OBTAINING APPROXIMATIONS FOR RANGE-BASED FREE-NETWORK ADJUSTMENT

A. Ladai[1,*], C. Toth[2]

[1]Department of Manufacturing and Civil Engineering, Norwegian University of Science and Technology (NTNU) in Gjøvik, Teknologiveien 22, 2815 Gjøvik, Norway - andras.ladai@ntnu.no

[2]Department of Civil, Environmental and Geodetic Engineering, The Ohio State University, USA - toth.2@osu.edu

KEY WORDS: Geodetic Networks, Least Squares Adjustment, Approximations

ABSTRACT:

Airborne and ground mobile mapping platforms are increasingly used in group formations to increase productivity or complement each other in terms of improving observation capacity and efficiency in the surveyed area. Similarly, the navigation of assisted and autonomous vehicles presents a similar problem of sharing a small place where, for example, accurate relative positioning is essential to avoid collisions. Estimating platform relative positions from inter-platform range measurements is important in these applications, as they can provide valuable information to improve individual platform navigation or can potentially detect anomalies in these solutions that could be caused by unintentional or intentional disturbances. Free-network adjustment based on ranges forms the baseline solution to obtain relative positions. The challenge is to provide adequate platform position approximations for the least squares adjustment to achieve both quick convergence and fast execution time. Here we propose a preprocessing method that creates suitable approximations based on range values.

1. INTRODUCTION

Providing mobility in urban areas is an essential requirement for achieving quality life in modern societies. There is a large variety of platforms that support, for example, public transit, special needs of handicap people and the like. Recently, highly automated platforms have been introduced and deployed in large numbers, which are very effective to improve mobility services by extending range and making them affordable. As the number of platforms keeps increasing in the same space, there is a need to collaborate the navigation of the platforms that are close to each other. For example, delivery services based on ground vehicles and UASs are expected to rapidly expand in the near future. The collaboration between platforms is primarily aimed at improving safety and efficiency.

The concept of collaborative navigation or cooperative positioning is based on data sharing between navigation platforms, such as a mapping swarm (Masiero et al., 2021). If these platforms share their navigation data, such as integrated GNSS and IMU solution as well as range and angular observations between platforms, if available, then a combined navigation filter can compute an optimal solution for the entire group of platforms. In particular, if inter-platform range data is available, a totally independent relative position solution can be obtained by using a free-network adjustment. This could be valuable for collision detection or anomaly detection and mitigation of individual platform navigation solution.

There is a proliferation of sensors, as navigation as well as imaging sensors, such as camera, RF signals, LiDAR, etc., are increasingly deployed on ground and air platforms (Toth and Jozkow, 2015). Most of these sensors provide some range and/or angular measurements that allow for detecting, ranging and, in general, tracking other platforms, giving an opportunity for collaborative navigation.

In a previous effort, a free-network adjustment was developed and tested using a static network of nodes, including the investigation of convergence and execution time (Ladai and Toth, 2022). The early experiences have clearly indicated that the initial values, as expected, play a critical role in obtaining a valid solution. This study extends on the former work by introducing an approximation creating process to support the adjustment. Though the approximation estimation is designed for the 3D case, the current study reports only initial results using a 2D network.

2. ESTIMATING APPROXIMATIONS

Least squares adjustment is a proven and widely used technique for geodetic network computations. However, the success of the adjustment depends a lot on the initial approximations, as without relatively close initial values, convergence is not guaranteed. This is especially the case for free-network adjustment, when only range measurements are available, and thus creating approximations is a task on its own. Note that historically, there was rarely a case of a completely free-network approximations as some of the network points always had coordinates (anchor points). Earlier, we experienced with using random initial values, which interestingly provided solution in a relatively large number of cases for small and moderate size networks. But the chance for obtaining convergence is quickly diminishing as the number of point increases (Ladai and Toth, 2022). Here we propose an algorithm to estimate initial values free-network adjustment.

2.1 Concept

Given the internodal ranges of a set of points in two or three dimensions, our objective is to estimate the coordinates of a subset of the points based on the distance matrix, which contains all the ranges available for computation. Note that we do not necessarily have a range measurement between any two points.

An optional weight matrix (with values of 1 or less), if defined, specifies our level of confidence in the range measurements. Not using weight matrix, zero numbers in the distance matrix indicate missing range measurements. Our process does not limit the number of points, it can be any, but not fewer than 4. There is,

* Corresponding author

however, some minimum requirement also for the spatial distribution of measurements as we will explain it in detail later. In case the number of points is large, i.e., greater than a predefined constant used in our implementation, or we have a incomplete distance matrix, i.e., we do not have range measurements between any 2 points, we will create an estimate only for a subset of the points. Since there are only ranges, first we need to define a local coordinate system before determining the coordinates of a potentially smaller subset of points which has all the distances measured between them; we call such points fully connected. The set of points, the coordinates of which will be determined by the process are called core points. The core point set should be fully connected, i.e., we need to have range measurements between any two points, and its size should not exceed some predefined (configurable in our implementation) number of points, $\boldsymbol{N_n}$.

2.2 Algorithm

The flowchart of the proposed algorithm is shown in Fig. 1.

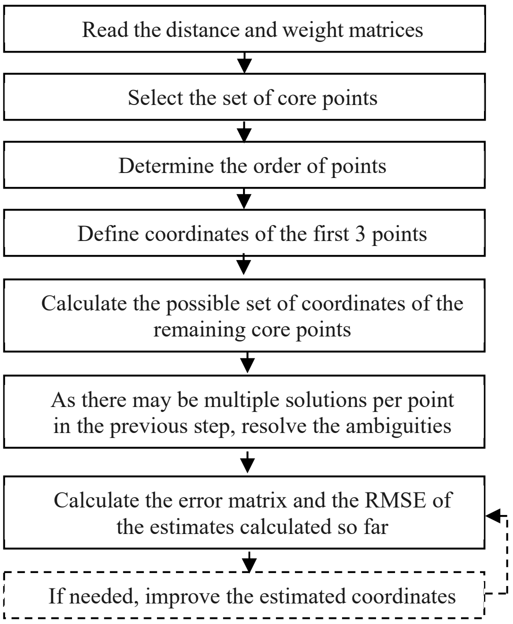

Figure 1. Flowchart of the algorithm

Select the set of core points, i.e., the set of points the coordinates of which will be determined by this process. In this step, we use a recursive algorithm to find a subset of the points which are fully connected (measured). Note that the algorithm will stop once the number of such points found has already reached our predefined limit. If there are fewer than 4 points with range measurements between any 2 points, the algorithm will exit with an insufficient measurements error message.

Determine the order of points to be calculated (limited to the core points) includes the following steps:

First, we assign a rank to each point based on the following:

- Largest distance measured from the point multiplied by the weight (accuracy) of that range measurement
- Number of measurements available for the point; important only for incomplete distance matrices
- Sum of distance squared of all other points from the point being evaluated

Ultimately, our algorithm gives preference to points with the highest reliable range measurement from each other and with the most measurements available (again, for incomplete distance matrices). Each of the factors mentioned above are taken into account using a few parameters we configure based on observations, and finally, we sort the points by the rank number calculated in the previous step high-to-low for all the core points

Define coordinates for the first 3 points, illustrated in Fig. 2, as:

- The first point will be the origin of the coordinate system
- The second point will lie on the X axis, i.e., the second point will have non-zero X coordinate, but 0 for Y and Z coordinates
- The 3rd point will have 0 as its Z coordinate, and its X and Y coordinates are calculated based on its distance from the first 2 points using the law of cosine

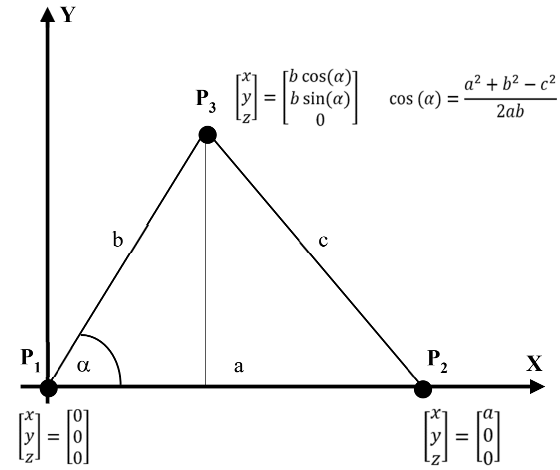

Figure 2. Assigning coordinates to the first 3 points

Calculate the possible set of coordinates of the remaining core points in the order determined before. We calculate the coordinates of the remaining core points based on their distances from the first 3 points. There could be zero, one, or two potential coordinate solutions. For each point *n*:

- If a distance value would result in inconsistent measurements, i.e., for any of the first 3 points, j and k
$$d(j,n) + d(n,k) < d(j,k)$$
we log the discrepancy so that the measurements can be potentially verified, and the distances $d(j,n)$ and $d(n,k)$ used in this step are proportionately adjusted (increased) until the inconsistency goes away; note that we still keep the original distance values for any further steps
- We determine the set of possible coordinates by the intersection of 3 spheres centered around the first 3 points based on the distances between those points and point *n*. There can be 0, 1, and 2 solutions.

Fig. 3 shows the determination of a point, P_4 where all the ranges, $\binom{4}{2}$ are available, so this point becomes a core point.

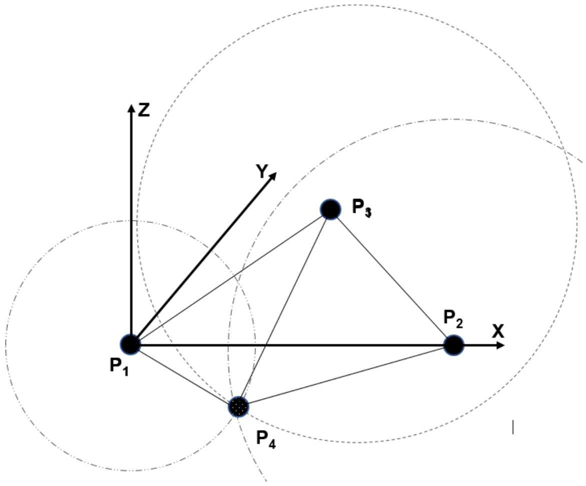

Figure 3. Determining the coordinates of point 4 by the intersection of the 3 spheres around points 1, 2, and 3

Resolving ambiguities is needed, as there might be multiple solutions in the previous step, which must be filtered for validity. Since the number of core points is limited to a predefined number, we can simply enumerate all combinations of the points; for *n* points, maximum 2^n checks is needed. We select the best combination using least squares, i.e., the combination which has the lowest sum of squared of errors.

Calculating the error matrix and the root mean square deviation (RMSE) of the points calculated so far is straightforward.

Improving the estimated coordinates is an optional step. Similar to outlier detection, the point with the highest error value is selected. The ranges to the other points are analyzed and correction is applied.

The approximation algorithm was implemented in Java and used to create initial coordinates to support the investigation of the free-network computation. Note that the free-network adjustment is implemented in Matlab environment.

3. PERFORMANCE EVALUATION

The free-network formation algorithm developed earlier (Ladai and Toth, 2022) was tested on multiple measurement configurations with applying various approximations. The 2D ground control network used in the May 2022 data acquisition campaign at OSU (Suleymanoglu et al., 2023) was used in this investigation. While we had ranges measured between any 2 points, the network code was also tested with incomplete distance matrices by omitting some of the distance measurements.

Various network configurations and parameter combinations were tested, including

- The number of approximated points used
- Different distribution of missing range measurements
- Spatial distribution of points with approximations
- The impact of the orientation between the network and initial approximations

3.1 The network

An early version of this free-network adjustment was tested on a control point network at the NTNU campus (Ladai and Toth, 2022). In this study, we used the OSU ground network, shown in Figure 4, which has 13 node points and 78 range observations, ***N***; note there are no angular/orientation observations.

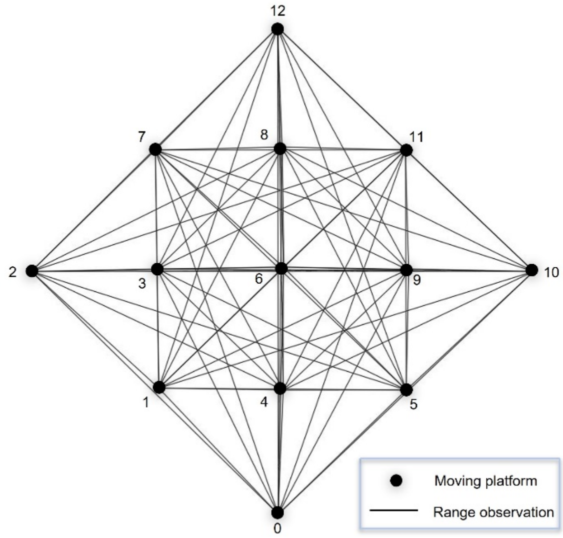

Figure 4. The OSU ground network

There are several benefits of using this physically existing network for our investigations:

- The OSU dataset includes measurements from multiple static and mobile platforms, and thus, it offers 3D networks for continuing investigations
- It has a symmetric shape which is challenging, and therefore, ideal for performance testing
- Since it is a physically existing network, small random deviations from the symmetric shape ensure that our results are representative of real-world situation
- The OSU network and measurement data have been used multiple research efforts

Below, typical examples for using approximations in various configurations are shown, providing an initial overview of the performance of the iterative adjustment method. In terms of statistics, the number of approximations, the number of range measurements and the number of iterations are listed and analyzed for comparisons.

3.2 Approximations

For any iterative adjustment process, initial values are needed. In most practical cases, there are realistic initial values available, but for a free-adjustment based on ranges only, there is no obvious way to provide approximations for all the network nodes. There are a few options:

1. Assign random values to the nodes
2. Assign values based on geometric primitives, such as line, circle, etc., spatial distribution
3. Create a dedicated process to generate initial values for some of the nodes

In general, only the last option is considered approximation as the initial values related to the real situation. In our earlier work, we used approach (2), more specially, creating initial values distributed along a line, which are not realistic approximations.

First, we tested how our algorithm performs on the OSU network when all the observations are available with and without approximations. When no approximations are given, i.e., initial values are based on a line pattern, the algorithm fails on network: it stops after 40 iterations with no valid solution. As expected, when all the nodes have approximated positions, the algorithm runs flawless and gives valid solution. Not surprisingly, when some of the range observations are removed, the adjustment produces good results without approximations. This simply means that the line pattern initial values better match the network with less range data; in other words, there is less constraint on the network. The statistics of the three cases are listed in Table 1, and the solutions are visualized in Fig. 5. The left column shows the status of the network nodes, whether there is approximation available for nodes. The mid column depicts the initial values; note the line shape of the network when no approximation is available. Finally, the right column displays results of the adjusted network. Note that test #3 is just one example of many valid solutions with incomplete distance matrix.

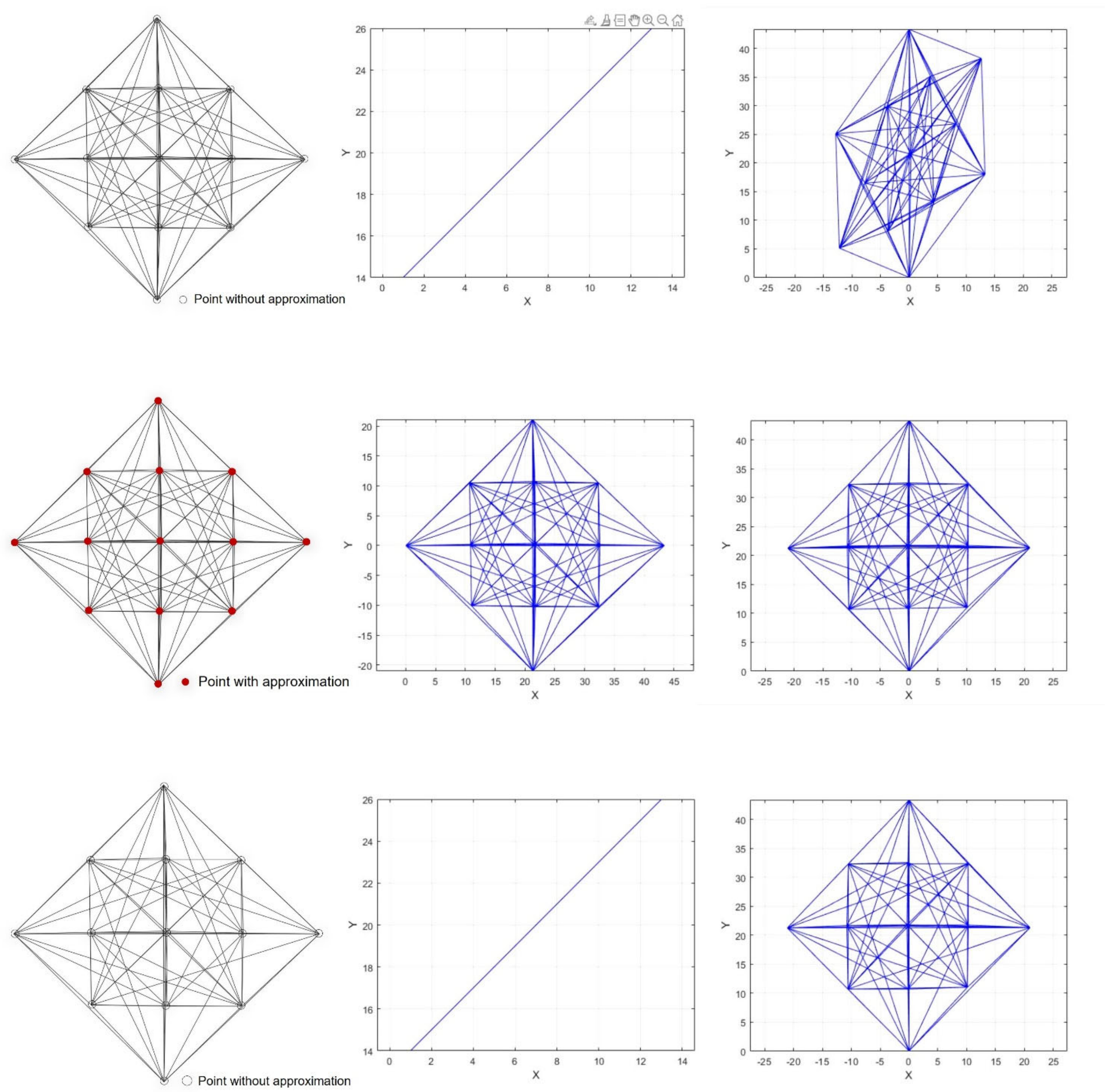

Figure 5. Some comparisons of full approximated network vs. no approximations

Test ID	Active ranges	Inactive ranges	Approximations	Number of iterations	Solution
1	78	0	0	40	Invalid
2	78	0	13	6	Valid
3	69	9	0	6	Valid

Table 1. Statistics of full approximated network vs. no approximations

3.3 Complete range matrix with various patterns of 4 approximations

This is an important case for investigation, as the premise is that providing good approximation for some of the network nodes, the chances are higher to get a valid solution; i.e., no need to have realistic values for all nodes. Therefore, the four-approximation case was investigated, since the four approximated points could provide a minimal redundancy in the 13-point 2D network. Various spatial distribution patterns of the approximated points were tested systematically. The overall results of this investigation are that all the configurations provide valid adjustment results. However, the number of iterations varies a lot between 6 and 24 as shown in Table 2.

Cases #7-10 should be identical in a fully symmetric system. However, they seem to perform slightly differently. The same stands for the pair of #11 and #12. While cases #4, #5 and #6 are not identical, they still show relatively similar patterns. These examples indicate that our algorithm is sensitive for the orientation of the network, which relates to the fact that the points without approximations have the linearly distributed initial values, which is clearly not symmetrical.

Test ID	Approximated nodes (IDs)	Number of iterations	Solution
4	0, 2, 12, 10	24	Valid
5	1, 7, 11, 5	10	Valid
6	4, 3, 8, 9	20	Valid
7	0, 1, 4, 5	6	Valid
8	2, 7, 3, 1	6	Valid
9	7, 8, 11, 12	15	Valid
10	5, 9, 10, 11	7	Valid
11	0, 4, 8, 12	20	Valid
12	2, 3, 9, 10	16	Valid

Table 2. Statistics of cases with complete range matrices and various patterns of 4 approximations

3.4 Various patterns of incomplete range matrix with 4 approximations

Next, we investigated different scenarios of incomplete range matrices with four approximated points. By taking away range observations one-by-one, we tried to determine the minimum number of range observations necessary for successful network adjustment.

Given the very large number of range measurement combinations that can be removed from the distance matrix (for *m* removed ranges, there are $\binom{N}{m}$ possibilities), it is difficult to systematically test the impact of using less ranges with four approximations.

By taking away range observations one-by-one, we experienced issues after missing 12 observations; again, this highly depends on the pattern, as described above. Also, in some cases, incomplete range matrix gives better performance (fewer iterations) than the complete range matrix while having the same geometry.

The general experiences are that no clear tendencies can be identified based on our limited tests. The performance of the algorithm does not necessarily correspond to the changes in the parameters; i.e., the pattern of approximations and then the pattern of active range observations. As experienced earlier, the higher number of active observations do not mean always less iterations. Also, the minimum number of necessary observations is not defined, since the spatial distribution of the point does have impact on whether the process is successful or not.

3.5 Specific cases

Each point is connected only to the 4 nearest neighbors. This case is intended to simulate a situation where range observations are available only to the four nearest neighbor platforms; for example, due to week signal reception or by design such as a bigger network is subdivided to smaller subnetworks. The network is shown on the left side of Fig. 6. The center image shows clearly the nodes organized along the approximated positions of the center lines. Also, the four points with only initial approximations are not connected (upper left corner of the center image), since there is no range observation between them. While the adjustment did not provide correct solution, the image on the right side of this figure shows that the process was partially correct, as due to symmetry the network folded along one axis. Note that in this case, there are 28 ranges available from the 78 potentially available ones. In summary, this network arrangement works only with complete and near complete node approximations.

We were also interested in seeing the inverse of the 4 neighbors' network case; in other words, all the ranges are used except the four to the closest neighbors. This configuration has significantly more active range observations, 50 instead of 28, so better results are expected. However, the tests show similar performance to the 4-neighbor case; the adjustment with full approximations works well, while lowering the number of approximations will not provide correct results. For comparison, Fig. 7 shows the results of the inverse pattern case with the same approximated nodes. In Table 3, statistics are also provided for better comparison of the 4 nearest neighbors' case and its inverse.

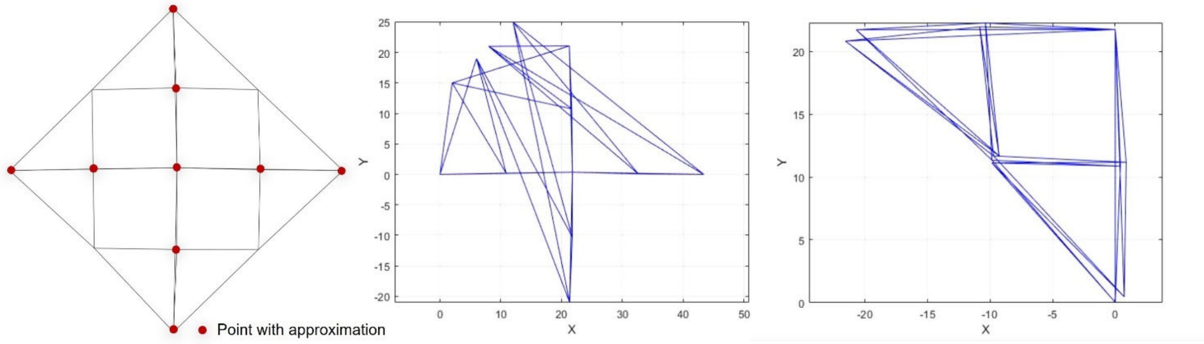

Figure 6. 4 nearest neighbors' network with 9 approximations

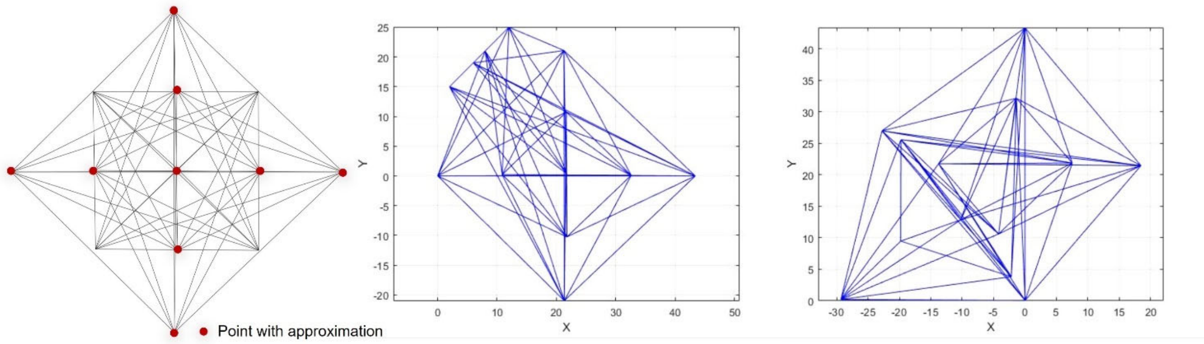

Figure 7. The inverse case of the 4 nearest neighbors' network with 9 approximations

Test ID	Active ranges	Inactive ranges	Approximations	Number of iterations	Solution
13	28	50	13	12	Valid
14	28	50	9	14	Invalid
15	50	28	13	5	Valid
16	50	28	9	42	Invalid

Table 3. Statistics of the 4 nearest neighbors' network and its inverse

4. CONCLUSION

The importance of having adequate approximation for the nodes of a range observation-based free-network adjustment has been demonstrated through the examples shown in this paper. The given examples are focused on specific cases, including full and no approximations, minimum constrain approximations, etc. In addition, the number and the pattern of the available range observations are varied too. While the advantage of having approximations is clear, the optimal pattern of their distribution for achieving the most efficient adjustment calculation is uncertain. Clearly, the performance of the network adjustment depends not only on the number and pattern of the approximated nodes and the active range observations, but their relative orientation and the position of the nodes with respect to the initial approximations have significant impact on the performance of the adjustment calculation.

These results determine the future directions of our research. The impact of the shape and orientation of the distribution of the default approximation must be investigated. In general, the shape of the network can be described by using internal reliability measures. The correlation between the performance and the network's internal reliability should be determined too. Also, introducing weights based on the quality of the approximations might have positive impact on the performance of the algorithm.

REFERENCES

Ladai, A., C. Toth (2022) Range-based Network Formation to Support Georeferencing Of Mapping Swarm Platforms, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci*., XLVI-M-2-2022, 147–152, 2022, https://isprs-archives.copernicus.org/articles/XLVI-M-2-2022/147/2022/

Masiero, A.; Toth, C.; Gabela, J.; Retscher, G.; Kealy, A.; Perakis, H.; Gikas, V.; Grejner-Brzezinska, D. (2021) Experimental Assessment of UWB and Vision-Based Car Cooperative Positioning System. *Remote Sens.* 2021, 13, 4858. https://doi.org/10.3390/rs13234858

Matlab Computer Vision Toolbox™ https://www.mathworks.com/products/computer-vision.html

Sluis, B., Toth, Ch. (2021) 2-Dimensional Geometric Analysis of a Simple Free Network, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume V-4-2021, pp.147-152.

Snow, K., Schaffrin, B. (2021) *Adjustment Computations.* Based on Courses GS 650, GS 651, & GS 762, The Ohio State University.

Suleymanoglu, B., Toth, C., Masiero, A., and Ladai, A.: Monitoring the Environment in Smart Cities: The Importance of Geospatial Location Referencing, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLVIII-M-1-2023, 355–360, https://doi.org/10.5194/isprs-archives-XLVIII-M-1-2023-355-2023, 2023.

Toth, C., Jozkow, G. (2015) Remote Sensing Platforms and Sensors: A Survey, *ISPRS Journal of Photogrammetry & Remote Sensing*, 115 (2016), pp. 22-36.