THE "DANISH METHOD" OF WEIGHT REDUCTION FOR GROSS ERROR DETECTION

Jens Juhl
Aalborg University
Denmark

ISP Commission III

Abstract

During the last decade sophisticated statistical methods have been applied to find gross errors in adjustments.

In 1977 the author introduced a weight reduction function in a bundle adjustment programme. This function had been used successfully in geodetic computations by the Danish geodesist T. Krarup. With slight modifications it showed to be a very efficient and cheap method for gross error detection, especially in well-conditioned blocks. The method is simple to program, even in existing programmes.

The paper deals with the practical implementation and the results, but gives no theoretical explanation.

After good results in the OEEPE/ISP test on gross error detection, the method seems to find followers.

Introduction

Since the 22th ISPRS congress in Hamburg in 1980 the Danish Method for gross error detection in photogrammetric adjustments has been discussed energetically. Good results obtained with this method combined with very easy programming - even in existing programmes - have caused the Danish Method to be implemented in several programmes in the world. In this paper I will not describe the theory of the method, simply because I do not have any. But I will describe how the method is implemented in my bundle adjustment programme, sANA, and how it works in practice.

The Idea

The idea of only using residuals combined with an iteration process in an automatic procedure for gross error elimination has been used by the Geodetic Institute in Denmark for more than 15 years. The procedure was proposed by the geodesist T. Krarup around 1968. It has never been described in any official paper. Roughly speaking, the gross error detection uses the residuals or more correctly a function of the residuals (r) and σ_0 from the previous iteration as a weight for the actual iteration.

The residuals from the previous iterations are computed in the actual iteration as the "right hand side" used for computation of the normal equations. There are thus no storage nor computation problems. The value σ_0 can be computed in the previous iteration step by a summation of the right hand side, but the value σ_0 is thus "two iterations old". By adding one line to the normal equations equal to the right hand side, it is possible to compute σ_0 only "one iteration old". In practice both values of σ_0 works, but the "two step old" value sometimes costs an extra iteration. The function used for computing the weights is organized in such a way that a great residual gets a small weight and vice versa.

In 1977 I became aware of the method during my Ph.D. study, and it caught my interest. My idea was to use it in photogrammetry rather than in geodetic networks. At that time the method had only been used by the Geodetic Institute for geodetic adjustments.

Further Development of the Method

As there was no theory for the method and no written material, I did a lot of experiments to make the procedure work in photogrammetry, but good results were only obtained after some changes in the original procedure.

The problem was that a gross error did not necessarily show among the greatest residuals. For that reason the iteration procedure was often unable to "catch" the gross errors. After many tests I came to a weight procedure working in three steps:

1) In the first step I am making a normal bundle adjustment after the method of least squares. Usually, 2-3 iterations are used.

2) In the second step a weight function is introduced which is rather drastic, saying that quite a lot (10-25%) of all observations have gross errors. This is done saying that even smaller residuals relative to unit weight are erroneous. Now there is a greater chance to find the gross errors between the 10-25% of the greatest residuals.
Due to the lower weight the residuals will increase in size, and in the next iteration the process will accelerate. But this will only happen to the observations with gross errors. The major part of the weightreduced observations will only increase slightly in spite of their weight being smaller than one.
Usually, 2-3 iterations are used. Two iterations are used if σ_0 decreases with more than 20% during these iterations. If σ_0 decreases with more than 20%, an extra iteration is used.

3) In the third step a new weight function is introduced which is a soft function compared to 2). Only $\frac{1}{2}$-2% of all observations are usually weightreduced with this function.
In this step all observations, which were not erroneous and therefore in step 2) had only a minor increase in the residuals, are brought back in the usual least square adjustment (weights equal to one or near one).

The observations with gross errors will stay weight-reduced due to their greater residuals obtained in step 2).
As stop criterium for the iteration I use a maximum correction of the ground coordinates less than 0.01°/oo of the flying height. Usually, 3-5 iterations are necessary.

The Weight Function

The weight function used in step 2) and step 3) is:

(1) $$p = \exp(-0.05(\frac{k \cdot \sqrt{ap} \cdot r}{\sigma_o})^a)$$

where

p: The computed weight reduction ($0<p<1$). The value is computed for each observation.

k: The value k is equal to 1.0 for step 2) and equal to 0.6 for step 3).

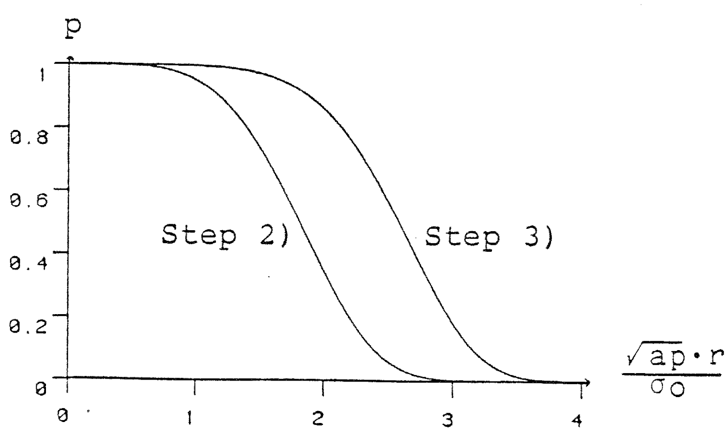
ap: This is the apriori weight of the observation. For image observations in bundle adjustments the value usually is set to one.

r: The "residual" of the observation in the actual iteration. For point observations in the image $r = \sqrt{\frac{1}{2}(r_x'^2 + r_y'^2)}$. For coordinate observations (control points) $r = \sqrt{\frac{1}{2}(r_x^2 + r_y^2)}$. X and y coordinate observations are in the same way as image observations given the same weight. For z coordinate observations $r = |r_z|$.

σ_o: RMSE of unit weight computed from the previous iteration. If p for this iteration is less than 0.1, the residuals do not count in the computation.

a: The value of a is equal to 4.4 for step 2) and equal to 6.0 for step 3).

A graph of the two functions used in step 2) and step 3) is shown below:



Modification of the Weight Function

To save computing time the value p is set to one, if the value $\frac{\sqrt{ap}\cdot r}{\sigma_o}$ is less than 1.0.

To prevent overflow during computation of the exponential function the weight function (1) is replaced by a polynomial, if $\frac{\sqrt{ap}\cdot r}{\sigma_o}$ is greater than 3.2 for step 2) and greater than 6.0 for step 3). The polynomial is

$$(2) \qquad p = \frac{0.0225}{\left(\frac{\sqrt{ap}\cdot r}{\sigma_o}\right)^4}$$

To make it easier to catch a gross error, the residuals (r) are multiplied with a factor. This factor is usually set to one, but if a point is only observed in two photos, the factor is set to 1.6 for image observations. If the point is observed in three photos, the factor is set to 1.3. This is done only in step 2).

Gross Errors Only

The weight function is only used for gross error detection. Systematic errors are in the programme sANA treated with added parameters.

Extra Facilities

As an extra facility in sANA it is possible to use a weight function simulating e.g. an adjustment with the norm 1 ($\min\Sigma|\sqrt{ap}\cdot r|$) instead of norm 2 (least squares). The weight function is:

$$(3) \qquad p = |\sqrt{ap}\cdot r|^{(b-2)}$$

where b: Norm of adjustment.

To prevent overflow $p = 10^{-(\text{number of iterations})}$, if $|\sqrt{ap}\cdot r| < 10^{-(\text{number of iterations})}$.

Different tests, not mentioned in this paper, have shown that a norm between 0.8 and 1.2 gives the best result concerning gross error. But the weight function (1) is a far better function for gross error detection.

Easy Programming

As mentioned in the introduction the programming effort is very small even in existing programmes: For each iteration in step 2) and 3) the weight p is computed for each observation. As residual for the observation the right side of the equation is used. σ_o is computed from the previous step. The method does not need any storage requirements, any inversion of the normal equations or similar. Only a few extra iterations and some extra programme lines are necessary.

No Theoretical Foundation

The iterative weight reduction method is not built on any theoretical foundation, but is only based on practical experiences gained from many bundle adjustments. In other words: The whole adjustment procedure is based on a theoretical knowledge (mean square) combined with experiences gained by making bundle adjustments through several years.

Conclusions

The weight function has now - without any changes - been in use for more than five years with good results in well-conditioned Danish large-scale blocks. We also use it in other photogrammetric computations like relative and absolute orientations. The method is less tested in triangulations with sparse control. We are still uncertain about small gross errors, but look forward to the results of the OEEPE/ISPRS test.