Estimating Body Segment Motion by Tracking Markers

William C. Rose, James G. Richards

Department of Kinesiology and Applied Physiology, University of Delaware

Corresponding author:
William C. Rose
University of Delaware
541 S. College Ave.
Newark, DE 19716
rosewc@udel.edu
302-831-1064, 302-831-9636 (fax)

Submitted as a Short Communication

Key words: movement analysis, rotation matrix, singular value decomposition

Word count: 1544
Estimating Body Segment Motion by Tracking Markers

William C. Rose, James G. Richards

Abstract

The estimation of segment motion is an essential task in biomechanical research and practice. In the present study we explain the key equations necessary to implement the algorithm of Spoor and Veldpaus. We identify issues that can prevent successful use of the algorithm, and explain how to overcome these issues. We used real experimental data and simulated data to test and verify Matlab and Labview implementations of the algorithm. The instructions in this communication will facilitate implementation of the Spoor and Veldpaus algorithm successfully in Matlab, Labview, or other programming environments.

Introduction

We track segment motion by attaching 3 or more markers to the segment and tracking the markers during motion. A 3D camera system produces files with the 3D coordinates of each marker at successive time points. As the segment moves, we wish to estimate the rotation and translation of the segment relative to its initial position. In other words, we wish to estimate a 3x3 rotation matrix $R$ and the 3-element translation vector $v$.

Algorithms for estimating segment motion

We assume that 3D marker coordinates have been measured in a reference position and during motion. Let the positions of $n$ markers in the initial, or reference, position be represented by a matrix $a$:

$$
\mathbf{a} = \begin{bmatrix}
a_{x1} & a_{x2} & a_{x3} & a_{x4} \\
a_{y1} & a_{y2} & a_{y3} & a_{y4} \\
a_{z1} & a_{z2} & a_{z3} & a_{z4}
\end{bmatrix}
$$

(1)
(We have assumed \( n=4 \) in this example.) The reference marker positions may be determined experimentally by averaging multiple frames of data from a static pose. Let the observed positions of the markers, at a particular instant of motion, be given by a matrix \( p \):

\[
p = \begin{bmatrix} p_{x1} & p_{x2} & p_{x3} & p_{x4} \\
p_{y1} & p_{y2} & p_{y3} & p_{y4} \\
p_{z1} & p_{z2} & p_{z3} & p_{z4} \end{bmatrix}
\]

(2)

We wish to find the rotation matrix \( R \) and translation vector \( v \) which transform the initial markers, specified by \( a \), into the moved markers, \( p \):

\[
Ra + v \approx p
\]

(3)

A perfect fit will generally not be possible, due to measurement errors and because the markers are not a true “rigid body”. Therefore we look for a fit that is best in a least squares error sense. Spoor and Veldpaus (1980) proposed that “best fit” estimates for \( R \) and \( v \) can be obtained by the following procedure. Our notation differs slightly from theirs.

Let the vectors \( \bar{a} \) and \( \bar{p} \) be the centroids of the markers:

\[
\bar{a} = \frac{1}{n} \sum_{j=1}^{n} a_j = \begin{bmatrix} \bar{a}_x \\
\bar{a}_y \\
\bar{a}_z \end{bmatrix}, \quad \bar{p} = \frac{1}{n} \sum_{j=1}^{n} p_j = \begin{bmatrix} \bar{p}_x \\
\bar{p}_y \\
\bar{p}_z \end{bmatrix}
\]

(4)

where \( a_j \) and \( p_j \) are the \( j \)th columns of \( a \) and \( p \) respectively. Then compute 3x3 matrix \( M \):

\[
M = \frac{1}{n} ap^T \quad - \quad \bar{a} \bar{p}^T
\]

(5)

where \( \bar{a} \bar{p}^T \) is the outer product of vectors \( \bar{a} \) and \( \bar{p} \). Then compute 3x3 matrices \( V \) and \( D^2 \) by eigenvector analysis of \( M^T M \):

\[
M^T M = V D^2 V^T
\]

(6)

where the columns of \( V \) are the eigenvectors of \( M^T M \), and \( D^2 \) is a diagonal matrix whose diagonal elements, \( d_1^2, d_2^2, d_3^2 \) are the eigenvalues of \( M^T M \). The rotation matrix \( R \) which gives the best fit is
\[
R = \begin{bmatrix}
    \frac{m_1}{d_1} & \frac{m_2}{d_2} & \frac{m_1 \times m_2}{d_1 d_2}
\end{bmatrix} V^T
\]  

(7)

where \( m_1, m_2 \) are columns 1 and 2 of 3x3 matrix \( MV \); \( MV = \begin{bmatrix} m_1 & m_2 & m_3 \end{bmatrix} \), and \( d_1, d_2 \) are the positive square roots of eigenvalues \( d_1^2 \) and \( d_2^2 \). The natural choice of \( m_3/d_3 \) for the third column in equation 7 was rejected because \( d_3 \) approaches zero if the markers lie in a plane. Spoor and Veldpaus stated the assumptions that (1) the columns of \( V \) (eigenvectors of \( MTM \)) are normalized to have unit length, and that (2) the eigenvalues in \( D^3 \), and the corresponding columns of \( V \), are ordered from largest to smallest, i.e. \( d_1 > d_2 > d_3 \). The translation vector that gives the best fit is

\[
v = \bar{p} - R\bar{a}
\]  

(8)

This completes the description of the algorithm of Spoor and Veldpaus.

We wrote programs in Matlab and Labview to implement the equations above. We tested our programs with simulated marker data and with real experimental data. The results we obtained were not always acceptable. This led us to examine the assumptions of Spoor and Veldpaus.

The assumption of Spoor and Veldpaus that the eigenvalues and corresponding eigenvectors are ordered from largest to smallest is not met by the Labview and Matlab routines just mentioned. Those routines return eigenvalues and eigenvectors ordered from smallest to largest (\( d_1 < d_2 < d_3 \)). (The Labview routine only return eigenvalues ordered from smallest to largest if it is called with the symmetric matrix option, which is appropriate, since \( MTM \) is always symmetric. If it is not called with the symmetric matrix option, the eigenvalues are unsorted.) Therefore we replaced the first column of \( MV \) instead of the third, i.e. we used the following equation for \( R \), instead of equation 7:

\[
R = \begin{bmatrix}
    \frac{m_2 \times m_3}{d_2 d_3} & \frac{m_2}{d_2} & \frac{m_3}{d_3}
\end{bmatrix} V^T
\]  

(9)
Even with this change, our testing showed that our programs in Matlab and Labview did not always give acceptable results. Therefore we investigated a different way of implementing the algorithm of Spoor and Veldpaus. Specifically, we use singular value decomposition (SVD) to find orthogonal matrices $U$ and $V_{svd}$ and diagonal matrix $S$ that satisfy

$$M = USV_{svd}^T$$

(10)

It is a result from linear algebra theory that $V_{svd}$ from equation 10 should equal $V$ from equation 6, and that the diagonal elements of $S$ should equal the square roots of the diagonal elements of $D^2$ in equation 6. Therefore we compute

$$R_{svd} = \begin{bmatrix} \frac{m_1}{s_1} & \frac{m_2}{s_2} & \frac{m_1 \times m_2}{s_1 s_2} \end{bmatrix} V^T$$

(11)

where $m_1$, $m_2$ are columns 1 and 2 of matrix $MV_{svd}$: $MV_{svd} = [m_1 \quad m_2 \quad m_3]$, and $s_1$, $s_2$ are the two largest singular values, from the diagonal of $S$. Both Matlab (svd.m) and Labview (Real SVD Decomposition.vi) return singular values from largest to smallest. Therefore we do not need to consider an equation analogous to equation 9. A Labview program using equations 10 and 11 gave acceptable results, but a Matlab program using the same equations did not. As we already mentioned, Matlab and Labview programs using equations 6 and 9 did not give acceptable results. Theoretically, all the programs should have worked. A closer look at the paper of Spoor and Veldpaus shows an unstated assumption that the eigenvectors follow the right hand rule, i.e. $v_1 \times v_2 = +v_3$, where $v_1$, $v_2$, $v_3$, are the columns of $V$ or of $V_{svd}$. Testing shows that this is not a valid assumption for the eigenvectors returned by Matlab’s or Labview’s eigenvector routines (eig.m and Eigenvalues and Vectors.vi), nor is it valid for the vectors returned by Matlab’s SVD (svd.m). For each of those routines, there is about a 50-50 chance that the eigenvectors will form a right-handed set ($v_1 \times v_2 = +v_3$) or left-handed set ($v_1 \times v_2 = -v_3$). The eigenvectors returned by Labview’s SVD always form a right handed set. This explains why the
Labview program using SVD worked reliably but the other programs did not. This problem can be ameliorated by checking the handedness of the vectors $v_1$, $v_2$, and $v_3$, and adjusting the sign of the replaced column accordingly.

**Summary of the recommended algorithm**

We now summarize the steps necessary to estimate the rotation matrix $R$ and translation vector $v$ of equation 3. We give two alternatives: one using eigenvalue decomposition and one using SVD. Both alternatives use the approach of Spoor and Veldpaus, with adjustments to account for the ordering of eigenvalues or singular values, and for the handedness of the vectors of $V$.

Use equations 1, 2, 4, and 5 to compute $M$. If eigenvalue decomposition is utilized, compute $V$ and $D^2$ according to equation 6. (We assume the eigenvalues are sorted in ascending order, which is true for Matlab’s eig.m, for LAPACK’s dsyev, and for Labview’s Eigenvalues and Vectors.vi, if it is called with the symmetric matrix option, as it should be. For other programming environments, the user should verify the validity of this assumption.) Compute $d_2$ and $d_3$, the positive square roots of eigenvalues $d_2^2$ and $d_3^2$, which are diagonal elements of $D^2$.

Check whether the columns of $V$ obey $v_2 \times v_3 = +v_1$ or $v_2 \times v_3 = -v_1$. If the first equation is true, compute $R$ using equation 9 above. If the second equation is true, compute $R$ using

$$R = \left[ \begin{array}{ccc} -\frac{m_2 \times m_3}{d_2 d_3} & \frac{m_2}{d_2} & \frac{m_3}{d_3} \end{array} \right] V^T$$

(12)

Then compute the translation vector $v$ using equation 8.

If SVD is utilized, compute $M$ as before, using equations 1, 2, 4, and 5. Then compute $S$ and $V_{svd}$ according to equation 10. (We assume the singular values are sorted in descending order, which is true for Matlab’s and Labview’s singular value routines, and for dgesvd in LAPACK. For other programming environments, the user should verify the validity of this assumption.) Check whether the columns of $V_{svd}$ obey $v_1 \times v_2 = +v_3$ or $v_1 \times v_2 = -v_3$. (For Labview, the first
equation is always true.) If the first equation is true, compute $R_{svd}$ using equation 11 above. If the second equation is true, compute $R_{svd}$ using

$$R_{svd} = \frac{m_1}{s_1} \frac{m_2}{s_2} - \frac{m_1 \times m_2}{s_1 s_2} \mathbf{v}^T$$

(13)

where $s_1$ and $s_2$ are two largest singular values, on the diagonal of $\mathbf{S}$. Then compute the translation vector $\mathbf{v}$ using equation 8.

**Results**

We wrote four programs to test the recommendations above: eigenvalue decomposition in Matlab and in Labview, and SVD in Labview and Matlab. We tested the programs with simulated data and with experimental data. The simulated data included marker sets of three to eight markers, translated and rotated by various amounts, with pseudorandom noise (standard deviation of 1 mm) added to simulate measurement noise and non-rigidity of the marker set. The experimental data included a plastic shell with four markers, taped to the thigh during walking, and an “artificial segment” made of a PVC pipe with four planar markers and four non-planar markers, which was translated and rotated.

All four programs gave identical results for all simulated and experimental test data. For the simulated data, the “true” rotation and translation was known, and could be compared to the estimated translation and rotation. Figures 1 and 2 show an example with four markers. This comparison and others showed that all four programs correctly estimated the rotation and translation.

**Discussion**

The published algorithm of Spoor and Veldpaus is based on stated and unstated assumptions. The algorithm works well, if the ordering of eigenvalues or singular values, and the handedness
of the columns of $V$, is determined and properly taken in to account, in the way we have explained.

The algorithm may be implemented using eigenvalue decomposition of $M^TM$ or singular value decomposition of $M$. Equivalent results are obtained with either method, and with both Matlab and Labview.

Conflict of interest statement

The authors have no conflict of interest.

Acknowledgements

This study was supported by the University of Delaware.

References

Figure 1. True (simulated) rotation angle and the estimated rotation angle from each of four programs. Rotation was about X, then Y, then Z. Lines show true rotation angles (solid, $\theta_x$; dashed, $\theta_y$; dotted, $\theta_z$). Symbols show estimates: $\circ$, eigenvalue decomposition in Matlab; $\times$, SVD in Matlab; $\Delta$, eigenvalue decomposition in Labview; $+$, SVD in Labview. Estimates are shown for every tenth frame for clarity.
Figure 2. True (simulated) translation and the estimated translation from each of four programs. Lines show true translations (solid, $v_x$; dashed, $v_y$; dotted, $v_z$). Symbols as in Figure 1; estimates are shown for every tenth frame for clarity.