{'}, \widetilde{w}_{m,t}^{'}]^T = KEM_tP_m \tag{5}$$

Suppose the 3-D line L_m is composed of two 3-D end-points. The 2-D projections of the end-points on the left view are $\overline{p}_{m,t,1}$ and $\overline{p}_{m,t,2}$. Similarly, the 2-D projections of the end-points on the right view are $\overline{p}_{m,t,1}^{'}$ and $\overline{p}_{m,t,2}^{'}$. $l_{m,t}$ and $l_{m,t}^{'}$ can be found by the cross product of their corresponding projected end-points:

$$l_{m,t} = \bar{p}_{m,t,1} \times \bar{p}_{m,t,2} \quad (6)$$

$$l'_{m,t} = \bar{p}'_{m,t,1} \times \bar{p}'_{m,t,2} \quad (7)$$

f is the focal length and $[s_u \quad s_v]$ denotes the principal point. The normalized form $\tilde{l}_{m,t}$ and $\tilde{l}'_{m,t}$ of line $l_{m,t}$ on the left view and $l'_{m,t}$ on the right view are respectively given by

$$\tilde{l}_{m,t} = \begin{bmatrix} \tilde{a}_{m,t} \\ \tilde{b}_{m,t} \\ \tilde{c}_{m,t} \end{bmatrix} = \begin{bmatrix} a_{m,t}/f \\ b_{m,t}/f \\ (c_{m,t} + s_u a_{m,t} + s_v b_{m,t})/f^2 \end{bmatrix} \quad (8)$$

$$\tilde{l}'_{m,t} = \begin{bmatrix} \tilde{a}'_{m,t} \\ \tilde{b}'_{m,t} \\ \tilde{c}'_{m,t} \end{bmatrix} = \begin{bmatrix} a'_{m,t}/f \\ b'_{m,t}/f \\ (c'_{m,t} + s_u a'_{m,t} + s_v b'_{m,t})/f^2 \end{bmatrix} \quad (9)$$

$\theta_{m,t}$ and $\lambda_{m,t}$ are the slope and polar radius of $l_{m,t}$. Likewise, $\theta'_{m,t}$ and $\lambda'_{m,t}$ are the slope and polar radius of $l'_{m,t}$.

$$\begin{cases} \theta_{m,t} = tan^{-1}(\tilde{a}_{m,t}/-\tilde{b}_{m,t}) \\ \theta'_{m,t} = tan^{-1}(\tilde{a}'_{m,t}/-\tilde{b}'_{m,t}) \end{cases} \quad (10)$$

$$\begin{cases} \lambda_{m,t} = \dfrac{\tilde{b}_{m,t}\tilde{c}_{m,t}}{\tilde{b}_{m,t} sin(\theta_{m,t}) + \tilde{a}_{m,t} cos(\theta_{m,t})} \\ \lambda'_{m,t} = \dfrac{\tilde{b}'_{m,t}\tilde{c}'_{m,t}}{\tilde{b}'_{m,t} sin(\theta'_{m,t}) + \tilde{a}'_{m,t} cos(\theta'_{m,t})} \end{cases} \quad (11)$$

The aim of the proposed method is to find the 3-D camera motion ξ_t recursively given only the 2-D image measurements $p_{m,t}$, $p'_{m,t}$, $l_{m,t}$ and $l'_{m,t}$.

III. Outline of The Algorithm

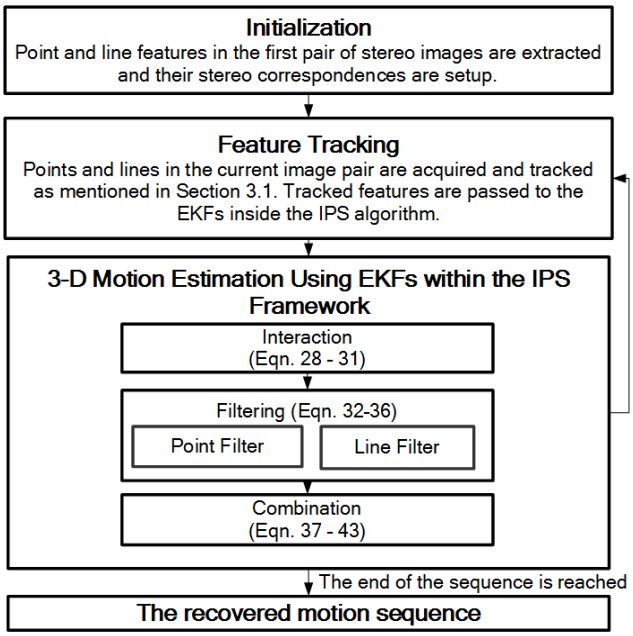

Fig. 2: A summary of the proposed 3-D motion tracking method that uses multiple extended Kalman filters within the interacting probabilistic switching framework.

An outline of the proposed tracking method is shown in Fig. 2. Features are extracted from the first image pair at the beginning, i.e. at t=1. The feature correspondences between two views of the reference image pair are then established. Then we enter into the main filtering loop by setting the current time-step to t=2. From t=2, matched correspondences are tracked from one stereo pair to the next and the details will be explained in Section III.A. The trifocal tensors are arranged in a special way to lock up the features. Trifocal constraints are applied to transfer points and lines in the measuremnt models. Their arrangement will be discussed in Section III.B. The 2-D features are then passed to the extended Kalman filters (EKFs) within the proposed interacting probabilistic switching (IPS) framework. The IPS leads us to a flexible framework with a two-channel filter structure. The two channels, one for the point measurements and the other for the line measurements, are complementary in that each compensates for the weaknesses in one another, as points and straight lines bear distinguishing properties. The IPS algorithm makes use of the residual information from the channels to evaluate the contribution of each channel to the final computed pose using a probability framework. This provides a mechanism for the algorithm to combine the pose deduced from point and line features elegantly. The mathematical details will be presented in Section IV.

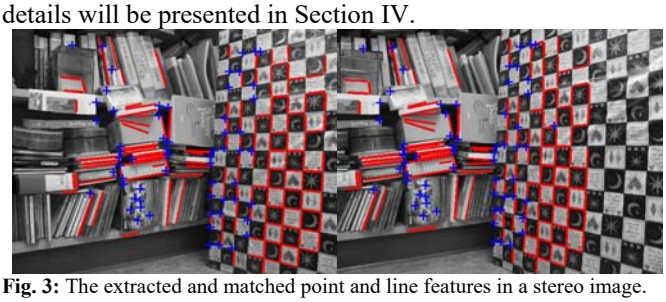

Fig. 3: The extracted and matched point and line features in a stereo image.

A. Feature Extraction, Matching and Tracking

Existing techniques are applied to extract, match and track features in the stereo image sequence. Our method uses two kinds of features at the same time. One of them is a texture-based feature, which is represented as an interest point in the space. To extract point features, the FAST detector [33] is employed. The interest points in a stereo pair are then matched using BRIEF descriptor [34], together with the brute force classifier based on the k-nearest neighbor (KNN) algorithm. Epipolar constraints [8] is added to enhance the matching robustness. For efficiency, the brute force classifier is templated on the distance metric called Hamming distance. The Kanade-Lucas-Tomasi (KLT) tracker described in [28] is then used to track the matched points from frames to frames in the images.

Another type of features used in our system is the line feature. Lines are first detected in the reference image pair by a conventional edge detection algorithm such as the Canny algorithm. Then, the line correspondences between two views of the reference image pair are established. We follow the method by Schmid and Zisserman [19] using mean-standard deviation line descriptors (MSLD) [20] to establish line-to-line matches in a stereo image. Matched correspondences are tracked from one frame to the next using Lucas-Kanade optical flow applied to the end-points as described in [7]. Fig.

3 shows the extracted point features and straight lines in a sample stereo image.

B. Defining the Trifocal Tensor Relations

In our algorithm, we propose to use several trifocal tensors [8] to set up constraints among four views, i.e. I_1, $I_1^{'}$, I_t and $I_t^{'}$, for motion tracking as illustrated in Fig. 4.

The trifocal tensor introduced by Hartley in [8] is a 3x3x3 array of numbers that incorporates all projective geometric relationships among three views. It is independent of the scene structure, and depends only on the relative pose among the three views and their intrinsic calibration parameters. The trifocal tensor actually relates the coordinates of corresponding points or lines in three views. This gives rise to the trifocal constraint. With such a constraint, point and line transfer among views becomes possible.

The first triplet consists of the images of the first stereo image pair I_1, $I_1^{'}$ in an image sequence and the current image I_t taken by the left camera. Hence the first trifocal tensor T constitutes the set of images I_1, $I_1^{'}$ and I_t. The second trifocal tensor $T^{'}$ and the third trifocal tensor T^{*} are formed in a similar manner such that $T^{'}$ constitutes images I_1, I_t and $I_t^{'}$, and T^{*} comprises the image set $I_1, I_1^{'}$ and $I_t^{'}$.

Three trifocal tensors are used to form two sets of geometric constraints. The first set is composed of the trifocal tensor pair T and $T^{'}$. It helps the system ensure the consistency of slopes and displacements of the line features in every four views. The second set comprises the trifocal tensor pair T and T^{*}, and is used to lock up the positions of point features in every four images.

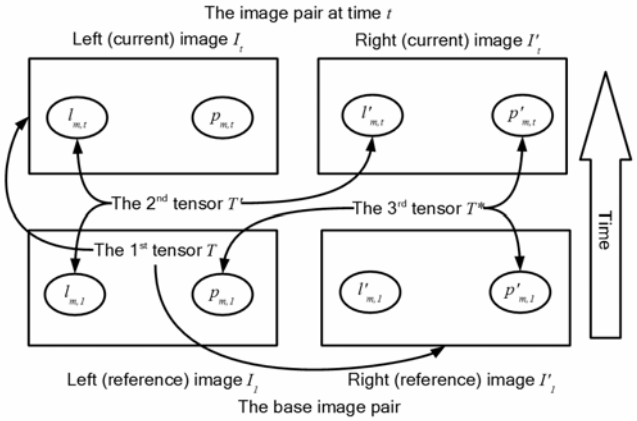

Fig. 4: A visual representation of the trifocal tensor arrangement. The first tensor T involves both points and lines in images I_1, $I_1^{'}$ and I_t. The second tensor $T^{'}$ involves only lines $l_{m,1}$, $l_{m,t}$ and $l_{m,t}^{'}$. The third tensor T^{*} involves only points $p_{m,1}$, $p_{m,1}^{'}$ and $p_{m,t}^{'}$.

With the normalized 2-D coordinates, T, $T^{'}$ and T^{*} can be expressed in tensor notation as:

$$T_i^{jk} = a_i^{'j} a_4^k - a_4^{'j} a_i^k \tag{12}$$

$$T_i^{'jk} = a_i^{''j} a_4^{'k} - a_4^{''j} a_i^{'k} \tag{13}$$

$$T_i^{*jk} = a_i^{''j} a_4^{\ k} - a_4^{''j} a_i^{\ k} \tag{14}$$

a_i^j, $a_i^{'j}$, $a_i^{''j}$ are respectively the extrinsic parameters E of the stereo system, the elements of the upper 3-by-4 component of the rigid transformation matrix M_t and the matrix product EM_t such that $E = \left[a_i^j\right]$, $[I_{3\times3} \quad 0_{3\times1}]M_t = \left[a_i^{'j}\right]$ and $EM_t = \left[a_i^{''j}\right]$.

These tensors are used in the form of transfer formulae in the EKFs that will be discussed in Section IV. Two of them are the point transfer formulae and can be written in the tensor notation as

$$[U_{m,t}]^k = [U_{m,1}]^i [l_{m,1}^*]_j T_i^{jk} \tag{15}$$

$$[U_{m,t}^{'}]^k = [U_{m,1}]^i [l_{m,1}^*]_j T_i^{*jk} \tag{16}$$

where $U_{m,t}$ and $U_{m,t}^{'}$ are respectively the normalized homogenous general form of $p_{m,t}$ and $p_{m,t}^{'}$.

$$U_{m,t} = \left[u_{m,t}/f \quad v_{m,t}/f \quad 1\right]^T \tag{17}$$

$$U_{m,t}^{'} = \left[u_{m,t}^{'}/f \quad v_{m,t}^{'}/f \quad 1\right]^T \tag{18}$$

$l_{m,t}^*$ is a line passing through the m^{th} feature point $p_{m,t}^{'}$ in the image pair of the right view at time t. $l_{m,t}^*$ is constructed by first finding the epipolar line $l_{m,t}^*$ through the coordinates $U_{m,t}^{'}$. Then $l_{m,t}^*$ is joining $U_{m,t}^{'}$ and perpendicular to the epipolar line.

$$l_{m,t}^* = \left[l_{m,t,1} \quad l_{m,t,2} \quad l_{m,t,3}\right]^T = e_{12} \times U_{m,t}^{'} \tag{19}$$

$$l_{m,t}^* = \left[l_{m,t,2} \quad -l_{m,t,1} \quad -u_{m,t}^{'} l_{m,t,2}/f + v_{m,t}^{'} l_{m,t,1}/f\right] \tag{20}$$

where e_{12} is an estimate of the epipole observed from the right camera and is calculated in the initialization step.

The normalized homogenous general form of $l_{m,1}$, $l_{m,1}^{'}$, $l_{m,t}$ and $l_{m,t}^{'}$ are $\widetilde{L}_{m,1}$, $\widetilde{L}_{m,1}^{'}$, $\widetilde{L}_{m,t}$ and $\widetilde{L}_{m,t}^{'}$, respectively. The rests are line transfer formulae and are in tensor notation as

$$[\widetilde{L}_{m,1}]^i = [\widetilde{L}_{m,t}]^k [\widetilde{L}_{m,1}]_j T_i^{jk} \tag{21}$$

$$[\widetilde{L}_{m,1}]^i = [\widetilde{L}_{m,t}^{'}]^k [\widetilde{L}_{m,t}]_j T_i^{'jk} \tag{22}$$

C. Feature Replacement

A simple scheme catering for the replacement of 2-D features into the scene is used. For each set of geometric constraint, it has its own pool of feature correspondence and reference image pair. Correspondences that are extracted from the set of views related by the trifocal tensors are fed into the filters to find the innovation residual, which will be discussed in the next section. If the number of available features are below two or any greater-than-2 integer k_c defined by the user, the views at the current time-step will be set as the new reference image pair and the tracker will be bootstrapped. In other words, the features extracted the reference image pair $I_1, I_1^{'}$ are assumed to be static and observable throughout the sequence before bootstrapping.

IV. The Interacting Probabilistic Switching Framework and the Extended Kalman Filters

The proposed algorithm consists of two extended Kalman filters (EKFs) embedded within the interactive probabilistic switching (IPS) framework. One EKF takes point features as measurement input, i.e. the point filter, while the other one uses line features as measurements, i.e. the line filter. The IPS provides a probabilistic framework for the EKFs to interact. With the combination of estimates from the EKFs, the velocity of the camera system, and in turn its position and orientation, can be estimated.

A. The Dynamic Systems

The state vector $s_t(i)$ of the i^{th} EKF, $i \in \{1,2\}$, representing the pose is defined as:

$$s_t(i) = \begin{bmatrix} \dot{x}_t & \dot{y}_t & \dot{z}_t & \dot{\alpha}_t & \dot{\beta}_t & \dot{\gamma}_t \end{bmatrix}^T \quad (23)$$

$\dot{x}_t$, $\dot{y}_t$, $\dot{z}_t$ are the amounts of translational velocities along the axes. $\dot{\alpha}_t, \dot{\beta}_t, \dot{\gamma}_t$ are the angular velocities of the motion on the x, y and z axis respectively. $\tilde{s}_t(i)$ is the matrix form of $s_t(i)$ according to equations (1) and (2). M_t can be regarded as an integral of velocity from the initial time to the current moment. Let η_t be the zero mean Gaussian noise and the dynamic system equations of the filters are:

$$M_t = M_{t-1} exp(\tilde{s}_t(i)) \quad (24)$$

$$s_t(i) = s_{t-1}(i) + \eta_t \quad (25)$$

It is assumed that the sampling rate of the measurements is high with small object motion between successive images.

B. The Measurement Models

The two EKFs that make use of different kinds of features as inputs have distinct measurement models. For the point filter, its measurement model is defined as:

$$\varepsilon_t^* = g_t^*(M_t) + \nu_t^* \quad (26)$$

where g_t^* is the $4N^* \times 1$-output trifocal tensor point transfer functions (15) (16) in Section III.B ν_t^* is a $4N^* \times 1$ vector representing zero-mean Gaussian noise imposed on the images captured. Here N^* is the number of point features extracted from scene. Using the point measurements ε_1^* from the first stereo image pair in the sequence, the pose information M_t, together with the extrinsic parameters of the stereo rig E, the estimated coordinates of the feature points at current time *t* can be computed.

The measurement model for the line filter, which relates the pose M_t and the measurements $[\varepsilon_t, \varepsilon_t']$ taken from the system, is defined as:

$$\begin{cases} \varepsilon_1 = g_t(M_t, \varepsilon_1', \varepsilon_t) + \nu_t \\ \varepsilon_1 = g_t'(M_t, \varepsilon_t, \varepsilon_t') + \nu_t \end{cases} \quad (27)$$

where ε_t is the line measurements from the image taken by the left camera at time-step t while ε_t' is the line measurements on the right view at time t. ν_t is a $2N \times 1$ vector representing zero-mean Gaussian noise imposed on the images captured. N is the number of extracted line features. g_t and g_t' are the $2N \times 1$ -output line transfer functions (21)(22) in Section III.B with the results converted to the form of slope and polar radius using equations (10) (11). A line $l_{m,1}$ can be transformed into slope $\theta_{m,1}$ and polar radius $\lambda_{m,1}$ with these formulae.

The line transfer function works in a reverse manner. Line measurements from the current image of the left camera ε_t are projected back to the first left view with the predicted motion M_t, the rigid transformation E, and line measurements in the first image from the right camera ε_1' by solving the non-linear measurement equation. The estimation of line measurements on the left reference view ε_1 can also be computed from a quartette consisting the straight lines from the current left and right images, i.e. ε_t and ε'_t, the predicted motion M_t and the rigid transformation matrix E.

C. Interaction of the IPS Algorithm

The basic IPS algorithm consists of several steps, which can be visualized in Fig. 2. Firstly, the likelihood of each filter $\lambda_{t-1}(i)$ is updated according to the 2×2 switching matrix $J(i, j)$

$$\lambda_t^*(i) = \sum_j J(i,j) \lambda_{t-1}(j) \quad (28)$$

where $\lambda_t^*(i)$ is the likelihood probability of the filter after interacting with the switching matrix $J(i, j)$. $J(i, j)$ denotes the probability of switching from filter i to filter j. The switching matrix is set by assuming that the algorithm relies on either one kind of features for an extended period of time with an occasional transition to another type of feature measurements. The other formulae for filter interaction are

$$\hat{s}_{t-1,t-1}^*(i) = \frac{\sum_j J(i,j) \lambda_{t-1}(j) \hat{s}_{t-1,t-1}(j)}{\lambda_t^*(i)} \quad (29)$$

$$\hat{P}_{t-1,t-1}^*(i) = \frac{\sum_j J(i,j) \lambda_{t-1}(j) \left(P_{t-1,t-1}(j) + \delta_{t-1}(i,j) \right)}{\lambda_t^*(i)} \quad (30)$$

$$\delta_t(i,j) = \left[\hat{s}_{t-1,t-1}(j) - \hat{s}_{t-1,t-1}(i) \right] \left[\hat{s}_{t-1,t-1}(j) - \hat{s}_{t-1,t-1}(i) \right]^T \quad (31)$$

where $\hat{s}_{t-1,t-1}^*(i)$ and $\hat{P}_{t-1,t-1}^*(i)$ are respectively the state estimates and its covariance of filter i after the interaction with the switching matrix $J(i, j)$. They are then passed to the EKFs for prediction and smoothing with their own measurements in the current time-step.

D. Filtering and Smoothing

The prediction equations of filter i for calculating the optimal estimates are

$$\hat{s}_{t,t-1}^*(i) = \hat{s}_{t-1,t-1}^*(i) \quad (32)$$

$$\hat{P}_{t,t-1}^*(i) = \hat{P}_{t-1,t-1}^*(i) + Q_t \quad (33)$$

$\hat{s}^*_{t,t-1}(i)$ is the estimates of state $s_t(i)$ after prediction. $\hat{P}^*_{t,t-1}(i)$ is a 6×6 covariance matrix of $\hat{s}^*_{t,t-1}(i)$. Q_t is the covariance of the noise terms η_t

The update equations of filter i for the corrections of estimates:

$$\hat{s}^*_{t,t}(i) = \hat{s}^*_{t,t-1}(i) + W(i)r_t(i) \quad (34)$$

$$\hat{P}^*_{t,t}(i) = \hat{P}^*_{t,t-1}(i) - W(i)\nabla g_M(i)\hat{P}^*_{t,t-1}(i) \quad (35)$$

$$W(i) = \hat{P}^*_{t,t-1}(i)\nabla g_M{}^T(i)(\nabla g_M(i)\hat{P}^*_{t,t-1}(i)\nabla g_M{}^T(i) + R_t(i))^{-1} \quad (36)$$

where $\hat{s}^*_{t,t}(i)$ is the estimate of state $s_t(i)$ after update. $\hat{P}^*_{t,t}(i)$ is a 6×6 covariance matrix of $\hat{s}^*_{t,t}(i)$. $W(i)$ is Kalman gain matrix. $\nabla g_M(i)$ is the Jacobian of the non-linear observation equations of filter i evaluated at $\hat{s}^*_{t,t-1}(i)$. $R_t(i)$ is the covariance of the measurement noise. $r_t(i)$ is the innovation vector as stated in equation (37). The outputs of the i^{th} filter after the prediction phase are $\hat{s}^*_{t,t-1}(i)$ and $\hat{P}^*_{t,t-1}(i)$ while that of the smoothing phase are $\hat{s}^*_{t,t}(i)$ and $\hat{P}^*_{t,t}(i)$.

E. Likelihood Update and Final State Computation

After the Kalman filtering cycle, the likelihood of each filter $\lambda_t(i)$ is updated with regard to the innovation vector $r_t(i)$ and its corresponding residual covariance $S_t(i)$ of the filters.

$$r_t(i) = \begin{cases} \varepsilon^*_t - g^*_t(M_t) & \text{for } i = 1 \\ \left[\varepsilon_1 - g_t(M_t, \varepsilon'_1, \varepsilon_t) \quad \varepsilon'_1 - g'_t(M_t, \varepsilon_1, \varepsilon'_t)\right]^T & \text{for } i = 2 \end{cases} \quad (37)$$

$$S_t(i) = \nabla g_M(i)\hat{P}^*_{t,t-1}(i)\nabla g_M{}^T(i) + R_t(i) \quad (38)$$

$$\lambda_t(i) = \kappa_t \frac{\lambda^*_t(i) exp\left[-0.5 r_t^T(i) S_t^{-1}(i) r_t(i)\right]}{\sqrt{\|S_t(i)\|}} \quad (39)$$

$$\sum_i \lambda_t(i) = \kappa_t \quad (40)$$

κ_t is a normalization factor and $\lambda_t(i)$ is computed according to a zero-mean normal distribution function. Lastly, the usable output state vector $\hat{s}^*_{t,t}$ and covariance matrix $\hat{P}^*_{t,t}$ at the current time-step are generated with the following equations:

$$\hat{s}^*_{t,t} = \sum_i \lambda_t(i)\hat{s}^*_{t,t}(i) \quad (41)$$

$$\hat{P}^*_{t,t} = \sum_i \lambda_t(i)\left(\hat{P}^*_{t,t}(i) + \delta_t(i)\right) \quad (42)$$

$$\delta_t(i) = \left[\hat{s}^*_{t,t}(i) - \hat{s}^*_{t,t}\right]\left[\hat{s}^*_{t,t}(i) - \hat{s}^*_{t,t}\right]^T \quad (43)$$

The final state output $\hat{s}^*_{t,t}$ of the system is a linear sum of the smoothed state and covariance estimates of each filter weighted by the corresponding updated filter likelihood.

V. EXPERIMENT RESULTS

A. Experiments with synthetic data

There are few types of errors that could affect the accuracy of the camera poses recovered by the proposed algorithm. The first type is the 2-D measurement error. To make the analysis, a uniformly distributed random error v_t' was imposed on each measurement. 40 synthetic features consisting of both points and lines were randomly generated. They were centered at 0.5m away from the camera The motion parameters per frame were randomly set with their maximum change of 0.5 degrees and 0.0005 meters in rotation and translation, respectively. The focal length of a camera was 4.6mm. The pixel dimension was 5.42 x 10^{-3} mm by 5.42 x 10^3mm. Each simulation experiment for a particular measurement error value contained 50 trials and each test case involved a 150-frame-long synthetic image sequence. Fig. 5 shows how the accuracy of the recovered poses varied as a function of the measurement error. The errors in the plots are the average accumulated total rotation and translation errors measured at the 150$^{\text{th}}$ frame. The errors were relatively small.

The second type of error that we are going to investigate is the calibration error induced by inaccurate focal length. Fig. 6 shows the results. The formulae for calculating the errors are the same as the previous test. The proposed algorithm was not susceptible to erroneous focal length.

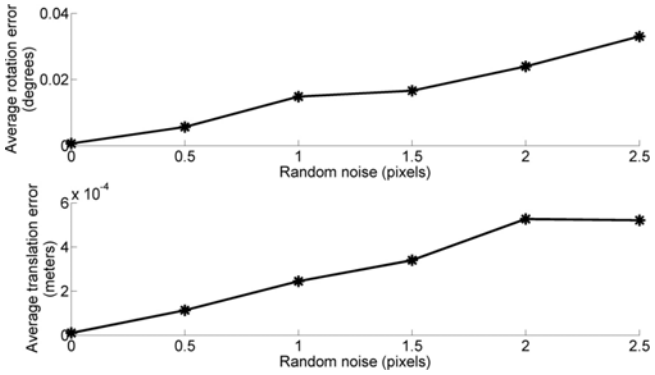

Fig. 5: The accuracy of the recovered poses varies as a function of measurement noise.

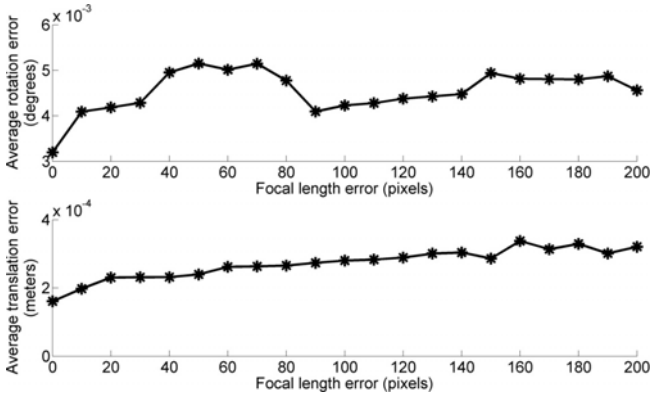

Fig. 6: The accuracy of the recovered poses varies as a function of errors in the calibrated focal length.

In the third experiment, we want to compare our method with other existing methods. An empirical comparison among the proposed hybrid method, the line-based approach by Elqursh and Elgammal [5], and the point-based algorithm by Yu et. al. [11] was made using synthetic data. Configurations of the simulation were the same as before but with an accurate focal length. Fig. 7 shows the errors of 50 random test cases. The lines with different markers (asterisk for the proposed method, triangle for the approach in [5] and square for the

method in [11]) are used to represent the experiment results of different methods. The proposed approach had errors smaller than that of Yu's point-based algorithm [11] and Elqursh's line-based method [5] for most of the time during tracking.

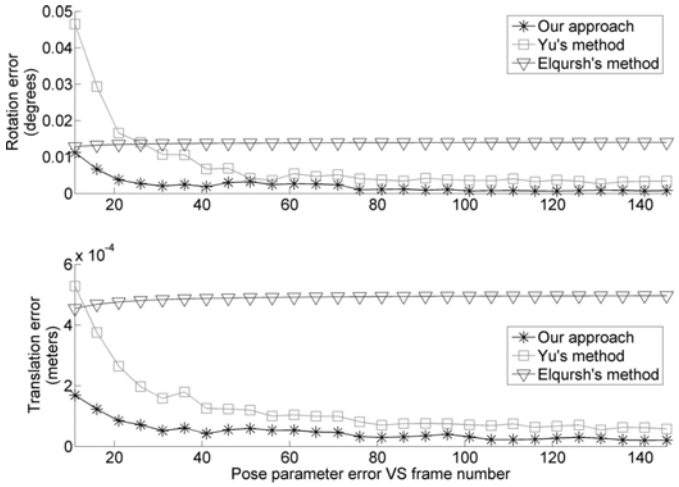

Fig. 7: The mean accumulated rotation (top) and translation (bottom) errors versus frame number of the algorithms under comparison.

TABLE I: A COMPARISON OF COMPUTATIONAL TIME AMONG THE THREE ALGORITHMS

Algorithm\ Feature number	Minimum Number	10	20	30
The proposed approach	0.0030s	0.0050s	0.0095s	0.0146s
Yu's method	0.0009s	0.0016s	0.0032s	0.0051s
Elqursh's method	0.0002s	N/A	N/A	N/A

Table 1 shows the core speed of all algorithms under comparison. The computation time required per frame is stated. The algorithms were implemented using C++ and tested using a desktop computer. Having 10 features in the scene, our method could operate at 200Hz.

B. Experiments with real images

A stereo image sequence was taken to evaluate the algorithm. We want to show if our algorithm can accurately estimate the 3-D motion of a robot. The robot moved in front of the bookshelf and images were taken with a stereo camera pair on the top. The ground-truth motion data was acquired while the robot was moving. The resolution of the images captured was 640x480 pixels.

Fig. 8 shows the comparison of the recovered motion with the ground truth. The average total errors in translation and rotation were lower than 0.003 meters and 1 degree, respectively. These errors were quite small and were mainly induced by the deviations of the calibrated values of the system parameters from the actual values.

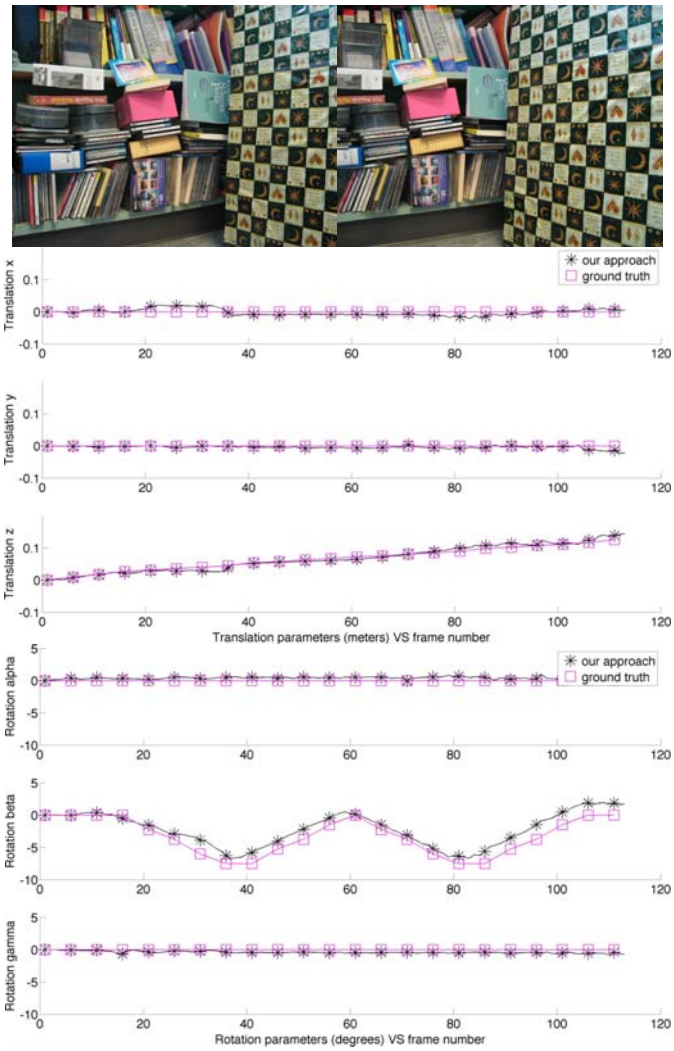

Fig. 8: A comparison of the recovered motion parameters from the real sequence with the ground truth values in the Bookshelf sequence.

VI. CONCLUSION

We have presented a recursive model-less algorithm to compute the 3-D camera motion from 2-D images in this paper. Both point features and straight lines are utilized in the computation process. At each time step, the motion is first predicted using the dynamic models in the extended Kalman filters (EKFs). It is then re-estimated with the help of the trifocal tensor point and line transfer functions in the measurement models. The final 3-D motion is computed by the likelihood probability of each EKF in the interacting probabilistic switching (IPS) framework. The trifocal constraints are incorporated into the system to eliminate the step of 3-D structure reconstruction. The core part of the proposed method takes 0.005s to process an image pair with 10 features. The system is able to operate with a minimum of 2 lines or 2 points. This requirement is so lenient that one can easily extract such a number of features in most realistic scenes. The proposed method has been applied to both synthetic and real data to demonstrate its performance. It is shown that our algorithm outperformed the latest recursive

model-less approach [11] that uses only point features, and another recent method that depends solely on line features [5].

ACKNOWLEDGMENT

This work is supported by a direct grant (Project Code: 4055045) from the Faculty of Engineering of the Chinese University of Hong Kong.

REFERENCES

[1] M.Chandraker, J.Lim and D.Kriegman, "Moving in stereo: Efficient structure and motion using lines", in *Proc. of the IEEE Intl. Conf. on Comput. Vision 2009*, pp. 1741–1748, 2009.

[2] M.Pressigout and E.Marchand, "Real-time 3d model-based tracking: Combining edge and texture information", presented at IEEE Intl. Conf. on Robotics and Automation 2006, Orlando, May 2006.

[3] A.Chiuso, P.Favaro, H.Jin and S.Soatto, "Structure from motion causally integrated over time", *IEEE Trans. on Pattern Anal. and Machine Intell.,* vol. 24, no. 4, pp. 523–535, 2002.

[4] K.Yubin and A.Kalle, "Pose estimation with unknown focal length using points, directions and lines", presented at the IEEE Intl. Conf. on Comput. Vision 2013, Sydney, December 2013.

[5] A.Elqursh and A.Elgammal, "Line-based relative pose estimation", in *Proc. of the IEEE Computer Society Conf. on Comput. Vision and Pattern Recognit. 2011,* pp. 3049–3056, 2011.

[6] P.Moulon, P.Monasse and R.Marlet, "Global Fusion of Relative Motions for Robust, Accurate and Scalable", presented at the IEEE Intl. Conf. on Comput. Vision 2013, Sydney, December 2013.

[7] B.Lucas and T.Kanade, "An iterative image registration technique with an application to stereo vision", in *Proc. of Image Understanding Workshop*, pp. 121–130, 1981.

[8] R.I.Hartley and A.Zisserman, *Multiple View Geometry in Computer Vision,* Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[9] Y.K.Yu, K.H.Wong, M.M.Y.Chang and S. H. Or, "Recursive camera-motion estimation with the trifocal tensor", *IEEE Transactions on Syst., Man, and Cybern., Part B: Cybern.,* vol. 36, no. 5, pp. 1081-1090, 2006.

[10] Y.K.Yu, K.H.Wong and M.M.Y.Chang, "Merging artificial objects with marker-less video sequences based on the interacting multiple model method", *IEEE Trans. on Multimedia,* vol. 8, no. 3, pp. 521–528, 2006.

[11] Y.K.Yu, K.H.Wong, S.H.Or and M.M.Y.Chang, "Robust 3-d motion tracking from stereo images: A model-less method", *IEEE Trans. on Instrumentation and Measurement,* vol. 57, no. 3, pp. 622–630, 2008.

[12] Y.K.Yu, K.H.Wong, and M.M.Y.Chang, "Recursive three-dimensional model reconstruction based on kalman filtering", *IEEE Transactions on Syst., Man, and Cybern., Part B: Cybern.,* vol. 35, no. 3, pp. 587-592, 2005.

[13] S.Hinterstoisser, V.Lepetit, S.Ilic, S.Holzer, G.Bradski, K.Konolige and N.Navab, "Model-based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes", in *Proc. of Asian Conf. on Comput. Vision 2012*, pp. 548–562, 2013.

[14] J.Y.Bouguet, Camera calibration toolbox for matlab, 2010.

[15] A.I.Comport, E.Marchand, M.Pressigout and F.Chaumette, "Real-time markerless tracking for augmented reality: the virtual visual servoing framework", *IEEE Trans. on Visualization and Computer Graphics*, vol.12, no. 4, pp. 615–628, 2006.

[16] T.Lemaire and S.Lacroix, "Monocular-vision based slam using line segments", in *Proc. of IEEE Intl. Conf. on Robotics and Automation 2007*, pp. 2791–2796, 2007.

[17] L-J.Qin and F.Zhu, "A new method for pose estimation from line correspondences", *Acta Automatica Sinica*, vol. 34, no. 2, pp. 130–134, 2008.

[18] S.Ramalingam, S.Bouaziz and P.Sturm, "Pose estimation using both points and lines for geo-localization", in *Proc. of IEEE Intl. Conf. on Robotics and Automation 2011*, pp.4716–4723, 2011.

[19] C.Schmid and A.Zisserman, "Automatic line matching across views", in *Proc. of IEEE Computer Society Conf. on Comput. Vision and Pattern Recognit. 1997*, pp. 666–671, 1997.

[20] Z.Wang, F.Wu and Z.Hu, "MSLD: A robust descriptor for line matching", *Pattern Recognition*, vol. 42, no. 5, pp. 941 – 953, 2009.

[21] M.E.Spetsakis and J.Aloimonos, "A unified theory of structure from motion", presented at DARPA Image Understanding Workshop, 1990.

[22] R.I.Hartley, "A linear method for reconstruction from lines and points", in *Proc. of IEEE Computer Society Conf. on Comput. Vision and Pattern Recognit. 1995*, pp. 882-887, 1995.

[23] A.I.Comport, E.Malis and P.Rives, "Real-time quadrifocal visual odometry", *The Intl. Journal of Robotics Research* vol. 29, no. 2-3, pp. 245-266, 2010

[24] G. Taylor, and L. Kleeman, "Fusion of multimodal visual cues for model-based object tracking", presented at Australasian Conf. on Robotics and Automation, 2003.

[25] A. I. Comport, E. Malis, and P. Rives, "Accurate quadrifocal tracking for robust 3d visual odometry". in *Proceeding of IEEE Intl. Conf. on Robotics and Automation 2007*, pp. 40–45, 2007.

[26] J.Weng, T.S.Huang and N.Ahuja, "Motion and structure from line correspondences; closed-form solution, uniqueness, and optimization", *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 14, no. 3, pp. 318–336, 1992.

[27] C.Rasmussen and G.D.Hager, "Joint probabilistic techniques for tracking multi-part objects", In *Proc. of IEEE Computer Society Conf. on Comput. Vision and Pattern Recognit.*, pp. 16–21, 1998.

[28] C.Tomasi, and T.Kanade, *Detection and tracking of point features.* School of Comput. Science, Carnegie Mellon Univ., 1991.

[29] F.Zhang, D.Clarke and A.Knoll, "Visual Odometry based on a Bernoulli Filter", *Intl. Journal of Control, Automation, and Systems*, vol. 13, no.3, pp. 530-538, 2015.

[30] M.Persson, T.Piccini, M.Felsberg and R.Mester, "Robust stereo visual odometry from monocular techniques", in *Proc. of IEEE Intell. Vehicles Symposium (IV) 2015*, pp. 686-691, 2015.

[31] T.Koletschka, L.Puig and K.Daniilidis, "MEVO Multi-environment stereo visual odometry", in *Proc. of IEEE/RSJ Intl. Conf. on Intell. Robots and Syst. 2014*, pp. 4981 - 4988, 2014.

[32] H. Silva, A.Bernardino and E.Silva, "Probabilistic Egomotion for Stereo Visual Odometry, *Journal of Intell. & Robotic Systems*, vol. 77, no. 2, pp. 265-280, February 2015.

[33] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. on Pattern Anal. and Machine Intell.,* vol. 32, no. 1, pp.105–119, 2010.

[34] M.Calonder, V.Lepetit, C.Strecha and P.Fua, "BRIEF: Binary Robust Independent Elementary Features", *in Proc. 11th European Conf. on Comput. Vision 2010*, pp. 778-792, 2010.

[35] C. Rother, "Linear multiview reconstruction of points, lines, planes and cameras using a reference plane", In *Proc. of 9th Intl. Conf. on Comput. Vision*, vol. 2, pp. 1210-1217, 2003.

[36] Y.K.Yu, K.H.Wong and M.M.Y.Chang, "Pose estimation for augmented reality applications using genetic algorithm", *IEEE Transactions on Syst., Man, and Cybern., Part B: Cybern.,* vol. 35, no. 6, pp. 1295-1301, 2005.