2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Image Moments for Higher-Level Feature Based Navigation

Ashwin Dani, Ghazaleh Panahandeh, Soon-Jo Chung, Seth Hutchinson

Abstract— **This paper presents a novel vision-based localization and mapping algorithm using image moments of region features. The environment is represented using regions, such as planes and/or 3D objects instead of only a dense set of feature points. The regions can be uniquely defined using a small number of parameters; e.g., a plane can be completely characterized by normal vector and distance to a local coordinate frame attached to the plane. The variation of image moments of the regions in successive images can be related to the parameters of the regions. Instead of tracking a large number of feature points, variations of image moments of regions can be computed by tracking the segmented regions or a few feature points on the objects in successive images. A map represented by regions can be characterized using a minimal set of parameters. The problem is formulated as a nonlinear filtering problem. A new discrete-time nonlinear filter based on the state-dependent coefficient (SDC) form of nonlinear functions is presented. It is shown via Monte-Carlo simulations that the new nonlinear filter is more accurate and consistent than EKF by evaluating the root-mean squared error (RMSE) and normalized estimation error squared (NEES).**

I. Introduction

In this paper, we introduce a localization and mapping algorithm that represents the scene using arbitrary shapes, such as planar features or 3D objects. By analyzing the variation of image moments of various segmented regions, 3D geometric information of the objects is estimated. The estimated geometric information of the objects is used to localize the robot. Typically, SLAM algorithms represent the information about its environment using a dense set of point features. To reduce the computational complexity of SLAM filtering, there have been efforts to extract geometric features, such as small edges (called edglets), lines, angles, etc [1]–[4]. In this paper, we propose to use a minimal set of attributes to represent a region; e.g., a planar region feature can be defined using a normal vector and distance to a local coordinate frame attached to the plane from the world coordinate frame. The algorithm in this paper represents a map using planar region features. An extension to 3D objects can be achieved with the same framework. Motivated by our recent work in [5], [6], we also propose a new optimal nonlinear filter based on a state-dependent coefficient (SDC) form of a nonlinear system. The filter minimizes the variance of the state estimation errors using convex optimization and is similar to discrete-time EKF in terms of implementation.

This project was supported by the Office of Naval Research (ONR) under Award No. N00014-11-1-0088.

Ashwin Dani, Soon-Jo Chung, and Seth Hutchinson are with the University of Illinois at Urbana-Champaign, Urbana, IL 61801. Email: {adani, sjchung, seth}@illinois.edu

G. Panahandeh is with the KTH-Royal Institute of Technology, Stockholm, Sweden. Email: ghpa@kth.se

Image moments provide generic representation of encoding the global characteristics of the objects. The concept of using image moments for the visual SLAM is novel and has not yet been presented in the literature. Typically, the outdoor or indoor scene is partially structured with higher level shapes, such as planar surfaces of arbitrary shapes or 3D shapes. This paper exploits the structure of an indoor environment or the partial structure of complex outdoor scenes, such as a riverine environment. The representation of a scene using planar objects of arbitrary shape reduces the state space for the SLAM algorithm. Another advantage of using higher-level structures is that the computational complexity in data association step is reduced.

Data association in visual SLAM requires matching numerous feature points in successive images. Large and rapid camera displacements result in erroneous feature matching that can lead to outliers. In order to reduce computationally expensive data association and feature matching between successive images, region tracking is a more robust and less resource intensive method. We take advantage of image moments that are invariant to rotation, translation, and scaling, such as Hu's moments [7] to perform data association/tracking for visual SLAM and to estimate the parameters of various region features. In certain scenarios, region tracking might be a very difficult task. In such cases, regions can still be represented using a smaller set of feature points. Once a region is segmented and tracked in the successive image frames, the variation of image moments of the tracked regions can be computed to estimate the parameters of the regions (the normal vector inversely scaled by the depth of the plane) in the camera frame. This information can be used to localize the camera in the world frame and to create a better map of the environment than a sparse map created using feature points.

The contributions of the paper are:

- Image moments are used to track the region features in the images and extract the parameters of the regions in the camera frame, which are used to represent the map. The use of region features reduces the number of landmarks; significantly reduces the dimension of the states for the navigation filter; and provides a better (possibly more meaningful) representation of the map which may be useful for a decision making stage. Another advantage is that the image moments capture the aggregate properties of the features and are more robust than the point feature-based computation.
- We present a new optimal discrete-time SDC-based nonlinear estimator, which is more accurate and improves the consistency of the nonlinear filter compared to the

extended Kalman filter (EKF). The consistency of two estimators is measured using average normalized estimation error squared (NEES) and accuracy is measured using the root-mean squared error (RMSE) computed over several Monte-Carlo runs. The estimator presented in this paper is based on the ideas in [5], [6] but is different in the sense that it is discrete time, uses different optimization criteria, and has a formulation similar to widely-used EKF.

- By using the image-moment features, we obtain a minimalistic representation in visual-SLAM framework, where the goal is to exploit the geometry of the structure and represent them using a minimum number of parameters. The shape of the regions can be arbitrary, e.g., a planar structure with arbitrary boundaries. The proposed framework is along the paradigm of environment representation using non-geometric landmarks (cf. [8]).

II. Related Work

Recently, higher-level structures have been used for SLAM [1], [2], [4], [9]. In [4], a curve-based representation of SLAM has been presented. In [1], a SLAM algorithm for hand-held camera along with planar feature identification capability using scale invariant feature transform (SIFT) features is developed. The work in [9] uses planar features instead of feature points for map building but the method depends on computation of homography between planar objects to estimate the parameters of the planar object. For homography computation, the method requires to track feature points on a planar object. In [2], planes and lines are fitted to the point features. However, it still requires matching numerous feature points in the images and also has an additional step of fitting higher-level structures in the tracked feature points. In [10], a region-based SLAM framework is presented that considers rectangular objects to create a map of the indoor environment. For outdoor environments the use of natural landmarks, such as trees, has been explored in [11]. In [12], [13], algorithms to extract tree features using laser scanners are presented with application to ground robot navigation in natural environments. In this paper, image moments are used to estimate the properties of higher-level objects in the scene. Using image moments for SLAM presents a more generalized approach in the sense that it can be used for planar objects of any complex shapes, and even for the 3D shapes, such as cylinders, spheres, etc. Image-moments have been used for visual servoing tasks [14]–[16] and are popular in computer vision for pattern recognition [7], [17].

Various approaches for SLAM using monocular vision are present in the literature [18], [19]. These approaches suffer from a drifting problem since the absolute scale of the scene is unobservable using a single camera. To avoid the drift in loop-closure, a camera-inertial measurement unit (IMU) combination has recently been used [9], [20], [21]. It has been shown that using the IMU significantly improves the loop closure properties of SLAM algorithms. In this paper, a camera-IMU combination is used to avoid the scale ambiguity problem. The algorithm in this paper reduces the SLAM state space compared to MonoSLAM [18] and its variant [19] because only a few number of characteristic features (e.g., center of gravity point) for a planar region is maintained in the SLAM state. By looking at the variation of the image moments, the map of any point inside the planar region can be generated.

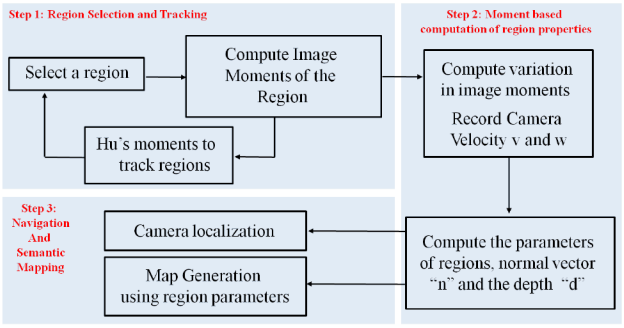


Fig. 1: Overview of the algorithm.

In [8], [22], a tutorial describing the existing solutions to the SLAM problem is presented. Three main approaches to the SLAM filtering are EKF-based filter methods [23], [24], particle-filter based methods [25], [26], and graph-based methods [27], [28]. In this paper, a nonlinear observer algorithm is developed for the SLAM problem that does not need linearization of nonlinear dynamics but relies on an 'extended linearization' parametrization or state-dependent coefficient (SDC) form of nonlinear systems. This parametrization is not unique and provides a flexible design choice to improve the accuracy and observability properties of the filtering problem. It has been shown that SDC-based filters are asymptotically optimal in the presence of nonlinearities in the system dynamics (cf., [29]–[32]).

III. Overview of Algorithm

We briefly describe the algorithm in this paper, which is divided into tracking/association (assuming regions are segmented or a small number of feature points are selected on the regions), localization and mapping tasks, as shown in Fig. 1.

Step 1: First the region features are tracked in the images to exploit the structure of the environment.

Step 2: The variation in 2D image moments of environment features in successive images is related to the linear and angular velocities of the camera and the geometric parameters of the region. For a planar region structure these parameters include a normal vector to the plane, and distance of the local coordinate frame from the world frame, w.l.o.g. we attached a local coordinate frame to a feature point on the plane. The structural parameters are estimated using a minimum variance estimator from the image observations in the camera/robot reference frame.

Step 3: The camera/robot is localized in the world coordinate frame using a new optimal nonlinear filter and the estimated structural parameters as the measurements.

Steps 2 and 3 are coupled in the sense that Step 2 uses linear and angular velocities of the camera estimated by the filter in Step 3 and Step 3 uses the output of Step 2 as measurements. Hence, this is a coupled estimation problem. The algorithm pseudo-code is given in Algorithm 1. Even though steps 2 and 3 are coupled, the mapping and localization part are decoupled where the map is created using least-square estimator and the camera/robot is localized using an optimal nonlinear filter.

IV. Image Moments and Global Characteristics of Structures

In this section, a relationship between the variation of image moments and the parameters of higher-level structures is presented. Only a case of planar objects is shown here but a similar development for spherical or cylindrical objects can also be derived [14], [15]. Let O be the planar object and $I(t)$ be the image of the object observed at current time t. Image moments of the segmented object $\bar{m}(t) = \{m_{ij} | \, i,j = 0,..,n_1\}$ and the corresponding central moment $\mu_{ij}(t)$ are defined as

$$m_{ij} = \iint_{I(t)} x^i y^j dxdy \tag{1}$$

$$\mu_{ij} = \iint_{I(t)} (x-\bar{x})^i (y-\bar{y})^j dxdy \tag{2}$$

where i, j defines the order of the moment, x and y are the image pixel coordinates, $(\bar{x}, \bar{y})$ is the centroid of the image segment. The variation of the image moments, m_{ij} and μ_{ij}, of a planar object is related analytically to the camera velocities as follows [14]

$$\dot{m}_{ij} = L_{m_{ij}}(\theta_1, \theta_2, \theta_3, m_{ij})\, \bar{v} \tag{3}$$

$$\dot{\mu}_{ij} = L_{\mu_{ij}}(\theta_1, \theta_2, \theta_3, \mu_{ij})\, \bar{v} \tag{4}$$

where $L_{m_{ij}}$ and $L_{\mu_{ij}}$ are the interaction matrices [14], $\bar{v}(t) = \begin{bmatrix} v^C & \omega^C \end{bmatrix}^T \in \mathbb{R}^6$, $v^C(t) = [v_x,\, v_y,\, v_z]$ denotes the camera linear velocity expressed in the coordinate frame C attached to the camera, $\omega^C = [\omega_x,\, \omega_y,\, \omega_z]$ is the camera angular velocity, $\theta(t) \triangleq \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T \in \mathbb{R}^3$ are the parameters θ related to the normal vector $\bar{n}$ of the planar object measured in the camera frame by [14]

$$\theta = [\theta_1\, \theta_2\, \theta_3]^T = -\frac{\bar{n}}{d} \tag{5}$$

where d is a perpendicular distance to the planar object from the origin of the camera. The Z coordinate of a point on the planar object in the camera frame C can be obtained using

$$\frac{1}{Z} = \theta_1 x + \theta_2 y + \theta_3. \tag{6}$$

The interaction matrices are given by

$$L_{m_{ij}} = \begin{bmatrix} m_{vx} & m_{vy} & m_{vz} & m_{\omega x} & m_{\omega y} & m_{\omega z} \end{bmatrix} \tag{7}$$

$$L_{\mu_{ij}} = \begin{bmatrix} \mu_{vx} & \mu_{vy} & \mu_{vz} & \mu_{\omega x} & \mu_{\omega y} & \mu_{\omega z} \end{bmatrix} \tag{8}$$

where the details of the terms in (7) and (8) are given in [14]. Observing that the right-hand side of (4) is linear in $\theta(t)$, (4) can be rewritten in the following form

$$\dot{m}_{ij} = J\left(m_{ij}, v^C\right)\theta + g\left(m_{ij}, \omega^C\right) + w_1 \tag{9}$$

where $J : \mathbb{R}^{n_1} \times \mathbb{R}^3 \to \mathbb{R}^{n_1 \times 3}$ and $g : \mathbb{R}^{n_1} \times \mathbb{R}^3 \to \mathbb{R}^{n_1}$ are given in Appendix A and w_1 is a process noise represented using a normal distribution $N\left(0, \bar{Q}\right)$. To define the planar object in the map, $\bar{n}$ and Z should be estimated that are related to θ by (5) and (6). The parameter θ can be estimated by using (9), variation of the image moments m_{ij} and camera velocities measured in C. The camera velocities are measured in the IMU coordinate frame, denoted by B. The velocities in C and B are related by $v^C = R_B^C v^B$ and $\omega^C = R_B^C \omega^B$.

A. Computation of the Parameters of the Plane

The parameters θ can be computed using a linear unbiased estimator given the measurements of m_{ij} and three or more image moments of the image segment. The estimate $\hat{\theta}(t) \in \mathbb{R}^3$ of $\theta(t)$ is given by

$$\hat{\theta} = \left(J^T \bar{Q}^{-1} J\right)^{-1} J^T \bar{Q}^{-1} \left(\dot{m}_{ij} - g\left(m_{ij}, \omega\right)\right) \tag{10}$$

The estimator (10) minimizes the cost function

$$J_c = \left(\dot{m}_{ij} - J\hat{\theta} - g\right)^T \bar{Q}^{-1} \left(\dot{m}_{ij} - J\hat{\theta} - g\right) \tag{11}$$

The covariance of the estimate, denoted by $S \in \mathbb{R}^{3\times 3}$ is given by

$$S = \left(J^T \bar{Q}^{-1} J\right)^{-1} \tag{12}$$

The estimator is simple to implement and provides an unbiased optimal estimate in the sense of the least variance. The estimated parameter $\hat{\theta}$ are used as measurements for the SLAM filter. The inverse depth $\frac{1}{Z}$ coordinate of any point on the object in the camera coordinate frame can be computed using $\hat{\theta}$ and the relationship in (6). Since the computed inverse depth is noisy due to image segmentation noise and quantization noise of image moments, it is filtered using a Kalman filter using $\frac{1}{Z}$ as a measurement and following inverse depth dynamic model

$$\frac{d}{dt}\left(\frac{1}{Z}\right) = (-\omega_2 x + \omega_1 y)\frac{1}{Z} + v_z \frac{1}{Z^2} \tag{13}$$

The 3D coordinates of a point in camera reference frame can be computed using the estimated Z, the camera calibration parameters and location of a point in the image using the relationship

$$X = \frac{xZ}{f}, \qquad Y = \frac{yZ}{f}. \tag{14}$$

where f is the focal length of the camera and $[X, Y, Z]$ are the 3D coordinates of the feature points in the camera frame.

V. Navigation Problem Formulation

In this section, a formulation of the navigation problem is presented. The landmarks are represented using planar structures, which are defined using parameters of a plane instead of range and bearing to each of the feature points belonging to the plane. The navigation filter estimates the

camera position, linear velocity, IMU biases using the measurements of positions of features in the region (e.g., the center of mass of the region) estimated in the camera body frame C using the estimator in Section IV-A. The map is estimated in the camera frame C using the estimator in Section IV-A and is transformed into the world frame W using the estimated states of the camera. The coordinate frame attached to the IMU is denoted by B.

A. State Vector

Let the state vector for SLAM be defined by $x_v = \left[r\left(t\right), v^B\left(t\right), a_b\right]^T$, and x_l is a set of N_l landmarks $x_l = \left[x_1, x_2, .., x_{N_l}\right] \in \mathbb{R}^{3N_l},\ i = 0, 1, .., N_l$, where each landmark is denoted by a vector $x_i = \left[X_i, Y_i, Z_i\right]^T$. The vehicle state x_v is composed of the robot position $r \in \mathbb{R}^3$ measured in W, the robot linear velocity $v^B \in \mathbb{R}^3$ measured in B, and the bias in the measurements of linear acceleration denoted by a_b measured in B. The landmark vector x_l consists of the points on the regions.

B. Process Model

The dynamics of a vehicle can be represented by

$$\dot{r} = R_B^W v^B, \qquad \dot{v}^B = -\left[\omega\right]_\times v^B + a - a_b + \left(R_B^W\right)^T g,$$
$$\dot{a}_b = n_{ab}, \qquad \dot{x}_l = 0 \tag{15}$$

where $g \in \mathbb{R}^3$ denotes the gravity vector expressed in W, R_B^W denotes a rotation matrix from B to W, n_{ab} denote zero-mean, uncorrelated process noise.

C. Measurement Model

The measurement model is formed using the normalized coordinates of the points on the planar regions measured in the camera frame C. Using (6), the feature point $x_i^C \in \mathbb{R}^3$ can be estimated in C and is related to the point x_i in W by $x_i^C = R_B^C\left(R_W^B\right)^T\left(x_i - r\right) + T_B^C$ where R_B^C is a rotation and T_B^C is the translation between camera-IMU coordinate frames. The components of the measurement vector $y\left(t\right) \in \mathbb{R}^{2N_l}$ are given by

$$y_i = \left[\frac{X_i^C}{Z_i^C}, \frac{Y_i^C}{Z_i^C}\right] + \nu \quad i = 1, .2, .., N_l \tag{16}$$

where ν is a zero-mean noise with covariance S. We also use the attitude data obtained using a 3-axis magnetometer to compute the rotation between W and C. Hence, the quaternion $q \in \mathbb{R}^4$ is a measured along with the linear acceleration a and angular velocity ω from the IMU package.

The SLAM formulation uses a parametric description of higher-level structures as landmarks instead of a dense cloud of feature points to describe higher-level structures. Parametric description of landmarks using higher-level structures reduces the large state space of the standard SLAM formulation and would significantly reduce the computation time of the SLAM filter. Let $n_i^C \in \mathbb{R}^3,\ i = 1, 2, .., n$ denote the normal vectors to the planar objects in the camera coordinate frame C. The vectors n_i^C and $\bar{n}_i$ are related by

$$n_i^C = R_B^C R_W^B \bar{n}_i. \tag{17}$$

Using (17) and the attitude data, normal vector of the features and the feature points on the planar regions can be estimated in the world frame W.

D. Feature Tracking

Once a new planar patch is detected in an image, it can be tracked in successive images by computing the centers of gravity of the planar objects in the new image and compare them with the centers of gravity from the previous images. We use Hu's invariant moments [7] for this task. From two successive images of a planar object, normal vector to the plane and position of the center of gravity can be computed in the camera reference from C. The newly-computed landmark is used to augment the measurement vector y. To track the segmented landmarks in the images, Hu's moments are used which are invariant to rotation, translation and scaling. Hu's moments are derived from central moments defined in (2). For certain scenes it is hard to extract region features directly. In that case, a planar region can be represented using a small set of feature points.

E. Filter Formulation

In this section, a filter for the SLAM formulation proposed in Section V is developed. The filter is based on a state dependent coefficient (SDC) form and is a modification of the filter in [5], [6].

1) Stochastic Nonlinear System Representation: Consider a dynamic system represented by a stochastic differential equation

$$\dot{x}_v = f\left(x_v, t\right) + B\left(x\right) w_2 \tag{18}$$
$$y = h\left(x_v, t\right) + \nu \tag{19}$$

where $x_v\left(t\right)$ is the filtering state, $y\left(t\right) \in \mathbb{R}^m$ is the measurement, $f\left(x_v, t\right) : \mathbb{R}^9 \times \mathbb{R} \to \mathbb{R}^9$, $h\left(x_v, t\right) : \mathbb{R}^9 \times \mathbb{R} \to \mathbb{R}^m$, $B\left(x\right) \in \mathbb{R}^{9 \times d_1}$, and w_2 and v are d_1 and m-dimensional white noise processes. The nonlinear functions $f\left(x_v, t\right)$ and $h\left(x_v, t\right)$ are given by

$$f\left(x_v, t\right) = \begin{bmatrix} R_B^W v^B \\ -\left[\omega\right]_\times v^B + a - a_b + R_W^B g \\ 0 \end{bmatrix},\ h\left(x_v, t\right) = \left[\frac{X_i^C}{Z_i^C}, \frac{Y_i^C}{Z_i^C}\right]. \tag{20}$$

The nonlinear functions (20) can also be expressed in a state dependent coefficient (SDC) form

$$f\left(x_v, t\right) = A\left(x_v, t\right) x_v = \begin{bmatrix} 0 & R_B^W & 0 \\ 0 & -\left[\omega\right]_\times & -I \\ 0 & 0 & 0 \end{bmatrix} x_v \tag{21}$$

and

$$h\left(x_v, t\right) = C\left(x_v, t\right) x_v = \begin{bmatrix} -f_1\left(x_i^C\right) R_B^C \left(R_W^B\right)^T & 0 & 0 \end{bmatrix} x_v \tag{22}$$

where the nonlinear function $f_1\left(x_i^C\right)$ takes following two forms

$$f_1\left(x_i^C\right) = \begin{bmatrix} \frac{1}{Z_i^C} & 0 & 0 \\ 0 & \frac{1}{Z_i^C} & 0 \end{bmatrix},\ f_1\left(x_i^C\right) = \begin{bmatrix} 0 & 0 & \frac{X_i^C}{\left(Z_i^C\right)^2} \\ 0 & 0 & \frac{X_i^C}{\left(Z_i^C\right)^2} \end{bmatrix} \tag{23}$$

The nonlinear functions $f\left(x_v, t\right)$ and $h\left(x_v, t\right)$ can be represented in the SDC form using $f\left(x_v, t\right) = A\left(x_v, t\right) x_v$

and $h(x_v,t) = C(x_v,t)\,x_v$ where functions $A(x_v,t) : \mathbb{R}^9 \times \mathbb{R} \to \mathbb{R}^{9\times 9}$ and $C(x_v,t) : \mathbb{R}^9 \times \mathbb{R} \to \mathbb{R}^{m\times 9}$. For a given nonlinear vector function, various forms of SDC parametrization exist such that $f(x_v,t) = A_i(x_v,t)\,x_v$ where $i = 1,..,s_1$ and $h(x_v,t) = C_j(x_v,t)\,x_v$ where $j = 1,..,s_2$.

2) Observer Equations: The estimated state is propagated using the SDC-based stochastic observer for the system in (18) as

$$\hat{x}_v^-(t+dt) = \hat{x}_v(t) + f(\hat{x}_v,t)dt \tag{24}$$

where $\hat{x}_v^-(t+dt) \in \mathbb{R}^9$ is the propagated state and the error covariance is propagated using

$$P_i^-(t+dt) = A_i(\hat{x}_v,t)\,P(t)\,A_i(\hat{x}_v,t)^T + B(\hat{x})\,Q B^T(\hat{x}) \tag{25}$$

where $A_i(\hat{x}_v,t)$, $\forall i = \{1,..,s_1\}$ are SDC parametrization, $P_i^-(t+dt) \in \mathbb{R}^{9\times 9}$ is the propagated error covariance and $Q \in \mathbb{R}^{d_1\times d_1}$ is a process noise covariance. The state estimates are corrected when the current measurement $y(t)$ is obtained using

$$\hat{x}_v(t+dt) = \hat{x}_v^-(t+dt) + K(\hat{x}_v,t+dt)\Big(y(t) - h(\hat{x}_v,t)\Big) \tag{26}$$

The observer gain $K(\hat{x}_v,t+dt) \in \mathbb{R}^{n\times m}$ is given by

$$K = \bar{P}(\eta,t+dt)\,\bar{C}^T(\hat{x}_v,t)\,R^{-1} \tag{27}$$

where $\bar{C} = \frac{1}{s_2}\sum C_j$, and the covariance matrix $\bar{P}(\eta,t+dt)$ is obtained using a convex combination

$$\bar{P}(\eta,t+dt) = \sum_{k=1}^{q} \eta_k P_k(t+dt),\ \sum_{k=1}^{q}\eta_k = 1,\ \eta_k \geq 0 \tag{28}$$

where $\eta = \{\eta_k | k = 1,...,q = s_1 \times s_2\}$ are weight parameters and $P_k(t+dt)$, $\forall k = \{1,..,q\}$ are obtained using

$$\begin{aligned} P_k(t+dt) =& P_i^-(t+dt) - P_i^-(t+dt) \\ & C_j^T(\hat{x}_v,t)\,\bar{S}^{-1} C_j(\hat{x}_v,t)\,P_i^-(t+dt) \end{aligned} \tag{29}$$

where $P_k(t+dt) \in \mathbb{R}^{n\times n}$ is the corrected error covariance corresponding to the parametrization C_j, $\bar{S} \in \mathbb{R}^{m\times m}$ is the measurement noise covariance, the index $k = \{1,..,q\}$, $i = \{1,..,s_1\}$, $j = \{1,...s_2\}$. An algorithm based on posterior Cramer-Rao lower bound (PCRLB) to update the weights η_i is presented in next section.

3) Fisher information for selection of weight parameters: Fisher information matrix FIM $\mathcal{I}$ represents the inverse of CRLB [33], which in turn lower bounds the error covariance of the filter

$$\mathcal{I}^{-1} = \text{CRLB} \leq E\left[(x_v - \hat{x}_v)(x_v - \hat{x}_v)^T\right] = \bar{P}(\eta,t+dt). \tag{30}$$

The Fisher information $\mathcal{I}$ of (30) is a theoretical bound and can only be estimated with the real data. The filter is efficient if the equality in (30) is achieved which is only possible for a linear dynamic and measurement model with additive Gaussian noise characteristics. Since the error covariance of the SDC filter $\bar{P}$ depends on the weight parameters η_k we can compute an optimal $\bar{P}$ by choosing η_k according to some objective function. We choose to maximize the determinant of the information matrix, which represents the amount of information used by the filter or minimizes the ellipsoidal uncertainty of the estimated error covariance. To compute η_i a following optimization problem is formulated

$$\begin{aligned} &\max_{\eta_k} \log\det\left(\mathcal{I}(\eta_k)\right) \\ &\text{sub to } \ \bar{P} > 0,\ \sum_{i=1}^{q}\eta_k = 1,\ \eta_k \geq 0 \end{aligned} \tag{31}$$

where the optimization objective is heuristic in the sense that the theoretical value of $\mathcal{I}$ is a property of a particular nonlinear dynamics and measurement equation and not the particular approximation of the nonlinear system. The intuition behind the optimization problem in (31) is that the information content in the representation the various SDCs (which is an approximation) is maximized or *equivalently* the uncertainty in the state estimation is minimized by optimally weighing each individual SDC parametrization. This optimization problem is referred to as a D-optimization problem and is a convex problem, which can be solved efficiently using the interior point tools [34]. The optimization problem requires a computation of $\mathcal{I}$ which can be achieved in a Riccati-like recursive form using algorithm in [33]. Alternately, an equivalent convex optimization problem can be solved

$$\begin{aligned} &\min_{\eta_k} \log\det\left(\bar{P}(\eta)\right) \\ &\text{sub to } \ \bar{P} > 0,\ \sum_{i=1}^{q}\eta_k = 1,\ \eta_k \geq 0 \end{aligned} \tag{32}$$

where the equivalence of objective functions $-\log\det(\mathcal{I})$ and (30) is used [34]. An alternate convex objective function $\text{trace}(\bar{P}(\eta))$ can be used in (32) to minimize the variance of the state estimation errors. The convex optimization problems can be easily solved by fast interior point methods.

TABLE I: Comparison of the Average Errors

	RMSE	Average NEES Mean
SDC-based Estimator	14.7956	8.8423
EKF	15.7246	14.5679

VI. Simulations and Experiments

A. Numerical Simulations

Numerical simulations are conducted to test the performance of the moment-based SLAM framework presented in this paper. Selection of image moments to compute the depth of features in the image plane is based on condition number of the J matrix in Section IV-A. For the simulation we use $s = (m_{00}, x_g, y_g, \mu_{11}, \mu_{02}, \mu_{20})$ as set of features. The condition number of J gives a sensitivity criteria for selection of image moment features. It is observed through simulations that the above combination of moment features gives us the least noisy estimation result. The localization and mapping results for 150 seconds are shown in Fig. 2.

Algorithm 1 MomentSLAMAlgorithm

Data: linear acceleration a, angular rates ω, attitude q, images I_t
Result: state $x_v(t)$, map $x_l(t)$, map points inside regions
while *imu data* **do**
  Propagate the states using (21)
  Propagate P using (25)
  if *measurement* **then**
    Segment image
    Match regions using Hu's moments
    Compute moments m_{00}, x_g, y_g, μ_{00}, μ_{11}, μ_{02}, μ_{20}
    Estimate $\hat{\theta}$ using (10)
    Estimate $\frac{1}{Z_i^C}$ using (6) and filter it using KF using (13) and measurement $\frac{1}{Z_i^C}$
    Estimate x_i^C using (6) and (14)
    Correct the state using (26)-(29)
  end
  if *dense map of region is required* **then**
    Estimate x_i^C inside the region in W
    using estimated R and corresponding image pixels
  end
end

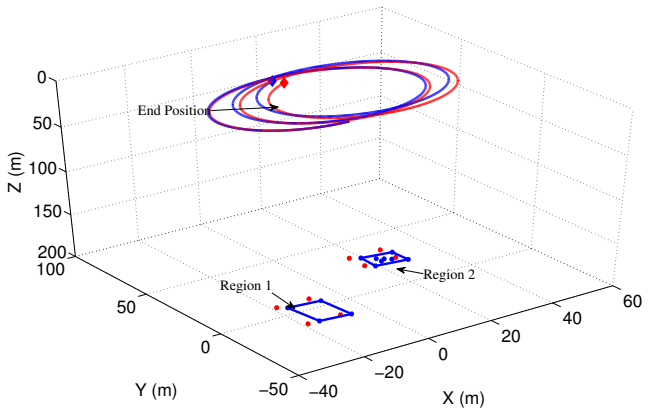
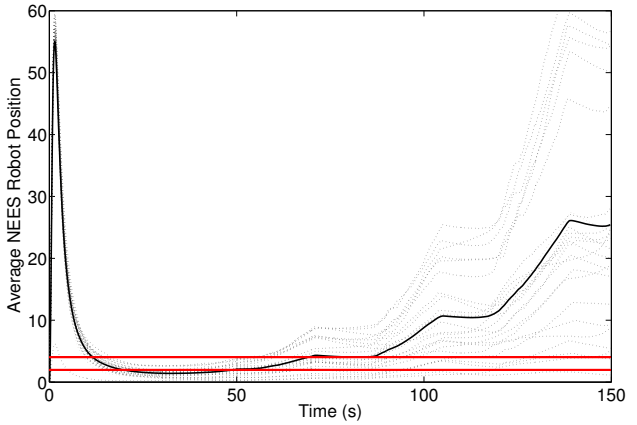
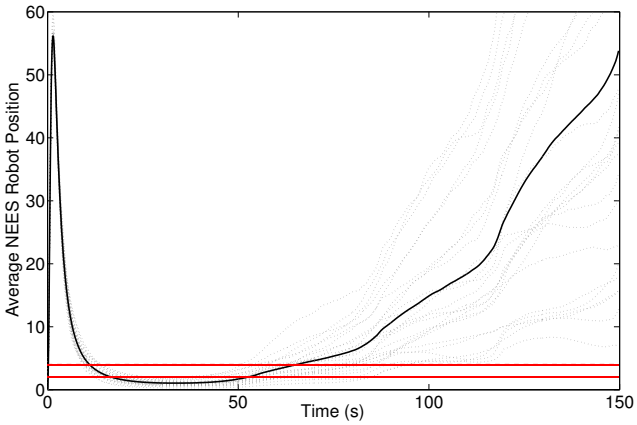


Fig. 2: Localization and mapping of planes using information extracted by variation of image moments. Blue line shows true robot trajectory, red dashed line shows estimated robot trajectory. Red stars show the true landmark locations, blue stars show estimated landmark locations. Any point inside the region can be estimated using the parameters of the plane, as depicted by blue stars.

To compare the consistency of the proposed estimator and EKF, average NEES is computed over 25 Monte-Carlo runs of each estimator ([35], [36]). The plots of individual NEES and their average value for the proposed estimator and EKF are shown in Fig. 3. The bounds of the double-sided 95% probability concentration region for 3-dimensional robot pose state for 25 Monte-Carlo runs are computed using $\chi^2_{(3\times25)}$ distribution. The upper and lower bounds of 3.9797 and 2.0202 are shown using two solid red lines in Fig. 3. To compare two estimators for the accuracy of estimation, average room-mean squared error of the robot pose for 25



(a) SDC-based estimator



(b) EKF

Fig. 3: Comparison of average NEES for 25 Monte-Carlo runs. Solid black line shows the averaged NEES value. Dotted gray lines show NEES for individual runs. Two red solid lines show the 95% consistency region.

Monte-Carlo runs is reported in Table I along with the mean value of the average NEES.

B. Experimental Setup

The experimental platform for the experiments result in this paper consists of a camera-IMU combination. We plan to conduct experiments on a quadcopter equipped with rigidly mounted camera and IMU in outdoor environments. We use the IMU embedded in the autopilot unit called ArduPilot - an open source unmanned aerial vehicle (UAV) platform. The IMU provides measurements of the linear acceleration and the angular velocity at 200Hz and the camera images were stored at 15fps. The IMU and camera data is time-synchronized. To estimate the relative IMU camera rotation the IMU-camera calibration toolbox [37] is used; assuming the relative translation can be neglected. The camera intrinsic parameters were estimated offline using the Matlab toolbox and are assumed constant. Images are captured with resolution 640×480 pixels. The accelerometer and gyro biases were initialized by keeping the platform static for about 60 sec and averaging out the measured linear acceleration and angular velocity. Accordingly, their corresponding variances

Fig. 4: Experimental platform.

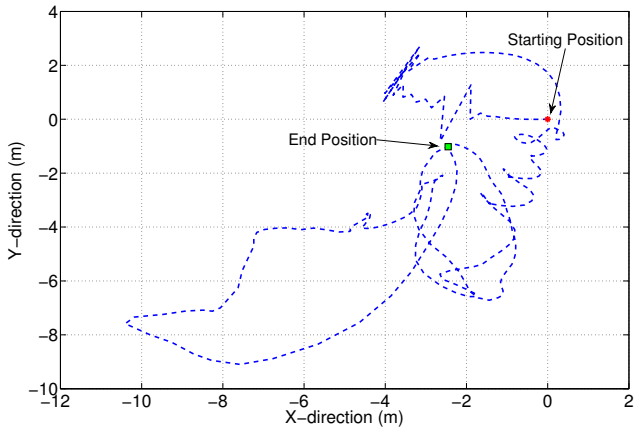


Fig. 6: 2D X-Y plot of the estimated position of the camera motion.

were used to initialize state covariance matrix. The initial position of the world coordinate frame W is assumed to be same as the starting position of the IMU coordinate frame.

C. Experimental Results

This section describes preliminary experimental results of the algorithm on the indoor office scenario. Our current efforts are to conduct experiments with data collected in different scenarios, e.g., outdoor semi-structured environments and testing the algorithm for a longer period of time to reconstruct a large map.

In Fig. 5 a region tracking sequence is shown by tracking four corner features of the region based on a LK feature point tracker. The regions are associated in the successive images by computing the invariant image moment values and using a small threshold (set to 0.1) to compare the Hu's moment as described in Section V-D. The algorithm described in Algorithm 1 is implemented in Matlab. To estimate the image moments we choose $s = (m_{00}, x_g, y_g, \mu_{11}, \mu_{02}, \mu_{20})$ as the set of moments, which corresponds to the area of the region, and X and Y location of the center of gravity of the region, and three central moments respectively. This particular set of moments is less susceptible to noise. The image moments in consecutive images are stored and the derivative of each moment $\dot{m}_{ij}$ is computed using a fourth order finite difference method. The corresponding J and g vectors for these moments are given in Appendix I. The estimator in (10) is used to generate the measurement for the filter in Section V-E. The $X - Y$ plot (top-down view) of the camera estimated position is shown in Fig. 6.

VII. Conclusion

In this paper, we introduced a new vision-based SLAM algorithm which incorporates higher-level structures, such as planar surfaces in visual navigation by using the variation of the image moments of the planar regions in 2D images. The map is represented by using regions instead of a large number of feature points. We have derived a camera-IMU SLAM formulation which represents the scene using a minimal set of parameters. We have conducted Monte-Carlo runs of the simulated environment and compared the performance of the proposed estimator to EKF. The RMSE and NEES comparison shows that the proposed estimator outperforms EKF in terms of accuracy and consistency. We have reported preliminary experimental results conducted using an indoor office scene to reconstruct camera trajectories. Our current efforts focus on testing the proposed algorithm in various outdoor environments and using discrete moments instead of continuous moments using discrete point features.

References

[1] R. Castle, D. Gawley, G. Klein, and D. Murray, "Towards simultaneous recognition, localization and mapping for hand-held and wearable cameras," in *IEEE Int. Conf. on Robotics and Autom.*, 2007, pp. 4102 – 4107.

[2] A. Gee, D. Chekhloc, A. Calway, and W. Mayol-Cuevas, "Discovering higher level structures in visual SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 980 – 990, 2008.

[3] G. Klein and D. Murray, "Improving the agility of keyframe-based SLAM," in *European Conference on Computer Vision (ECCV'08)*, October 2008, pp. 802–815.

[4] D. Rao, S.-J. Chung, and S. Hutchinson, "CurveSLAM: An approach for vision-based navigation without point features," in *IEEE/RSJ Int. Conf. Intell. Robots and Sys.*, 2012, pp. 4198–4204.

[5] A. P. Dani, S.-J. Chung, and S. Hutchinson, "Observer design for stochastic nonlinear systems using contraction analysis," in *Proc. IEEE Conf. Decision Control*, Maui, Hawaii, 2012, pp. 6028–6035.

[6] ——, "Observer design for stochastic nonlinear systems via contraction-based incremental stability," *IEEE Trans Autom. Control*, 2013 conditionally accepted.

[7] M. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inform. Theory*, vol. 8, pp. 179 – 187, 1962.

[8] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping: part II," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108 – 117, 2006.

[9] F. Servant, P. Houlier, and E. Marchand, "Improving monocular plane-based SLAM with inertial measures," in *IEEE/RSJ Conf. on Intelligent Robots and Systems*, Oct 2010, pp. 3810 – 3815.

[10] J. Oberlander, K. Uhl, J. Marius Zollner, and R. Dillmann, "A region-based SLAM algorithm capturing metric, topological, and semantic properties," in *IEEE Int. Conf. on Robot Autom.*, 2008, pp. 1886–1891.

[11] P. Forsman and A. Halme, "3-d mapping of natural environments with trees by means of mobile perception," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 482–490, 2005.

[12] M. Song, F. Sun, and K. Iagnemma, "Natural feature based localization in forested environments," in *IEEE Intel. Robot and Systems*, 2012, pp. 3384–3390.

[13] ——, "Natural landmark extraction in cluttered forested environments," in *IEEE Int. Conf. Robot Autom.*, 2012, pp. 4836–4843.

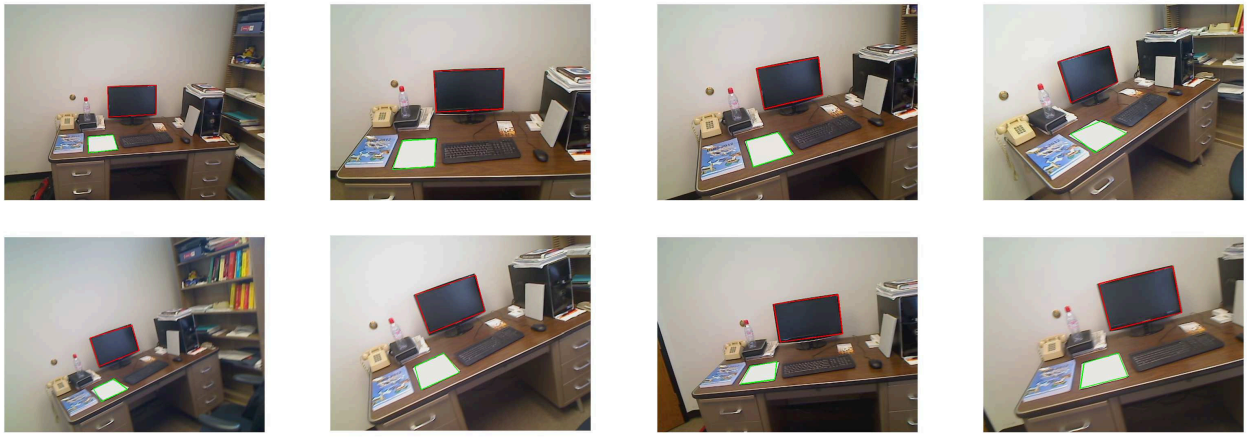


Fig. 5: Tracking of regions in successive images based on feature point tracking.

[14] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Trans on Robotics*, vol. 20, no. 4, pp. 713–723, Aug 2004.

[15] O. Tahri and F. Chaumette, "Point-based and region-based image moments for visual servoing of planar objects," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1116–1127, 2005.

[16] A. De Luca, G. Oriolo, and P. B. Giordano, "Feature depth observation for image-based visual servoing: Theory and experiments," *Int J Robot Res*, vol. 27, no. 10, pp. 1093–1116, 2008.

[17] R. Prokop and A. Reeves, "A survey of moment-based techniques for unoccluded object representation and recognition," in *Proc. Computer Vision, Graphics, Image Processing Conf.*, vol. 54, 1992, pp. 438 – 460.

[18] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052 – 1067, 2007.

[19] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 932–945, 2008.

[20] P. Pinies, T. Lupton, S. Sukkarieh, and J. Tardos, "Intertial aiding of inverse depth SLAM using a monocular camera," in *IEEE Int. Conf. on Robot. and Autom.*, 2007, pp. 2797 – 2802.

[21] T. Lupton and S. Sukkarieh, "Removing scale biases and ambiguity from 6DoF monocular SLAM using inertial measurements," in *IEEE International Conference on Robotics and Automation*, 2008, pp. 3698 – 3703.

[22] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99 – 110, 2006.

[23] S. Williams, P. Newman, M. Dissanayake, and H. Durrant-Whyte, "Autonomous underwater simultaneuous localization and map building," in *IEEE Int. Conf. on Robot. and Autom.*, 2000, pp. 1793 – 1798.

[24] J. Kim and S. Sukkarieh, "Airborne simultaneuous localization and map building," in *IEEE International Conf. on Robotics and Autom.*, 2003, pp. 406–411.

[25] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fast-SLAM: a factored solution to the simultaneuous localization and mapping problem," in *Proc. AAAI Nat. Conf. Artif. Intell.*, 2002, pp. 593 – 598.

[26] ——, "Fast-SLAM 2.0: an improved particle filtering algorithm for simultaneuous localization and mapping that provably converges," in *Proc. Int. Joint Conf. Artif. Intell.*, 2003, pp. 1151 – 1156.

[27] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *Int. J. Robot. Research*, vol. 25, no. 5, pp. 403 – 429, 2006.

[28] R. Kümmerle, B. Steder, C. Dornhege, A. Kleiner, G. Grisetti, and W. Burgard, "Large scale graph-based SLAM using aerial images as prior information," *Autonomous Robots*, vol. 30, no. 1, pp. 25 – 39, 2009.

[29] D. Haessig and B. Friedland, "State dependent differential Riccati equation for nonlinear estimation and control," in *15th IFAC World Congress*, Spain, 2002.

[30] H. Beikzadeh and H. Taghirad, "Robust $\mathcal{H}_\infty$ filtering for nonlinear uncertain systems using state dependent Riccati equation technique," in *Conf. on Decision and Control*, Dec 2009, pp. 4438–4445.

[31] H. T. Banks, B. M. Lewis, and H. T. Tran, "Nonlinear feedback controllers and compensators: a state-dependent Riccati equation approach," *Comput. Optim. and Appl.*, vol. 37, no. 2, pp. 177–218, 2007.

[32] T. Cimen, "State dependent Riccati equation control: a survey," in *17th IFAC World Congress*, 2008, pp. 3761–3775.

[33] P. Tichavsky, C. H. Muravchik, and A. Nehorai, "Posterior Cramer-Rao bounds for discree-time nonlinear filtering," *IEEE Trans Signal Proc.*, vol. 46, no. 5, pp. 1386–1396, 1998.

[34] J. G. VanAntwerp and R. D. Braatz, "A tutorial on linear and bilinear matrix inequalities," *J. Process Control*, vol. 10, pp. 363–385, 2000.

[35] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley and Sons, 2001.

[36] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the ekf-slam algorithm," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006, pp. 3562–3568.

[37] J. Lobo and J. Dias, "Relative pose calibration between visual and inertial sensors," *Int J Robot Res*, vol. 26, no. 6, pp. 561–575, 2007.

APPENDIX I
DERIVATION OF MATRICES IN SECTION IV

The image moment dynamics in (3) can be written in the form (9). The matrices J and g in (9) are derived for a planar object as $J = [J_{00},\ J_{x_g},\ J_{y_g}]^T$, where the elements of $J\left(m_{ij}, v^C\right)$ and $g\left(m_{ij}, \omega\right)$ in (9) are given by:

$$J_{00} = \begin{bmatrix} -av_x + 3ax_g v_z \\ -av_y + 3ay_g v_z \\ 2av_z \end{bmatrix} \quad J_{x_g} = \begin{bmatrix} -x_g v_x + \left(x_g^2 + 4n_{20}\right) v_z \\ -y_g v_x + \left(x_g y_g + 4n_{11} v\right) v_z \\ -v_x + x_g v_z \end{bmatrix} \tag{33}$$

$$J_{y_g} = \begin{bmatrix} -x_g v_y + \left(x_g y_g + 4n_{11}\right) v_z \\ -y_g v_y + \left(y_g^2 + 4n_{02}\right) v_z \\ -v_y + y_g v_z \end{bmatrix} \tag{34}$$

where $n_{ij} = \mu_{ij}/m_{00}$, and μ_{ij} is a centered moment, and

$$\begin{aligned} g_{00} &= 3ay_g\omega_x - 3ax_g\omega_y \\ g_{x_g} &= \left(x_g y_g + 4n_{11}\right)\omega_1 - \left(1 + x_g^2 + 4n_{20}\right)\omega_y + y_g\omega_z \\ g_{y_g} &= \left(1 + y_g^2 + 4n_{02}\right)\omega_x - \left(x_g y_g + 4n_{11}\right)\omega_y - x_g\omega_z \end{aligned} \tag{35}$$