Photogrammetria (PRS), 43 (1989) 337-346
Elsevier Science Publishers B.V., Amsterdam — Printed in The Netherlands

A fast recursive method for repeated computation of reliability matrix *QvvP*

SHAN JIE

Department of Photogrammetry Zhengzhou Surveying and Mapping Institute, Longhai West Road, Zhengzhou, Henan, P.O. Box 450052, P.R. China

(Received 05 May 1988; revised and accepted 26 October 1988)

ABSTRACT

Shan Jie, 1989. A fast recursive method for repeated computation of reliability matrix *QvvP*. Photogrammetria, 43: 337-346.

In quality analysis, optimization design and blunder detection of many photogrammetric problems, repeated computation of reliability matrix *QvvP* is often required. In this paper, a new recursive method for repeated computation of *QvvP* is developed. From a test on a bundle block adjustment, it is shown that this new method possesses much higher computational efficiency than the conventional methods.

1. INTRODUCTION

In order to discover and delete gross errors in a photogrammetric block adjustment, the reliability matrix *QvvP* sometimes has to be calculated repeatedly with the change of the weight matrix *P* of the observations each time (El Hakim, 1982; Li Deren, 1983). In other areas, like quality analysis, optimization design, and a posteriori variance-covariance components estimation, such kind of repeated computation of *QvvP* is needed too (Förstner, 1979; Li Deren, 1983).

In the conventional methods of dealing with such kind of computations, very large mormal equations have to be inverted repeatedly, which is a very time-consuming procedure on computers (Förstner, 1979; El Hakim, 1982; Li Deren, 1983). In this paper a much more efficient algorithm is introduced for this purpose. The problem is to find an efficient algorithm to recompute the matrix *QvvP* from a proceeding one, after the weight matrix *P* has been altered.

The derivation here starts from the conventional formulas expressing the general relations between matrices *QvvP* and *P*. But these conventional formulas have been changed in form through differentiation and then integration, so that they can better serve the present purpose.

0031-8663/89/$03.50

2. THE GENERAL RELATIONS

Assume we have an adjustment model:

$$\boldsymbol{V}=A\boldsymbol{X}-\boldsymbol{L} \tag{1}$$

where

$\boldsymbol{L}$ = vector of observations,
$\boldsymbol{X}$ = vector of unknowns,
$\boldsymbol{V}$ = vector of residuals,
A = design matrix and
P = weight matrix of observations.

By the method of least squares, we obtain (Förstner, 1983):

$$\boldsymbol{V}=-(E-AN^{-1}A^{T}P)\boldsymbol{L}=-R\boldsymbol{L} \tag{2}$$

where

$$\begin{aligned} N&=A^{T}PA \\ R&=(E-AN^{-1}A^{T}P)=QvvP \end{aligned} \tag{3}$$

Equation (3) represents the general relation between $R=QvvP$ and P. Matrix R may be called a reliability matrix since its diagonal elements represent the redundancies of observations and hence reflect directly the effectiveness of gross error detection.

When the weight matrix is diagonal (which is almost always the case in practice), i.e.:

$$P=\text{diag}(P_{11}, P_{22}, \dots\dots P_{nn})$$

and the matrix A is blocked as:

$$A^{T}=(a_1, a_2, \dots\dots a_n)$$

we have:

$$(R)_{ij}=r_{ij}=\delta_{ij}-a_i^{T}N^{-1}a_jP_{jj} \tag{4}$$

where

$$\begin{aligned} \delta_{ij}&=\begin{cases}1 & i=j\\ 0 & i\neq j\end{cases} \\ N&=\sum_{k=1}^{n}(a_kP_{kk}a_k^{T}) \end{aligned} \tag{5}$$

3. THE DIFFERENTIAL RELATIONS

The differential relations include the following derivatives:

$$\frac{\mathrm{d}r_{ii}}{\mathrm{d}P_{jj}}, \frac{\mathrm{d}r_{ii}}{\mathrm{d}P_{ii}}, \frac{\mathrm{d}r_{ij}}{\mathrm{d}P_{ii}}, \frac{\mathrm{d}r_{ij}}{\mathrm{d}P_{jj}}, \frac{\mathrm{d}r_{ij}}{\mathrm{d}P_{kk}} \qquad (i \neq j \neq k, \text{below the same})$$

By the use of formulas:

$$\frac{\mathrm{d}(C \times D)}{\mathrm{d}x} = \frac{\mathrm{d}C}{\mathrm{d}x} D + C \frac{\mathrm{d}D}{\mathrm{d}x}$$

$$\frac{\mathrm{d}C^{-1}}{\mathrm{d}x} = -C^{-1} \frac{\mathrm{d}C}{\mathrm{d}x} C^{-1}$$

we can derive the above five derivatives sequentially as follows.

From eqs. (4) and (5), we have:

$$r_{ii} = 1 - a_i^{\mathrm{T}} \left(\sum_k a_k P_{kk} a_k^{\mathrm{T}} \right)^{-1} a_i P_{ii}$$

then:

$$\frac{\mathrm{d}r_{ii}}{\mathrm{d}P_{jj}} = -a_i^{\mathrm{T}} \frac{\mathrm{d}(\sum a_k P_{kk} a_k^{\mathrm{T}})^{-1}}{\mathrm{d}P_{jj}} a_i P_{ii} = a_i^{\mathrm{T}} N^{-1} a_j a_j^{\mathrm{T}} N^{-1} a_i P_{ii}$$

noticing that:

$$a_j^{\mathrm{T}} N^{-1} a_i P_{ii} = -r_{ji}$$

we obtain:

$$\frac{\mathrm{d}r_{ii}}{\mathrm{d}P_{jj}} = \frac{r_{ji}^2}{P_{ii}} = \frac{P_{ii}}{P_{jj}^2} r_{ij}^2 \tag{6}$$

Similarly, we have:

$$\frac{\mathrm{d}r_{ii}}{\mathrm{d}P_{ii}} = -a_i^{\mathrm{T}} \frac{\mathrm{d}N^{-1}}{\mathrm{d}P_{ii}} a_i P_{ii} - a_i^{\mathrm{T}} N^{-1} a_i \frac{\mathrm{d}P_{ii}}{\mathrm{d}P_{ii}} = -r_{ii} a_i^{\mathrm{T}} N^{-1} a_i$$

That is:

$$\frac{\mathrm{d}r_{ii}}{\mathrm{d}P_{ii}} = -r_{ii} (1 - r_{ii}) / P_{ii} \tag{7}$$

In the same way, we can derive the following equations:

$$\frac{\mathrm{d}r_{ij}}{\mathrm{d}P_{ii}} = -r_{ij} a_i^{\mathrm{T}} N^{-1} a_i = -r_{ij} (1 - r_{ii}) / P_{ii} \tag{8}$$

$$\frac{\mathrm{d}r_{ij}}{\mathrm{d}P_{jj}} = -r_{jj} a_i^{\mathrm{T}} N^{-1} a_j = r_{jj} r_{ij} / P_{jj} \tag{9}$$

$$\frac{\mathrm{d}r_{ij}}{\mathrm{d}P_{kk}}=\frac{r_{ik}r_{jk}}{P_{kk}^2}P_{jj} \tag{10}$$

Equations (6)–(10) are the corresponding differential relations.

From eqs. (6) and (7), it is easy to see that $\mathrm{d}r_{ii}/\mathrm{d}P_{jj}\geqslant 0$, and $\mathrm{d}r_{ii}/\mathrm{d}P_{ii}\leqslant 0$. This means that with the increase of one specific P_{ii}, the local redundancy of that observation r_{ii} will decrease while the local redundancies of all the other observations will increase. Similarly, if s $(1<s<n)$ weights get greater, the other $n-s$ local redundancies will become greater, with decrease of the sum of the s local redundancies. This shows that the redundancy of a specific adjustment system will be redistributed among all the observations when the weight matrix P is altered.

4. THE FUNCTIONAL RELATIONS

We now integrate both sides of eqs. (6)–(10) to obtain the relations between r and P as already expressed in eq. (4) but in a different form this time.

From eq. (7):

$$\frac{\mathrm{d}r_{ii}}{r_{ii}(1-r_{ii})}=-\frac{1}{P_{ii}}\,\mathrm{d}P_{ii}$$

after integration we have:

$$r_{ii}=\frac{1}{1+C_{ii}(\bar{P}_{ii})P_{ii}} \tag{11}$$

where $C_{ii}(\bar{P}_{ii})>0$ is a constant which is independent of the weight P_{ii} and is determined by the other $n-1$ weights and the geometry of the network.

As for eq. (8), after the replacement of eq. (11) and integration, we get:

$$r_{ij}=\frac{C_{ij}(\bar{P}_{ii})}{1+C_{ii}(\bar{P}_{ii})P_{ii}}=C_{ij}(\bar{P}_{ii})r_{ii} \tag{12}$$

Similarly, from eq. (9) we have:

$$r_{ij}=\frac{P_{jj}C_{ij}(\bar{P}_{jj})}{1+C_{jj}(\bar{P}_{jj})P_{jj}}=P_{jj}C_{ij}(\bar{P}_{jj})r_{jj} \tag{13}$$

As to eq. (6), substituting eq. (13) in it at first, we have:

$$\frac{\mathrm{d}r_{ii}}{P_{jj}^2C_{ij}^2(\bar{P}_{jj})r_{jj}^2}=\frac{P_{ii}}{P_{jj}^2}\,\mathrm{d}P_{jj}$$

and then carrying eq. (11) in the above equation, we obtain:

$$\mathrm{d}r_{ii}=P_{ii}C_{ij}^2(\bar{P}_{jj})\frac{1}{(1+C_{jj}(\bar{P}_{jj})P_{jj})^2}\,\mathrm{d}P_{jj}$$

Integrating both sides we obtain:

$$r_{ii}=C_{ii}(\bar{P}_{jj})-\frac{P_{ii}C_{ij}^{2}(\bar{P}_{jj})}{C_{jj}(\bar{P}_{jj})}r_{jj} \tag{14}$$

Substituting eqs. (13) and (11) in eq. (10), and then integrating, we have:

$$r_{ij}=C_{ij}(\bar{P}_{kk})-C_{ik}(\bar{P}_{kk})C_{jk}(\bar{P}_{kk})\frac{P_{jj}}{C_{kk}(\bar{P}_{kk})}r_{kk} \tag{15}$$

In the above equations, (11)-(15) each are formulas of one specific function expressed differently under different arguments. We may summarize above formulas as follows. r_{ii} is a linear fractional function of P_{ii}. Each r_{ij} and r_{ii} can be expressed in the form of a linear function of r_{kk}, like $a+b\times r_{kk}$. r_{ij} is proportional to r_{ii}. In these equations, the coefficients which decide each formula are independent of its related weight. Exception is the relation between r_{ij} and r_{jj}, where r_{ij} is proportional to $P_{jj}\times r_{jj}$ instead of r_{jj}.

In the derivation of the above relations, we see that we have differentiated the functions at first and then integrated them. Since these two operations are reversed, we can certainly obtain the above results in a more direct way. Thus, from the formula:

$$(C^{-1}+BD^{-1}A)^{-1}BD^{-1}=CB(D+ACB)^{-1}$$

we may get

$$r_{ii}=1-a_i^{\mathrm{T}}\left(\sum_{k\neq 1}a_kP_{kk}a_k^{\mathrm{T}}\right)^{-1}a_i\left(P_{ii}^{-1}+a_i^{\mathrm{T}}\left(\sum_{k\neq i}a_kP_{kk}a_k^{\mathrm{T}}\right)^{-1}a_i\right)^{-1}$$

let:

$$\bar{N}_i=\sum_{k\neq i}a_kP_{kk}a_k^{\mathrm{T}}$$

then

$$r_{ii}=1-a_i^{\mathrm{T}}\bar{N}_i^{-1}a_i\frac{1}{P_{ii}^{-1}+a_i^{\mathrm{T}}\bar{N}_i^{-1}a_i}$$

that is

$$r_{ii}=1/(1+a_i^{\mathrm{T}}\bar{N}_i^{-1}a_iP_{ii}) \tag{16}$$

Comparing with eq. (11), we have:

$$C_{ii}(\bar{P}_{ii})=a_i^{\mathrm{T}}\bar{N}_i^{-1}a_i \tag{17}$$

The other constants may be obtained similarly. However, the strategy of differentiation and then integration is more convenient and clearer to establish the independency of the constants on certain weight. The recursive method for repeated computation of $QvvP$ in this paper takes the advantage of these independencies of the related constants. As a matter of fact, it can also be developed from the sequential least squares (see Appendix A).

5. THE RECURSIVE METHOD

The repeated computation of $QvvP$ can be described as follows. Assume that the design matrix A is unchanged. Given the weight matrix P_{old} and its corresponding reliability matrix R_{old}, if we have an increment matrix ΔP, i.e. $P_{\text{new}}=P_{\text{old}}+\Delta P$, then, calculate the corresponding R_{new}.

It has to be pointed out that the functional relations given in Section 4 are true only when one weight is changed. If two or more weights are altered, the constants are not independent of one specific weight. So those equations can not be used directly to solve the above problem.

We now decompose the increment matrix ΔP as:

$$\Delta P=\Delta P_1+\Delta P_2+\dots\Delta P_m=\sum_k \Delta P_k \quad , \quad (1\leqslant m\leqslant n)$$

where

$$\Delta P_k=\operatorname{diag}(0, 0, \dots \Delta \mathrm{P}_{m_k m_k}, 0, \dots 0)$$

If we use the symbol:

$$P^{(k)}+\Delta P_{k+1}=P^{(k+1)} \quad , \quad k=0, 1, 2, \dots, m-1$$

we have the following corresponding equations:

$$\begin{array}{ll}
P_{\text{old}}=P^{(0)} & R_{\text{old}}=R^{(0)} \\
P^{(1)}=P^{(0)}+\Delta P_1 & R^{(1)} \\
P^{(2)}=P^{(1)}+\Delta P_2 & R^{(2)} \\
\vdots & \vdots \\
P_{\text{new}}P^{(m)}=P^{(m-1)}+\Delta P_m & R^{(m)}=R_{\text{new}}
\end{array}$$

Thus now each weight matrix $P^{(k+1)}$, has only one changed weight with respect to $P^{(k)}$ and therefore the functional relations listed in Section 4 can be used recursively. Obviously, at most by m calculations of $R^{(k+1)}$ with respect to $P^{(k+1)}$, we can obtain the required reliability matrix R_{new}. This is the principle of the recursive method. The concrete computational formulas and procedure follow.

With equations in Section 4, we have:

$$C_{ii}(\bar{P}_{ii}) = (1-r_{ii})/r_{ii}P_{ii} \tag{a}$$

$$C_{ij}(\bar{P}_{ii}) = r_{ij}/r_{ii} \tag{b}$$

$$C_{ji}(\bar{P}_{ii}) = r_{ji}/P_{ii}r_{ii} \tag{c}$$

$$C_{jj}(\bar{P}_{ii}) = r_{jj} + P_{jj}\cdot C_{ji}^2(\bar{P}_{ii})r_{ii}/C_{ii}(\bar{P}_{ii}) \tag{d}$$

$$C_{jk}(\bar{P}_{ii}) = r_{jk} + C_{ji}(\bar{P}_{ii})\cdot C_{ki}(\bar{P}_{ii})\cdot P_{kk}r_{ii}/C_{ii}(\bar{P}_{ii}) \tag{e}$$

In the above equations, (d) is the special case of (e) when $k=j$. Therefore only equations (a), (b), (c), (e) are needed in the repeated computation.

Imagining that all the constants $C(\bar{P}_{ii})$ make up a matrix C, from the above equations we know that only the i-th column of the matrix C will be used repeatedly, while its other elements may be calculated temporarily and therefore do not take up extra storage cells.

Let us assume that the reliability matrix $R^{(s-1)}$ has already been obtained after $(s-1)$ recursive computational steps. The new reliability matrix $R^{(s)}$ generated by the weight increment ΔP_{ii} can be obtained as follows.

(1) Calculate the constant:

$$C_{ii}^{(s-1)}(\bar{P}_{ii}) = (1-r_{ii}^{(s-1)})/(r_{ii}^{(s-1)}P_{ii}^{(s-1)})$$

and the element of $R^{(s)}$

$$r_{ii}^{(s)} = 1/(1+C_{ii}^{(s-1)}(\bar{P}_{ii})P_{ii}^{(s)})$$

(2) Calculate the constants:

$$C_{ij}^{(s-1)}(\bar{P}_{ii}) = r_{ij}^{(s-1)}/r_{ii}^{(s-1)})$$

$$C_{ji}^{(s-1)}(\bar{P}_{ii}) = r_{ji}^{(s-1)}/(P_{ii}^{(s-1)}r_{ii}^{(s-1)}) \qquad (j\neq i)$$

As elements $r_{ij}^{(s-1)}$ and $r_{ji}^{(s-1)}$ are only used once, the constants $C_{ij}^{(s-1)}(\bar{P}_{ii})$ and $C_{ji}^{(s-1)}(\bar{P}_{ii})$ can be temporarily stored in cells $R(i,j)$ and $R(j,i)$, which are used to store elements r_{ij} and r_{ji} of matrix R.

(3) Calculate the constant:

$$C_{jk}^{(s-1)}(\bar{P}_{ii}) = r_{ik}^{(s-1)} + C_{ji}^{(s-1)}(\bar{P}_{ii})C_{ki}^{(s-1)}(\bar{P}_{ii})P_{kk}^{(s-1)}r_{ii}^{(s-1)}/C_{ii}^{(s-1)}(\bar{P}_{ii})$$

and the element:

$$r_{jk}^{(s)} = C_{jk}^{(s-1)}(\bar{P}_{ii}) - C_{ji}^{(s-1)}(\bar{P}_{ii})C_{ki}^{(s-1)}(\bar{P}_{ii})P_{kk}^{(s-1)}r_{ii}^{(s)}/C_{ii}^{(s-1)}(\bar{P}_{ii})$$
$$(j\neq i,\ k\neq i)$$

and then store it in cell $R(j,k)$.

(4) Calculate the elements:

$$r_{ij}^{(s)} = C_{ij}^{(s-1)}(\bar{P}_{ii}) r_{ii}^{(s)}$$
$$r_{ji}^{(s)} = C_{ji}^{(s-1)}(\bar{P}_{ii}) P_{ii}^{(s)} r_{ii}^{(s)} \qquad (j \neq i)$$

and store them in cells $R(i,j)$ and $R(j,i)$.

We have now obtained all the elements of $R^{(s)}$. In practice, the procedure can be simplified. Because of the symmetry with weight, i.e.

$$r_{jk}/P_{kk} = r_{kj}/P_{jj}$$

the computation may be manipulated only for the upper or lower triangular part of the matrix R.

It should be noted that in almost all adjustment problems only the diagonal elements of R are needed. In these cases the computer processing time can be reduced still further.

Moreover, since weights are relative in nature, we can always keep the weights of one sort of observations unchanged. In our case now, we would certainly choose the weights of that sort of observations unchanged, which has the most number of observations. In this way, the number of recursive steps needed can be made much less than the total number of observations. In gross error detection, we often adjust the weights so as to reduce or delete their effects. Since the number of gross errors are very small, the recursive steps are also quite few. Moreover, with more gross errors detected, the needed number of recursive steps gets less and less.

6. TESTS AND CONCLUSION

The computational efficiency of the recursive method was tested with fictitious data of a bundle block adjustment of 3×7 photos. Each photo has 25 image points. We have altogether 465 image points and 169 object points, among which 12 points are chosen as control points. There are altogether 966 observations and 633 unknowns. The redundancy is 333. The computation was performed by a Siemens 7.570-C computer.

In the computation of matrix R by the conventional method, the banded characteristics of N and the sparse feature of A are adequately considered for both the computation of N^{-1} and $AN^{-1}A^{\mathrm{T}}P$.

Table 1 shows the CPU-time used to calculate the elements of matrix R by the new recursive method and the conventional method. We see that even when the number of changed weights amounts to 50 (approx. 5.2 percent of total number observations), the CPU-time used by the recursive method is half of that used by the conventional method. If only the diagonal elements of R need to be calculated, the CPU-time of the recursive method takes only about 2.3% of that of the conventional method.

Table 1 shows also that the tr(R) obtained by the two methods are practi-

TABLE 1

Comparison of the two methods

R	Fast recursive method						Conventional method
	$m=1$	$m=5$	$m=10$	$m=20$	$m=30$	$m=50$	
CPU-time (in seconds):							
All elements	11	35	65	126	186	305	604
Diagonal elements	5	5	5	6	8	13	560
Trace of matrix R (tr R):							
	333.0000	332.9999	332.9997	332.9995	332.9992	332.9987	333.000

cally the same (the figures are identical for all elements of R and just the diagonal elements). The discrepancies are due to the round-off error. As the calculation is recursive, the round-off error gets greater with the increase of the number of the recursive steps. However, this small amount of round-off error is not effective at all in practice.

In conclusion, the newly developed recursive method for repeated computation of reliability matrix R is correct, and is much more efficient than conventional methods, especially when only the diagonal elements need to be recomputed, which is almost always the case in practice.

REFERENCES

El-Hakim, S.F., 1982. Data snooping with weighted observations. In: International Archive of Photogrammetry, 24-III, pp. 126–133, Helsinki, Finland.

Förstner, W., 1979. Konvergenzbeschleunigung bei der a posteriori Varianzschätzung. Zeitschrift für Vermessungswesen, 104(3): 149–156.

Förstner, W., 1983. Reliability and discernability of extended Gauss - Markov models. Deutsche Geodätische Kommission, Reihe A, Heft 98, München, FR Germany.

Li Deren, 1983. Ein Verfahren zur Aufdeckung grober Fehler mit Hilfe der a posteriori Varianzschätzung. Bildmessung und Luftbildwesen, 51(5): 184–187.

APPENDIX A

As a change in the weight of observations can be considered as adding or subtracting an observation with suitable weight in or from the original normal equations, by the use of the sequential least squares, similar equations can be derived also. Let:

$$N=\sum_i a_i P_{ii} a_i^{\mathrm{T}}$$

$$\bar{N}=\sum_i a_i P_{ii} a_i^{\mathrm{T}} + a_k \Delta P_{kk} a_k^{\mathrm{T}}$$

Than we have:

$$\bar{N}^{-1}=N^{-1}-N^{-1}a_k(\Delta P_{kk}^{-1}+a_k^{\mathrm{T}}N^{-1}a_k)^{-1}a_k^{\mathrm{T}}N^{-1}$$

That is:

$$\bar{N}^{-1}=N^{-1}-\Delta P_{kk}N^{-1}a_k a_k^{\mathrm{T}}N^{-1}/(1+a_k^{\mathrm{T}}N^{-1}a_k\Delta P_{kk}) \tag{A-1}$$

From eq. (4) we have:

$$\bar{r}_{ij}=\delta_{ij}-a_i^{\mathrm{T}}\bar{N}^{-1}a_j P_{jj} \qquad (j\neq k)$$

Substituting eq. (A-1) we get:

$$\bar{r}_{ij}=r_{ij}+\Delta P_{kk}a_i^{\mathrm{T}}N^{-1}a_k a_k^{\mathrm{T}}N^{-1}a_j P_{jj}/(1+a_k^{\mathrm{T}}N^{-1}a_k\Delta P_{kk}) \qquad (j\neq k)$$

After noticing eq. (4), the above equation can be written as:

$$\bar{r}_{ij}=r_{ij}+\Delta P_{kk}(\delta_{ik}-r_{ik})(\delta_{kj}-r_{kj})/(P_{kk}+(1-r_{kk})\Delta P_{kk}) \qquad (j\neq k) \tag{A-2}$$

When $j=k$, we have:

$$\bar{r}_{ik}=\delta_{ik}-a_i^{\mathrm{T}}\bar{N}^{-1}a_k(P_{kk}+\Delta P_{kk})$$

Similarly, we can obtain:

$$\bar{r}_{ik}=\frac{P_{kk}r_{ik}+\Delta P_{kk}(r_{ik}-r_{kk}\delta_{ik})}{P_{kk}+(1-r_{kk})\Delta P_{kk}} \tag{A-3}$$

It can be proven that eqs. (A-2) and (A-3) are the same as eqs. (11)–(15) in nature. On the basis of formulas (A-2) and (A-3), the recursive algorithm can also be developed. It is equivalent to the one presented in this paper.