Experience Based Localization in Wide Open Indoor Environments

Rahul Gautam*, Harsh Jain, Mayank Poply, Rajkumar Jain, Mukul Anand, Rahul Kala
Department of Information Technology, Indian Institute of Information Technology Allahabad, Allahabad, India

*Corresponding Author: rahul.gautam21@gmail.com

Citation: R. Gautam, H. Harsh Jainm, M. Poply, R. Jain, M. Anand, R. Kala (2018) Experience based localization in wide open indoor environments. Paladyn, Journal of Behavioral Robotics 9(1): 82–94.

Final Version Available At: https://www.degruyter.com/view/j/pjbr.2018.9.issue-1/pjbr-2018-0006/pjbr-2018-0006.xml

Abstract: This paper solves the problem of localization for indoor environments using visual place recognition, visual odometry and experience based localization using a camera. Our main motivation is just like a human is able to recall from its past experience, a robot should be able to use its recorded visual memory in order to determine its location. Currently experience based localization has been used in constrained environments like outdoor roads, wherein the robot would be found at nearly the same place at every time of the visit. The paper adapts the same technology to wide open maps like halls wherein the robot could be at a new spot on every visit. When a robot is turned on in a room, it first uses Visual Place Recognition using Histogram of Oriented Gradients features and a Support Vector Machine in order to predict which room it is in. The robot then scans it's surrounding and uses a nearest neighbor search on the robot's experience coupled with Visual Odometry for localization. We present the results of our approach tested on an environment comprising of three rooms with a dataset comprising of approximately 5000 monocular and 5000 depth images and tested over different rooms with some dynamic changes.

Keywords: Localization; Visual Place Recognition; Experience Based Localization; Machine Learning.

1 INTRODUCTION

With the growing industry of self-driving cars, humanoid robots and industrial robots there is a need for the robots to become autonomous and be able to navigate themselves in any environment. For this purpose, we need precise and accurate robot localization and tracking algorithms, many of which are inspired by the human visual system and use visual hardware such as stereo systems or RGBD (Red-Green-Blue-Depth) sensors.

Localization is the basis of every robotic application. In order for any robot to do its required task, it should answer the question "where am I?" All localization techniques determine the robot's location and orientation in the given environment. The process of mapping cannot be separated from the process of localization, as localization is usually performed with respect to the surroundings or the environment. Maps are usually represented in two frameworks, the metric framework which is uses a geometric model of the map and is the most commonly used framework; and a topological framework in which we consider the places and the relations existing between the places.

Experience based localization is a recent technique used to solve the problem of localization, wherein the past experiences of the robot are stored and used for answering the localization queries. The technique is widely used in a topological or constrained environment wherein the robot travels by nearly the same position every time it travels the same place. The most common example is self-driving cars and corridors wherein there is a very little lateral dispersion that the robot may show on the road or corridor, and therefore at every instant of travel the percept is the same. This paper solves the problem of localization for wide open indoor environments wherein the robot could be anywhere inside a big hall. We also present a hybrid approach of localization using both experienced based localization and visual odometry.

The main aim of this paper is that whenever a robot is switched on at an unknown location, the robot should be able to identify its location. The problem is decomposed into finding out which room it is in, and the location and orientation in that room. The paper aims at solving the problem of indoor localization on a previously built static map represented using a topological framework. We use the past information/experience of the robot in order to recognize the room in which the robot is present using Visual Place Recognition. After that we perform experience based localization by extracting out only the experiences that are relevant from the database using a Nearest Neighbor classifier and perform visual odometry on the images. We use visual odometry so that we can adjust to some dynamic changes in the environment for which our experience based localization algorithm might not give the accuracy required. The entire process is summarized in figure 1.

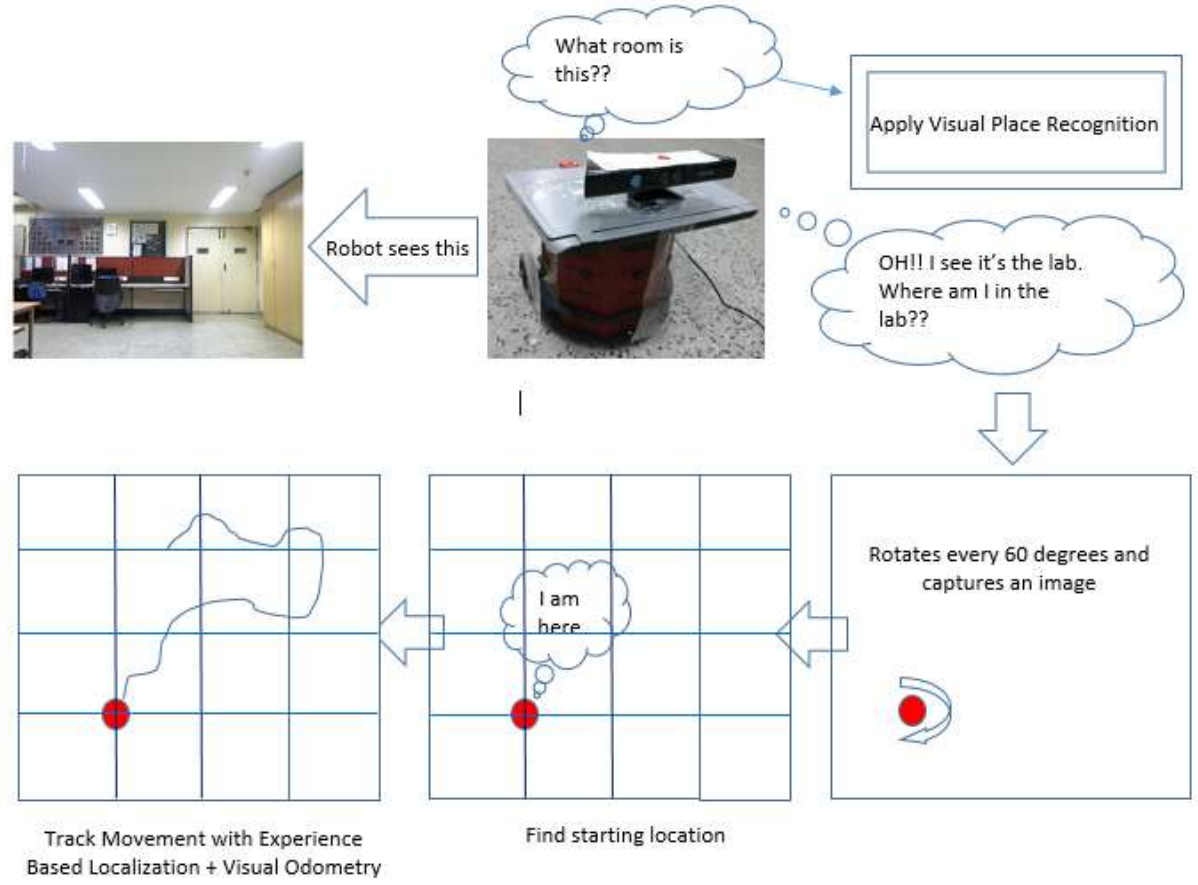

Figure. 1: Project Overview: This figure shows the entire process of the project approach starting with Visual Place Recognition, followed by finding the starting point and Experience Based Localization operating in parallel with Visual Odometry.

2 LITERATURE SURVEY

Lowe [1] extracted SIFT features for visual place recognition. The authors matched SIFT features against a database of features using a Hough transform and a fast nearest-neighbor algorithm, and finally the best pose parameters verification was done through the least-squares solution. Lowry et al. [2] created a topological graph and then performed localization via SIFT feature matching and for next frames search is only done among the neighbors of the detected node. The paper also presented the process of navigation in the topological graph.

Linegar et al. [3] used both visual odometry and visual place recognition in order to localize a robot in a time and weather varying environment. If visual place recognition failed, the robot was localized using the visual odometry results, till the time it was able to re-localize itself in the environment and continue with localization using visual place recognition. The process led to creation of sophisticated and robust maps. The authors present localization results of a car driven on a 37 km track.

Napier et al. [4] demonstrated online vehicle pose estimation with only the use of a stereo camera. The approach used visual appearances of the scenes in order to perform pose estimation of the vehicle along with the visual SLAM generated trajectory. Fiala and Ufkes [5] performed visual odometry by matching 3-D points using corresponding 2-D

descriptors for images. The paper used the 3D data along with SIFT or SURF features to find the camera's rotation and translation by using RANSAC (Random sample consensus) algorithm thus performing live visual odometry.

Pronobis et al. [6] used an appearance based approach for localization working under different illuminations and over a large span of time. The approach used a rich descriptor composed of a high dimension histogram of the image and a SVM. Shi and Thomasi [7] discussed on the problem of selecting feature points corresponding to physical world points which are good to track. The authors explained how usually points with big eigenvalues are sufficient to be detected as corners in images, however they may not necessarily be corners in the physical world. The authors modified the scoring function from Harris Corner Detector [8] to obtain better features for tracking.

Luo et al. [9] demonstrated an adaptive visual place recognition algorithm which was able to learn from its experience and continuously adapt to changes in the environment. The recognition algorithm was able to adapt to changes in its database and updated it. It used incremental SVM in order to keep the algorithm running within the memory requirements. Pronobis et al. [10] showcased the ability of a software to learn in realistic settings. The authors used incremental SVM based learning in order to build recognition models of the environment. Pronobis and Caputo [11] approached the visual place recognition problem along with a degree of the confidence of the answer. The authors used a SVM classifier in order to classify the hypothesis and distance from the SVM classification hyperplane in order to get the confidence of the hypothesis.

Some other related approaches are also briefly discussed. Llorca et al. [12] presented a vehicle logo recognition algorithm using a sliding window combined with a HOG and SVM classifier. Shimosaka et al. [13] presented an indoor localization mechanism with the use of Zig-Bee devices. Kala [14] described the different localization schemes in a self-driving car technology and explained the importance of localization in planning and decision making. Kadota et al. [15] explained the problem of pedestrian recognition on an embedded system using a simplified HOG algorithm. Henry et al. [16] built dense 3-D maps of indoor environments. The paper used visual and depth information combined with loop closure detection and finally optimized the poses obtained in order to build maps which are globally consistent. Dryanovski et al. [17] used a RGBD sensor and then used real-time visual odometry and mapping. The authors used visual odometry with Kalman filter. The authors used a registration algorithm which was able to detect small loops online. Irie et al. [18] presented an outdoor stereo vision based localization system. The system took care of illumination changes by forming two dimensional occupancy grid maps from the three dimensional point clouds. The robot's pose was estimated using a particle filter combining visual odometry and map matching.

Kaundal et al. [19] localized a fixed node by pursuit nodes in outdoor localization using two different methods, first using LNSM (Log Normal Shadowing Model) which used ZigBee protocol and the received signal strength for localization; and the second using a method based on Hybrid TLBO (Teacher Learning Based Optimization Algorithm)-Unilateral technique. The paper also compared the localization results for both the methods and presented them. It also showed that the fixed node becomes 100 per cent discoverable by use of the second method. Rathnam and Birk [20] presented a 3D exploration algorithm using a group of robots which are always within communication range of each other. Since the method is based on a greedy computation strategy using a heuristic function, it becomes computationally efficient. The paper also presented an efficient strategy to recover from deadlocks resulting from local minimums. The exploration algorithm was tested on a simulator for Autonomous Underwater Vehicles (AUV) taking into account their sensors. The effect of increasing and decreasing the number of robots and the communication range of the robots on the algorithm was also investigated.

Li et al. [21] presented an adaptive clustering technique based on exploration and exploitation strategies in contextual multi-armed bandit settings to improve the performance of context based filtering and classic collaborative filtering methods which do not perform well under dynamic recommendation domains such as news recommendation and computational advertisement. The algorithm used the collaborative effects arising between the interactions of users with the items and then dynamically grouped the users based on the items under consideration. Li [22] also proposed algorithmic solutions to the networked bandit problem so as to integrate the strong social components in the bandit algorithms which increased the performance of the algorithm drastically. Starting from the global Laplacian strategy where each node has its own bandit algorithm and is allowed to share signals with its neighboring nodes, the paper aimed to develop and experimentally tested more strategies of clustering nodes which were more scalable. Li and Kar [23] proposed a Context-Aware clustering of bandits (CAB) algorithm which captured collaborative effects. CAB dynamically clustered the users based on the content universe in situations such as the real-world recommendation system where multi armed bandits performed pretty well. Korda et al. [24] provided algorithms for solving the linear bandit problem in peer to peer networks with limited communication capabilities. First the authors assumed all peers solved the same linear bandit problem and prove that the algorithm achieves optimal asymptotic regret rate. Then the authors assumed that within clusters there

were clusters of peers solving the same bandit problem and the algorithm discovered these clusters while within each cluster it achieved the optimal asymptotic regret rate.

Kar et al. [25] presented online stochastic algorithms for quantification-specific measures in order to perform class prevalence that is what fraction of the population belongs to a particular class. The paper also presented hybrid algorithms to balance out quantification and classification performance. Yoo et al. [26] proposed a semi-supervised machine learning algorithm. It improved localization performance as training data comes from received signal strengths of wireless communication link and thus reduced the needs of calibrating labelled data from the unlabeled data. The algorithm was used for evaluating the position of a smartphone mobile robot. The experimental results showed that the algorithm did not compromise with the computation speed and was more robust compared to state-of-the-art semi supervised learning algorithms. Park and Roh [27] presented a global localization technique based on a 2-D range scan place recognition technique. The paper using a support vector machine (SVM) to train a set of classifiers for place recognition. The paper used a bag-of-words approach to create the feature vectors for training the SVM classifiers for place recognition. The algorithm first produced coarse localization for selecting the best candidate places where the robot may be located based on place recognition. After that fine localization was done via fast spectral scan matching and particle filter algorithm. The paper also presented the results of extensive simulations and experiments.

Lu et al. [28] presented an image-based indoor localization system based on thermal imaging which can be used in case of emergencies such as a blackout. Learning was applied in order to enrich the thermal imaging classification. Active learning enhanced the performance of the algorithm and the algorithm was able to properly localize the robot in a dark environment. Winterhalter et al. [29] presented an accurate indoor localization method for RGB-D smartphone or tablet. The method only used the two dimensional outline of environment as the map from architectural drawings in order to perform localization using the floor measurements and differences between floor plans and real world data using the sensors present on the device. The algorithm used a particle filter to estimate the 6DoF pose of the device using the sensors. Results of the localization approach are also presented on a Google Tango device.

Most of the above mentioned techniques have been tried in outdoor scenarios wherein the map is topological in nature and the vehicle can show a very small lateral movement. This means the experience will be the same every time the vehicle passes the same regions. The algorithms are adaptive to corridor like situations where also the map is topological. For halls and other rooms with wide open spaces, the robot may be found at any place and the same place may be different every time the robot passes by, thus creating possibilities of a very large number of experiences possible. Such a setting is not actively researched. Further, the proposed approach integrates Visual Place Recognition with Experience Based Localization to limit the number of experiences to be considered during search. In other cases wherein wireless sensors are used, the wireless sensors are known to be prone to noise. Other algorithms which propose effective localization do not give localization approaches when the robot is travelling short distances in which case the visual odometry becomes far more efficient than experience based localization.

3 BACKGROUND

3.1 GFTT (Good Features to Track)

The method is similar to Harris Corner Detection [30] where we move a window over the entire image and find values where Eigen vectors have maximum values which are found out by finding the derivatives along X and Y directions. Shi and Thomasi made some modification to the original formula such that tracking is improved and can overcome problems like occlusions and features that do not correspond in the real world. The proposed work uses GFTT features for selecting the feature points in the images to compute visual odometry between two images.

3.2 Visual Odometry

Using visual odometry we estimate the pose of an object which in our case is represented by X-coordinate value, Y-coordinate value and orientation angle measured in anti-clockwise direction from the X-axis (θ).

PnP (Perspective-*n*-Point) [31] problem is estimating the pose of a given calibrated camera from given sets of 3D points.

We follow the below steps in order to perform visual odometry between two images:-

1. Take input images from camera or any other input stream.

Extract GFTT features and descriptors from the image. First we select the features in the images. The GFTT features selected in the image for a sample image are shown in figure 2.

Figure. 2: GFTT Features: The green dots show the GFTT feature points detected in our image.

2. Match the descriptors between the images using FLANN (Fast Approximate Nearest Neighbor Search Library) matcher. Now we see the correspondence between the feature points in our image and the world coordinates as shown in figure 3. The figure also shows how 3-D points are mapped to corresponding $(u, \quad v)$ pairs between the images via red lines i.e. the 2-D projections of these 3-D points in the camera coordinate system.

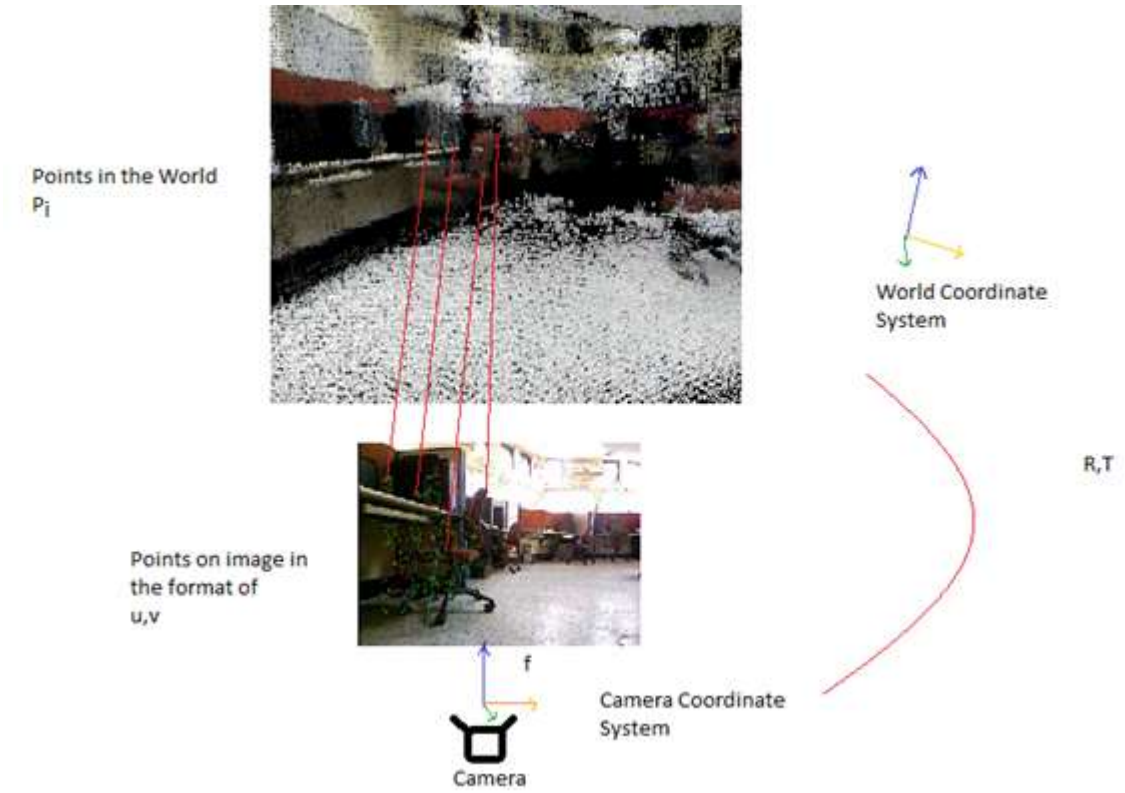

Figure. 3: Transformation from 2-D image to 3-D world coordinates: This image shows the correspondence between points on the image $(u, \quad v)$ and in the physical world i.e. P_i.

3. Perform pose estimation between the images using PnP (Perspective-n-Point problem) and RANSAC [32] *algorithm.* We aim to retrieve the pose (rotation *R* and translation *T*) using these parameters and the focal length of

the camera. During the whole process of computing visual odometry we use the pinhole model of the camera as shown in figure 4.

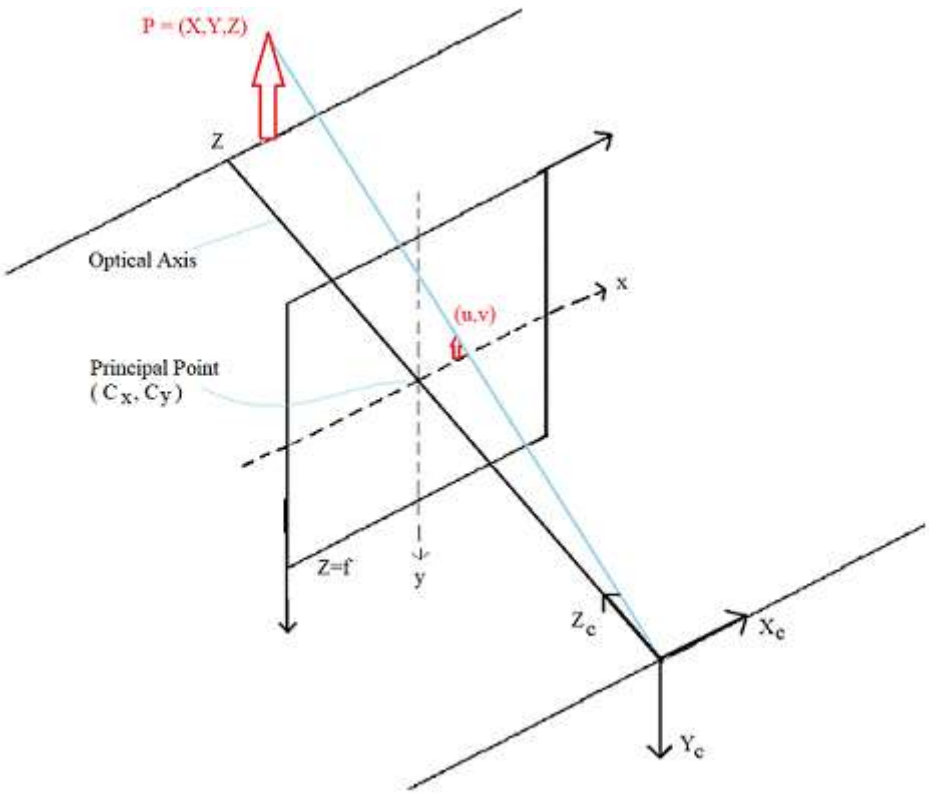

Figure. 4: Pinhole Camera Model.

We first project the obtained features into their corresponding 3-D points using equations (1) and (2):

$$sm' = A[R|T]M' \quad (1)$$

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13}t_1 \\ r_{21} & r_{22} & r_{23}t_2 \\ r_{31} & r_{32} & r_{33}t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2)$$

In equations (1) and (2), s is the scaling factor, $(X, \quad Y, \quad Z)$ are the world coordinates projection of the point pixel $(u, \quad v)$ in the image obtained via a camera having (f_x, f_y) as the focal length and (c_x, c_y) as the principal center. A represents the camera matrix i.e. the matrix of the camera's intrinsic parameters. The matrix [R|T] is called as the joint rotation and translation matrix .It is used to describe the camera motion in a static scene.

After we have obtained the 3-D points corresponding to the 2-D features we find the motion by applying PnP RANSAC. RANSAC is a process through which we are able to take care of these outliers by iteratively selecting a subset of points non-deterministically and finding inliers (points which have correct 3-D corresponding points) in them. We continue this same process many times till an approximate good data of inliers can be obtained. We usually select sufficiently high number of RANSAC iterations so that probability of finding inliers becomes high. After we are done with finding the set of inliers we proceed to the final step i.e. the camera pose estimation via PnP algorithm. The PnP algorithm returns the camera pose by finding the differences between inliers of two successive images and outputs the final rotation and translation vectors that the camera has gone through. The process is explained in figure 5 and algorithm 1.

Algorithm 1: Visual Odometry

1. Initialize_Camera using Calibration_File
2. $\tau(0) \leftarrow [R_Z(\theta_0)\ (x_0\ y_0\ 0)^T\ ;0\ 0\ 0\ 1]$ # θ is initial angle
3. t=0, I(t) ← 3Dsensor_Capture()
4. while (3Dsensor is 'ON'):
5. I(t) ← 3DSensor_Capture()
6. Transform T← Compute_Transformation (I(t), I(t-1))
7. If T is NULL:
8. Display "Could Not Compute Motion"
9. Else:
10. $\tau(t+1) \leftarrow \tau(t) \times T$
11. $x_{t+1}, y_{t+1}, \theta_{t+1} \leftarrow$ Generate_3Df(Pose)
12. Display x, y, theta
13. t←t+1

The function Compute_Transformation takes as input the previous image and the live image and returns the camera pose transformation matrix. τ is the trajectory of the camera, stored as a set of discrete poses. $I(t)$ is the input image at time t. The initial position is (x_0, y_0) and the initial orientation is θ_0. Performance of visual odometry is good for short distances however in case of longer runs the cumulative error becomes too large and hence we need to reinitialize the location of the robot and we suggest doing this using the output of experience based localization.

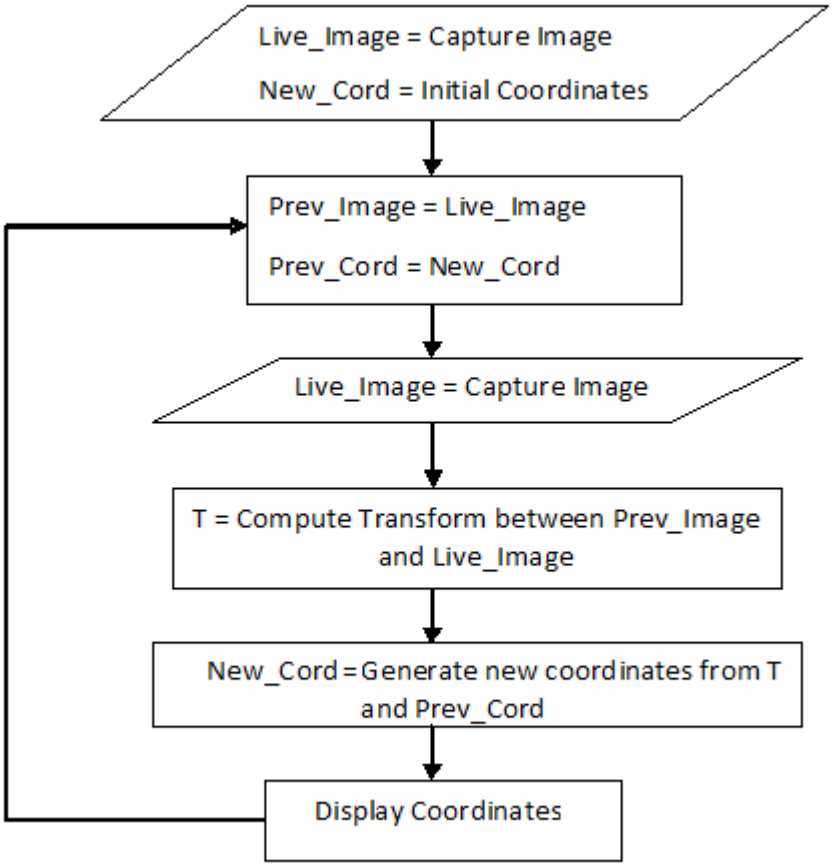

Figure. 5: Flowchart of Visual Odometry: This figure shows the process of visual odometry for robot pose estimation.

3.3 Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients (HOG) is a feature descriptor explained by Dalal and Triggs [33]. This feature with SVM is used for Visual Place Recognition. HOG is calculated by first calculating gradient images and making histograms using direction range as bin and magnitude as frequency. Figure 6 shows the process.

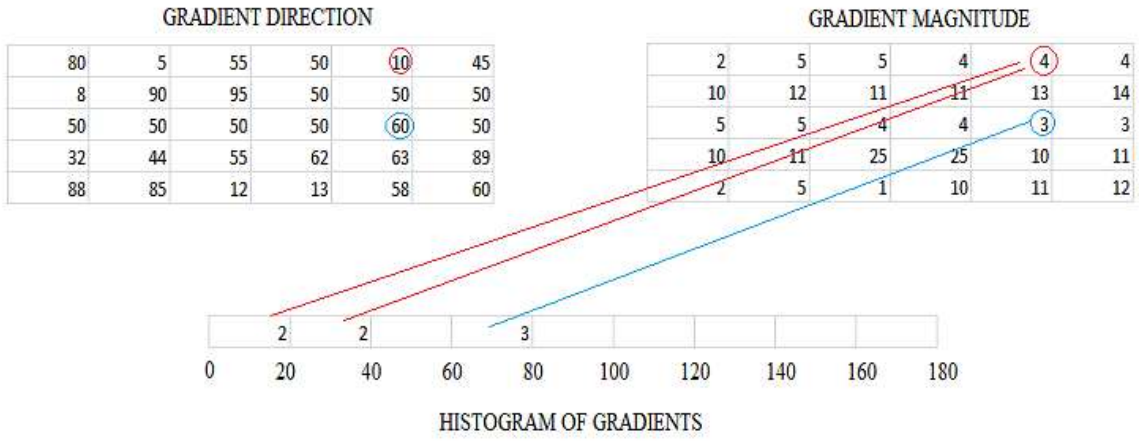

Figure. 6: Histogram of Gradients: Calculation of each pixel is weighted that is magnitude of 10 degree is shared equally by 0 and 20 therefore the magnitude in HoG is increased in both 0 and 20 bins.

4 METHODOLOGY

In order to carry out localization, a map needs to be known, which for this problem corresponds to a topological map of the environment. Difficulty or error arises when the map is not accurate. Given an input image we predict the starting point using Visual Place Recognition. Two methodologies are used. In the first, we separately applied visual odometry and experience based localization; and compared the two methodologies with each other. In the second methodology, we applied a fusion of experience based localization and visual odometry. Figure 7 shows a Kinect sensor attached on a Firebird 6 robot.

Figure. 7: Robot Platform Used for Experiment: The Kinect has been attached to a Firebird 6 robot.

4.1 Visual Place Recognition

The problem of visual place recognition is to find out which 'place' or geographical location the robot is in by looking at the surroundings. So as to localize we first need to find where currently the robot is and what is its starting position. Consider the situation where a human wakes up and the first thing he/she does is to observe surroundings to localize himself/herself. We have also used the same approach. Finding the starting position is a challenging task as the starting position result affects the results of visual odometry.

Let a robotics area have n places called ρ_1, ρ_2, $\rho_3 \ldots \rho_n$ while the robot only has a monocular camera that takes an image (I). Visual Place Recognition is the Classifier $C_1(I)= \rho_i$ that takes an image as input and returns the place label as an output. Because the image may be very large in size, we have used HOG descriptors as a feature to train a SVM classifier [34], making the classifier $C_2(\text{HOG}(I))=\rho_i$

Knowing the place, one needs to know the starting positon inside the place or room. The problem is to find $(x_0,y_0,\theta_0)|(x_0,y_0) \in \rho_i$, given the sensory percepts of the robot. A typical way to do so would be to make a database $D(\rho_i)=\{I, (x,y,\theta)\}$ where the image I is the input and (x,y,θ) is the labelled output. A classifier $C_3(I)=(x_0,y_0,\theta_0)|(x_0,y_0) \in \rho_i$ can then be made to solve the problem of start point computation. The dataset D may be made recording all possible (x,y,θ) combinations. However the assumption here is operation in a wide open hall or room wherein the robot motion is not restricted on any axis, unlike roads wherein there is a very small lateral dispersion. In such contexts sometimes by a single image we may not get adequate features to compare and that may result in poor results. Say the image to be compared is very near to that of a plain wall that has no features. Sometimes even we cannot identify the correct location by just looking at a wall. So we just turn or rotate a little in order to correctly identify the starting location. Using the same motivation we take multiple images at an angular dispersion from the same (x_0,y_0) location. Taking multiple images from the same coordinate can get sufficient features and increase accuracy.

To find the starting point we take 6 images at a difference of $\pi/3$ radians each. Each of the image is searched in the desired room database recorded earlier and 1-Nearest Neighbor classifier is used to find the resulting image. Image matching is done by finding the ORB (Oriented Fast and Rotated Brief) descriptors [35] and using FLANN [36] library to find approximate Nearest Neighbors. Let $C_4(\text{ORB}(I_j))= (x_j,y_j,\theta_j)|(x_j,y_j) \in \rho_i$, where j is the image number taken in intervals of $\pi/3$ radians. All classifications (x_j,y_j,θ_j) corresponding to each of the image is recorded. The final starting position is taken as the one that is the most centrally placed, or the one that minimizes the metric given by equation (5). In equation (5) d() is the distance function. In short, we find the most centrally located point among the six points.

$$(x_0,y_0,\theta_0)=(x_j,y_j,\theta_j),\ j=\arg\min_k \sum_{l\neq k} d([x_k,y_k]^T,[x_l,y_l]^T) \quad (5)$$

The choice of the classifier is again due to the nature of the problem. In order to create a dataset we visited all the places once and recorded various poses at every coordinates creating a single experience of visiting every possible orientation of the robot. Unlike literature, the map is wide open and creation of a dataset requires a huge effort due to multitude of positions possible for the robot. Every image is mapped to a single coordinate and hence the classifier.

4.2 Experience Based Localization

For experience based localization, the map first needs to be converted into a topological map. The underlying map represents wide indoor space, which does not naturally represent a topological map. However to embed a graph into the same, all possible locations with some uniform sampling are taken as the vertices, $V = \{(x,y,\theta)\}$. Two vertices are connected to each other by an edge if they are neighbors of each other using a neighborhood function δ. $(v_i, \delta(v_j)) \in E$. It was easy to create a topological map. The only conditions we need to keep in mind was that the robot/machine can go anywhere except on reaching the limits in X and Y directions. There were points which were inaccessible so complete black images were mapped to these points meaning it is a wall or obstacle.

The initial position (x_0, y_0, θ_0) is known from the Visual Place Recognition algorithm. Assume that the position at any time t is known to be (x_t, y_t, θ_t) which is covered by the vertex v_t of the sampled topological graph. The robot should be able to localize itself as it moves to the next position or as it takes a step. As per modelling, the maximum that the robot can move within a time iteration is less than the vertices being covered by the neighborhood function, that is $d([x_t, y_t, \theta_t]^T,[x_{t+1}, y_{t+1}, \theta_{t+1}]^T) < \max_{vi \in \delta(vt)}(d(v_i,v_t))$. Hence, knowing the positon at time t to be in the vertex v_t, the position at time t+1 may be at either of the vertices in $\delta(v_t)$. We only need to check for this range of coordinates. Topological map for an obstacle-free region is shown in figure 8(a). For simplicity Figure 8(a) does not show the orientation axis. Figure 8(b) shows the coordinate axis.

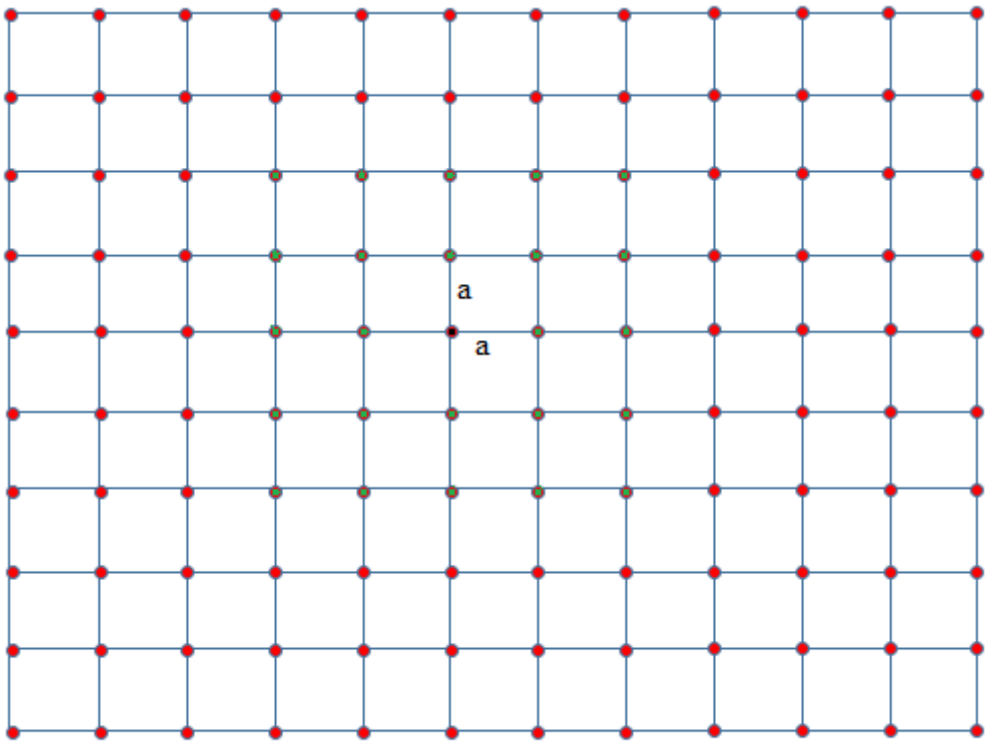

Figure. 8(a): Grid Map: Suppose black dot is previously visited coordinate then its neighbors are the green dots and the black dot. 1 unit in X is equal to 'a'. 1 unit in Y is equal to 'a' which is difference in distance at which we have recorded images.

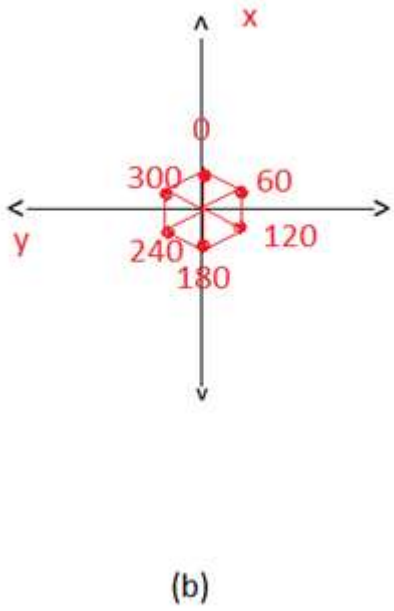

Figure. 8(b): Coordinate axis with Robot Orientations: All theta values around a single coordinate are displayed. 'Theta' is the rotation around Z-axis. Z-axis is normal to X-Y plane.

The choice of the neighborhood function $\delta(v_t)$ is such that any two vertices at a geometric distance of $2\sqrt{2}a$ or less will be connected, irrespective of the orientation, that is $\delta(v_t<x_t,y_t,\theta_t>)=\{v_i<x_i,y_i,\theta_i>:d([x_t\ y_t]^T, [x_i\ y_i]^T)<= 2\sqrt{2}a\ \}$. Here *d* is the Euclidian distance function. From figure 8(a) we see that there are 25 coordinates in the XY place. Each coordinate

has 6 images mapped to it in the orientation plane, so there are 150 images that needs to be matched with the current online image. The number will be larger if there are more than one image per location. A localizer takes the online image and compares it with these 150 images from experience to give the best possible current location. Let $D(x,y,\theta)$ denote the image corresponding to an output entry (x,y,θ) in the database. Since (x,y,θ) is discretized in the dataset, the function represents a hash table with (x,y,θ) as the key and the image as the value. Algorithm 2 shows how the algorithm is carried through a pseudo code. In the algorithm the term position and vertex are used interchangeably even though the former is a continuous quantity and the latter is a discrete quantity because all operations returning position are only capable of returning discrete outputs corresponding to which a vertex always exists. Figure 9 summarizes the algorithm of Experience based localization with the use of a flowchart.

Algorithm 2: Localization

1. $\tau(0) \leftarrow (x_0,y_0,\theta_0)$, $t \leftarrow 0$, $v_0 \leftarrow (x_0,y_0,\theta_0)$
2. while (3Dsensor is 'ON'):
3. I(t)← 3DSensor_Capture()
4. Compute $\delta(v_t)$
5. $v_{t+1} \leftarrow v_j$, $j=\arg\min_{j \in \delta(vt)}(d(ORB(I(t)), ORB(D(v_j))))$
6. $\tau(t+1) \leftarrow v_{t+1}$
7. $t \leftarrow t+1$

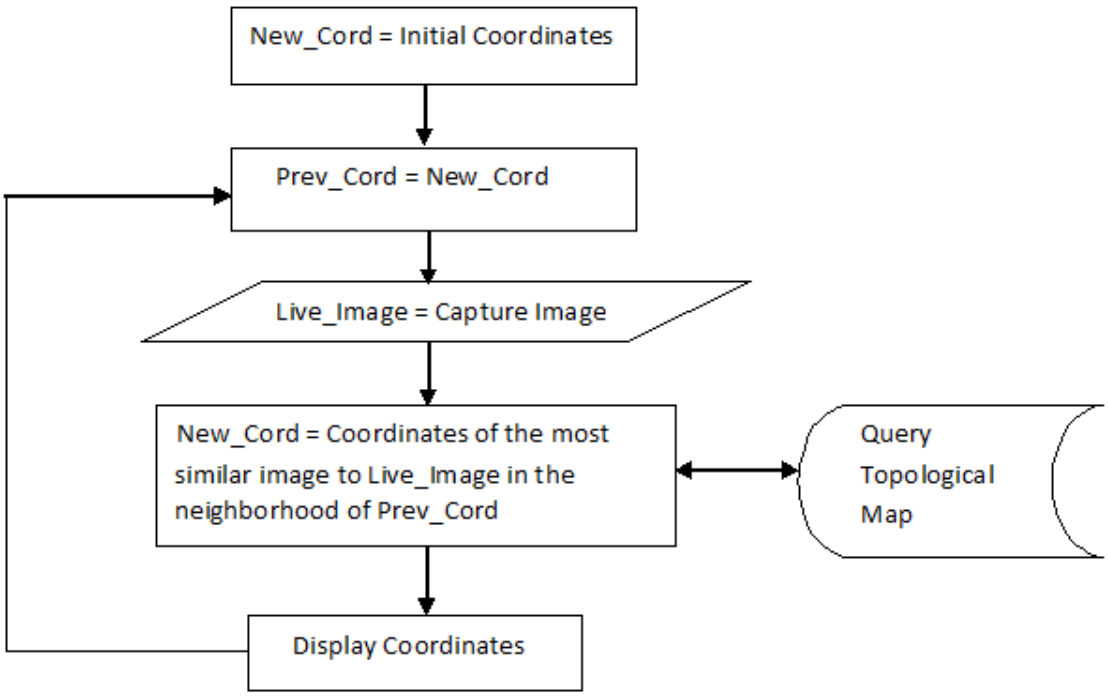

Figure. 9: Flowchart of Experience Based Localization: This figure shows the process of Experience Based Localization for robot pose estimation.

4.3 Combined Approach

The previous sections presented the use of experience based localization and visual odometry independently. In this section we find the best matched coordinate of the online image using experience based localization and then we use the location with the online image to get better results. Visual odometry is a very good mechanism for localization, however the errors keep getting accumulated that may result in a very poor localization over the long run. However, the output is continuous in nature. The experience based localization finds the location from a dataset using live images and hence the errors do not easily accumulate. However the output is limited to the stored experiences and in this case discrete in nature. By fusing the two techniques we get rid of the additive nature of the errors of localization using experience based

localization for macro localization and get a continuous output based on the visual odometry for micro localization. Further, the database of experience based localization can also aid in visual odometry by providing better images to be used to compute the transformation. The betterment of the approach is illustrated in Figure 10. If the original location of the robot was the one indicated by the red symbol, the current location by green and the one obtained by using experience based localization by blue, then computing visual odometry between the green and blue poses gives a better result rather than computing visual odometry between the green and red poses. Thus, first we detect the robot's position using experience based localization and after that in order to find out the live location of the robot we compute the pose of the robot from applying visual odometry between the live image and the image obtained from experience based localization.

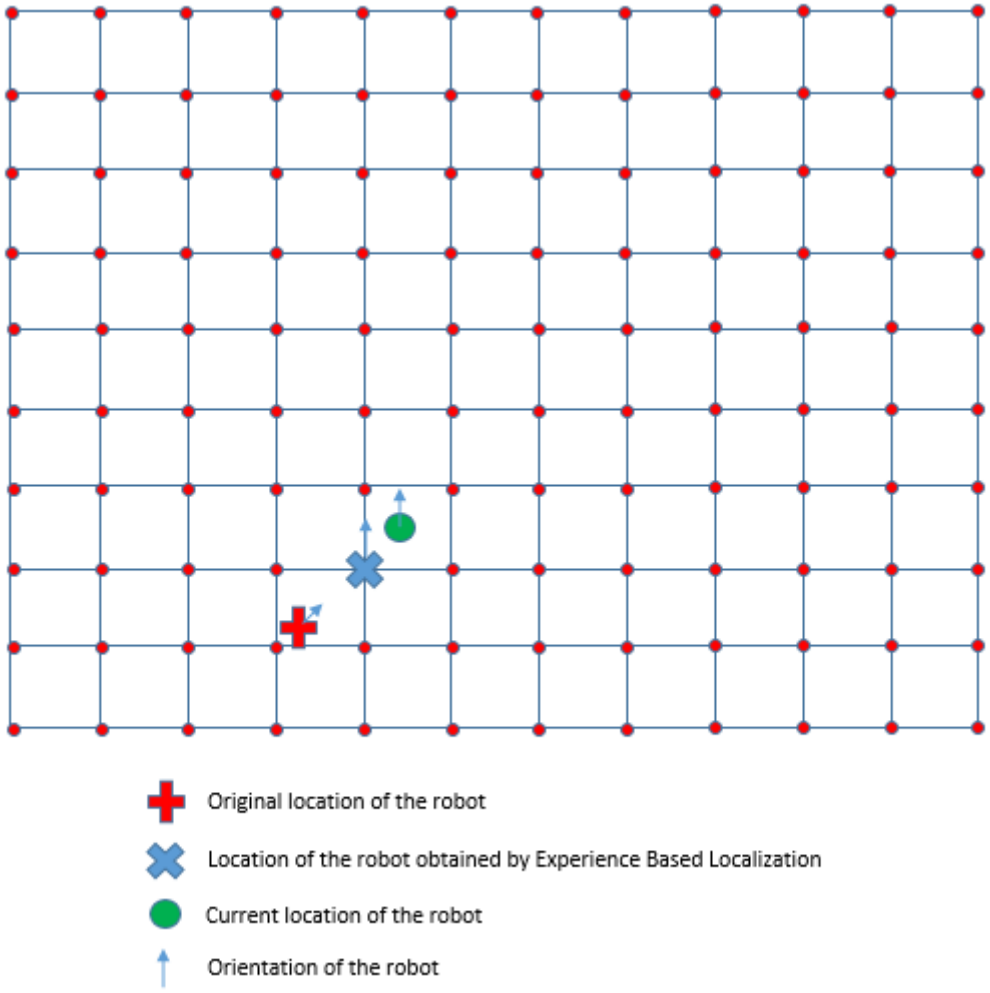

Figure. 10: The different orientations of the robot: The arrowhead part indicates the direction of the robot.

Figure 11 summarizes the algorithm discussed through a flowchart. The hybrid algorithm is given by Algorithm 3. Visual Odometry takes two images and computes the distance between them. Algorithm 3 is a slight modification of Algorithm 2.

Algorithm 3: Combined Approach

1. $\tau(0) \leftarrow (x_0, y_0, \theta_0)$, $t \leftarrow 0$, $v_0 \leftarrow (x_0, y_0, \theta_0)$
2. while (3DSensor is 'ON'):
3. I(t)← 3DSensor_Capture()
4. Compute $\delta(v_t)$
5. $v_{t+1} \leftarrow v_j$, $j=\arg\min_{j \in \delta(vt)}(d(ORB(I(t)), ORB(D(v_j))))$
6. $\tau' \leftarrow v_{t+1}$
7. $\Delta \leftarrow$ Visual_Odometry_Distance(I(t), $D(v_{t+1})$)
8. $\tau(t+1) \leftarrow \Delta \times \tau'$
9. $t \leftarrow t+1$

The function $D(v_{t+1})$ returns the image corresponding to the location given by experience based localization. The image is fed as input to the function Visual_Odometry_Distance which returns the transformation between the live image and image obtained by experience based localization.

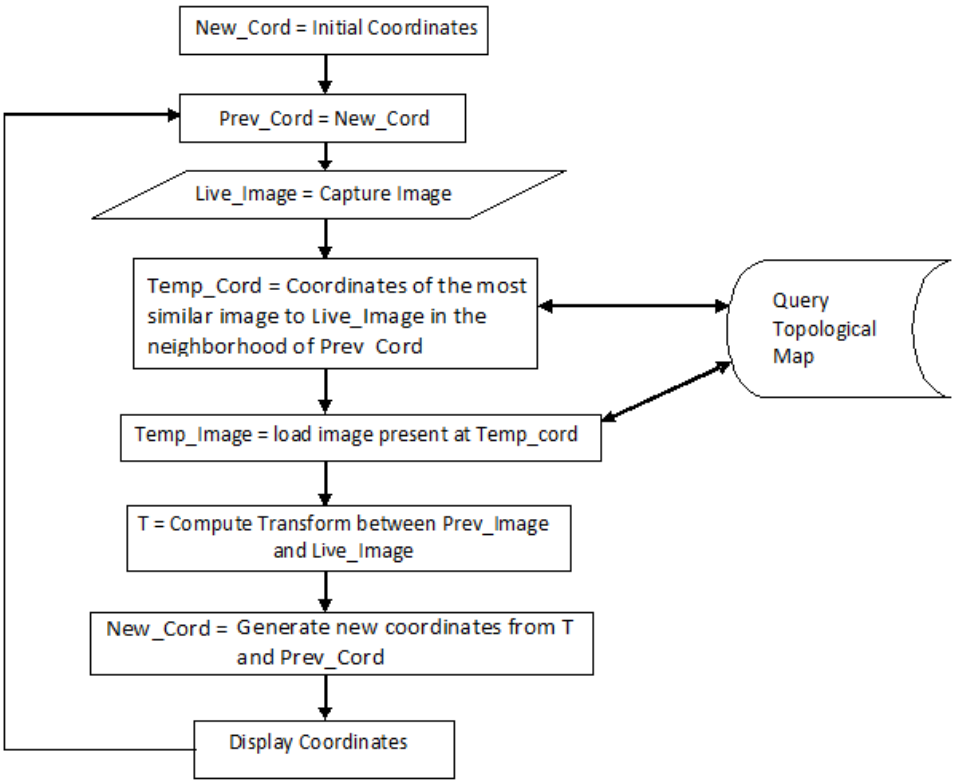

Figure. 11: Flowchart of combined approach: The results of experience based localization gives discrete coordinates. These coordinates are mapped to some images in dataset. This image and online image are inputs to visual odometry which gives the final output pose of the robot.

5 DATASET

Since no standard dataset was suited for our approach, the data set had to be exclusively created by us. The problem being solved was specific in nature which was to operate a robot in the robotics arena of the institute and hence only dataset of that place was taken. As per policies only that particular location could be physically hand-annotated (causing physical changes to the area) and the robots could be operated only in the same region. That said, the robotics arena of the institute is itself very large and diverse. The area in question was exhaustively covered by considering all positions and poses and we did everything possible in order to record a rich dataset out of the resources available in the need of

the broader project. Dataset consists of 640 × 480 pixel images recorded by a Kinect sensor at a height of about 107 centimeters above the floor. The dataset has been recorded on three different rooms namely Classroom, Lab and Office. Floor of each of the room is a two dimensional surface (XY) not having any variations around Z-axis (normal to the floor). On each coordinate (x, y) six images at an angle $\pi/3$ clockwise have been recorded. Classroom has 18×20 coordinates consisting of total 2160 monocular images. Lab has 16×30 coordinates consisting of total 2880 monocular images. Office has 8×10 coordinates consisting of total 400 monocular images (in the office dataset we recorded on 72 degrees). We have also depth images recorded in the manner above. The lab dataset was not recorded on a single night. Frequently changing positions of books, bottles and some minor changes occurred during recording the dataset. Figure 12 shows an example of recorded dataset for coordinate (11, 5) in lab dataset.

Figure. 12: Dataset Images: (a), (b), (c), (d), (e), (f) shows images at 0 degree, 60 degree, 120 degree, 180 degree, 240 degree and 300 degree respectively at coordinate (11, 5) from lab dataset.

6 RESULT

All three dataset have some minor changes. We have used Root Mean Square (RMS) as our evaluation metric as shown in equation (6).

$$rms = \sqrt[2]{\frac{\sum_{i=1}^{N} error_i^2}{N}} \quad (6)$$

Another metric used for evaluation is the misclassification accuracy, wherein a result is termed as correct if the error is less than the resolution of recording. The development was done in C++ and python programming language. We used a system with Ubuntu 16.04 and 8 GB RAM.

6.1 Result of Experience Based Localization

Table 1 shows results of Experience based localization. *a* is the length of coordinate axis or the resolution by which the dataset was recorded. No two images exists at a distance smaller than *a*. For classroom and lab *a* is 38 centimeters and for office *a* is 50 centimeters. All results are reported relative to this distance to make the results invariant of the recording resolution. hit(1*a*) denotes the probability of correctly classifying gird within *a* distance, that is the resolution of recording the dataset. For office dataset out of 33 points 8 points are correctly classified within *a* distance. hit(2*a*) denotes the probability of correctly classifying gird within 2*a* distance. For office dataset out of 33 points 26 points are correctly classified within 2*a* distance.

We see that RMS error of lab dataset is more as compared to other datasets (rooms) as in lab between recording the dataset and the test runs there were some minor changes as positions of mouse, books and keyboard were not fixed always. There were some computer screens that were 'on' in initial recording of dataset and 'off' during the later recordings of the dataset and vice versa. We tried to keep the position of chairs same as before during the whole recording of the dataset.

	rms(a)	hit(1a)	hit(2a)
Office	1.31152	8/33	26/33
Classroom	1.34602	14/33	28/33
Lab	1.39014	15/20	17/20

Table. 1: Results of Experience Based Localization: Column contains name of Datasets (Room) and Rows contain result metrics.

6.2 Results of Visual Odometry

Table 2 shows results of visual odometry, rms (in meters) shows root mean square distance calculated in meters and RMS (*a*) shows root mean square distance calculated in terms of side *a*. Table 2 column contains points or steps. The first row shows the overall performance of the test runs while the second row shows the performance of first ten steps taken by the robot or say first ten points where the image is taken and so on. We see that RMS is relatively low for the first ten points and keeps increasing thereafter due to accumulation of errors. Thus for a long run we need to reinitialize the visual odometry original location with the pose obtained from experience based localization so as to restrict the cumulative error from becoming too large.

	Rms (in meters)	rms(a)
Overall	0.5407	1.421
first 10	0.3235	0.842
11 to 20	0.6051	1.578
after 20	0.6213	1.631

Table. 2: Results of Visual Odometry: Column shows the points range and Rows contain result metrics.

6.3 Results with Dynamic Obstacles

We tested both above algorithms for dynamic obstacles but the results were 4.23a and 2.92a respectively. Visual Odometry results show twice as much as error in normal case. The reason is that it may not get more similar features in consecutive frames because of dynamic obstacles that leads to poor results than that in case of normal conditions. But 2.92a is approximately 1.1 meters which means it is still able to localize itself over a rough estimation.

6.4 Results of Start Point

RMS error of start point by visual place recognition is 0.81a. It is low because we are taking six images each at a $\pi/3$ radians difference rotated clockwise and the finding the most centrally located point among them. It has to be low because it is used to initialize the odometry module and its accuracy is very important in ensuring all the following algorithms works accurately.

6.5 Results of Visual Place Recognition

There are 300 training images from dataset. There are 93 testing images. All testing images are different from training images. The accuracy is been calculated using precision (p) and recall(r). Accuracy is given by equation (7), precision is given by equation 7 and recall is calculated by equation (9). Table 3 summarizes the results for all three classes (Classroom, Lab and Office).

$$accuracy = \frac{2pr}{(p+r)} \quad (7)$$

$$p = \frac{truepositives}{truepositives+falsepositives} \quad (8)$$

$$r = \frac{truepositives}{truepositives+falsenegatives} \quad (9)$$

	TP	FP	FN	Precision	Recall	Accuracy
Classroom	23	9	3	0.71	0.88	0.78
Lab	25	0	10	1	0.71	0.83
Office	32	4	0	0.89	1	0.94

Table. 3: Results of Visual Place Recognition. TP, FP and FN are true positives, false positives and false negatives.

6.6 Results of the combined approach

RMS error of using experience based localization with visual odometry is 2.32a. The error is a little large because visual odometry requires good initialization but the problem is that we are not able to provide good initial angle by localization. The poor resolution of recording the dataset in terms of angle is the limiting factor. Nevertheless, as apparent otherwise from the results, the combined approach is able to give realistic location estimates by using the combination of both methodologies. There was no large growth in error as was dominant in visual odometry, while the system could interpolate between points and was not restricted to giving outputs between the points where experience was recorded. Currently it was possible to spend significant time in recording an experience dataset with a very high resolution. In a

more realistic setting, the experience dataset will be a lot coarser and the benefits of the combined approach will be much larger.

7 CONCLUSION

For localization, Visual Odometry works well for small distances that are suitable for indoor localization but the problem is that it needs to be initialized accurately in a prebuilt map and the errors may accumulate with time. The paper used Visual Place Recognition for setting a starting point. A naïve implementation of Visual Place Recognition works gives poor performance for wide-open scenarios as the results are based on a single image. The paper proposes the use of multiple images at different orientations for improved performance. Experience based localization performs well in static maps with non-cumulative errors to a very large extent, while visual odometry performs well in dynamic maps and gives continuous outputs to localization. The paper proposed the fusion of the two methodologies. The resultant algorithm performed well, while the results could have been even better if more accurate orientation was predicted from experience based localization using a better resolution of orientation while recording the dataset.

The biggest limitation of this project is that currently we are recording the dataset manually. In the future we can design a robot which when left in a room can make the entire dataset. The current algorithm is also highly susceptible to changes in the illumination of the room and for large changes in the environment we have to recreate the database. We can in the future modify the algorithm such that changes in the environment are incrementally updated in the original database. If we take the camera at a low height the results for both visual odometry and experience based localization are very poor as a lot of ground features are detected which are difficult to match. The dataset cannot be shared across the robot as different robots have different heights. It should be possible to make adaptations to changing robots. These all limitations can be taken care of or the effect be mitigated by making changes to the algorithm in the future.

REFERENCES

[1] D. G. Lowe, Distinctive Image Feature from Scale-Invariant Keypoints, International Journal of Computer Vision, 60(2): 91–110, 2004.

[2] S. Lowry, N. Sunderhauf, P. Newman, J.L. Leonard, D. Cox, P. Corke, M.J. Milford, Visual Place Recognition: A Survey, IEEE Transactions on Robotics 32(1): 1-19, 2016.

[3] C. Linegar, W. Churchill, and P. Newman, Work Smart, Not Hard: Recalling Relevant Experiences for Vast-Scale but Time-Constrained Localisation, In: 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, 2015, pp. 90-97.

[4] A. Napier, G. Sibley, and P. Newman, Real-Time Bounded-Error Pose Estimation for Road Vehicles Using Vision. In 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, Portugal, 2010, pp. 1141–1146.

[5] M. Fiala and A. Ufkes, Visual Odometry Using 3-Dimensional Video Input. In 2011 Canadian Conference on Computer and Robot Vision, St. Johns, NL, 2011, pp. 86-93.

[6] A. Pronobis, B. Caputo, P. Jensfelt and H. I. Christensen, A Discriminative Approach to Robust Visual Place Recognition. In 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, 2006, pp. 3829-3836.

[7] J. Shi, C. Tomasi, Good Features to Track in IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, 1994, pp. 593-600.

[8] C. Harris, M. Stephens, A Combined Corner and Edge Detector, Plessey Research Roke Manor, United Kingdom, 1988.

[10] A. Pronobis, J. Luo, B. Caputo, The More you Learn, the Less you Store: Memory-controlled Incremental SVM for Visual Place Recognition. Image and Vision Computing 28(7): 1080-1097, 2010.

[11] A. Pronobis, B. Caputo, Confidence-based cue integration for visual place recognition. In 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, 2007, pp. 2394-2401.

[12] D. F. Llorca, R. Arroyo and M. A. Sotelo, Vehicle logo recognition in traffic images using HOG features and SVM. In 16th International IEEE Conference on Intelligent Transportation Systems, The Hague, 2013, pp. 2229-2234.

[13] M. Shimosaka, O. Saisho, T. Sunakawa, H. Koyasu, K. Maeda, R. Kawajiri: ZigBee based wireless indoor localization with sensor placement optimization towards practical home sensing. Advanced Robotics 30(5): 315-325, 2016.

[14] R. Kala, On-Road Intelligent Vehicles: Motion Planning for Intelligent Transportation Systems, Elsevier, Waltham, MA, 2016.

[15] R. Kadota, H. Sugano, M. Hiromoto, H. Ochi, R. Miyamoto and Y. Nakamura, Hardware Architecture for HOG Feature Extraction. In 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kyoto, 2009, pp. 1330-1333.

[16] P. Henry, M. Krainin, E. Herbst, X. Ren, D. Fox, RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. International Journal of Robotics Research 31(5): 647-663, 2012.

[17] I. Dryanovski, R. G. Valenti and Jizhong Xiao, Fast visual odometry and mapping from RGB-D data. In 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, 2013, pp. 2305-2310.
[18] K. Irie, T. Yoshida, M. Tomono, Outdoor Localization Using Stereo Vision under Various Illumination Conditions, Advanced Robotics 26(3-4): 327-348, 2012.
[19] V. Kaundal, P. Sharma & M. Prateek , Wireless Sensor Node Localization based on LNSM and Hybrid TLBO- Unilateral technique for Outdoor Location. International Journal of Electronics and Telecommunications, 2017, 63(4), pp. 389-397. Retrieved 14 Dec. 2017.
[20] R. Rathnam, & A. Birk, A Distributed Algorithm for Cooperative 3D Exploration under Communication Constraints. Paladyn, Journal of Behavioral Robotics, 2013, 4(4), pp. 223-232. Retrieved 14 Dec. 2017
[21] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile, Collaborative Filtering Bandits. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR '16), 2016, ACM, New York, NY, USA, 539-548.
[22] Shuai Li, The Art of Clustering Bandits, Università degli Studi dell'Insubria, Dissertation, 2016
[23] Shuai Li, Purushottam Kar, Context-Aware Bandits, CoRR abs/1510.03164, 2017
[24] Nathan Korda, Balázs Szörényi, and Shuai Li, Distributed clustering of linear bandits in peer to peer networks, In Proceedings of the 33rd International Conference on International Conference on Machine Learning, 2016, Volume 48 (ICML'16),1301-1309
[25] Purushottam Kar, Shuai Li, Harikrishna Narasimhan, Sanjay Chawla, and Fabrizio Sebastiani, Online Optimization Methods for the Quantification Problem, In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), 2016, ACM, New York, NY, USA, 1625-1634.
[26] Jaehyun Yoo, Karl H. Johansson, Semi-Supervised Learning for Mobile Robot Localization using Wireless Signal Strengths, International conference on indoor positioning and indoor navigation (IPIN), 2017
[27] Soonyong Park and Kyung Shik Roh, Coarse-to-Fine Localization for a Mobile Robot Based on Place Learning With a 2-D Range Scan, IEEE Transactions on Robotics, June 2016, Volume 32, no. 3, pp. 528-544
[28] Guoyu Lu, Yan Yan, Li Ren, Philip Saponaro, Nicu Sebe, Chandra Kambhamettu, Where am I in the dark: Exploring active transfer learning on the use of indoor localization based on thermal imaging, Neurocomputing, Volume 173, Part 1, 2016, Pages 83-92
[29] Wera Winterhalter, Freya Fleckenstein, Bastian Steder, Luciano Spinello and Wolfram Burgard, Accurate indoor localization for RGB-D smartphones and tablets given 2D floor plans, 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, 2015, pp. 3138-3143.
[30] J. Luo, A. Pronobis, B. Caputo, P. Jensfelt, Incremental learning for place recognition in dynamic environments. In 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, 2007, pp. 721-728.
[31] L. Quan, Z.D. Lan. Linear N-Point Camera Pose Determination. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21 (8): 774–780, 1999.
[32] S. Jia, K. Wang, X. Li, T. Xu, A Novel Improved Probability-Guided RANSAC Algorithm for Robot 3D Map Building, Journal of Sensors, vol. 2016, Article ID 3243842, 18 pages, 2016.
[33] N. Dalal, B. Triggs. Histograms of Oriented Gradients for Human Detection. In International Conference on Computer Vision & Pattern Recognition, 2005, San Diego, pp.886-893.
[34] C.W. Hsu and C.J. Lin, A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks, 13(2): 415-425, 2002.
[35] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, ORB: An efficient alternative to SIFT or SURF. In International Conference on Computer Vision, Barcelona, 2011, pp. 2564-2571.
[36] M. Muja, D. G. Lowe, Scalable Nearest Neighbor Algorithms for High Dimensional Data. IEEE Transactions on Pattern Analysis and Machine Intelligence 36(11): 2227-2240, 2014.