Robot Localization based on Navigation Methods of the Brain

Aditya Mohan
2017A7PS0945G

Akhil Tarikere
2017A7PS1916G

Pranav Guruprasad
2017A7PS1918G

Rishikesh Vanarse
2017A7PS1913G

Srisreyas Sundaresan
2017A7PS0065G

Ved Sirdeshmukh
2017AAPS1934G

Abstract:

An important task in robotics is localization, i.e., knowing the position and orientation of the robot with respect to the environment. Visual odometry proves to be a promising approach to perform the task of localization. Vision-based navigation methods are similar to methods used by the brain for navigation. In this paper, we propose a new method for visual odometry, based on navigation in humans. In this method, a stereo depth camera is used to obtain an RGB image of the surroundings and the corresponding depth map. Using object detection, certain 'objects of interest' are tracked in the live stream. The depth map is used to estimate a 3D position of these objects with respect to the robot. Change in this relative vector in consecutive frames is used to estimate the global change in the position of the robot.

1. Introduction:

In humans as well as all complex animals, navigation is a fundamental task. The cognitive map[3] that lies in the hippocampus aids in navigation and localization in humans. The process mainly involves two steps:
1) Construction of the cognitive Map [4]
2) Matching current visual cues to localize self in the map [5]
SLAM (Simultaneous Localization and Mapping) [17] is a method of great research in robotics,

which involves continuously building/updating a map of the environment and simultaneously localizing the robot within it. It is evident that this method significantly resembles navigation by the brain.
During the process, humans are able to perceive depth through stereoscopic vision. Due to the physical separation between the eyes, slightly different images are generated on the retina of the left and right eyes, and the brain superposes them to interpret depth[15].

Attempts in studying spatial navigation in animals [10][8] have lead to some useful insights on cognitive maps and localization. However, these insights have not been effectively implemented on robots due to requirement of high computing and lack of sufficient knowledge regarding the semantics of human navigation. The former often proves to be a bottleneck to research on brain-based navigation methods in robots. However, advances in computing capabilities of GPUs and TPUs may reduce the impracticality of these methods. It is therefore required to study the advantages and limitations of such localization methods.

Landmarks used in existing SLAM methods are usually physical features of the environment [26][27]. Humans, however, use surrounding ‘objects’ as the points of reference for the same task. The proposed method involves usage of stationary objects as landmarks for reference. A depth map (through IR) is used to predict the 3D positions. Motion of the robot is predicted through averaging the relative motion of the detected objects. The paper is organised as follows:
Section 3 discusses the existing research in the field. Section 4 discusses the proposed methodology, along with the hardware/software used. Section 5 describes the experiments performed and provides the corresponding results. Sections 5,6 and 7 discuss the conclusions, future scope.

2. Literature Survey:

2.1 Traditional Approaches in SLAM

Mainstream approaches to Simultaneous Localization and Mapping generally do not use object detection for performing localization. Even when the approach uses Visual odometry [1] , researchers prefer using classical image processing algorithms. The SIFT algorithm (Scale Invariant Feature Transform) [21] is used along with the Harris Corner detection algorithm [22] for landmark detection in most existing SLAM based navigation stacks. This is used in combination with the Christopher Longuet-Higgins’s 8-Point algorithm [24] and the Random Sample Consensus (RANSAC) [23] algorithm.

2.2 Biological Inspiration

There has been a significant amount of work on biologically inspired approaches to SLAM. A 2004 paper by Milford et al. [8] proposed an approach to SLAM called RatSLAM, based on computational models of the rodent hippocampus, which was capable of recovering from path integration errors and ambiguous landmark information in real time on robots. In 2005, Howard et al. [6] proposed a computational framework to simulate the function of the medial temporal lobe (MTL) and the support of the entorhinal cortex in changing temporal context in spatial navigation. Work in neural networks by Fox and Prescott [7] presented a mapping of the hippocampus onto a Temporal Restricted Boltzmann Machine in order to implement a method of noise-free temporal sequence of navigation, with the findings published by IEEE in 2010.

2.3 How humans and other organisms navigate

After decades of research on rodents by numerous researchers, in 2004, Sutherland and Hamilton [10] proposed a descriptive model of rodent navigation that takes into account reference frame, information and movement control. The ‘cognitive map’ hypothesis mentioned in Epstein et al. [28] proposes that the brain builds a unified representation of the spatial environment to support memory and guide future action. Many years of research in rodents suggests that cognitive maps are ‘neurally instantiated by place, grid, border, and head direction cells in the hippocampal formation and related structures’ [28], and that the human brain has a similar functional organization.

3. Proposed Method:

The proposed method consists of the following steps:

1. Obtaining a continuous stream of RGB images and their corresponding depth maps
2. Detecting standard stationary objects in the RGB images and estimating their (x, y, z) position with respect to the camera using the depth map
3. Detecting the same objects in consecutive frames and estimating the change in their relative position
4. Using a weighted mean of the changes in relative vectors to estimate the global change in position of the robot.

3.1 Obtaining RGB-D data:

Initial attempts of obtaining RGB-D data involved the use of Monocular Vision. The depth map was attempted to be estimated using a single-camera stream through deep neural networks. Pre-trained models of Monodepth [11] and SharpNet [12] were used for the same.

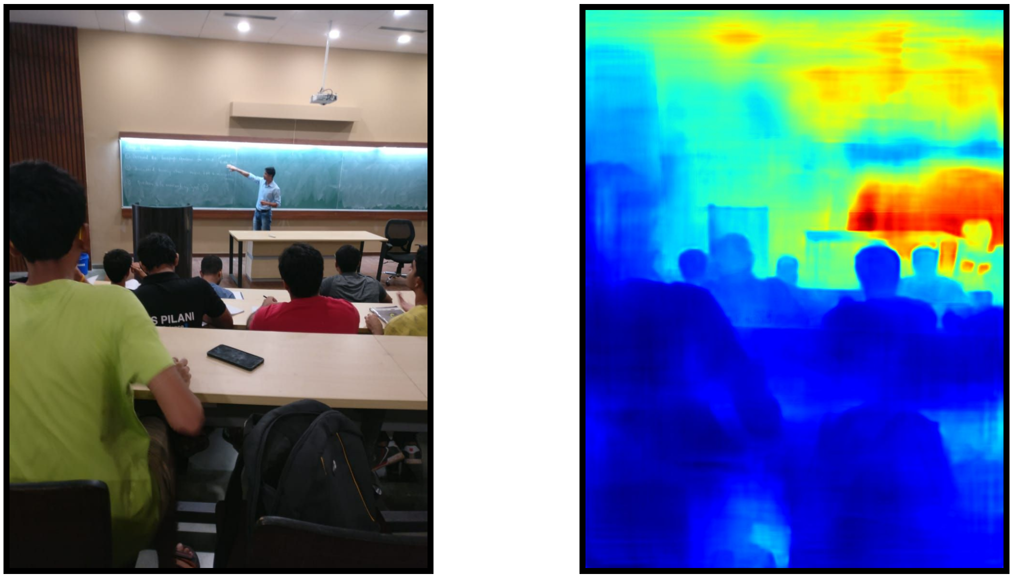

Fig-1: *An RGB image and its corresponding depth map generated by SharpNet*
The hue of the pixels map to the predicted depth

This procedure, however, posed the following drawbacks;
i. High computation time, posing difficulties for real-time implementation
ii. Low accuracy of the results, leading to an unreliable depth map
Iii. High sensitivity to lighting conditions and nature of surroundings, leading to a very limited set of environments where the procedure would work.
The method of obtaining RGB-D data was therefore shifted from monocular camera to Stereo Depth sensors. A Microsoft XBox Kinect 360 v1 was used in the experiments. The live RGB images and the corresponding point cloud was obtained using the ROS package freenect_stack [14] with ROS Kinetic on Ubuntu 16.04.

3.2 Object Detection and Relative position estimation:

In this step, the RGB frames are fed into an object detection model. YOLOv3 [13] was used for this purpose due to its ability to perform fast real-time object detection on general purpose GPUs. As experiments performed were in indoor environments, only specific classes of objects found in such environments were included. The centre of the output bounding boxes are assumed to be single-point approximations of the 2D location of the object. The 2D pixel coordinates are then looked up in the depth map and an (n x n) Gaussian filter around this point estimates the depth (ϱ-coordinate, i.e. radial distance from the camera) of the point (in the spherical coordinate system). In the experiments, the value of n used was 7. When a mean filter was used instead of a gaussian, the kernel size used was 5x5. The step of depth lookup is postponed and carried out after the consecutive frame object matching, in order to improve speed and remove redundant calculations.

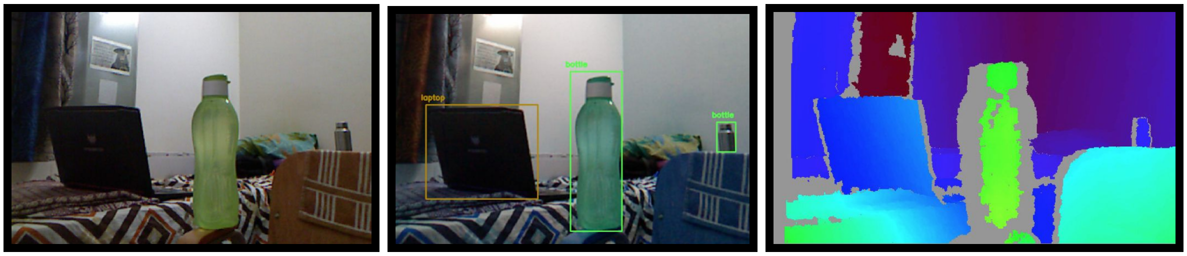

Fig-2: *YOLOv3 detects the objects in the scene and gives out their pixel-coordinates (centre) . The distance of these points from the camera can be extracted from the depth map (right) generated using the disparity between the stereo images.*

3.3 Object Matching in Multiple Frames:

A list of classes (along with their positions) is generated through YOLO for every frame. Pairs of consecutive frames are taken for comparison. Whenever a class is present in both lists, the euclidean distance between its two pixel-positions is calculated. If this distance falls within a predefined threshold, it is assumed that the object in both the frames is the same.

3.4 Global Position Estimation:

The x and y pixel-coordinates of the object are converted to corresponding θ (Angle in the X-Z plane) and Φ (Angle in the Y-Z plane), as follows:

$$\begin{aligned} \Phi &= \frac{(y - h/2)}{h}(\Phi_r) \\ \theta &= \frac{(x - w/2)}{w}(\theta_r) \end{aligned}$$

where,

h is the height of the image in pixels
w is the width of the image in pixels
(x,y) are the pixel coordinates
Φ_r and θ_r are the horizontal and vertical angles of view (AOV) of the stereo depth sensor.

The spherical coordinate vectors of the detected objects are converted to (x, y, z)-vectors as follows:

$$\begin{aligned} x &= \rho \, sin(\theta) \, cos(\phi) \\ y &= \rho \, sin(\theta) \, sin(\phi) \\ z &= \rho \, cos(\theta) \end{aligned}$$

The (x, y, z)-vectors of identical objects from both the frames are then subtracted to get the displacement of the object with respect to the robot.

$$\frac{1}{n}\sum_{j=0}^{n}\left\{[x_{i+1}\ \ y_{i+1}\ \ z_{i+1}]_{(j)} - [x_i\ \ y_i\ \ z_i]_{(j)}\right\}$$

Here, *n* is the total number of objects matched between the two consecutive frames. The process is done for each object detected in consecutive frames, to obtain a set of displacement vectors.

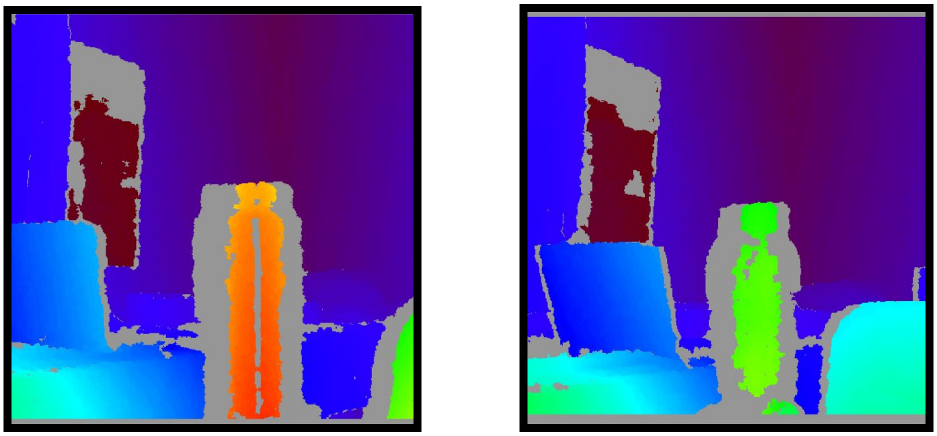

Fig-3: *Disparity depth map of the same scene after moving the bot away from the objects. The image on the right shows a clear change in depth after moving the bot.*

4. Experimental Results:

Three categories of experiments were conducted, as shown in figure ___. In each experiment, the RGB-D values were obtained from the kinect, before and after the movement. These two sets of values act as two consecutive frames of the real-time feed. If the system works for two frames, then inductively, it should work for multiple frames.

The following were the three sets of movements that were experimentally studied.

1. Movement along only Z-axis
2. Movement along X-axis only
3. Rotation about Y-axis

All the experiments had three trials. The first two trials were carried out with a single object in the field of view of the kinect. The final trials of all experiments includes multiple objects of interest.

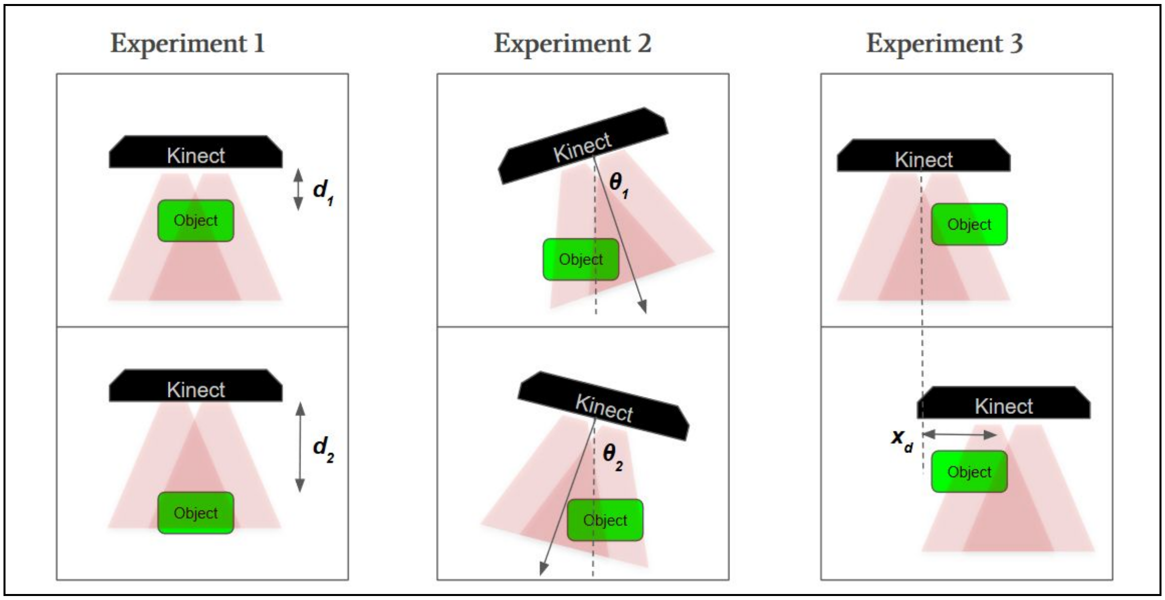

Fig-4: Experimental Setup

In experiment 1, a displacement of (d_2-d_1) is done along the z-axis. In experiment 2, a displacement of (θ_1 + θ_2) is made about the y-axis. In experiment 3, a displacement of x_d is done along the x-axis.

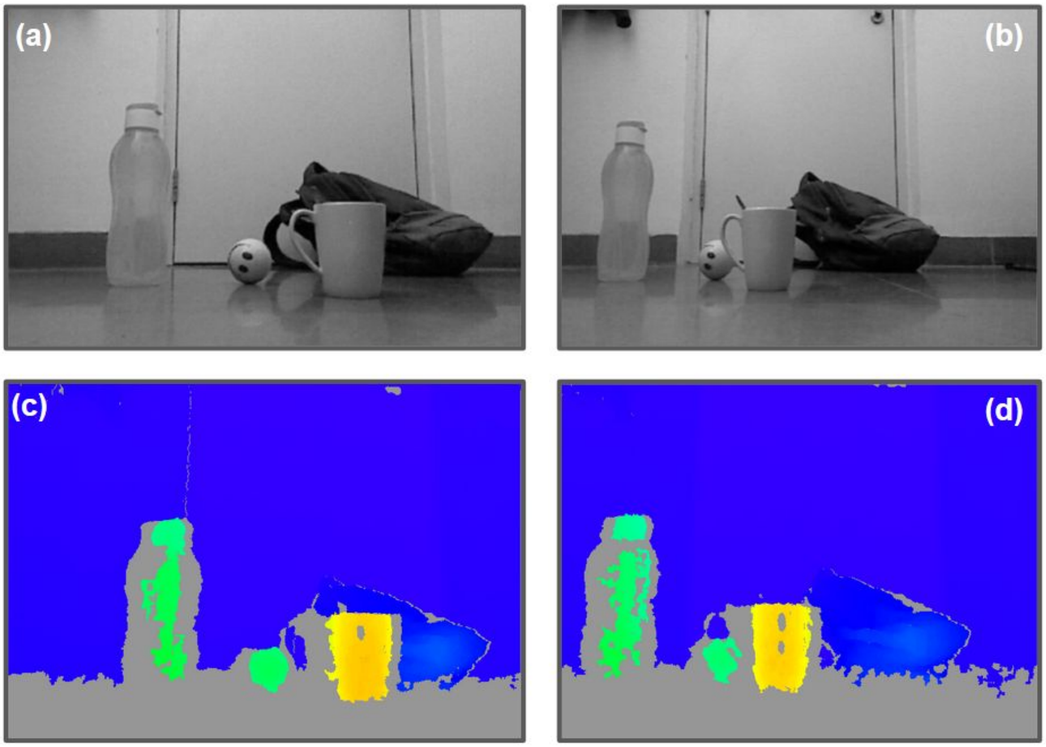

Fig-5: Multiple objects

Fig 5a and 5b show images containing multiple objects, taken from 2 different locations. 5c and 5d show their corresponding depth maps

The known values of the angular and linear displacements of the kinect were compared to the output of the algorithm. The results are shown below:

Table 1: *When a single object of interest was used (Bottle).*

Sr. No.	**Experiment**	**True displacement (x, y, z) in cm**	**Experimentally predicted displacement (x, y, z) in meters**
1	Linear motion along Z-axis only	*(0, 0, 15)*	*(1.035, 0.012, 14.93)*
2	Angular displacement (25°) about Y-axis (yaw) only	*(0, 0, 0)*	*(65.05, 0.088, 1.022)*
3	Linear motion along X-axis only	*(18, 0, 0)*	*(23.69, 0.012, 0.0)*

Table 2: *When multiple objects of interest were used (bottle, cup, sponge-ball & backpack).*

Sr. No.	**Experiment**	**True displacement (x, y, z) in cm**	**Experimentally predicted displacement (x, y, z) in meters**
1	Linear motion along Z-axis only	*(0, 0, 15)*	*(0.28, 0.04, 14.97)*
2	Angular displacement (25°) about Y-axis (yaw) only	*(0, 0, 0)*	*(43.78, 0.043, 0.015)*
3	Linear motion along X-axis only	*(18, 0, 0)*	*(17.3, 0.053, 0.0)*

Through the experimental results, the following general results are evident:

1. The method proves to be very accurate for displacement perpendicular to the Kinect.
2. The method is not suited for Yaw and Pitch. The reason for this is that the final displacement vector calculation involves direct subtraction of the two relative vectors. Although the robot does not change its position in experiment 2, the relative positions of the objects changes with respect to the robot. This causes a significant error in localization.
3. For displacement along X-axis, it is seen that multiple objects of interest are required to obtain an accurate measurement.

Limitations:

The depth map is obtained using the disparity between the two cameras on the Kinect [16]. When objects fall within a certain range (usually less than 60cm) from the kinect, the corresponding points from the two cameras are not matched correctly, leading to effective depth to be calculated as zero. Secondly, although YOLO is built for real-time object detection, it still requires significant computing power. Hence, a sufficiently powerful GPU is required for the software to run in realtime.

One of the major limitations of the method is, however, the inability to localize correctly after angular displacements. The algorithm needs to be modified to include this scenario.

5. Discussions and Future Scope:

Simultaneous Localization and Mapping is generally does not use objects as landmarks due to the computational load of detecting objects. Simple image processing is sufficient to detect the landmarks required for SLAM. However, since humans are experts at localizing themselves with ease, it is essential to study the methods used by humans in doing so. The method described is a naive implementation of the spatial navigation methods of the brain. The novelty lies in the following aspects: firstly, the usage of objects of interest rather than geographical landmarks, and secondly, the fact that the method does not rely on any other means of localization other than visual.

The method proposed has plenty of scope for further development. Firstly, the algorithm can be modified to resolve the flaw with respect to angular movement. Parameters can be optimized to get a more accurate measurement. Closed-loop SLAM can be implemented by introducing a method to store the locations of the objects of interest. Finally, the current method assumes all objects of interest to be stationary, which might not be the case in practice. Hence, algorithms can be further developed to take into account the movement of the objects to make the system more versatile.

6. Conclusion:

In this report we successfully demonstrated the proof of concept of our algorithm through experiments involving two frames. It was shown that object tracking can be an effective method of localization, provided sufficient onboard computation is available. The method proposed was shown to be suitable for linear displacement and requires modification to be extended for angular displacement.

7. References:

1. D. Nister, O.Naroditsky, J.Bergen. 2004. Visual Odometry. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*

2. Simon Frintrop, Patric Jensfelt and Heric Christensen. Simultaneous Robot Localization and Mapping using Visual Attention System

3. Thackery I. Brown (2016, Jun 10). Prospective Representation of Navigational Goals in the Human Hippocampus. *Science Vol 352, Issue 6291,* pp. 1323 - 1326

4. Adam Johnson. 2008. On the Use of Cognitive Maps, *University of Minnesota, 2008*

5. “How does the brain localize itself?”. *,* 2008, Jun 19. *Science*

6. Mark W Howard, Mrigankka. S. Fotedar, Aditya V. Datey. 2005. The Temporal Context Model in Spatial Navigation and Relational Learning: Toward a Common Explanation of Medical Temporal Lobe function across domains. *Psychol Rev. 2005 Jan; 112(1) pp.* 75-116.

7. Charles Fox, Tony Prescot. 2010. Hippocampus as unitary coherent Particle Filter. *2010 International Joint Conference on Neural Networks (IJCNN),* (18-23 July 2010), Barcelona, Spain

8. M.J.Milford, G.F. Wyeth, D. Prasser. 2004. RatSLAM: A Hippocampal Model for Simultaneous Localization and Mapping. *Proceedings of the 2004 IEEE International Conference of Robotics and Automation,* New Orleans, April 2004.

9. L.Zang, L.Wei, P. Shen, W. Wei. 2018. Semantic SLAM based on Object Detection and Improved Octamap. *IEEE Access Vol-6:* pp. 75545-75559

10. Sutherland RJ, Hamilton DA. 2004. Roden Spatial Navigation: At the Crossroads of Cognition and Movement, *in Neuroscience and Biobehavioral Reviews, 2004 Nov 28* pp. 687-97

11. Clement Godard, Osin Mac Aodha, Gabriel J. Ostow. 2017. Unsupervised Monocular Depth Estimation with Left-Right Consistency, *12 Sep 2017*

12. Michael Ramamonjioa, Vincent Lepetit. 2019. SharpNet: Fast and Accurate Recovery of Occluding Contours in Monocular Depth Estimation, *International Conference of Computer Vision (ICCV) Workshops*

13. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. 2016. You Look Only Once: Unified, real-time Object Detection, in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2016,* pp. 779-788

14. Piyush K. “Freenect_stack” *ROS Wiki* https://wiki.ros.org/freenect_stack

15. “How do I See Depth?” http://scecinfo.usc.edu/geowall/stereohow.html

16. Rostam Affendi Hamzah, Haidi Ibrahim. 2016. Literature Survey on Stereo Vision Disparity Map Algorithms, in *Journal of Sensors, Vol-2016 Article 8742920*

17. H. Durant-Whyte, T. Bailey. 2006. Simultaneous Localization and Mapping. In *IEEE Robotics and Automation Magazine, Vol-13 Issue 2 (June 2006),* pp. 99-110.

18. Yolila Arora, Inshan Patil and Thao Nguyen. Fully Convolutional Network for Depth Estimation and Semantic Segmentation

19. Y. Ma and S. Soatto and J. Koeck and S.S. Sastry, ”An invitation to 3-D vision from images to geometric models,” *Springer, New York, 2004*, pp. 121.

20. Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. J. of Computer and System Sciences, 55(1):119–139, 1997.

21. David G Lowe. 2004. Distinctive Image Features from Scale Invariant Keypoints, *International Journal for Computer Vision, 2004.*

22. Chris Harris & Mike Stephens. 1988. A Combined Corner and Edge Detector

23. Random Sample Consensus (RANSAC), *Wikipedia.* https://en.wikipedia.org/wiki/Random_sample_consensus

24. Christopher-Longuet Higgins - 8 point algorithm, *Wikipedia,* https://en.wikipedia.org/wiki/Eight-point_algorithm

25. Munir Zaman. 2007. High Resolution Relative Localisation using two Cameras, in *Robotics and Autonomous Systems 55 (2007)* pp. 685-692

26. Kurt Konilige, Motilal Agrawal, Robert C. Bolles, Cregg Cowan, Martin Fischler, Brian Gerkey. Outdoor Mapping and Navigation using Stereo Vision

27. Boxin Zhao, Tianjiang Hu, Lincheng Shen. 2015, Visual Odometry: A review of Approaches. In *Proceedings of 2015 IEEE International Conference on Information and Automation,* Lijiang, China (August 2015)

28. Russel. A. Epstein, Eva Zeta Patai, Joshua B. Julian, Hugo J. Spiers. 2017. The Cognitive Map in Humans: Spatial Navigation and Beyond, in *Nature Neuroscience 20(11) , (2017, Oct 26) 1504-1513*