Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 10, No. 4, 2022, 493-503.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

A UKF-based Approach for Indoor Camera Trajectory Estimation

Seyyed Ali Hoseini[1*] and Peyman Kabiri[2]

1. Faculty of Electrical and Computer Engineering, University of Birjand, Birjand, Iran.
2. School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.

Article Info

Article History:
Received 05 January 2021
Revised 07 April 2022
Accepted 16 June 2022

DOI:10.22044/jadm.2022.11550.2315

Keywords:
Feature Extraction, Unscented kalman Filter, Robot Vision, Simultaneous Localization and Mapping, Camera Tracking.

**Corresponding author: sa.hoseini@birjand.ac.ir (S. A. Hoseini).*

Abstract

When a camera moves in an unfamiliar environment, for many computer vision and robotic applications, it is desirable to estimate the camera position and orientation. Camera tracking is perhaps the most challenging part of the Visual Simultaneous Localization and Mapping (Visual SLAM) and Augmented Reality problems. This paper proposes a feature-based approach for tracking a hand-held camera that moves within an indoor place with a maximum depth of around 4-5 m. In the first few frames, the camera observes a chessboard as a marker to bootstrap the system and construct the initial map. Thereafter, upon the arrival of each new frame, the algorithm pursues the camera tracking procedure. This procedure is carried out in a framework that operates using only the extracted visible natural feature points and the initial map. The constructed initial map is extended as the camera explores new areas. In addition, the proposed system employs a hierarchical method on the basis of the Lucas-Kanade registration technique to track the FAST features. For each incoming frame, the 6-DOF camera pose parameters are estimated using an Unscented Kalman Filter (UKF). The proposed algorithm is tested on real-world videos, and the performance of UKF is compared against the other camera tracking methods. Two evaluation criteria (i.e. relative pose error and absolute trajectory error) are used in order to assess the performance of the proposed algorithm. Accordingly, the reported experimental results show the accuracy and effectiveness of the presented approach. The conducted experiments also indicate that the type of extracted feature points does not have a significant effect on the precision of the proposed approach.

1. Introduction

Camera tracking is the problem of estimating camera pose parameters from a sequence of video frames. Some known problems in computer vision such as 3D reconstruction, image registration, and augmented reality have a close relationship with camera pose estimation. Meanwhile, in robotics, this problem is known as visual Simultaneous Localization And Mapping (visual SLAM). The purpose of visual SLAM is to estimate the camera's moving path and making map of the observed scene simultaneously. The constructed map can be represented in dense or sparse manners. The Visual SLAM techniques continuously extend the constructed map of the observed environment, and then locate the camera position by means of this map and its projection on image plane. However, if the camera is equipped with a RGBD sensor or range scanner, depth calculation for the detected natural feature points won't be necessary. By contrast, a monocular camera can only provide the 2D measurements of a 3D environment. On the other hand, in the visual SLAM solutions, it is required to obtain the depth of each new extracted feature point since no information about the depth of the observed scene is given using a monocular

camera. In other words, due to the noisy feature point measurements and inaccurate camera poses, depth calculation for newly detected natural feature points is prone to error. As a consequence, the process of camera localization and map extension is more complicated in the context of monocular camera tracking.

Augmented Reality (AR) is an active field of research in computer vision that should be equipped with camera tracking capability. The purpose of AR is to neatly overlay the computer-generated models on video frames. Accurate estimation of camera pose with respect to the world coordinate system is an important problem here. The primary AR systems often use fiducial markers or Computer-Aided Design (CAD) models to locate camera in a 3D scene. In other words, providing 3D-2D correspondences between 3D scene and its projection on image plane, the AR system can calculate the camera pose with respect to a reference coordinate system often known as the world coordinate system. However, for an automatic camera tracking, it is desirable to use natural landmarks. In the following the main contributions of this work are listed:

- Feature tracking routine is performed in a coarse to fine scheme, which is accurate and robust in the presence of quick camera movement.
- Initial metric map construction, which employs a marker with salient feature points whose 3D positions are known.
- The propagation of depth information, which enables the proposed system to determine the camera position as the camera explores new places.

The structure of this paper is organized in the following way. The Second section discusses the related works. The Third section explains the proposed approach in details. The experimental results are presented in the Fourth section. Conclusion and future works are included in the Fifth section.

2. Related Works

Depth estimation of newly extracted feature points is a challenging task once the camera explores unknown environment, and it is demanding to estimate the camera position with respect to its surrounding. Moreover, successive iteration of camera pose estimation and depth computation of newly detected feature points need to be performed continuously as the camera observes new regions. This repetition gradually increases cumulative error, which introduces drift in the camera position. In fact, utilizing only 2D measurements of feature points in the absence of a robust 3D map, it is only feasible to estimate 3D position of new feature points up to a scale factor.

In general, the researchers adopt two strategies for addressing the scale ambiguity in the camera tracking problem. In the first strategy, some markers with known structures are placed in front of the camera. Detecting these markers in captured images, the camera pose parameters are easily computed with a high precision. Ababsa and Mallem [1] have employed this method to handle the scale ambiguity problem. Additionally, using markers in the scene enables the system to control accumulation of the estimation error. Using the reference calibrated images is the second strategy used for camera tracking in unknown environments to deal with scale ambiguity [2], [3]. Calibrated images contain easily detectable objects with known position in the world coordinate system. Once the reference images are available, the camera tracking problem reduces to association of observed features in each new frame and the reference images.

Detection of loop closures in camera tracking is another approach, which can control the cumulative error of the estimated camera pose parameters [4]-[6]. However, it is only feasible when the camera explores those parts of the scene that are observed before. Whenever the camera explores new regions, we require extracting new feature points to retrieve the camera pose parameters. In this work, the problem of loop closure detection is not addressed. Instead, the reported work tries to propagate depth information of the initially constructed map to newly detected features points the camera observes within new sceneries.

Camera trajectory estimation for video sequences is extensively studied. Structure from Motion (SfM) and filtering methods are two major strategies presented to tackle this problem. Early studies based on the SfM strategy were mainly employing epipolar geometry principles for estimation of the camera pose parameters [7]. Additionally, the initial algorithms proposed to solve the camera tracking problem were mostly for a short video sequences or small set of images. However, later works were addressing longer image sequences [8], [9]. Parallel Tracking and Mapping (PTAM) [10] is a forerunner work developed for small scale workspaces that has a real-time performance. In PTAM, map construction and camera tracking routines are run in two separated threads. This improves the execution speed of the algorithm. S-PTAM [11] is

another feature-based method for camera tracking and mapping, which uses stereo-images. Exploiting stereo-images allows robust initialization of extracted feature points.

ORB-SLAM [12] and its successor ORB-SLAM2 [13] as well as LSD-SLAM [14] are prominent works recently developed in the visual SLAM community. ORB-SLAM and ORB-SLAM2 are feature-based methods, and LSD-SLAM is a direct (feature-less) approach that estimates camera trajectory by using directly pixel intensities. In addition, all of them benefit from strong loop closure detection methods, which improve precision of the algorithm substantially.

It is worth noting that camera tracking is the core component of the visual SLAM and visual odometry techniques. However, visual SLAM benefits from loop closure detection but visual odometry does not. Taking advantage of loop closure detection could significantly improve the accuracy of camera tracking when the camera explores previously visited regions. Nevertheless, the proposed approach has not utilized loop closure detection. Instead, our algorithm tries to do its best to improve the accuracy of camera tracking by concentration on accurate feature tracking and configuring a well-tuned UKF framework.

The advent of RGBD cameras encourages some researchers to take advantages of using RGBD images for camera tracking. RGBD-SLAM [16] and *k*-SLAM [17] are two visual SLAM approach developed for the RGBD sequences. It is worth noting that since the RGBD images contain depth values of image pixels, it is not required to triangulate newly detected features. This issue to a large extent improves performance of the algorithm.

Contrary to the SfM approach, in the filtering methods, the problem casts in the form of a dynamic system. The internal state of this dynamic system consists of the camera pose parameters. Similar to any dynamic system, the state transition of this dynamic system is a non-linear relation regarding the physical nature of rigid body motion occurring in 3D space. Furthermore, the internal state of the system has a complicated non-linear relation with measurements from the scene. Often, the Kalman filter is used to estimate the internal state of dynamic system given observations. However, these non-linearities in the camera tracking problem require employment of the Extended Kalman Filter (EKF) for estimation of the camera pose parameters [18]. In this context, MonoSLAM [19] is a leading work in which, map of the environment is represented using a probabilistic approach. Additionally, to estimate the position and orientation of the camera, a full covariance EKF is employed. Furthermore, a top-down approach is introduced to provide feature matchings along the consecutive frames. Although the reported method is categorized as a filtering approach, contrary to MonoSLAM, a deterministic map was employed. In other words, in the proposed method, the state of the filter only includes the camera pose parameters.

Some feature-based camera tracking algorithms select some frames as keyframes. Mostly, the goal is to minimize the camera pose accumulative error in keyframes. Accordingly, an optimization step is performed on selected keyframes. Two popular methods extensively used for this purpose are Bundle Adjustment (BA) [20] and pose map optimization [21]. BA is a non-linear optimization technique, which minimizes the reprojection error between the location of tracked feature points and the projection of their associated 3D points on image plane. Since these optimization routines are computationally expensive, to achieve a real-time performane, it is essential to run them in a separate thread as implemented in PTAM and ORB-SLAM.

3. Proposed Method

In Figure 1, an outline of the proposed approach is illustrated. Upon capturing every new frame, the camera tracking procedure is carried out in two phases. Firstly, the corresponding feature points in the previous frame should be located in the current frames. In the second phase, the camera position and orientation are computed using the obtained correspondences in an UKF framework.

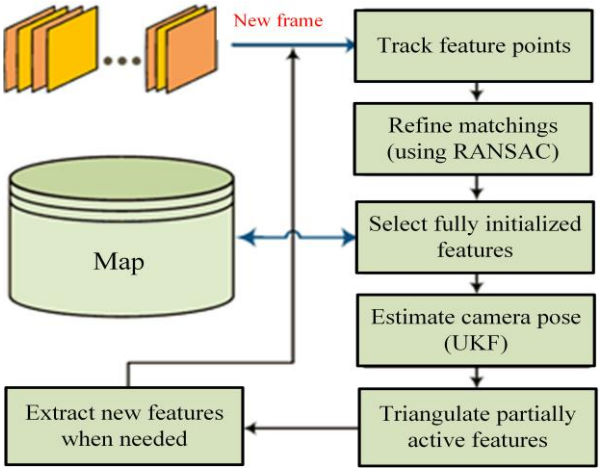

Figure 1. Overview of proposed approach.

UKF is a derivative-free approach that approximates the state distribution using special

samples drawn from distribution called sigma points. Like EKF, UKF consists of two phases, i.e. state predication and state update, except they are preceded by extra routine of sigma point selection. During the UKF steps, these sigma points are propagated through predication and observation models. Calculating weighted average of the propagated sigma points, the new state of the filter is generated. At the same time, covariance matrix of the filter state that implies uncertainty of estimated state is also constructed. It is worth noting that the number of sigma points is 2L + 1, where *L* is the dimension of filter's state.

3.1 Preliminaries

The relation between the world coordinate system and the camera coordinate systems is represented using a rigid body transformation.

$$\tilde{z}_C = T_{CW}\tilde{z}_W, \quad T_{CW} = \begin{bmatrix} R_{CW} & t_{CW} \\ 0_{1\times 3} & 0 \end{bmatrix} \in SE(3) \tag{1}$$

where $\tilde{z}_C = [z_C\ 1]^T, \tilde{z}_W = [z_W\ 1]^T$ are homogeneous coordinates of an arbitrary point in the world and camera coordinate systems, respectively. $R_{cw} \in SO(3)$ is a rotation matrix, and $t_{cw} \in R^3$ is a translation vector representing the origin of world in the camera frame. In this paper, the pose of the camera for frame *k* is denoted by T_{cw}^k. According to the pinhole camera model, projection of any arbitrary 3D point z_w in the world coordinate frame on image plane in term of pixel is calculated using (2).

$$\tilde{y} = \pi(z_W) = \frac{1}{d(z_W)} K(R_{CW}z_W + t_{CW}) \tag{1}$$

where $\tilde{y}$ is the homogeneous representation of y, $d(z_w)$ is the depth of 3D point z_w in the camera coordinate frame, and *K* is the camera calibration matrix. Furthermore, let $Z_k = \{z_1, z_2, \ldots, z_n\}$ be a set of 3D points in the scene that are already initialized in the constructed map and are successfully tracked in frame *k*. The projection of $z_i \in Z_k$ on frame k is denoted by u_i that is obtained through the feature tracking routine. Accordingly, $U_k = [u_1, u_2, \cdots, u_n]^T \in R^{2n}$ indicates the associated vector for the observed feature points in frame k.

Given a group of 3D points in the world coordinate frame and their projection on image frame, the parameters of world to camera transformation are produced by minimizing the sum of squared error of (3).

$$E(R,T) = \sum_{i=1}^{n} (\pi(z_i) - u_i)^2 \tag{3}$$

where u_i is the pixel coordinate corresponding to the z_i point in image plane obtained through the feature tracking procedure. R is the rotation matrix (often represented in the quaternion format as a unit vector belongs to R^4), and $T \in R^3$ is the translation vector that relates the world and camera coordinate systems as shown in Figure 2. In the computer vision literature, this problem is referred to by Perspective-n-Point (PnP). The non-linear optimization techniques such as Gauss-Newton and Levenberg-Marquardt are often used to minimize the error function defined in (3).

3.2. UKF details

In this work, a filtering method is employed for estimation of the posteriori density of the camera trajectory parameters. To this end, UKF is adopted to estimate the 6-D pose parameters of camera in each frame. State of the filter is a 14 × 1 random vector, as defined in (4).

$$x_k = [t_k\ q_k\ v_k\ \omega_k]^T \tag{4}$$

where t_k is the camera center relative to the world coordinate system, and q_k is the quaternion representation of rotation matrix relative to the world coordinate system at time step *k*. v_k and ω_k are the linear and angular velocities of the camera, respectively. In each step, the filter is initialized with the estimated state of the previous step. Here, it is assumed that $\bar{x}_{k-1}$ and P_{k-1} are the mean and covariance of the filter's state estimated in time step *k-1*, from which, a collection of *2L+1* sigma points (along with their weights) are constructed using (5).

$$\begin{aligned}
&x_{k-1}^0 = \bar{x}_{k-1} \\
&x_{k-1}^i = \bar{x}_{k-1} + (\sqrt{(L+\lambda)P_{k-1}})_i \quad i = 1\cdots L \\
&x_{k-1}^i = \bar{x}_{k-1} - (\sqrt{(L+\lambda)P_{k-1}})_{i-L} \quad i = L+1\cdots 2L \\
&W_m^0 = \lambda/(L+\lambda) \\
&W_c^0 = \lambda/(L+\lambda) + (1-\alpha^2+\beta) \\
&W_m^i = W_c^i = \frac{1}{2(L+\lambda)} \quad i = 1,\cdots,2L
\end{aligned} \tag{5}$$

where $\lambda = \alpha^2(L+\eta) - L$ is a scaling factor, and the three parameters α, β, η are used to tune the UKF filter [22]. The expression $(\sqrt{(L+\lambda)P_{k-1}})_i$ is the i-th row of the matrix square root.

In the developed UKF, a constant velocity model

is used for state transition between the time steps, as given in (6).

$$x_{k|k-1} = f(x_{k-1}) = \begin{pmatrix} t_{k-1} + (v_{k-1} + V_c) \\ q_{k-1} \otimes (\omega_{k-1} + \Omega_c) \\ v_{k-1} + V_c \\ \omega_{k-1} + \Omega_c \end{pmatrix} \quad (6)$$

where V_c and Ω_c are the Gaussian white noises that indicate the uncertainties for the linear and angular velocities of the camera, respectively. $\otimes$ denotes the quaternion product operation.

In the prediction phase of UKF, these sigma points are propagated through state transition.

$$x^i_{k|k-1} = f(x^i_{k-1}) \quad i = 0, \cdots, 2L \quad (7)$$

Later, the mean and covariance of the filter's state is calculated using a weighted average of propagated sigma points.

$$\begin{aligned} \bar{x}_{k|k-1} &= \sum_{i=0}^{2L} W^i_m x^i_{k|k-1} \\ P_{k|k-1} &= \sum_{i=0}^{2L} W^i_c [x^i_{k|k-1} - \bar{x}_{k|k-1}][x^i_{k|k-1} - \bar{x}_{k|k-1}]^T + Q_k \end{aligned} \quad (8)$$

where Q_k is the covariance matrix of the additive process noise. In the state update phase, new sigma points are reconstructed using the predicted state.

$$\begin{aligned} x^0_{k|k-1} &= \bar{x}_{k|k-1} \\ x^i_{k|k-1} &= \bar{x}_{k|k-1} + (\sqrt{(L+\lambda)P_{k|k-1}})_i \quad i = 1, \cdots, L \\ x^i_{k|k-1} &= \bar{x}_{k|k-1} - (\sqrt{(L+\lambda)P_{k|k-1}})_{i-L} \quad i = L+1, \cdots, 2L \end{aligned} \quad (9)$$

If the camera pose parameters for the frame k are encoded in x_k, then the projection of each $z_i \in Z_k$ on frame k is denoted by y_i that is obtained using (2). $Y_k = [y_1, y_2, \cdots, y_n]^T \in R^{2n}$ is the predicted observation vector whose components are projection of 3D points of Z_k on frame k, obtained individually using (2). In other words, the predicted observation is obtained using a non-linear function, as given in (10).

$$Y_k = h(x_{k|k-1}, Z_k) = \begin{bmatrix} \pi(z_1) \\ \vdots \\ \pi(z_n) \end{bmatrix} \quad (10)$$

In the state update phase of UKF, initially, new sigma points are passed through the non-linear observation model.

$$Y^i_k = h(x^i_{k|k-1}, Z_i) \quad (11)$$

Later, the mean and covariance of the observations are calculated.

$$\begin{aligned} \bar{Y}_k &= \sum_{i=0}^{2L} W^i_m Y^i_k \\ P_{Y_k Y_k} &= \sum_{i=0}^{2L} W^i_c [Y^i_k - \bar{Y}_k][Y^i_k - \bar{Y}_k]^T + R_k \end{aligned} \quad (12)$$

where R_k is covariance of the observation noise. The cross-covariance of state-observation can be calculated using (13).

$$P_{x_k Y_k} = \sum_{i=0}^{2L} W^i_c [x^i_{k|k-1} - \bar{x}_{k|k-1}][x^i_{k|k-1} - \bar{x}_{k|k-1}]^T \quad (13)$$

Kalman gain (κ_k) and innovation vector (b_k) are obtained, as seen in (14).

$$\begin{aligned} \kappa_k &= P_{x_k Y_k} P^{-1}_{Y_k Y_k} \\ b_k &= \bar{Y}_k - U_k \end{aligned} \quad (14)$$

The updated state and of the filter is calculated by adding the predicted state with weighted innovation vector.

$$x_k = x_{k|k-1} + \kappa_k b_k \quad (15)$$

The updated covariance matrix of filter state is calculated using (16).

$$P_{k|k} = P_{k|k-1} - \kappa_k P_{y_k y_k} \kappa^T_k \quad (16)$$

3.3. Feature extraction and tracking

Detecting salient and distinguished feature points, which can be accurately tracked along a sequence of video frames, is of great importance in the feature-based camera tracking problem. In this work, the FAST feature detector [23] is employed to extract the salient feature points. Once new features are detected, they should be tracked in successive frames. In camera tracking process, the feature points should be tracked with as much accuracy as possible. This is because feature point selection accuracy directly affects the performance of the camera pose estimation. Due to small displacement of feature points within consecutive frames, feature points can be tracked easily using correlation window. In the reported work, a hierarchical technique is employed [24] to improve the precision of the feature tracking process. To do so, at first, a multi-level pyramid of a window whose center is located at the feature position is constructed. In the constructed pyramid, the Lucas-Kanade iterative algorithm [25] is used to compute the motion vector at each level. The Lucas-Kanade method performs image registration between two images to align them within the best possible way. The spatial intensity gradient information is used to conduct search procedure to find the best matching points. This

coarse to fine process hierarchically proceeds from the coarsest level down to the finest level of the produced pyramid. The resulted matching point in each level is determined as an initial guess in the next level. This pyramidal routine introduces a motion vector for each feature point, which indicates displacement of the tracked feature point in two successive frames. Another result of foregoing pyramidal tracking is the tracking score, which measures the similarity between the image points considered as matched features in two successive frames. The number of pyramid levels is often set to 3 or 4 for the VGA quality images. However, for high quality videos, it is better to use more levels for the pyramid.

The existence of repetitive textures and blurriness in video frames may introduce noisy or wrong feature correspondences. In noisy correspondences, the matching error within two successive frames is low (i.e. 3-5 pixels). However, wrong correspondences are often affected from significant displacement error. Noisy and wrong correspondences can be removed using robust estimators. The RAndom SAmple Consensus (RANSAC) algorithm [26] is an outlier removal algorithm that is employed for elimination of noisy and wrong correspondences.

During execution of the algorithm, two groups of feature points are tracked. The first group includes those, which are already triangulated and tracking them in each new frame provides the algorithm with a collection of 3D-2D correspondences. This group of feature points is called fully active features that directly utilized for camera pose estimation. The second group includes those features that are not yet triangulated. This group of features is called the partially active features. The coordinates of every partially active feature, from the frame in which it is detected until the initialization point should be stored. Due to large error in depth estimation, after several frames, some partially active features cannot be initialized anymore, and hence, they are no longer tracked. In the proposed method, any partially active feature whose depth is not estimated after 20 frames will be ignored, and consequently, removed from the partially active features list.

3.4. Initial map construction

The proposed system uses a metric map, which is initialized after few frames. The initial pose of the camera is computed using a chessboard placed in front of the camera. The cell size of the chessboard is known, and hence, pose of the camera in the first frame is obtained using a collection of 3D-2D correspondences from 3D coordination of chessboard cell corners and their projections on the captured frame. There is no way to estimate the depth of newly extracted features except using the structure of features with pre-determined 3D position. In fact, using a collection of 2D-2D feature matchings in two or more frames, the depth of the corresponding features is estimated up to scale. Prior information on the 3D geometry of the observed scene is required to perform a metric visual SLAM. Generally, a sparse group of initialized landmarks is enough for this purpose [27].

In the reported experiments, a marker-based strategy is pursued to estimate the camera pose parameters in the initial frames. Accordingly, a chessboard whose cell size is known is placed on a computer desk. As depicted in Figure 2, the chessboard lies on the XY plane of the world coordinate frame and the Z axis is perpendicular to it. Additionally, one can easily detect the corners of the chessboard. Since their 3D coordination in the world coordinate frame is available, a set of 3D-2D feature correspondences is collected, which is sufficient for computation of the camera pose parameters using solutions of the PnP problem. Meanwhile, those feature points that extracted in the first frame are tracked. Having 2D coordinate of feature points in the first few frames, the initial map is constructed. This job is feasible since the camera pose parameters in these frames are available.

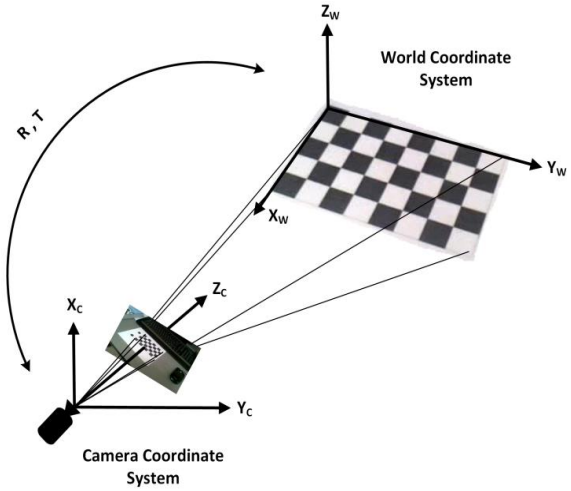

Figure 2. View of world and camera coordinate systems in first frame.

In the reported work, we only employ the natural feature points to obtain the 6Dof camera pose parameters. Finding the depth value for newly detected feature points is performed using the camera pose parameters of both the reference frame and the frame in which the feature displacement exceeds its threshold.

3.5. Depth estimation of new features

Camera pose estimation and map extension are tightly coupled issues in visual SLAM since precision of each one is directly affecting the efficiency of the other. Map extension includes triangulation of newly detected feature points, which is prone to a remarkable error in the presence of noisy feature correspondences. If the camera explores previously visited scenes, then one may try to detect loop and alleviate quick growth of camera position drift. In the reported approach, the problem of loop detection is not addressed. The proposed approach is aimed on achieving promising results through precise data association and reliable camera pose estimation. Therefore, once a new feature is detected, the intention is to initialize it with the highest precision. The new feature cannot be initialized immediately because a single image contains no knowledge about the depth of its points. Calculating the depth of each new feature point, it requires having it present in at least two frames. Since successive frames of a given video forms narrow-baseline images, triangulation of any new feature point using two successive frames produces a large error. This error will reduce the accuracy of the camera pose parameters in the future frames.

Dealing with this problem, Davison [28] has employed a particle filter to represent the initial depth of each new feature. The initial depth is represented by a uniform distribution within a predetermined range. The observation of these new feature points in the succeeding frames is then used to modify the initial depth. This routine is continued until a Gaussian posteriori distribution with small variance is resulted. Eade and Drummond [29] have pursued a similar method. However, they considered the initial distribution for inverse depth of each feature instead of its depth. The foregoing methods suffer from a significant error in feature initialization once the projections of a scene point on subsequent frames are small.

In the proposed approach, the linear triangulation approach is employed to initialize the features. However, this calculation is delayed until the displacement of its projections on subsequent frames exceeds a predefined threshold.

This threshold is set to 30 pixels in our experiments. Having enough number of fully active features available, the depth estimation of the newly extracted features can be postponed since there is no need for their immediate initialization.

3.6. Feature management

The proposed approach utilizes an automatic feature management procedure. It is obvious that once the camera moves, its pose changes and some parts of the scene will disappear from camera field of view while some new scenery will come in to camera's view. In this way, on some occasions, it is required to add new features to the map, and sometimes it is required to remove a few of feature points from the currently applied map. In an efficient camera tracking algorithm, these decisions should be made automatically. Also it is required to make decision about the optimized number of feature points needed for the map. However, given only four non-coplanar 2D-3D point matchings, it is possible to compute the camera position and orientation. Attaining more reliable and robust results, it is recommended to constantly track as many feature points as possible. Using more features points in the camera tracking routine improves the accuracy of the result at the expense of efficiency reduction. In the reported experiments, at least 60 feature points are continuously tracked. Controlling the computation cost of the algorithm, the maximum number of feature points in each frame is limited to 150 feature points. Furthermore, the extracted feature points are selected in such a way that they are uniformly distributed across the whole frame, and hence, the observed area is properly represented.

4. Experimental Results

In the conducted experiments, two video sequences taken with a freely moving handheld camera are used. In Table 1, the details of the video sequences are summarized. The video sequences are captured at 30 fps and the place where the sequences are captured is a cluttered computer desk. To obtain ground truth data for the camera pose parameters, a chessboard pattern is placed on the desk since chessboard corners can easily detect the camera position and orientation are effortlessly calculated using a group 3D-2D correspondences. It is worth noting that the detected points on chessboard are exclusively used for calculation of the ground truth pose parameters. Therefore, the detected corner points of the chessboard squared patterns will be treated as regular features, which are required to be initialized in spite of their known 3D position with respect to the world coordinate system.

Before preparation of the aforementioned video sequences, the camera is calibrated using a flexible method introduced by Zhengyou [30]. To

this end, a collection of images taken from a planar chessboard are utilized. The chessboard cell size is known, and images are taken from different viewpoints.

Table 1. Specifications of used video sequences.

	Frame count	Resolution	Avg. angular velocity (deg/s)	Avg. translational velocity (m/s)
Seq 1	1052	640×480	13.82	0.14
Seq 2	1091	1280×720	10.58	0.13

Fig. 3 shows that some feature points are tracked until their depths are calculated. The marked magenta squares indicate fully active feature points and small blue squares are newly extracted feature points. New feature points are tracked along the following few frames (green dots in Figure 3(b)). Every new feature point is triangulated (yellow asterisks in Figure 3(b)) once its 2D coordinate has an adequate displacement with respect to the frame in which it is detected.

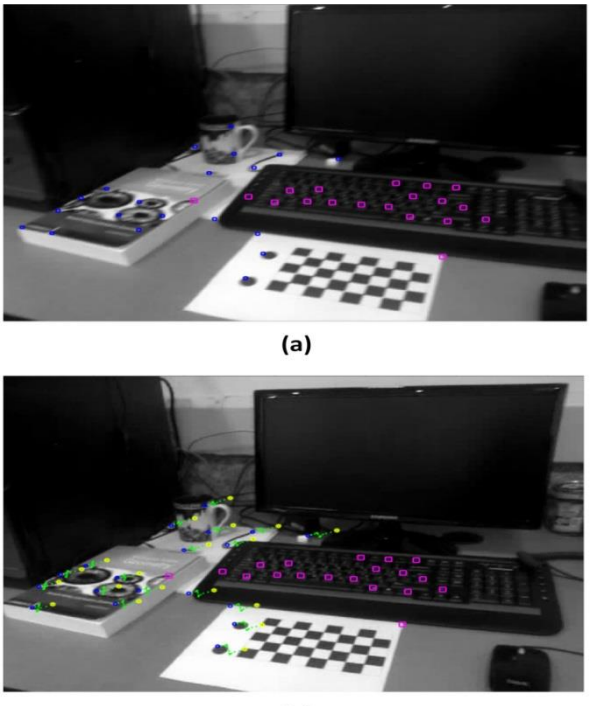

Figure 3. (a) Fully active features + partially active features (b) Fully active features along with tracking path of partially active features until initialization.

In Figure 4, projection of the estimated camera trajectories along with the projection of ground truth trajectories on XY plane for Seq 1 (Figure 4 (a)) and Seq 2 (Figure 4 (b)) are depicted. As it can be seen, the estimated trajectories are tracked camera path with high precision.

In Figure 5, the components of translation part of camera pose along with ground-truth data for both sequences are shown. As it is illustrated in Fig. 5, in spite of long duration of the input sequences, camera trajectories are tracked with a high accuracy. Moreover, the translation errors in x and y directions for both sequences are negligible, and only a small error is seen along the z axis.

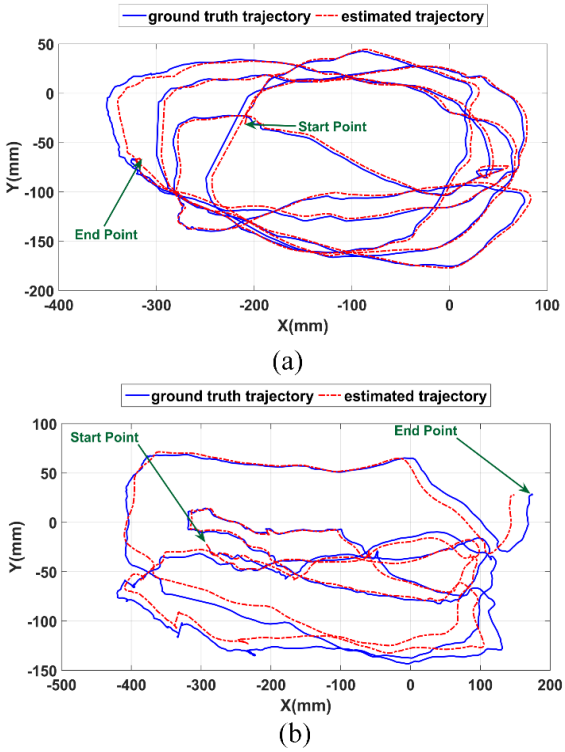

Figure 4. Ground truth and estimated camera trajectories projected on XY plane (a) for Seq 1 and (b) for Seq 2.

Furthermore, the proposed UKF-based algorithm is compared against EKF and non-linear minimization solution of PnP (NLPnP), as given in (3) using the LM method. For comparison, the Relative Pose Error (RPE) and Absolute Trajectory Error (ATE) [31] criterions for both translation and rotation parts of the camera pose are used.

Generally, RPE is a measure for the local accuracy of the algorithm that indicates the trajectory error between the successive frames, and ATE specifies the global consistency of the estimated trajectory.

Table 2 shows the obtained results in terms of RPE for the two sequences introduced in Table 1. In Table 2, the reported results for the camera's translation part of RPE show that the UKF-based approach outperforms the EKF and NLPnP methods. However, the RPE values for camera rotation in both sequences show that neither of them has a significant superiority over the others.

Table 3 reports the performance of the proposed algorithm as well as the EKF and NLPnP approaches using the ATE measure. In terms of the translation part of ATE, one can easily see that our algorithm produces more promising results. Also similar to RPE, the obtained results for the rotation part of ATE are approximately the same among the UKF, EKF, and NLPnP methods.

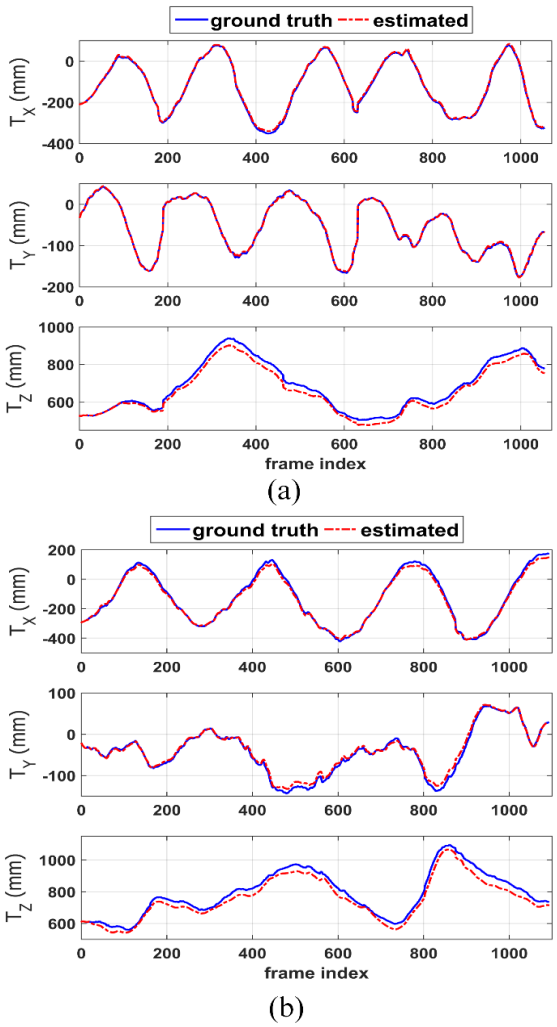

Figure 5. Ground truth versus estimated camera position (a) for Seq 1 (b) for Seq 2.

In order to investigate the influence of feature detection method on the performance of camera tracking routine, we repeated the proposed algorithm using well-known feature point detection methods, i.e. FAST, HARRIS [32], and MINEIGEN [33]. The quantitative results in terms of the RPE and ATE criterions for different feature extractors are reported in Table 4 and Table 5. As presented in these two tables, the reported results (especially for RPE) for both sequences are almost the same. As a conclusion, the type of feature extractor has no significant impact on the accuracy of the estimated trajectories.

Table 2. Comparison of translation and rotation parts of RPE for different camera pose estimation methods.

Tracking method	Seq 1		Seq 2	
	RPE_{Trans} (mm)	RPE_{Rot} (deg)	RPE_{Trans} (mm)	RPE_{Rot} (deg)
UKF	1.7	0.4	2.1	0.7
EKF	1.9	0.4	2.3	0.8
NLPnP	2.9	0.4	4.1	0.5

Table 3. Comparison of translation and rotation parts of ATE for different camera pose estimation methods.

Tracking method	Seq 1		Seq 2	
	ATE_{Tran} (mm)	ATE_{Rot} (deg)	ATE_{Tran} (mm)	ATE_{Rot} (deg)
UKF	25.7	7.2	38	8.3
EKF	25.7	7.7	38.4	7.9
NLPnP	195.4	7.4	172.9	7.9

Table 4. Comparison of translation part of RPE and ATE for different feature extractors.

Feature type	Seq 1		Seq 2	
	RPE (mm)	ATE (mm)	RPE (mm)	ATE (mm)
HARRIS	2.2	38.9	3.2	53.5
MINEIGEN	2.2	35	3	53.2
FAST	2.2	36.2	3	59.3

Table 5. Comparison of rotation part of RPE and ATE for different feature extractors

Feature type	Seq 1		Seq 2	
	RPE (deg)	ATE (deg)	RPE (deg)	ATE (deg)
HARRIS	0.8	8.3	1.6	7.8
MINEIGEN	0.8	6.8	1.5	7.9
FAST	0.7	7.8	1.6	8.7

5. Conclusion

In this paper, the problem of camera tracking in unknown environments was addressed. The proposed method could be used in any AR or visual SLAM application. In the reported work, a feature-based strategy was adopted to locate the camera. The proposed algorithm initializes a metric map. Using a metric map, the scale ambiguity problem for the parameters of the camera translation and the extended map is solved. Furthermore, the constructed map is extended solely based on the initially triangulated feature points and the knowledge acquired by tracking of detected feature points. Hence, with any increase in the number of input video frames, the cumulative error for the camera pose parameters will also increase. One strategy to handle this issue is to employ loop detection algorithms or to use known markers in the scene.

A fundamental property of our approach is that promising results are obtained without utilizing any optimization method. The intention is to develop a more accurate map extension strategy to reduce the accumulation of camera pose error.

References

[1] F.e. Ababsa and M. Mallem, "Robust camera pose estimation using 2d fiducials tracking for real-time augmented reality systems," in *Proceedings of the ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, Singapore, 2004, pp. 431-435.

[2] K. Xu, K. W. Chia, and A. D. Cheok, "Real-time camera tracking for marker-less and unprepared augmented reality environments," *Image and Vision Computing*, Vol. 26, pp. 673-689, 2008.

[3] Z. Dong, G. Zhang, J. Jia, and H. Bao, "Efficient keyframe-based real-time camera tracking," *Computer Vision and Image Understanding*, vol. 118, pp. 97-110, 2014.

[4] L. A. Clemente, A. J. Davison, I. D. Reid, J. Neira, and J. D. Tardós, "Mapping Large Loops with a Single Hand-Held Camera," in *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, 2007, pp. 352-360.

[5] E. Eade and T. Drummond, "Unified Loop Closing and Recovery for Real Time Monocular SLAM," in *Proceedings of the British Machine Vision Conference*, Leeds, UK, 2008, pp. 136-145.

[6] O. Guclu and A. B. Can, "Fast and Effective Loop Closure Detection to Improve SLAM Performance," *Journal of Intelligent & Robotic Systems*, Vol. 93, pp. 495-517, 2019/03/01 2019.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd Ed. New York, NY, USA: Cambridge University Press, 2003.

[8] A. W. Fitzgibbon and A. Zisserman, "Automatic camera recovery for closed or open image sequences," in *Proceedings of 5th European Conference on Computer Vision*, Freiburg, Germany, 1998, pp. 311-326.

[9] M. Pollefeys, R. Koch, and L. Van Gool, "Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters," in *Sixth International Conference on Computer Vision*, Bombay, India, 1998, pp. 90-95.

[10] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, Nara, Japan, 2007, pp. 1-10.

[11] T. Pire, T. Fischer, G. Castro, P. De Cristóforis, J. Civera, and J. Jacobo Berlles, "S-PTAM: Stereo Parallel Tracking and Mapping," *Robotics and Autonomous Systems*, Vol. 93, pp. 27-42, 2017/07/01/ 2017.

[12] R. Mur-Artal, J. M. M. Montiel, J. D. Tard, and x00F, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, Vol. 31, pp. 1147-1163, 2015.

[13] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, Vol. 33, pp. 1255-1262, 2017.

[14] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," in *Proceedings of 13th European Conference on Computer Vision*, Zurich, Switzerland, , 2014, pp. 834-849.

[15] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D Mapping With an RGB-D Camera," *IEEE Transactions on Robotics*, vol. 30, pp. 177-187, 2014.

[16] O. Guclu and A. B. Can, "k-SLAM: A fast RGB-D SLAM approach for large indoor environments," *Computer Vision and Image Understanding*, Vol. 184, pp. 31-44, 2019/07/01/ 2019.

[17] M. Maidi, F. Ababsa, M. Mallem, and M. Preda, "Hybrid tracking system for robust fiducials registration in augmented reality," *Signal, Image and Video Processing*, vol. 9, pp. 831-849, 2015.

[18] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, pp. 1052-1067, 2007.

[19] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle Adjustment — A Modern Synthesis," in *International workshop on vision algorithms*, Corfu, Greece, 1999, pp. 298-372.

[20] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011, pp. 3607-3613.

[21] E. A. Wan and R. V. D. Merwe, "The unscented Kalman filter for non-linear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153-158.

[22] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Tenth IEEE International Conference on Computer Vision*, Beijing, China, 2005, pp. 1508-1515.

[23] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," 2001.

[24] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th international joint*

conference on Artificial intelligence, Vancouver, BC, Canada, 1981, pp. 674-679.

[25] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, Vol. 24, pp. 381-395, 1981.

[26] Q. Long and L. Zhongdan, "Linear N-point camera pose determination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, pp. 774-780, 1999.

[27] A. J. Davison, "Real-Time Simultaneous Localisation and Mapping with a Single Camera," in *Proceedings of the Ninth IEEE International Conference on Computer Vision*, Nice, France, 2003, pp. 1403-1410.

[28] E. Eade and T. Drummond, "Scalable Monocular SLAM," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 469-476.

[29] Z. Zhengyou, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, pp. 1330-1334, 2000.

[30] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, 2012, pp. 573-580.

[31] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, 1988, pp. 147–151.

[32] J. Shi and C.Tomasi "Good features to track," in *9th IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, 1994, pp. 593-600.

یک روش مبتنی بر فیلتر کالمن غیرمعطر برای تخمین مسیر حرکت دوربین در فضای داخل ساختمان

سیدعلی حسینی[1،*] و پیمان کبیری[2]

[1] دانشکده مهندسی برق و کامپیوتر، دانشگاه بیرجند ، بیرجند، ایران.

[2] دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران.

ارسال ۲۰۲۲/۰۱/۰۵؛ بازنگری ۲۰۲۲/۰۴/۰۷؛ پذیرش ۲۰۲۲/۰۶/۱۶

چکیده:

هنگامی که دوربین در یک محیط ناآشنا حرکت می‌کند، تخمین موقعیت و جهت دوربین برای بسیاری از کاربردهای حوزه بینایی ماشین و رباتیک مهم است. دنبال نمودن مسیر حرکت دوربین احتمالا پر چالش‌ترین بخش مسائلی نظیر مکان‌یابی و نقشه‌سازی همزمان و واقعیت افزوده است. این مقاله یک روش مبتنی بر ویژگی برای دنبال نمودن مسیر حرکت دوربین پیشنهاد می‌دهد. محیط حرکت دوربین در فضای داخل ساختمان درنظر گرفته شده و حداکثر عمق و فاصله اشیاء تا دوربین حدود ۴ تا ۵ متر است. در چند فریم آغازین، دوربین با دیدن یک صفحه شطرنجی، یک نقشه اولیه از محیط پیرامونی ساخته و سامانه پیشنهادی راه‌اندازی می‌شود. در ادامه با رسیدن هر فریم جدید روال تعقیب مسیر حرکت دوربین دنبال می‌شود. این روال درون یک چارچوب که تنها از ویژگیهای استخراج شده طبیعی در کنار نقشه اولیهِ ساخته شده استفاده می‌کند، انجام می‌شود. همزمان با مشاهده نواحی جدید توسط دوربین، نقشه اولیه ساخته شده از محیط نیز گسترش می‌یابد. درضمن، سامانه پیشنهادی از یک روش سلسله مراتبی بر مبنای روش ثبتِ تصاویرِ لوکاس-کاناده برای تعقیب نقاط ویژگیِ استخراج شده بهره می‌برد. الگوریتم پیشنهادی بر روی ویدیوهای واقعی آزمایش شده و کارایی فیلتر کالمن غیر معطر با سایر روشهای تعقیب دوربین، مقایسه شده است. دو سنجه خطای نسبی موقعیت دوربین و خطای مطلق مسیر دوربین برای ارزیابی عملکرد الگوریتم پیشنهادی مورد استفاده قرار گرفته است. آزمایشهای انجام شده حاکیست روش پیشنهادی از دقت بالایی برخوردار بوده و علاوه بر آن، نوع ویژگی استخراج شده تاثیر شایانی بر دقت عملکرد الگوریتم ندارد.

کلمات کلیدی: استخراج ویژگی، فیلتر کالمن غیرمعطر، بینایی ربات، مکان‌یابی و نقشه‌سازی همزمان، دنبال کردن دوربین.