A Survey of Geodetic Approaches to Mapping and the Relationship to Graph-Based SLAM

Pratik Agarwal[1] Wolfram Burgard[1] Cyrill Stachniss[1,2]

[1] University of Freiburg, Institute of Computer Science, 79110 Freiburg, Germany
[2] University of Bonn, Institute of Geodesy and Geoinformation, 53115 Bonn, Germany

***Abstract*— The ability to simultaneously localize a robot and build a map of the environment is central to most robotic applications and the problem is often referred to as SLAM. Robotics researchers have proposed a large variety of solutions allowing robots to build maps and use them for navigation. Also the geodetic community addressed large-scale map building for centuries, computing maps which span across continents. These large-scale mapping processes had to deal with several challenges that are similar to those of the robotics community. In this paper, we explain key geodetic map building methods that we believe are relevant for robot mapping. We also aim at providing a geodetic perspective on current state-of-the-art SLAM methods and identifying similarities both in terms of challenges faced as well as in the solutions proposed by both communities. The central goal of this paper is to connect both fields and to enable future synergies between them.**

I. Introduction

The problem of simultaneously localization and mapping (SLAM) is essential for several robotic applications in which the robot is required to autonomously navigate. A mobile robot needs a map of its environment to plan appropriate paths towards a goal location. Furthermore, following the planned paths in turn requires the robot to localize itself in its map. Many modern SLAM methods follow the graph-based SLAM paradigm [1], [2], [3], [4]. In this approach, each pose of the robot or each landmark position is represented as a node in a graph. A constraint between two nodes, which results from observations, is represented by an edge in the graph. The first part of the overall problem is to create the graph, based on sensor data and such a system is often referred to as the front-end. The second part deals with finding the configuration of the nodes that best explains the constraints modeled by the edges. This step corresponds to computing the most-likely map (or the distribution over possible maps) and a system solving it is typically referred to as a back-end.

In the geodetic mapping community, one major goal has been to build massive survey maps, some even spanning across continents. These maps were supposed then be used either directly by humans or for studying large scale properties of the earth. In principle, geodetic maps are built in a similar way to the front-end/back-end approach used in SLAM. Constraints are acquired through observations between physical observation towers. These towers correspond to positions of the robot as well as the landmarks in the context of SLAM. Once the constraints between observation towers are obtained, the goal is to optimize the resulting system of equations to get the best possible locations of the towers on the surface of the earth.

The aim of this paper is to survey key approaches of the geodetic mapping community and put them in relation to recent SLAM research. This mainly relates to the back-ends of graph-based SLAM systems. We believe that this step will enable further collaborations and exchanges among both fields. As we will illustrate during this survey, both communities have come up with sophisticated approximations to the full least squares approach for computing maps with the aim of reducing memory and computational resources. A central starting point for the survey of the geodetic approaches is the "North American Datum of 1983" by Charles R.Schwarz [5] and we go back to the work by Helmert [6] published in 1880. This paper extends [7] and presents a comprehensive review of the geodetic mapping techniques that we believe are related to SLAM.

In the remainder of this paper, we first identify the key similarities between the challenges faced in geodetic and robotic mapping. Second, we introduce graph-based SLAM and provide a short overview on it. Third, we explain the problem formulation for geodetic mapping and introduce commonly used terminologies. We continue to explain the approaches to geodetic mapping including the motivation and insights central to the developed solutions. We then highlight the relationships between individual SLAM approaches and geodetic solutions.

II. Common Challenges in Geodetic and Robotic Mapping

SLAM and geodetic mapping have several problems in common. The first challenge is the large size of the maps. The map size in the underlying estimation problems is represented by the number of unknowns. In geodetic mapping, the unknown are the positions of the observation towers, while for robotics, the unknowns corresponds to robot positions and observed landmarks. For example, the system of equations for the North American Datum of 1927 (NAD 27) required solving for 25,000 tower positions and the North American Datum of 1983 (NAD 83) required solving for 270,000 positions [8]. The largest real-world SLAM datasets have up to 21,000 poses [9], while simulated datasets with 100,000 poses [10] and 200,000 poses [4] have been used for evaluation. At the time of NAD 27 and 83, the map building problems could not

be solved by standard least squares methods as no computer was capable to handle such large problems. Even nowadays, computational constraints are challenging in robotics. SLAM algorithms often need to run on real mobile robots, including battery-powered wheeled platforms, small flying robots, and low-powered underwater vehicles. For autonomous operation, the memory and computational requirements are often constrained so that approximate but online algorithms are often preferred over more precise but offline ones.

The second challenge results from outliers or spurious measurements. The front-ends for both, robotics and geodetic mapping, are affected by outliers and noisy measurements. In robotics, the front-end is often unable to distinguish between similar looking places, which leads to perceptual aliasing. A single outlier observation generated by the front-end can lead to a wrong solution, which in turn results in a map that is not suitable to perform navigation tasks. To deal with this problem, there recently has been a more intense research for reducing the effect of outliers on the resulting map by using extensions to the least squares approach [11], [12], [13], [14]. For geodetic mapping, the front-end consisted of humans carefully and meticulously acquiring measurements. However, even this process was prone to making mistakes [5].

The third challenge comes from the non-linearity of constraints, which is frequently the case in SLAM as well as geodetic mapping. A commonly used approximation is to linearize the problem around an initial guess. However, this approximation to the non-linear problem may lead to a suboptimal solution if the initial guess is not in the basin of convergence. There are various methods, which enable finding a better initial guess, but this still remains a challenge [15], [16]. The importance of the initial guess in SLAM has also motivated the study of the convergence properties of the corresponding nonlinear optimization problem [17], [18], [19], [20].

Fourth, both geodetic mapping and SLAM ideally require an incremental and online optimization algorithm. In robotics, a robot is constantly building and using the map. It is advantageous if the system is capable of optimizing the map incrementally [21], [22]. In geodetic mapping, new survey towers are build and new constraints are obtained as and when required. It is not feasible to optimize the full network from scratch when new areas or constraints are added. Thus, also geodetic methods must be able to incorporate new information into existing solutions with minimum computational demands.

Given these similarities, we believe that studying the achievements of the geodetic scholars is likely to inspire novel solutions to large-scale, autonomous robotic SLAM.

III. Graph-Based SLAM

In the robotics community, Lu and Milios [1] were the first who introduced a least squares-based direct method for SLAM. In their seminal paper, they propose the graph-based framework in which each node models a robot pose and each edge represents a constraint between the poses of the robot. These constraints can represent odometry measurements between sequential robot poses produced by wheel encoders and inertial measurement units or produced by sensor fusion techniques such as scan matching [23], [24], [25], [26].

Graph-Based SLAM back-ends aim at finding the configuration of the nodes that minimize the error induced by constraints. Let $\mathrm{x} = (\mathrm{x}_1, \ldots, \mathrm{x}_n)^T$ be the state vector where x_i describes the pose of node i. This pose x_i is typically three-dimensional for a robot living in the 2D plane. We can describe the error function $\mathbf{e}_{ij}(\mathrm{x})$ for a single constraint between the nodes i and j as the difference between the obtained measurement z_{ij} and the expected measurement $f(\mathrm{x}_i, \mathrm{x}_j)$ given the current state

$$\mathbf{e}_{ij}(\mathrm{x}) \quad = \quad f(\mathrm{x}_i, \mathrm{x}_j) - z_{ij}. \tag{1}$$

The actual realization of the measurement function f depends on the sensor setup. For pose to pose constraints, one typically uses the transformation between the poses. For pose to landmark constraints, we minimize the reprojection error of the observed landmark into the frame of the observing pose. The error minimization can be written as

$$\mathrm{x}^* \quad = \quad \underset{\mathrm{x}}{\operatorname{argmin}} \sum_{ij} \mathbf{e}_{ij}(\mathrm{x})^T \Omega_{ij} \mathbf{e}_{ij}(\mathrm{x}), \tag{2}$$

where Ω_{ij} is the information matrix associated to a constraint and x^* is the optimal configuration of nodes with minimum sum of error induced by the edges. The solution to Eq. 2 is based on successive linearization of the original non-linear cost typically around the current estimate. The non-linear error function is linearized using Taylor series expansion around an initial guess $\breve{\mathrm{x}}$:

$$\mathbf{e}_{ij}(\breve{\mathrm{x}} + \Delta\mathrm{x}) \quad \simeq \quad \mathbf{e}_{ij}(\breve{\mathrm{x}}) + J_{ij}\Delta\mathrm{x}, \tag{3}$$

with the Jacobian

$$J_{ij} \quad = \quad \left.\frac{\partial \mathbf{e}_{ij}(\breve{\mathrm{x}} + \Delta\mathrm{x})}{\partial \Delta\mathrm{x}}\right|_{\Delta\mathrm{x}=0}. \tag{4}$$

This leads to a quadratic form. In the end, the Jacobians from all constraints can be stacked into a matrix and the minimum of the quadratic form can be found through direct methods such as Gauss-Newton, by solving the linear system

$$H\Delta\mathrm{x}^* \quad = \quad -b, \tag{5}$$

where $H = \sum_{ij} J_{ij}^T \Omega_{ij} J_{ij}$ and $b = \sum_{ij} J_{ij}^T \Omega_{ij} e_{ij}$ are the elements of the quadratic form that results form the linearized error terms and $\Delta\mathrm{x}^*$ is the increment to the nodes in the graph configuration that minimizes the error in the current iteration at the current linearization point:

$$\mathrm{x}^* = \breve{\mathrm{x}} + \Delta\mathrm{x}^*. \tag{6}$$

As the measurements z and the poses x do not form an Euclidean space, it is advantageous to not subtract the expected and obtained measurement in Eq. 1 but to perform the operations in a non-Euclidean manifold space [27], [28]. This is especially helpful for constraints involving orientations.

In their seminal paper [1], Lu and Milios compute Eq. 5 by inverting the quadratic matrix H. This is reported to be the most computationally expensive operation as it scales cubic with the number of nodes. A lot of graph-based SLAM research focuses on efficiently solving Eq. 5 using domain

Fig. 1. Observing towers called Bibly towers built for triangulating other towers. Figure courtesy of [45].

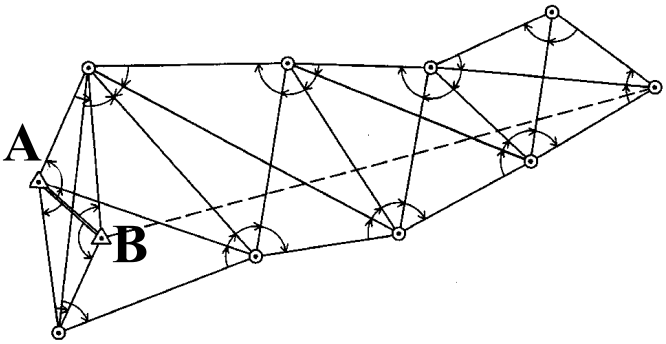


Fig. 2. A simple triangle net. The distance between towers A and B is physically measured. Angular constraints to all other towers are measured and the lengths of all other segments are computed with respect to the baseline AB with the measured angles. Figure courtesy of [46].

knowledge and sparse linear algebra methods. Gutmann and Konolige propose a method, which could incrementally build a map and required the expensive matrix inversions only after certain pre-determined steps [29]. Konolige provided a computationally efficient algorithm with a complexity of $O(n)$ for single loop closures and $O(n \log n)$ for multiple looped maps [30]. He identified the sparse connectivity of the information matrix resulting from SLAM graphs and used Conjugate Gradient preconditioned with an incomplete Cholesky factor to reduce computational requirements. Folkesson and Christensen formulate the least squares problem in Eq. 2 as an energy-minimization problem [2]. They also incorporate data association within the energy function, which implicitly performs a χ^2 test. They furthermore reduce the complexity and size of the graph by collapsing subgraphs into a single node, which they call star nodes.

Dellaert and Kaess explore the graph SLAM formulation as a factor graph using smoothing and mapping [31]. They call their method $\sqrt{\text{SAM}}$ as it uses matrix factorization methods such as QR, LU, and Cholesky decomposition to solve Eq. 5. Besides the offline approach, an efficient variant for incremental updates using Givens rotation is available [32].

Other authors minimize Eq. 2 using relaxation techniques, where parts of graphs are incrementally optimized [33], [34], [35]. Olson et al. solve the least squares problem using stochastic gradient descent [36]. In stochastic methods, the error induced by each constraint is reduced, one at a time, by moving the nodes accordingly. Grisetti et al. propose a reparametrization of the problem to improve convergence [37]. Stochastic methods are typically more robust to bad initial estimates and can be run incrementally by tuning the learning rate [38], [39]. Also hierarchical and submapping approaches have shown to efficiently compute solutions by decomposing the optimization problem at different levels of granularity [40], [41], [27], [16], [42], [43], [44].

IV. Geodetic Mapping

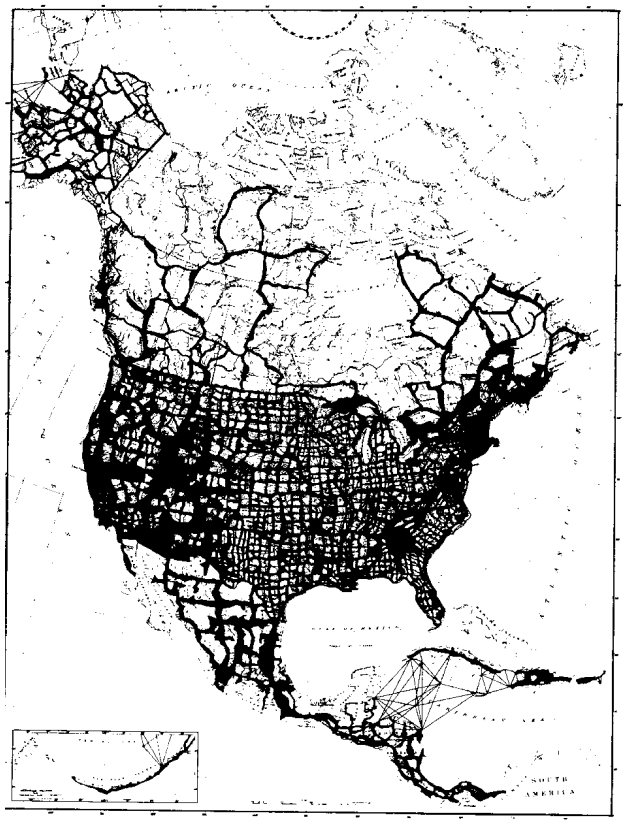
Geodetics, also known as geodesy, is the science that studies the earth at various global and local scales. It measures large scale changes in the earth's crustal movements, tidal waves, magnetic and gravitational fields, etc. The aspects of geodesy, which we are most interesting in the context of SLAM, are related to geodetic mapping. In this section, we first describe how the massive geodetic surveys, which typically result in *triangle nets*, are created. We then explain various geodetic back-ends and compare them to current graph-based SLAM methods for obtaining a minimal error configuration.

The basic principle behind geodetic surveying is triangulation. Various observation towers, called Bibly towers, typically 20-30 meters in height, are built in line-of-sight with neighboring towers [46]. An example of a Bibly tower with surveyors on top is shown in Figure 1. Geodetic surveyors built large interconnected mesh of observation towers by triangulating measurements between them. The resulting mesh of towers and constraints is commonly called a triangle net. A simple example of a triangle net is shown in Figure 2. Each line segment in Figure 2 is a constraint between two observation towers. Some of these constraints are directly measured while others are computed using trigonometrical relationships.

The method of obtaining constraints between the observing towers has evolved over time. Initially, constraints were distance only or angle only measurements. The distance measurements were obtained using tapes and wires of Invar, which has a very low coefficient of thermal expansion. Later, more sophisticated instruments, such as parallax range-finders, stadimeter, tellurometer, and electronic distance measuring instruments were used. Angular measurements were obtained using theodolites. Measuring distance using tapes and wires is more cumbersome compared to computing angles with theodolites. Hence, only a few distance measurements called baselines are computed and the other distance measurements are deduced based on angular measurements and trigonometrical relationships. In Figure 2, the towers A and B are the baseline and only angular measurements to other towers are measured. The distances between all other towers are deduced using the measured baseline AB, angles, and trigonometrical relations.

Moreover, measurement constraints used in geodetic surveys can be differentiated as absolute and relative measurements. Absolute measurements involve directly measuring the posi-

(a) The network of constraints used for the European Datum of 1987. Figure courtesy of [47].



(b) The network of constraints existing in 1981 for mapping the North American continent. Figure courtesy of [5].

Fig. 3. Triangulation networks spanning across Europe (left) and North America (right).

tion of a tower on the surface of the earth. These include measuring latitudes by observing stars at precise times and then comparing them to the existing catalogs. Stars were also used as fixed landmarks for bearing only measurements from different base towers at precisely the same time. Later, more sophisticated techniques such as very long baseline interferometry (VLBI) and GPS measurements, which lead to an improved measurement accuracy, were introduced. Angular measurements obtained with theodolites for large scale geodetic mapping were abandoned after the introduction of VLBI and the triangulation techniques have been replaced by trilateration around 1960. In the US, High Accuracy Reference Network (HARN) was built using VLBI and GPS only [48] by 1990. The Continuously Operational Reference Station (CORS) introduced in 1995 uses only GPS measurements and is regularly updated. National Spatial Reference System of 2007 known as NAD 83(NSRS2007) is the latest readjustment, containing solely of static GPS measurements [49].

Examples of large scale triangle nets are shown in Figure 3. Figure 3(a) displays the geodetic triangle net used for mapping Central Europe in 1987 whereas Figure 3(b) shows the triangulation network in existence for mapping North-America in 1983 (NAD 83). The thick lines in NAD 83 are long sections of triangulation nets comprising of thousands of Bibly towers. Connections between multiple sections are small triangle networks and are called junctions.

The geodetic mapping community typically chose the earth's center of gravity as the sensor origin. This eased the process of acquiring measurements as it is a quick and easy to standardize calibration technique. The primary form of triangulation was done using theodolites, all of which contain a level, which allows aligning the instruments with respect to the earth's center of gravity (E_{cg}). All other instruments use some form of a plumb-line as a reference, which aligned them with E_{cg}. In fact, choosing E_{cg} is also preferred when using satellites for acquiring measurements as they orbit the earth's E_{cg}. The exact shape of the earth is not a sphere but a geoid, which can be approximated as an ellipsoid. However, the center of this approximate ellipsoid does not coincide with E_{cg}. This is because earth's mass is not uniformly distributed. Hence, although choosing the sensor origin as E_{cg} is practically a good choice, it is mathematically inconvenient. It requires additional mathematical computations for mapping measurements to the center of the ellipsoid. This problem is commonly called computing the "deflection of the vertical".

The geodetic mapping problem can be broken down into two major sub problems, the first being the "adjustment of the vertical", the second being "adjustment of the net". The problem of "adjustment of the net" is finding the least squares system of the planar triangulation net whereas "adjustment of the vertical" involves finding parameters to wrap this mesh network on a geoid representing earth [50].

V. Adjustment of the Net – Geodetic Back-Ends

The problem of adjustment of the net is similar to the graph-based SLAM formulation. In the SLAM notation, the geodetic mesh network consists of various observation towers constrained by non-linear measurements. These physical towers are similar to the unknown state vector $\mathrm{x} = (\mathrm{x}_1, \ldots, \mathrm{x}_n)^T$ in the SLAM problem. In SLAM, 2D and 3D robot headings (roll, pitch and yaw) are estimated in addition to the the Cartesian coordinates (x, y, z). For geodesy, the surveyors were mainly interested in Cartesian coordinates of the towers on the surface of the earth. Angular components were used until the 1960ies in geodesy before GPS and VLBI systems were available.

The task of the back-end optimizer in geodesy is to find the best possible configuration of the towers on the surface of the earth to minimize the sum of errors from all constraints. All non-linear constraints can be linearized using Taylor series expansion and stacked into a matrix [8]. The least squares solution can be computed directly by solving the least square system in Eq. 5. Both, SLAM and geodetic problems inherently contain non-linear equations constraining a set of positional nodes. The process of linearizing the constraints and solving the linearized system is repeated to improve the solution.

The biggest challenge faced by the geodetic community over centuries was limited computing power. Even with the most efficient storage and sparse matrix techniques, there was no computer available which could solve the full system of equations as a single problem. For example, the NAD 83 mesh network, illustrated in Figure 3(b), requires solving 900,000 unknowns with 1.8 million equations [51]. A dense matrix of size 900,000 times 900,000 would require more than 3000 GB just to store it. Even when using sparse matrix data-structures such as compressed sparse rows or columns, NAD 83 would still require roughly 30 GB of memory to store the matrix of normal equations (considering that only 0.5% of the matrix elements are non-zeros). In the next section, we explain some of the geodetic back-end optimizers used for mapping large continental networks such as that of North America and Europe.

A. Helmert Blocking

When geodetics started to develop the large scale triangulations in the mid-1800s, the only way of solving large geodetic networks was by dividing the problem so that multiple people could work on solving each subproblem. Helmert proposed the first method for solving the large optimization problem arising from the triangulation network in parallel. His strategy, which was proposed in 1880, is possibly the oldest technique to partition the set of equations into smaller problems [6], [52]. Although graph partitioning and submapping has been frequently used in robotics, to the best of our knowledge, Helmert' method has not been referenced in the robot mapping community—only Triggs et al. [53] mention it as an optimization procedure in their bundle adjustment survey.

Helmert observed that by partitioning the triangle net in a particular way, one can solve the overall system of equations

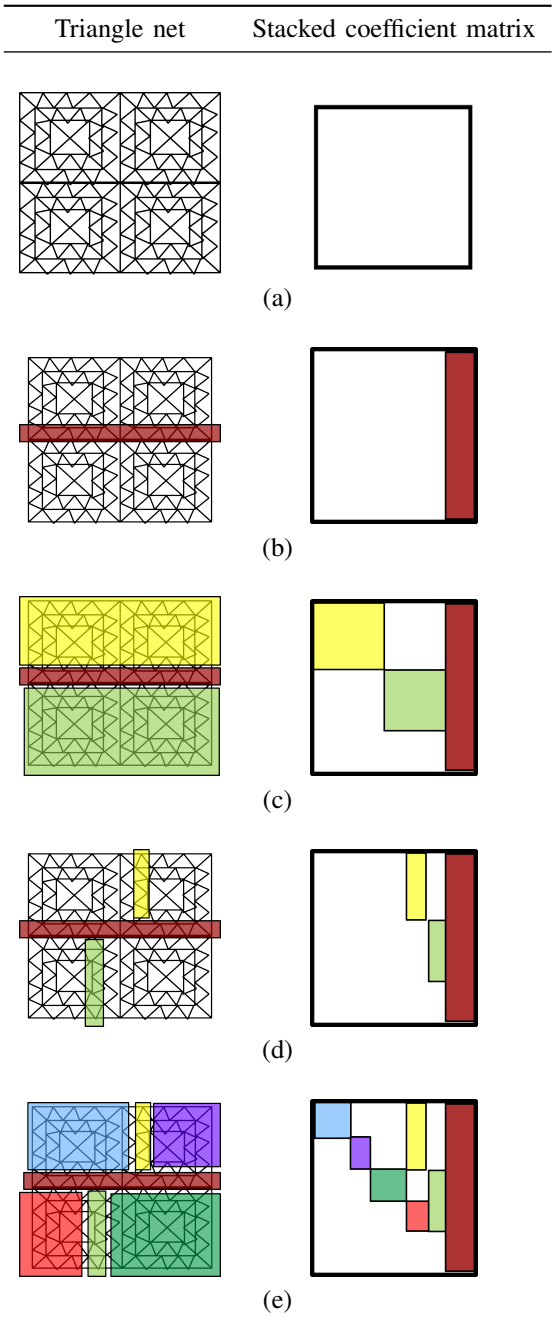


Fig. 4. Helmert blocking in action. The left column shows a toy example of triangle net. The right column shows the corresponding stacked coefficient matrix for each net.

in parallel. He outlined that the whole triangle net can be broken into multiple smaller subnets. All nodes that have constraints only within the same subnet are called inner nodes and can be eliminated. All separator-nodes, i.e., those that connect multiple subnets, are then optimized. The previously-eliminated inner nodes can be computed independently given the values of the separator nodes. Most importantly, the so-formed subnets can be solved in parallel. Helmert's blocking method is outlined in Alg. 1 and is explained more precisely as a mathematical algorithm by Wolf in [54].

Consider a simple triangle net shown in Figure 4. In

Algorithm 1 Helmert Blocking

1: Given a partitioning scheme, establish the normal equations for each partial subnet separately
2: Eliminate the unknowns for all nodes which do not have constraints with neighboring partial subnets
3: Establish the main system after eliminating all the inner nodes containing only intra-subnet measurements
4: Solve the main system of equations containing only separator nodes
5: Solve for inner nodes given the value of separator nodes

Figure 4(a), each line segment is a constraint and the end of segments represent a physical observation tower. Helmert observed that if he divides the triangle net into two halves, for example as illustrated in Figure 4(b), the top half of the towers will be independent of the bottom half given the values of the separators as shown in Figure 4(c). Such a system can be solved using reduced normal equations [54], [5].

Let us represent the whole system of equations from the triangle net in Figure 4(a), as:

$$Ax = b. \tag{7}$$

This equation can be subdivided into 3 parts in the following manner:

$$\begin{bmatrix} A_s & A_1 & A_2 \end{bmatrix} \begin{bmatrix} x_s \\ x_1 \\ x_2 \end{bmatrix} = b. \tag{8}$$

Here, A_s and x_s represent the coefficients and unknowns respectively, arising from the central separator. A_1 and A_2 are coefficients of the top and bottom subnets. The coefficient matrix $\begin{bmatrix} A_s & A_1 & A_2 \end{bmatrix}$ in Eq. 8 is shown on the right-hand side of Figure 4(c). The corresponding system of normal equations is:

$$\begin{bmatrix} N_s & N_1 & N_2 \\ N_1^T & N_{11} & 0 \\ N_2^T & 0 & N_{22} \end{bmatrix} \begin{bmatrix} x_s \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_s \\ b_1 \\ b_2 \end{bmatrix}. \tag{9}$$

The towers in x_1 do not share any constraints with towers in x_2. Both, x_1 and x_2 share constraints with x_s but not with each other. The key element in Eq. 9 is the block structure of N_{11} and N_{22}. The system of equation in Eq. 9 can be reduced such that:

$$\bar{N}_s x_s = \bar{b}_s, \tag{10}$$

where $\bar{N}_s$ is computed as:

$$\begin{aligned} \bar{N}_s &= N_s - \begin{bmatrix} N_1 & N_2 \end{bmatrix} \begin{bmatrix} N_{11}^{-1} & 0 \\ 0 & N_{22}^{-1} \end{bmatrix} \begin{bmatrix} N_1^T \\ N_2^T \end{bmatrix} \\ &= N_s - \sum_{i=1,2} N_i N_{ii}^{-1} N_i^T, \end{aligned} \tag{11}$$

and $\bar{b}_s$ is computed as:

$$\bar{b}_s = b_s - \sum_{i=1,2} N_i N_{ii}^{-1} b_i^T. \tag{12}$$

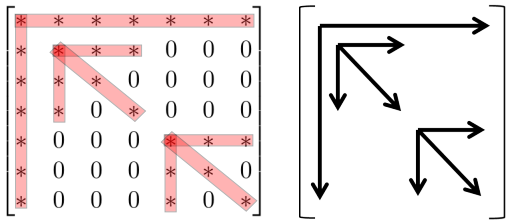


Fig. 5. The structure of the coefficient matrix arising from two levels of Helmert blocks. The arrows show non-zero cells.

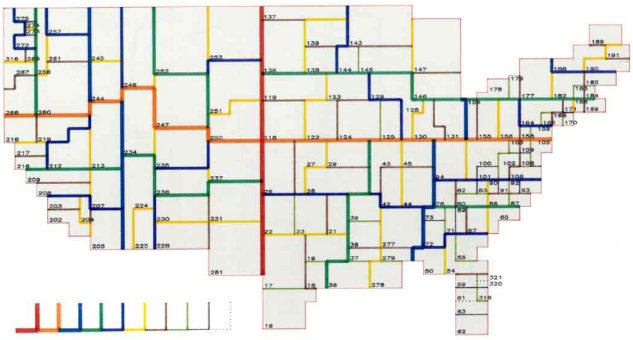


Fig. 6. Helmert blocking applied to North America with ten levels. The legend in the bottom left corner shows progressive levels of partitions. The first partition, cuts the continent into east-west blocks, whereas the second cut partitions each half into north-south blocks. Hence, each geographical block is partitioned into four regions and this method is recursively performed on each sub-block. Figure courtesy of [5].

Here, $\bar{N}_s$ is called the reduced normal equations. Once N_s has been solved, x_1 and x_2 can be computed by solving:

$$N_{11} x_1 = b_1 - N_s^T x_s \tag{13}$$

$$N_{22} x_2 = b_2 - N_s^T x_s. \tag{14}$$

Moreover, Wolf states that matrices should never be inverted for computing Eq. 10, 13, and 14 [54]. Instead, Cholesky decomposition or Doolittle-based LU decomposition should be employed. The steps outlined for computing the reduced normal forms in Eq. 11 and 12 represent the Schur complement. The inverse computation in this step is trivial if the subnets result in block diagonal matrices. If not, both Wolf and Schwarz mention that each of the subnets can themselves be sparse and can be further sub divided as illustrated in Figure 4(d) [54], [5]. Part IV of [49] also contains a detailed methodology for applying Helmert Blocking for optimizing geodetic networks used for National Spatial Reference System of 2007.

The structure of the normal matrix represented in Figure 4(d) looks like the matrix represented in Figure 5. In the example explained in Figure 5, we have two levels of Helmert blocks. The number of levels can be arbitrary and each subnet can have a different number of recursive sub-blocks. Elimination takes place in a bottom-up manner. Then, the solution at the highest level is used to recursively propagate back to smaller subnets. Helmert also outlines the strategy to select separators at each level. This is discussed later in Section V-G.

The Helmert blocking method was used for creating the North American Datum of 1983 (NAD 83). Ten levels of Helmert blocks were created with a total of 161 first level blocks. This is illustrated in Figure 6. The colored lines show the partitions of the graph at different levels. On average each of these 161 subnets contained 1,000 nodes, but roughly 80% of these nodes contained only internal measurements and can be eliminated [5]. Three non-linear iterations of the whole system were performed, which took roughly four months each [5].

The Helmert blocking method is inherently multi-core and parallel but it still requires solving a large number of equations when solving for the reduced normal equation for a large separator set. The complexity of the exact method can be reduced if networks are built in sections as shown in Figure 7(b). Whenever the triangular network was not built using long sections, the net was approximated using polygon filling methods, which are explained below.

B. Polygon Filling

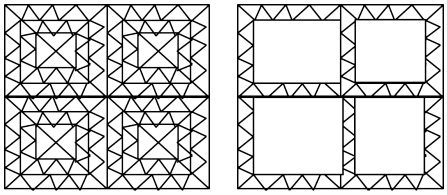
Polygon filling methods are used to convert a dense net into sections. An example for a dense network is shown in Figure 7(a) and the corresponding sparse mesh of sections in Figure 7(b). Inner regions in Figure 7(b) are completely removed from the initial optimization. Only the sections as shown in Figure 7(b) are optimized. In a subsequent optimization, the structure of the grids is fixed and the interior dense nets are optimized. This is an approximation as the optimization of the sections does not contain any measurements from the interior nets. Any error in the sections is distributed over the interior, shaded subnets [55].

Helmert's method can also be exploited in meshes comprising of sections to reduce the number of separator nodes. The colored small squares in Figure 7(c) are the separators after applying Helmert's method on the section and their size is much smaller compared to those depicted in Figure 4. The new separators are formed at the intersections of the sections rather than the long partitions in the original method. This difference can be observed by comparing Figure 7(c) and Figure 4(e).

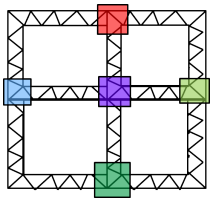
The polygon filling method is also the preferred technique for incremental optimization. As triangulation networks evolve over time, large sections are built first and are subsequently triangulated densely—if and when required. This means that the outer sections are fixed and are never updated with measurements from inner nodes. Thus, any error in the initial optimization of the outer section has to be distributed over the inner regions. Even when adding new sections from new surveys covering unmapped area, the initially obtained sections are often kept fixed. Although this procedure is an approximation, it allows for increasing the size of a triangulation net without having to start the optimization from scratch. Thus, it is an efficient incremental map building method.

C. The Bowie Method

The use of networks of sections and the polygon filling method reduces the size of the optimization problem significantly but it is still computationally demanding considering the

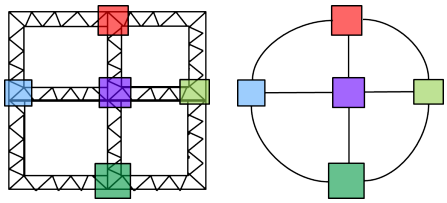


(a) Original triangle net. (b) Polygon filled net.



(c) Helmert's method applied to polygon filled net.

Fig. 7. The original triangle net (a) and the corresponding net after applying the polygon filling method (b). This leads to a smaller and better tractable problem at the expense of reduced accuracy. Once the sparse solution is computed, the interior stations can be computed by fixing the sections. Also here, Helmert blocking can be applied leading to the separator nodes illustrated by the colored squares (c).



(a) (b)

Fig. 8. Bowie's approximate triangle net. Size of separators are much smaller once the polygon filling method is used. The separators are shown in colored squares (left). The Bowie method approximates this problem by abstracting the sections into a single constraint and all separator nodes as a single node (right).

resources of the early 1900s. Bowie approximated the above methods further to create the North American Datum in 1927. Bowie's main insight was that he can approximate the net comprising of sections by collapsing intersections into a single node and the sections into a single virtual constraint. This is illustrated in Figure 8. This much smaller least squares system is solved to recover the positions of the intersections, which can then be used to compute the sections independently.

The core steps of the Bowie method consist of separating the sets of unknowns into segments and junctions. The junctions act as a small set of separator nodes. The junction nodes are shown as colored squares in Figure 8. These are not a single node but a small subnet of towers, which separates the sections. A generic junction is shown in Figure 9. All nodes in one junction are optimized together using least squares adjustment but ignore any inter-junction measurements. After this optimization, the structure of the junction does not change, i.e., each node in a junction is fixed with respect to other nodes in that particular junction.

As a next step, new latitudinal and longitudinal constraints are created between junctions. This is done by approximating each section with a *single* constraint. Each of such single constraints is a two-dimensional longitude and latitude constraint.

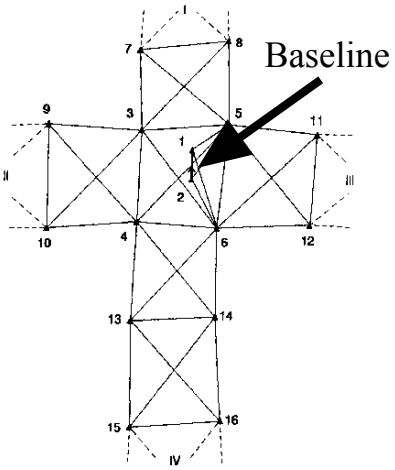


Fig. 9. Example of a typical junction connecting segments in the four directions. The baseline and azimuth of stations 1 and 2 were measured directly. All other measurements were relative to other stations. Figure courtesy of [5].

As a result, each junction turns into a single node and each section into a single constraint. This leads to a much smaller but approximate problem, which is optimized using the full least-squares approach.

The above described steps of the Bowie method are summarized in Alg. 2, see also Adams [56]. The full least squares problem is not solved by matrix inversion but by a variant of Gaussian elimination called Doolittle decomposition (see Wolf [57]). The Doolittle algorithm is a LU matrix decomposition method, which decomposes a matrix column-wise into a lower triangular and upper triangular matrix. The solution can be computed using forward and backward substitution steps as with other matrix decomposition methods as well.

In essence, the Bowie method generates new, virtual constraints from sections. He introduces a weight for each virtual measurement, which are chosen as the ratio of the length of the section with respect to the sum of all the section lengths. Hence, these weights are proportional to the length of the sections so that the larger proportion of the error is distributed over long sections compared to shorter ones.

Another computational trick to lower the efforts is to use diagonal covariances for the two-dimensional, virtual, latitude/longitude constraints. This enables to separate the system of equations for longitude and latitude. This yields two least squares problems with half of the original size.

The partitioning of the triangular net into junctions and sections is done manually. Each junction has to contain at least one measured baseline and one measured azimuth direction. This is sometimes referred to as astronomical stations. Occasionally, the size of junctions were enlarged to include an azimuth measurement because azimuth measurement have not been taken at all towers. Figure 10 illustrates the original triangle net and Bowie's approximated net into segments and junctions used for NAD 27. In Figure 10, the small circles represent junctions and all lines connecting junctions are sections of triangle nets. These sections and junctions individually represent a subset of the constraints connecting stations.

In sum, the Bowie method is an approximation of Helmert Blocking. The approximation uses single level subnets and an approximate optimization of the junction nodes. The optimization of the highest level in Helmert Blocking consists of junction nodes and is computed via reduced normal equations. In contrast to that, the Bowie method uses virtual constraints at the highest level. The main difference is that the system of equations created by the virtual constraints is much smaller and sparser and hence easier to optimize than the full set of reduced normals as in the exact Helmert blocking method.

To the best of our knowledge, the Bowie method is the first implementation of a large scale approximate least squares system. It was effectively used in creating the NAD 27 as the triangulation nets were build in sections forming large loops. The Bowie method exploits this structure and also allows incremental optimization. New loops in the triangulation nets were not optimized as a whole, instead, they were integrated into the existing system by keeping the previous positions fixed. This led to inaccuracies but was better tractable than optimizing the system as a whole.

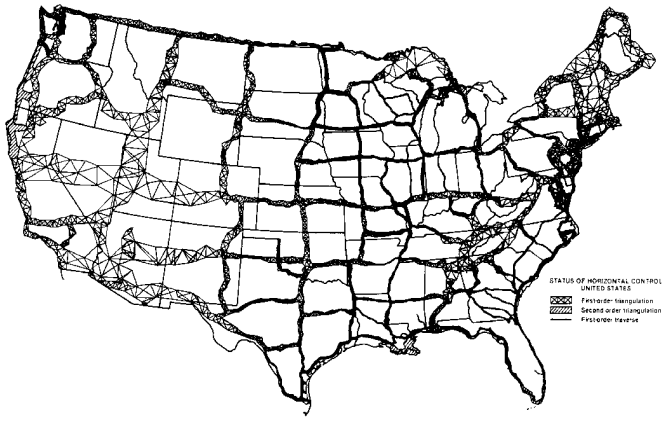
Algorithm 2 Bowie method

1: Separate triangle net into junctions and segments.
2: Optimize each junction separately.
3: Create new virtual equations between junctions treated as a single node
4: Solve the abstract system of equations comprising of each junction as a single node and each section as a single constraint
5: Update the resulting positions of stations in the segments using the new junction values

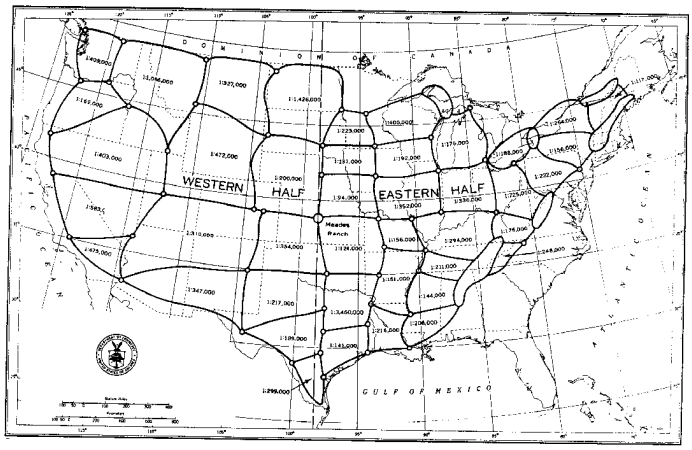
D. Modified Bowie Method for Central European Triangulation

The approximation introduced by the Bowie method yielded suboptimal results for the European Datum of 1950 [55]. For example, the virtual latitudinal and longitudinal constraints are artificially generated and not directly measured constraints but this fact is not fully considered in the optimization. Furthermore, cross correlations between latitudes and longitudes are ignored by using diagonal covariances and the junctions are fixed as a whole. Another issue in the Bowie method results from the assumption that the size of sections are much longer than junctions, which was the case for NAD 27, but the triangulation nets in the European Datum of 1950 (ED 50) are more dense. This results in the amplification of the errors introduced by the approximations [55].

The European geodetics thus proposed two modifications to the Bowie method to cope with the above mentioned problems for optimizing the ED 50 [55]. The first one addresses the virtual measurements. Line 3 of Alg. 2 sets up as many equations as sections. Instead, one virtual equation for every loop was created, enforcing a zero error around the loop.



(a) The triangle net used for creating NAD 27.



(b) The Bowie method as applied for solving NAD 27.

Fig. 10. Bowie method as used for NAD 27 triangle net (top). The figure below illustrates Bowie's approximation. Each small circle represent a junction and the junctions are connected by sections. Given the values of the junctions, the segments are independent of the rest. The western half contains 26 junctions and 42 sections creating 16 loops. The eastern half contains 29 junctions and 55 sections forming 26 loops. All measurements were computed with respect to Meades Ranch located in Kansas shown by the big circle in the center. The numbers inside the regions depict the error in a loop after the optimization is completed. Figure courtesy of [5].

The second modification addresses the way the linear system is solved efficiently. Instead of using the Doolittle method, they used the Boltz method, as explained below, which allowed for computing the matrix inversion in one shot.

E. Boltz Method

The Boltz method is an alternative to the Gaussian elimination with LU decomposition for solving large set of equations in *one shot* [58], [59]. An explanation of the Boltz method was provided by Wolf, which basically says that Boltz was tabulating matrices and their corresponding inverted solution [57]. Boltz was able to simply *look up* the solutions for subproblems from a table instead of recomputing the inverse.

The central question here is how to setup the linear system so that the matrix of the normal equations, which needs to be inverted, reappears during the calculations. In general, for this method to work infinitely many matrices would need to be tabulated. To keep the number of tabulated matrices tractable, Boltz proposed to divide the equations into two groups:

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \end{bmatrix} = \begin{bmatrix} \mathrm{v} \\ \mathrm{w} \end{bmatrix}. \tag{15}$$

Here, A, B and C are coefficients of the normal matrix and x and y are unknowns. Boltz proposed to cache A^{-1} and solve the reduced system of equation via

$$\left[C - \left[B^T A^{-1} B\right]\right] \mathrm{y} = [\mathrm{w} - B^T A^{-1} \mathrm{v}]. \tag{16}$$

The key idea is to separate angle-only measurements from all others. Let x be the unknowns that arise from angle-only measurements, while y arise from all other measurements (see also [55]). The constraints in x are simple given the triangular structure of the net: the sum of the angles in a triangle equals $180°$, the sum of angles around a point must add up to $360°$. Hence, the coefficients of the matrix A in Eq. 15 and Eq. 16 can be written so that it contains only 1's, 0's and -1's. This, in combination with domain knowledge about the structure of the triangular nets, allows for efficiently caching A^{-1}. Boltz method is designed for triangle nets in which the structure repeats itself. This is often the case in geodetic mapping.

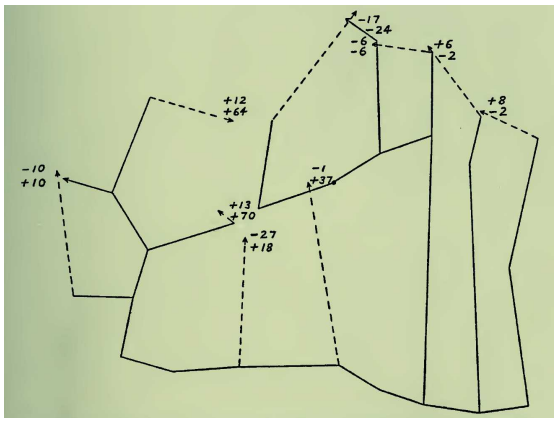
The look up was done by humans but we could not find details about the procedure for physically storing and retrieving the inverted matrices. One might even spot similarities between Boltz' proposal of caching matrix inversions and techniques like lifted probabilistic inference [60]. Caching can result in a substantial performance gain if majority of operations are done manually as was the case for the ED 50. Boltz' method was successfully used for eliminating up to 80 by 80-sized matrices when creating ED 50 [55].

Note that reduction in Eq. 16 is the Schur compliment but the motivation behind using the Boltz method is the fact that A^{-1} is tabulated and looking up the inverse is substantially faster than computing it (by hand). The geodetic researchers prior to 1950 mention that the Boltz method allowed solving least squares without biasing the solution towards any particular unknown. They mention that the Boltz method treated all unknown equally unlike the Gaussian elimination which caused the last variable to contain larger errors[1].

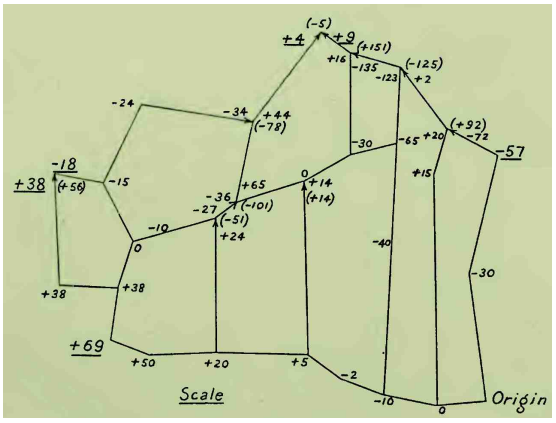
F. The Indian Method of 1938

Bomford proposed a different approach than the above mentioned methods for solving the triangulation net for the Indian subcontinent in 1938 [62]. He mentions that the Bowie method was infeasible for optimizing the network of towers in India, as the number of astronomical stations was too small and each junction in the Bowie method requires to have an astronomical station with independent latitude and longitude observations.

[1]From [61]: "A method devised by Boltz for solving the normal equations occurring in the adjustment of a triangulation network; it allows a large set of equations to be solved in one straight operation by the method of least squares without biasing the solution towards any particular unknown. Initially, the normal equations were solved using the Gaussian method of successive elimination. This method, however, causes the last determined value to contain larger errors than the first determined value. Boltz's method treats all unknowns equally and is particularly suitable for solving large systems of equations."

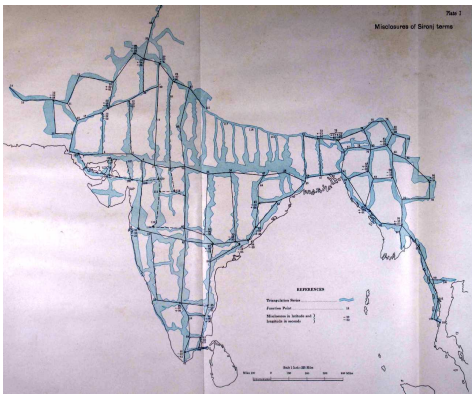


(a) The mesh net with circuit errors on a zoomed section.

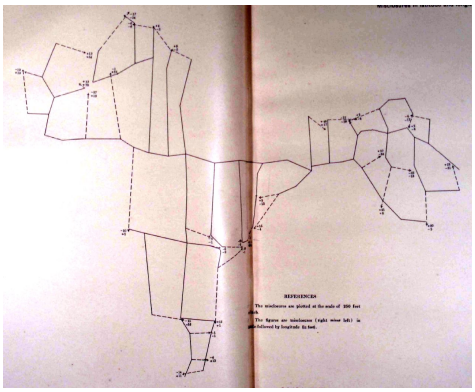


(b) The resulting optimized net with the correction for each junction.

Fig. 12. A zoomed in view on the North-East section of the Indian subcontinent. It shows the initial and final configuration of the triangle net used in 1939 for surveying India. The solid lines represent the initial spanning tree. The dotted lines are sections which induce circuit points and thus a residual error. Each circuit point contains the latitudinal residual error (top) and longitudinal residual error (bottom). Figure courtesy of [63].



(a) The Indian triangulation net on the map of the Indian subcontinent. The blue region shows the area covered by sections of nets and the solid line is the represented single section.



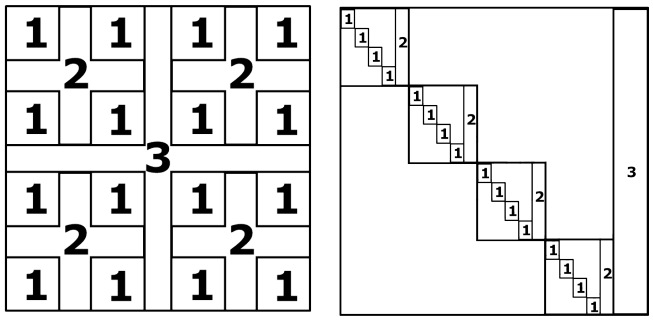
(b) The triangulation net with circuit errors at intersections.

Fig. 11. The Indian triangulation net (top) and the section network with errors in circuit points (bottom). The dotted lines are sections, which complete loops. The point of intersection between a solid and a dotted line is a circuit point, which has a latitude and longitudinal error induced by the dotted line. Figures courtesy of [62].

The starting point of the Indian method is similar to the Bowie method. Junctions and sections are created from the triangle nets. These junctions are points of intersection of sections and do not require any astronomical constraints. This is illustrated in Figure 11(a). A *spanning tree* consisting of sections and junctions is chosen from the mesh network of the triangle nets. The initial spanning tree does not have errors as there are no loops and it is depicted by the solid lines in Figure 11(b). All other sections that are not part of the spanning tree will introduce a residual error (see dotted lines). Bomford refers to the points where multiple sections meet and have a residual error as *circuits*. These circuit points have a residual in latitude and longitude. A zoomed-in portion of the North-east part of the map is shown in Figure 12. Fig.12(a) illustrates the errors in latitude and longitude at each circuit point. Bomford proposed that the error in circuit points could be solved by distributing it around the loop. He had no automated system to distribute errors around the loop. He manually performed trial and error methods to reduce the total error induced in the circuit nodes. For example, he states that he first reduced the longitudinal error in the North West regions, shown in Figure 12(a), by adjusting longitudinal values of the lower section. He incorporates the "stiffness" of sections, which he approximated by the length and uncertainty of the sections [62]. This method does not appear to be as rigorous as the Bowie or Helmert methods but still resulted in accurate maps of the Indian subcontinent.

G. Variable Ordering

In Sec. V-A we outlined the Helmert Blocking method but did not mention how the partitioning was performed. Helmert himself provided simple instructions for creating subnets and for partitioning the blocks, which is critical for his method. He first instructed to pick a latitude such that it partitions all the towers roughly into halves. Next, a longitude is chosen for each upper and lower half to partition the Northern and Southern regions into Eastern and Western partitions. Each obtained geographical rectangular block is recursively



(a) A planar mesh partitioned according to nested dissection.

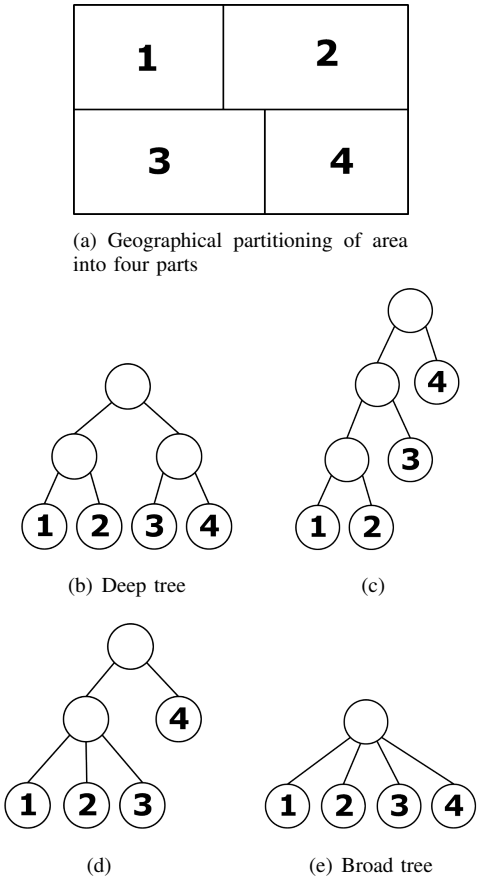
(b) The corresponding matrix picture for this net.

Fig. 13. Nested dissection and the corresponding matrix arrangement. The four higher block with nodes numbered 1 and 2 are independent given the separator 3. All sub-blocks numbered 1 are independent given the sub-block numbered 2.

partitioned into four further blocks. This strategy is illustrated in Figure 4(d) and Figure 4(e). The proposed method is simple but effective, since the triangulation network is built roughly as a planar graph and the density of the net was approximately similar across different locations. Helmert's approach to partition the triangle nets also shares similarities to the nested dissection variable reordering strategies [64] used to efficiently factorize sparse matrices. The use of variable reordering significantly improves the computation and memory requirements for matrix decomposition methods [65], [66].

The nested dissection variable reordering scheme was initially proposed for solving a system of equations for n×n grid problems arising from finite-element discretization of a mesh [64]. It partitions the graph with "+" shapes, as illustrated in Figure 13. Each resulting block is then recursively partitioned with a "+" shape. The number on each partition corresponds to the entry in the coefficient matrix. Helmert's proposal to divide a triangulation net recursively along latitudes and longitudes was used by Avila and Tomlin for solving large least squares using the ILLIAC IV parallel processor for optimizing geodetic networks [67]. Later, Golub and Plemmons used the Helmert blocking variable ordering strategy for solving large system of equations using orthogonal decomposition techniques such as QR decomposition for the system of equations arising from the geodetic network [68].

Given that both, Helmert's proposal and the nested dissection algorithm, are so similar, researchers performed a study to understand the best way to create the separators and to join the blocks given a four-way partitioning for NAD 83 [5]. Figure 14 shows four ways of joining a block partitioned into four sub-blocks. The partitioning can be done using either Helmert's strategy or nested dissection. The key design choice is whether to use a deep tree or a broad tree strategy as shown in Figure 15. The deep tree strategy creates smaller and denser final blocks while broad trees result in larger and sparser blocks. This can be seen from Figure 15 (left) and Figure 15 (right) for a simple example. The large sparse matrix in the broad tree strategy implies further variable reordering to minimize the matrix fill-in. In the deep tree strategy, parallelism can be exploited better, but it requires more matrix



(a) Geographical partitioning of area into four parts

(b) Deep tree

(c)

(d)

(e) Broad tree

Fig. 14. Possible different ways for joining four Helmert blocks according to [5]. The top row shows a rectangular area partitioned into four blocks, which can be joined in four ways.

allocations. In the simple example shown in Figure 15, seven matrices are allocated for the deep tree compared to five allocations for the broad tree. Hence, the deep tree strategy was preferred for NAD 83 to enable more parallelism and as the created, dense sub-blocks do not require further reordering.

H. Removing Outliers for Geodetic Mapping

The task of traditional geodetic mapping involves hundreds of surveyors working in parallel to obtain measurements. This also results in many faulty constraints [5]. The typical source of faulty constraints are human errors, errors in instruments, and sometimes errors when transferring entries from physical journals into the database management systems. For NAD 83, erroneous constraints are detected and removed through a block validation process. The block validation process validates constraints and towers in small geographical regions, one block at a time. The blocks are created by a geographical partitioning of stations as shown in Figure 16. These also represent the lowest level of Helmert blocks.

The block validation process consists of two iterating steps. First, by identifying under-constrained nodes and second, by checking for outliers given small blocks of nodes. If any node in a block is under-constrained, either additional constraints are added from neighboring blocks or the node is removed from the block. This avoids any singularities within the least

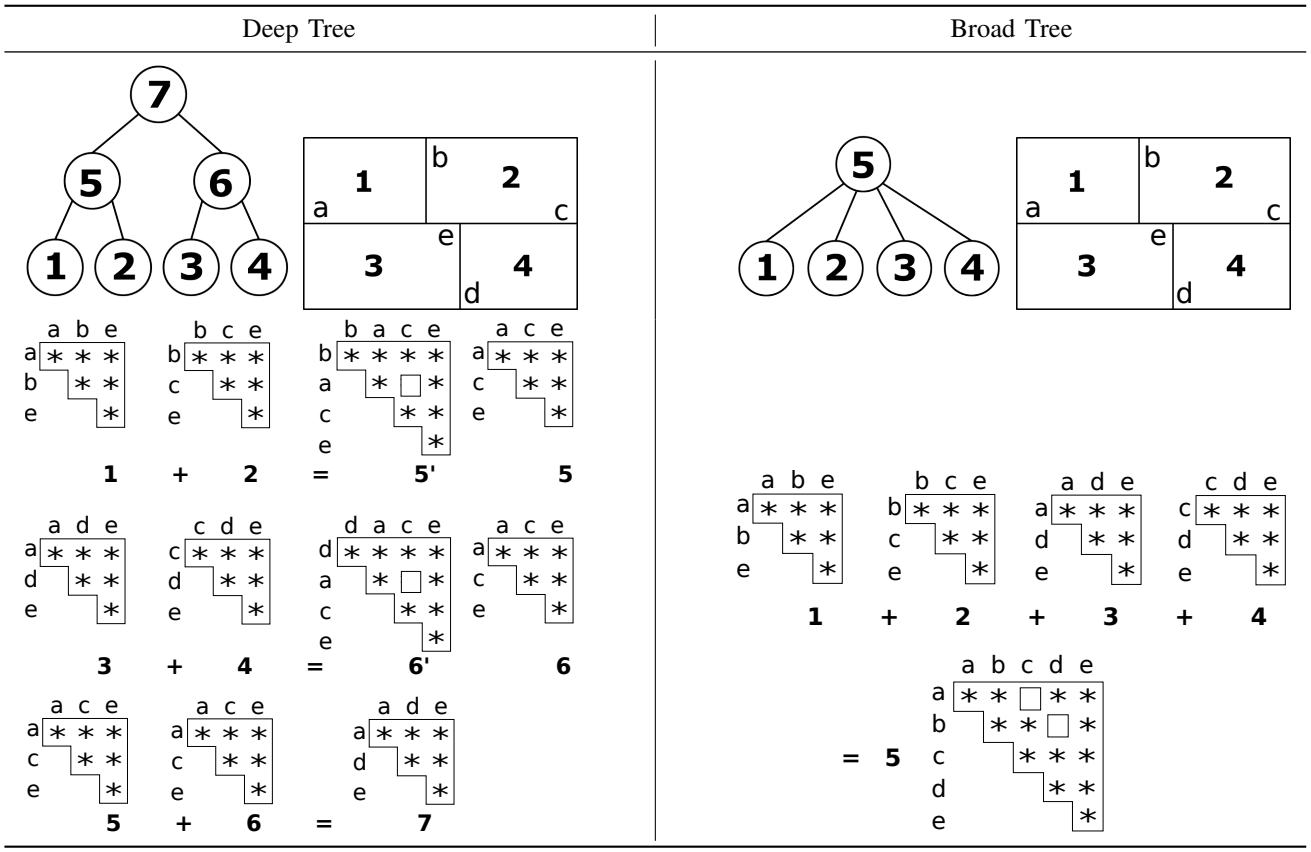


Fig. 15. Comparison between deep tree (left) and broad tree (right) ordering according to [5]. The area is partitioned into four blocks labeled 1, 2, 3 and 4. *a* to *e* represents separator variables. The variable set *e* are separators with constraints in all 4 blocks, *a* are variables with constraints only between blocks 1 and 3, *b* are variables constraining between blocks 1 and 2 only, etc. In this example, the deep tree approach would merge two subgraphs at a time, while the broad tree approach merges four subgraphs at a time. In both methods all low level subgraphs labeled 1 to 4 are processed in parallel. The deep tree method requires two more matrix allocation compared to the broad tree approach but the broad tree approach results in a larger matrix with zero blocks, which require additional variable re-ordering to reduce computation.

squares solution. After this issue is resolved, the block is optimized and all constraints having a weighted residual error of greater than three $(\chi^2 > 3)$ are evaluated. These are regarded as possible erroneous constraints. A constraint with a large residual error $(\chi^2 > 3)$ is deleted if there is another constraint between the same nodes having a similar sensor modality but a smaller residual error. A constraint with high residual error is also deleted if there are additional constraints between other nodes capable of constraining the nodes under consideration. If no additional constraints are found, the standard deviation of the possible outlier constrained is doubled until $\chi^2 < 3$. In other words, the assumed sensor noise for this constraint is increased, which is a similar principle behind m-estimators in robust statistics [69], [70].

The whole process of optimizing a sub-block and evaluating constraints with large error is repeated till the block is free of outliers. For NAD 83, a total of 843 blocks were evaluated using block validation methods, as illustrated in Figure 16. Each block consisted of 300 to 500 nodes and required roughly one person month to validate [5]. This process for outlier detection was also used as recently as 2007 for NAD83(NSRS2007), where seven trial solutions were carried out to account for outliers, singularities and handle weakly determined areas [49].

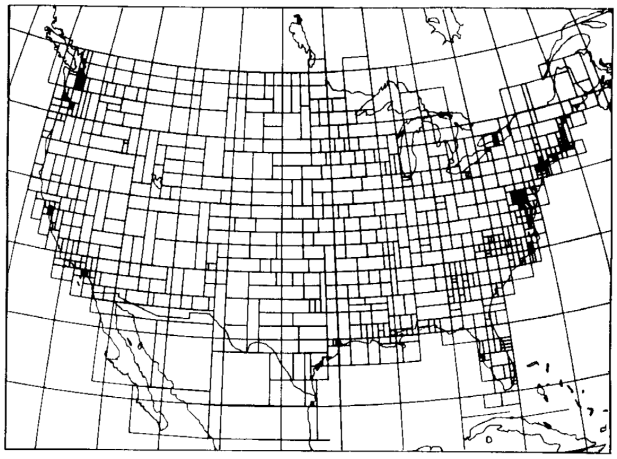


Fig. 16. Boundaries of blocks of data used in the block validation process. Figure courtesy of [5].

VI. Relation to SLAM back-ends

The use of sparse linear algebra and matrix decomposition methods have been introduced in robotics only recently [31], [21], [71], [22]. In contrast to that, the geodetic scholars were using LU, QR and Cholesky decomposition for a long time—Cholesky decomposition was actually developed for geodetic

mapping purposes. This section aims at highlighting some of the similarities between the methods developed by both communities.

A. Hierarchical and Sub-Map-Based SLAM Methods

Grisetti et al. [27] propose an efficient hierarchical multi-level method for optimizing pose-graphs. The higher the level in the hierarchy, the smaller the pose-graph. At each level, a subgraph from a lower level is represented by a single node on the higher level. The optimization is primarily carried out at the higher levels and only propagated down to the lower level if there is a significant update to the pose of a node. Each edge in the higher level is computed via a new deduced virtual measurement, which is created after a local optimization. The gain in speed by this method results from the fact that computationally expensive optimization is only performed when the pose of a node in the coarse representation gets affected more than a certain threshold (which was chosen as 5 cm or 0.05° by the authors). There exists also a extension to general SLAM graphs and bundle adjustment problems [16].

The virtual measurements created by Grisetti et al. are similar in idea to those in the Bowie method explained in Section V-C. The Bowie method creates a two-level hierarchy instead of the multiple levels as in [27]. Both methods use a single node from a dense sub-graph in the higher, coarser level and add a virtual deduced measurement between the smaller new problem instances. A difference is that given a geodetic network consisting of sections, each edge in the Bowie method represents a sub-graph whereas in the hierarchical approach of Grisetti et al., only nodes represent sub-graphs. Furthermore, Grisetti et al. compute the uncertainty of a virtual constraint explicitly.

Also Ni et al. propose back-ends, which divide the original problem into multiple smaller sub problems [42], [43]. These smaller problems are then optimized in parallel. The first approach, [42], partitions the problem into a single level of sub-maps, while [43] partitions each map recursively multiple times. The speed-up is mainly due to caching the linearization result of each subgraph. In both methods, all nodes in a subgraph are expressed with respect to a single base node. This allows for efficiently re-using the linearization of the constraints within each subgraph. This insight results in a reduction of the total computation time. Boundary nodes and constraints connecting multiple subgraphs need to be recomputed at each iteration, while results for nodes within a subgraph can be reused. The methods of Ni et al. would result in a batch solution if each non-linear constraint within a subgraph is re-linearized at every iteration, but they show experimentally that by not doing so, a high speed-up in optimization time is achieved at the expense of very small errors.

The sub-maps methods proposed by Ni et al. show many similarities to Helmert's approach of partitioning the triangle nets into subnets. The idea of anchoring sub-maps in [43] has a similar motivation as Bowie's idea for anchoring junctions and moving them as a whole by shifting the anchor nodes and not re-optimizing each subgraph. Krauthausen et at. prove that approximately planar SLAM graphs can be optimized in $O(n^{1.5})$ by using the nested dissection reordering [72]. The nested-dissection algorithm, which is at the heart of [43] and [72], is similar to Helmert's strategy of recursively partitioning a planar triangulation net [68]. Furthermore, the out-of-core parallel Cholesky decomposition-based non-linear solver for geodetic mapping proposed by Avila and Tomlin in 1979 [67] uses Helmert blocking and is thus strongly connected to [43]. It should, however, be noted that the geodetic community was generating the partitioning manually with domain knowledge and that the triangle-nets have a simpler and more planar structure than typical SLAM graphs.

B. Stochastic Methods

Olson et al. propose stochastic gradient descent for solving pose-graphs with a bad initialization [36]. They re-parametrize the constraints from a global pose to a relative pose representation. In the global pose, all positional nodes are represented with global coordinates in the world frame, whereas in the relative pose representation, each pose is represented with respect to its previous pose in the odometry chain. This allows the authors to distribute the errors of each constraint within the loop induced by it. Grisetti et al. [37], [73] improve this approach by using a tree parameterization based on a spanning tree instead of the odometry chain. The basic idea of both approaches is to stochastically select a constraint and move nodes based on the parameterization to reduce the error induced by the constraint. A learning rate controls the step size and is gradually decreased to prevent oscillations. Both approaches also assume spherical covariances and intelligent data-structures to efficiently distribute the errors. These methods are also capable of online and incremental approaches by increasing the learning rate for active areas [38], [74], [39].

A major intuition of using stochastic methods is equating error around a loop to zero. This is somewhat similar to the motivation for the modified Bowie method, where one virtual equation is set-up for each loop in the sparse mesh and the error around every loop is equated to zero. Examples of distributing errors around loops has also been explored in the "zero-sum property" of [15] and "trajectory bending" of [75].

The Indian method of 1938 [62] described in Section V-F has a similar motivation to the stochastic methods described above. In the Indian method, an initial spanning tree is manually chosen, which is a similar initialization strategy than the one of Grisetti et al. The Indian method as a whole, however, is rather informal and distributes the errors manually in a trial and error fashion. In contrast to that, Olson et al. and Grisetti et al. minimize the error using stochastic gradient descent and provide a more formal treatment of the approach.

C. Robust Methods for Outlier Rejection

For quite a while, SLAM back-ends suffered from data association failures that result in wrong constraints between nodes in the graph. Recently, a set of different approaches have been proposed that are robust even if a substantial number of constraints are outliers.

Latif et al. [13] propose RRR, which is a robust SLAM back-end, capable of rejecting false constraints. RRR first clusters mutually consistent and topological related constraints together. Each cluster is checked for intra-cluster consistency by comparing the residual of each constraint with a theoretical bound. Constraints that do not satisfy the bound are removed. This approach is similar to the block validation technique used for NAD 83. The individual blocks in block validation and the clusters in RRR conceptually represent similar sub-graphs. The blocks are geographical partitions but are actually a set of nodes and constraints similar to what clusters represent in RRR. Again, it should be noted that the triangular networks of the geodetic community have a simpler structure than SLAM graphs and thus the verification step is easier to conduct.

The dynamic covariance scaling (DCS) approach by Agarwal et al. [14] is a robust back-end that is able to reject outliers by scaling the covariance of the outlier constraint. DCS can be formulated as a generalization of switchable constraints [12], another state-of-the-art back-end for robust operation in the presence of outliers. In DCS, the covariance matrix of constraints with large residuals is scaled such that the error stays within a certain bound. This is related to the "doubling of standard deviation of each constraint" strategy used in [5]. It is, however, not clear if and how the scaling in NAD 83 was modified between different iterations.

D. Linear SLAM

A good initial guess is critical for iterative non-linear methods to converge to the correct solution. For providing a good initialization, Carlone et al. [15] propose a linear approximation for planar pose-graphs. Their approach yields an approximate solution to the non-linear problem without any initial guess and thus can be used as a high quality initial guess for state-of-the-art back-ends. Carlone et al. formulate the SLAM problem in terms of graph-embedding and suggest to partition the system of equations into:

$$A_2^T \rho = R(\theta)\Delta^l \tag{17}$$

$$A_1^T \theta = \delta. \tag{18}$$

Here, θ contains angular unknowns and ρ represents the positional unknowns. The terms A_1 and A_2 are their respective coefficient matrices, Δ^l and δ are the corresponding constraint residuals, and $R(\theta)$ is the stacked rotation matrix. Eq. 18 is solved first and the computed value of θ is used for solving Eq. 17. The authors provide both, a theoretical proof and real-world examples for the algorithm in [15].

Related to Linear SLAM, Boltz was reordering the equations in Eq. 15 and eliminating angular unknowns first. His motivation was to cache matrix inversions for a faster human look-up, but geodetics used it not only for gains in speed but also because it "did not bias the solution towards any particular unknown like Gaussian elimination" [61]. In our geodetic survey, we did not find any proof for this statement, but as both approaches share similarities, we suspect that this statement is related to the properties of Linear SLAM. The theoretical justification of [15] also holds in the case of Boltz' reordering of eliminating angular constraints first.

E. Dense Sub-Blocks

The manuscript [5] describing NAD 83 offers several interesting aspects ranging from database management to memory and cache-friendly algorithms. During the project, engineers used punch card machines—rather inconvenient and cumbersome tools for solving large matrix problems compared to modern computers. Physical file management system and efficient card storage are reminiscent of sophisticated techniques used in modern software. We have decided to not discuss all topics in detail here but in essence, the NAD 83 engineers were arranging data similar to the block matrix representations. The punch cards describing the data comprising station positions and constraints were also physically stored as low-level Helmert Blocks. All punch cards corresponding to a single block were stored together for faster retrieval.

Such dense sub-blocks are central elements in some of the fastest SLAM back-ends. Konolige et al. [10] describes a 2D SLAM implementation where the computation time is significantly reduced by storing each Jacobian as a 3×3 block matrix. Kümmerle et al. [71] generalize the block storage and indexing strategy for 6×6 and other types of feature nodes. Finally, most modern graph-based SLAM implementations use the Cholesky decomposition algorithm from the SuiteSparse library by Davis [76]. The fastest super-nodal Cholesky decomposition routine, CHOLMOD, also exploits dense sub-blocks [77].

F. Further Remarks

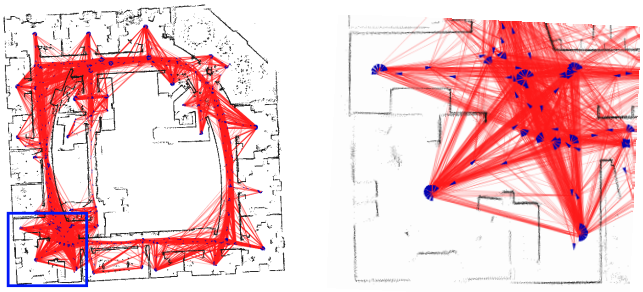
As an additional note, the Cholesky decomposition, which is commonly used in error minimization, was developed in the early 1900s by André-Louis Cholesky for geodesy and map building while he was in the French Geodetic section. The χ^2 distribution was also published by Helmert in [78]. This is further elaborated in [79] and details with respect to Pearson's report can be found in Sec. 7.3 of [80].

VII. Discussion

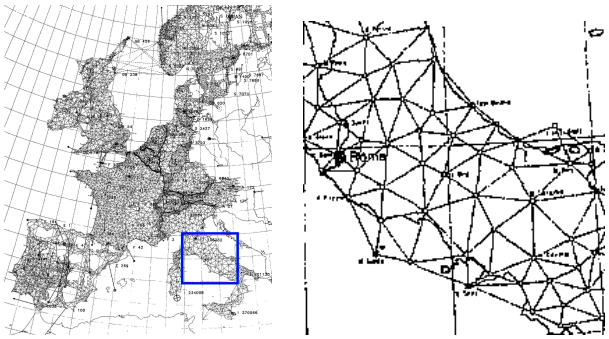
Although there are similarities between the problems of both communities, it is also important to highlight the additional challenges that autonomous robots, which rely on working SLAM solutions, face compared to geodetic mapping.

First, SLAM systems are completely autonomous while geodetic mapping inherently involves humans at all levels of the process. It is difficult for automated front-end data-association methods to distinguish between visually similar but physically different places and this is likely to occur, for example, in large man-made buildings. Perceptual aliasing creates false constraints, which often introduces errors in the localization and map building process.

Second, the quality of the initial guess is often different. The initial guess that is available for geodetic triangle networks are typically substantially better than the pose initializations of typical wheeled robots using odometry as well as flying or walking robots. A good initial guess substantially simplifies and even enables the use of polygon filling and other types of approximations.



(a) Intel dataset with 875 robot positions and 15675 constraints (left) and a zoomed in section (right). The robot positions are shown as blue triangles and constraints in red. The zoomed in section of the bottom left portion intuitively shows the non-planarity of a SLAM graph.



(b) Triangle net for ED87 (left) and a zoomed-in section for Italy (right). The constraints covering Italy intuitively show that triangle nets used for geodesy were almost planar. Figure courtesy of [47].

Fig. 17. Intuitive illustration of planarity of geodetic triangle net compared to a SLAM graph.

Third, the geodetic triangle networks are almost planar, while most SLAM graphs are not. This can be intuitively seen from Figure 17. Additionally Eiffel-tower type landmarks which connect all poses, create highly non-planar SLAM graphs [81]. Helmert's simple partitioning scheme of segmenting along latitudes and longitudes works for Geodetic networks because the graphs are almost planar. In [72], the authors prove that planar or approximately planar SLAM graphs can be optimized in $O(n^{1.5})$ by using the nested dissection reordering. Comparable results can be expected for Helmert's Blocking strategy as both are rather similar [67]. The way most modern SLAM methods work, however, leads to a highly non-planar graph with a high crossing number [82].

VIII. Conclusion

This paper provides a survey of geodetic mapping methods and aims at providing a geodetic perspective on SLAM. We showed that both fields share similarities when it comes to the error minimization task: maps are large, computational resources are limited and incremental methods are required, non-linear constraints require iterative procedures and data associations is erroneous. There are, however, also differences: geodetic triangular nets have a simpler structure, which can be exploited in the optimization, methods for robotics must be completely autonomous while the geodetic surveys always have humans in the loop, and often the geodetic community had a better initial configuration to begin with.

Besides the elaborated similarities and differences between geodetic mapping and SLAM, we surveyed several core techniques developed by the geodetic community and related them to state-of-the-art SLAM methods. The central motivation for this paper is to connect both fields and to enable future synergies among them. While surveying the geodetic methods, we experienced strong respect towards the geodetic scholars. Their achievements, especially given their lack of computational resources, are outstanding.

SLAM researchers have often gone back to the graph theory and sparse linear algebra community for efficient algorithms. It is probably worth to also look into the geodetic mapping literature given that they addressed large-scale error minimization and developed highly innovative solutions to solve them. Actually, several research activities by linear algebra researchers have been motivated by the large problem instances of geodetic mapping.

There might still be more methods in geodetic mapping that are unknown outside their community but could inspire other fields. Interested readers should begin with the excellent document by Schwarz on the history of North American Datum of 1983 [5].

Acknowledgement

We like to thank Edwin Olson for introducing the work of Schwarz [5] to the authors and our librarian Susanne Hauser for locating some of the old geodetic journals and manuscripts. We would also like to thank Google scholar for managing to crawl and find several of the discussed papers. We would also like to thank Heiner Kuhlmann, Giorgio Grisetti, Steffen Gutmann, Frank Dellaert as well as the reviewers for their valuable feedback.

This work has partly been supported by the European Commission under FP7-600890-ROVINA and ERG-AG-PE7-267686-LifeNav and by the BMBF under contract number 13EZ1129B-iView.

References

[1] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, 1997.

[2] J. Folkesson and H. I. Christensen, "Graphical SLAM - a self-correcting map," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2004, pp. 383–390.

[3] S. Thrun and M. Montemerlo, "The GraphSLAM algorithm with applications to large-scale mapping of urban structures," *International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.

[4] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Transactions on Intelligent Transportation Systems Magazine*, vol. 2, pp. 32–43, 2010.

[5] C. R. Schwarz, "North American datum of 1983," US Department of Commerce, National Oceanic and Atmospheric Administration (NOAA), Rockville, MD, Tech. Rep. NOAA Professional Paper NOS 2, 1989. [Online]. Available: http://www.ngs.noaa.gov/PUBS_LIB/NADof1983.pdf

[6] F. Helmert, *Die mathematischen und physikalischen Theorien der höheren Geodasie-Einleitung*. Leipzig, Germany: B.G. Teubner, 1880.

[7] P. Agarwal, W. Burgard, and C. Stachniss, "Helmert's and Bowie's geodetic mapping methods and their relationship to graph-based slam," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[8] G. B. Kolata, "Geodesy: dealing with an enormous computer task," *Science*, vol. 200, no. 4340, pp. 421–466, 1978.

[9] H. Johannsson, M. Kaess, M. Fallon, and J. J. Leonard, "Temporally scalable visual slam using a reduced pose graph," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 54–61.

[10] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent, "Efficient sparse pose adjustment for 2d mapping," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 22–29.

[11] E. Olson and P. Agarwal, "Inference on networks of mixtures for robust robot mapping," in *Proceedings of Robotics: Science and Systems (RSS)*, 2012, pp. 313–320.

[12] N. Sünderhauf and P. Protzel, "Towards a robust back-end for pose graph slam," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 1254–1261.

[13] Y. Latif, C. C. Lerma, and J. Neira, "Robust loop closing over time," in *Proceedings of Robotics: Science and Systems (RSS)*, 2012, pp. 233–240.

[14] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, "Robust map optimization using dynamic covariance scaling," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 62–69.

[15] L. Carlone, R. Aragues, J. Castellanos, and B. Bona, "A linear approximation for graph-based simultaneous localization and mapping," in *Proceedings of Robotics: Science and Systems (RSS)*, 2011, pp. 41–48.

[16] G. Grisetti, R. Kummerle, and K. Ni, "Robust optimization of factor graphs by using condensed measurements," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 581–588.

[17] H. Wang, G. Hu, S. Huang, and G. Dissanayake, "On the structure of nonlinearities in pose graph slam," in *Proceedings of Robotics: Science and Systems (RSS)*, 2012, pp. 425–432.

[18] L. Carlone, "A convergence analysis for pose graph optimization via gauss-newton methods," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 965–972.

[19] G. Hu, K. Khosoussi, and S. Huang, "Towards a reliable slam back-end," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 37–43.

[20] P. Agarwal, G. Grisetti, G. D. Tipaldi, L. Spinello, W. Burgard, and C. Stachniss, "Experimental analysis of dynamic covariance scaling for robust map optimization under bad initial estimates," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[21] M. Kaess, A. Ranganathan, and F. Dellaert, "Fast incremental square root information smoothing," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2007, pp. 2129–2134.

[22] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[23] F. Lu and E. Milios, "Robot pose estimation in unknown environments by matching 2D range scans," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994, pp. 935–938.

[24] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2004, pp. 2743–2748.

[25] E. Olson, "Real-time correlative scan matching," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009, pp. 4387–4393.

[26] G. D. Tipaldi, L. Spinello, and W. Burgard, "Geometrical FLIRT phrases for large scale place recognition in 2d range data." in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013, pp. 2693–2698.

[27] G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg, "Hierarchical optimization on manifolds for online 2D and 3D mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 273–278.

[28] C. Hertzberg, "A framework for sparse, non-linear least squares problems on manifolds," Master's thesis, Universität Bremen, department of Computer Science, 2008.

[29] J.-S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proceedings of the IEEE International Symposium on Computational Intelligence (CIRA)*, 1999, pp. 318–325.

[30] K. Konolige, "Large-scale map-making," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2004, pp. 457–463.

[31] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[32] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Fast incremental smoothing and mapping with efficient data association," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 1670–1677.

[33] A. Howard, M. J. Mataric, and G. Sukhatme, "Relaxation on a mesh: a formalism for generalized localization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, 2001, pp. 1055–1060.

[34] T. Duckett, S. Marsland, and J. Shapiro, "Learning globally consistent maps by relaxation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, San Francisco, CA, 2000, pp. 3841–3846.

[35] U. Frese, P. Larsson, and T. Duckett, "A multilevel relaxation algorithm for simultaneous localization and mapping," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 196–207, April 2005.

[36] E. Olson, J. Leonard, and S. Teller, "Fast iterative optimization of pose graphs with poor initial estimates," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 2262–2269.

[37] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, "A tree parameterization for efficiently computing maximum likelihood maps using gradient descent," in *Proceedings of Robotics: Science and Systems (RSS)*, 2007, pp. 65–72.

[38] E. Olson, J. Leonard, and S. Teller, "Spatially-adaptive learning rates for online incremental SLAM," in *Proceedings of Robotics: Science and Systems (RSS)*, 2007, pp. 73–80.

[39] G. Grisetti, D. L. Rizzini, C. Stachniss, E. Olson, and W. Burgard, "Online constraint network optimization for efficient maximum likelihood map learning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 1880–1885.

[40] M. Bosse, P. Newman, J. Leonard, and S. Teller, "Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework," *International Journal of Robotics Research*, vol. 23, no. 12, pp. 1113–1139, December 2004.

[41] C. Estrada, J. Neira, and J. Tardós, "Hierachical SLAM: Real-time accurate mapping of large environments," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 588–596, 2005.

[42] K. Ni, D. Steedly, and F. Dellaert, "Tectonic SAM: Exact; out-of-core; submap-based slam," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 1678–1685.

[43] K. Ni and F. Dellaert, "Multi-Level submap based SLAM using nested dissection," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 2558–2565.

[44] L. Paz, J. Neira, and J. Tardós, "Divide and conquer: Ekf slam in o(n)," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1107–1120, 2008.

[45] Natural Resources Canada, "100 years of geodetic surveys in canada," 2009. [Online]. Available: http://www.geod.nrcan.gc.ca/timeline/timeline_e.php

[46] R. K. Burkard, *Geodesy for the Layman*. US Department of commerce, National Oceanic and Atmospheric Administration, St. Louis, Missouri, 1983. [Online]. Available: http://www.ngs.noaa.gov/PUBS_LIB/GeoLay.pdf

[47] W. Ehrnsperger, "The ED87 Adjustment," *Journal of Geodesy*, vol. 65, no. 1, pp. 28–43, 1991.

[48] K. O. Milbert, "An evaluation of the High Accuracy Reference Network relative to the Continuously Operating Reference Stations," National Geodetic Survey, National Ocean Service, NOAA. [Online]. Available: http://www.ngs.noaa.gov/PUBS_LIB/HARN_CORS_COMP/eval_harn_to_cors.html

[49] Dale G. Pursell and Mike Potterfield, "NAD 83(NSRS2007) National Readjustment Final Report," National Oceanic and Atmospheric Administration, National Ocean Service, Silver Spring, MD, Tech. Rep. NOS NGS 60, 2008. [Online]. Available: http://www.ngs.noaa.gov/PUBS_LIB/NSRS2007/NOAATRNOSNGS60.pdf

[50] H. Wolf, "Triangulation adjustment general discussion and new procedure," *Bulletin Géodésique*, vol. 16, no. 1, pp. 87–104, 1950.

[51] J. D. Bossler, "Geodesy solves 900,000 equations simultaneously," *Eos, Transactions American Geophysical Union*, vol. 68, no. 23, pp. 569–569, 1987.

[52] Aeronautical Chart and Information Center, *Preface and Part I: The Mathematical Theories*, St. Louis 18, MO., 1964.

[53] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustmenta modern synthesis," in *Vision algorithms: theory and practice*. Springer, 2000, pp. 298–372.

[54] H. Wolf, "The helmert block method, its origin and development," in *Proceedings of the Second International Symposium on Problems Related to the Redefinition of North American Geodetic Networks*, 1978, pp. 319–326.

[55] F. W. Hough, "The adjustment of the central european triangulation network," *Bulletin Géodésique*, vol. 7, no. 1, pp. 64–93, 1948.

[56] O. S. Adams, *The Bowie Method of Triangulation Adjustment, as applied to the first-order net in the Western part of the United States*. US Government Printing Office, 1930.

[57] H. Wolf, "Gaußscher Algorithmus (Doolittle-Methode) und Boltzsches Entwicklungsverfahren," *Bulletin Géodésique*, vol. 26, no. 1, pp. 445–452, 1952.

[58] H. Boltz, *Entwickelungs-Verfahren zum Ausgleichen Geodätischer Netze nach der Methode der kleinsten Quadrate*. Veröffendlichungen des Geodätischen Instituts Potsdam, 1923, no. 90.

[59] ——, *Substitutions-Verfahren zum Ausgleichen grosser Dreiecksnetze in einem Guss nach der Methode der kleinsten Quadrate*. Veröffendlichungen des Geodätischen Instituts Potsdam, 1938, no. 108.

[60] K. Kersting, "Lifted probabilistic inference," in *Proceedings of 20th European Conference on Artificial Intelligence (ECAI–2012)*, 2012, pp. 33–38.

[61] American Congress on Surveying, American Society for Photogrammetry and Remote Sensing, and American Society of Civil Engineers, "The glossary of the mapping sciences," American Society of Civil Engineers, Bethesda, Maryland, 1994.

[62] G. Bomford, "The readjustment of the Indian triangulation," *Professional paper No. 28, Survey of India, Dehra Dun*, 1939.

[63] ——, *Geodesy*. Oxford University Press, London, 1952.

[64] A. George, "Nested dissection of a regular finite element mesh," *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 345–363, 1973.

[65] T. A. Davis, *Direct methods for sparse linear systems*. Society for Industrial and Applied Mathematics, 2006, vol. 2.

[66] P. Agarwal and E. Olson, "Variable reordering strategies for SLAM," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2012, pp. 3844–3850.

[67] J. Avila and J. Tomlin, "Solution of very large least squares problems by nested dissection on a parallel processor," in *Proceedings of the 12th Symposium on the Interface*, 1979, pp. 9–14.

[68] G. H. Golub and R. J. Plemmons, "Large-scale geodetic least-squares adjustment by dissection and orthogonal decomposition," *Linear Algebra and Its Applications*, vol. 34, pp. 3–28, 1980.

[69] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[70] Z. Zhang, "Parameter estimation techniques: A tutorial with application to conic fitting," *Image and Vision Computing Journal*, vol. 15, pp. 59–76, 1997.

[71] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 3607–3613.

[72] P. Krauthausen, F. Dellaert, and A. Kipp, "Exploiting locality by nested dissection for square root smoothing and mapping," in *Proceedings of Robotics: Science and Systems (RSS)*, 2006, pp. 73–80.

[73] G. Grisetti, C. Stachniss, and W. Burgard, "Nonlinear constraint network optimization for efficient map learning," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, no. 3, pp. 428–439, 2009.

[74] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard, "Efficient estimation of accurate maximum likelihood maps in 3D," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 3472–3478.

[75] G. Dubbelman, P. Hansen, B. Browning, and M. B. Dias, "Orientation only loop-closing with closed-form trajectory bending," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 815–821.

[76] T. Davis, "Suitesparse," 2006. [Online]. Available: http://www.cise.ufl.edu/research/sparse/SuiteSparse/

[77] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, "Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate," *ACM Trans. Math. Softw.*, vol. 35, pp. 22:1–22:14, 2008.

[78] F. Helmert, "Über die Wahrscheinlichkeit der Potenzsummen der Beobachtungsfehler und über einige damit im Zusammenhang stehende Fragen," *Zeitschrift für Mathematik und Physik*, vol. 21, pp. 192–218, 1876.

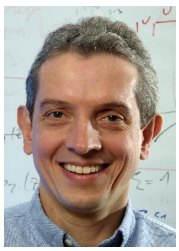
[79] W. Kruskal, "Helmert's distribution," *The American Mathematical Monthly*, vol. 53, no. 8, pp. 435–438, 1946.

[80] O. Sheynin, "Helmert's work in the theory of errors," *Archive for history of exact sciences*, vol. 49, no. 1, pp. 73–104, 1995.

[81] F. Dellaert, J. Carlson, V. Ila, K. Ni, and C. E. Thorpe, "Subgraph-preconditioned conjugate gradients for large scale slam," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 2566–2571.

[82] P. Hliněný and G. Salazar, "On the crossing number of almost planar graphs," in *Graph Drawing*. Springer, 2007, pp. 162–173.

Pratik Agarwal received his B.E. degree in Computer Science and Engineering from Manipal Institute of Technology, India in 2010, and M.S.E. in Computer Science and Engineering from University of Michigan, Ann Arbor in 2012, where he also worked as a research assistant at the APRIL robotics laboratory. Currently, he is a Doctoral Research Assistant at the Autonomous Intelligent Systems Lab, University of Freiburg, Germany. His research interests include robust robot mapping, sparse linear algebra and multi-object multi-sensor tracking.



Wolfram Burgard studied computer science at the University of Dortmund, Germany, and received his Ph.D. degree from the Department of Computer Science of the University of Bonn in 1991. In 1999 Wolfram Burgard became professor at the Department of Computer Science of the University of Freiburg where he heads the research laboratory for Autonomous Intelligent Systems. His areas of interest lie in artificial intelligence and robotics. They cover various aspects of mobile robotics including mobile robot navigation, multi-robot systems, state estimation, robot learning, and human robot interaction.

Cyrill Stachniss is currently a professor for photogrammetry at the University of Bonn. Before that, he was a lecturer at the University of Freiburg. He received the habilitation in 2009 after being an academic advisor at the University of Freiburg and a senior researcher at the Swiss Federal Institute of Technology. In 2006, he finished his PhD with a thesis entitled "Exploration and Mapping with Mobile Robots" at the University of Freiburg. From 2008-2013, he was an associate editor of the IEEE Transactions on Robotics and he is a Microsoft Research Faculty Fellow since 2010. In 2012, he was selected as a Robotics: Science and Systems Early Career Spotlight and received the IEEE RAS Early Career Award in 2013. In his research, he focuses on probabilistic techniques in the context of mobile robotics, perception, mapping, and navigation problems.