Metric Reconstruction with Missing Data under Weak Perspective

Ákos Pernek, Levente Hajder, Csaba Kazó
Budapest University of Technology and Economics
Department of Automation and Applied Informatics
Goldmann György tér 3. 1111-Budapest, Hungary
and MTA SZTAKI Kende u. 13-17. 1111-Budapest, Hungary
{pernek,hajder,kazo}@vision.sztaki.hu

Abstract

3D reconstruction with missing data has been a challenging computer vision task since the late 90s. This paper proposes a novel metric reconstruction algorithm dealing with the missing data problem. The algorithm is the adaption of the Fast Alternation method published by us in CAIP2007. We concentrate on metric instead of affine reconstruction because the quality of metric reconstruction is significantly better as it is demonstrated in this study. The solution is an alternation which consists of several substeps. All of these substeps are optimal with respect to the parameters that are being optimized. It is proved that the proposed algorithm converges to a local minimum. The solutions to the optimization subproblems in our approach are given by closed-form formulas, therefore the proposed method is relatively fast.

1 Introduction

Recovering scene geometry and camera motion has been attracting attention of the computer vision community since the late 80s. The classical factorization method for the full case – when the so-called measurement matrix is factorized into 3D motion and structure matrices – was developed by Tomasi and Kanade[12] in 1992. The weak-perspective extension was published by Weinshall and Kanade [13]. The factorization was extended to the paraperspective [8] case as well as to the real perspective [11] one.

The problem of missing data has already been addressed by Tomasi and Kanade [12]. They proposed a naive approach which transformed the missing data problem to the full matrix factorization by estimating the missing entries. Shum et al. [10] proposed a method to reconstruct the objects from range images. Their method is successfully applied to the Structure from Motion (SfM) problem by Buchanan et al. [3].

The mainstream idea to the factorization with missing data is to factorize the rank 4 measurement matrix into affine structure and motion matrices which are of size 4. (The Shum-method [10, 3] also computes affine structure and motion matrices, but the size of those matrices is 3.) This can be done by the mathematical method called Principal Component Analysis with Missing Data (PCAMD). This problem has been already addressed by mathematicians since the middle 70s [9]. These methods can be applied directly to the

BMVC 2008 doi:10.5244/C.22.68

SfM problem as it is written in [3]. Hartley & Schaffalitzky [6] proposed the PowerFactorization method which is based on the Power method. Power method is an iteration to compute the dominant n-dimensional subspace of a given matrix. Buchanan & Fitzgibbon [4] handled the problem as an alternation consisting of two nonlinear iterations to be solved. They suggested using the Damped-Newton method with line search to compute the optimal structure and motion matrices.

We consider the weak-perspective factorization in this paper. The weak-perspective methods can be used for perspective reconstruction if the elements of the measurement matrix are multiplied with the corresponding projective depths as it is proposed by Sturm et al. [11], or the perspective camera parameters can be estimated from weak-perspective camera reconstruction. Finally, the results can be refined by the well-known bundle adjustment method [2].

2 The factorization problem

Given P feature points of a rigid object tracked across F frames, the goal of the reconstruction is to recover the structure of the object as well as the 3D motion of the camera.

If the weak-perspective camera model is applied, a feature point can be written as $w_{fp} = q_f R_f S_p + t_f$, where S_p is the 3D vector of the p^{th} point of the object to be reconstructed, R_f is the first two rows of the 3D orientation matrix of the f^{th} camera, q_f is the nonzero scale factor of the same camera. t_f is the 2D offset vector of the origin while w_{fp} is a 2D vector containing the position of the p^{th} point in the f^{th} frame.

If all feature points are collected, the SfM problem can be written in matrix form:

$$W = [M|t]\begin{bmatrix} S \\ 1 \end{bmatrix}, \tag{1}$$

where the measurement matrix W consists of the trajectories of the feature points. Matrix $M = [M_1, M_2, \ldots, M_F]^T$ is called motion matrix, while $S = [S_1, S_2, \ldots, S_P]$ structure matrix. Motion submatrix M_f is written as $M_f = q_f R_f$. Therefore the following equation can be written: $M_f^T M_f = q_f^2 I \ \forall f$. We call this the weak-perspective motion constraint in the rest of this paper.

The goal of this study is to compute structure and motion matrices from the measurement matrix considering the motion constraint. Our solution is an alternation one that minimize the following reprojection error:

$$\left|\left| H \odot \left(W - [M|t]\begin{bmatrix} S \\ 1 \end{bmatrix} \right) \right|\right|_F^2 \tag{2}$$

subject to $M_f M_f^T = q_f^2 I$, $\forall f$. $||.||_F$ denotes the Frobenius norm of the error matrix, $A \odot B$ denotes the Hadamard product of matrices A and B[1]. H is the mask matrix: if $h_{2i-1,j} = h_{2i,j} = 1$ then the j^{th} feature point is seen in the i^{th} frame. It that is not seen, $h_{2i-1,j} = h_{2i,j} = 0$.

[1] $A \odot B = C$ if $c_{ij} = a_{ij} \cdot b_{ij}$.

3 Proposed method: Fast Alternation with Missing Data (FAMD)

As it is discussed in the introduction, the published SfM methods dealing with missing or uncertain data give affine and not metric results. We propose a novel method which is the modification of the Fast Alternation (FA) method published by us [5] in 2007. The FA algorithm does not deal with the missing data problem. With our modification, the case of missing data is handled as well.

Algorithm 1 Summary of FAMD algorithm

$M^{(0)}$,$t^{(0)}$,$S^{(0)}$ $\leftarrow$ Parameter Initialization
$\tilde{H}$, $\tilde{W}^{(0)}$, $\tilde{M}^{(0)}$, $\tilde{t}^{(0)}$ $\leftarrow$ Complete(H, W,$M^{(0)}$,$t^{(0)}$, $S^{(0)}$)
$k \leftarrow 0$
repeat
  $k \leftarrow k+1$
  $\tilde{t}^{(k)}$,$\tilde{W}^{(k)}$ $\leftarrow$ t-Step($\tilde{H}$,$\tilde{W}^{(k-1)}$,$S^{(k-1)}$)
  $\tilde{M}^{(k)}$ $\leftarrow$ M-Step($\tilde{H}$,$\tilde{W}^{(k)}$,$S^{(k-1)}$)
  $\tilde{W}^{(k)}$ $\leftarrow$ Complete(W,$\tilde{H}$,$\tilde{M}^{(k)}$,$S^{(k-1)}$)
  $S^{(k)}$ $\leftarrow$ S-Step($\tilde{H}$,$\tilde{W}^{(k)}$,$\tilde{M}^{(k)}$)
  $\tilde{W}^{(k)}$ $\leftarrow$ Complete(W,$\tilde{H}$,$\tilde{M}^{(k)}$,$S^{(k)}$)
  $\tilde{M}^{(k)}$, $S^{(k)}$, $\tilde{t}^{(k)}$ $\leftarrow$ Line Search ($\tilde{M}^{(k-1)}$, $\tilde{M}^{(k)}$, $S^{(k-1)}$, $S^{(k)}$, $\tilde{t}^{(k-1)}$, $\tilde{t}^{(k)}$)
  $\tilde{W}^{(k)}$ $\leftarrow$ Complete(W,$\tilde{H}$,$\tilde{M}^{(k)}$,$S^{(k)}$)
until $\left\|\tilde{H} \odot \left(\tilde{W}^{(k)} - \left[\tilde{M}^{(k)}|t^{(k)}\right]\begin{bmatrix} S^{(k)} \\ 1 \end{bmatrix}\right)\right\|_F^2$ converges.

The proposed method is a down-hill alternation to minimize the reprojection error defined in Eq. 2. Each cycle is divided into the following main steps:

1. Parameter Initialization: The reprojection error is based on the difference between the feature points and the reprojected points. For the M-step (which is described later) 3D vectors are needed. For this reason, the measurement matrix W and motion matrix M should be completed.

   The completion should be carried out for the motion matrix M and the offset vector t. The motion submatrix M_f can easily be completed: M_f consists of two rows which are (quasi) orthogonal. It is completed by the third row which should be orthogonal to the first two row-vectors. Its length is chosen to be the average length of the other two vectors. This third row is denoted by $m_{(f,3)}$. The offset vector t_f is simply completed by a zero element. (The third element of the offset vector is denoted by $t_{(f,3)}$ in the rest of this paper.)

   The corresponding measurement matrix W_f is completed by applying the projection of the corresponding part. Therefore the third row of W_f is computed as $W_{(f,3)} = m_{(f,3)}S$.

2. S-step: Tha aim of this step is to determine the structure matrix optimally. The elements of the structure matrix can be arbitrary. This is a linear problem w.r.t. the structure matrix which can be solved by the pseudoinverse operator both for

full factorization and for factorization with missing data. The solution is described in [10] in detail.

3. M-step. The goal of the M-step is to improve the motion matrix M_f while the other matrices and vectors are fixed. It is trivial that the value of M_i is independent from that of M_j if $i \neq j$. Therefore, the task is to minimize the following error for all f: $||W_f - M_f S||$, where matrices W_f and S are fixed. The problem should be solved considering the $M_f^T M_f = q_f^2 I$ constraint. This is a 3D point set registration problem where the 3D points are contained by the columns of matrices W_f and S. The solution to the registration problem is given in the appendix.

4. t-step. The optimal offset t_f is also given in the appendix: it is the difference between the centers of gravity of the vectors contained by matrices S, and W_f.

5. Line Search. Numerous iteration steps might be needed if the surface of the reprojection error is very flat. To speed up the algorithm, a line search method can be inserted. We have applied a line search method based on the stronge Wolfe-conditions using cubic interpolation.

6. Completion Step. After each iteration, the third rows of the measurement submatrices (W_f-s) should be completed. This is carried out by the following equation: $W_{(f,3)} = m_{(f,3)} S + t_{(f,3)} \cdot [1, \ldots, 1]$.

The proposed method guarantees that a local optimum will be reached because each step of the algorithm descreases the reprojection error. The algorithm runs until the difference between the subsequent error values drops below a given limit ε.

4 Parameter initialization

The proposed method requires initial values of the matrices. The key idea of our initialization is that the factorization with missing data can be divided into full matrix factorizations.

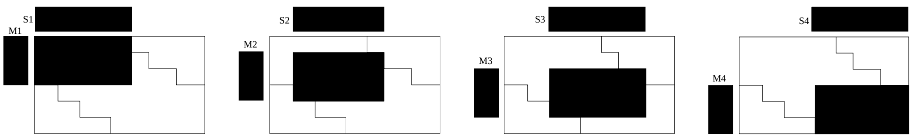

Figure 1: Problem divided into factorization of submatrices.

The Tomasi-Kanade factorization needs at least 3 frames to compute structure and motion. Therefore, the selected feature points should be visible in frames 1, 2, and 3 as it is visualized in the left image of Fig. 1. Then the Tomasi-Kanade factorization [12] with the extension of Weinshall and Kanade [13] is run with the selected feature points. Motion and structure matrices (M_1 and S_1) and offset vector t_1 are obtained by full matrix factorization.

Then another full matrix is formed with the feature points visible in frames 2, 3, and 4. By applying the factorization, motion matrix M_2, structure matrix S_2, and offset vector t_2 are computed. Matrices M_3, S_3, and offset vector t_3 are computed from the points visible in frames 3,4, and 5. This process is repeated until the matrices M_{F-2}, S_{F-2}, and t_{F-2} are obtained. The steps of parameter initialization are summarized in Algorithm 2.

Algorithm 2 Parameter Initialization

$W_1 \leftarrow$ Feature points common in frames 1,2,3.
$M_1, S_1, t_1 \leftarrow$ TomasiKanade(W_1)
for $i = 2$ to $(F-2)$ **do**
  $W_i \leftarrow$ Common feature points of frames i,$i+1$,$i+2$.
  $M_i, S_i, t_i \leftarrow$ TomasiKanade(W_i)
  Put points common in S and S_i into S' and S'_i, respectively.
  Register matrices S' and S'_i to each other.
  Update matrices M and S and offset vector t
end for

The key problem of the initialization is to register the results of the new factorization to the already registered ones which are contained by matrices M, S, and vector t. New factorization in the i^{th} cycle can be written as

$$[W_i|t_i]\begin{bmatrix} S_i \\ 1 \end{bmatrix}. \tag{3}$$

The feature points common in S and S_i are denoted by S', and S'_i, respectively. These point sets should be registered to each other by the method described in the appendix. If s'^j and $s_i'^j$ denote the j^{th} elements of the point sets, the registration is given by

$$s'^j \approx qR(s_i'^j - o_2) + o_1, \tag{4}$$

where q, R, o_1, and o_2 are the scale factor, rotation matrix, center of gravity of the first, and that of the second point set, respectively. The original structure matrix S_i should be transformed as well. After transformation, the new points are added to the point set S.

The last two rows of M_i and t_i are inserted at the end of M, and t, respectively, if the matrices M_i and t_i are transformed as follows:

$$M_i \leftarrow \quad \frac{1}{q} M_i R^T, \tag{5}$$

$$t_i \leftarrow \quad t_i + M_i o_2 - \frac{1}{q} M_i R^T o_1. \tag{6}$$

5 Tests on synthesized data

Several experiments with synthetic data have been carried out to study the properties of the proposed method. In this section, our FAMD method is compared with the following methods: (i) **PowFac:** PowerFactorization [6] of Hartley&Schaffalitzky. (ii) **Shum-Bucha:** The method of Shum et al. [10] applied to the SfM problem as it is written in [3]. (iii) **DamNew:** Damped Newton by Buchanan&Fitzgibbon [4].

In order to compare the affine methods listed above to the proposed alternation algorithm, the computation of the metric 3D structure is carried out by the classical weak-perspective Tomasi-Kanade factorization. The $2F \times 4$ affine motion is multiplied by the $4 \times P$ affine structure matrix, and a full measurement matrix is obtained. Then this

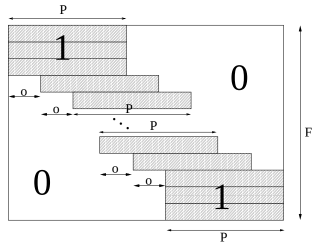

Figure 2: Structure of mask matrix.

measurement matrix is factorized by Tomasi-Kanade algorithm [12] with the Weinshall-Kanade [13] extension.

The rival algorithms were coded in Matlab by Buchanan and they can be downloaded from the author's homepage [2]. The FAMD algorithm was also implemented in Matlab. The methods were run under Octave[3] on an Intel P4 2.4 GHz PC with 512 MByte memory.

5.1 Generation of the measurement matrix

The input (measurement matrix) is composed of 2D trajectories. These trajectories are generated in the following way: (i) Random three-dimensional coordinates are generated by a zero-mean Gaussian random number generator with variance σ_{3D}. (ii) The generated 3D points are rotated by random angles. (iii) The rotated points are projected onto the image plane using weak perspective projection. (iv) Noise is added to the projected coordinates. It is generated by a zero-mean Gaussian random number generator as well. Its variance is set to σ_{2D}. (v) Finally, the measurement matrix W is composed with the projected points. (vi) Motion and structure parameters are initialized as it is described in Sec. 4. For each test case, 40 measurement matrices are generated and the rival reconstuction methods are executed 40 times. The results shown in this section is calculated as the average of the 40 executions.

5.2 Generation of the mask matrix

The mask generator algorithm has three parameters: (i) P: Number of the visible points in each frame, (ii) F: Number of frames, (iii) O: Offset between the neighboring frames. For the tests, offset was set to $O = 2$.

The structure of the generated mask is illustrated in Fig. 2. The missing data ratio in the matrix is calculated by the following equation: $md_{\%} = 100PF/(F(P+(F-5)O))$.

General remarks. The tests show that the FAMD algorithm outperforms the other methods in every test case. Only the Shum-Buchanan algorithm [10, 4] can be up to the proposed algorithm in the second test (Fig. 4). Examining the time demand, it is clear that the fastest algorithm is the Damped-Newton [4], the second one is the proposed FAMD algorithm. Note that the measured time does not contain the time required for initialization.

[2] http://www.robots.ox.ac.uk/$\sim$amb/

[3] Octave is a Matlab compatible interpreter. See www.octave.org

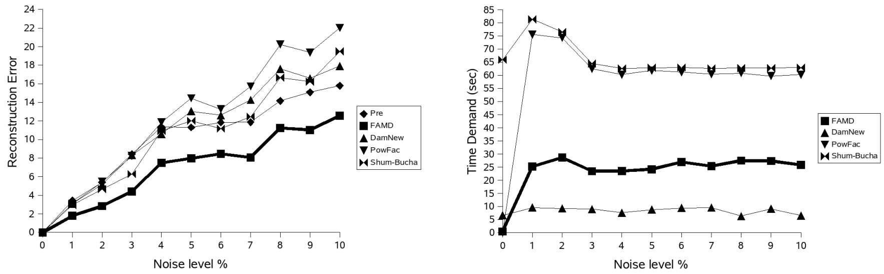

Figure 3: Reconstruction error and time demand w.r.t. 2D noise.

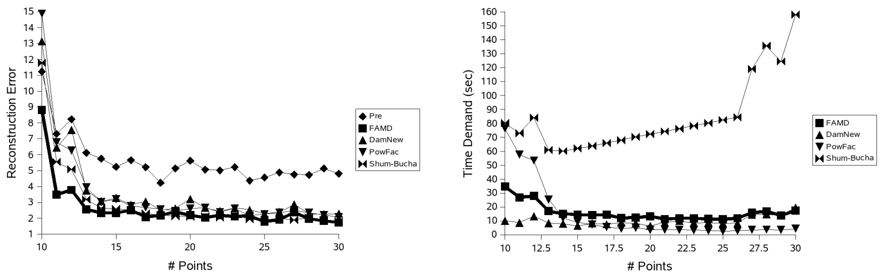

Figure 4: Reconstruction error and time demand w.r.t. number of points.

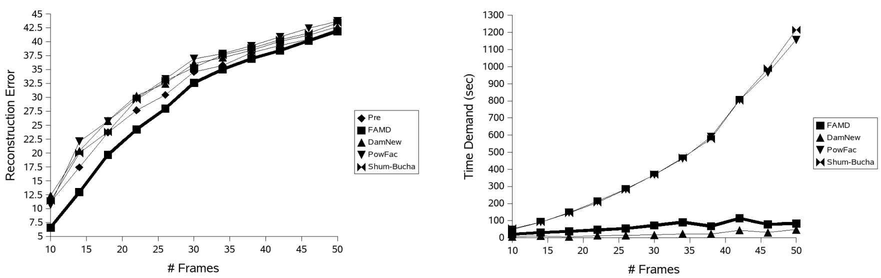

Figure 5: Reconstruction error and time demand w.r.t. number of frames.

Error versus noise (Figure 3) The methods are run with gradually increasing noise level. The reconstruction error increases approximately in a linear way for all the methods. The test sequence consists of 10 frames, and $P = 10$ was set. The missing data ratio is 50%. The noise level is calculated as $100\sigma_{2D}/\sigma_{3D}$.

The test indicates that the proposed FAMD significantly outpowers the rival algorithms. The FAMD algorithm is fast, only the Damped Newton is faster.

Error versus number of points (Fig. 4) P increases from 10 to 30. (The missing data rate decreases from 50% to 25%.) The noise level is 5%, and the sequence consists of 10 frames. The FAMD yields better results than the other algorithms, except for 3 test cases when the Shum-Buchanan [10, 4] algorithm achieves the better reconstruction. The other two algorithms yield significantly more error.

Error versus number of frames (Fig. 5). F increases from 10 to 50. The missing data ratio increases from 50% to 90%.The noise level was 5%, and $P = 10$ was set. In each test case, the most efficient algorithm was the proposed one. The plot shows that the error increases with the number of frames because the missing data ratio is high when the test sequence consists of many frames. It is important to note that the Shum-Buchanan [10, 3] and PowerFactorization [6] methods are very slow, their time demand seems to be a nonlinear function of the number of frames while the running time of the Damped Newton and the FAMD algorithms are approximately linear w.r.t. frame number.

Figure 6: Original images and corresponding masks of 'Cat' sequence.

6 Test on real data

We have tested the proposed FAMD algorithm on our 'Cat' sequence. The cat statuette was rotated on a table and 92 photos were taken by a commercial digital camera. The regions of the statuette in the images were automatically determined as it is demonstrated in fig 6.

Feature points were detected using the widely used KLT algorithm, and the points were tracked by correlation-based template matching method. The measurement matrix of the sequence consists of 2290 points and 92 frames. The missing data ratio is 82%.

The 3D reconstructed points are visualized on Fig. 7. The running time of the FAMD algorithm was 3914 sec. The reconstruction is made by DampedNewton, PowerFactorization, Shum-Buchanan algorithms as well. The time demand was 29656, 48540, and 41840 seconds, respectively. The FAMD algorithm is significantly faster in the real test case than the rival ones. The test on synthesized data have suggested that DampedNewton is the fastest algorithm. This is not true for the real test, because the main iteration in

DampedNewton has a matrix inversion and the size of this matrix is $16PF \times 16PF$ where P is the number of points, F that of frames. Thus, the algorithm can be very slow when F and/or P are high which is tipical when real object reconstruction is carried out.

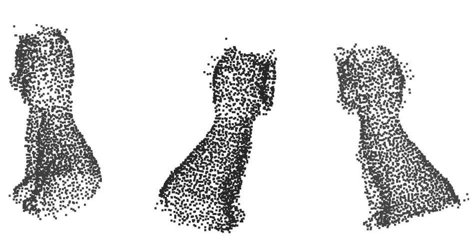

Figure 7: Reconstructed 3D points of 'Cat' sequence.

7 Conclusion

We have presented a novel algorithm to compute metric reconstruction from measurement matrix containing the moving feature points of a rigid object. Elements are allowed to be missing in the matrix. Our tests imply that the proposed algorithm gives better results than the previously published ones. The novel algorithm is relatively fast because the steps of the proposed alternation method solve subproblems for which closed-form solutions exist.
Acknowledgement. The authors would like to express his gratitude to Prof. Dmitry Chetverikov for his council and for providing his mask generator algorithm.

References

[1] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Trans. on PAMI*, 9(5):698–700, 1987.

[2] Bill Triggs and Philip McLauchlan and Richard Hartley and Andrew Fitzgibbon. Bundle Adjustment – A Modern Synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

[3] A. M. Buchanan. Investigation into matrix factorization when elements are unknown. Technical report, University of Oxford, `http://www.robots.ox.ac.uk/~amb`, 2004.

[4] A. M. Buchanan and A. W. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *CVPR05*, volume 2, pages 316–322, 2005.

[5] L. Hajder, Á. Pernek, and Cs. Kazó. Fast and Precise Weak-Perspective Factorization. In *The 12th International Conference on Computer Analysis of Images and Patterns*, pages 498–505, 2007.

[6] Richard Hartley and Frederik Schaffalitzky. Powerfactorization: 3d reconstruction with missing or uncertain data, 2003.

[7] B.K.P. Horn, H.M. Hilden, and S. Negahdaripourt. Closed-form Solution of Absolute Orientation Using Orthonormal Matrices. *Journal of the Optical Society of America*, 5(7):1127–1135, 1988.

[8] C. J. Poelman and T. Kanade. A Paraperspective Factorization Method for Shape and Motion Recovery. *IEEE Trans. on PAMI*, 19(3):312–322, 1997.

[9] A. Ruhe. Numerical computation of principal components when several observations are missing. Technical report, Departure of Information Processing, Umea Univesity, Sweden, 1974.

[10] Heung-Yeung Shum, Katsushi Ikeuchi, and Raj Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(9):854–867, 1995.

[11] P. Sturm and B. Triggs. A Factorization Based Algorithm for Multi-Image Projective Structure and Motion. In *ECCV*, volume 2, pages 709–720, 1996.

[12] Tomasi, C. and Kanade, T. Shape and Motion from Image Streams under orthography: A factorization approach. *Intl. Journal Computer Vision*, 9:137–154, 1992.

[13] D. Weinshall and C. Tomasi. Linear and Incremental Acquisition of Invariant Shape Models From Image Sequences. *IEEE Trans. on PAMI*, 17(5):512–517, 1995.

A Optimal 3D point set registration

Given two point sets, both containing N points, the goal is to minimize the registration error with respect to the rotation matrix R, scale factor s and 3D offset vector o. The registration error is defined by the following formula: $\sum_{i=1}^{N}(a_i-(sRb_i+o))^T(a_i-(sRb_i+o))$, where a_i and b_i are the i^{th} 3D vector of the first and second point set, respectively. Horn et al. [7] proved that the registration error defined above is minimized optimally with respect to the offset vector if o is the difference between the centers of gravity of the two 3D point sets.

Let us subtract the center of gravity from the datasets. Let denote the i^{th} points of the new point sets by a'_i and b'_i, respectively. The registration error is modified as follows: $\sum_{i=1}^{N}(a'_i-sRb'_i)^T(a'_i-sRb'_i)$ It can be shown by calculating the derivative of the error function with respect to R that the minimization problem is equivalent to maximizing the $\sum_{i=1}^{N}sa_i'^T Rb'_i$ expression w.r.t. R. Scale vector does not influence the maximum, thus the problem is to maximize $\sum_{i=1}^{N}a_i'^T Rb'_i$. Arun et al. [1] proved that if the SVD of matrix $\sum_{i=1}^{N}b'_i a_i'^T$ is UDV^T, then the optimal solution is $R=V^TU$.

Let us rotate the second point set by R. The rotated vectors are denoted by double prime: $b''_i=Rb'_i$. The registration error becomes $\sum_{i=1}^{N}(a'_i-sb''_i)^T(a'_i-sb''_i)$. The scale factor can be calculated optimally by differentiating the registration error with rescpect to the scale: $\sum_{i=1}^{N}\left(sb_i''^T b''_i-a_i'^T b''_i\right)=0$ The solution is given by the following formula:$s=\sum_{i=1}^{N}a_i'^T b''_i/\sum_{i=1}^{N}b"_i^T b"_i$. Note that Horn et al. [7] also proposed a formula to compute the scale, but their solution is not optimal.