A universal and robust computation procedure for geometric observations

Author

Prof. Dr.-Ing. Rüdiger Lehmann
University of Applied Sciences Dresden
Faculty of Spatial Information
Friedrich-List-Platz 1
D-01069 Dresden
GERMANY
Tel +49 351 462 3146
Fax +49 351 462 2191
mailto:Ruediger.Lehmann@htw-dresden.de
https://www.researchgate.net/profile/Ruediger_Lehmann2

Abstract

This contribution describes an automatic and robust method, which can be applied to all classical geodetic computation problems. Starting from given input quantities (e.g. coordinates of known points, observations) computation opportunities for all other relevant quantities are found. For redundant input quantities there exists a multitude of different computation opportunities from different minimal subsets of input quantities, which are all found automatically, and their results are computed and compared. If the computation is non-unique, but only a finite number of solutions exist, then all solutions are found and computed. By comparison of the different computation results we may detect outliers in the input quantities and produce a robust final result. The method does not work stochastically, such that no stochastic model of the observations is required. The description of the algorithm is illustrated for a practical case. It is implemented on a webserver and is available for free via internet.

Keywords: geodetic computations, geodetic networks, outlier detection

Accepted Manuscript of Survey Review, Volume 50, Issue 361, Pages 364-372 (2018)
https://dx.doi.org/10.1080/00396265.2017.1279844

1 Introduction

Today, for simple geodetic computations we use standard methods to derive desired quantities from observations. But there are complicated situations, where it is not immediately clear, if a desired quantity can be computed at all from the given quantities. Even so, it is not immediately clear whether there is a unique solution or not. (Can the area of a quadrangle be computed from three sides and two diagonals? Yes, but the solution might not be unique, see figure 1 for illustration.) For large geodetic networks we need approximate values for the coordinates of new points. This is a computation ultimately hard to automate, especially for less transparent network topologies. A computer algebra system (Maple, Mathematica etc.), which uses methods of symbolic computation to solve geometrical problems *(e.g. Fontijne and Mann 2007)*, is not useful here.

There are long-standing methods of the automatic generation of approximate values for the coordinates in geodetic networks (*Benning and Ahrens 1979, Benning 1978, Benning and Förstner 1979*). The methods have been advanced by *Vetter (1992, 2007)* using matrices from graph theory. Since that time geodetic network adjustment software automatically compute approximate values, but they are restricted to standard cases, e.g. where the new points can be generated by successive computations of polar points sticking to known points. The new approach presented by *Lehmann (2015)* is much more general and can handle any situation, however complicated it may be.

If there are redundant observations, i.e. more than they are actually required to compute the desired quantities, we perform a geodetic adjustment, usually in the sense of least squares (*e.g. Teunissen 2007*). If gross errors in the observations cannot be excluded, least squares adjustment is no longer optimal because those gross errors are not optimally adjusted. Here a robust adjustment is recommended, mostly a method from the large toolbox of M-estimation *(Huber 2009)*. By robustness we mean the property of the solution being insensitive to gross errors in the observation values. See (*Rousseeuw and Leroy 1987*) for more information.

As an alternative to robust estimation, the data snooping method is very popular in geodesy. Here gross errors causing outliers in the observations are not tolerated, but detected, identified and rejected. The basis of the method is statistical hypothesis testing, (see e.g. *Teunissen 2006*). Recently, this procedure has been extended by *Lehmann and Lösler (2016)* towards multiple testing and compared to the method of model selection by information criteria.

However, if the redundancy is too low, geodetic adjustment is sometimes not applied, because the improvement is very meagre. Even worse, it may not be achieved at all, because the introduced stochastic model (weight matrix, correlations etc.) is not realistic. In this case the optimality property of the adjustment results is not in effect. Instead it is advisable to compute the desired quantities in two different ways and compare them. (If we have observed all sides and diagonals of a quadrangle, the desired area can be computed twice by dismissing one diagonal at a time.) If the difference is small, the result is validated. In this simple method no stochastic model is employed.

If the redundancy is high, we could also apply this method, but we should try to find computation schemes, which use least-overlapping or even disjoint subsets of the observations. If there is a gross error in an observation value, then we have good chances that it falsifies only one of the computed values and will therefore show up in their difference. In a non-trivial case it is very difficult to find manually two computation schemes with this least-dependence or independence. We present such an example in section 3.

Lehmann (2015) presented an algorithm, which starting from a set of given quantities (start quantities) generates all possible computation schemes for a set of desired quantities. One could then select the two least-dependent or independent schemes. Moreover, one could compute all such schemes and compare all values obtained for a desired quantity. If there are gross errors in some start quantities, the values computed from them will often become the smallest or the largest in this set. If this affects less than 50% of those values, their median would then be unfalsified and can be used as a highly robust estimate of the desired quantity.

Accepted Manuscript of Survey Review, Volume 50, Issue 361, Pages 364-372 (2018)
https://dx.doi.org/10.1080/00396265.2017.1279844

In this presentation, Lehmann's algorithm is developed further. If the computation is non-unique, but only a finite number of solutions exist, then all solutions are found and computed. Additionally, if some computation schemes suffer from bad error propagation, they are dropped. Finally, the algorithm is extended towards the detection of outliers in the observation. The ultimate algorithm is illustrated for a practical case: the determination of an inaccessible point with auxiliary triangles. The promised robustness is demonstrated.

The improved algorithm is implemented on a webserver and can be used freely via internet (see appendix).

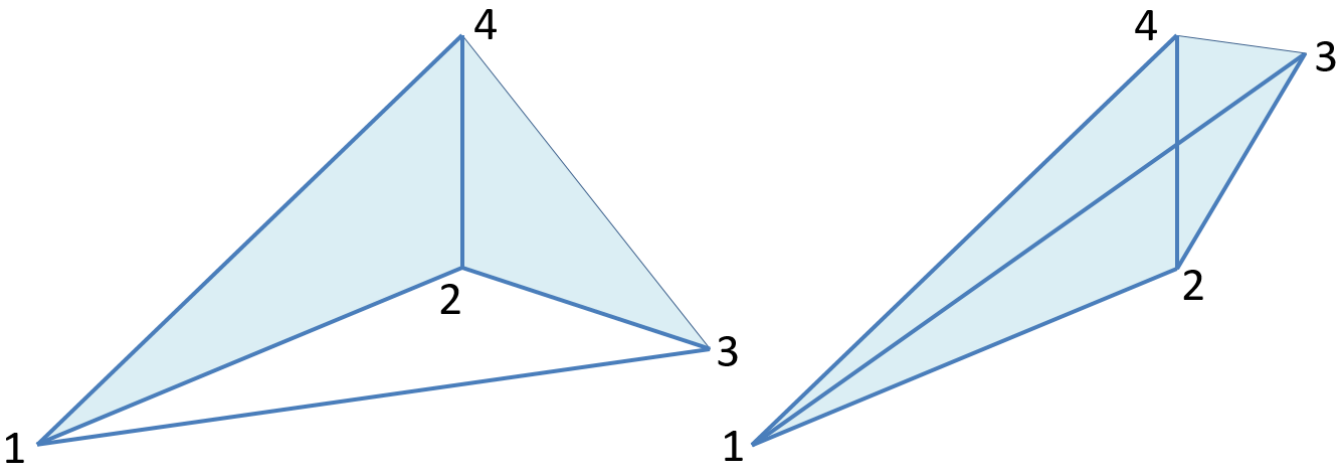


Figure 1: Two quadrangles (filled areas) agreeing in three sides 4-1,1-2,2-3 and two diagonals 1-3,2-4

2 The basic algorithm

2.1 Compiling a list of relevant quantities

Let us define a number of points in space, usually of dimension 2 or 3. Some coordinates X, Y, Z of these points may be known, some of them may be desired. Other points could possibly become relevant as intermediate points during the computation procedure.

Furthermore, let us introduce a set of geometric quantities defined for and between these points:

- point coordinates
- slope distances
- horizontal distances
- height differences
- azimuths
- horizontal angles, i.e. readings of the horizontal scale
- orientation angles, i.e. azimuth of zero reading
- zenith angles, i.e. readings of the vertical scale
- instrument heights and target heights
- areas of triangles, quadrangles, polygons
- etc.

Some of these quantities are observed, other quantities are desired. And again, other quantities are pure auxiliary quantities, which could possibly become relevant as intermediate quantities during the computation of the desired quantities. A list of such quantities (point coordinates and other quantities) is compiled.

Accepted Manuscript of Survey Review, Volume 50, Issue 361, Pages 364-372 (2018)
https://dx.doi.org/10.1080/00396265.2017.1279844

2.2 Compiling a catalogue of template computation rules

Let us introduce a full catalogue of template computation rules, which are satisfied by the true values of the quantities introduced in the preceding subsection:

- all basic geometric angle and triangle relationships
- all geodetic intersections (arcs-intersections, lines-intersections, etc.)
- all conversions between cartesian coordinates and polar quantities (distances, angles)

Each template rule is a formula, instructing the computer to obtain a target quantity from given values of other quantities.

Example 1: A template rule is

$$d_{PQ} = \sqrt{(X_Q - X_P)^2 + (Y_Q - Y_P)^2} \tag{1}$$

where d_{PQ} denotes the horizontal distance between points P and Q. Note that P and Q are not yet actual points, but only wildcards.

2.3 Compiling a list of applicable computation rules

Wherever possible, we apply this catalogue to the list of quantities set up in subsection 2.1, regardless of whether they are known or observed or desired or auxiliary quantities.

Example 1 (cont'd): (1) is applied to any pair of points, for which this computation rule could be relevant during the computation procedure. If in doubt, we could apply it to any pair of points, but this could make the computation slow. Often (1) is only required for the computation, if points are connected by a line of sight.

From this list of rules, all rules are successively deleted, which are not applicable, because there is no applicable rule for the computation of some unknown input quantity. This deletion continues, until no more rules can be deleted.

Example 1 (cont'd): If (1) is applied to a pair of points A and B, and the coordinates X_B, Y_B of B are neither known, nor exists a rule in the list of rules, which allows one to compute them, this rule is deleted.

We arrive at a complete list of applicable computation rules.

2.4 Setting up a sequence of computation rules

Let us introduce the term *start quantities* for all known point coordinates and known or observed geometric quantities. First, we search for a computation rule, which only uses start quantities as input quantities and apply this rule generating a new known quantity, expanding the set of known quantities by one. Now, we repeat this step by sequential application of computation rules to known quantities, until no more known quantities can be generated. However, we do not yet actually compute the values of the quantities, but only set up a sequence of rules.

Example 2: Consider the planar trilateration network in figure 2, where A and B are known points, 1,2,3 are unknown points and only the lengths of the drawn lines have been observed. First, we find a rule for computing the area A_{A12} of triangle A12 from side lengths (Heron's formula)

$$A_{A12} = \frac{1}{4}\sqrt{(d_{A1} + d_{12} + d_{A2})(-d_{A1} + d_{12} + d_{A2})(d_{A1} - d_{12} + d_{A2})(d_{A1} + d_{12} - d_{A2})} \tag{2}$$

Next, we find the arcs-intersection rule for computing 1 from A and B, then we find the arcs-intersection rule for computing 2 from 1 and A. Then, we find a rule for computing the area of triangle A12 from coordinates of vertices (Gaussian formula).

Accepted Manuscript of Survey Review, Volume 50, Issue 361, Pages 364-372 (2018)
https://dx.doi.org/10.1080/00396265.2017.1279844

$$A_{A12} = \frac{1}{2}\left(Y_A(X_1 - X_2) + Y_2(X_A - X_1) + Y_1(X_2 - X_A)\right) \quad (3)$$

Of course, the numerical values of (2) and (3) would be identical, but remember that at this step we do not yet compute values. Finally, we find the arcs-intersection rule for computing 3 from 2 and B.

In case of redundant start quantities, almost any quantity can be computed by many different computation schemes and consequently will later assume many different values. Here two computation schemes for one quantity are only considered as different if for the two sets of used start quantities it holds that one set is not a subset of the other. In this case we create multiple instances of this quantity, otherwise the computation rule using the larger, hence reducible, set of start quantities is dropped.

Example 2 (cont'd): The area of the triangle A12 has been computed twice by (2) and by (3). While the first only uses the lengths of the sides as start values, the latter additionally uses the coordinates of A. Thus, (3) is reducible and dropped.

Example 3: Reconsider the planar trilateration network in figure 2, but now also 3 is a known point. Point 2 can be computed in three different ways, see table 1. The three sets of used start quantities used for point 2 overlap, but none is a subset of the other set, see table 1. Therefore, all three instances for 2 are retained. Now, the area of A12 could be computed by (2) and by (3) using the two different instances of 2 involving point 3. In this way we arrive at four different irreducible instances of the area of A12, see table 2.

In case of redundant start values, even for start quantities we often obtain new instances by computing them from other start quantities. For each quantity it is tried to create as many different irreducible instances as possible.

Example 3 (cont'd): All three instances of point 2 can be used to compute d_{23} by (1) applied to points 2 and 3. But only the first possibility gives a valid new instance for d_{23}. Also for all other observations in this network, multiple instances are created. The same happens with the coordinates of the known points, angles, areas etc.

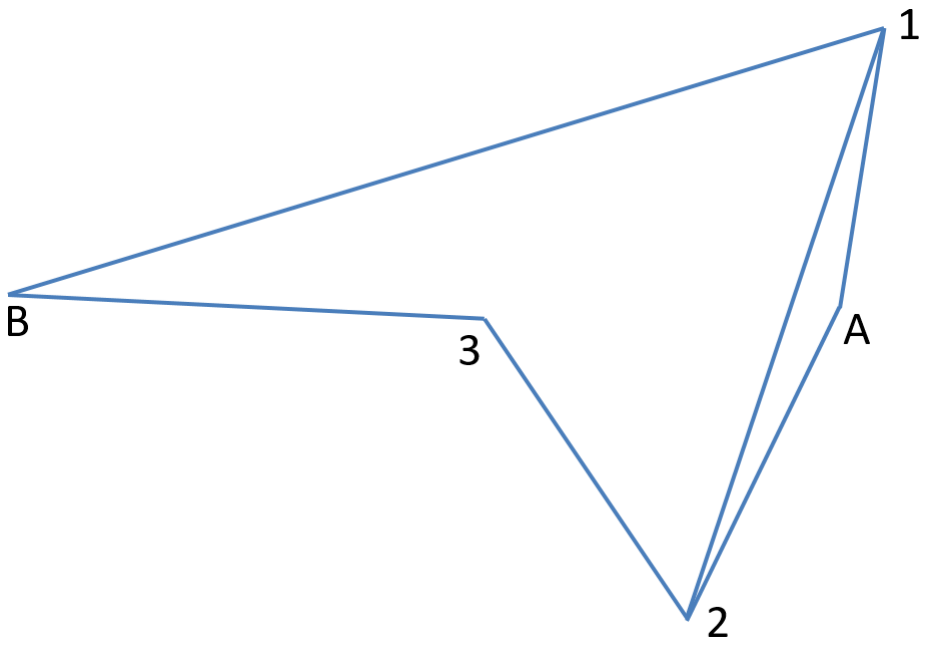


Figure 2: Planar trilateration network.

2.5 Computing values

The computation rules are now applied to the actual numerical values. If for some computation rule there is a non-unique solution, as e.g. for the arcs-intersection, it is tried to resolve the non-uniqueness by comparison with other

Accepted Manuscript of Survey Review, Volume 50, Issue 361, Pages 364-372 (2018)
https://dx.doi.org/10.1080/00396265.2017.1279844

instances of the same quantity or with the start value, if any. The solution not matching other results is dropped. If this fails permanently, then the computation scheme is duplicated and continued with both solutions in parallel threads. If this case arises repeatedly, then a large number of threads may arise. Those are all computed in parallel. Also the case can arise, that in some thread a computation rule produces no solution, e.g. an arcs-intersection, when the circles do not intersect. Then this thread is terminated and dropped.

In case of a gross error in the start values or some other mistake, we could be lucky that each thread terminates prematurely. This would show a problem with the start values.

Example 2 (cont'd): When computing point 1 in the planar trilateration network by arcs-intersection, two solutions arise. Since there is no other instance of point 1, it is not possible to decide, which solution is the true one. Thus, the computation scheme is duplicated and continued in parallel with both threads. When computing point 2 by arcs-intersection, we could be lucky that in the wrong thread the arcs do not intersect. Then this thread is terminated and deleted, otherwise we continue with four threads. The same happens when computing point 3 by arcs-intersection. Finally we could obtain up to 8 solutions for each quantity, depending on the numerical start values. In the actual network configuration displayed in figure 2, we obtain four solutions: Besides the solution in figure 2 we also obtain the three solutions displayed in figure 3.

Example 3 (cont'd): Remember that if point 3 is also a known point, then in each thread we can compute three instances of point 2, all by arcs-intersections. In the thread matching reality, all three instances must have nearly identical coordinates. Possible deviations are caused by small observation errors in the start values. Consider the two solutions of 2 not using point 3. These are the solutions displayed in figures 2 and 3. Note that the apparently four solutions of 2 are pairwise coincident: 2 in figure 2 matches 2 in figure 3b and 2 in figure 3a matches 2 in figure 3c. Denote these solutions as 2a and 2b. Now in each thread we have more instances of 2. E.g. from points A and 3 we get by arcs-intersection two more instances of 2 denoted as 2α and 2β. In figure 4 we see, that only 2b and 2β are coincident. The other threads can be terminated and dropped and we come up with a unique solution. Nonetheless, point 2 has two instances 2b and 2β, and it gets one more by arcs-intersections from 1 and 3.

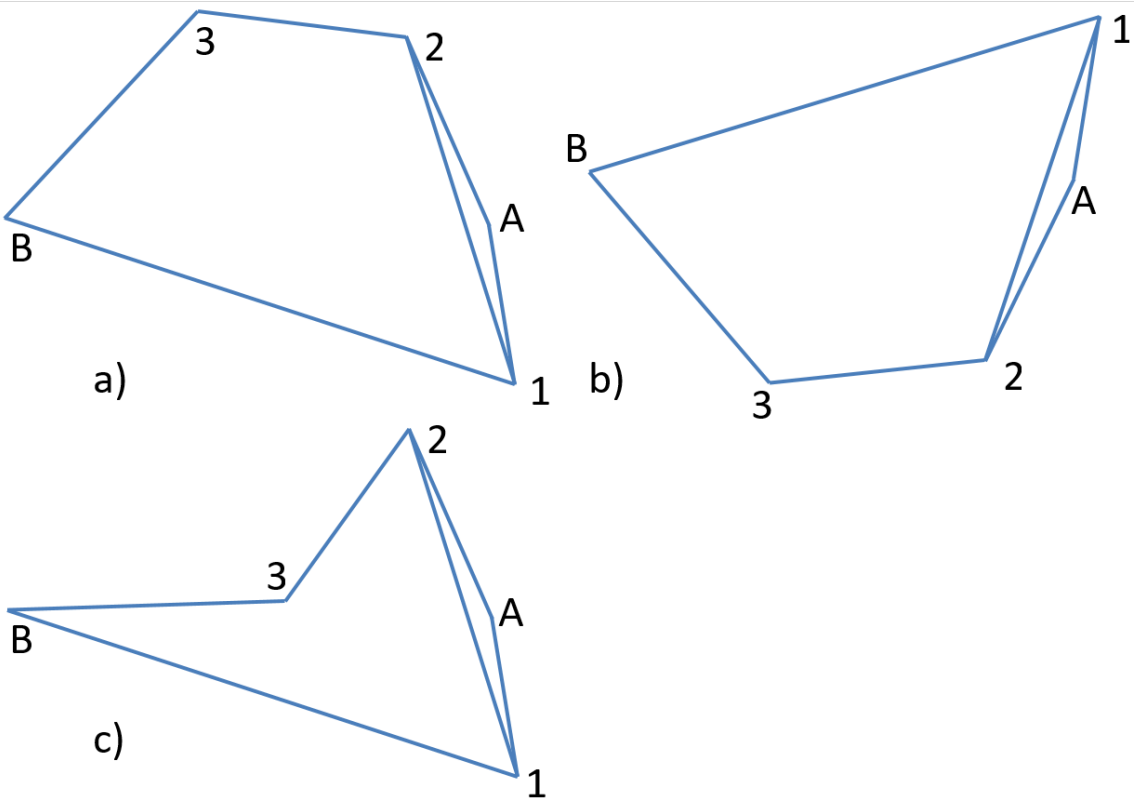


Figure 3: Three extra solutions of the planar trilateration network of figure 1 with known points A,B.

Accepted Manuscript of Survey Review, Volume 50, Issue 361, Pages 364-372 (2018)
https://dx.doi.org/10.1080/00396265.2017.1279844

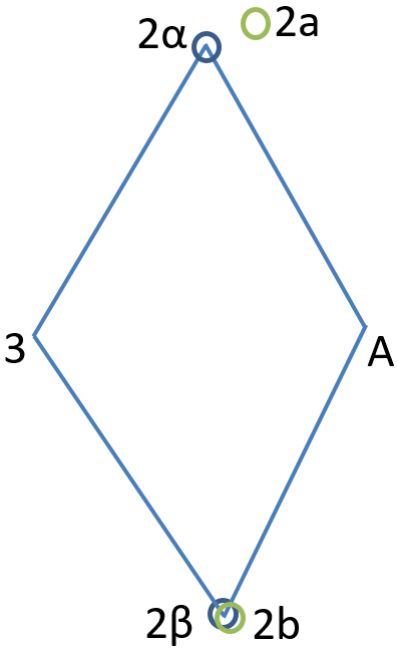


Figure 4: Arriving at a unique solution in example 3: Instances 2a and 2α are not nearly coincident and are dropped.

2.6 Assigning final values to all computed quantities

If multiple instances of a quantity are computed, they will show a certain scattering for two reasons:

- The start values contain small observation errors. Usually this yields only a small scattering of the computed values. However, it may occur that a computation shows conditions of bad error propagation, usually if an acute triangle is involved or an intersection has an acute intersection angle. We have good chances that the instances suffering from bad error propagation give very large or very small values, compared to other instances of the same quantity.
- Some start values may be falsified by gross errors. This would influence only those instances, which are computed from these start values. Again, we have good chances that these values are very large or very small.

We can get rid of the effect of bad error propagation as well as of a few gross errors by taking the median of the computed values of the same quantity and assign it as a final value of that quantity. Therefore, the solution shows features of robustness.

Example 3 (cont'd): We have obtained three instances of point 2, see again table 1. According to figure 2, the instance computed by arcs-intersection from 1 and A has an acute intersection angle. It is likely that the coordinates of this instance are the smallest or largest of the three values. Taking the median of the coordinate values means to automatically ignore the bad intersection and to use one of the two good intersections as the final result.

It is obvious that the power of this approach only comes into effect, if the number of instances is really large, say 5 or larger. There is some opportunity for improvement of this approach:

- Inspecting the range (i.e. maximum - minimum) of the instances does not immediately show gross errors in the start values because large range values could also be the consequence of bad error propagation. Therefore, it is beneficial to keep track of bad intersections and acute triangles during the computation procedure. This helps to sort out extreme values suffering from bad error propagation before taking medians and ranges, but it requires the number of instances to be even larger.

Accepted Manuscript of Survey Review, Volume 50, Issue 361, Pages 364-372 (2018)
https://dx.doi.org/10.1080/00396265.2017.1279844

- Taking the median may leave some discrepancies in the final values. E.g. the median coordinates of P and Q and the median value of d_{PQ} do not always exactly satisfy (1). Measures against these discrepancies are not yet proposed.

Note that it is not necessary to specify, which quantities are actually desired by the user. The computation procedure generates all quantities, which are computable from the start quantities. Later the user selects the desired quantities from the output. If some desired quantity is not found in the output, it is not computable from the input.

2.7 Outlier detection

Outlier detection means identifying grossly falsified coordinates or observations. In geodesy, the common approaches use mathematical statistics and require a good stochastic model of the coordinates and observations. *Lehmann and Lösler (2016)* compare the widespread methods of hypothesis testing with the elegant and plausible method of model selection by information criteria. The latter seems to have some interesting advantages.

Here we do not use an explicite stochastic model, which rules mathematical statistics out. However, the proposed algorithm allows a very natural extension towards outlier detection, which will be outlined below. As any method of outlier detection, it requires redundant start values and the results improve greatly, if the redundancy is increased.

If a start value is grossly falsified, then almost all instances of computed quantities, which use this start value, are falsified as well. They will typically become the largest or the smallest of all instance values of a quantity and therefore increase its range. This is expected to happen for many computed quantities. We have to find out, which start value all those outlying instance values share. This is the candidate for an outlier.

Practically, we propose to run through all start values, drop all instances using this value and find out, how many ranges decrease, and how much. Many drastic decreases of ranges is an unmistakable sign of an outlier. Finally eliminating an outlying start value does not require to repeat the entire procedure, but only to ultimately drop all instances using this value. To detect multiple outliers, this procedure can be repeated many times.

This approach is remotely related to a cross validation technique (*Geisser 1993*) called 'leave-one-out' cross validation, LOO or LOOCV for short (*Cawley and Talbot 2003*), later extended to 'leave-one-block-out', LOBO for short (*Biagi and Caldera 2013*). Other relationships exist to Baselga's "Exhaustive Search Procedure" *(Baselga 2011)* and to "Case-Deletion Diagnostics" *(Guo 2013)*. All those approaches are of combinatorial type: The aim is to find a clean subset of observations. However, in contrast to our contribution, all those approaches use a stochastic model of the observations. Therefore, we do not go into the details here.

Example 3 (cont'd): This example has a very poor redundancy, such that the prospects for detecting outliers are dismal. Using the configuration in figure 2 we simulate a gross error by increasing d_{A1} by 10% of its true value. All computed quantities with multiple instances have considerable ranges, clearly indicating gross errors. Inspecting all largest and smallest instances we find that only d_{A1} and point A are used in all those instances. This shows that either d_{A1} or the coordinates of A are grossly falsified. Which one is undecidable.

3 Illustrating example: Determination of an inaccessible point with auxiliary triangles

3.1 Setting up the example

An inaccessible point 1121 was observed from three tacheometry stations 515, 632, P, see figure 5. The height of 1121 is desired. The benchmark height of point 515 amounts to 107.483 m, the instrument height is here 1.54 m. The benchmark height of point 632 amounts to 107.832 m, the instrument height is here 1.39 m.

Accepted Manuscript of Survey Review, Volume 50, Issue 361, Pages 364-372 (2018)
https://dx.doi.org/10.1080/00396265.2017.1279844

The height of the auxiliary point P is unknown. An arbitrary value can be assigned to the instrument height of P, say 0.00 m. The target heights equal the instrument heights on the same point, because matching instruments and reflectors have been used. The inaccessible point 1121 was directly sighted without distance observation, such that the corresponding target height equals 0.00 m. The observation values are given in table 3.

Now we will apply the algorithm worked out in the previous section to this problem. Before we can begin, we need to introduce a coordinate frame. For this purpose we manually compute the horizontal distance $d_{515,632} = 952.233$ m by (1) and define a local frame basis as follows:

$$X_{515} = 1000 \text{ m}, Y_{515} = 1000 \text{ m}, Z_{515} = 107.483 \text{ m},$$

$$X_{632} = 1000 \text{ m}, Y_{632} = 1952.233 \text{ m}, Z_{632} = 107.832 \text{ m}$$

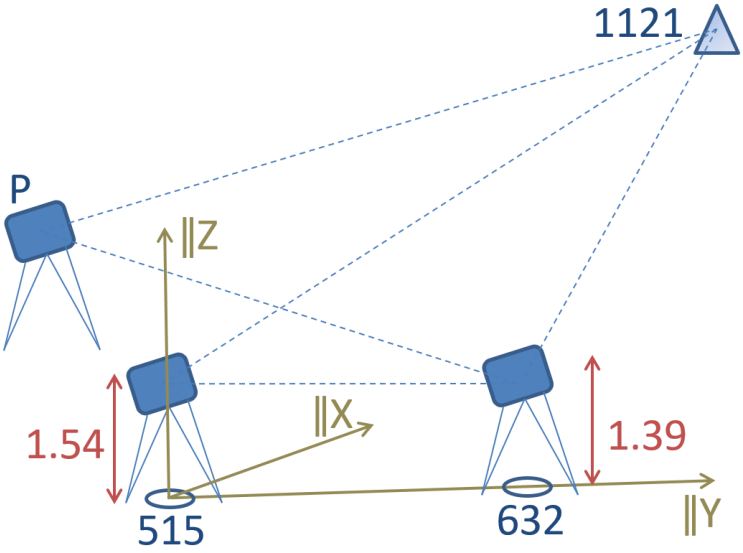


Figure 5: Determination of an inaccessible point 1121 with auxiliary triangles.

3.2 Computation rules

Summarizing, we have 32 start quantities. We always count X and Y of a point as one quantity only, because each computation rule involving X, also involves Y of the same point and vice versa, e.g. (1). Target heights count from each station as a different quantity, such that we get seven target heights.

Running the algorithm described in section 2, we come up with 47 computable quantities (here of course not counting areas etc.). They are listed in table 4. Together with some incomputable start quantities, these quantities are related by 178 applicable computation rules. Setting up a sequence of computation rules, 98 of these rules turn out to be useful. The total sequence has 2590 steps, which will give in total 2590 different values (instances) of the computable quantities.

Table 4 shows the number of instances to be obtained for each relevant computable quantity. Remember that all these instances are computed from different irreducible sets of start values. The maximum is 207 instances for the slope distance between P and 1121. The actually desired quantity Z_{1121} has 84 instances. The number of start values used for these 84 instances varies between 10 and 20. Three start values are used for all those instances: XY_{515}, XY_{632} and the horizontal angle from station 632 to target 515. All other start values could have been missing without losing the possibility to compute Z_{1121} at least once. (One could expect the target height of 1121 being indispensable for Z_{1121}, but remember that in our implementation they count as three quantities, one from each station.) Two start values are used for none of those instances: the slope distances from station 632 to target 515 and back. This is because they are already materialized in Y_{515}, Y_{632}.

Accepted Manuscript of Survey Review, Volume 50, Issue 361, Pages 364-372 (2018)
https://dx.doi.org/10.1080/00396265.2017.1279844

We hope that among the 84 instances of Z_{1121} there are two not sharing a start quantity except the three indispensable quantities named above, such that a gross error in any other start quantity shows up in their difference. Unfortuntely this is not exactly so. But we find several pairs of instances sharing only one further start quantities. This gives a reasonable protection against gross errors by only computing and comparing two instances. More can be expected in cases of better redundancy.

3.3. Performing the computation

Remember that up to now the numerical values of the computable quantities have not yet been computed. But now the start values are really used to compute them. Although the 2590 computation steps involve a number of non-unique steps like arcs-intersections, all non-uniquenesses can be resolved using other instances of the computed quantities and we finally come up with a unique solution.

The curvature of the Earth is corrected during the computation. The resulting values of the 84 instances of Z_{1121} are displayed in figure 6. The statistics are

Minimum: 201.0881 m Median: 201.2115 m Maximum: 201.2585 m Range: 0.1704 m

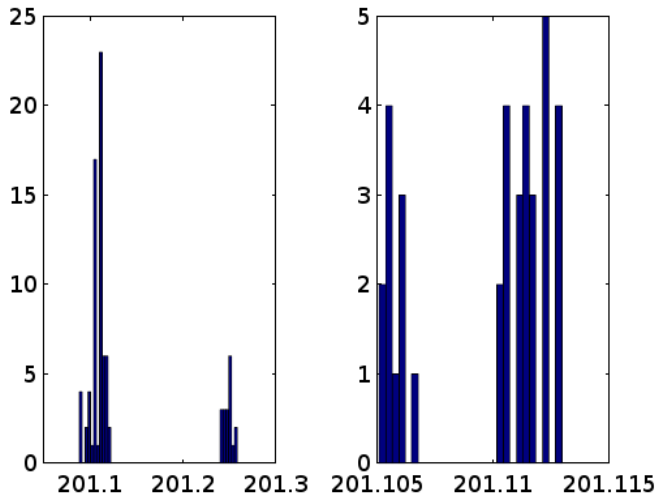


Figure 6: Histogram of the instances of the height of point 1121. left: all 84 instances, right: only 41 instances not suffering from bad error propagation (note the different scales).

The range is quite large. However, 43 instances of Z_{1121} suffer from bad error propagation due to bad intersections and acute triangles. E.g. triangle 551-1121-P has an angle of only 0.4 gon at 1121. If we keep track of such badnesses during the computation procedure as proposed in subsection 2.6, we can improve the solution considerably. The remaining 41 instances show the following statistics:

Minimum: 201.1045 m Median: 201.1106 m Maximum: 201.1129 m Range: 0. 0084 m

Thus, the range is reduced by 95%. Nonetheless, there are still two clusters of values (see again figure 6), the lower cluster contains 16 instances and the upper cluster contains 25 instances.

3.4 Robustness

If there are some gross errors in the start values, we can hope that they do not affect the medians too much. In this investigation we simulate such gross errors by falsifying some start values.

Accepted Manuscript of Survey Review, Volume 50, Issue 361, Pages 364-372 (2018)
https://dx.doi.org/10.1080/00396265.2017.1279844

As an example we increase each horizontal angle by 1 gon, one at a time. Note that such a gross error usually falsifies Z_{1121}, considerably. E.g. a change of a horizontal angle in the triangle 515-632-1121 by 1 gon can change an instance of Z_{1121} by more than 2 metres.

We will investigate, how robust the algorithm behaves. Table 5 shows the medians and ranges after changing one out of seven horizontal angles by 1 gon. The values refer to the solution excluding the 43 instances of bad error propagation. The new medians are quite close to the non-falsified value (201.1106 m). An exception is clearly the angle from 632 to 1121. This value is needed in almost all instances of Z_{1121}, such that here a gross error cannot be tolerated. In any case, the new ranges of more than 1 metres clearly indicate that a gross error occurred.

Finally, we falsify (increase) two horizontal angles by 1 gon: from 515 and P to 1121. The new median of Z_{1121} is 201.2904 m. As a result, it differs from the old value by 0.1798 m. This is certainly not bad for two gross errors at a time in a not too redundant set of observations. The new range of Z_{1121} is 2.14 m.

3.5 Outlier detection

We continue the investigation of the last case with two falsified horizontal angles. Table 6 shows the ratio of quantities (total = 49) and the ratio of the corresponding instance values (total = 2590) computable after one suspected outlier has been dropped. E.g., after dropping the horizontal angle from P to 1121, four observed horizontal angles cannot be double-checked by other observations anymore, such that 92% of the quantities are still computable. Z_{1121} can still be computed, but only 18 out of 84 values are left. Summing up all quantities, we find that 38% of the values are still computable.

More important is the number of quantities, where after dropping a start value the range is reduced. Dropping the (previously falsified) horizontal angle from P to 1121 reduces 68% of the ranges. The same for the other angle at station P because both always go together. This is the maximum ratio among all start values and indicates that one of these start values is grossly falsified.

Unfortunately, the other falsified start value is not clearly detected in table 6. This is due to insufficient redundancy of this example.

4 Conclusions

The presented computation procedure for the processing of geometric observations in geodesy has the following advantages:

1. It is simple to use, because once the catalogue of template computation rules has been implemented, only the start values must be specified.
2. It is universal, because all problems involving geometric observations can be processed: area segmentations, geodetic intersections, traverses, networks, 2D or 3D, redundant or non-redundant, unique or non-unique.
3. It is automatic, because the user is not supposed to control the algorithm.
4. It is robust, because if the redundancy is sufficiently high, a small number of gross errors in the start values can be tolerated. Moreover, there is a good chance to detect those grossly falsified values (outliers).
5. It is implemented on a webserver and can be used freely (see appendix)

Nonetheless, there are also four disadvantages:

1. We do not obtain precision measures like standard deviations for the computed quantities. This is not possible without a stochastic model for the observations. In the case that such a model is available and we know that it fits reality well, classical geodetic adjustment must of course be applied in one way or another.
2. The algorithm is of combinatorial type: It is tried to combine computation rules to get new results. This search is very smart, but for large-scale problems it may require some time to really find all possible

Accepted Manuscript of Survey Review, Volume 50, Issue 361, Pages 364-372 (2018)
https://dx.doi.org/10.1080/00396265.2017.1279844

computation schemes. However, to guarantee quality and robustness of the results, one is often already satisfied with a sufficiently large number of them. Then, the search can be terminated prematurely at any time.

3. If in the case of redundant observations a stochastic model is available and if we are lucky that it really fits the stochastic properties of the observations, e.g. normal distribution, independent observations, no gross errors, etc., then the classical geodetic adjustment by least squares (*e.g. Teunissen 2007*) of course yields better results. However, the results of the presented computation procedure can be used here as automatically generated approximate values of an adjustment procedure.
4. As pointed out at the end of subsection 2.6, taking the median may leave some discrepancies in the final values.

The latter point will be tackled in the future.

References

Baselga S (2011) Exhaustive Search Procedure for Multiple Outlier Detection. Acta Geod. Geophys. 46(4) 401-416. DOI: 10.1556/AGeod.46.2011.4.3

Benning W (1978) Zur Auswertung geodätischer Messungen mit automatisierter Fehlersuche. Allgemeine Vermessungsnachrichten 1/1978, S. 16-26

Benning W, Ahrens B (1979) Konzept und Realisierung eines Systems zur automatischen Fehlerlokalisierung und automatischen Berechnung von Näherungskoordinaten. Nachrichten aus dem öffentlichen Vermessungsdienst Nordrhein-Westfalen, Heft 2/1979 S. 107-124.

Benning W, Förstner W (1979) Datenbereinigung und automatische Berechnung von Näherungskoordinaten in geodätischen Lagenetzen – das Programm NAEKO. Zeitschrift für Vermessungwesen 2/1979, S. 52-60.

Biagi L, Caldera S (2013) An Efficient Leave One Block Out approach to identify outliers. J. Appl. Geodesy, 7(1) 11–19. DOI 10.1515/jag-2012-0030

Cawley GC, Talbot NLC (2003) Efficient leave-one-out cross-validation of kernel Fisher discriminant classifiers, Pattern Recognition 36

Fontijne LDD, Mann St (2007) Geometric algebra for computer science : an object-oriented approach to geometry. Morgan Kaufmann, San Francisco, CA. ISBN 0123694655

Geisser S (1993) Predictive Inference. Chapman and Hall, New York. ISBN 0-412-03471-9

Guo J (2013) The case-deletion and mean-shift outlier models: equivalence and beyond. Acta Geod. Geophys. 48(2) 191-197. DOI 10.1007/s40328-013-0017-5

Huber PJ (2009) Robust Statistics (2nd ed.). Hoboken, NJ: John Wiley & Sons Inc., ISBN 978-0-470-12990-6

Lehmann R (2015) Ein automatisches Verfahren für geodätische Berechnungen. Allgemeine Vermessungsnachrichten 122 (2015) 3, 102-115

Lehmann R Lösler M (2016) Multiple Outlier Detection: Hypothesis Tests Versus Model Selection by Information Criteria. J Surv Eng 142(4). DOI 10.1061/(ASCE)SU.1943-5428.0000189

Rousseeuw PJ, Leroy AM (1987) Robust Regression and Outlier Detection, John Wiley & Sons, New Jersey. ISBN 978-0-471-85233-9

Teunissen PJG (2006) Testing theory. Series on Mathematical Geodesy and Positioning, VSSD Delft, ISBN 978-90-407-1975-2

Teunissen PJG (2007) Adjustment theory – an introduction. Series on Mathematical Geodesy and Positioning, VSSD Delft, ISBN 978-90-407-1974-5

Vetter M (1992) Automatische Berechnung zweidimensionaler Näherungskoordinaten. Allgemeine Vermessungsnachrichten, 99 (1992) 6, 245 – 255

Vetter M (2007) Näherungskoordinatenberechnung und robuste Fehlersuche. In: Derenbach H, Illner M, Schmidt G, Vetter M, Vielsack S (Eds.) Ausgleichungsrechnung – Theorie und aktuelle Anwendungen aus der Vermessungspraxis. Schriftenreihe des Studiengangs Geodäsie und Geoinformatik (2007), 4. Universitätsverlag Karlsruhe. ISBN 978-3-86644-124-8

Appendix: Aspects of technical implementation on the IN DUBIO PRO GEO geodetic cloud computation webserver

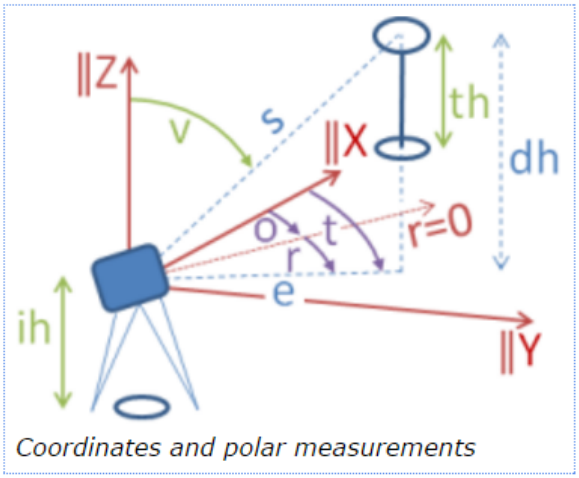
The algorithm is implemented on the IN DUBIO PRO GEO geodetic cloud computation webserver (www.in-dubio-pro-geo.de) in the computation tool named 'universal computer' and can be used freely. (Click on the British flag for the English language). In this implementation all quantities from the list in subsection 2.1 are realized, except areas. The total number of start values is limited. The limit depends on the structure of the problem. In the worst case the maximum number of start values is 126, but usually it is higher. The total computation time of a single run is limited to 40 s and can be limited even more by the user. To save time, the re-computation of some or all start values can also be suppressed. The complete procedure for the example in section 3 takes 3 s. The computation procedure can be analysed in detail because not only the values of the instances are displayed, but also the entire detailed sequence of steps with intermediate results. Figure 7 shows the output for the computation of point 2 in example 3.

Symbols

o	angle of orientation	r	horizontal angle	e	horizontal distance
ih	instrument height	t	azimuth	s	slope distance
th	target height	v	zenith angle	dh	height difference
AA	arcs-intersection	RE	resection	LL	straightlines-intersection
LA	straightline-arc-inters.	Rec2Pol/Pol2Rec		coordinate conversion cartesian⇔polar	

8 start values used for XY(2)

XY(3) = 127.67;157.18
XY(A) = 129.15;192.92
XY(B) = 130.16;107.49
e(1→2) = 63.623
e(1→A) = 28.54
e(1→B) = 94.147
e(2→3) = 36.784
e(2→A) = 35.714



Coordinates and polar measurements

Computation procedure and intermediate values for XY(2)

$XY(1)_1$ = AA(XY(A),e(1→A),XY(B),e(1→B)) = 157.29677845208;197.64127766284

Results (sorted in ascending order) for XY(2)

$XY(2)_1$ = AA(XY(3),e(2→3),XY(A),e(2→A)) = 96.966520006816;177.43731898125
$XY(2)_2$ = AA($XY(1)_1$,e(1→2),XY(3),e(2→3)) = 96.966816325464;177.4377680964
$XY(2)_3$ = AA($XY(1)_1$,e(1→2),XY(A),e(2→A)) = 96.967989996649;177.43426371966

values	minimum	median	maximum	range	inter quartile range (IQR)
	96.9665200	96.966816325	96.9679900	0.0015	
3	177.434264	177.43731898	177.437768	0.0035	

Figure 7: Sequence of steps for computation of point 2 in example 3 with IN DUBIO PRO GEO universal computer

Accepted Manuscript of Survey Review, Volume 50, Issue 361, Pages 364-372 (2018)
https://dx.doi.org/10.1080/00396265.2017.1279844