IEIE Transactions on Smart Processing and Computing, vol. 5, no. 3, June 2016
http://dx.doi.org/10.5573/IEIESPC.2016.5.3.178

IEIE Transactions on Smart Processing and Computing

Mobile Robot Path Finding Using Invariant Landmarks

Kajal Sharma

Flat No. 301, Building 206, Piorville Apartment, 24 Namyang-Dong, Seongsangu, Changwon, Korea kajal175@gmail.com

Received May 25, 2016; Revised June 21, 2016; Accepted June 29, 2016; Published June 30, 2016

** Regular Paper*

Abstract: This paper proposes a new path-finding scheme using viewpoint-invariant landmarks. The scheme introduces the concept of landmark detection in images captured with a vision sensor attached to a mobile robot, and provides landmark clues to determine a path. Experiment results show that the scheme efficiently detects landmarks with changes in scenes due to the robot's movement. The scheme accurately detects landmarks and reduces the overall landmark computation cost. The robot moves in the room to capture different images. It can efficiently detect landmarks in the room from different viewpoints of each scene. The outcome of the proposed scheme results in accurate and obstacle-free path estimation.

Keywords: Landmark, Path finding, Image processing

1. Introduction

Path finding is an important concept in designing a wide range of applications in the field of robotics, obstacle avoidance and autonomous vehicle–navigation. Although, a number of path-finding techniques for robotic applications have improved accuracy and efficiency when determining a path using sonar and laser sensors, the problem still attracts researchers who want to deal with the problem by using vision sensors. Vision sensors are in great use these days for determining reference points in objects and can be used to detect a robot's position using image information appearing in front of the camera. The image information can also be used to reconstruct a 3D object model. The points of interest are detected in 2D/3D environments, and these keypoints are designated as landmarks to localize the robot path so it can navigate with respect to these landmarks.

This paper proposes a path-finding scheme based upon keypoint detection from scene information. The main contributions proposed in this paper are as follows.

- The proposed scheme improves landmark detection performance in terms of computation time, compared to the recent state-of-the-art schemes, by reducing the feature dimensions of the feature vectors.
- The method detects invariant landmarks, which are used to estimate and localize the position of the mobile robot. It makes it possible to determine an obstacle-free path for the robot.
- Landmark estimation from the proposed scheme guarantees path estimation with stable landmarks, and improves robot navigation with fast computation of a 3D map from different viewpoints.

Section 2 reviews the literature on robot mapping and localization and the importance of vision sensors in path finding and feature-based mapping techniques. Section 3 proposes the path-finding scheme using landmarks detected by matching invariant features from different viewpoint dataset images. An efficiency and accuracy analysis of the proposed scheme is discussed in Section 4, and the resulting outcomes are detailed in this section. The conclusion is presented in Section 5.

2. Related Work

Numerous algorithms have been proposed in the field of path finding, navigation, and simultaneous localization and mapping (SLAM) [1-4]. Most of the techniques rely on laser or sonar sensors, while others are based on recently developed vision sensors [5-7]. SLAM algorithms are classified according to the sensing method used in the application. These sensors are used to gather 2D and 3D information in the scene. Nagataniet et al. [8] used laser range finders to gather 3D data and to obtain depth information, but their algorithm cannot provide color data. In addition, it uses slope sensors and incurs a high cost to implement the data. Another range of sensor-based algorithms has been proposed by researchers to design vision applications, including the monocular camera [7], the stereo vision bumblebee camera [9-11], and the Microsoft Kinect camera [12, 13].

Zhou et al. [14] proposed a 3D SLAM algorithm using an RGB camera and a time-of-flight (TOF) camera. The TOF camera uses an infrared (IR) camera for IR projection and capturing 3D data, but its resolution is low, which provides inaccurate map generation. A trinocular camera was used to capture 3D data with three cameras [15], which is an advancement over the stereo camera for searching pixels in the corresponding different scene images. Another method using the bumblebee stereo camera was proposed [9, 10] to obtain 3D scene information. It obtains pixel correspondence between left and right images, which is used to generate a disparity map. Recently, the Microsoft Kinect sensor has seen high demand due to its motion-sensing capability and high-resolution depth map. Both RGB and depth data can be gathered with the Kinect, which has gesture and audio sensing capabilities [12].

A number of path-finding and navigation algorithms rely on detecting features and points of interest in the environment from changes in the robot's motion. Many of the SLAM algorithms are based on feature extraction using the Harris corner detection method [16]. The algorithms using Harris corners require extensive computation to detect features in beacon-free environments, and cannot detect features with changes in viewpoints in the scenes. Some other approaches have been employed to detect the points of interest in environment images captured during movement of a mobile robot. Shi and Tomasi [17] proposed feature tracking in the images, and this algorithm resulted in a more robust approach, compared to the Harris corner method. Another feature-based algorithm [18], determines invariant-landmark clues with a change in viewpoint, which can be employed in various path-finding and SLAM algorithms. The approach is efficient but has a computation drawback because Gaussian pyramid computation is extensive. Thus, this algorithm cannot be directly incorporated into the design of SLAM algorithms in real time, and a demand arises to determine the points of interest with fewer computations. Comparative observations of all the algorithms used for SLAM and feature-based navigation are detailed in Table 1.

Table 1. Comparative observations for SLAM and feature-based navigation algorithms.

Sensing method and feature-based algorithms	Observations
Laser range finders and slope sensors [8]	3D information is detected by the robot with respect to the horizontal plane; this method is difficult to implement.
Time-of-flight camera [14]	This method has low resolution, and the two cameras are aligned to capture the same object image. The method is based on the time of flight concept for ray projection and perception.
Trinocular stereo camera [15]	The trinocular camera is used to speed the capture process, and it searches the triplets corresponding to 2D segments in the 3 images. It suffers from the alignment problem.
Bumblebee stereo vision camera [9, 10]	The bumblebee stereo camera generates dense 3D maps, but it results in less accurate maps. The cost of the camera is high.
Kinect camera [12, 13]	The camera provides high-resolution RGB and depth images able to recognize human action, gestures, and audio.
Harris corner method [16]	The method detects features in the scene with better location accuracy, and implementation is faster.
Shi and Tomasi [17]	This is a feature-tracking algorithm to detect features in scene images; it is faster and more robust than the Harris corner detection method.
Lowe SLAM algorithm [18]	This viewpoint-invariant feature-detection method obtains robust and stable features. It can be used to track an object in real-time implementation of object-tracking and object-detection applications. It suffers from slow algorithm performance at the corners, and needs more computations.

3. The Proposed Scheme

This section discusses the basic and detailed procedures to determine robot movement, which includes landmark matching, estimation of intrinsic parameters and extrinsic parameters, point cloud generation, path finding and object modeling.

3.1 Landmark Scheme

In the sensor described in this paper, a landmark can be estimated by matching different images. The robot, named KOBOT, moves in a room to sense the environment and capture different images. KOBOT uses the Kinect sensor to capture images and to capture various image datasets (for instance, an illumination scene–change dataset, a rotation scene–change dataset, a blurred and affine scene–change dataset). A schematic of the proposed scheme, depicting the steps employed in this research, is given in Fig. 1.

The operating principle of the proposed scheme is as follows: the scene dataset is sent to the interest-point detection block and the cluster landmark block. To match and detect landmarks in the corresponding images, the points of interest are detected relative to the points of interest in the next image. A series training cycle is employed to cluster the similar points of interest, so that similar features for each image between the current scene and the different viewpoints can be detected. Self-organizing map (SOM) networks are employed to detect the nearest neighbor pixels in each cluster of features. Only one neuron in the output grid will have one neuron. The remaining clusters are mapped to the neuron having similar values. The output SOM map results in reduced

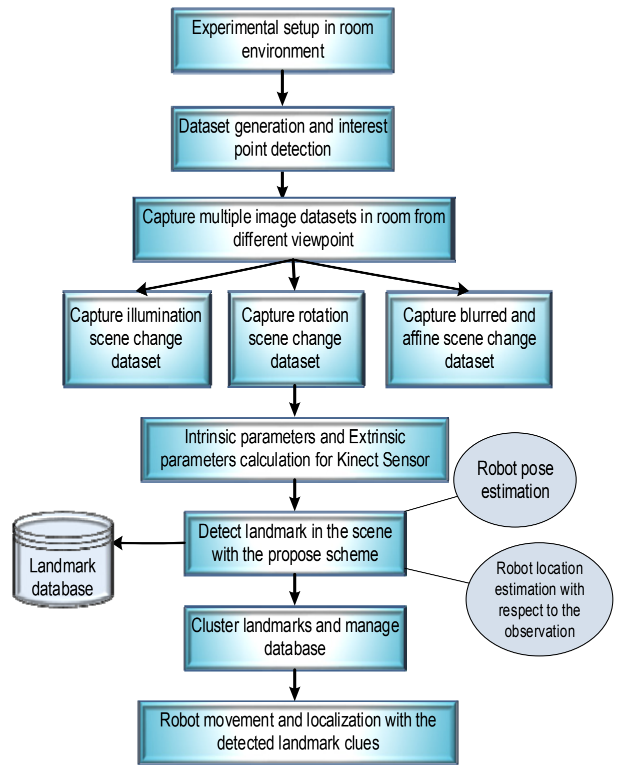

Experimental setup in room environment
Dataset generation and interest point detection
Capture multiple image datasets in room from different viewpoint
Capture illumination scene change dataset
Capture rotation scene change dataset
Capture blurred and affine scene change dataset
Intrinsic parameters and Extrinsic parameters calculation for Kinect Sensor
Robot pose estimation
Landmark database
Detect landmark in the scene with the propose scheme
Robot location estimation with respect to the observation
Cluster landmarks and manage database
Robot movement and localization with the detected landmark clues

Fig. 1. A schematic of the proposed scheme depicting the steps employed in the research work.

landmarks. The intrinsic parameters and extrinsic parameters are calculated and discussed in the next subsection.

3.2 Intrinsic Parameters and Extrinsic Parameters for Kinect Sensor

It is necessary to determine the intrinsic parameters of the depth and RGB camera for 3D map building. For each camera disparity image, the intrinsic parameters are used to form a point cloud. The intrinsic parameters for the calibration process were determined by using the mobile robot programming toolkit (MRPT) [19] library. It was concluded that the intrinsic calibration parameters did not correspond to the disparity image owing to limited bandwidth for the USB connection. The parameters calculated for the experiments are listed in Table 2, which gives a brief overview to better understand the calibration parameters for RGB/depth images from the Kinect sensor. Example images from the calibration process are shown in Fig. 2.

For each object point, the calibration parameters denote the relation between the image measurements (x, y, z') and object coordinates (*X*, *Y*, *Z*). The image size of the disparity scene obtained by chip processing is of reduced size, in comparison with the scene captured using the IR sensor. The size of the disparity scene is 640 × 480 pixels. As can be seen from Table 2, various calibration parameters are calculated for the RGB and depth scenes, for instance, focal lengths (*fx*/*fy*), the optical center (*cx*/*cy*), and distortion parameters.

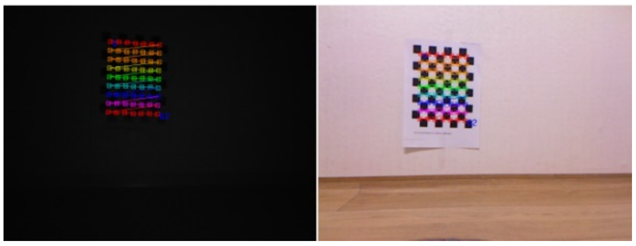


Fig. 2. Example images showing calibration process and the reprojected corners.

Table 2. Intrinsic and extrinsic calibration parameters calculated for the RGB/depth images.

Parameters	Dataset 1		Dataset 2	
	Infrared/ Depth	*RGB*	*Infrared/ Depth*	*RGB*
fx	556.70	542.21	285.48	647.44
fy	533.59	595.10	298.72	655.94
cx	-285.32	413.64	515.45	381.85
cy	163.77	224.60	241.30	75.21

3.3 Point cloud formation

The point clouds received for each room scene with the Kinect sensor have more than 300,000 points. The numbers in the point clouds are of reduced size because the process needs a number of training cycles to iterate. It is preferable to take the front scene image to analyze the point cloud, and then discard a point area that is too high for KOBOT. In addition, the scene images that are at too low a position for the Kinect are also discarded. Each point in the point cloud is represented by a 3D coordinate (*x*, *y*, *z*). A filter is applied to remove the points that have too-high and too-low values, and a threshold is set to keep the points that contribute to the landmark estimation and path determination block.

The algorithm is enhanced by reducing the number of point-cloud points using a voxel grid approach. The voxel grid is designed to reduce the number of points in the point cloud, and results in a point cloud in a 2D point space. The proposed scheme uses a voxel grid of 1 cm x 1 cm x 1 cm. A centroid single point is calculated to replace all the voxel grid points with a single point. The 3D point information is stored in a voxel grid, which is a 3D occupancy grid where cells are marked as occupied, free, or unknown. The MRPT library was used, which consists of two major stages: depth/RGB image conversion to a 3D point cloud, and a point-cloud projection. MRPT provides a CColouredPointsMap function to convert the range scan to a point cloud. The points of the point cloud are copied into a Point Cloud Library (PCL) point cloud.

project3DPointsFromDepthImageInto can be used to directly project the point into the PCL. To do this, MRPT must be precompiled with the PCL before projection of the points. The sequence of steps to compute the 3D point cloud is as follows.

1. The out-of-range points are removed from the

dataset, i.e. the values that are too high or too low are discarded. A threshold is defined to select the points in the output space. The *HeightFilterMin* and *HeightFilterMax* properties are defined by the user to select the 3D point space.
2. Copy the point cloud to a visualization cloud if *PointCloudOut* is set to enable.
3. Use *VoxelGrid* class to downsample the point cloud.
4. The *NormalEstimation* method is used to estimate the normals.

The sensor observations share a common virtual base class, and these classes are used to store laser scanners, 3D range images, monocular and stereo images, GPS data, odometry readings, etc. The class mrpt::slam::Csensory Frame is used as a set of observations that were collected at approximately the same instant. The class mrpt::slam:: CRawlog is used to load, edit, and explore the robotic dataset. The rawlog dataset file is used as input, and is visualized and manipulated by *RawlogViewer*. The rawlog file consists of all the data gathered by the sensor during movement through the environment. The format of the rawlog file is defined as a sequence of *Actions* and *Observations*, where *Actions* include robot motor actuations (odometry) and *Observations* includes readings of laser scanners, and images from cameras and sonar ranges, etc.

3.4 Path-finding algorithm and object modeling

The proposed approach is based on processing the landmarks and the 3D point cloud, and using the resulting information to execute a 2D navigation block. The path-finding block consists of generating a 3D point cloud, which is processed to obtain an obstacle-free path. The robot moves in the room using the invariant landmarks in scenes to reach the goal. The best path is determined via minimum-distance calculation from the robot position to the target, which resulted in an obstacle-free shortest path.

A probabilistic model for the collection of features with flexible appearance is used, which consists of a set of features of a scene. For two classes, ω_1 and ω_2, a Bayesian decision is defined, which is used to set the threshold value for path finding:

$$X^* = \arg\max_{x=1,2} p(w_x|F) = \arg\max_{x=1,2} p(F|w_x)p(w_x) \quad (1)$$

where X is the feature of the front scene captured from the camera. A hypothesis is called for the detected matched features of a scene with different viewpoints. The likelihood term in Eq. (1) can be expanded, as seen below:

$$p(F|w_x) = \sum_{h\varepsilon H} p(F,h|w_x) = \sum_{h\varepsilon H} p(F|h,w_x)p(h|w_x) \quad (2)$$

where x=1, 2, and H is the set of matched hypotheses. It is assumed that segmenting the object in the scene from the background has no occlusions, and the matching hypothesis is considered many-to-1. In order to overcome the large hypothesis feature space, a threshold-based hypothesis is considered, which can be defined as follows.

For each p^{th} detected feature of the segmented object, and the q^{th} part of the object model, the most probable hypothesis, H*, is determined for each feature. The corresponding mean feature vectors are computed by calculating the least X2 distance to the segmented object features, which are denoted by $\overline{x_p}$ and $\overline{x_q}$. The hypothesis can be defined with the following equation:

$$p(F|w_x) \approx p(F,h^*|w_x) \quad (3)$$

The defined hypothesis condition makes sure that the object does not contain repeating patterns, and correct or false matches can be obtained from positive and negative matches based on the hypothesis for each detected feature of the scene. The differences between the corresponding matched keypoints are computed for the reference region of interest and the next scene captured by the camera. The matched hypothesis pair is selected as the positive feature match if the least X2 distance to the segmented object feature is less than or equal to the threshold value.

4. Simulation Results

In order to determine the effectiveness of the proposed scheme, 100 images were selected for test cases, and points of points were detected. The experiments were performed with images captured by the mobile robot. The mobile robot named KOBOT was fitted with an RGB-D Kinect sensor, and moved through the room environment. Fig. 3 shows a snapshot of RGB and grayscale images captured by KOBOT at a time instance of 30 sec.

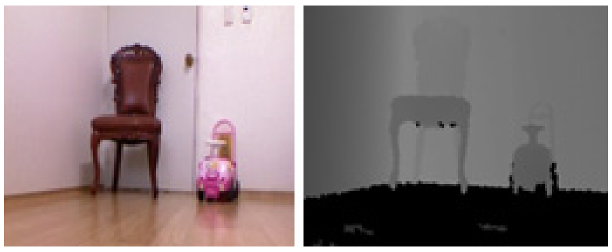


Fig. 3. Snapshot of RGB and grayscale images captured by KOBOT.

A laser source is incorporated in the Kinect device, which splits a single beam into multiple beams that create a constant speckle of patterns projected on the object. This beam is based on the concept of a high-power projector beam, and this speckle pattern is then interpreted by a set of points known as a point cloud. Each point consists of three coordinates, and an example is shown in Fig. 4. The MRPT library is used in the implementation of various tasks that provide access to the Kinect device. The library is used to convert depth and RGB data to 3D point–cloud information. A threshold of 0.5 m was set to estimate object distance. Distances less that 0.5 m and greater than

6 m were discarded to maintain better accuracy in the results. Fig. 5 shows the results of the distances between the camera and the object. A 3D view is shown, which was used as a reference in the landmark detection process.

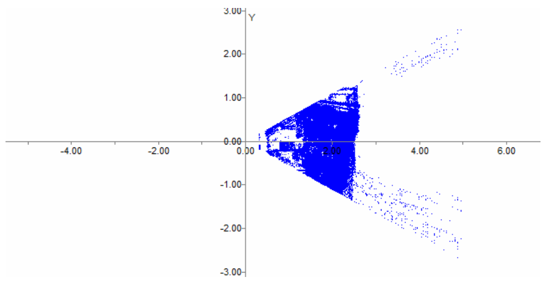


Fig. 4. Speckle pattern shown as a point cloud, and each point is denoted by 3D coordinates.

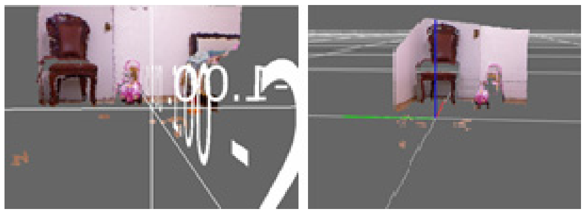
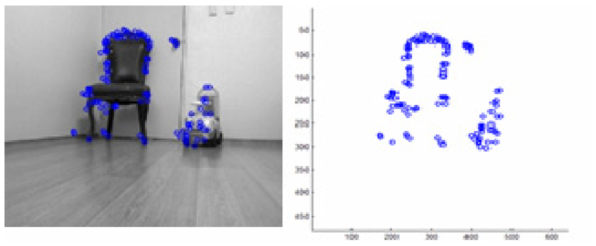
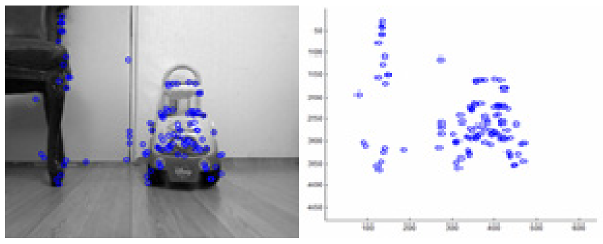


Fig. 5. This 3D view shows the position of the object with respect to the camera. The distance of the object is shown on the white line.

Fig. 6 shows the detection of the landmarks in the room at 30 sec and 50 sec. The locations of the landmarks are stored in the database for further metric evaluation. An average of 50 landmarks were detected in each of the 100 images in the different viewpoint datasets, and an accuracy of 95 % was achieved.



(a) Detected landmarks at 30 sec and the corresponding landmarks



(b) Detected landmarks at 50 sec and the corresponding landmarks.

Fig. 6. Detection of the landmarks at 30 sec and 50 sec, and the landmark position in a 2D plot.

The performance of the herein-presented algorithm was demonstrated with a number of different room images. The proposed scheme was compared with the other recent state-of-the-art methods mentioned in the literature. For the captured room image, the SIFT method was used to calculate the points of interest by determining the difference-of-Gaussian (DOG). These points are invariant to viewpoint change, i.e. the (scale invariant feature transform) SIFT method resulted in rotation- and scale-invariant points of interest. The SIFT method generates a 128-dimensional descriptor vector. This vector is passed to the self-organizing neural network to reduce the dimensions of the detected landmarks. The dimensions of the descriptor are based on the SOM topological network. The best results are obtained for a 7 x 7 rectangular topological neural network. The descriptor vector is mapped onto the SOM network. This reduces the computational overhead for the large-room images database in the path-finding scheme. Each image is passed to the landmark extraction process and passes through the database for further optimization and storage. The proposed method detected landmarks in 0.03245 sec, which is comparatively less than the SIFT method, which detected fewer landmarks in each image in 0.23467 sec. Fig. 7 shows 3D and 2D representations of the point cloud generated in the scene captured during movement of the robot.

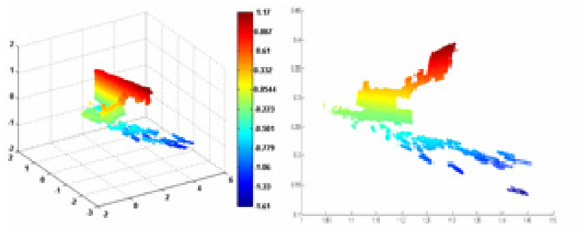


Fig. 7. 3D and 2D point clouds of an image captured during robot movement.

Figs. 8 and 9 offer an idea of how the robot moved during the experiments with the use of landmark clues. The map shown was built using the detected landmarks, and hence, white areas in Fig. 9 may contain many obstacles that are avoided during robot navigation. The obstacles are invisible with a laser sensor, but the Kinect camera has the advantage of detecting these obstacles using the RGB camera, and it can estimate landmarks for optimal path planning. The trajectory shows the areas where the robot did not visit, and set the field of view that was used during the navigation.

Fig. 9 shows the path of the mobile robot from using the similar landmarks. Similar landmarks were detected using SOM landmark matching, which were then plotted onto a 2D graph. These landmarks are rotation- and scale-invariant and are stored in the database to detect the path of the robot. The robot starts at the initial position and then navigates using the landmarks to move in the room. The final path was determined by the robot movement via the landmarks that are detected by matching the room images.

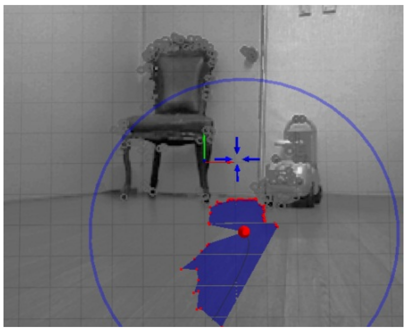


Fig. 8. Robot movement using the landmark clues detected using the proposed method.

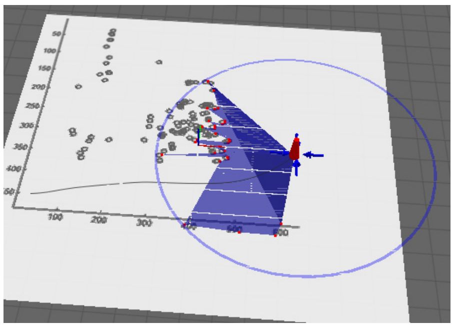


Fig. 9. Example showing holonomic path navigation, where the white space consists of obstacles, and the red color shows the robot's initial position.

5. Conclusion

In this paper, a new path-finding scheme for mobile robots is proposed to determine paths in a room. The algorithm is robust against viewpoint change, and can determine landmarks from room images. By employing a suitable feature-matching detector, the resulting estimates are accurate and precise from different scenes. The detected landmarks are prominent and stable for path analysis during movement of the robot. Furthermore, the available landmarks provide essential clues for many different types of images, particularly with regard to autonomous vehicle navigation, and object detection and localization.

References

[1] M. Kaess, A. Ranganathan, F. Dellaert,"iSAM: Incremental smoothing and mapping," IEEE Trans. on Robotics, vol. 24, no. 6, pp. 1365–1378, 2008. Article (CrossRef Link)

[2] R. Manduchiet. al., "Obstacle detection and terrain classification for autonomous off-road navigation,"Autonomous Robots, vol. 18, no. 1, pp. 81–102, 2005. Article (CrossRef Link)

[3] G. Grisetti, G. Stachniss, andW. Burgard,"Non-linear constraint network optimization for efficient map learning,"IEEE Trans. on Intelligent Transportation systems, vol. 10, no. 3, pp. 428–439, 2009. Article (CrossRef Link)

[4] K. Sharma, I. Moon, andS. Kim, "Extraction of visual landmarks using improved feature matching technique for stereo vision applications,"IETE Technical Review, vol. 29, no. 6, pp. 473-481, 2012. Article (CrossRef Link)

[5] J. St¨uhmer,S. Gumhold, andD. Cremers,"Real-time dense geometry from a handheld camera," Proc. DAGM Symposium on Pattern Recognition, 2010. Article (CrossRef Link)

[6] D. Nist´er,"Preemptive ransac for live structure and motion estimation," Machine Vision and Applications, vol. 16, no. 5, pp. 321–329, 2005. Article (CrossRef Link)

[7] K. Koeser,B. Bartczak, and R. Koch, "An analysis-by-synthesis camera tracking approach based on free-form surfaces," Proc. German Conf. on Pattern Recognition, pp. 121–131, 2007. Article (CrossRef Link)

[8] K. Nagataniet. al., "Three-dimensional localization and mapping for mobile robot in disaster environments," Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3,pp. 3112-3117, 2003. Article (CrossRef Link)

[9] K. Sharma, S. Kim, and M. Singh, "An improved feature matching technique for stereo vision applications with the use of self-organizing map," Int. J. of Prec. Eng. and Manufac., vol. 13, no. 8, pp. 1359-1368, 2012. Article (CrossRef Link)

[10] L. Paz et. al. "Large-scale 6-DOF SLAM with stereo-in-hand,"IEEE Trans. Robotics,vol. 24, no. 5, pp. 946–957, 2008. Article (CrossRef Link)

[11] A. Comport, E. Malis, and P. Rives, "Real-time quadrifocal visual odometry," Int. J Robotics Research, vol. 29, no. (2-3), pp. 245–266, 2010. Article (CrossRef Link)

[12] P. Henry et. al., "RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments,"Proc. Intl. Symp. on Experimental Robotics (ISER), 2010. Article (CrossRef Link)

[13] P. Henry et. al., "RGB-D mapping: Using Kinect-style depth cameras for dense 3D modelling of indoor environments," TheInt. J. of Robotics Research, vol. 31, no. 5, pp. 647-663, 2012. Article (CrossRef Link)

[14] W. Zhou, J. MirÓ, and G. Dissanayake, "Information-efficient 3-D visual SLAM for unstructured domains," IEEE Trans. on Robotics vol. 24, no. 5, pp. 1078–1087, 2008. Article (CrossRef Link)

[15] D. Marzorati, M. Matteucci, and D.Sorrenti, "Particle-based sensor modeling for 3D-vision SLAM,"Proc. IEEE Int. Conf. on Robotics and Automation, Roma, pp. 4801-4806, 2007. Article (CrossRef Link)

[16] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," IEEE Trans. on Pattern Analysis and MachineIntelligence, vol. 19, no. 5, pp. 530-534, 1997. Article (CrossRef Link)

[17] J. Shi and C. Tomasi, "Good Features to Track," Proc. of the 9th IEEE Conference on Computer Vision and PatternRecognition, pp. 593-600, 1994. Article

(CrossRef Link)

[18] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110 (2004). Article (CrossRef Link)

[19] J. L. B. Claraco, "Development of Scientific Applications with the Mobile Robot Programming Toolkit (MRPT)," Machine Perception and Intelligent Robotics Laboratory, University of Malaga, 2010. Article (CrossRef Link)

Kajal Sharma received a BEng in computer engineering from the University of Rajasthan, India, in 2005, and an MTech and PhD in computer science from Banasthali University, Rajasthan, India, in 2007 and 2010, respectively. From October 2010 to September 2011, she worked as a post-doctoral researcher at Kongju National University, Korea. From October 2011 to April 2013, she worked as a post-doctoral researcher at the School of Computer Engineering, Chosun University, Gwangju, Korea. Her research interests include image and video processing, neural networks, computer vision, and robotics. She has published many research papers for various national and international journals and conferences.