The 7-Parameter Helmert Transformation

David Eberly, Geometric Tools, Redmond WA 98052
https://www.geometrictools.com/

Created: November 11, 2024

Contents

1 Introduction **2**

2 Derivatives of the Least-Squares Function **2**

2.1 Solving for Euler Angle 0 3

2.2 Solving for Euler Angle 1 4

2.3 Solving for Euler Angle 2 4

3 Iterative Algorithm for Minimization **5**

1 Introduction

The 7-parameter Helmert transformation [1] is designed to rotate (by a matrix R), translate (by a vector $\boldsymbol{T}$), and uniformly scale (by a scalar $s > 0$) one 3D point set, $\{\boldsymbol{q}_i\}_{i=0}^{n-1}$, to be as close as possible to another 3D point set, $\{\boldsymbol{p}_i\}_{i=0}^{n-1}$. The transformation is based on a least-squares algorithm. The error function to be minimized is

$$E(R, \boldsymbol{T}, s) = \sum_i |sR\boldsymbol{q}_i + \boldsymbol{T} - \boldsymbol{p}_i|^2 \tag{1}$$

The rotation matrix is represented as a product of Euler rotations, say, $R(a_0, a_1, a_2) = R_0(a_0)R_1(a_1)R_2(a_2)$. The translation vector is $\boldsymbol{T} = (t_0, t_1, t_2)$. The uniform scale is s. The error function is therefore dependent on 7 parameters.

Helmert's approach uses the input point sets as is. However, the framework is simplified by translating both point sets by the average of the $\boldsymbol{q}$-points given by $\overline{\boldsymbol{q}} = (\sum_i \boldsymbol{q}_i)/n$. The average of the $\boldsymbol{p}$-points is $\overline{\boldsymbol{p}} = (\sum_i \boldsymbol{p}_i)/n$. Define $\boldsymbol{u}_i = \boldsymbol{p}_i - \overline{\boldsymbol{q}}$ and $\boldsymbol{v}_i = \boldsymbol{q}_i - \overline{\boldsymbol{q}}$ for all i. The average of the $\boldsymbol{v}$-points is the zero vector.

Observe that

$$\begin{aligned} sR\boldsymbol{q}_i + \boldsymbol{T} - \boldsymbol{p}_i &= sR(\boldsymbol{v}_i + \overline{\boldsymbol{q}}) + \boldsymbol{T} - (\boldsymbol{u}_i + \overline{\boldsymbol{q}}) \\ &= sR\boldsymbol{v}_i + (\boldsymbol{T} - \overline{\boldsymbol{q}} + sR\,\overline{\boldsymbol{q}}) - \boldsymbol{u}_i \\ &= sR\boldsymbol{v}_i + \boldsymbol{B} - \boldsymbol{u}_i \end{aligned} \tag{2}$$

where the last equality defines $\boldsymbol{B}$. After the translation by the average of the $\boldsymbol{q}$-points, the function to minimize is

$$\begin{aligned} F(R, \boldsymbol{B}, s) &= \textstyle\sum_i |sR\boldsymbol{v}_i + \boldsymbol{B} - \boldsymbol{u}_i|^2 \\ &= s^2 \textstyle\sum_i \boldsymbol{v}_i^\top \boldsymbol{v}_i + 2s \sum_i (\boldsymbol{B} - u_i)^\top R\boldsymbol{v}_i + \sum_i (\boldsymbol{B} - \boldsymbol{u}_i)^\top (\boldsymbol{B} - \boldsymbol{u}_i) \\ &= s^2 \textstyle\sum_i \boldsymbol{v}_i^\top \boldsymbol{v}_i - 2s \sum_i \boldsymbol{u}_i^\top R\boldsymbol{v}_i + \sum_i (\boldsymbol{B} - \boldsymbol{u}_i)^\top (\boldsymbol{B} - \boldsymbol{u}_i) \end{aligned} \tag{3}$$

where $\sum_i \boldsymbol{B}^\top R\boldsymbol{v}_i = \boldsymbol{B}^\top R \sum_i \boldsymbol{v}_i = \boldsymbol{0}$ because the average of the $\boldsymbol{v}$-points is the zero vector.

In the next section we will see that $\boldsymbol{B} = \overline{\boldsymbol{p}}$. The Helmert paper has $\boldsymbol{T} = \overline{\boldsymbol{p}} - sR\,\overline{\boldsymbol{q}}$. In this document we have $\boldsymbol{T} = \overline{\boldsymbol{p}} + \overline{\boldsymbol{q}} - sR\,\overline{\boldsymbol{q}}$. The $\boldsymbol{q}$-points average is added to undo the translation by that average when generating $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ from $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$, thereby producing the alignment in the original coordinate system.

2 Derivatives of the Least-Squares Function

Half the s-derivative is

$$\frac{1}{2}\frac{dF}{ds} = s\sum_i \boldsymbol{v}_i^\top \boldsymbol{v}_i - \sum_i \boldsymbol{u}_i^\top R\boldsymbol{v}_i \tag{4}$$

which has the s-root

$$s = \sum_i \boldsymbol{u}_i^\top R\boldsymbol{v}_i \Big/ \sum_i \boldsymbol{v}_i^\top \boldsymbol{v}_i \tag{5}$$

Half the $\boldsymbol{B}$-derivative (a 3-tuple gradient) is

$$\frac{1}{2}\frac{dF}{d\boldsymbol{B}} = \sum_i \boldsymbol{B} - u_i = n\boldsymbol{B} - \sum_i \boldsymbol{u}_i \tag{6}$$

which has the $\boldsymbol{B}$-root

$$\boldsymbol{B} = \frac{1}{n}\sum_i \boldsymbol{u}_i = \overline{\boldsymbol{u}} \tag{7}$$

This is the average $\overline{\boldsymbol{u}}$ of the $\boldsymbol{u}$-points.

Half the a_j-derivative is

$$\frac{1}{2}\frac{dF}{da_j} = -2s\sum_i \boldsymbol{u}_i^{\mathsf{T}}\frac{\partial R}{\partial a_j}\boldsymbol{v}_i \tag{8}$$

The first fractional term on the right-hand side of the equation is the uniform scale s. This is assumed to be positive, so the a_j-root of the derivative is a solution to

$$\sum_i \boldsymbol{u}_i^{\mathsf{T}}\frac{\partial R}{\partial a_j}\boldsymbol{v}_i = 0 \tag{9}$$

An iterative numerical method for minimization can be formulated to search the 3-parameter space of Euler angles for the minizing angles. One such method cycles through the angles by keeping 2 of the Euler angles constant and solving equation (9) for the other Euler angle.

2.1 Solving for Euler Angle 0

Let $j = 0$, where the rotation corresponding to the Euler angle and its derivative are

$$R_0(a_0) = \begin{bmatrix} \cos a_0 & -\sin a_0 & 0 \\ \sin a_0 & \cos a_0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \frac{\partial R_0}{\partial a_0} = \begin{bmatrix} -\sin a_0 & -\cos a_0 & 0 \\ \cos a_0 & -\sin a_0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{10}$$

For a specified rotation matrix $R(a_0, a_1, a_2) = R_0(a_0)R_1(a_1)R_2(a_2)$, the derivative equation for which we seek the minimum with respect to a_0 is

$$0 = \sum_i \boldsymbol{u}_i^{\mathsf{T}}\frac{\partial R}{\partial a_0}\boldsymbol{v}_i = \sum_i \boldsymbol{u}_i^{\mathsf{T}}\frac{\partial R_0}{\partial a_0}R_1R_2\boldsymbol{v}_i = \sum_i \boldsymbol{\ell}_i^{\mathsf{T}}\frac{\partial R_0}{\partial a_0}\boldsymbol{r}_i \tag{11}$$

where $\boldsymbol{\ell}_i = \boldsymbol{u}_i$ and $\boldsymbol{r}_i = R_1R_2\boldsymbol{v}_i$ for all i. As 3-tuples, define $\boldsymbol{\ell}_i = (\ell_{i0}, \ell_{i1}, \ell_{i2})$ and $\boldsymbol{r}_i = (r_{i0}, r_{i1}, r_{i2})$; then

$$0 = \sum_i \boldsymbol{\ell}_i^{\mathsf{T}}\frac{\partial R_0}{\partial a_0}\boldsymbol{r}_i = (\cos a_0)\sum_i(\ell_{i1}r_{i0} - \ell_{i0}r_{i1}) - (\sin a_0)\sum_i(\ell_{i0}r_{i0} + \ell_{i1}r_{i1}) \tag{12}$$

The equation can be solved for

$$(\sin a_0, \cos a_0) = \pm\frac{\left(\sum_i(\ell_{i1}r_{i0} - \ell_{i0}r_{i1}), \sum_i(\ell_{i0}r_{i0} + \ell_{i1}r_{i1})\right)}{\sqrt{\left(\sum_i(\ell_{i1}r_{i0} - \ell_{i0}r_{i1})\right)^2 + \left(\sum_i(\ell_{i0}r_{i0} + \ell_{i1}r_{i1})\right)^2}} \tag{13}$$

2.2 Solving for Euler Angle 1

Let $j = 1$, where the rotation corresponding to the Euler angle and its derivative are

$$R_1(a_1) = \begin{bmatrix} \cos a_1 & 0 & \sin a_1 \\ 0 & 1 & 0 \\ -\sin a_1 & 0 & \cos a_1 \end{bmatrix}, \quad \frac{\partial R_1}{\partial a_1} = \begin{bmatrix} -\sin a_1 & 0 & \cos a_1 \\ 0 & 0 & 0 \\ -\cos a_1 & 0 & -\sin a_1 \end{bmatrix} \tag{14}$$

For a specified rotation matrix $R(a_0, a_1, a_2) = R_0(a_0)R_1(a_1)R_2(a_2)$, the derivative equation for which we seek the minimum with respect to a_1 is

$$0 = \sum_i \boldsymbol{u}_i^\mathsf{T} \frac{\partial R}{\partial a_1} \boldsymbol{v}_i = \sum_i \boldsymbol{u}_i^\mathsf{T} R_0 \frac{\partial R_1}{\partial a_1} R_2 \boldsymbol{v}_i = \sum_i \boldsymbol{\ell}_i^\mathsf{T} \frac{\partial R_1}{\partial a_1} \boldsymbol{r}_i \tag{15}$$

where $\boldsymbol{\ell}_i = R_0^\mathsf{T} \boldsymbol{u}_i$ and $\boldsymbol{r}_i = R_2 \boldsymbol{v}_i$ for all i. As 3-tuples, define $\boldsymbol{\ell}_i = (\ell_{i0}, \ell_{i1}, \ell_{i2})$ and $\boldsymbol{r}_i = (r_{i0}, r_{i1}, r_{i2})$; then

$$0 = \sum_i \boldsymbol{\ell}_i^\mathsf{T} \frac{\partial R_1}{\partial a_1} \boldsymbol{r}_i = (\cos a_1) \sum_i (\ell_{i0} r_{i2} - \ell_{i2} r_{i0}) - (\sin a_1) \sum_i (\ell_{i0} r_{i0} + \ell_{i2} r_{i2}) \tag{16}$$

The equation can be solved for

$$(\sin a_1, \cos a_1) = \pm \frac{\left(\sum_i (\ell_{i0} r_{i2} - \ell_{i2} r_{i0}), \sum_i (\ell_{i0} r_{i0} + \ell_{i2} r_{i2})\right)}{\sqrt{\left(\sum_i (\ell_{i0} r_{i2} - \ell_{i2} r_{i0})\right)^2 + \left(\sum_i (\ell_{i0} r_{i0} + \ell_{i2} r_{i2})\right)^2}} \tag{17}$$

2.3 Solving for Euler Angle 2

Let $j = 2$, where the rotation corresponding to the Euler angle and its derivative are

$$R_2(a_2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos a_2 & -\sin a_2 \\ 0 & \sin a_2 & \cos a_2 \end{bmatrix}, \quad \frac{\partial R_2}{\partial a_2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\sin a_2 & -\cos a_2 \\ 0 & \cos a_2 & -\sin a_2 \end{bmatrix} \tag{18}$$

For a specified rotation matrix $R(a_0, a_1, a_2) = R_0(a_0)R_1(a_1)R_2(a_2)$, the derivative equation for which we seek the minimum with respect to a_2 is

$$0 = \sum_i \boldsymbol{u}_i^\mathsf{T} \frac{\partial R}{\partial a_2} \boldsymbol{v}_i = \sum_i \boldsymbol{u}_i^\mathsf{T} R_0 R_1 \frac{\partial R_2}{\partial a_2} \boldsymbol{v}_i = \sum_i \boldsymbol{\ell}_i^\mathsf{T} \frac{\partial R_2}{\partial a_2} \boldsymbol{r}_i \tag{19}$$

where $\boldsymbol{\ell}_i = (R_0 R_1)^\mathsf{T} \boldsymbol{u}_i$ and $\boldsymbol{r}_i = \boldsymbol{v}_i$ for all i. As 3-tuples, define $\boldsymbol{\ell}_i = (\ell_{i0}, \ell_{i1}, \ell_{i2})$ and $\boldsymbol{r}_i = (r_{i0}, r_{i1}, r_{i2})$; then

$$0 = \sum_i \boldsymbol{\ell}_i^\mathsf{T} \frac{\partial R_2}{\partial a_2} \boldsymbol{r}_i = (\cos a_2) \sum_i (\ell_{i2} r_{i1} - \ell_{i1} r_{i2}) - (\sin a_2) \sum_i (\ell_{i1} r_{i1} + \ell_{i2} r_{i2}) \tag{20}$$

The equation can be solved for

$$(\sin a_2, \cos a_2) = \pm \frac{\left(\sum_i (\ell_{i2} r_{i1} - \ell_{i1} r_{i2}), \sum_i (\ell_{i1} r_{i1} + \ell_{i2} r_{i2})\right)}{\sqrt{\left(\sum_i (\ell_{i2} r_{i1} - \ell_{i1} r_{i2})\right)^2 + \left(\sum_i (\ell_{i1} r_{i1} + \ell_{i2} r_{i2})\right)^2}} \tag{21}$$

3 Iterative Algorithm for Minimization

The Euler angles for the minimum of F in equation (3) are constructed by cycling through the angles and computing the solutions of equations (12), (17), and (21). For each iteration, compute $R = R_0(a_0)R_1(a_1)R_2(a_2)$ for 2 of the Euler angles treated as constants and then solve the correpsonding equation for the othe Euler angle. Once the rotation is known, equation (5) is used to compute the uniform scale s. The translation $\boldsymbol{B}$ is provided by equation (7) but must be adjusted to be in the original coordinate system.

The steps are as follows.

1. Inputs: $\{\boldsymbol{u}_i\}_{i=0}^{n-1}$ and $\{\boldsymbol{v}_i\}_{i=0}^{n-1}$.
2. Compute the averages $\overline{\boldsymbol{u}} = (\sum_i \boldsymbol{u}_i)/n$ and $\overline{\boldsymbol{v}} = (\sum_i \boldsymbol{v}_i)/n$.
3. Replace $\boldsymbol{u}_i \leftarrow \boldsymbol{u}_i - \overline{\boldsymbol{v}}$ and $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i - \overline{\boldsymbol{v}}$. The centroid of the adjusted $\boldsymbol{v}$-points is now at the origin. The adjusted $\boldsymbol{u}_i$-points are translated by the centroid amount.
4. Set $\boldsymbol{B} = \overline{\boldsymbol{u}} - \overline{\boldsymbol{v}}$.
5. The initial guess at rotation R is the identity matrix I, so $a_0 = a_1 = a_2 = 0$. The matrixes $R_0(a_0)$, $R_1(a_1)$, and $R_2(a_2)$ are stored during the iterations to find the minimizer angles.
6. Cycle through the Euler angles for a gradient-descent-like algorithm. The subroutine for this algorithm is listed after the current set of steps. The subroutine returns updated R and s values. The cycles terminate when $F(R, \boldsymbol{B}, s)$ meets a user-specified criterion. For example, the criterion might be that the difference in F-values for 2 consecutive cycles is sufficiently small.
7. Restore $\boldsymbol{B}$ to the original coordinates: $\boldsymbol{T} = \boldsymbol{B} + \overline{\boldsymbol{v}} - sR\,\overline{\boldsymbol{v}}$. Although it appears that you could choose $\boldsymbol{B} = \overline{\boldsymbol{u}}$ and $\boldsymbol{T} = \boldsymbol{B} - sR\,\overline{\boldsymbol{v}}$, this is not correct because the evaluation of $F(R, \boldsymbol{B}, s)$ relies on $\boldsymbol{B}$ to be the difference of the averages.
8. Outputs: R, $\boldsymbol{T}$, and s.

The cycle through the Euler angles consists of the following. The steps for updating $(\sin a_0, \cos a_0)$ are:

1. Compute $\boldsymbol{r}_i = R_1(a_1)R_2(a_2)\boldsymbol{v}_i$ for all i.
2. Solve equation (12) for $(\sin a_0, \cos a_0)$.
3. Compute $R_0(a_0)$ using equation (10). The choice of sign is irrelevant. It might be negative initially, but on convergence, it should be positive. In worst case, if the final result has a negative scale, replace s by $-s$ and R by $-R$.
4. Compute $R = R_0(a_0)R_1(a_1)R_2(a_2)$.
5. Compute s using equation (5).
6. Compute $F(R, \boldsymbol{B}, s)$ using equation (3).

The steps for updating $(\sin a_1, \cos a_1)$ are:

1. Compute $\boldsymbol{\ell}_i = R_0^{\mathsf{T}}(a_1)\boldsymbol{u}_i$ for all i.

2. Compute $\boldsymbol{r}_i = R_2(a_2)\boldsymbol{v}_i$ for all i.

3. Solve equation (17) for $(\sin a_1, \cos a_1)$. The choice of sign is irrelevant. It might be negative initially, but on convergence, it should be positive. In worst case, if the final result has a negative scale, replace s by $-s$ and R by $-R$.

4. Compute $R_1(a_1)$ using equation (14).

5. Compute $R = R_0(a_0)R_1(a_1)R_2(a_2)$.

6. Compute s using equation (5).

7. Compute $F(R, \boldsymbol{B}, s)$ using equation (3).

The steps for updating $(\sin a_2, \cos a_2)$ are:

1. Compute $\boldsymbol{\ell}_i = (R_0(a_0)R_1(a_1))^{\mathsf{T}}\boldsymbol{u}_i$ for all i.

2. Solve equation (21) for $(\sin a_2, \cos a_2)$. The choice of sign is irrelevant. It might be negative initially, but on convergence, it should be positive. In worst case, if the final result has a negative scale, replace s by $-s$ and R by $-R$.

3. Compute $R_2(a_2)$ using equation (18).

4. Compute $R = R_0(a_0)R_1(a_1)R_2(a_2)$.

5. Compute s using equation (5).

6. Compute $F(R, \boldsymbol{B}, s)$ using equation (3).

The implementation is in the Geometric Tools file

https://www.geometrictools.com/GTE/Mathematics/HelmertTransformation7.h

References

[1] G.A. Watson. Computing helmert transformations. *Journal of Computational and Applied Mathematics*, 197:384–394, February 2005.
https://core.ac.uk/reader/81174210.