

II. THE FULL SLAM PROBLEM

Below we review how the full SLAM problem reduces to solving a succession of linear least-squares problems, following [4]. In particular, inference in the full SLAM problem corresponds to the following *non-linear* least-squares problem for the unknown robot poses $\{x_i\}$ and landmarks $\{l_j\}$:

$$\sum_{i=1}^M \|x_i - f_i(x_{i-1}, u_i)\|_{\Lambda_i}^2 + \sum_{k=1}^K \|z_k - h_k(x_{i_k}, l_{j_k})\|_{\Sigma_k}^2 \quad (1)$$

Here $\|\cdot\|_{\Sigma}^2$ denotes the squared Mahalanobis norm with covariance Σ . This assumes, as standard in the SLAM literature [9]–[11], that we use Gaussian process and measurement models,

$$x_i = f_i(x_{i-1}, u_i) + w_i \quad z_k = h_k(x_{i_k}, l_{j_k}) + v_k \quad (2)$$

where $f_i(\cdot)$ is the *motion model*, $h_k(\cdot)$ is the *measurement model*, and w_i and v_k are normally distributed zero-mean noise terms with covariance matrices Λ_i and Σ_k . We assume both known control inputs $\{u_i\}$ and data association $\{i_k, j_k\}$.

Non-linear optimization methods solve a succession of linear approximations to (1) by Taylor expansions around a linearization point (x^0, l^0) . In particular, the first-order linearized version of the process model in (2) is given by

$$x_i^0 + \delta x_i = f_i(x_{i-1}^0, u_i) + F_i^{i-1} \delta x_{i-1} + w_i \quad (3)$$

with F_i^{i-1} the Jacobian of $f_i(\cdot)$ at the value x_{i-1}^0 . The linearized measurement equations are obtained similarly,

$$z_k = h_k(x_{i_k}^0, l_{j_k}^0) + H_k^{i_k} \delta x_{i_k} + J_k^{j_k} \delta l_{j_k} + v_k \quad (4)$$

with $H_k^{i_k}$ and $J_k^{j_k}$ the Jacobians of $h_k(\cdot)$ with respect to a change in x_{i_k} and l_{j_k} , evaluated at $(x_{i_k}^0, l_{j_k}^0)$. We now obtain a linear least-squares problem in the variables $\theta \triangleq \{\delta x_i, \delta l_j\}$,

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^M \|F_i^{i-1} \delta x_{i-1} + G_i^i \delta x_i - a_i\|_{\Lambda_i}^2 + \sum_{k=1}^K \|H_k^{i_k} \delta x_{i_k} + J_k^{j_k} \delta l_{j_k} - c_k\|_{\Sigma_k}^2 \quad (5)$$

where we defined the constants $a_i \triangleq x_i^0 - f_i(x_{i-1}^0, u_i)$, $c_k \triangleq z_k - h_k(x_{i_k}^0, l_{j_k}^0)$, and $G_i^i = -I$. Below we will assume that we are working on one of these linearized problems and hence drop the δ notation in the interest of clarity.

A. Sparse Linear Algebra View

Equation (5) can be written concisely using matrix notation, by collecting the Jacobian matrices into the (sparse) matrix A , and the vectors a_i and c_k into a right-hand side vector b ¹:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \|A\theta - b\|_2^2 \quad (6)$$

Figure 2 illustrates the typical block-structured sparsity of A where the non-zero blocks correspond to the individual measurement Jacobians F , G , H , and J as defined above.

¹and pre-multiplying the rows of A and entries of b to reflect the Mahalanobis norm in the case of non-isotropic noise.

Solving linear systems of the form (6) is normally done either through sparse QR factorization of the measurement Jacobian A (numerically stabler), or Cholesky factorization of the *information matrix* $\mathcal{I} \triangleq A^T A$ (faster) [6], yielding the algorithms below:

Algorithm 1 Solving the sparse linear system (6).

QR	Cholesky
1. Factor $A \rightarrow QR$	1. Factor $A^T A \rightarrow R^T R$
2. Equate $\begin{bmatrix} d & e \end{bmatrix}^T = Q^T b$	2. Solve $R^T y = A^T b$
3. Solve $R\theta^* = d$	3. Solve $R\theta^* = y$

The key question considered in this paper regarding these algorithms concerns their computational complexity. The amount of computation needed for the factorization and back-substitution is determined by the number of non-zero elements (NNZ) in the initial matrix A , as well as the NNZ in R . While we cannot influence the former, it is well known [4] that one *can* influence the NNZ introduced by the factorization *by reordering the columns of the matrix A* , leading to *Alg. 2*:

Algorithm 2 Iterative least-squares with reordering

- Reorder $A_\pi \leftarrow^\pi A$
- *Symbolic* Factorization of the sparsity pattern of A
- **Repeat**
 - 1) Linearize
 - 2) *Numeric* Factorization of A
 - 3) Back-substitution
- **Until** convergence
- Backorder solution $\theta \leftarrow^{\pi^{-1}} \theta_\pi$

B. Graphical View of Matrix Factorization

In the following we attempt to give the reader an intuition into the factorization process from a graphical point of view. In particular, let the graph G be constructed as in Section I such that the vertices V represent the variables and the edges E represent the dependencies between them. Then the adjacency matrix of the graph G corresponds exactly to the sparse block structure of the information matrix $\mathcal{I} \triangleq A^T A$ [4]. *Elimination* of the matrix $A^T A$ is defined as recursively eliminating one variable at a time, until there is only one variable left. By *eliminating* we mean expressing this variable as a linear combination of adjacent vertices. However, eliminating a variable v_i introduces dependencies in the remaining graph, forcing one to connect every variable adjacent to v_i into a completely connected subgraph or *clique*. We refer the interested reader to [4], [8], [12] for more detailed explanations.

In matrix terms, the extra edges thus introduced correspond to *fill-in* in the Cholesky factor R . The number of edges introduced by the elimination depends on the order in which the variables are eliminated. Thus, we wish to find the ordering that results in the least fill-in.

III. COMPLEXITY OF TYPICAL SLAM SCENARIOS

For a few typical scenarios it is not too difficult to derive complexity bounds. In particular, if either the number of poses M or the number of landmarks N is bounded, we obtain linear algorithms in $n \triangleq M+N$. Likewise, in a continual exploration mode the complexity of Smoothing and Mapping is linear.

A. An Environment of Bounded Size

In the case that the environment is of bounded size, the complexity of SAM is $O(M)$, where M is the number of robot poses. One of the objectives of mobile robotics is to deploy robots for an extended time in relatively small environments. The canonical examples are museum tour guides [13], some of which have been deployed for a period of years [14].

In this case, with N constant and $M \gg N$, we can first eliminate all poses with a bounded work per pose, as the number of landmarks adjacent to each pose is bounded by N (typically much smaller). After eliminating all poses, the landmarks will in general be completely connected, and hence we will have to factorize a dense matrix of $O(N)$ size. Hence, the complexity of this scenario is $O(M + N^3)$. However, as N is assumed constant, the final complexity is $O(M)$.

Note that this strategy of first eliminating all poses is a well known heuristic in the structure from motion literature, where it is known as the ‘‘Schur complement trick’’ [15], [16].

B. A Fixed Sensor Network

Likewise, in the case that M is bounded and $N \gg M$, the complexity of SAM is bounded by $O(N)$ by an analogous argument. One application where this occurs is when a team of robots is deployed to fixed positions after which they continually monitor the environment, e.g. tracking people in their field of view. These then play the role of ‘‘landmarks’’, and the SAM algorithm ‘‘calibrates’’ the deployed sensor network while at the same time yielding all people tracks. As $N \gg M$, a good ordering strategy is to first eliminate all ‘‘landmarks’’, then factorize the fully connected $O(M)$ sensor network, for a complexity of $O(M^3 + N) = O(N)$ with M constant.

C. Continual Exploration with no Loop Closures

Finally, if a robot is in a continual exploration mode without performing any loop-closures, the complexity of SAM is again linear. For example, this is a typical case when the robot is traversing a long indoor hallway or other linear structure. Given a bounded sensor range d , it is obvious that in this case the number of landmarks seen from any pose is bounded, and vice versa any landmark is only seen from a bounded number of poses. As a result, and in the absence of any loops, if the variables are ordered chronologically the information matrix I will be *band-limited*.

It is well known that factorization in that case is linear. However, in Section VI we give an alternative proof based on a recursive decomposition of the measurement graph.

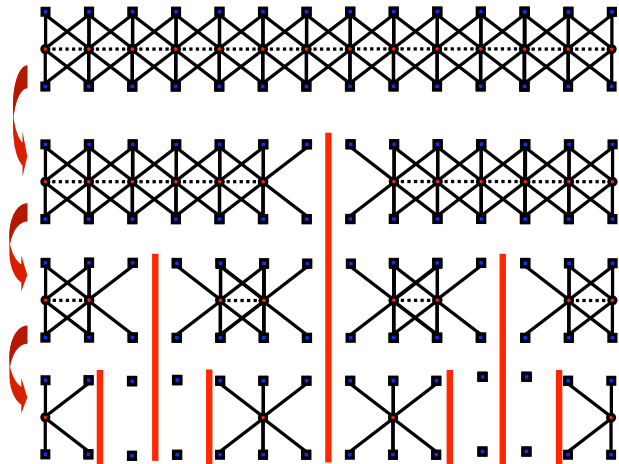


Fig. 3. An ND partitioning of an indoor hallway scenario.

IV. ORDERING HEURISTICS FOR GENERAL SCENARIOS

For *general* SLAM problems it is difficult to provide simple bounds. Finding an optimal elimination order is an NP-complete problem, which has long been known in the linear algebra and scientific computing community [12], [17]–[19]. Fortunately, a lot of useful heuristics have been developed to approximate an optimal ordering, most notably the *minimum degree* type algorithms (MD) [17], and *Nested Dissection* (ND) [8]. These two popular heuristics are empirically found to perform similarly in general SLAM scenarios.

A. Minimum Degree Heuristics

The two most widely used sparse matrix ordering algorithms for scientific computation are based on the heuristic of first eliminating the least constrained variables of G . This family of algorithms is known as *minimum degree* (MD) algorithms. A first approach is to eliminate all variables of minimal degree in one call of the elimination function, multiple (elimination) MD (MMD), while also eliminating *indistinguishable* nodes in the same step. MMD saves time both on updating the graph and determining the next elimination candidates [20]. A second approach avoids computing the exact vertex degrees when eliminating one or more variables, by collecting nodes into cliques. Only the degrees for the cliques are calculated, thus reducing the bookkeeping effort. This algorithm is known as *Approximate Minimum Degree* (AMD) [17], [21].

Both AMD and MMD produce equally good orderings, but AMD is faster. It was shown in [4] that reordering according to an AMD heuristic applied to the block-structure of the problem, can drastically reduce the factorization cost for SAM.

B. Nested Dissection

The Nested Dissection (ND) ordering algorithm follows a *divide-and-conquer* paradigm. An indoor hallway example of this recursive process is shown in Fig. 3. ND algorithms recursively partition the measurement graph and return a *post-fix* notation of the partitioning tree as the ordering.

Algorithm 3 Nested Dissection

Let $G = (V, E)$ be a graph, with a set of vertices V and a set of edges E and $|G| = n$.

- 1) Partition G into subgraphs A, B and *separator* C
 - 2) Repeat *Step (1)* until $|A|, |B| \leq \epsilon$ or $|A|, |B| = 1$
 - 3) Obtain the ordering by putting the binary tree of the recursive partitioning in post-order, with the nodes of the separating set C last for every triple of sets.
-

The pseudo-code for a general ND scheme is given above. Almost all ND variants use a two step approach for determining the separators C in the graph. First, they try to find good areas for a cut that preserve the balance between the induced subgraphs A and B . Second, a refinement algorithm like [27] or [28] is applied. These can be understood as variants of bipartite graph matching algorithms as they aim at finding the minimal cut between a set of nodes.

The application of ND orderings is especially relevant to SLAM, as environments mapped by robots often contain parts that are spatially separable. In conventional EKF SLAM based methods this property of the SLAM problem is exploited by using “submaps” [22]–[26]. Thus, a *divide-and-conquer* approach appears very promising.

C. Comparison of AMD and ND Heuristics

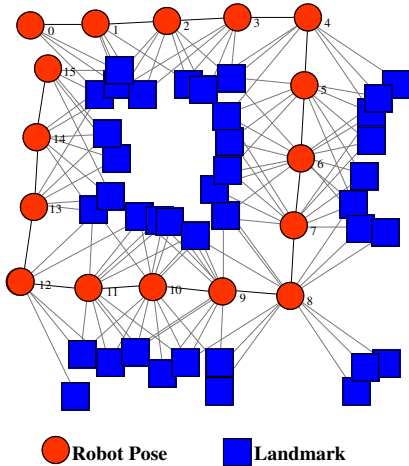
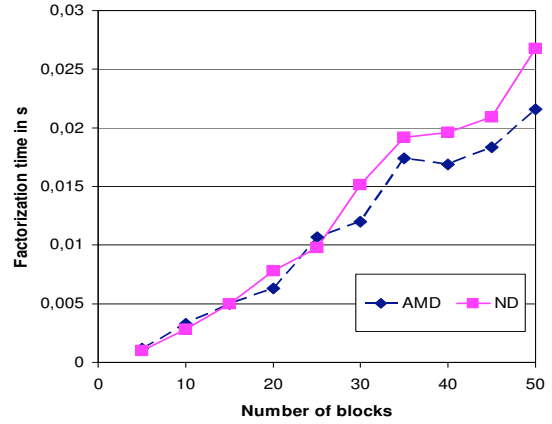


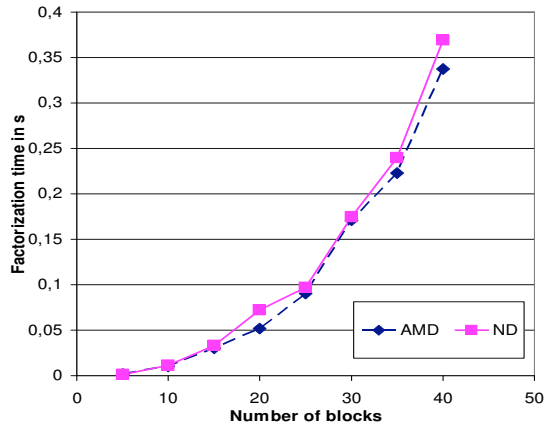
Fig. 4. Resulting measurements from a simulated walk with 15 poses.

We compared the performance of ND to AMD reorderings using indoor scenario simulations, showing that ND is about as efficient as AMD on general scenarios. The implementations used were for AMD [21], [29] and for ND [19].

The simulation data for the comparison was produced using simulated block-worlds. The result of a loop around one block can be seen in Figure 4. It takes the robot 4 steps to pass a block. At each step the robot senses between 8 and 12 landmarks. For simplicity we count the number of measurements in number of blocks. For example this means that for a path straight down a hallway with 1,000 blocks to



(a) Straight walk with blocks to the left and right



(b) Random walk through a square block world

Fig. 5. Factorization times vs. number of blocks. In (b) the number of blocks denotes the number of blocks of one side of the square block world, and the robot travels for a distance of n^2 blocks.

the left and right a measurement matrix with about 4,000 rows and about 40,000 non zero entries is obtained.

In Fig. 5 the factorization times for varying numbers of measurements are depicted. Fig. (a) shows the factorization times for such a straight hallway path for varying numbers of blocks. Fig. 5 (b) shows factorization times for simulated *random* walks in $n \times n$ block worlds of varying sizes. All results show that the factorization of SLAM measurement matrices are roughly the same for AMD and ND reorderings.

V. COMPLEXITY OF NESTED DISSECTION

A significant advantage of the nested dissection scheme is that it lends itself to a rigorous analysis. While AMD holds a slight performance edge over ND in our experiments, only the latter allows us to establish bounds on the complexity of SLAM. Below we review the seminal work in this area.

The key to the analysis of ND is the existence of so-called *separator theorems*. The basis of any ND algorithm

is the ability to recursively partition graphs into subgraphs of roughly equal size with a small separating subgraph. It is crucial that these *separators* be small as this keeps the fill-in in the factorization process low, by maximizing the number of nodes that are mutually independent.

A. Separator Theorems

Lipton et al. [7], [8] formalized the existence of small separators through the $f(n)$ -separator theorem:

Theorem 1: ($f(n)$ -SEPARATOR THEOREM) *A graph G is $f(n)$ -separable if there exist constants $\alpha < 1$ and $\beta > 0$ such that the vertices of G can be partitioned into three sets A, B, C in a way that no edge joins a vertex in A with one in B , A nor B contains more than αn vertices, and C contains no more than $\beta f(n)$ vertices.*

For all classes of graphs for which a $f(n)$ -separator theorem holds an efficient *divide-and-conquer* ordering can be found. Note that one needs to guarantee that the algorithms for the graph partitioning and local subgraph elimination are less complex than the factorization itself.

B. From Separator Size to Complexity

The link between separator size and the complexity of ND is given by the following theorem from [8]:

Theorem 2: (GENERALIZED NESTED DISSECTION) *Let S be any class of graphs closed under the subgraph criterion for which an n^σ separator theorem holds, with $\sigma > 0.5$. Then for any n -vertex graph G in S , there is an elimination ordering with $O(n^{2\sigma})$ fill-in size and $O(n^{3\sigma})$ multiplication count.*

Thus we can derive complexity bounds for the factorization. These bounds depend on how tight $f(n) = n^\sigma$ is. Hence, in order to use this theorem we *only* need to show that there exists a graph partitioning algorithm fulfilling an $f(n)$ -separator theorem which is less complex than $O(n^{3\sigma})$.

VI. AN $O(n)$ BOUND FOR CONTINUAL EXPLORATION

An even stronger theorem from [8] will yield the alternative proof of linear-time SAM for continual exploration:

Theorem 3: (LINEAR NESTED DISSECTION) *Let S be any class of graphs closed under the subgraph criterion for which an n^σ separator theorem holds, with $\sigma < 1/3$. Then for any n -vertex graph G in S , there is an elimination ordering with $O(n)$ fill-in size and multiplication count.*

Thus, it suffices to prove that in the exploration case, we can always cut the graph such that the separator size is bounded. Let G be a measurement graph with M poses and N landmarks, and denote by M_C and N_C the number of poses and landmarks, respectively, in an arbitrary connected subgraph C of G . Let us make the following assumptions:

- 1) For every connected subgraph C in G we have a constant ratio $\phi = \frac{N}{M} = \frac{N_C}{M_C}$.
- 2) Every landmark is seen from a constant number of consecutive poses $d \geq 1$
- 3) We use an omni-directional sensor with limited range

Assumptions (2) and (3) imply that closely located landmarks will be observed from the same set of poses. For a

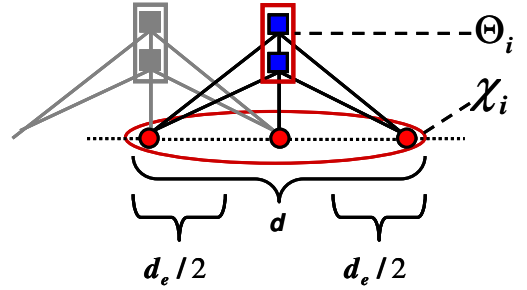


Fig. 6. An abstract measurement graph. Φ_i is the set of landmarks seen from pose x_i , while χ_i is the set of poses seeing landmarks in Φ_i is marked.

pose x_i we define the closest ϕ landmarks by Φ_i . The order of the measurements from x_i is indexed by k . All landmarks $l_{ik} \in \Phi_i$ are seen from the same set of poses χ_i symmetric around x_i . Intuitively $\chi_i = \{x_j \mid \exists l_{ik} \in \Phi_i : (l_{ik}, x_j) \in E\}$ is the set of poses that are connected via a measurement in G to a landmark in Φ_i . According to assumption (3) an edge has a maximum length d . This translates into the graphical model by landmarks l_{ik} only being observed by the set of poses $\{x_{i-\lfloor d/2 \rfloor}, \dots, x_{i+\lfloor d/2 \rfloor}\}$ —given an ideal sensor and d being odd. We define

$$d_e = \psi(d) = \begin{cases} d-1 & \text{if } d \text{ is odd} \\ d & \text{else} \end{cases}$$

to reduce the even case to the symmetric odd case - with $d_e/2$ poses left and to the right of the i -th pose sensing Φ_i . Using assumption (2) we can now refine our above statement to $\chi_i = \{x_j \mid |j-i| \leq (d_e/2)\}$.

Lemma 1: (CONTINUAL EXPLORATION SEPARATOR) *Given a SLAM measurement graph $G = (E, V)$ fulfilling the assumptions (1)-(3) the subgraph $C_i = \Phi_i \cup \chi_i$ of G with $|C_i| = \phi + d$ separates G into subgraphs A, B such that no vertex in A is adjacent to a vertex in B .*

Proof: imagine a measurement graph of a hallway. We wish to cut this graph next to a pose x_i . In order to separate the pose chain adding x_i to the separator is sufficient. But due to the pose-landmark edges there exist connections from one subgraph to the other if $C = \{x_i\}$. The outreach is bound topologically by d . Thus with the definitions from above, no landmark in the set Φ_i will be connected to poses other than $\{x_{i-d_e/2}, \dots, x_{i+d_e/2}\}$. By extending the separator to $C_i = \Phi_i \cup \chi_i$ we can guarantee a separation, thus $|C_i| = \phi + d$. ■

VII. AN $O(n^{1.5})$ BOUND ON LARGE-SCALE SLAM

For a special class of large-scale SLAM problems, in which the measurement graph has a planar meta-structure, we can prove an $O(n^{1.5})$ bound. We will do so by first examining the typical SLAM measurement graphs at the fine level and then continuing with the *meta* structure of the graph. Using the complexity bounds for the ND algorithm on planar graphs [7], [8] we can then bound the complexity for a large class of SLAM graphs.

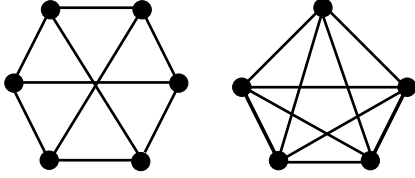


Fig. 7. It is sufficient for a graph to have one of the above graphs as a subgraph to be non-planar (a complete bipartite and a Kuratowski graph).

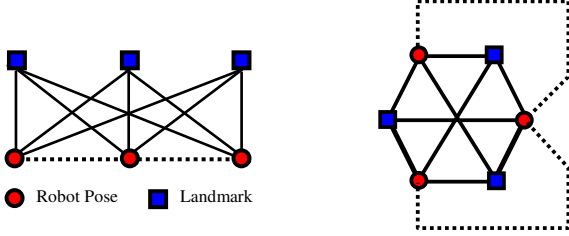


Fig. 8. A typical measurement pattern in SLAM which is isomorphic to a non-planar graph containing a fully connected bipartite graph as in Fig. 7.

A. Nested Dissection for Planar Graphs

Planar graphs are an important class of graphs, and frequently arise in large physics and mechanical simulations. A graph is planar iff it contains neither a complete bipartite graph on two sets of three vertices nor a complete graph on five vertices (*Kuratowski-Theorem*, Figure 7).

Planar graphs satisfy an \sqrt{n} -separator theorem [7]:

Theorem 4: (\sqrt{n} -SEPARATOR THEOREM) *Any n -vertex planar graph G satisfies an $f(n)$ -separator theorem with $f(n) = \sqrt{n}$, $\alpha = \frac{2}{3}$, and $\beta = 2\sqrt{2}$. In addition, the partitions can be found in $O(n)$ time.*

In [8] the above \sqrt{n} -separator theorem was used to prove bounds on the complexity of matrix factorizations when the matrix structure equals a planar graph:

Theorem 5: (PLANAR NESTED DISSECTION) *Let G be any planar graph. Then G has an elimination ordering which produces a fill-in of size $c_1 n \log n + O(n)$ and a multiplication count of $c_2 n^{3/2} + O(n \log^2 n)$, where $c_1 \leq 129$ and $c_2 \leq 4002$. Such an ordering can be found in $O(n \log n)$ time.*

For planar graphs satisfying an \sqrt{n} -separator theorem, [7], [8] give the following result:

Theorem 6: [8] *Let S be any class of graphs satisfying a \sqrt{n} -separator theorem then any graph $G \in S$ has an elimination ordering which produces a fill-in of size $c_1 n \log n + O(n)$ and a multiplication count of $c_2 n^{3/2} + O(n \log^2 n)$, where*

$$c_1 = \beta^2 \left(\frac{1}{2} + 2\sqrt{\alpha} / (1 - \sqrt{\alpha}) \right) / \log_2(1/\alpha)$$

$$c_2 = \beta^2 \left(\frac{1}{6} + \beta\sqrt{\alpha}(2 + \sqrt{\alpha} / (1 + \sqrt{\alpha})) + 4\alpha / (1 - \alpha) \right) / (1 - \sqrt{\alpha}) / (1 - \delta)$$

with $\delta = \alpha^{\frac{3}{2}} + (1 - \alpha)^{\frac{3}{2}}$.

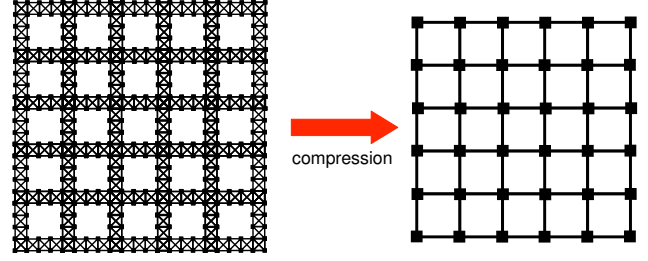


Fig. 9. Measurement graph for a typical indoor scenario (left) and the corresponding *meta graph* (right).

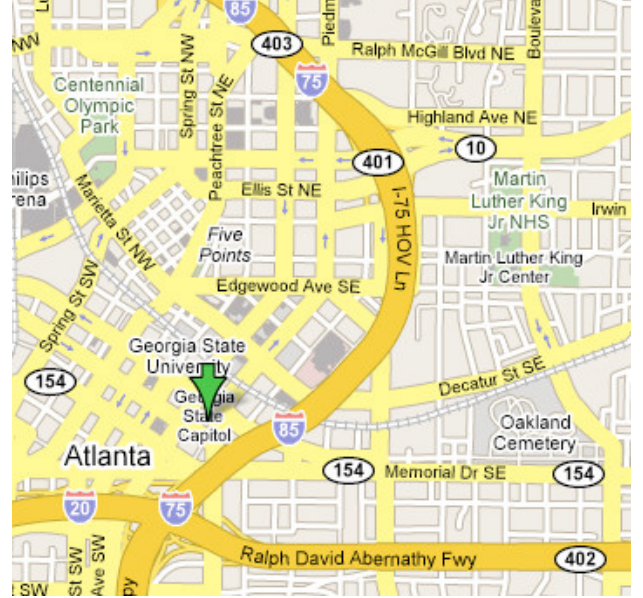


Fig. 10. Example scenario for an urban mapping example (Atlanta).

B. Compressing SLAM Graphs into a Meta Graph

Unfortunately, as shown in Figure 8, measurement graphs in SLAM are typically non-planar. The denser the measurement graph is, the *less* planar the measurement graph G will be.

However, in many realistic SLAM scenarios it is possible to recognize structure at a level higher than at the level of the detailed measurements. Imagine a robot exploring the interior of a building as in Fig. 9, or an extended area in large city as in Fig. 10. If we abstract from the fine graph structure in SLAM we obtain a coarser graph with vertices corresponding to intersections and edges corresponding to hallways or streets, as depicted in Fig. 9. In graph theoretic terms this process is called *compression*. We denote the result of the compression as the *meta graph*.

C. Complexity if the Meta Graphs is Planar

If a SLAM problem instance has a *planar meta graph*, defined as above, we can use Theorem 4 to determine the number of meta nodes in the separator as a function of the total number of meta nodes. Using bounds on the constituent parts of the fine graph we can then obtain a separator theorem for

this class of SLAM graphs for which - with changed constants - the complexity bounds of Theorem 5 hold, and hence also an $O(n^{3/2})$ bound on the multiplication count.

As the *meta* graph is planar the separator theorem for planar graphs holds and thus for any *meta cut* C' we have

$$|C'| \leq 2\sqrt{2N}$$

where $|C'|$ is the number of intersections in the cut and N the total number of intersections. If n_{max} is the maximum number of (fine) nodes in any intersection, then the size of the fine separator C is bound by

$$|C| \leq 2n_{max}\sqrt{2N} \quad (7)$$

In addition, let n_{min} be the *minimum* number of (fine) nodes in any intersection *or* edge and let the meta graph be connected. Then we have at least as many edges as nodes, and the total number of fine nodes n satisfies $2n_{min}N \leq n$. Dividing by n_{min} and taking the square root on both sides yields

$$\sqrt{2N} \leq \frac{\sqrt{n}}{\sqrt{n_{min}}}$$

Inserting this result into (7) yields the following bound on the size of any meta cut:

$$|C| \leq 2n_{max} \frac{\sqrt{n}}{\sqrt{n_{min}}} = \beta\sqrt{n} \text{ with } \beta = \frac{2n_{max}}{\sqrt{n_{min}}}$$

Hence, β is a constant in the sense of the $f(n)$ -separator-theorem as in [7].

Finally, from Section VI we know that the hallways/streets have a constant separator of size $k(\phi + d)$, where k is the maximum number of edge traversals. Then n also has to satisfy

$$k(\phi + d) \leq \beta\sqrt{n}$$

That can be guaranteed by stopping the ND recursion when

$$n < \frac{k^2(\phi + d)^2}{\beta^2} = \frac{k^2(\phi + d)^2 n_{min}}{4n_{max}^2}$$

Summarizing, we have derived a \sqrt{n} -separator theorem in the sense of [7], [8] and thus can use their results. Thus, we obtain the same complexity bounds as if the graph was planar, except for the constants c_1 and c_2 .

VIII. EXAMPLE SCENARIOS

In this section we demonstrate the partitioning process for typical scenarios as depicted in Fig. 9 or Fig. 10. We also show the impact of the changed constants.

We start with a grid block-world scenario with 50 nodes in the intersections and at least 50 nodes per hallway. This scenario is challenging as there are no “natural” partitions like rooms to the sides of a hallway and the graph is evenly dense. These might save cuts or keep the cuts small.

The example grid block-world and possible cuts according to our separator theorem are shown in Fig. 11. The partitionings correspond to the cuts of a depth-first partitioning recursion. Note that the measurement graph contains only parts where the robot passed along two times at maximum.

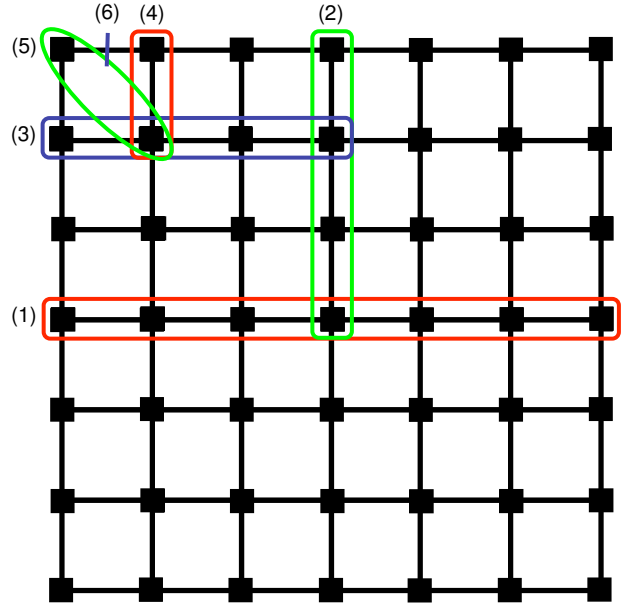


Fig. 11. Grid World Example

	n	\sqrt{n}	$\beta\sqrt{n}$	$ C $	$ A , B $
(1)	86,250	294	4153	350	42,950
(2)	42,950	207	2931	200	21,375
(3)	21,375	146	2068	200	10,588
(4)	10,588	103	1455	100	5,244
(5)	5,244	72	1024	100	2,572
(6)	2,572	51	717	18	1,277
(7)	1,277	36	505	18	629
(8)	629	25	355	18	306
(9)	306	17	247	18	144
(10)	144	12	170	18	63
(11)	63	8	112	18	22
(12)	22	5	67	18	2

TABLE I

THE LEVEL OF THE RECURSION, NUMBER OF NODES n IN THE EACH PARTITION, \sqrt{n} , THE SIZE OF THE NEXT PARTITIONS AND THE SEPARATOR, WITH 1,000 NODES PER HALLWAY AND 50 NODES PER INTERSECTION.

Assuming that the intersections contain twice as much nodes as a separator the 50 nodes correspond to a hallway separator with $d = 3$, $k = 2$ of at least $\phi = 3$. In Table I the relevant figures are shown for a grid with roughly 90,000 nodes. Note that after the final partition all nodes form a subgraph and the recursion stops.

We expand this scenario to the challenging outdoor urban mapping from Fig. 10. We assume 10,000 features per street, 1,000 features per intersection and a 20×20 block world. Table II contains the results one would obtain for the first 9 partitionings.

We want to highlight that the constants c_1 and c_2 in the derived theorem are not negligible. For the first scenario we obtain $\beta = \frac{2n_{max}}{\sqrt{n_{min}}} = 14,14$, thus $c_1 \approx 3,200$ and $c_2 \approx 499,700$. It is the subject of future work to empirically validate how tight these bounds are for different scenarios.

	n	\sqrt{n}	$\beta\sqrt{n}$	$ C $	$ A , B $
(1)	8,837,000	2973	188,011	21,000	4,408,000
(2)	4,408,000	2100	132,786	11,000	2,198,000
(3)	2,198,000	1483	93,776	11,000	1,093,750
(4)	1,093,750	1046	66,144	5,500	544,125
(5)	544,125	738	46,653	5,500	269,313
(6)	269,313	519	32,821	2,750	133,281
(7)	133,281	365	23,089	2,750	65,266
(8)	65,266	255	16,157	1,375	31,945
(9)	31,945	179	11,304	1,375	15,285
(10)	...				

TABLE II

NUMBER OF NODES n IN THE EACH PARTITION, \sqrt{n} , THE SIZE OF THE NEXT PARTITIONS AND THE SEPARATOR, WITH 10,000 NODES PER STREET AND 1,000 NODES PER INTERSECTION.

IX. CONCLUSION

In this paper we have shown that the key computation in the full SLAM problem is the factorization of a sparse matrix, and that the ordering of eliminating the unknowns dramatically changes the complexity of the resulting algorithm. We have explained how the nested dissection algorithm exploits the locality inherent in the SLAM problem, and have described its properties and complexity bounds in the context of SLAM. The contributions of this paper can be summarized as follows:

- For typical measurement patterns bounds on the separator sizes and an $f(n)$ – separator theorem can be derived.
- Hence, in contrast to the complexity of $O(n^3)$ for a dense matrix factorization we can obtain a complexity bound of $O(n \log n)$ for the total fill-in and $O(n^{\frac{3}{2}})$ for the multiplication count of the factorization of the pre-ordered matrix.

As a caveat, these bounds *can* be linked to high constant factors that need to be taken into consideration, as was illustrated by applying the algorithm in a typical indoor scenario. Nevertheless, this scenario was in many respects a difficult one and we expect that smaller constants will occur in real environments.

REFERENCES

- [1] U. Frese and T. Duckett, "A multigrid approach for accelerating relaxation-based SLAM," in *Proc. of the IJCAI-03 on Reasoning with Uncertainty in Robotics*, 2003.
- [2] M. Paskin, "Thin junction tree filters for simultaneous localization and mapping," in *Intl. Joint Conf. on Artificial Intelligence (IJCAI)*, 2003.
- [3] J. Folkesson and H. I. Christensen, "Graphical SLAM - a self-correcting map," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, vol. 1, 2004, pp. 383 – 390.
- [4] F. Dellaert, "Square Root SAM: Simultaneous location and mapping via square root information smoothing," in *Robotics: Science and Systems (RSS)*, 2005.
- [5] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT press, Cambridge, MA, 2005.
- [6] G. Golub and C. V. Loan, *Matrix Computations*, 3rd ed. Baltimore: Johns Hopkins University Press, 1996.
- [7] R. Lipton and R. Tarjan, "A separator theorem for planar graphs," *SIAM Journal on Applied Mathematics*, vol. 36, no. 2, pp. 177–189, 1979.
- [8] —, "Generalized nested dissection," *SIAM Journal on Applied Mathematics*, vol. 16, no. 2, pp. 346–358, 1979.

- [9] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in Robotics," in *Autonomous Robot Vehicles*, I. Cox and G. Wilfong, Eds. Springer-Verlag, 1990, pp. 167–193.
- [10] J. Leonard, I. Cox, and H. Durrant-Whyte, "Dynamic map building for an autonomous mobile robot," *Intl. J. of Robotics Research*, vol. 11, no. 4, pp. 286–289, 1992.
- [11] J. Castellanos, J. Montiel, J. Neira, and J. Tardos, "The SPMAP: A probabilistic framework for simultaneous localization and map building," *IEEE Trans. Robot. Automat.*, vol. 15, no. 5, pp. 948–953, 1999.
- [12] P. Heggernes and P. Matstoms, "Finding good column orderings for sparse QR factorization," in *Second SIAM Conference on Sparse Matrices*, 1996.
- [13] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Creemers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Probabilistic algorithms and the interactive museum tour-guide robot minerva," *International Journal of Robotics Research*, vol. 19, no. 11, pp. 972–999, November 2000.
- [14] I. Nourbakhsh, C. Kunz, and T. Willeke, "The Mobot museum robot installations: A five year experiment," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2003.
- [15] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – a modern synthesis," in *Vision Algorithms: Theory and Practice*, ser. LNCS, W. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer Verlag, 2000, pp. 298–375.
- [16] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [17] P. R. Amestoy, T. Davis, and I. S. Duff, "An approximate minimum degree ordering algorithm," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 4, pp. 886–905, 1996.
- [18] A. Gupta, G. Karypis, and V. Kumar, "Highly scalable parallel algorithms for sparse matrix factorization," *IEEE Trans. Parallel and Distributed Systems*, vol. 8, no. 5, pp. 502–520, 1997.
- [19] G. Karypis and V. Kumar, "Multilevel algorithms for multi-constraint graph partitioning," in *Supercomputing '98: Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM)*. Washington, DC, USA: IEEE Computer Society, 1998, pp. 1–13.
- [20] J. W. H. Liu, "Modification of the minimum-degree algorithm by multiple elimination," *ACM Trans. Math. Softw.*, vol. 11, no. 2, pp. 141–153, 1985.
- [21] T. Davis, J. Gilbert, S. Larimore, and E. Ng, "A column approximate minimum degree ordering algorithm," *ACM Trans. Math. Softw.*, vol. 30, no. 3, pp. 353–376, 2004.
- [22] H.-J. S. Feder and J. Leonard, "Decoupled Stochastic Mapping, Part II: Performance Analysis," submitted to *IEEE Transactions on Robotics and Automation*.
- [23] J. J. Leonard and H. J. S. Feder, "Decoupled stochastic mapping," *IEEE Journal of Oceanic Engineering*, pp. 561–571, October 2001.
- [24] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An Atlas framework for scalable mapping," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2003.
- [25] J. Nieto, J. Guivant, and E. Nebot, "The hybrid metric maps (hymms): A novel map representation for denseslam," in *IEEE International Conference on Robotics and Automation*, 2004.
- [26] U. Frese, "Treemap: An $O(\log n)$ algorithm for simultaneous localization and mapping," in *Spatial Cognition IV*. Springer Verlag, 2005, pp. 455–476.
- [27] B. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *The Bell System Technical Journal*, vol. 49, no. 2, pp. 291–307, 1970.
- [28] C. Fiduccia and R. Mattheyses, "A linear time heuristic for improving network partitions," in *Proc. 19th IEEE Design Automation and Conference*, 1982, pp. 175–181.
- [29] T. A. Davis, "Algorithm 8xx: a concise sparse Cholesky factorization package," Univ. of Florida, Tech. Rep. TR-04-001, January 2004, submitted to *ACM Trans. Math. Software*.