Information-based Reduced Landmark SLAM

Siddharth Choudhary*, Vadim Indelman†, Henrik I. Christensen*, Frank Dellaert*

Abstract—In this paper, we present an information-based approach to select a reduced number of landmarks and poses for a robot to localize itself and simultaneously build an accurate map. We develop an information theoretic algorithm to efficiently reduce the number of landmarks and poses in a SLAM estimate without compromising the accuracy of the estimated trajectory. We also propose an incremental version of the reduction algorithm which can be used in SLAM framework resulting in information based reduced landmark SLAM. The results of reduced landmark based SLAM algorithm are shown on Victoria park dataset and a Synthetic dataset and are compared with standard graph SLAM (SAM [6]) algorithm. We demonstrate a reduction of 40-50% in the number of landmarks and around 55% in the number of poses with minimal estimation error as compared to standard SLAM algorithm.

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is one of the basic problems in mobile robotics. SLAM allows the robot to incrementally build a consistent map of the environment while simultaneously using the map to localize itself. It has a wide range of applications from service robotics to reconnaissance operations. The complexity of the existing SLAM approaches grow with the length of the robot’s trajectory which challenges long term autonomy. Some recent approaches solve this problem by compressing the pose graph to reduce its complexity [9], [17], [14], [12], [2], [23], [18]. These methods sparsify the pose graph to reduce the number of poses and do not consider the size of the generated map.

As compared to pose graph based SLAM, landmark based SLAM explicitly maintains the landmark and robot states. Keeping landmarks in the SLAM estimate has advantages over marginalizing out. For example, semantic landmarks like planes and objects can be used to perform tasks which require higher cognition capability. Additionally, it can be associated with the current robot pose and therefore be used to localize the robot [24], [4]. However storing all landmarks and poses can become expensive in the long term.

Therefore, in this paper we address this problem by developing an information-based approach to select a reduced number of landmarks and poses for a robot to successfully localize itself and simultaneously build an accurate map. We present an incremental and active minimization approach (Section IV) for doing reduced landmark based SLAM. The active minimization approach uses an information theoretic algorithm (Algorithm 3) to reduce the number of landmarks and poses. Performing the minimization in an incremental manner (Algorithm 5) results in a reduced landmark based SLAM algorithm. We evaluate our approach using two datasets, Victoria Park [11] and a synthetic dataset comparing it to standard graph SLAM (SAM [6]) and demonstrating a reduction of 40-50% in the number of landmarks and around 55% in the number of poses with minimal estimation error. Figure 1 shows the comparison of robot trajectory and landmark locations estimated using standard graph SLAM and reduced landmark based SLAM on Victoria Park dataset.

Our contributions in this paper are as follows:

1) We develop an information theoretic algorithm to efficiently reduce the number of landmarks and poses without compromising the accuracy of the estimated trajectory.
2) We propose an incremental version of this algorithm which can be used in a SLAM framework required for online operations.

In Section II, we discuss the related work in this area. Section III reviews landmark based SLAM and proposes a formulation for reduced landmark based SLAM. Section IV describes the incremental and active minimization algorithms for doing the same. In Section V we give an extensive evaluation of our approach on two datasets and we conclude in Section VI.

II. RELATED WORK

SLAM is an active area of research in robotics. One of the initial solutions to the SLAM problem was proposed by Smith and Cheeseman who used the Extended Kalman Filter (EKF) to jointly represent the landmark position with the
pose [21]. Durrant-Whyte and Bailey provide a survey of
the SLAM literature [8].

Dissanayake et al. [7] showed that it is possible to remove
a large percentage of the landmarks from the map without
making the map building process statistically inconsistent.
Folkesson and Christensen introduced GraphSLAM system
which finds the best robot trajectory using a non linear
optimization technique [10]. Dellaert and Kaess exploited
the inherent sparsity of the SLAM problem to make the
process more efficient [6]. These approaches solve a batch
least squares optimization which become computationally
expensive for large-scale problems.

Various other methods have been developed to reduce the
computational time. Sibley et al. used sliding window filter
whereas Ni et al. divided the whole graph into multiple local
sub-graphs [19], [20]. Incremental smoothing and mapping
algorithm incrementally updates the square root information
matrix with the new measurements [16]. In this approach,
the length of the trajectory depends on the exploration time
rather than the explored area which becomes computationally
expensive for long term SLAM.

Kretzschmar and Stachniss proposed an information theo-
retic approach to compress the pose graph by selecting
the most informative laser scans with respect to the map
[17]. Ila et al. [13] proposed to add only non-redundant
and informative links to the graph. Eade et al. [9] reduced
the complexity of the graph by marginalizing out past robot
poses that are not useful in subsequent operations. Vial et al.
[25] presented a conservative sparsification technique that
minimizes the KL divergence of the information matrix for
sparsely approximating multi-dimensional Gaussian distribu-
tions. Huang et al. [12] propose a consistent graph sparsi-
fication scheme to marginalize out old nodes. Wang et al.
[26] performs pose graph reduction using a greedy pruning
based on KL Divergence measure between the reduced graph
and full graph. Johannsson et al. [14] used a reduced pose
graph representation to bound the size of the pose graph
with respect to the explored area. Carlevaris-Bianco et al.
[2] proposed a generic node removal technique to produce
a new set of linearized factors over the elimination cliques
that represents either the true or a sparse approximation
of the true marginalization. Mazuran et al. [18] formulated
sparsification as a convex minimization problem. All of these
compress the pose graph by reducing views or poses to
reduce the complexity of graph optimization where as our
approach reduces the number of landmarks and poses in a
landmark based SLAM framework.

Suger et al. [23] present an approximate SLAM based on
hierarchical decomposition to reduce the memory consump-
tion. Cao and Snavely [1] proposed a probabilistic K-cover
algorithm to reduce the size of 3D reconstructions. Chli et
al. used mutual information between the measurements to
guide the search during feature matching step of a visual
SLAM system [3]. Kaess and Dellaert ignore the feature
measurements whose information gain falls below a specified
threshold [15]. In contrast we estimate the information
provided by a landmark rather than a measurement.

III. APPROACH

While the robot is navigating, reduced landmark based
SLAM selects a reduced number of landmarks and poses
from the environment for a robot to localize itself and simul-
taneously build an accurate map. We develop an incremental
and active minimization approach (Section IV) for doing
reduced landmark-based SLAM. The active minimization
approach uses an information theoretic algorithm (Algorithm
3) to reduce the number of landmarks and poses. Incremental
minimization performs the same in an iterative manner
(Algorithm 5).

In the following sections we review the standard graph
SLAM formulation (using all landmarks and poses) and
propose a formulation for reduced landmark-based SLAM.

A. Landmark-based SLAM

In landmark-based SLAM a robot while navigating tries
to localize itself and at the same time build a map of the
environment (represented using landmarks). Assuming the
pose of the robot at $i^{th}$ time step is $x_i$ with $i \in \{0 \ldots M\}$,
a landmark is $l_j$ with $j \in \{1 \ldots N\}$ and a measurement is $z_k$,
with $k \in \{1 \ldots K\}$, the joint probability model is given as,

$$P(X, L, Z) = P(x_0) \prod_{i=1}^{M} P(x_i|x_{i-1}, u_i) \prod_{k=1}^{K} P(z_k|x_{i_k}, l_{jk})$$

where $P(x_0)$ is a prior on the initial state, $P(x_i|x_{i-1}, u_i)$
is the motion model, parameterized by a control input or
odomentry measurement $u_i$ and $P(z_k|x_{i_k}, l_{jk})$ is the landmark
measurement model where $x_{i_k}$ ($i^{th}$ pose) and $l_{jk}$ ($j^{th}$
landmark) correspond to the $z_k$ measurement. $X$ is the
set of poses, $L$ is the set of landmarks and $Z$ is the set
of measurements. Figure 2 shows the corresponding Bayes
Net. Assuming the motion and measurement models are
Gaussian, $P(x_i|x_{i-1}, u_i) \propto \exp(-\frac{1}{2}\|f_i(x_{i-1}, u_i) - x_i\|^2_{\Lambda_i})$
and $P(z_k|x_{i_k}, l_{jk}) \propto \exp(-\frac{1}{2}\|h_k(x_{i_k}, l_{jk}) - z_k\|^2_{\Sigma_k})$
where $f()$ is the robot motion equation and $h()$ is a landmark
measurement equation with $\Lambda_i$ and $\Sigma_k$ as the respective
covariances.

![Belief Net corresponding to the Landmark based SLAM problem. The pose of the robot at $i^{th}$ time step is $x_i$ with $i \in \{0 \ldots M\}$, a landmark is $l_j$ with $j \in \{1 \ldots N\}$ and a measurement is $z_k$, with $k \in \{1 \ldots K\}$.](image-url)
or $P(x_i|x_{i-1}, u_i)$ or $P(z_k|x_{ik}, l_{jk})$. Therefore the joint probability model can be written as

$$P(X, L, Z) \propto g(\Theta) = \prod_i g_i(\Theta_i)$$

where $\Theta_i$ is the set of variables $\theta_j$ adjacent to the factor $g_i$. Given all the measurements, we obtain the maximum a posteriori (MAP) estimate $\Theta^*$ by maximizing the joint probability $P(X, L, Z)$.

$$\Theta^* = \arg \max_\Theta P(X, L|Z) = \arg \min_\Theta (-\log g(\Theta))$$

which leads to the following non-linear least squares problem:

$$\Theta^* = \arg \min_\Theta \frac{1}{2} \sum_{i=1}^M \| f_i(x_{i-1}, u_i) - x_i \|^2_{A_i} + \sum_{k=1}^K \| h_k(x_{ik}, l_{jk}) - z_k \|^2_{R_{ik}}$$

The non-linear least squares problem in Equation 3 is solved using non-linear optimization method such as Levenberg-Marquardt algorithm which solves a succession of linear approximations in order to approach the minimum.

In each iteration of the non-linear least squares problem, we linearize around a linearization point $\tilde{\Theta}$ to obtain a linear least squares problem of the form $\delta^* = \arg \min_\delta \| A\delta - b \|^2_2$ where $A$ represents the Jacobian matrix, $b$ is the set of measurements (odometry or landmark) and $\delta$ represents the change around the current estimate of $X$ and $L$. For a full rank matrix $A$, the least squares solution can be found by solving the normal equations $A^T A \delta^* = A^T b$. Cholesky factorization yields $A^T A = R^T R$ where $R$ is an upper-triangular matrix. A forward substitution on $R^T y = A^T b$ followed by backward substitution $R^T \delta^* = b$ gives the update $\delta^*$. We use the Georgia Tech Smoothing and Mapping (GTSAM) library to jointly optimize the robot poses and landmarks [5].

**B. Reduced Landmark-based SLAM**

The goal of reduced landmark-based SLAM is to navigate in an unknown environment using the reduced amount of landmarks and poses. It can be formulated as incrementally finding a reduced subset of landmarks and robot poses $(L_s, X_s)$ such that the difference between trajectory estimated using a reduced subset of landmarks and robot poses and the trajectory estimated using all landmarks and robot poses is minimal. However, there is a trade-off between the memory requirement and the estimation error because reducing the number of landmarks (memory requirement), increases the error in the estimated trajectory.

To represent the trade-off between the memory requirements and the estimation accuracy, we formulate the problem as finding the subset of landmarks $L_s \in L$ and robot poses $X_s \in X$ which minimizes the objective function given as,

$$\rho(L_s, X_s) = (1-\lambda)d(\Theta^*_s) + \lambda m(L_s, X_s) \quad (4)$$

where $d()$ is the distance function representing the accuracy of the trajectory estimated ($\Theta^*_s$) using the subset of landmarks $L_s$ and poses $X_s$. $m()$ is the amount of memory required to store the same and $\lambda$ is the weight parameter. The objective function represents the trade-off using $\lambda$ as the weight parameter. Higher $\lambda$ will force the optimization to remove as many landmarks and poses as possible whereas lower $\lambda$ will be more conservative by retaining more landmarks and poses. $d()$ and $m()$ are normalized to have values between 0 and 1. Equation 4 is discussed in Section IV-B.

The reduced set of landmarks and poses is estimated by minimizing Equation 5.

$$\{L^*, X^*\} = \arg \min_{L_s, X_s} \rho(L_s, X_s) \quad (5)$$

Directly minimizing the above objective function is infeasible since we have to iterate over all combinations of landmarks and poses which is of the order $2^N$ where $N$ is total number of poses and landmarks. Below we describe an incremental and active minimization approach for the same.

**IV. REDUCED LANDMARK BASED SLAM VIA INCREMENTAL AND ACTIVE MINIMIZATION**

We estimate the subset of landmarks and poses by actively removing the least informative landmark until the objective function (Equation 4) has its minimum value. The corresponding poses that do not see any landmark are marginalized out. Below we describe information gain and the objective function used followed by the description of active and incremental minimization algorithms.

**A. Information Gain**

Mutual information $I(\alpha, \beta)$ is the measure of the expected information gain in $\alpha$ on measuring the exact value of $\beta$ or vice versa. The mutual information of two continuous multivariate PDFs $p(\alpha)$ and $p(\beta)$ is

$$I(\alpha; \beta) = H(\alpha) - H(\alpha|\beta) = \mathbb{E}\left[\log_2 \frac{p(\alpha|\beta)}{p(\alpha)}\right]$$

In the case of multivariate Gaussian describing a state vector $x$ by mean vector $\mu$ and covariance matrix $\Sigma$ such that

$$x = \begin{pmatrix} x_\alpha \\ x_\beta \end{pmatrix}, \mu = \begin{pmatrix} \mu_\alpha \\ \mu_\beta \end{pmatrix}, \Sigma = \begin{bmatrix} \Sigma_{\alpha\alpha} & \Sigma_{\alpha\beta} \\ \Sigma_{\beta\alpha} & \Sigma_{\beta\beta} \end{bmatrix}$$

where $x_\alpha$ and $x_\beta$ are the two disjoint partitions of the state vector $x$. The mutual information between the two partitions $x_\alpha$ and $x_\beta$ is given by

$$I(\alpha; \beta) = \frac{1}{2} \log_2 \frac{\Sigma_{\alpha\alpha}}{\Sigma_{\alpha|\beta}} \quad (6)$$

where $\Sigma_{\alpha|\beta}$ is the covariance matrix corresponding to $p(\alpha|\beta)$ and $\Sigma_{\alpha\alpha}$ is the covariance matrix corresponding to $p(\alpha)$. The computation of log covariance determinant $\log_2 |\Sigma|$ can be expensive for a large state space $x$. Since SAM [6] maintains the square root information matrix $R$ we
can compute the log of covariance determinant using the determinant of R matrix as shown below.

\[ \log_2 |\Sigma| = \log_2 |A^T A|^{-1} = \log_2 |R^T R|^{-1} = -2 \log_2 |R| \]  

(7)

R is an upper triangular matrix and therefore, the determinant of R is the product of its diagonal terms. When R matrix is not available we have to re-linearize the factor graph \( g(\Theta) \) to get A followed by QR factorization to get R. Algorithm 1 shows the overall process.

**Algorithm 1 LogDet \((X, g(\Theta), \Theta^*)\)**

1. Linearize \( g(\Theta) \) around \( \Theta^* \) to get A
2. Eliminate X from A to get R (QR factorization)
3. \( \log \text{Det} = -2 \times \sum_i (\log_2 R_{ii}) \) \( \triangleright \) Equation 7
4. return \( \log \text{Det} \)

**Algorithm 2 InfoGain \((X, L, g(\Theta), \Theta^*)\)**

1. \( X_r \leftarrow \{X / L\} \)
2. \( \log \text{Det}_{th} \leftarrow \log \text{Det}(X_r, g(\Theta), \Theta^*) \) \( \triangleright \) \( \log_2 |\Sigma_{x/L}| \)
3. \( g_r(\Theta) \leftarrow \text{Remove factors having L from } g(\Theta) \)
4. \( \Theta^*_r \leftarrow \{\Theta^*_s / \Theta^*_L\} \)
5. \( \log \text{Det}_{th} \leftarrow \log \text{Det}(X_r, g_r(\Theta), \Theta^*_r) \) \( \triangleright \) \( \log_2 |\Sigma_{x/L}| \)
6. return \( \frac{1}{2} \times (\log \text{Det}_{th} - \log \text{Det}_{th}) \)

\( \frac{1}{2} \times \log_2 |\Sigma_{x/L}| \)

**B. Objective Function**

We now discuss in detail the components of the objective function \( \rho(L_s, X_s) \).

**Distance Function \( d(\Theta^*_s) \):** The distance function represents the accuracy of the trajectory \( \Theta^*_s \) estimated using a reduced set of poses \( X_s \) and landmarks \( L_s \). For a consistent estimator, the accuracy of the estimated trajectory is proportional to the uncertainty in \( \Theta^*_s \). As shown in Equation 6, information gain represents the expected reduction in uncertainty on measuring a landmark and analogously it represents the expected increase in uncertainty on removing a landmark.

Therefore we use information gain of the least informative landmark as the distance function corresponding to uncertainty in the estimated trajectory. Information gain of the least informative landmark provides a lower bound on the information gain estimated using any of the remaining landmarks. It also represents the minimum increase in uncertainty on removing a landmark. This distance function works good in practice, although better distance function is an interesting area for future work.

**Memory Consumption \( m(L_s, X_s) \):** As we remove the least informative landmark during each iteration and estimate the value of the objective function, we use the number of landmarks remaining as the value of \( m(L_s, X_s) \). Figure 3 shows the normalized objective function for Victoria park dataset using \( \lambda = 0.5 \) having 6536 poses and 149 landmarks.

**C. Active Minimization**

Using the above formulations, we propose an active, batch minimization algorithm as summarized in Algorithm 3. The input to the algorithm is the current state of SLAM described by the set of landmarks \( L \), robot poses \( X \), measurements \( Z \), the optimal landmark location and trajectory estimate \( \Theta^* \) obtained by maximizing the probability \( P(X, L, Z) \propto g(\Theta) \) and the weight vector \( \lambda \). The value of \( \lambda \) is set according to how conservative the minimization has to be. By default, we use \( \lambda = 0.5 \) to get a good balance between the memory function value and the distance function value.

**Algorithm 3 Active Minimization \((X, L, \Theta^*, g(\Theta), \lambda)\)**

1. Initialize \( L_o \leftarrow L, X_s \leftarrow X, \Theta^*_o \leftarrow \Theta \)
2. Initialize \( \rho(L_o, X_s) \leftarrow \lambda m(L_s, X_s) = \lambda |L_s| \)
3. repeat
   4. \( (L_o, IG) \leftarrow \text{LandmarkSelection}(X_o, L_o, g(\Theta), \Theta^*) \)
   5. \( X_r \leftarrow X_s \backslash L_o \) \( \triangleright \) Remove \( L_o \) from \( L_s \)
   6. \( \Theta^*_r \leftarrow \{\Theta^*_s / \Theta^*_L\} \) \( \triangleright \) Remove \( \Theta^*_o \) from \( \Theta^*_s \)
   7. \( d(\Theta^*_r) \leftarrow IG, m(L_r, X_r) \leftarrow |L_r| \)
   8. \( \rho(L_r, X_r) \leftarrow (1 - \lambda)d(\Theta^*_r) + \lambda m(L_r, X_r) \) \( \triangleright \) Eq 4
   9. if \( \rho(L_r, X_r) \geq \rho(L_s, X_s) \) then
      10. break \( \triangleright \) Minimum value
   11. end if
   12. \( L_s \leftarrow L_r, X_s \leftarrow X_r, \Theta^*_s \leftarrow \Theta^*_r \)
   13. \( \rho(L_s, X_s) \leftarrow \rho(L_r, X_r) \)
   14. until it breaks out of the loop
   15. \( X_r \leftarrow \text{Poses not seeing any landmark } i \in L_s \)
   16. \( X_s \leftarrow \{X_s / X_r\} \) \( \triangleright \) Marginalize out \( X_r \)
   17. return \( \{X_s / L_s\} \)

**Algorithm 4 LandmarkSelection \((X, L, g(\Theta), \Theta^*)\)**

1. \( \min IG \leftarrow \infty \)
2. selectedLandmark \( \leftarrow \emptyset \)
3. for all landmarks \( L_i \) in \( L \) do
   4. \( IG(X; L_i) \leftarrow \text{InfoGain}(X, L_i, g(\Theta), \Theta^*) \)
   5. if \( IG(X; L_i) < \min IG \) then
      6. \( \min IG \leftarrow IG(X; L_i) \)
      7. selectedLandmark \( \leftarrow L_i \)
   8. end if
9. end for
10. return \( \{\text{selectedLandmark}, \min IG\} \)

During each iteration of the active minimization algorithm the least informative landmark is removed until Equation 4
has reached its minimum value. A landmark is removed by deleting the landmark and the corresponding factors from the graph. The least informative landmark is selected using Algorithm 4. It evaluates the mutual information between each landmark and rest of the states by computing the difference between log of covariance determinants before and after removing the landmark (Algorithm 2). Additional results comparing various landmark selection algorithms can be found at http://www.cc.gatech.edu/~choudhar/icra15/supplementary.pdf.

When Equation 4 has reached its minimum value, robot poses that do not see those landmarks are marginalized out. The optimal trajectory and landmark location is re-estimated using the reduced set of landmarks and poses. This is a batch algorithm.

Landmarks that are currently being measured have low information gain and can be removed by the active minimization algorithm. However these landmarks can become important in the future. To avoid throwing away the landmarks that are being measured right now, we do not consider landmarks seen by the last $P$ poses during the active minimization process. We use $P = 5$ during the experiments to ensure a reasonable pose lag.

Algorithm 3 removes only one landmark at a time until the value of the objective function has reached the minimum. One possible improvement to this algorithm is to remove a cluster of landmarks instead of one landmark at a time [15].

The current algorithm requires all the landmarks and poses to be available before running the minimization as a batch process. Below we give an incremental version of this algorithm.

D. Incremental Minimization

To be used in a SLAM framework, landmark and pose minimization has to be done in an incremental manner. We propose an incremental minimization algorithm where active minimization (Algorithm 3) is performed after every $Ψ$ poses of running SLAM. The reduced set of poses and landmarks are used from that point forward instead of using all landmarks and poses. The incremental minimization algorithm is shown in Algorithm 5.

![Algorithm 5 Incremental Minimization($Ψ, λ$)]

```
1: Initialize $L \leftarrow \emptyset$, $X \leftarrow \emptyset$, $Θ^\ast \leftarrow \emptyset$, $g(Θ) \leftarrow \emptyset$
2: repeat
3:   Add a new pose to $X$ and new landmarks to $L$.
4:   Add the corresponding new factors to $g(Θ)$
5:   $Θ^\ast \leftarrow \arg \min_{Θ^\ast} \left(-\log g(Θ)\right)$ // Eq: 2
6: if poses added since last minimization $> Ψ$ then
7:   $\{X, L\} = \text{Active Minimization} (X, L, Z, Θ^\ast, g, λ)$
8: end if
9: until the robot is navigating
```

Incremental minimization iteratively throws away less informative landmarks while using the remaining landmarks to navigate through the environment. It is less optimal and is greedy as compared to active minimization since active minimization considers all the landmarks and poses present till that moment to make the decisions where as incremental minimization keeps throwing away landmarks and does not have access to all the landmarks available at a particular time. For example, incremental minimization algorithm can throw away landmark which was uninformative at time $T$ but which might result in a loop closure constraint at $T + t$ and hence become very informative in the future. A possible solution to solve the optimality issue is to maintain a uniform distribution of landmarks over the explored region regardless of their information gain. Higher value of $Ψ$ (calling active minimization less frequently) will result in the estimation given by incremental minimization to approach the estimation given by active minimization.

V. EXPERIMENTAL RESULTS

In this section, we describe the datasets and evaluation metrics used in the experiments and compare reduced landmark-based SLAM algorithm with the standard graph SLAM (SAM [6]) algorithm (using all landmarks and poses).

A. Datasets

We tested the reduced landmark-based SLAM algorithm on two datasets: Victoria Park dataset [11] and Synthetic dataset. Table I summarizes the datasets used and Figure 4 show the datasets. Both the datasets are planar where each pose is represented by its translation $(x, y)$ and heading direction $θ$.

![Table I: Statistics of the datasets used.](image)

**Table I: Statistics of the datasets used.** $L$ is #landmarks, $P$ is #poses and $Z$ is #measurements.

B. Evaluation Metrics

1) Absolute trajectory error (ATE (m)) : As proposed by Sturm et al. [22], absolute trajectory error compares the absolute distance between the trajectories estimated using standard graph SLAM and reduced landmark-based SLAM. ATE evaluates the RMSE (root mean squared error) over the difference of pose translation (for the reduced set of poses) estimated using all landmarks and poses and reduced set of landmarks and poses. Formally ATE is defined as follows,

$$ATE = \left( \frac{1}{n} \sum_{i=1}^{n} \left\| \text{trans} (X_i^1 - X_i^2) \right\|^2 \right)^{\frac{1}{2}}$$ (8)
where $X^i$ is the $i^{th}$ pose estimated using all landmarks and poses and $X^i_\text{RED}$ is the $i^{th}$ pose estimated using reduced set of landmarks and poses.

2) **Absolute landmark error (ALE (m))**: Similar to ATE, absolute landmark error computes the RMSE over the reduced set of landmarks.

$$ALE = \left( \frac{1}{n} \sum_{i=1}^{n} ||L^i - L^i_\text{RED}||^2 \right)^{\frac{1}{2}}$$  \hspace{1cm} (9)

3) **Average rotation error (ARE (deg))**: Average rotation error is evaluated by averaging the angular difference in the pose heading directions over the reduced set of poses.

$$ARE = \frac{1}{n} \sum_{i=1}^{n} ||\text{rot} \left( X^i - X^i_\text{RED} \right)||$$ \hspace{1cm} (10)

4) **Eigen Envelope (EIG)**: As proposed by Carlevaris-Bianco et al. [2], we compute the range of eigen values of the difference between the marginal covariances of the reduced set of poses and landmarks when using all landmarks and poses as compared to using reduced number of landmarks and poses.

$$EIG = \text{range} \left( \text{eig} \left( \Sigma_\text{TRUE} - \Sigma_\text{REDUCED} \right) \right)$$ \hspace{1cm} (11)

5) **Percentage increase in covariance determinant of the latest pose (UD)**: Uncertainty of the latest pose is computed by marginalizing out the latest pose and evaluating its covariance determinant. We compute the percentage increase in uncertainty of the latest pose when using all landmarks and poses as compared to using reduced number of landmarks and poses.

$$UD = \frac{\|\Sigma^L_\text{REDUCED}\| - \|\Sigma^L_\text{TRUE}\|}{\|\Sigma^L_\text{TRUE}\|} \times 100$$ \hspace{1cm} (12)

C. **Results**

We compare the two components of the reduced landmark-based SLAM algorithm, active minimization and incremental minimization with standard graph SLAM algorithm. The minimization algorithms are analyzed by comparing the pose trajectory and landmark location estimated by the reduced set of landmarks and poses as compared to the estimation by standard SLAM using all landmarks and poses. The accuracy of the solution is analyzed using the evaluation metrics given in Section V-B. In the following experiments we use $\lambda = 0.4$ as the weight factor to conserve more landmarks and $P = 5$ as the pose lag.

1) **Victoria Park dataset**: For Victoria Park dataset, we use $\Psi = 500$ as a pose interval at which active minimization is called in an incremental manner. Figure 5 plots the estimated pose trajectories and landmark location for Victoria Park dataset using standard graph SLAM and reduced landmark-based SLAM (active minimization and incremental minimization) at three different time intervals. Table II compares active minimization and incremental minimization algorithm w.r.t standard graph SLAM using the evaluation metrics for the trajectories shown in Figure 5. Figure 6 compares active minimization and incremental minimization algorithm using the evaluation metrics at every $500^{th}$ pose.

We can see from Figure 5 and Table II that the trajectory and landmark location estimated using standard graph SLAM is similar to the estimation given by reduced landmark-based SLAM. The final absolute trajectory error between the estimation given by reduced landmark-based SLAM (incremental minimization) as compared to standard graph SLAM is 2.3 m with a 50% reduction in the number of landmarks and 54% reduction in the number of poses. The absolute landmark error is 2.05 m as well. The average error in heading direction is 0.0205 degrees. As compared to incremental minimization, active minimization uses all the landmarks and poses and therefore has better accuracy of the estimated trajectory (Table II and Figure 6). This also shows the sub-optimal nature of the incremental minimization as compared to active minimization. Another reason of high errors in case of incremental minimization as compared to active minimization is because minor trajectory errors added initially in time propagate through out rest of the estimation as it progresses. The number of landmarks and poses selected by active minimization is more than the number of landmarks and poses selected by incremental minimization. Positive values of eigen envelope (EIG) show that the estimates are conservative. Percentage increase in uncertainty of the latest pose (UD) is reasonable for both active minimization and incremental minimization algorithms.

2) **Synthetic dataset**: For this dataset, we use $\Psi = 30$ as a pose interval at which active minimization is called in an incremental manner. Figure 7 plots the estimated pose trajectories and landmark location for Synthetic dataset using standard graph SLAM and reduced landmark-based SLAM (active minimization and incremental minimization) at two different time intervals. Table III compares active minimization and incremental minimization algorithm w.r.t standard graph SLAM using the evaluation metrics.

We can see in Table III that the ATE and ALE of the estimation given using reduced landmark-based SLAM as compared to using standard graph SLAM is 0.2231 m and 0.3365 m with a 39% reduction in the number of landmarks. Since after removing uninformative landmarks there is no pose that doesn’t see any landmark, we do not marginalize out any pose and therefore there is no reduction in the number of poses. The average error in the heading direction is 0.0158 degrees. For the synthetic dataset the trajectory estimated using active minimization is the same as the trajectory and landmark location estimated using incremental minimization since there are no loop closures. In the presence of loop closures, landmarks which are less informative at certain point of time can become informative in the future. Percentage increase in the uncertainty of latest pose (UD) is low for active minimization algorithm (0.08%), whereas it is high for incremental minimization algorithm (16.9%) as compared to full SLAM. This is due to sub-optimal removal of landmarks in the case of incremental minimization.
TABLE II: Statistics of Minimization algorithms as compared to Standard Graph SLAM for Victoria Park dataset. L is #landmarks. P is #poses. ATE is absolute trajectory error, ALE is Absolute landmark error, ARE is average rotation error, UD is percentage increase in covariance determinant of last pose and EIG is eigen envelope. [%] is percentage change.

<table>
<thead>
<tr>
<th>L</th>
<th>P</th>
<th>Active Minimization</th>
<th>Incremental Minimization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>L [%]</td>
<td>P [%]</td>
</tr>
<tr>
<td>29</td>
<td>522</td>
<td>5 [13%]</td>
<td>508 [42%]</td>
</tr>
<tr>
<td>68</td>
<td>2532</td>
<td>58 [14%]</td>
<td>1345 [47%]</td>
</tr>
<tr>
<td>82</td>
<td>4532</td>
<td>63 [23%]</td>
<td>2359 [48%]</td>
</tr>
<tr>
<td>139</td>
<td>6532</td>
<td>74 [46%]</td>
<td>2971 [54%]</td>
</tr>
</tbody>
</table>

TABLE III: Statistics of Minimization algorithms as compared to Standard Graph SLAM for Synthetic dataset. L is #landmarks. P is #poses. ATE is absolute trajectory error, ALE is absolute landmark error, ARE is average rotation error, UD is percentage increase in covariance determinant of last pose and EIG is eigen envelope. [%] is percentage change.

<table>
<thead>
<tr>
<th>L</th>
<th>P</th>
<th>Active Minimization</th>
<th>Incremental Minimization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>L [%]</td>
<td>P [%]</td>
</tr>
<tr>
<td>6</td>
<td>322</td>
<td>15 [16%]</td>
<td>52 [10%]</td>
</tr>
<tr>
<td>15</td>
<td>622</td>
<td>15 [10%]</td>
<td>62 [10%]</td>
</tr>
<tr>
<td>23</td>
<td>962</td>
<td>18 [22%]</td>
<td>96 [10%]</td>
</tr>
</tbody>
</table>

Fig. 5: Estimated pose trajectory and landmark location for Victoria Park dataset using standard graph SLAM and reduced landmark-based SLAM at four different time intervals. P is #poses. The algorithms compared here are standard graph SLAM (using all landmarks and poses), Active minimization (Alg: 3) and Incremental minimization (Alg: 5).

Fig. 6: Comparison minimization algorithms w.r.t varying number of poses for Victoria Park dataset. The algorithms are compared on the basis of number of landmarks and poses remaining after minimization, absolute trajectory error (Section V-B.1) and absolute landmark error (Section V-B.2).
Fig. 7: Estimated pose trajectory and landmark location for Synthetic dataset using Standard Graph SLAM and reduced landmark-based SLAM for different trajectories. P is #Poses. The algorithms compared here are Standard Graph SLAM (using all landmarks and poses), Active minimization (Algorithm 3) and Incremental minimization (Algorithm 5). Green ellipses represent pose covariances. Blue ellipses represent landmark covariances.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we developed an information theoretic algorithm to efficiently reduce the number of landmarks and poses without compromising the estimation accuracy. We proposed an incremental version of the algorithm resulting in reduced landmark-based SLAM algorithm. We showed the results on Victoria Park dataset and Synthetic dataset and compared trajectory and landmark locations estimated using standard graph SLAM (SAM [6]) and reduced landmark-based SLAM algorithm showing a reduction of 40-50% landmarks and around 55% poses with minimal estimation error as compared to standard SLAM algorithms.

As a future work, we are working on a parallel minimization and mapping algorithm where active minimization can be run in parallel to the standard graph SLAM algorithm.

REFERENCES