Path Planning for Autonomous Mobile Robots

Khalid Bashir[1], Sohail Abbasi[2], Waqas Nawaz Khokhar[1]

[1]Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah, KSA
[2]School of Electrical Engineering and Computer Science, National University of Science and Technology, Islamabad, Pakistan

Summary
Building autonomous and intelligent robots has been an elusive dream for researchers for some time. Simultaneous Localization and Mapping (SLAM) systems have contributed towards achieving this goal by making robots better in navigating through complex environments. Until now it has only been possible to train and teach robots to move around in particular environments using a certain set of rules and heuristics. With the sudden surge in interest in AI and Machine Learning, a lot of effort has been put in into making robots intelligent and for them to automatically learn their paths in unknown environments (also referred to as Path Planning). This however has been met with mixed results as either the solution proposed is not too practical (e.g. requires too much training) or has limited success (e.g. works in specific environments). In this research, we develop a novel autonomous path planning framework using Deep Learning which can learn to navigate in unknown environments. The system has been tested on state-of-the-art Active Vision Dataset with promising results.

Key words:
Autonomous Robots, Robot Navigation, Path Planning, Simultaneous Localization and Mapping, Convolutional Neural Networks.

1. Introduction

Simultaneous Localization and Mapping (SLAM) "is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it" [1]. SLAM solves the problem of detecting the location of a moving sensor. This is the fundamental ability required by autonomous agents to plan actions in the environment. Visual SLAM (vSLAM) deals with the SLAM problem with the use of one or more cameras providing a constant stream of visual imagery.
Fig. 1 shows the vSLAM problem. The robot receives a sequence of successive (mostly overlapping) images from which it generates a 3D map of the environment while simultaneously keeping track of its current location in the environment.

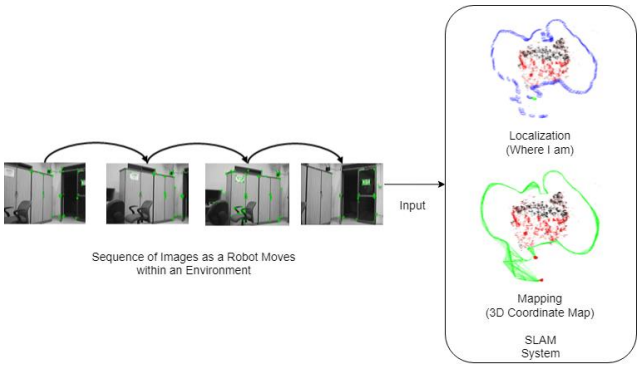

Fig. 1 The Visual SLAM Problem

This area has been of particular interest to the Robotics Community where Monocular SLAM systems (vSLAM systems built using a single camera) have been extensively explored due to their low cost and simplicity. Over the years, vSLAM has evolved a great deal with slow and inefficient systems giving way to efficient and real time solutions. ORB-SLAM [2] and SVO-SLAM [3] are two examples of such systems where solution to the SLAM problem is efficient, quick and robust. These systems use feature descriptors to keep track of the scene locations from within the image stream. Some of the more prominent features used by SLAM systems include FAST, SIFT, SURF and ORB, etc., each having their own set of advantages to support their use in specific use cases. A lot of research in the SLAM community has been focused on making robots more autonomous by enabling them to navigate unknown environments through the use of path planning.
Path planning is a known problem in robotics, where it is meant to equip the robot with a method to navigate to a particular location in a scene/environment. Historically, researchers have focused on an algorithmic approach where a set of heuristics/rules is used to decide how to navigate the robot through the environment. The algorithmic approach however, generally fails when the robot is introduced into a new environment and requires generation of a new set of rules for every unique scenario.
In this work we propose a method to enable the robot to navigate an environment by using a machine learning approach to scene understanding. Instead of programming via a set of rules, we will take an approach where the robot

Manuscript received November 5, 2019
Manuscript revised November 20, 2019

learns these rules by using a machine learning approach. The robot is first shown a set of images and the associated set of actions needed to navigate from one scene to another. The set of images are taken from a sequence of overlapping images (the sort that a robot sees when incrementally navigating an environment). These set of labeled images and actions are used to train a Convolutional Neural Network (CNN) which implicitly understands the image contents and infers what actions are required to be taken to reach a goal. By providing the robot many examples of what moves to make under different situations, it will gain the ability to navigate in unknown cases as well. The proposed system is demonstrated on example images from the gold standard Active Vision Dataset [4] with promising results.

A peek into the upcoming sections is as follows: Section 2 discusses the related work in the field. Section 3 provides the details of the proposed system for the path planning using learning methods. Section 4 shows the experiments, along with their results and discussion. Section 5 concludes the paper.

2. Related Work

Like many problems in the computational vision domain Visual Simultaneous Localization and Mapping (vSLAM) does not come as easily to machines as it does to humans. A seemingly easy problem for humans is a challenging task for machines under real world conditions. In an effort to make it feasible a wide variety of techniques have been proposed to tackle various aspects of vSLAM ranging from features based methods to RGB-D based camera approaches. A survey of vSLAM methods can be found in the work of [5]. General surveys of SLAM methods are to be found in [6] and [7]. A more approach centered survey focusing on visual aspects is provided by [8].

Path planning is a major research area in SLAM systems with a goal of making robots more autonomous. Maps built by standard SLAM based systems cannot be used to plan paths. A potential solution to this problem is proposed in [9] using Pose SLAM graph to determine the path between two robot configurations with lowest accumulated pose uncertainty. A slightly different approach using a belief-function is presented in [10]. By representing different types of uncertainty in evidential grid maps. In addition, the paper also proposes optimal navigation and exploration for better path planning. Both these approaches are heuristics based and have their limitations when new and unknown scenarios are encountered in real-world environments.

Losing localization is an analogous problem to path planning in real world vSLAM systems. Robots tend to lose localization due to the uncontrolled movement of the camera, lack of sufficient matching between current and previous images and uncertainties of the environment. Visual inertial SLAM system in [11] provides a solution to overcome this problem using visual-inertial odometry and provides locally consistent trajectory and map estimates, while global consistency is achieved using online loop-closing and non-linear optimization. Binary features have been reported to have orders of magnitude better performance than the traditional feature descriptors [12]. Visual place recognition schemes using a hashing mechanism based on these binary features to speed up the process have been proposed [12, 13] to recover from localization errors and globally relocalizing the robot. In [14] advances in key point recognition are used to determine the camera pose.

These re-localization methods for recovering from lost state mainly use a previously visited place and compare the feature descriptors of the current image with the database of stored image descriptors. This process re-localizes the robot in the environment and the robot continues with the SLAM process.

Similar to the approaches used in re-localization, wherein lost state is recovered using previously visited places, we can use previously collected visual information to train a machine learning model for predicting next actions of the robot in order to achieve better results in path planning. Our approach learns the program (Robot movements) from the available data (Image sets) by training a Convolutional Neural Network that outputs next movement action, incrementally moving the robot from one place to another. The details are provided in the next section.

3. The Proposed System

Convolutional Neural Networks (CNNs) have proven themselves to be invaluable resource in scene-understanding problems due to their capability of preserving spatial information. The proposed solution to the path planning problem builds upon this concept and uses a custom designed CNN to predict what motion to undertake to enable a robot to move from one scene to another. Using the CNN for this problem is a process involving two steps: training and inference. During training the CNN is shown a pair of images; the first corresponding to the scene that the robot is presently positioned at, and the second where we intend the robot to go to. With each pair of images, the CNN is also shown a motion label depicting the motion undertaken in order to move from the first scene to the second. Having been fed a series of such “training samples” the CNN adjusts its parameters so that it can later suggest the action to undertake in order to move from one scene to another. These set of parameters need to be adjusted only once and can be saved to disk as a “model” file. This process is shown in Fig. 2.

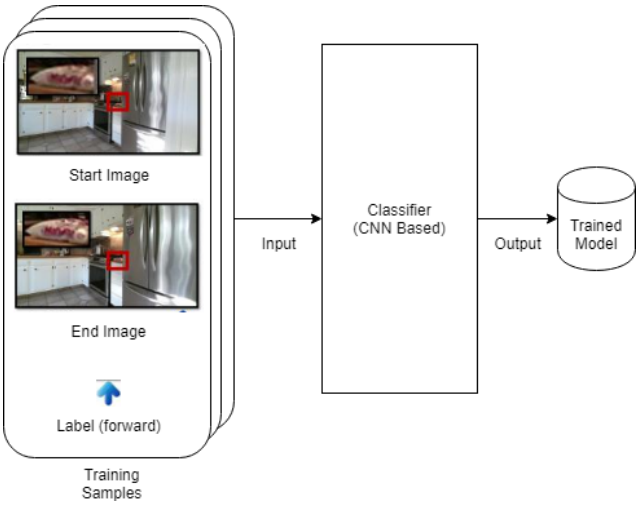

Fig. 2 Training

In the inference phase, the model trained in the training step is utilized to make a prediction on what action the robot must undertake in order to move from the first scene to the other. This process is shown in Fig. 3.

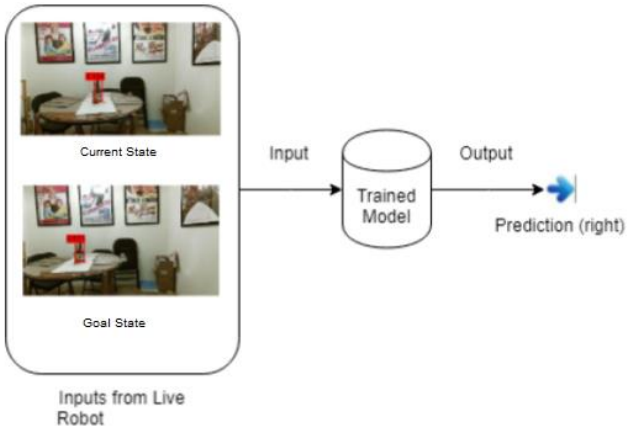

Fig. 3 Inference

3.1. System Architecture

In order to counter the above-mentioned path planning problem, we need a dataset of sequential images from an example path exploration. We extract these from the Active Vision Dataset and feed it to our CNN as described above in order to train it.

3.2. CNN Architecture

Convolutional Neural Networks are a variation of regular neural networks especially suited to image recognition and understanding tasks [15]. The network represents a differentiable function from image pixels to class probabilities.

CNNs are chosen for this problem because Conventional Neural Nets don't scale well to images. If a regular Neural Network is used for this problem of path planning we would require all the image pixels to be passed through multiple fully connected layers, which in-turn would mean having to fine tune a large number of weights. In the first layer this would mean having a large number of weights (image_width x image_height x number_of_channels). In the case of a CNN, this number is dramatically reduced (number_of feature_maps x filter_size). This is depicted in Fig.4. This dramatic reduction in number of weights follows from the observation that all the pixels of the image need to be operated on in a similar manner and thus can share their weights. The computation size is further reduced in subsequent layers through the use of a pooling mechanism (mostly max-pooling). Another added advantage of CNNs is their ability to preserve spatial information, which is extremely useful in image related tasks.

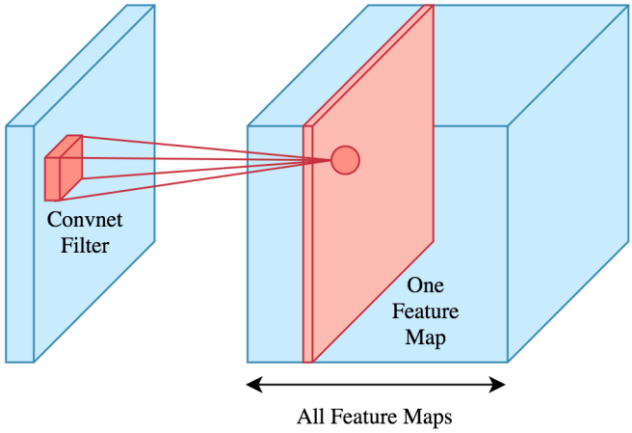

Fig. 4 A Single layer of a Convolutional Neural Network

We have reduced the size of each training image at the first layer of the convolutional neural network by scaling and then randomly cropping a 224 x 224 portion of the image. This is followed by batch normalization operation around the mean. Random cropping helps in generalizing the neural network to unseen possibilities while batch normalization

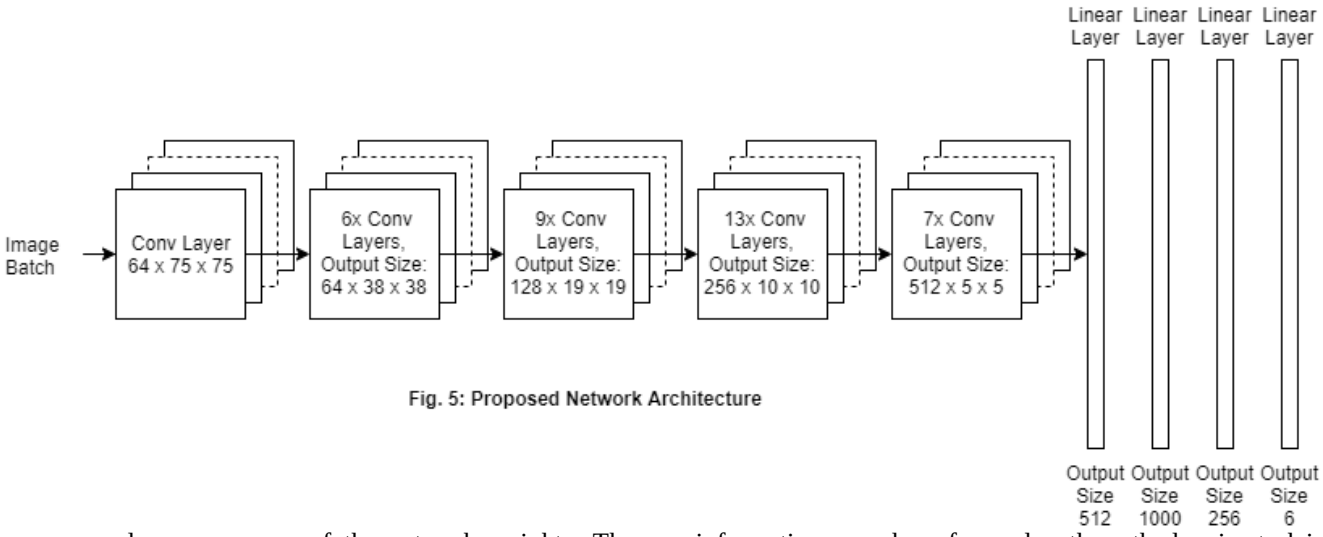

Fig. 5: Proposed Network Architecture

ensures early convergence of the network weights. The inference images also need to go through the above set of transformations. However, a random crop is not needed and it is instead center-cropped to conform to the input dimensions.

The Convolutional Neural Network designed for this problem takes as input the correctly cropped and transformed images as described earlier. The architecture is depicted in Fig. 5. An optimized architecture has been decided upon by systematically tuning the hyper parameters. The transformed image is passed to a Convolutional Layer and subsequent max-pooling layer which produces 64 feature maps each of dimension 75 x 75. This is followed by a sequence of six convolution-only layers which are finally max-pooled to arrive at 128 feature maps of dimension 19x19. The process is repeated (as depicted in Fig. 5). The output from the final convolutional layer is flattened to produce a linear vector of size 512. The subsequent network is a fully-connected neural network which culminates into an output of size 6 representing the probabilities of the 6 possible actions to be taken by the robot. The actions represented by the outputs are Forward, Backward, Strafe-Left, Strafe-Right, Rotate-CCW and Rotate-CW.

The system is trained and tested on state-of-the-art Active Vision Dataset [4] details of which are described next.

3.3. Dataset

Active Vision Dataset is a dataset of images acquired by a real-world robot moving through a variety of indoor environments. It is specifically designed for tasks related to robot vision. It comprises of 30,000+ images across 15 distinct environments out of which we use a subset of ~10,000 images for this research. Images are randomly chosen without any bias. The source dataset also contains depth information for images and bounding boxes for object detection tasks. However, in this research we do not use this information as we have focused on the path planning task in a monocular SLAM setting. Sample images from the dataset are shown in Fig. 6. The annotation files for the dataset provide the movement action of the robot as it moves from one frame to another. The possible action labels provided are Forward, Backward, Strafe-Left, Strafe-Right, Rotate-CCW and Rotate-CW. In the dataset, extent of movement of the robot after taking each action is kept constant. Sample annotation structure for the JSON annotation file is shown below:

```
"000610000010101.jpg":{
        "bounding_boxes":[
                [1308,500,1331,520,25,5]
        ],
        "rotate_ccw":"000610000020101.jpg",
        "rotate_cw":"000610000120101.jpg",
        "forward":"000610000130101.jpg",
        "backward":"",
        "left":"000610002170101.jpg",
        "right":""
},
"000610000020101.jpg":{
        "bounding_boxes":[
                [570,705,671,796,16,3]
        ],
        "rotate_ccw":"000610000030101.jpg",
        "rotate_cw":"000610000010101.jpg",
        "forward":"000610001100101.jpg",
        "backward":"",
        "left":"",
        "right":"000610000140101.jpg"
}
```

In the annotation sample above, the robot at frame “000610000010101.jpg” moves to frame “000610000020101.jpg” as it takes the Rotate-CCW action.

In the next section we discuss the usage of the Active Vision Dataset in our experimental setting for use in our Path Planning problem along with the results and discussion.

Fig. 6 Sample Images from different scenes of the Active Vision Dataset

4. Experiments, Results and Discussion

4.1 Experiments

As mentioned earlier, the annotation files for the Active Vision Dataset provide the relative position of the robot as it moves from one frame to another. Using these annotation files, we extract a pair of images with an associated robot movement action that depicts the action as the robot moves between the pair of images. The pair of images are then converted into grayscale and fused into a single multi-channel image. In the training phase these dual-channel images are fed into the network along with their action labels. The images thus produced are split into training and validation sets with a ratio of 80-20. These inputs are fed into the network in batches of 16 images until the weights converge to acceptable loss value for the validation set. The learning rate for the network is to 0.00001.

An unseen test set comprising of around 2000 images is used for measuring the performance of the proposed architecture. In the inference phase, dual-channel images are generated from image pairs of the test set in a manner similar to the one used in the training phase. The pair of images depict starting and end way-point in a robot's proposed path. Ground truth values of movement actions from the annotation files are used for calculating the system's performance in the inference phase.

4.2 Results and Discussion

Training/Validation was conducted on ~50,000 dual-channel images generated from the Active Vision Dataset as explained earlier. Although the source data contains only ~10,000 images, the image pairs generated for each action label are much higher because a single image is related to many other images through different actions. The training loss and accuracy graphs over 10 epochs of training are presented in Figs. 7 and 8 respectively:

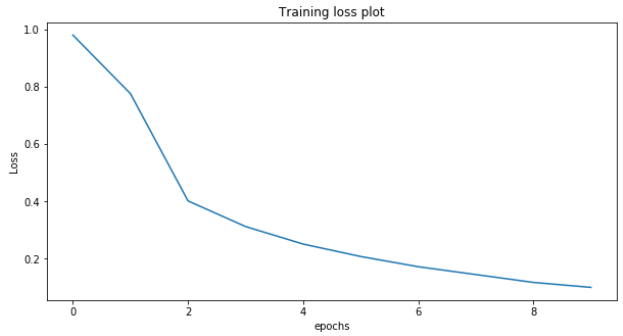

Fig. 7 Training Loss Plot

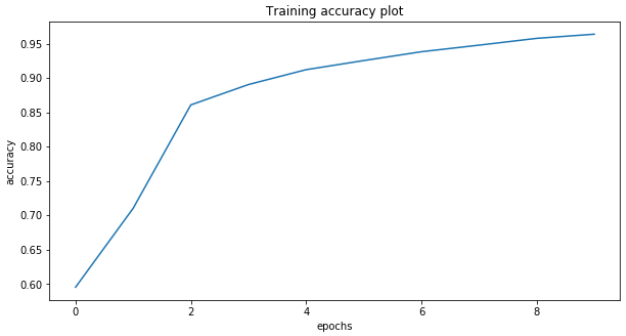

Fig. 8 Training Accuracy Plot

The above plots clearly show that the validation accuracy is steadily increasing and the loss is respectively decreasing over successive epochs. This depicts the efficacy of the proposed method and validates it to be well-suited to the problem at hand.

Performance results from testing the system on the test set of ~2000 images from unseen environments is presented below in the form of a confusion matrix (Fig. 9). As highlighted earlier in training dataset results, although the source data contains only ~2,000 images, the image pairs generated for each action label are much higher (approx. 15,000 image pairs).

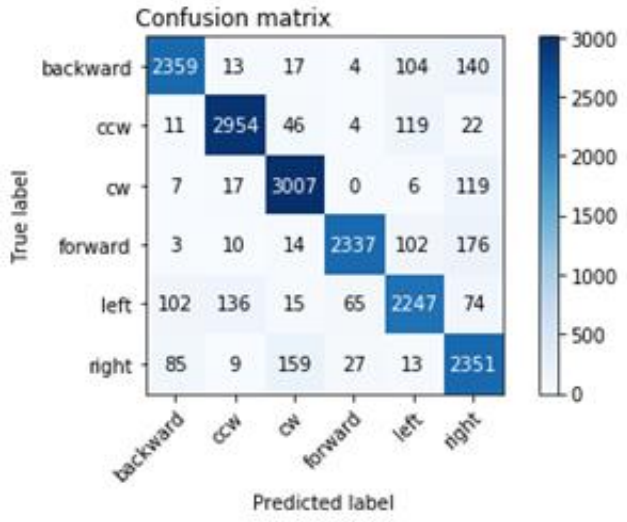

Fig. 9 Confusion Matrix

It is clear from Fig. 9, that the majority of the results fall in the diagonal entries, which indicates that the proposed method correctly predicts the action to undertake for planning the robot's path between two frames. The results also show that most cases of misclassification arise from confusion between the Left and CCW movement and Right and the CW movement. This is expected since Left and CCW Motion produce similar changes in the images captured by the robot. The same is true for Right and Clockwise actions.

The overall test result metrics (Precision, Recall, Accuracy and F1-Score) are tabulated below:

Table 1: Test Results

S. No.	Metric	Value
1.	Precision	90.41
2.	Recall	90.14
3.	Accuracy	90.40
4.	F1-Score	90.21

The final test accuracy is reported to be 90.40. Precision, recall and F1-Score are all similarly high. It can be deduced that increasing the number of training images pairs will further increase the accuracy and should result in reducing the misclassifications between similar action pairs.

5. Conclusion

Path planning for robots has generally been tackled using heuristic rule based approaches. In this work we have demonstrated how proven deep learning techniques can be used to improve path planning for autonomous robots. Results indicate that the technique can prove to be a viable alternative and can even lead to improvement in path planning.

Acknowledgment

This work was supported by the Deanship of Research, Islamic University of Madinah, Kingdom of Saudi Arab [Project Title: "Path Planning for Autonomous Mobile Robots using Learning ", the 10th (Takamul) Program of Academic Year 1440-1441 AH]

References

[1] Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localization and mapping: part I. IEEE robotics & automation magazine, 13(2), 99-110.

[2] Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE transactions on robotics, 31(5), 1147-1163.

[3] Forster, C., Zhang, Z., Gassner, M., Werlberger, M., & Scaramuzza, D. (2016). SVO: Semidirect visual odometry for monocular and multicamera systems. IEEE Transactions on Robotics, 33(2), 249-265.

[4] Ammirato, P., Poirson, P., Park, E., Košecká, J., & Berg, A. C. (2017, May). A dataset for developing and benchmarking active vision. In 2017 IEEE International Conference on Robotics and Automation (ICRA) (pp. 1378-1385). IEEE.

[5] Taketomi, T., Uchiyama, H., & Ikeda, S. (2017). Visual SLAM algorithms: a survey from 2010 to 2016. IPSJ Transactions on Computer Vision and Applications, 9(1), 16.

[6] Bresson, G., Alsayed, Z., Yu, L., & Glaser, S. (2017). Simultaneous localization and mapping: A survey of current trends in autonomous driving. IEEE Transactions on Intelligent Vehicles, 2(3), 194-220.

[7] Jamiruddin, R., Sari, A. O., Shabbir, J., & Anwer, T. (2018). Rgb-depth slam review. arXiv preprint arXiv:1805.07696.

[8] Fuentes-Pacheco, J., Ruiz-Ascencio, J., & Rendón-Mancha, J. M. (2015). Visual simultaneous localization and mapping: a survey. Artificial Intelligence Review, 43(1), 55-81.

[9] Valencia, R., Morta, M., Andrade-Cetto, J., & Porta, J. M. (2013). Planning reliable paths with pose SLAM. IEEE Transactions on Robotics, 29(4), 1050-1059.

[10] Clemens, J., Reineking, T., & Kluth, T. (2016). An evidential approach to SLAM, path planning, and active exploration. International Journal of Approximate Reasoning, 73, 1-26.

[11] Kasyanov, A., Engelmann, F., Stückler, J., & Leibe, B. (2017, September). Keyframe-based visual-inertial online SLAM with relocalization. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 6662-6669). IEEE.

[12] Vysotska, O., & Stachniss, C. (2017, September). Relocalization under substantial appearance changes using hashing. In Proceedings of the IROS Workshop on Planning, Perception and Navigation for Intelligent Vehicles, Vancouver, BC, Canada (Vol. 24).

[13] Feng, Y., Wu, Y., & Fan, L. (2017). Real-time SLAM relocalization with online learning of binary feature indexing. Machine Vision and Applications, 28(8), 953-963.

[14] Williams, B., Klein, G., & Reid, I. (2011). Automatic relocalization and loop closing for real-time monocular

SLAM. IEEE transactions on pattern analysis and machine intelligence, 33(9), 1699-1712.

[15] Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham.

Khalid Bashir received the B.Eng. (Computer Engineering) from National University of Engineering and Technology, Pakistan in 2002 and M.S. in Computer Science from Queen Mary University of London, UK in 2007. He did his Ph. D in Computer Science from Queen Mary University of London, UK in 2011. He has been an active researcher and Associate Professor at Faculty of Computer and Information Systems, Islamic University of Madinah. His research interests are Computer Vision, Machine Learning and Deep Learning.

Sohail Abbasi is a Computer Engineer and received his bachelor's degree from National University of Science and Technology, Pakistan. He has also got a Masters in Computer Engg. from University of Engineering and Technology, Taxila, Pakistan. He was awarded the President's Gold Medal for Best Research Project in his degree. He is a keen researcher with interests in Machine Vision and Learning Methods.

Waqas Nawaz is Assistant Professor at Islamic University Almadinah, Kingdom of Saudi Arabia. He served as post doctoral fellow from 2015 to 2016 at Innopolis University, Russia. He has received his Ph.D. degree from Kyung Hee University, South Korea in July, 2015. He has completed his B.S. and M.S. from University Institute of Information Technology and National University of Computer and Emerging Sciences (NUCES-ISB), Pakistan, in 2008 and 2010, respectively. He is currently pursuing his research work in the areas of large graph mining, data mining, big data analytics, clustering, image processing, computational intelligence, intelligent mobility and databases.