Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Stereo Vision based Localization of a Robot using Partial Depth Estimation and Particle Filter

Selvaraj Prabu * **Guoqiang Hu** **

* *School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, (e-mail: prabu003@e.ntu.edu.sg)*
** *School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, (e-mail: gqhu@ntu.edu.sg)*

Abstract: In this paper, we present a novel method to localize a robot throughout its navigation path using a stereo camera. We first estimate the 3D locations of the feature points in the images using the partial depth estimation technique and compute the motion estimate among the set of subsequent images. Then those motion estimates are filtered using a particle filter method in order to minimize the error in the motion estimates and reliably localize the moving robot. In the partial depth estimation technique, we determine the disparity of the feature points between the stereo images and estimate the depth of the feature points. The main novelty of the paper is the formulation of a vision based localization algorithm which combines the partial depth estimation and particle filter techniques. Experiments were conducted on mobile robots and the obtained localization results are analysed.

Keywords: Stereo vision, Mobile robots, Localization, 3D transformation, SVD, Particle Filter.

1. INTRODUCTION

Vision based automation methods are interesting topics in robotics since the cost of cameras have become cheap. Among these topics, localizing a robot using pure vision based method is one of the most widely researched and also a challenging topic. The term *localization* is the process of finding the current location of a robot. Robot localization is a key technology for many robotic tasks such as obstacle avoidance, path planning, indoor and outdoor exploration, etc. The use of GPS (Global Positioning System) is the most common way of localizing a robot in outdoor environments. Other sensors such as Laser, IMUs (Inertial Measurement Units), etc are also used to achieve localization. A camera represents the environment with millions of pixels. Each image gives us abundant information about the environment using various colors, gradients, shapes, etc. These significant image properties make the camera as one of the most powerful and cost effective sensors.

1.1 Objective and Challenges

The main objective of this paper is to determine the problems involved in localizing a robot using vision based methods and then propose a novel method to localize a robot robustly and analyse the accuracy, repeatability and drift rate in the proposed method. The main challenges involved in vision based robot localization include estimation of camera parameters of the stereo camera, identifying feature points and disparity between the stereo images tracking the feature points and determining the pose estimates using the tracked feature points and filtering the pose estimates using particle filter method.

1.2 Contributions

The main contribution of this paper is the process of combining partial depth estimation with the particle filtering method. Partial depth estimation is the process of estimating the depth of only the feature points present in the image whereas computing the depth of all the pixels in the image is full depth estimation. Particle filtering ensures that noisy motion estimates are filtered out during the estimation of location of the robot and it also suppresses the uncertainty present in the pose estimates.

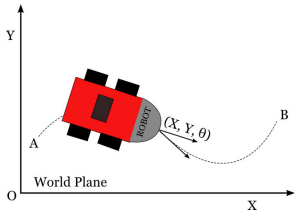
1.3 Related Work

Robot localization is important for many higher level robot tasks such as motion planning, autonomous navigation, etc. Over the past two decades, many researchers all over the world have been working on the vision based robot localization problem, in order to unlock the full potential of the camera sensor. In the late 1990s, the Bouguet and Perona [1995] and Fox et al. [1999] stated the importance of robot localization. Fox et al. [1999] proposed a promising localization algorithm based on Monte-Carlo technique. Later, Fox et al. proposed a variant of the previously proposed method in [Thrun et al., 1999] and successfully implemented the new method in a museum robot. Se et al. [2005] used image features as the main parameter for robot odometry and developed vision based SLAM (Simultaneous Localization And Mapping) system for a robot. Yuen et al. [2005] proposed a vision based localization method using a landmark matching technique. Maimone et al. [2007] discusses the entire algorithm of how visual odometry was implemented on the Mars Exploration Rovers (MERs). This application clearly highlighted the importance of vision based localization and navigation system Recently, Kitt et al. [2010] proposed another visual odometry algorithm based on RANSAC outlier rejection technique. All the above mentioned research works provided a fundamental basis for the localization algorithm developed in this paper.

2. BACKGROUND PRELIMINARIES

Some of the basic preliminary topics that are required for implementing the method proposed in this paper are,

(a) Stereo camera calibration using multiple chessboard poses.



(b) Schematic for robot localization.

Fig. 1. Preliminaries

2.1 Stereo Geometry

An image obtained from a camera is a projection of 3D world to 2D pixel coordinates. So the depth information is lost in the camera projection process. *Stereo Geometry* is the method used to recover the lost depth information using the basic geometry between the left and right cameras.

2.2 Stereo Calibration

In order to recover the 3D coordinates from the 2D coordinates, we need to determine those camera parameters using calibration process. For the purpose of localizing the robot, we used the camera calibration methods as explained in [Bouguet, 2000]. Fig. 1(a) shows few chessboard poses used in the calibration process of the stereo camera.

2.3 Feature Extraction and Matching

The total number of pixels in the image (with resolution 640×480) is 307200 pixels. It is evident that a normal camera gives abundant information for manipulation. Out of that abundant information, only a very few information in the image are meaningful and unique called features. Features are broadly classified into two types: *feature points* and *feature templates or patches*. Feature points are very unique single pixels such that they are very different from the surrounding neighbourhood pixels while templates are *unique group of pixels* that are different from the neighbouring *group of pixels*. So, point features are easy and computationally efficient when compared with the template feature matching. Among the various existing feature points, we chose SIFT (Scale Invariant Feature Transform) features as the feature point detector in our localization. SIFT features are more robust than any other features, since they are scale and rotation invariant as well as partially invariant to illumination and affine transformation.

3. PROBLEM FORMULATION

The problem of localizing a robot is shown in Fig. 1(b). Consider a robot moving from point A to point B in an environment. In order to map the robot in that particular environment, it is necessary to trace the path of the robot by localizing the robot. The process of determining localization information using pure vision based methods is susceptible to errors due to motion blur, lighting conditions, region similarity, etc. Thus, this paper provides a robust algorithm to solve this localization problem of the robot. The Odometry information required for the successful localization of the robot is given by (x, y, θ), where (x, y) and θ gives the position and heading angle of the robot at any instant, respectively.

4. LOCALIZATION ALGORITHM

The basic steps used by the localization algorithm of this paper are as follows.

i) Compute SIFT features and extract the location of the SIFT features from the stereo images.
ii) Estimate Disparity of the features between the left and right images.
iii) Compute 3D world coordinates of each features from the estimated disparity of the features.
iv) Perform the steps 1 to 3 again for the subsequent frame (Say n^{th} interval frame). Decide the value of n based on the motion velocity of the robot.
v) Estimate the 3D transformation (rotation and translation) between the initial frame and the subsequent frame.
vi) Steps 1 to 5 are repeated for a set of frame pairs, i.e., transformation is computed for a set of frame pairs such as between i^{th} and $(i+n)^{th}$, $(i+1)^{th}$ and $(i+1+n)^{th}$,..... $(i+N)^{th}$ and $(i+N+n)^{th}$ frame pairs etc.
vii) Compute the mean translation and rotation among all frame pairs and also compute the global location of all the SIFT features with respect to the origin (starting point of the robot motion).
viii) Finally, use particle filter to eliminate the noise in the localization estimates.

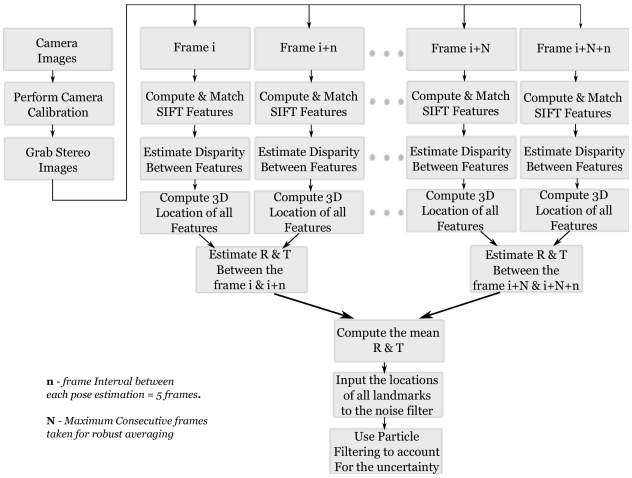


Fig. 2. Flowchart of Localization Algorithm using Partial Depth Estimation and Particle Filter. Here we show the flow only for $[i,\ i+n]$ and $[i+N,\ i+N+n]$ frame pairs.

The Fig. 2 explains visually the steps of the localization algorithm with N frame pairs and frame interval of n for computing the mean transformation. The following sections describes the detailed mathematical formulation of all the steps explained in Fig. 2.

4.1 Feature Extraction and Matching

Assuming the stereo camera is calibrated, the images are then grabbed from the stereo camera. Using the calibration parameters of the camera, undistortion and rectification of the stereo image pairs are also performed. The calibrated images are then used for the feature extraction process. The features used for localization must be robust and consistent. This is the reason for using SIFT features in the localization process. SIFT features are extracted from the images based on the mathematical technique proposed in [Lowe, 2004]. Here, the SIFT features are

extracted from the left and right stereo image pairs. Let n and m be the total number of features extracted from the left and right images respectively and the extracted features from the stereo images are matched using a correspondence algorithm. FLANN (Fast Library for Approximate Nearest Neighbor) matching algorithm is used for matching the features of left image with the features of the right image as proposed in [Muja and Lowe, 2009]. Only j number of features are matched between the stereo images. Thus, the set of matching features of the stereo images are denoted as,

$$F_i^l = \{f_{i1}^l, f_{i2}^l, ...f_{ij}^l\} \quad (1)$$

$$F_i^r = \{f_{i1}^r, f_{i2}^r, ...f_{ij}^r\} \quad (2)$$

where and right image respectively. Here the correspondence of f_{i1}^l is f_{i1}^r. Similar correspondence applies to all the j features in the feature set. Each feature point is represented as,

$$f_{ij} = (x_{f_{ij}^l}, y_{f_{ij}^l}), \forall\, f_{ij} \in F_i^l, F_i^r, \quad (3)$$

where $(x_{f_{ij}^l}, y_{f_{ij}^l}) \in \mathbb{R}^2$ is the x and y coordinates of the feature point respectively.

4.2 Feature Sorting

Once the features have been extracted, the feature points are sorted either according to their 2D x coordinate or y coordinate. The algorithm used for the purpose of sorting the features is Quick sort, proposed in [Knuth, 1998]. This algorithm is chosen since its average computation complexity is $O(n \log n)$. This sorting step will ease the process of searching the correspondence between the two successive frames in the subsequent steps of the localization algorithm. Both the left and right image features are sorted along 2D y coordinate of the feature points such that they are represented as,

$$(y_{f_{i1}^l} < y_{f_{i2}^l} < y_{f_{i3}^l} ..... < y_{f_{ij}^l}),\ \forall\, f_{ij} \in F_i^l \quad (4)$$

$$(y_{f_{i1}^r} < y_{f_{i2}^r} < y_{f_{i3}^r} ..... < y_{f_{ij}^r}),\ \forall\, f_{ij} \in F_i^r \quad (5)$$

4.3 Partial Depth Estimation

Estimating the entire depth image by stereo block matching methods as proposed in [Konolige, 1998, Hirschmuller, 2008] is computationally expensive. Full depth image can be obtained from the block matching methods which is unnecessary for the localization process unless we want to develop a SLAM system. In order to avoid these expensive computation task, we estimate the depth value of the SIFT feature points only. Feature points need to be represented in 3D world coordinates from 2D pixel coordinates. The projection equation for the stereo vision is described in [Hartley and Zisserman, 2004] and is given by,

$$Q \begin{bmatrix} x_{ij} \\ y_{ij} \\ d_{ij} \\ 1 \end{bmatrix} = \begin{bmatrix} X_{ij} \\ Y_{ij} \\ Z_{ij} \\ W_{ij} \end{bmatrix},\ \forall\, f_{ij} \in F_i^l, F_i^r \quad (6)$$

where, d is the disparity between the feature points in the left and right images and Q is the reprojection matrix of the camera. The disparity d and reprojection matrix Q is given as,

$$d_{ij} = \sqrt{(x_{f_{ij}^l} - x_{f_{ij}^r})^2 + (y_{f_{ij}^l} - y_{f_{ij}^r})^2},\ \forall\, f_{ij} \in F_i^l, F_i^r \quad (7)$$

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/T_x & (c_x - c_x{}')/T_x \end{bmatrix}, \quad (8)$$

where c_x, c_y represents x and y coordinate of the principal point of the image, f is the focal length of the stereo cameras and T_x represents the baseline between the stereo cameras. From (7) and (8), we can obtain the 3D world coordinates of the feature points using the following equations,

$$W_{ij} = ((c_x - c_x{}') - d_{ij})/T_x, \quad X_{ij} = (x_{f_{ij}^l} - c_x)/W_{ij},$$
$$Y_{ij} = (y_{f_{ij}^l} - c_y)/W_{ij}, \quad Z_{ij} = f/W_{ij},$$

$$q_{ij} = (X_{ij}, Y_{ij}, Z_{ij})$$

Thus, q_{ij} represents the 3D homogeneous world coordinates of the feature points and the term Z_{ij} represents the depth of the feature points.

4.4 Removal of Features with Invalid Depth

After calculating the 2D pixel location and 3D world location of feature points, we need to remove the features with invalid depth values. For any stereo cameras, depth range is limited due to the baseline length between the left and right cameras. A feature point is discarded from the feature set if,

$$(Z_{ij}/W_{ij}) \geq D',\ \forall\, q_{ij} \in Q_i \quad (9)$$

where, D' is the maximum depth range that can be computed for a particular stereo set-up.

4.5 Feature Matching between i^{th} Frame and $(i+n)^{th}$ Frame

The above steps are used for computing features of the left and right image of the i^{th} frame only. The procedure is repeated for successive N number of frames. Choose the frame interval n (say, $n = N$) to compute the pose estimation. Finally, we obtain the set of 2D and 3D locations for the feature points in each frame. This is given by,

$$\begin{aligned} F_1^l &= \{f_{11}^l, f_{12}^l, ...f_{1j}^l\} \\ Q_1 &= \{q_{11}, q_{12}, ...q_{1j}\} \end{aligned} \quad ..... \quad \text{Frame 1}$$

$$\vdots$$

$$\begin{aligned} F_n^l &= \{f_{n1}^l, f_{n2}^l, ...f_{nj}^l\} \\ Q_n &= \{q_{n1}, q_{n2}, ...q_{nj}\} \end{aligned} \quad ..... \quad \text{Frame n}$$

where $F_n^l \in \mathbb{R}^2$ represents the set of 2D pixel location of feature points corresponding to left image of the n^{th} frame. The 2D pixel locations can be obtained either from the left or right images. $Q_n \in \mathbb{R}^3$ is the respective set of 3D location of feature points (F_n^l) at the n^{th} frame. After the frame interval n, the above feature manipulation processes are again performed on the $(i+n)^{th}$ frame as well as the correspondence relation between the i^{th} frame and $(i+n)^{th}$ frame are estimated. This correspondence estimation is the additional step performed for all the frames from the $(i+n)^{th}$ to $(i+N+n)^{th}$ frames. Only the left images of the i^{th} and $(i+n)^{th}$ frames are used to match features using FLANN Matching algorithm. A new feature set $F'_{i|i+n}$ is obtained as the result of matching between the i^{th} and $(i+n)^{th}$ frames such that,

$$F'_{i|i+n} = \{f'_{i1}, f'_{i2}, ...f'_{ij}\} \quad (10)$$

4.6 Estimation of 3D Location of the Common Features between the Previous and Current Frame

Next step is to obtain the 3D location of those common features at the i^{th} and $(i+n)^{th}$ instants. This task is easy as we have already computed the 3D location of feature points for

all the previous frames. For every common feature points determined in the previous step, we need to find their respective 3D locations in the i^{th} and $(i+n)^{th}$ frames. An efficient search algorithm named Binary search algorithm, proposed in [Knuth, 1998], is used to search the required features between the previous and current instants. Binary search algorithm requires the elements to be sorted before searching. This was the reason for sorting the feature points in the earlier steps. The $y_{f'_{ij}}$ coordinate value of the common feature point is chosen as the search key for the binary search algorithm. Once there is matching $y_{f'_{ij}}$ value in the i^{th} and $(i+n)^{th}$ frame, then their corresponding 3D locations is obtained from the 3D feature set Q_i. The new feature set contains only the features common between the two frames as,

$$\forall i,j,\ f^l_{ij} \in (F^l_i)_{new},\ |\ \exists F'_{i|i+n} \subseteq F^l_i \wedge y_{f^l_{ij}} \approx y_{f'_{ij}}, \quad (11)$$

$$q_{ij} \in (Q_i)_{new},\ \forall f^l_{ij} \in (F^l_i)_{new} \quad (12)$$

Thus, after this search process, the 2D $(F^l_i)_{new}$ and 3D $(Q_i)_{new}$ are the truncated feature sets of the i^{th} frame corresponding to the new feature set $F'_{i|i+n}$. Similarly, the feature sets $F^l_{(i+n)j}$ and $Q_{(i+n)j}$ from the $(i+n)^{th}$ frame is also truncated to $(F^l_{(i+n)j})_{new}$ and $(Q_{(i+n)j})_{new}$ accordingly.

4.7 Discarding Outlier in the 3D Feature Point Locations

There may be still few outlier features among the estimated common features. In order to discard those outliers, we use the Random Sample Consensus (RANSAC) outlier rejection algorithm proposed in [Fischler and Bolles, 1981]. RANSAC is an iterative regression algorithm which tries to fit a plane to the set of estimated 3D feature points. It effectively removes the feature points that are far away from the plane fitted to the feature data. This outlier rejection greatly helps in improving the accuracy of the localization result. RANSAC algorithm is used for both the left and right image 3D feature points. RANSAC iterative procedure is performed only on the feature points common between the i^{th} and $(i+n)^{th}$ frames.

4.8 Estimation of 3D Transformation

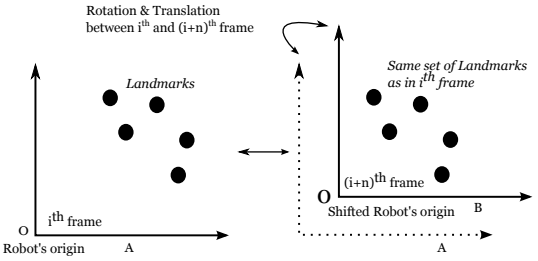


Fig. 3. 3D transformation between the i^{th} and $(i+n)^{th}$ frames.

Feature Matching and Outlier removal is performed for all the frame pairs from $[i^{th},(i+n)^{th}]$ frame pair to $[(i+N)^{th},(i+N+n)^{th}]$ frame pair. Now, we need to estimate the 3D transformation between the frame pairs as shown in Fig. 3. This transformation yields us the translation and rotation between the frame pairs, which indirectly gives us the translation and rotation of the robot from the previous instant to the current instant. Consider the two sets of 3D feature points, one from the i^{th} frame instant and the other from the $(i+n)^{th}$ frame instant. The transformation between the two 3D feature point sets is computed using a least squares fitting algorithm with Singular Value Decomposition (SVD) as proposed in [Arun et al., 1987]. Eggert et al. [1997] provided a comparison between four different R and T estimation methods and proved that the SVD method proves to be more efficient than the other methods. Let us consider the two 3D feature point sets Q_i and Q_{i+n} that are related by certain amount of rotation and translation as,

$$Q_{i+n} = R_{i|i+n}Q_i + T_{i|i+n} \quad (13)$$

$$Q^c_{i+n} = \frac{1}{j}\sum_{i=1}^{j} Q_{i+n} \quad \& \quad Q^c_i = \frac{1}{j}\sum_{i=1}^{j} Q_i \quad (14)$$

Now, compute the correlation matrix H (3×3) using,

$$H_{i|i+n} = \sum_{i=1}^{j}(Q_i - Q^c_i).(Q_{i+n} - Q^c_{i+n})^T. \quad (15)$$

Then, decompose the H matrix using SVD to get the two orthonormal matrices U and V and the diagonal matrix Σ,

$$[U\Sigma V] = \text{SVD}\,(H_{i|i+n}). \quad (16)$$

After obtaining the U and V matrices from the decomposition, the rotation matrix, $R_{i|i+n}$ and translation vector $T_{i|i+n}$ is obtained as

$$R_{i|i+n} = VU^T \quad (17)$$

$$T_{i|i+n} = Q_{i+n} - R_{i|i+n}Q_i. \quad (18)$$

The above estimation is performed only for the $[i^{th},(i+n)^{th}]$ frame pair instant. Similarly, we need to estimate the transformation for all the frame pairs such as $[(i+1)^{th},(i+1+n)^{th}]$,.....$[(i+N)^{th},(i+N+n)^{th}]$ frames. Finally, We need to calculate the mean transformation from all the R and T estimates. Fig. 4 explains the process of consecutive estimates of transformation from the i^{th} frame to $(i+N+n)^{th}$ frame.

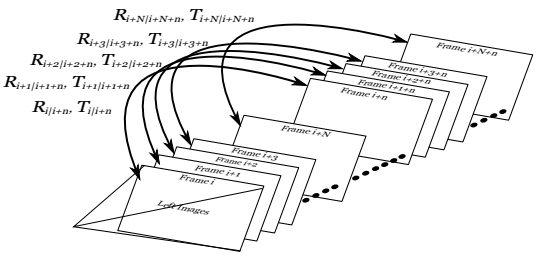


Fig. 4. R & T estimates between the consecutive $i+n$ frames. (Note: The n value can also be $n=N$).

The transformation matrix Tr is the effective transformation pose of the robot between the i^{th} and $(i+n)^{th}$ frame and it is given by,

$$Tr^{eff}_{i|i+n} = \begin{pmatrix} R_{i|i+n} & T_{i|i+n} \\ 0\ 0\ 0 & 1 \end{pmatrix} \quad (19)$$

where,

$$R_{i|i+n} = \begin{pmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{pmatrix} \quad (20)$$

$$T_{i|i+n} = (\ t_x\ t_y\ t_z\ )^T \quad (21)$$

The global transformation matrix Tr^{global}, is obtained from the effective pose transformation matrix as,

$$Tr^{global} = Tr^{global}\ .\ Tr^{eff} \quad (22)$$

This global transformation matrix gives the global pose of the robot (X,Y,Z) with respect to the world coordinate frame.

5. FINE TUNING WITH PARTICLE FILTER

In spite of rejecting most of the noise in the features selection process, there will still be some errors in the estimated transformation matrix. The most common type of filter employed in tracking or odometry applications is Kalman Filter. If the measurements are multimodal, then the variants of Kalman Filter such as Extended Kalman Filter (EKF) or Unscented Kalman Filter (UKF) can be employed. The disadvantage of the EKF or UKF is that both these filters try to linearize the non-linearity in the sensor measurements using approximation methods. Hence, Particle filtering technique is used to filter the errors in our algorithm. Kalman Filter predicts the position of the robot at the next instant from the obtained measurements from past and current instants while the Particle Filter samples the probable positions of the robot from 1^{st} position to say N possible positions. In [Fox et al., 1999] and [Thrun, 2002], the authors proposed a localization algorithm based on Monte-Carlo method using the particle filter technique. In our algorithm, a particle filter predicts the position of the robot based on the estimated feature locations. Since the robot is a Unmanned Ground Vehicle (UGV), the motion along the vertical axis of the robot is assumed to be negligible. So, both the pose estimates and feature location estimates in $\mathbb{R}^3$ is assumed to be in $\mathbb{R}^2$. We formulate the pose estimates from $(X, Y) \in \mathbb{R}^2$ into motion estimate of the robot in (r, δ). A motion model developed for the UGV with the formulated motion estimates is given as

$$\begin{pmatrix} x_p \\ y_p \\ \theta_p \end{pmatrix} = \begin{pmatrix} x_c \\ y_c \\ \theta_c \end{pmatrix} + 2 \begin{pmatrix} r\cos(\theta_c + \delta) \\ r\sin(\theta_c + \delta) \\ \theta_c + \delta \end{pmatrix}, \quad (23)$$

where (x_p, y_p, θ_p) is the predicted localization estimate from the current position estimates (x_c, y_c, θ_c) and the motion estimates (r, δ). Particle filter samples position of robot into multiple hypothetical positions called particles. All these particles will have their own belief confidences from the predicted position. The beliefs are influenced by the global landmark locations $\in Q_i$. This process of creating the multiple beliefs is called *Sampling process*. This sampling process takes care of the uncertainty introduced in the estimation of R and T. Once the pose is estimated for the i^{th} frame, then we need to sample the position of the robot. This process can be given as,

$$\mathcal{P}_i = \mathcal{X} * r^n, \quad (24)$$

where $\mathcal{X}$ is the pose estimate of the robot given by (x, y, θ) and $\mathcal{P}_i$ is the set of particles representing the probable position of the robot at the i^{th} instant. In our algorithm, we fix the size of the particle set to be constant at 1000 particles.The term r^n is the random noise, which induces noise in (x_p, y_p, θ_p) components for sampling the pose estimates into 1000 particles. In the *prediction step*, the pose for all the particles at the next frame instant is predicted as,

$$\mathcal{X}_i^{[m]} \sim p(\mathcal{X}_i | u_i, \mathcal{X}_{i-1}^{[m]}, Q_{i-1}), \quad (25)$$

where $\mathcal{X}_i$ is the predicted pose estimate for each particle $\mathcal{X}_i^{[m]}$ with respect to the previous pose estimates $\mathcal{X}_{i-1}^{[m]}$, motion control input u_i and feature set locations Q_{i-1}. This step is called as the *Belief propagation*. At the $(i+n)^{th}$ instant, we determine the next set of feature point locations Q_{i+n} and estimate the new motion estimate using the new measurements. This is the *measurement update step* where the predicted pose estimates are corrected with the newly obtained motion estimates. The Particle set $\mathcal{P}_i$ is updated as,

$$w_t^{[m]} = p(Q_i | \mathcal{X}_i^{[m]}) \quad (26)$$

$$\bar{\mathcal{P}}_i = \mathcal{P}_i + (\mathcal{X}_i^{[m]}, w_i^{[m]}), \quad (27)$$

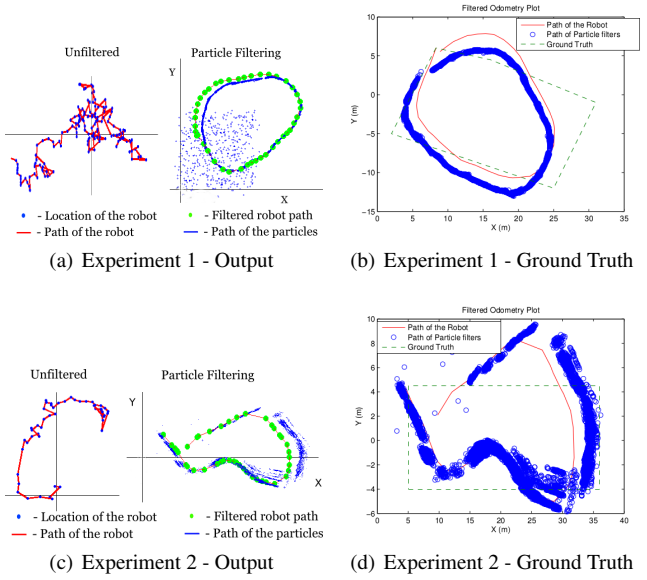
where $\bar{\mathcal{P}}_i$ is the corrected set of particles indicating the possible poses of the robot and $w_i^{[m]}$ is the confidence probability assigned to the each particles based on their distance with respect to the set of measurements Q_i at the current instant. Particles having least probability are replaced with the particles having high probability. This process is called *Low variance sampling*. So, only the particles with high beliefs are involved in the next prediction and measurement step. This process is continued throughout the robot's motion for each frame pairs to filter the errors in R and T and gradually all the particles will converge to the true robot location.

6. EXPERIMENTAL RESULTS

Experiments were performed with the proposed localization algorithm in actual robots. We used a commercial stereo camera *Bumblebee 2* mounted on a Pioneer P3-AT robot. The experimental setup is shown in Fig. 5. Initially the algorithm is used

Fig. 5. Experimental setup with a stereo camera.

without the particle filtering technique. Next, we included the particle filtering method in the algorithm, so as to achieve better localization results. The particle filter usually initializes the starting point of the robot as a random value. So, the initial start point will be a random point in Fig. 6(a) and 6(c). From the initial random point, the next pose is estimated based on the motion estimates calculated from the transformation matrix Tr^{eff} for each frame pair instants. The figures 6 show our experimental results that were performed to close a loop. It is clear from Fig. 6(a) and 6(c) that the unfiltered transformation estimation yields incorrect odometry path. We observed that the estimated R matrix is incorrect because of the presence of outliers in the 3D feature points and also depth estimation is incorrect due to occlusion and FOV limitation. But this can be improved by using a stereo setup with a very wide baseline T_x. However, our particle filter method reduces the errors by imposing constraints in the pose estimates (x, y, θ). we can observe that the starting point is at random position in both the experiments and also random noise is generated initially around the starting point. Then the random noise gradually converges to the actual location of the robot pose and finally closes the loop successfully. The table 1 and 2 shows the endpoint position of the robot with respect to the true ground truth. It can be seen that localization estimates obtained from the experiment 1 is better than the experiment 2 as the number of inlier features detected in the experiment 2 are less compared to the features detected in experiment 1. We can also observe that the unfiltered odometry results deviates away from the ground truth path. In Fig. 7, the mean squared error between the determined robot pose estimates and ground truth is plotted against the time of duration of the robot motion. In experiment 1, the error is relatively small at the end point even though it increases from the starting point but in experiment 2, the pose error increases as the duration of robot traversal increases. Based on the experimental results, it can be observed that in our proposed algorithm, the



(a) Experiment 1 - Output (b) Experiment 1 - Ground Truth

(c) Experiment 2 - Output (d) Experiment 2 - Ground Truth

Fig. 6. Localization Result of Experiments.

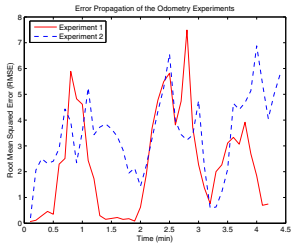


Fig. 7. Error Propagation with prolonged localization

Table 1. Particle Filter Results with Experiment 1

	X value (m)	**Y value** (m)	θ (rad)
Ground Truth Endpoint	8.30	6.00	0.00
Estimated Position Endpoint	8.894	5.547	3.269
Error	0.594	−0.453	3.269

Table 2. Particle Filter Results with Experiment 2

	X value (m)	**Y value** (m)	θ (rad)
Ground Truth Endpoint	4.50	5.00	0.00
Estimated Position Endpoint	9.769	2.093	1.659
Error	5.269	−2.907	1.659

particle filter tries to converge to true robot location thereby suppressing the error in the estimation of R and T.

7. CONCLUSION

In this paper, a new stereo vision based robot localization process is developed by combining the partial depth estimation and particle filter technique. The localization of the robot is achieved by effectively computing the robot pose estimates from the motion estimates of the features between each frame instants. Although there are various state-of-the-art algorithms already available for vision based robot localization, our method also proves to be one of the promising algorithms for robot localization. Experiments were conducted at our university campus and the accuracy and persistence of the localization path of the robot is analysed. Planned future works for the proposed localization algorithm include reducing computation time using a dynamic size for the particle set and using global minimization techniques to suppress the outlier errors in the estimation of transformation matrix.

REFERENCES

K.S. Arun, T. S. Huang, and S. D. Blostein. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9, 1987.

J. Y Bouguet and P. Perona. Visual navigation using a single camera. In *Proceedings of the International Conference on Computer Vision*, 1995.

J.Y. Bouguet. Matlab Camera Calibration Toolbox. 2000. URL `http://www.vision.caltech.edu/bouguetj/calib_doc`.

D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-D Rigid Body Transformations: a Comparison of Four Major Algorithms. *Machine Vision and Applications*, 9, 1997.

Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24, 1981.

Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. In *Proceedings of the National Conference On Artificial Intelligence (AAAI)*, 1999.

R. I. Hartley and A. Zisserman. *"Multiple View Geometry in Computer Vision"*. Cambridge University Press, 2004.

H. Hirschmuller. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30, 2008.

B. Kitt, A. Geiger, and H. Lategahn. Visual Odometry based on Stereo Image sequences with RANSAC-based Outlier Rejection Scheme. In *IEEE Intelligent Vehicles Symposium (IV),*, 2010.

Donald E. Knuth. *"The Art of Computer Programming, Volume 3: (2nd Ed.) Sorting and Searching"*. Addison Wesley Longman Publishing Co., Inc., 1998.

Kurt Konolige. Small Vision Systems: Hardware and Implementation. In Yoshiaki Shirai and Shigeo Hirose, editors, *Robotics Research*. Springer London, 1998.

David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60, 2004.

Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the Mars Exploration Rovers. *Journal of Field Robotics, Special Issue on Space Robotics*, 24, 2007.

Marius Muja and David G. Lowe. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In *VISAPP International Conference on Computer Vision Theory and Applications*, 2009.

Stephen. Se, D.G. Lowe, and J.J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21, 2005.

S. Thrun. Particle Filters in Robotics. In *Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)*, 2002.

S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Minerva: a second-generation museum tour-guide robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, 1999.

David Yuen, Bruce Macdonald, David C. K. Yuen, and Bruce A. Macdonald. Vision-based localization algorithm based on landmark matching, triangulation, reconstruction and comparison. *IEEE Transactions on Robotics*, 21, 2005.