Multimed Tools Appl (2018) 77:7955–7976
DOI 10.1007/s11042-017-4694-x

Improved visual SLAM: a novel approach to mapping and localization using visual landmarks in consecutive frames

Kajal Sharma[1]

Received: 19 May 2016 / Revised: 8 March 2017 / Accepted: 6 April 2017 /
Published online: 19 April 2017

Abstract Pathfinding is becoming more and more common in autonomous vehicle navigation, robot localization, and other computer vision applications. In this paper, a novel approach to mapping and localization is presented that extracts visual landmarks from a robot dataset acquired by a Kinect sensor. The visual landmarks are detected and recognized using the improved scale-invariant feature transform (I-SIFT) method. The methodology is based on detecting stable and invariant landmarks in consecutive (red-green-blue depth) RGB-D frames of the robot dataset. These landmarks are then used to determine the robot path, and a map is constructed by using the visual landmarks. A number of experiments were performed on various datasets in an indoor environment. The proposed method performs efficient landmark detection in various environments, which includes changes in rotation and illumination. The experimental results show that the proposed method can solve the simultaneous localization and mapping (SLAM) problem using stable visual landmarks, but with less computation time.

Keywords SLAM · Localization · Landmarks · Video processing

1 Introduction

Localization and mapping are becoming more important for the field of pathfinding in various challenging environments where the goal is to obtain the camera trajectory and a map from sensor data. Simultaneous localization and mapping (SLAM) began with the robotics community in mid-1986 with the development of a concrete representation of uncertainty in feature location by Smith and Durrant-Whyte. A major practical finding was introduced to deal with errors, and it was done with a combination of sensor readings (a laser scanner or sonar) and

✉ Kajal Sharma
kajal175@gmail.com

[1] Independent Researcher, Flat No. 301, Building 206, Piorville Apartment, 24 Namyang-Dong, Seongsangu, Changwon, South Korea

information about the control input (e.g., steering angle) and the measured robot state (e.g., counting wheel rotations). Later, the existence of a correlation between feature location errors and errors in motion, which affects all feature locations, was proven by Cheesman, Chatila, and Crowley. And the errors in feature location acquired by agents were correlated with one another. The correlation exists because an error in localization will have a universal effect on the perceived location of all features. The motivation behind solving the SLAM problem is to understand and utilize the relationship among errors in feature locations and robot pose.

A number of algorithms have been proposed for SLAM in the fields of robotics [10, 13, 21, 34, 42] and computer vision [1, 25, 28, 33, 38]. Most of the existing algorithms are based on sonar sensors or two-dimensional (2D) and 3D laser scanners [9, 20, 23, 26, 31]. Recently, visual sensors have become an important aspect of SLAM research, because an image is considered a rich source of information when gathering details of the environment. The goal of visual SLAM is to track a set of points obtained through successive video frames, and to determine the 3D position. On the other hand, robot pose is calculated using the estimated 3D points that have been observed during the movement of the robot. A wide range of sensor modalities was proposed in the past, including monocular cameras [15, 16, 36, 38], stereo systems [5, 17], and the recently developed Kinect sensor [11, 32, 35].

Most of the earlier approaches focused on artificial landmarks, and some approaches are based on detecting and tracking simple features using Harris corners without considering location data [22, 41]. These approaches suffer from the functioning problem in beacon-free environments. However, high computational overhead is needed to maintain the reliability and association of the detected features in cluttered and viewpoint-changed environments. Most of the traditional sensors, such sonar and infrared (IR) sensors, suffer from resolution and accuracy problems. To address these problems, vision-based systems are employed in a variety of robotic applications, including SLAM, object recognition, and obstacle avoidance [2, 8, 18, 27]. A vision sensor is used to select the appropriate reference points that enable the reconstruction of a 3D object, and it can be used in navigation to estimate the pose of a robot with respect to prominent landmark cues [3, 12].

A number of techniques have been proposed to recognize landmarks and to detect points of interest in a scene. Many feature-based algorithms have been proposed to track corner features that enable the creation of 3D structures, but the corners could not be tracked once they are lost. Although these techniques are robust to viewpoint change and lighting conditions, these algorithms are too slow to implement in real-time pathfinding applications. Other feature-based algorithms have been proposed to obtain the points of interest for objects or scenes [14, 29].

In this paper, a visual SLAM algorithm is proposed from detection of viewpoint-invariant landmarks in video frames of indoor environments. The recent scale-invariant feature transform (SIFT) algorithm [19, 30] allows detecting a 3D location that can be used as a landmark to identify a change in position from robot movement. The proposed method detects and recognizes the landmarks in consecutive frames from feature matching using a self-organizing map (SOM) [40], which is an unsupervised neural network method to map n-dimensional input space to a lower-dimensional output map. The SOM is an efficient tool for analyzing a dataset and extracting useful features, and is applied to divide the feature space into subspaces by clustering similar features together. SIFT landmarks are invariant to image translation, scaling, and rotation, and partially invariant to illumination. The robot pose is estimated with these landmark positions and is used for a mapping algorithm to generate a hypothesis about the robot pose and landmark positions. The overall process requires complex and long computations in the SIFT algorithm; thus, this paper introduces a new method for efficient detection of viewpoint-invariant landmarks, along with robot mapping and localization, requiring less computation and providing high accuracy. The landmarks are detected in less time, and a database is maintained to keep distinctive invariant features. In the proposed work, the system is equipped with the Kinect sensor (an RGB-D camera); thus, 3D positions of landmarks can be obtained from a scene. In consecutive images, feature matching is done using winner pixel calculation for the captured video dataset. Hence, a 3D map can be built by using the landmark cues, and the movement of the sensor can be localized simultaneously in three dimensions. The landmark feature database is mapped to the 3D environment and is constantly updated with changes in time and with respect to changes in environment conditions. The detected landmarks will serve as the basis for performing high-level tasks, such as mobile robot navigation and path estimation. The advantages of using SIFT for SLAM is that the invariant landmarks can be obtained using feature matching. The vision information provides the cues for pathfinding by detecting obstacles that are invariant to change in viewpoint. In this work, improved SIFT with SOM feature-matching is used for landmark extraction that provides results that are rotation-invariant, scale-invariant, and illumination- and blurring-invariant.

The research presented in this paper is a novel way to develop a SLAM application using the Kinect sensor. In this research, the work is focused on SLAM using the Kinect sensor, where landmarks are detected using feature matching in consecutive images. The relationships among the Kinect, landmarks, SOM, and SLAM are presented here to clearly explain the proposed problem and the algorithm. Due to the motion-sensing and vision capability of the Kinect sensor, it can acquire a large motion dataset for experimental purposes. The vision datasets gathered by the Kinect are used to obtain landmark information from the scenes at different time instances. However, due to the large amount of dataset information, the landmark database is reduced using an improved SOM feature-reduction method. This results in fast matching of features from different scene information at different time instances. The robot pose is estimated, and the 3D positions are further used to determine SLAM in real time using the Kinect sensor. The application is focused on the development of the SLAM application using the vision sensor. This improved method is novel, in comparison with ordinary SIFT. In addition, the proposed method is better in terms of computation cost, which guarantees landmark detection for real-time processing. The first SIFT stage is used as a base to compute the dataset keypoint feature vector, whereas later stages of the SIFT method are improved using the SOM method. The feature keypoints initially extracted are later passed to the SOM for determination of improved landmarks. The feature sets are reduced using the SOM method, and only stable landmarks are detected and passed to the next stage.

In Section 2, an overview of the depth-sensing process and the calibration method for the Kinect is explained. The proposed method for landmark detection, along with localization and mapping in RGB frames, is described in Section 3. Experimental results using different datasets in the indoor home environment, and a discussion on the findings, are presented in Section 4. The paper concludes with some remarks in Section 5.

2 Overview of the depth sensing process and calibration method for the Kinect RGB-D camera

The Kinect sensor has the ability to grab RGB images and infrared images of 640 × 480 pixels at 30 frames per second (fps). It has an angular field of view that ranges 57 degrees horizontally and 43 degrees vertically. The depth-sensing technology consists of emitting an IR pattern and simultaneously capturing an IR image with an attached complementary metal-oxide semiconductor camera (Fig. 1). The steps of the depth sensing process are detailed as follows. (1) The PrimeSense chip sends a signal to the IR emitter depth sensor, which is mounted as a camera on the Kinect. In actuality, it is an IR projector that has a single transparency with a fixed pattern to project a complex pattern of light dots onto an object. (2) The PrimeSense chip also sends a signal to the IR/Depth sensor to initialize the depth sensor. (3) Electromagnetic radiation is emitted onto objects in front of the camera. The infrared light projected on the objects is invisible, because the wavelengths of the radiation are longer than the wavelengths of visible light. (4) The depth information obtained from the reflected light is captured by the depth sensor, and the invisible dotted data are used to determine an object's distance from the sensor. The resulting dotted data are converted into depth data for further display operations. (5) The coded depth light is returned to the PrimeSense chip. The information is then processed to reconstruct a three-dimensional model of the object using the dot information of the IR light pattern. (6) The processed depth stream is ready to display an output depth image. The depth stream contains the number of depth frames; the pixels in each frame represent the distance information in millimeters.

The intrinsic parameters of the depth and RGB camera, as well as the pose difference between the two cameras of the Kinect, should be known for accurate 3D map–building based on the 2D depth images. The calibration parameters define the relation between the image measurements (x, y, d') and object coordinates (X, Y, Z) of each point. The camera's intrinsic calibration parameters is used to generate a point cloud from each disparity image. The calibration parameters of the infrared camera do not directly correspond to the disparity images

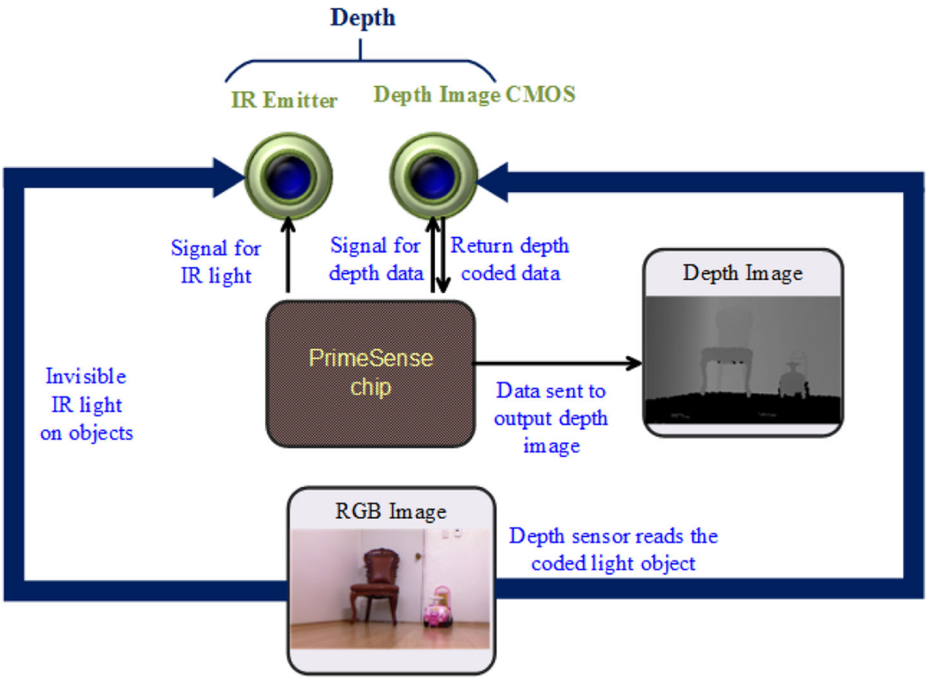


Fig. 1 Details of the depth sensing process: PrimeSense chip processing with an IR emitter and an IR/Depth sensor to form a depth image as output

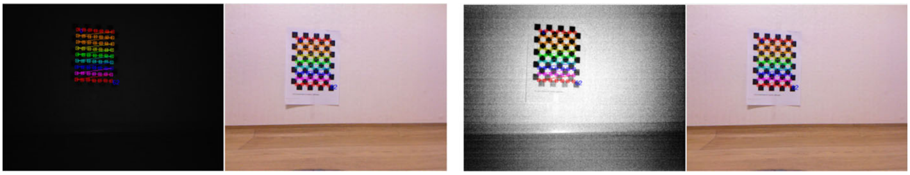
due to the bandwidth limitation of the universal serial bus (USB) connection. The size of the disparity images computed by the PrimeSense chip processor is smaller than the actual size of the infrared sensor. The infrared images are streamed at a reduced size of 640 × 480 pixels, corresponding to each disparity image. A pixel-by-pixel correspondence is performed over the reduced infrared images and the disparity images.

In the proposed method, the intrinsic calibration parameters for both RGB and IR/Depth cameras were estimated using the mobile robot programming toolkit (MRPT) [4] library. As a result of this calibration, the focal lengths (*fx*/*fy*), the optical center (*cx*/*cy*), and distortion parameters are obtained for both RGB and IR/Depth cameras. Fig. 2 shows the experiments performed for calibration to obtain the intrinsic parameters. The parameters obtained for the various sequences are summarized in Table 1.

3 Proposed method for landmark detection and localization

In this research, a new pathfinding approach is presented to obtain an obstacle-free path in the indoor environment by using stable landmarks. A Kinect sensor is attached to a four-wheeled mobile vehicle, which is used to capture a dataset for use by robots in an indoor environment. The consecutive RGB-D frames in the video dataset are then matched to estimate stable landmarks between pairs of RGB frames. Experiments were conducted under variable conditions (rotation, scaling, noise, affine, and so forth) to estimate the paths using stable landmarks with fewer computations. To better understand the proposed visual SLAM method, SOM-based stable landmark estimation to reduce the dimensions of the matched features in the consecutive frames is explained first, and then the SLAM method with landmarks to determine a path with improved matched landmarks is discussed.

SOM is a neural network that is used for visualization and abstraction of high-dimensional data through competitive unsupervised methods. In the proposed method, the robot's dataset is post-processed for offline image processing to generate a landmark database. A large dataset of landmarks detected from the frames of the video sequence needs to be reduced in order to enhance computation performance. The main advantages of using the SOM are computational efficiency and the intuitive way the results are presented to the user. A SOM is a prominent tool for data exploration, having capabilities for automated organization of digital libraries (for example, feature datasets in video frames). The motivation behind using a SOM is its data exploration capability to reduce the large dataset of features detected in this application. The overall clustering performance of the SOM is better, compared to other clustering methods such as the hard k-means algorithm (HKM) and the fuzzy k-means algorithm (FKM), and it also performs well for detection of noisy documents and topology preservation, thus making it



(a) Detected corners for a 7 x 9 chessboard. (b) Re-projected corners for the 7 x 9 chessboard.

Fig. 2 Checkerboard used for calibration: **a** detected corners for a 7 × 9 chessboard, and **b** re-projected corners for the 7 × 9 chessboard

Table 1 Intrinsic parameters of the RGB and IR/Depth cameras of the Kinect sensor, including focal length (*fx*/*fy*) and optical center (*cx*/*cy*)

Checkboard size (7 × 9)	Camera	*fx*	*fy*	*cx*	*cy*
Dataset 1	Infrared/Depth	556.70	533.59	−285.32	163.77
	RGB	542.21	595.10	413.64	224.60
Dataset 2	Infrared/Depth	285.48	298.72	515.45	241.30
	RGB	647.44	655.94	381.85	275.21

more suitable for some applications, such as navigation of document collection, multi-document summarization, etc.

In the proposed method, the RGB-D Kinect frames are used to obtain features from consecutive frames of the captured video dataset. In each RGB frame, the high-dimensional feature descriptor sets are extracted from the SIFT method, which are then used as input for the SOM. The whole algorithm is shown in steps in Fig. 3. Let RGB_I and RGB_{I+1} denote the consecutive frames in the video captured by the Kinect sensor. The points of interest are detected to generate SIFT descriptors, and then these consecutive frames are used to perform matching in consecutive frames in the video using the SOM method. Consecutive frame matching is performed for the creation of a landmark database, which is then used to estimate the robot pose and path. The neurons in the SOM network are arranged in a rectangular, hexagonal, or circular topological grid structure. In order to estimate distinctive landmarks, a SOM is applied to nonlinear projection of a multivariate feature set into a low-sized feature space. The projection and clustering of input feature space are performed with competitive learning and the preservation of input feature information in a low-sized output neuron grid.

The performance of a Kohonen self-organizing map is compared to other clustering methods, namely, the HKM and FKM algorithms. For comparison, four different normal distributed datasets were used; two or three dimensions and 1000 or 10,000 vectors were used per cluster. In the 2D case, the mean vector for the first cluster was $m1 = [0.3, 0.3]$, and for the second cluster, it was $m2 = [0.7, 0.7]$. In the 3D case, the mean vector for the first cluster was $m1 = [0.3, 0.3, 0.3]$, and for the second cluster, it was $m2 = [0.627, 0.627, 0.627]$.

The value of m is determined to be the mean of all the patterns within the vector. It can also be a random value that ranges between the values of the input data vectors. In the SOM method, the mean vector is a random value between 0 and 1. The weight values are assigned randomly according to the size of the input feature vector. In 1D SOM, weight vectors of processing elements correspond to cluster mean vectors, whereas in a 2D map, one or several weight vectors correspond to one cluster. The process of clustering is done via estimation of a density function of the data, which is accomplished by finding the winner element in each of the input vectors. The position of the weight vectors of the processing elements in the SOM are distributed according to the density function of the input vectors. The mean vector is computed by searching the nearest weight vectors and computing the local mean of the neighborhood of the weight vector.

The performance of different algorithms was compared with respect to quantization error and its deviation. The comparison results with mean value and deviation values are given in Table 2, which shows that the best results are obtained in the SOM clustering method. It can be noted from Table 2 that the value of the quantization error in FKM is smaller than the error in HKM, at first; but later, a decrease in error for FKM was slower than HKM. When the number of clusters was increased to eight from six, the error value from FKM increased because every

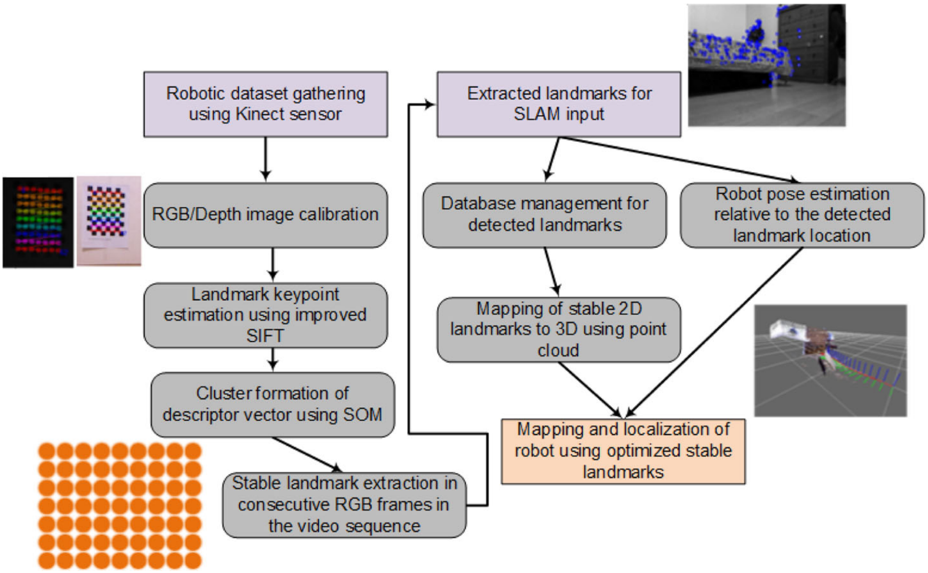

Robotic dataset gathering using Kinect sensor
RGB/Depth image calibration
Landmark keypoint estimation using improved SIFT
Cluster formation of descriptor vector using SOM
Stable landmark extraction in consecutive RGB frames in the video sequence
Extracted landmarks for SLAM input
Database management for detected landmarks
Robot pose estimation relative to the detected landmark location
Mapping of stable 2D landmarks to 3D using point cloud
Mapping and localization of robot using optimized stable landmarks

Fig. 3 Process for mapping and localization using stable landmarks and improved landmark matching between consecutive frames

feature vector affects the evaluation of the cluster mean vectors. On the other hand, the error value in HKM is smaller, compared to the error value in FKM, but deviations are noticeably larger. HKM is very sensitive to first values of cluster mean vectors evaluated by random number generator. In a real dataset, the cluster centers move close to each other, and two different cluster centers can move to the same place in feature space. The best results are obtained in SOM clustering with the smallest quantization error and its deviation (Table 2) for all the different cluster values. In all cases, the SOM was able to converge to (almost) the same solution every time, and performed best, compared to HKM and FKM.

The pixels values were normalized between [0, 1] before the clustering process. The number of clusters used for clustering ranged between 2 and 8. For different clusters, SOM performance was the best in comparison with HKM and FKM and quantization error, and their deviations were smallest in the SOM method. This resulted in the conclusion that the placement of cluster centers was best in the SOM. The original image was compared to the

Table 2 Comparison results with mean value and deviation values for the hard k-means algorithm (HKM), the fuzzy k-means algorithm (FKM), and the self-organizing map (SOM) method

Number of Clusters	Hard K-Means (mean)	Hard K-Means (deviation)	Fuzzy K-Means (mean)	Fuzzy K-Means (deviation)	SOM Clustering (mean)	SOM Clustering (deviation)
2	6461.7.2	3068.4	4987.6	0.0	4982.2	0.1
3	3871.0	2031.4	3806.0	0.9	2842.7	0.0
4	2355.2	883.8	3131.7	181.1	1640.2	0.9
5	1652.6	601.7	2673.6	189.5	1271.5	107.5
6	1195.3	185.7	2175.6	151.3	908.0	24.2
7	1026.5	359.7	2201.9	37.4	746.1	0.2
8	822.3	159.0	2323.7	27.7	653.6	12.8

clustering result, and the SOM performed best when examined visually. In FKM, some clusters were not natural, whereas the clusters in HKM were unstable. The SOM performed best, and is the reason for its use in this application.

The different experiments were performed with a SOM with different parameter settings. The size of the SOM grid selected was a 7 X 7 neuron grid for better performance of the feature set during the training process. The experiments were conducted for 7 X 7, 10 X 10, and 12 X 12 grids. The best results were obtained for the 7 X 7 grid, in terms of both optimum features and computation. In the experiments, a SOM with gain term $a_o = 0.02$ with a linear decrease in time, and neighborhood $NE_0 = 1$ were chosen.

3.1 Landmark clustering with SOM

In this work, the vectors extracted from SIFT are used to compose a topological map. SIFT is an efficient method to extract a set of keypoints in the RGB frames. It allows matching under numerous image transformations (i.e., rotation, scale, perspective) and generates a dense set of image features. At first, the large variety of SIFT patterns extracted from consecutive RGB frames in landmark space increases the potential difficulty in obtaining discriminate boundaries for classifying patterns into landmark classes by using only one classifier. The SOM is applied to divide the landmark space into subspaces by clustering similar landmarks together and representing each landmark cluster as a node on an output low-dimensional topological grid. Fig. 4 shows the clustering process and mapping of SIFT feature space onto a 2D neuron grid map, and each node in the SOM grid is connected with weight vector $w_i = [w_1 \ldots . w_n]$ T in the grid map. The landmarks are defined by using the matched features obtained by the SOM matching method. The matched features in the different RGB frames are designated as landmarks. The positions of these matched features in different frames are defined as landmarks, which are defined as the ground truth. If a moving object comes, then it will be treated as a landmark. The landmarks are determined by an object's motion in the subsequent frames of the videos. As the proposed method is invariant to change in scale and other viewpoint changes, its matched features can be extracted if the object positions change in the different frames due to motion. The appearance of any moving object is considered a target, and landmarks are extracted for the objects or person appearing.

The proposed approach operates in the SIFT feature space instead of a Kinect RGB database; in other words, the landmark database consists of a reduced feature set and the

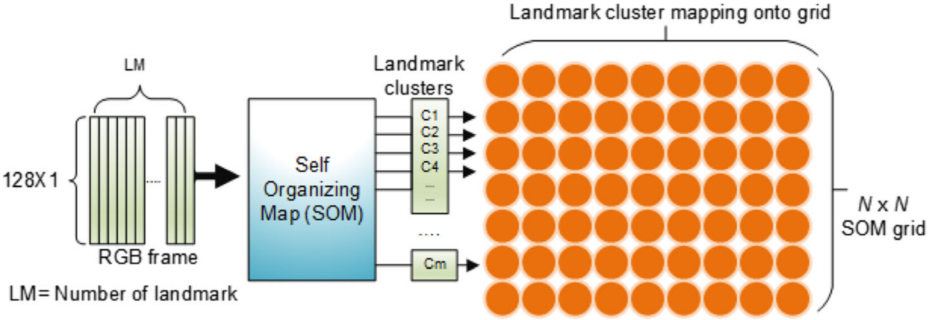


Fig. 4 Cluster formation of SIFT landmarks. The descriptor is mapped onto a SOM grid to detect the reduced set of features, which results in landmarks

feature pose with respect to a stable, distinctive, detected landmark at position (x, y). The set of descriptor spaces in the RGB frames are represented in terms of a reduced Kohonen map, in which the feature values are distributed over an $n \times n$ grid. When the new descriptor vector arrives, the topological map determines the best matching neuron using the concept of a nearest neighbor learning algorithm. Let $X_i = [RGB_1, RGB_2, RGB_3 \ldots \mathrm{RGB}_i] \in V_n$ be the set of input feature vectors, and W_i is the weight vector for each node connected to the SOM grid. The video datasets are captured during navigation in the indoor environment with different viewpoints. For the clustering process, there will be one winning neuron, N_i, the neuron where the weight vector lies closest to the input descriptor vector. The winner neuron represents the best-matched neuron (BMU) and the corresponding pixel location is included in the landmark database. During the learning process, the neurons that are connected to adjacent neurons and that are close in the topological grid map will activate one another, and the neighborhood function can be represented as

$$h_{ci}\mu(t) = \eta(t).\exp\left(-\frac{\|r_c - r_i\|}{2.(\sigma(t))^2}\right) \tag{1}$$

where $\eta(t)$ is the learning rate, and it depends on the number of iterations during the training process; $\eta(t)$ gradually decreases linearly as a function of time to reduce the neighborhood region in successive training iterations:

$$\eta(t) = a_0\left(1 - t\middle/\mathrm{tmax}\right) \tag{2}$$

where a_0 and *tmax* denote the initial learning rate and a maximum number of iterations of the training cycles, respectively. For each iteration, the input RGB frame is applied, and the winner neuron is calculated based on the Euclidean distance between the input feature vector and the node in the topological grid. The winning neuron represents the pixels in the consecutive frames that could be a match in the current frame and can be represented by

$$|x - W_c| = \min|x - W_i| \tag{3}$$

The nodes in the output SOM grid map will activate one another to learn from the same RGB frames. Feature information during the learning and the weights of the nodes in the grid map are updated by the following equation:

$$W_i(t+1) = W_i(t) + h_{ci}(t)[x(t) - W_i(t)] \tag{4}$$

The learning process repeats iteratively until all the input feature patterns in the RGB frames are mapped onto the SOM topological grid, and all the frames' feature patterns are clustered into each node in the SOM map. The process ends when the value of t reaches *tmax*, which indicates that the maximum limit of the training cycles was reached.

3.2 Localization and mapping using detected landmarks in RGB frames

In the proposed approach, the landmark vector in the first RGB frame is used as input in the localization and mapping process to determine the path of the robot. The overall localization procedure consists of multiple stages: stable landmark extraction in consecutive environment views, data association, state estimation, state update, and landmark database update. For each new robot position, stable landmarks are extracted from the new RGB frames of the environment, and matching is performed in the consecutive frames. A pixel is selected randomly in the new frame, and the corresponding feature vector is supplied to the SOM to detect the stable features. The matched landmarks in the consecutive views are then associated with observations of landmarks seen previously. The robot position is updated in the extended Kalman filter (EKF) using the re-observed landmarks. The newly detected landmarks are added to the EKF as new observations, so they can be re-observed in later stages.

The robot needs to build a map without external control and without a given map; it needs to navigate in an indoor environment with the estimation of stable landmarks in the different views that appear in front of the camera during navigation. The map is constructed using the matched landmarks, such as features of walls, doors, and objects that appear in the images in consecutive views when the robot navigates. The proposed visual SLAM is based on the improved matched features in consecutive views of the RGB frames. If a landmark is created, some unique scene features are added to the landmark database for later recognition. For each matched landmark in consecutive frames, pose vector $LM = [row,col.,scale,orien,disp,x,y,z]$ is obtained, where (row,col) denotes the calculated image coordinates in the reference camera, $(scale,orien,disp)$ are the scale, orientation, and disparity associated with each stable landmark, and (x, y, z) are the 3D coordinates of the landmark with respect to the camera position. In the proposed method, the goal is to estimate the motion of the camera solely from the visual RGB image information. At each time instance t, the camera provides an RGB-D frame and a corresponding depth map. Fig. 5 illustrates motion estimation using the visual information.

To build a map, visual odometry is used to estimate robot motion in consecutive frames, and is used to calculate the approximate movement (p, q) in X and Z directions, as well as to estimate orientation (θ). Let (x, y, z) denote the 3D coordinates of a matched landmark obtained with the proposed SOM feature matching method, and the new 3D coordinates (x', y', z') can be obtained using the following notation:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} (x-p)\cos\theta-(z-q)\sin\theta \\ y \\ (x-p)\sin\theta + (z-q)\cos\theta \end{bmatrix} \tag{5}$$

The camera calibration model discussed in Section 2 is used to project the 3D position to image coordinates by the following notation:

$$\begin{bmatrix} row' \\ col' \\ \mathrm{disp}' \end{bmatrix} = \begin{bmatrix} v_0-fy'\Big/z' \\ u_0+fx'\Big/z' \\ fI\Big/z' \end{bmatrix} \tag{6}$$

where (u_0, v_0) are the image center coordinates, I is the distance to the landmark, and f is the focal length. The new value for scale is given by the following equation:

$$s' = \mathrm{scale}^{*}z\Big/z' \tag{7}$$

where s' is the scale value that is inversely related to the distance of the landmark, while the orientation of the expected landmark remains unchanged. For each matched landmark in the consecutive views, the relative pose is determined using the 3D information and correspondences in the different views in the RGB frames. This results in robust and stable landmark

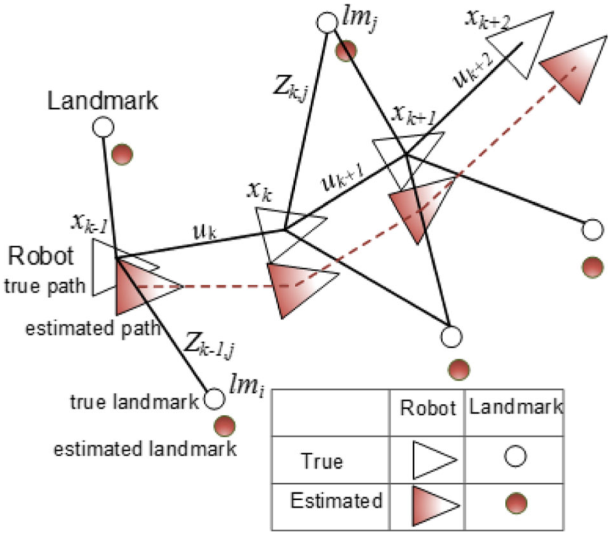


Fig. 5 The robot is represented by the triangle, and the circles represent landmarks in the RGB frames. The robot measures the path using the matched landmarks in consecutive views in the RGB frames. The initial locations of the landmarks are measured using the sensor information

recognition in different views over different time instances. A 3D point is associated with each landmark in the landmark database, and 2D to 3D feature correspondences are used to estimate the robot pose over a span of time. Each new RGB frame comprises 10 to 50 stable landmarks in the newly observed frame, which are later used for pose estimation. The accuracy of the estimated pose depends on the newly observed view, and is denoted as an accurate estimation if the newly detected landmarks lie within a radius of 1 m with respect to the previously observed landmark.

Let Φ denote the map, and let N denote the number of detected landmarks. Each reference to a landmark in the map is denoted by Φ_n. The pose of the robot is denoted by s_t and is denoted with a 3 × 1 vector $[x_t, y_t, \theta_t]T$, where x_t, y_t is the robot location at discrete time t, and θ_t is the heading at a particular time instance, t. The motion model is a state space model to implement visual SLAM (vSLAM) and is given by the following notation:

$$s_{t+1} = f(s_t, u_t) + w_t \tag{8}$$

where u_t is the visual odometry obtained in the time instance between t and $t + 1$; w_t is the error that occurred due to noise. The motion model depends on the odometry information and the kinematics information, along with the floor surface. The motion model used for vSLAM is denoted by the following equations:

$$\begin{aligned} &E(w_t) = 0 \\ &E\left(w_t {w_t}^T\right) = g(\theta_t, u_t, \sigma_T, \sigma_R) \end{aligned} \tag{9}$$

σT and σR are the change in translational and rotational odometry. Another measurement model is given by the following equation:

$$y_t = h(s_t, \varnothing_{nt}) + v_t \tag{10}$$

where $\varnothing_{nt}$ denotes the observed landmark at time instance t, and v_t is noise error with a change in odometry. The error-free measurement with a change in the visual information is given by $h(s_t,\varphi_{nt})$, $E(v_t) = 0$, and $E(v_t {v_t}^T) = \sum v^{vis}$. The robot path is denoted by the sequence

$s_t = s_1,\mathrm{s}_2,\ldots s_t$, which is a sequence estimated from time 1 to time t. The s_t vector represents the sequence of robot poses. Pose is determined using the estimation of s_{t-1}, measurements u_{t-1} and y_t, and landmark poses φ_{nt}. The landmark pose φ_{nt} is estimated at time t, and s_t and Φ are determined using the probability distribution $p(st,\Phi|nt,yt,ut)$. The assumption is shown using Bayesian calculus and probabilistic robotics:

$$p\left(s^t,\varnothing|n^t,y^{t-1},u^t\right) = p\left(s^t|s^{t-1},u^t\right).p\left(s^{t-1},\varnothing|n^{t-1},y^{t-1},u^{t-1}\right) \tag{11}$$

For the visual SLAM algorithm, the following factorization is performed to estimate the $N + 1$ poses and their cross-correlations:

$$p(s^t,\varnothing|n^t,y^t,u^t) = p\left(s^t|n^t,y^t,u^t\right).\prod_{i=1}^{N} p(\varnothing_i|s^t,n^t,y^t) \tag{12}$$

The landmark database is updated, along with updating the robot path, and then landmark pose estimation is performed. The Kalman filter is used to update the landmark pose, and each landmark is associated with a Kalman filter, which is used to update the path hypothesis regularly. For instance, if there are 100 particles and 20 landmarks, the number of Kalman filters is 2000. The Kalman filters are updated when there is a change in the state of the system.

4 Experimental results and discussion

Experiments were conducted in 12 different types of environment in both simple and complex indoor scenarios. Snapshots of the environments with detected landmarks in consecutive frames at time intervals of 30 s are shown in Fig. 6. The Kinect sensor was used for acquisition of the datasets under the various conditions. The resolution of the RGB frames for all 12 datasets was 640 × 480. The frame rate was 30 fps, and the tilt angle was 30 degrees for acquisition of the rotation and illumination datasets. The details of the six simple datasets used for the experiments are given below.

Bedroom (Fig. 6a): the room has a length of approximately 4 m with landmarks, including a bed, an almirah, and a study table. The floor was flat and coated with a wooden floor sheet.

Bedroom (rotation and illumination) (Fig. 6b): the room has a length of approximately 4 m and landmarks, including a bed, an almirah, and a study table. In this dataset, there is a change in illumination, and the rotation angle was set to 30 degrees.

Dining Room (Fig. 6c): the room has a length of approximately 6 m and landmarks, including a dining table, chairs, and obstacles in the path. In this dataset, there is no change in illumination and rotation.

Dining Room (rotation and illumination) (Fig. 6d): the room has a length of approximately 6 m and landmarks, including a dining table, chairs, and obstacles in the path. In this dataset, there is a change in illumination, and the rotation angle was set to 30 degrees.

Living Room (Fig. 6e): the room has a length of approximately 5 m and landmarks, including a sofa, some objects, and chairs. In this dataset, there is no change in illumination and rotation.

Living Room (rotation and illumination) (Fig. 6f): the room has a length of approximately 5 m and landmarks, including a sofa, some objects, and chairs. In this dataset, there is a change in illumination, and the rotation angle was set to 30 degrees.

A set of six experiments for simple scenarios was done under different viewpoint conditions (i.e., different indoor scenarios). In each of the experiments, the landmarks were detected with

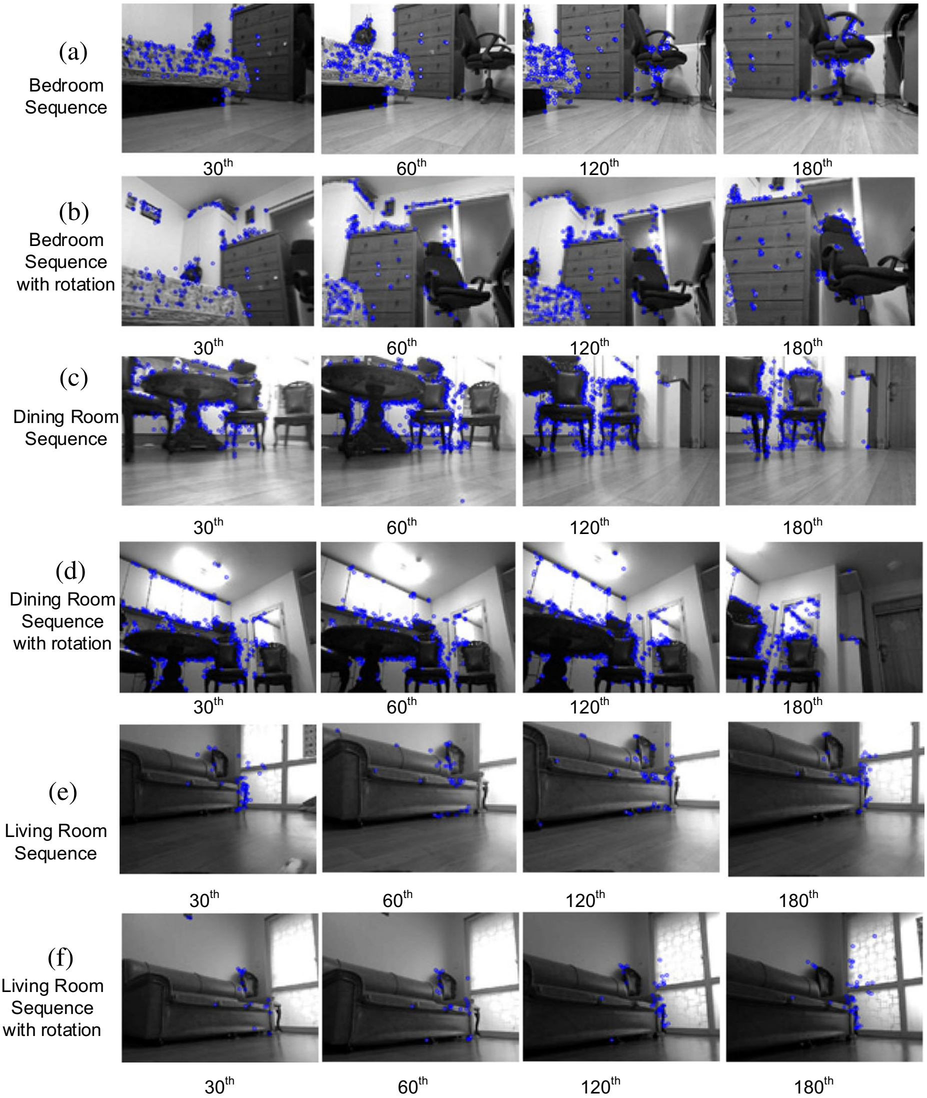


Fig. 6 The results of the viewpoint-invariant landmarks for the different datasets used in the experiments: **a** shows the extracted landmarks with the proposed method in an indoor bedroom with features (bed, almirah, and study table); **b** shows the extracted landmarks with the proposed method in the bedroom sequence with changes in rotation and illumination; **c** to **f** show the set of extracted landmarks for the dining room, the dining room with rotation and illumination, the living room, and the living room with rotation and illumination

the proposed feature matching method. The Kinect sensor was manually controlled to move randomly without a specific style, such as following the corners of the wall. The presented algorithm efficiently extracted the visual landmarks using the winner calculation method. The proposed method is invariant to object loss, scaling, and rotation, and can efficiently replicate an automatic real-time SLAM operation.

Fig. 7 shows the map obtained using the visual landmarks, where the odometry and visual information were plotted onto the map. Referring to Fig. 7, row 1 and row 2, the map was obtained using the detected visual landmarks in the bedroom scenario, and can efficiently localize position using point cloud information. Similarly, row 3, row 4, row 5, and row 6 show the map obtained using the detected visual landmarks in the dining room and living room scenarios. The mapping managed to correct some misalignments due to odometry errors. The lower landmarks underneath the dining table were also detected and mapped during the map building. On the other hand, some misalignment occurred in the map due to the narrow horizontal field of view. The results prove that the proposed method is able to obtain viewpoint-invariant landmarks in less time and can achieve real-time SLAM.

The six complex scenarios were multiple rooms in different conditions and with the appearance of a moving object in front of the camera. The landmarks were detected for the complex scenes to determine an obstacle-free path using the improved stable landmarks. The details of the complex scenes used in the experiments are given below.

Multi-rooms (Fig. 8a)
Multi-rooms (rotation) (Fig. 8b)
Multi-rooms with illumination and darkness (Fig. 8c)
Multi-rooms with illumination and darkness (rotation) (Fig. 8d)
Multi-rooms with appearance of a moving vehicle (Fig. 8e)
Multi-rooms with appearance of a moving vehicle (rotation) (Fig. 8f)

Snapshots of the multiple room environments with detected landmarks in consecutive frames are shown in Fig. 8. A number of experiments were carried out to validate the proposed approach and to evaluate its effectiveness in an indoor scenario. The experiments were performed using RGB-D video sequences where the landmarks are used to detect an obstacle-free path. In order to build the map, the pose of each frame is estimated using visual odometry. Fig. 9 shows the maps obtained for the multi-room scenarios using visual landmarks, where odometry and visual information were plotted onto the map. Optimization was carried out with the selection of optimum landmarks and estimates of the relative pose between two consecutive RGB-D observations. The drift of the proposed method is sufficiently small to determine the actual path trajectory.

The landmark extraction performance was compared with SIFT and the speed-up robust features (SURF) matching method, and it is concluded that the proposed method generated almost double the number of features, in comparison with SIFT and SURF. The proposed method incorporates the winner calculation technique for increasing the number of features, whereas some generated features were rejected in the SIFT and SURF methods during the orientation stage.

For each RGB-D sequence, the success and failure rates were recorded with different trajectories, together with the average length of the path covered by the sensor. The recognition rates in the performed experiments for the proposed method are given in Table 3. Recognition

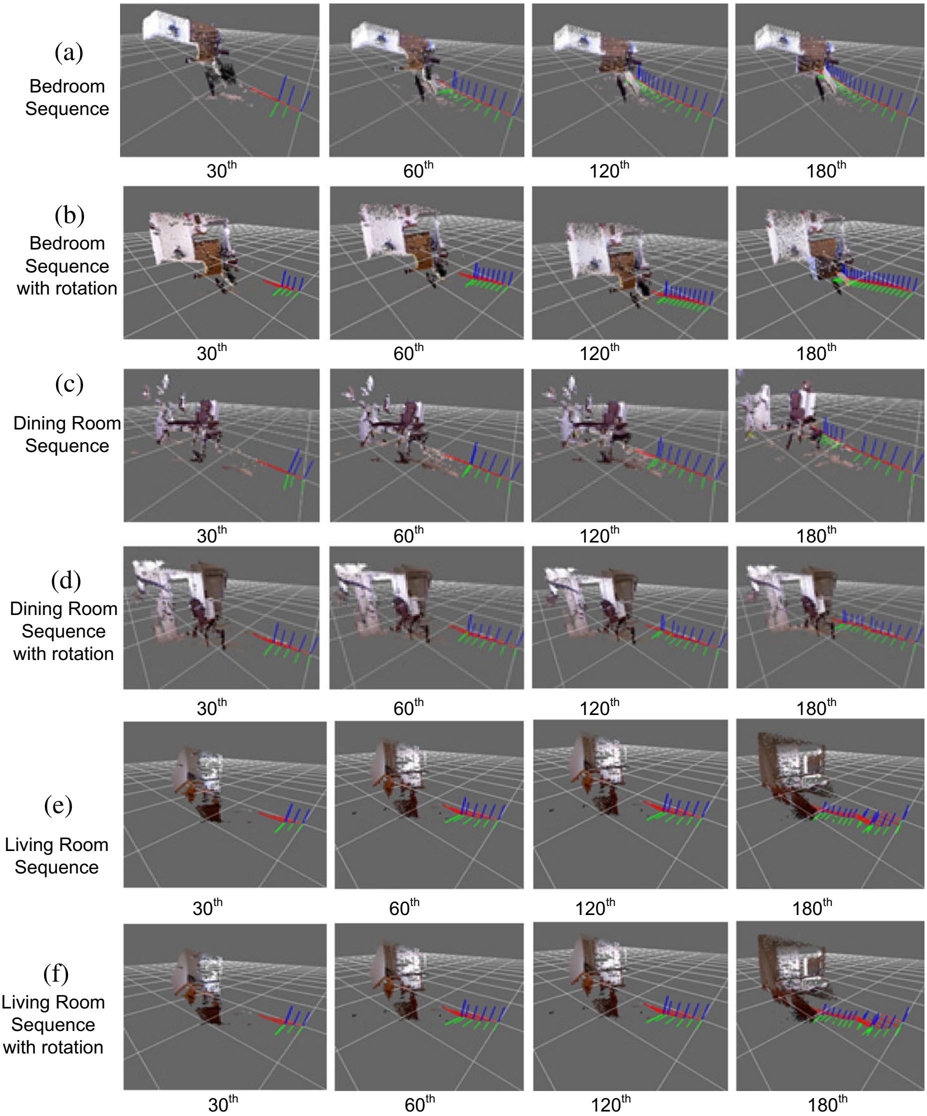


Fig. 7 SLAM output using visual extracted landmarks for the bedroom, bedroom with rotation and illumination, dining room, dining room with rotation and illumination, living room, and living room with viewpoint change and illumination

rate indicates the percentage of correct place recognition, whereas the failure rate indicates error in recognition of the place. The average length of the paths during navigation ranged between 4 m and 6 m. The average number of detected stable landmarks ranged between 50 and 400 during navigation. Nevertheless, the length of exploration is not directly related to the recognition rate, since even scenes with few distinctive landmarks and rooms with no landmarks can eventually be matched. An interesting feature of the proposed approach is that it can easily recognize places where there is little appearance information, a change in viewpoint, and object loss. The average computation time of the

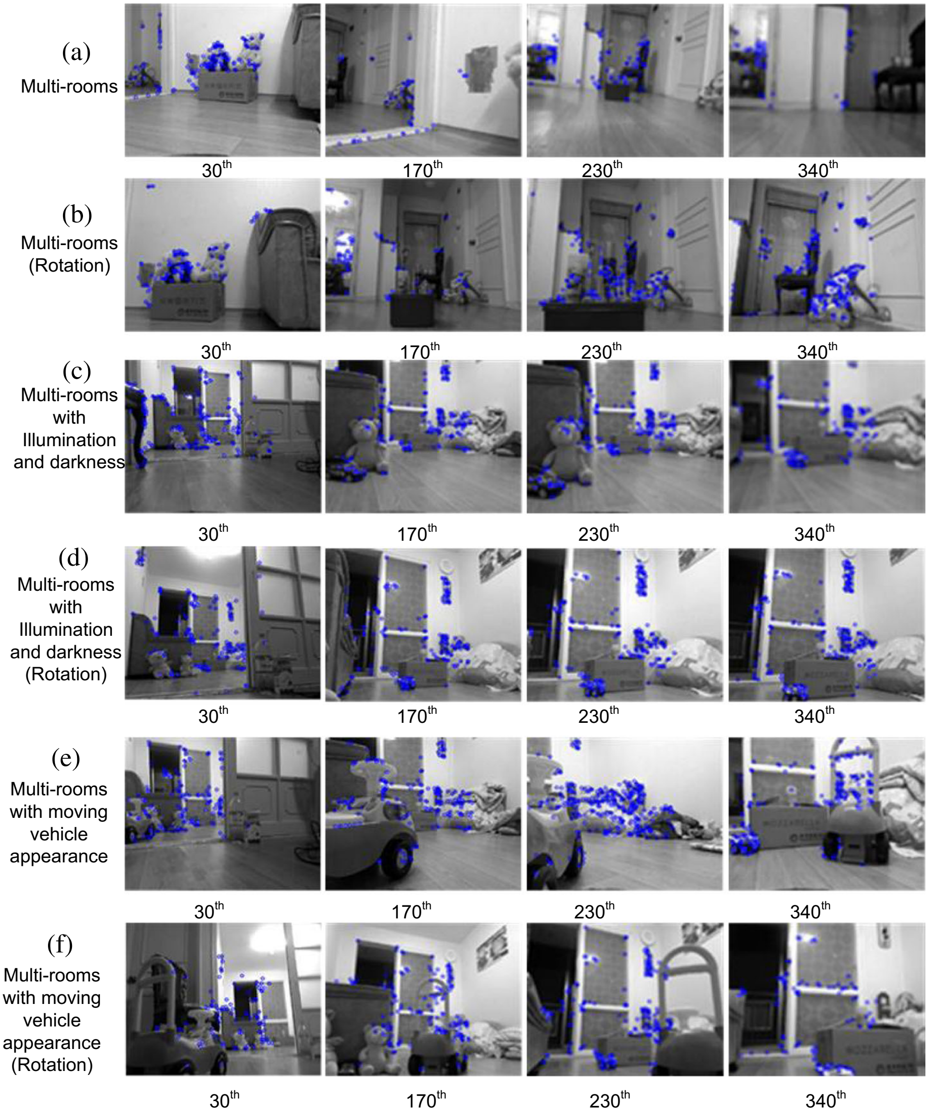

(a)
Multi-rooms
(b)
Multi-rooms (Rotation)
(c)
Multi-rooms with Illumination and darkness
(d)
Multi-rooms with Illumination and darkness (Rotation)
(e)
Multi-rooms with moving vehicle appearance
(f)
Multi-rooms with moving vehicle appearance (Rotation)
30th
170th
230th
340th

Fig. 8 The results of the viewpoint-invariant landmarks for the different datasets used in the experiments: **a** shows the extracted landmarks with the proposed method in multi-rooms with features (i.e., objects placed in different rooms); **b** shows the extracted landmarks with the proposed method in the multi-rooms sequence with a change in rotation; **c** to **f** show the set of extracted landmarks for multi-rooms with illumination and darkness, multi-rooms with illumination, darkness, and rotation, and multi-rooms with a vehicle appearance and for vehicle appearance with rotation

proposed approach is 0.05833 s, which is far less in comparison to the recent SIFT and SURF matching techniques.

In the proposed approach, the time cost is reduced compared to SIFT and SURF. The proposed method can be used for real-time processing with the proposed approach, and the comparative results are given in Table 3 for the proposed method. When using the SIFT and SURF methods, the computation time increases for feature detection and matching of features

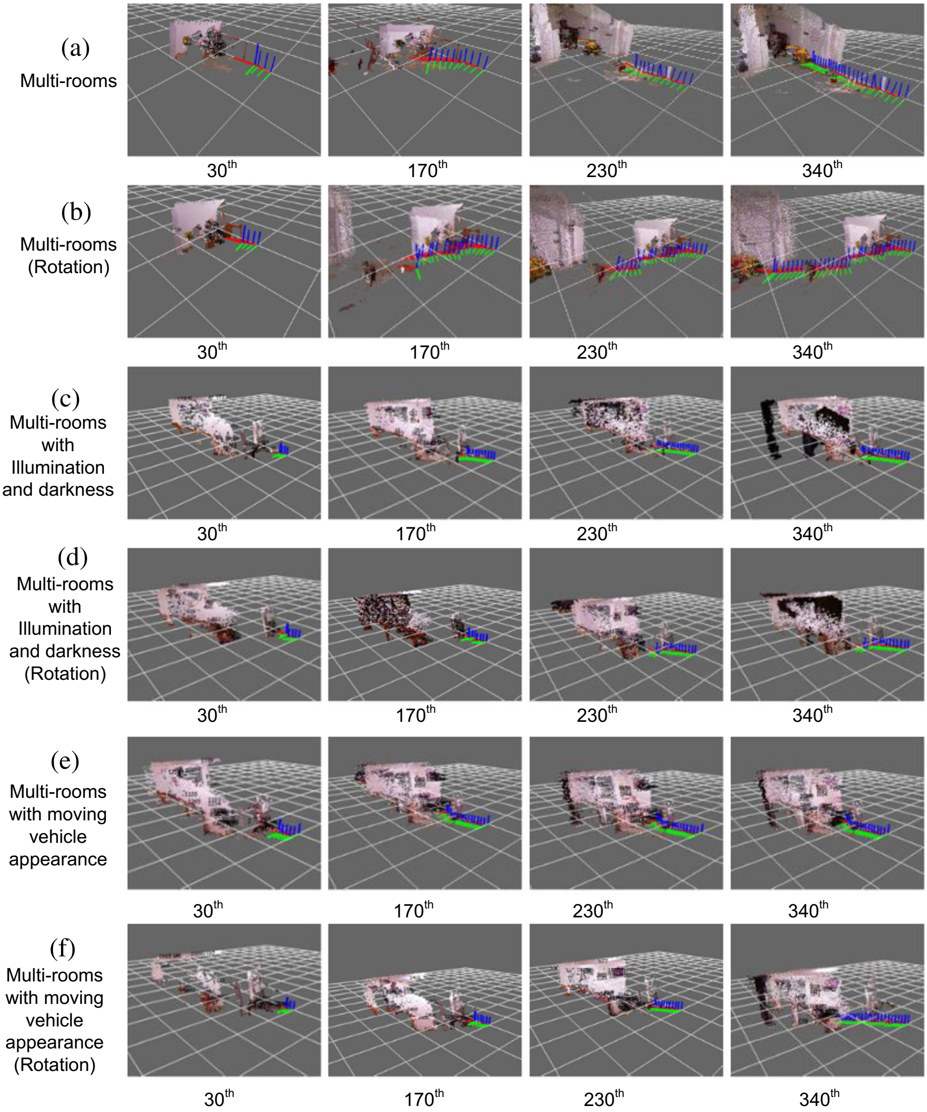

(a)
Multi-rooms
30th
170th
230th
340th
(b)
Multi-rooms (Rotation)
30th
170th
230th
340th
(c)
Multi-rooms with Illumination and darkness
30th
170th
230th
340th
(d)
Multi-rooms with Illumination and darkness (Rotation)
30th
170th
230th
340th
(e)
Multi-rooms with moving vehicle appearance
30th
170th
230th
340th
(f)
Multi-rooms with moving vehicle appearance (Rotation)
30th
170th
230th
340th

Fig. 9 SLAM output for complex videos using visually extracted landmarks for multi-rooms, the multi-rooms sequence with rotation, multi-rooms with illumination and darkness, multi-rooms with illumination, darkness, and rotation, and multi-rooms with a vehicle appearance, and vehicle appearance with rotation

in consecutive frames, compared to the proposed approach. The proposed approach is advantageous when meeting real-time processing and detecting stable landmarks, requiring less time in comparison with SIFT and SURF. The computation costs are increased in SIFT and SURF, but in the proposed method, the size of a descriptor vector is reduced, which reduces the overall computation time.

For evaluation, the proposed SLAM algorithm was compared with the recent state-of-the-art visual SLAM approaches, namely multi-resolution surfel maps (MRS-Map) [37], the RGB-D SLAM system [6, 7], and the point cloud library (PCL) implementation of KinectFusion

Table 3 Effectiveness of the proposed method in different indoor environments with different viewpoint changes, including rotation and illumination

S. No.	Indoor scenario	Frame Num-ber	Recogni-tion Rate for Proposed Method	Fail-ure Rate for Pro-posed Me-thod	Aver-age Path Length	Number of Detected Land-marks (Pro-posed Method)	Average Computa-tion Time for Landmark Detection (sec)-SIFT	Average Computa-tion Time for Landmark Detection (sec)-SURF	Average Computa-tion Time for Landmark Detection (sec)-Proposed Method
1.	Bedroom	30th	100%	0%	4 m	510	8.129044	4.721295	0.068540
		60th	100%	0%	4 m	199	8.201357	5.078524	0.031897
		90th	100%	0%	4 m	625	8.429638	4.272486	0.109481
		120th	100%	0%	4 m	288	5.564516	3.365378	0.056440
2.	Bedroom (Rotation-Illumination)	30th	100%	0%	4 m	731	8.025331	2.920260	0.137459
		60th	100%	0%	4 m	294	7.043448	3.603692	0.108662
		90th	100%	0%	4 m	215	8.160434	3.472364	0.032916
		120th	100%	0%	4 m	197	7.378901	3.938698	0.030392
3.	Dining Room	30th	95%	5%	6 m	646	8.025331	3.408349	0.115905
		60th	95%	5%	6 m	567	7.043448	3.875540	0.156598
		90th	95%	5%	6 m	376	5.379830	4.177699	0.087196
		120th	95%	5%	6 m	496	5.539445	4.299545	0.114063
4.	Dining Room (Rotation--Illumination)	30th	90%	10%	6 m	242	6.075970	2.922796	0.036609
		60th	90%	10%	6 m	346	5.874661	2.942832	0.067722
		90th	90%	10%	6 m	305	4.535895	4.411003	0.052156
		120th	90%	10%	6 m	274	4.934601	3.133584	0.047164
5.	Living Room	30th	90%	10%	5 m	73	4.600187	4.471121	0.042772
		60th	90%	10%	5 m	63	4.505334	4.138419	0.007699
		90th	90%	10%	5 m	73	4.363124	3.982278	0.008349
		120th	90%	10%	5 m	51	4.396687	4.209219	0.006566
6.	Living Room (Rotation-Illumination)	30th	90%	10%	5 m	77	4.572476	4.199959	0.010053
		60th	90%	10%	5 m	56	5.171028	4.754808	0.007046
		90th	90%	10%	5 m	69	5.121288	4.082094	0. 017242
		120th	90%	10%	5 m	43	5.738337	4.261545	0.006085

Table 4 Results for RMSE of absolute trajectory error (ATE) for the proposed improved SLAM in comparison with the recent state-of-the-art approaches. The results are compared with the multi-resolution surfel map [37], RGB-D SLAM [6, 7] and KinectFusion [24] methods. The proposed method performs best, compared to the other three methods

Datasets Used for Evaluation	TFC	RMSE ATE (in meters)			
		Proposed Improved SLAM	Multi-Resolution Surfel Map [37]	RGB-D SLAM [6, 7]	Kinect Fusion [24]
BR	128	0.012	0.023	0.019	0.122
BR (Rot, Illu.)	292	0.043	0.057	0.052	0.211
DR	429	0.038	0.046	0.042	0.168
DR (Rot., Illu.)	257	0.029	0.054	0.046	0.543
LR	239	0.034	0.051	0.039	0.431
LR (Rot., Illu.)	260	0.028	0.031	0.034	0.426
MR	363	0.025	0.064	0.083	0.512
MR (Rot.)	448	0.035	0.043	0.041	0.826
MR (Illu.,Dark)	487	0.039	0.052	0.065	0.924
MR (Illu., Dark, Rot.)	254	0.025	0.039	0.053	0.265
MR(MVA)	311	0.041	0.054	0.072	0.679
MR(MVA,Rot.)	221	0.021	0.039	0.057	0.143

BR Bedroom, *LR* Living Room, *DR* Dining Room, *MR* Multi-rooms, *Rot.* Rotation, *Illu.* llumination, *MVA* Moving vehicle appearance, *TFC* Total Frame count in each sequence

(KinFu) [24]. The two prominent methods given in TUM RGB-D benchmark [39] were used to calculate the absolute trajectory error (ATE) and the relative pose error (RPE). ATE is used to evaluate the error in the estimated trajectory by comparing it with the ground truth. Also, RPE is used for measuring the drift of a visual odometry system (for example, drift per second). The results of the root mean square error (RMSE) of the absolute trajectory error for the proposed improved SLAM in comparison with the recent state-of-art approaches are given

Table 5 Results for RMSE of relative pose error (RPE) for the proposed improved SLAM in comparison with the recent state-of-the-art approaches. The results are compared with the multi-resolution surfel map [37], RGB-D SLAM [6, 7] and KinectFusion [24] methods. The proposed method performs best, compared to the other three methods

Datasets Used for Evaluation	TFC	RMSE RPE (in meters)			
		Proposed Improved SLAM	Multi-Resolution Surfel Map [37]	RGB-D SLAM [6, 7]	Kinect Fusion [24]
BR	128	0.072	0.074	0.087	0.165
BR (Rot, Illu.)	292	0.094	0.104	0.104	0.289
DR	429	0.083	0.093	0.106	0.217
DR (Rot., Illu.)	257	0.052	0.126	0.121	0.874
LR	239	0.044	0.098	0.048	0.902
LR (Rot., Illu.)	260	0.059	0.072	0.064	0.953
MR	363	0.054	0.139	0.201	0.107
MR (Rot.)	448	0.075	0.087	0.217	0.195
MR (Illu.,Dark)	487	0.078	0.094	0.152	0.207
MR (Illu., Dark, Rot.)	254	0.043	0.085	0.132	0.301
MR(MVA)	311	0.092	0.192	0.161	0.927
MR(MVA,Rot.)	221	0.057	0.076	0.147	0.191

BR Bedroom, *LR* Living Room, *DR* Dining Room, *MR* Multi-rooms, *Rot.* Rotation, *Illu.* llumination, *MVA* Moving vehicle appearance, *TFC* Total Frame count in each sequence in Table 4. The first and second column in Table 4 shows the datasets used and the number of frames in each sequence. The average value for RMSE of the absolute trajectory error for the proposed, MRS-Map, RGB-D SLAM, and KinectFusion are 0.030 m, 0.046 m, 0.050, and 0.437 m, respectively. The results of the RMSE of the relative pose error for the proposed improved SLAM in comparison with the recent state-of-art approaches are given in Table 5. It can be seen that the proposed SLAM gave the best results in RMSE values for ATE and RPE.

5 Conclusion

In this paper, a vision-based map-building algorithm using improved, stable landmarks from consecutive frames is proposed. Being scale- and orientation-invariant, the SOM-optimized improved features are good natural visual landmarks for tracking over a long period of time from different viewpoints. Using the proposed methodology, the system is able to build the maps efficiently without keeping correlations between landmarks. The landmarks in the consecutive frames are classified in feature space using SOM clustering, which divides the landmark space into subspaces, and clusters are mapped on a grid. The presented experimental results demonstrated the effectiveness of the proposed approach at recognizing and mapping landmarks in a dataset composed of 12 indoor scenes from a bedroom, a dining room, a living room, multiple rooms, and multiple rooms with an appearance of a vehicle.

References

1. Avidan S (2004) Support vector tracking. IEEE Trans Pattern Anal Mach Intell 26(8):1064–1072
2. Chen S, Li Y (2005) Vision sensor planning for 3-D model acquisition. IEEE Trans Syst Man Cybern B Cybern 35(5):894–904
3. Chiang J, Hsia C, Hsu H (2013) A stereo vision-based self-localization system. IEEE Sensors J 13(5):1677–1689
4. Claraco JLB (2010) Development of scientific applications with the mobile robot programming toolkit (MRPT), machine perception and intelligent robotics. University of Malaga, Laboratory
5. Comport A, Malis E, Rives P (2010) Real-time quadrifocal visual odometry. Int J Robot Res 29(2):245–266
6. Endres F, Hess J, Engelhard N, Sturm J, Cremers D, Burgard W (2012) An evaluation of the RGB-D SLAM system, IEEE Intl. Conf. on Robotics and Automation (ICRA)
7. Engelhard N, Endres F, Hess J, Sturm J, Burgard W (2011) Realtime 3D visual SLAM with a hand-held camera, RGB-D workshop on 3D perception in robotics at the European Robotics Forum (ERF)
8. Gedik O, Alatan A (2013) 3-D rigid body tracking using vision and depth sensors. IEEE Transactions on Cybernetics 43(5):1395–1405
9. Grisetti G, Stachniss C, Burgard W (2007) Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE Trans Robot 23(1):34–46
10. Grisetti G, Stachniss G, Burgard W (2009) Non-linear constraint network optimization for efficient map learning. IEEE Trans Intell Transp Syst 10(3):428–439
11. Henry P, Krainin M, Herbst E et al (2010) RGB-D mapping: using depth cameras for dense 3D modeling of indoor environments. Intl. Symp. on Experimental Robotics (ISER)
12. Jiang C, Paudel D, Fougerolle Y et al (2016) Static-map and dynamic object reconstruction in outdoor scenes using 3-D motion segmentation. IEEE Robotics and Automation Letters 1(1):324–331
13. Kaess M, Ranganathan A, Dellaert F (2008) iSAM: incremental smoothing and mapping. IEEE Trans Rob 24(6):1365–1378
14. Ke Y, Sukthankar R (2004) PCA-SIFT: a more distinctive representation for local image descriptors. Proc. IEEE computer society Conf. Computer vision and pattern recognition. 2: 506–513Washington, DC

15. Koeser K, Bartczak B, Koch R (2007) An analysis-by-synthesis camera tracking approach based on free-form surfaces. German Conf. on Pattern Recognition
16. Konolige K, Bowman J (2009) Towards lifelong visual maps. Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems
17. Konolige K, Agrawal M, Bolles R et al (2007) Outdoor mapping and navigation using stereo vision. Proc. Intl. Symp. on Experimental Robotics (ISER)
18. Lee J, Roh K, Wagner D et al (2011) Robust local feature extraction algorithm with visual cortex for object recognition. Electron Lett 47(19):1075–1076
19. Lowe D (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
20. Magnusson M, Andreasson H, Nüchter A, Lilienthal A (2009) Automatic appearance-based loop detection from 3D laser data using the normal distributions transform. J Field Rob 26(11–12):892–914
21. Manduchi R, Castano A, Talukder A, Matthies L (2005) Obstacle detection and terrain classification for autonomous off-road navigation. Auton Robot 18(1):81–102
22. Mikolajczyk K, Schmid C (2004) Scale & affine invariant interest point detectors. Int J Comput Vis 60(1): 63–86
23. Montemerlo M, Thrun S, Koller D et al (2002) FastSLAM: a factored solution to the simultaneous localization and mapping problem. Proc. of the National Conf. On artificial intelligence (AAAI)
24. Newcombe RA, Izadi S, Hilliges O, Molyneaux D, Kim D, Davison AJ, Kohli P, Shotton J, Hodges S, and Fitzgibbon A (2011) KinectFusion: real-time dense surface mapping and tracking, IEEE Intl. Symp. On mixed and augmented reality (ISMAR)
25. Nistér D (2005) Preemptive ransac for live structure and motion estimation. Mach Vis Appl 16(5):321–329
26. Nüchter A, Lingemann K, Hertzberg J, Surmann H (2007) 6D SLAM–3D mapping outdoor environments: research articles. J Field Rob 24(8–9):699–722
27. Park J, Kim Y (2015) Collision avoidance for quadrotor using stereo vision depth maps. IEEE Trans Aerosp Electron Syst 51(4):3226–3241
28. Pollefeys M, Gool L (2002) From images to 3D models Commun. ACM 45(7):50–55
29. Schmid C, Mohr R (1997) Local grayvalue invariants for image retrieval. IEEE Trans Pattern Anal Mach Intell 19(5):530–534
30. Se S, Lowe D, Little J (2002) Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. Int J Robot Res 21(8):735–758
31. Segal A, Haehnel D, Thrun S (2009) Generalized-icp. Robotics: Science and Systems (RSS)
32. Shao L, Han J, Xu D et al (2013) Computer vision for RGB-D sensors: Kinect and its applications. IEEE Transactions on Cybernetics 43(5):1314–1317
33. Sharma K, Moon I (2013) Improved scale-invariant feature transform feature-matching technique-based object tracking in video sequences via a neural network and kinect sensor. J Electron Imaging 22(3): 033017–033017
34. Sharma K, Moon I, Kim S (2012) Extraction of visual landmarks using improved feature matching technique for stereo vision applications. IETE Tech Rev 29(6):473–481
35. Sharma K, Moon I, Kim S (2012) Depth estimation of features in video frames with improved feature matching technique using kinect sensor. Opt Eng 51(10): 107002(1–11).
36. Strasdat H, Montiel J, Davison A (2010) Scale drift-aware large scale monocular SLAM. Proc. of Robotics: Science and Systems (RSS)
37. Stuckler J, Behnke S (2012) Integrating depth and color cues for dense multi-resolution scene mapping using rgb-d cameras, IEEE Intl. Conf. on Multisensor Fusion and Information Integration (MFI)
38. Stühmer J, Gumhold S, Cremers D (2010) Real-time dense geometry from a handheld camera. DAGM Symposium on Pattern Recognition
39. Sturm J, Magnenat S, Engelhard N, Pomerleau F, Colas F, Burgard W, Cremers D, Siegwart R (2011) Towards a benchmark for RGB-D SLAM evaluation, Proc. of the RGB-D workshop on advanced reasoning with depth cameras at robotics: science and systems Conf. (RSS)
40. Vesanto J, Alhoniemi E (2000) Clustering of the self-organizing map. IEEE Trans Neural Netw 11(3):586–600
41. Wang J, Zha H, Cipolla R (2006) Coarse-to-fine vision-based localization by indexing scale-invariant features. IEEE Trans Syst Man Cybern B Cybern 36(2):413–422
42. Zamora E, Yu W (2013) Recent advances on simultaneous localization and mapping for mobile robots. IETE Tech Rev 30(6):490–496

Kajal Sharma received a B.E. degree in computer engineering from University of Rajasthan, India in 2005, and M.Tech. and Ph.D. degrees in computer science from Banasthali University, Rajasthan, India in 2007 and 2010. From October 2010 to September 2011, she worked as a postdoctoral researcher at Kongju National University, Korea. Since October 2011 to April 2013, she worked as a postdoctoral researcher at the School of Computer Engineering, Chosun University, Gwangju, Korea. Presently she is working as an independent researcher in Korea. Her research interests include image and video processing, neural networks, computer vision, and robotics. She has published many research papers in various national and international journals and conferences.