Point Based Rigid Registration

Ziv Yaniv

School of Engineering and Computer Science
The Hebrew University, Jerusalem, Israel.

The subject of this lecture is point based rigid registration, also known in the vision community as the absolute orientation problem. Registration is the task of finding a transformation from one coordinate system to another so that all features that appear in both data sets are aligned with each other. In this lecture we limit ourselves to rigid transformations with input given as points.

Problem Definition

Given the coordinates of points measured in two cartesian coordinate systems find the rigid transformation between the two systems.

This summary is divided into five sections: (1) 3D point data acquisition using a tracked pointer. (2) Registration using three points. (3) An iterative approach to registration. (4) A closed form solution to the registration problem. (5) How to analyze your results.

1 3D Point Acquisition

As stated in the problem definition the input we receive is point data measured in two coordinate systems. This section describes a 3D point acquisition setup using a pointing device and a tracking unit. The pointing device is equipped with a tracking plate which defines a local coordinate system which is relative to the tracking unit coordinate system, Figure 1.

The points are acquired by touching the required locations with the pointer tip. If the translation between the pointer coordinate system origin and tip is known we can compute the 3D coordinates of the object. This translation is computed by placing the pointer tip in a fixed location and rotating and translating the pointer while the tip remains stationary. This setting is shown in Figure 2.

Every pointer transformation $F(pointer)_i = [R_i, t_i]$ yields the following three equations in six unknowns:

$$R_i t_{tip} + t_i = t_{post}$$

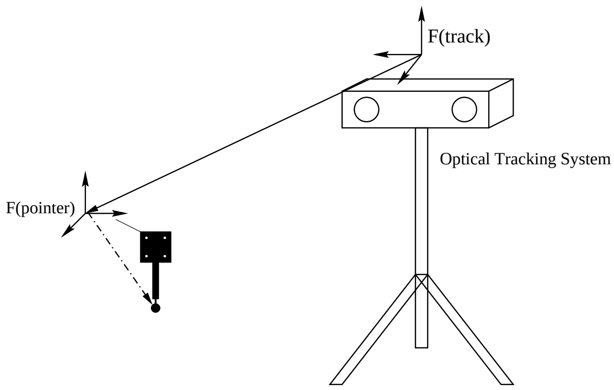

Figure 1: Pointer coordinate system.

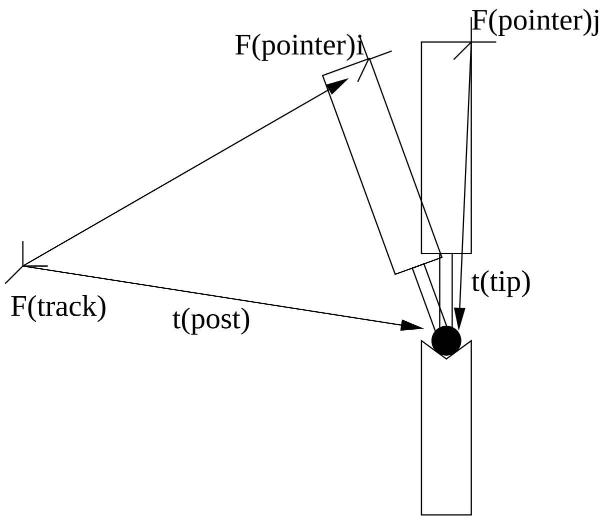

Figure 2: Pointer calibration setup.

We now acquire several transformations and solve a least squares problem:

$$\begin{bmatrix} R_0 | -I \\ \vdots \\ R_n | -I \end{bmatrix} \begin{bmatrix} t_{tip} \\ t_{post} \end{bmatrix} = \begin{bmatrix} -t_0 \\ \vdots \\ -t_n \end{bmatrix}$$

2 Three Point Registration

Given the coordinates of three points in two coordinate systems, let us call them "left" and "right". The points are p_{l1}, p_{l2}, p_{l3} and p_{r1}, p_{r2}, p_{r3}. We find the transformation between coordinate systems with the following construction:

1. Choose one of the points to be the origin, let us say p_1.
2. Construct the x axis:
$$x = \frac{p_2 - p_1}{|p2 - p1|}$$
3. Construct the y axis:
$$y = (p_3 - p_1) - [(p_3 - p_1) \cdot x]x$$
$$y = y/|y|$$
4. Construct the z axis:
$$z = x \times y$$
5. Build the rotation matrices for both point sets:
$$R_l = [x_l, y_l, z_l] \quad R_r = [x_r, y_r, z_r]$$
6. The rotation between the "right" coordinate system to the "left" is given by:
$$R = R_l R_r^t$$
7. The translation between the "right" coordinate system to the "left" is given by:
$$t = p_{l1} - Rp_{r1}$$

The problem with this formulation is that we assume that the data is given with infinite precision. This is not the case in real life, so we must resort to least squares techniques.

3 Iterative Registration

Given the coordinates of a set of points, more than three, in two coordinate systems we want to compute the rigid transformation between coordinate systems. Let us call the coordinate systems "left" and "right".
We formulate the problem as follows:
Given two points p_l and p_r we get the following expression for the distance between them:
$$d^2(p_l, p_r) = (p_l - p_r)^t(p_l - p_r)$$

The expression $p'_l = Rot(\omega)p_l + t$ provides the location of p_l in the "right" coordinate system, where ω is a rotation and t a translation with respect to the global coordinate system. The distance between the new point p'_l and p_r is:

$$d^2(p'_l, p_r) = (p'_l - p_r)^t (p'_l - p_r)$$

Let ω be a three vector $(\omega_1, \omega_2, \omega_3)$ representing a general fixed axis rotation with respect to the x, y, z axes. This defines the following rotation matrix, where c is cosine and s sine:

$$\begin{aligned} R_{xyz} &= R_z(\omega_3)R_y(\omega_2)R_x(\omega_1) \\ &= \begin{bmatrix} c(\omega_3) & -s(\omega_3) & 0 \\ s(\omega_3) & c(\omega_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c(\omega_2) & 0 & s(\omega_2) \\ 0 & 1 & 0 \\ -s(\omega_2) & 0 & c(\omega_2) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\omega_1) & -s(\omega_1) \\ 0 & s(\omega_1) & c(\omega_1) \end{bmatrix} \quad (1) \\ &= \begin{bmatrix} c(\omega_3)c(\omega_2) & c(\omega_3)s(\omega_2)s(\omega_1) - s(\omega_3)c(\omega_1) & c(\omega_3)s(\omega_2)c(\omega_1) + s(\omega_3)s(\omega_1) \\ s(\omega_3)c(\omega_2) & s(\omega_3)s(\omega_2)s(\omega_1) + c(\omega_3)c(\omega_1) & s(\omega_3)s(\omega_2)c(\omega_1) - c(\omega_3)s(\omega_1) \\ -s(\omega_2) & c(\omega_2)s(\omega_1) & c(\omega_2)c(\omega_1) \end{bmatrix} \end{aligned}$$

For small rotations, we apply the following standard approximation for R_{xyz}: $\cos(\omega_i) \simeq 1$, $\sin(\omega_i) \simeq \omega_i$, $\sin(\omega_i)\sin(\omega_j) = 0$,
this reduces 1 to:

$$R_{xyz} \simeq \begin{bmatrix} 1 & -\omega_3 & \omega_2 \\ \omega_3 & 1 & -\omega_1 \\ -\omega_2 & \omega_1 & 1 \end{bmatrix}$$

Also, the approximate position of the transformed point is:

$$p'_l = Rot(\omega)p_l + t \simeq (\omega \times p_l) + p_l + t$$

Given this approximation for R_{xyz} we obtain the following distance equation for $d^2(p'_l, p_r)$:

$$d^2(p'_l, p_r) \simeq (\omega \times p_l + p_l + t - p_r)^t(\omega \times p_l + p_l + t - p_r)$$

Let

$$\delta_i = p_r - p_l$$

and

$$P_{l,i} = \begin{bmatrix} 0 & p_{l3} & -p_{l2} \\ -p_{l3} & 0 & p_{l1} \\ p_{l2} & -p_{l1} & 0 \end{bmatrix}$$

which yields,

$$d^2(p'_l, p_r) \simeq ([P_{l,i}|I] \begin{pmatrix} \omega \\ t \end{pmatrix} - \delta_i)^t([P_{l,i}|I] \begin{pmatrix} \omega \\ t \end{pmatrix} - \delta_i) \quad (2)$$

Equation 2 is an approximation to the value of the distance which becomes more accurate as the distance is reduced.

We now formulate the problem by minimizing the cumulative distance between all points. We want to minimize the sum of the square distances of all p_l's :

$$\begin{aligned}
\Delta^2(\omega, t) &= \textstyle\sum_i d^2(p_l', p_r) \\
&= \textstyle\sum_i([P_{l,i}|I]\begin{pmatrix}\omega \\ t\end{pmatrix} - \delta_i)^t([P_{l,i}|I]\begin{pmatrix}\omega \\ t\end{pmatrix} - \delta_i) \\
&= \left(\begin{bmatrix} P_{l,1}|I \\ \vdots \\ P_{l,n}|I \end{bmatrix}\begin{pmatrix}\omega \\ t\end{pmatrix} - \begin{bmatrix}\delta_1 \\ \vdots \\ \delta_n\end{bmatrix}\right)^T \left(\begin{bmatrix} P_{l,1}|I \\ \vdots \\ P_{l,n}|I \end{bmatrix}\begin{pmatrix}\omega \\ t\end{pmatrix} - \begin{bmatrix}\delta_1 \\ \vdots \\ \delta_n\end{bmatrix}\right) \\
&= \|Ux - z\|^2
\end{aligned}$$

Finally, we obtain the minimization problem:

$$min\Delta^2(x) \equiv min\|Ux - z\|$$

Note that because of the angular approximation, the obtained transformation is generally not the actual required transformation. By repeating this process, the error is reduced below a desired threshold. Unfortunately these iterations do not guarantee that the algorithm will converge to the global minimum. Usually we initialize the iterations with a transformation which is "close" to the global minimum. The "closeness" is in terms of the angles which compose the transformation.

4 Closed Form Registration

This section is based on [3]. Given the coordinates of a set of points, more than three, in two coordinate systems we want to compute the rigid transformation between coordinate systems. Let us call the coordinate systems "left" and "right". We formulate the problem as follows:
Given two points p_l, p_r and the computed transformation $[R, t]$ the residual error between the actual and computed locations is given by:

$$e_i = p_r - R(p_l) - t$$

We want to minimize the sum of squares of these errors:

$$\sum_{i=1}^{n} ||e_i||^2$$

It turns out to be useful if we refer all measurements to the centroids which are given by:

$$\mu_l = \frac{1}{n}\sum_{i=1}^{n} p_{l,i} \qquad \mu_r = \frac{1}{n}\sum_{i=1}^{n} p_{r,i}$$

We now denote the new coordinates by

$$p'_{l,i} = p_{l,i} - \mu_l \qquad p'_{r,i} = p_{r,i} - \mu_r$$

Using the new coordinates we rewrite the error term as:

$$e_i = p'_{r,i} - Rp'_{l,i} - t'$$

where

$$t' = t - \mu_r + R\mu_l$$

The sum of square errors becomes

$$\begin{aligned} \sum_{i=1}^{n} ||e_i||^2 &= \sum_{i=1}^{n} ||p'_{r,i} - Rp'_{l,i} - t'||^2 \\ &= \sum_{i=1}^{n} ||p'_{r,i} - Rp'_{l,i}||^2 - 2t' \cdot \sum_{i=1}^{n} [p'_{r,i} - Rp'_{l,i}] + n||t'||^2 \end{aligned} \tag{3}$$

Noting that $\sum_{i=1}^{n} p'_{r,i} = \sum_{i=1}^{n} p'_{l,i} = 0$ it immediately follows that the middle summation expression in Equation 3 is zero. The last term in Equation 3 is non-negative and the first term does not depend on t' which means that the error is minimized when:

$$\begin{aligned} t' &= 0 \\ &\Downarrow \\ t &= \mu_r - R\mu_l \end{aligned}$$

We have reduced the minimization task to the minimization of the first term in Equation 3:

$$\sum_{i=1}^{n} ||p'_{r,i} - Rp'_{l,i}||^2 = \sum_{i=1}^{n} ||p'_{r,i}||^2 - 2\sum_{i=1}^{n} p'_{r,i} \cdot Rp'_{l,i} + \sum_{i=1}^{n} ||Rp'_{l,i}||^2 \tag{4}$$

The first and third terms of Equation 4 are constants independent of R (rotations preserve the vector norm). To minimize the error we only need to maximize the second term (The geomtric intuition for this is given in Figure 3).

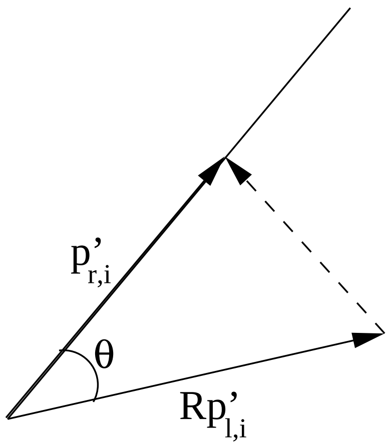

Figure 3: Maixmizing the sum $\sum_{i=1}^{n} p'_{r,i} \cdot Rp'_{l,i}$ is equivalent to maximizing $\sum_{i=1}^{n} |p'_{r,i}||Rp'_{l,i}| \cos\theta$. This sum is maximal when $\cos\theta = 1, \quad \theta = 0$. Geometrically we compute the rotation which minimizes the angle between the two vectors.

We choose to use unit quaternions as our rotational operator and we maximize the following sum:

$$\sum_{i=1}^{n} (q * p'_{l,i} * \overline{q}) \cdot p'_{r,i}$$

Which can be rewritten as:

$$\sum_{i=1}^{n} (q * p'_{l,i}) \cdot (p'_{r,i} * q)$$

Again we rewrite the sum using matrix notation for the quaternion multiplication:

$$p'_{r,i} * q = \begin{bmatrix} 0 & -x'_{r,i} & -y'_{r,i} & -z'_{r,i} \\ x'_{r,i} & 0 & -z'_{r,i} & y'_{r,i} \\ y'_{r,i} & z'_{r,i} & 0 & -x'_{r,i} \\ z'_{r,i} & -y'_{r,i} & x'_{r,i} & 0 \end{bmatrix} q = \Re_{r,i} q$$

and

$$q * p'_{l,i} = \begin{bmatrix} 0 & -x'_{l,i} & -y'_{l,i} & -z'_{l,i} \\ x'_{l,i} & 0 & z'_{l,i} & -y'_{l,i} \\ y'_{l,i} & -z'_{l,i} & 0 & x'_{l,i} \\ z'_{l,i} & y'_{l,i} & -x'_{l,i} & 0 \end{bmatrix} q = \overline{\Re_{l,i}} q$$

giving the following summation:

$$\sum_{i=1}^{n} (\overline{\Re_{l,i}} q) \cdot (\Re_{r,i} q)$$

$$\Downarrow$$

$$\sum_{i=1}^{n} q^T \overline{\Re_{l,i}}^T \Re_{r,i} q$$

$$\Downarrow$$

$$q^T (\sum_{i=1}^{n} \overline{\Re_{l,i}}^T \Re_{r,i}) q$$

$$\Downarrow$$
$$q^T(\sum_{i=1}^n N_i)q$$
$$\Downarrow$$
$$q^T N q$$

The matrix N is symmetric as it is a sum of symmetric matrices. The vector q which maximizes $q^T Nq$ is the eigenvector corresponding to the most positive eigenvalue of the matrix N (Theorem B.3).

Constructing the matrix N:

We introduce the following matrix:

$$\begin{aligned} M &= \sum_{i=1}^{n} p'_{l,i} p'^T_{r,i} \\ &= \sum_{i=1}^{n} [p_{l,i} p^T_{r,i}] - \mu_l \mu_r^T \\ &= \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix} \end{aligned}$$

where

$$S_{xx} = \sum_{i=1}^{n} x'_{l,i} x'_{r,i} \qquad S_{xy} = \sum_{i=1}^{n} x'_{l,i} y'_{r,i}$$

and so on. This matrix contains all the information needed to construct the matrix N.

$$N = \begin{bmatrix} trace(M) & \Delta^T \\ \Delta & M + M^t - trace(M) I_3 \end{bmatrix}$$

where

$$\Delta = \begin{bmatrix} (M - M^T)_{23} \\ (M - M^T)_{31} \\ (M - M^T)_{12} \end{bmatrix}$$

5 Result Analysis

This section discusses the issue of comparing the results of the registration algorithm with a known ground truth [2, 4, 6]. Let us assume that we are using matrices for rotation representation. Looking at the matrix entries does not give an indication whether we are close or not to our ground truth.

Example:

Given the ground truth matrix representing a fixed axis rotation

$$R_{xyz}(10^o, 20^o, 30^o) = \begin{bmatrix} 0.8138 & -0.4410 & 0.3785 \\ 0.4698 & 0.8826 & 0.0180 \\ -0.3420 & 0.1632 & 0.9254 \end{bmatrix}$$

We have the matrix:

$$\begin{bmatrix} 0.8138 & -0.4063 & 0.4155 \\ 0.4698 & 0.8808 & -0.0590 \\ -0.3420 & 0.2432 & 0.9077 \end{bmatrix}$$

The entries are different but by how much? Does this give any indication about the difference in rotations? The answer is of course no (the second matrix is $R_{xyz}(15^o, 20^o, 30^o)$).

A more meaningful comparison is to extract the rotation angles and the translation and compare these vectors $(\omega_x, \omega_y, \omega_z, dx, dy, dz)$. Unfortunately this measure is mostly qualitative and not quantitative. The reason for this is that the errors in the registration results do not correspond linearly with errors in Euclidean space. This is shown in Figure 4 where for the same error in rotation we have different errors in Euclidean space.

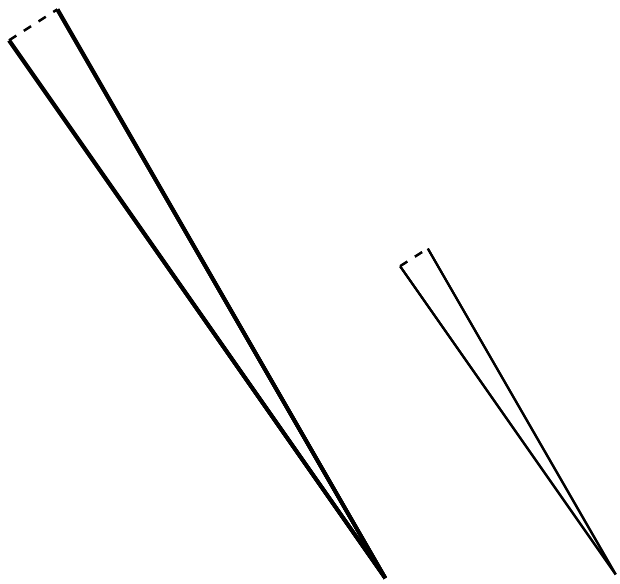

Figure 4: Given a 5^o error we have different errors in Euclidean space. The farther from the rotation center the greater the error.

The most meaningful comparison is the distance between expected and actual point locations. Usually we are aligning two data sets from which we sample points and use these as input for our algorithm. It is natural that these points will be the ones best aligned but we should remember that our goal is to align all the data and not just the sampled points. We view each point set as consisting of two sub sets, *target and fiducial* points. Target points are the actual points that we want to align. Fiducial points are the points used as input for the registration. If both sub sets are the same, fiducials are the targets, then our discussion up to this point is complete.

In medical applications this is not always the case. Fiducial points are usually identifiable anatomical landmarks or implanted markers while the target area/points are not always accessible or directly identifiable. We now consider three types of errors fiducial registration error (FRE) due to fiducial localization error (FLE) and the target registration error (TRE).

For a statistical derivation of TRE as a function of FRE and FLE the reader is referred to [6]. We will now look briefly into the relationship between FRE and TRE, why FRE is a bad estimator for TRE. We show that it is possible to have a small/large FRE and with a corresponding large/small TRE.

We first look at the case of a small FRE with a large TRE.
Assume an FLE of $+X$mm in the x direction for all fiducials (a bias). The registration will yield a small FRE but the TRE will be Xmm. This is a simple example of the fact that a bias added to the fiducials will not be detected when our error measure is FRE. For the general case consider the following: both point sets, fiducial P_f and target P_t, have undergone a transformation T_1 we add a bias, another transformation T_2, to P_f. According to our measurements the result of the registration will be $T = T_1T_2$ while the correct result should be T_1.

We now look at the case of a large FRE with a small TRE.
Assume an FLE of $+X$mm in the x direction for half of the fiducials and an FLE of $-X$mm in the x direction for the other half. The registration will yield the correct transformation, as the errors have canceled each other out. It is easily seen that the FRE will be Xmm while the TRE will be zero/small.

Finally, always keep in mind that the goal of registration is to align the target points, so error analysis should be meaningful with regard to this goal.

A Closed Form Solution in 2D

In section 4 we developed a closed form solution for two sets of 3D points. In this appendix we discuss the degenerate 2D case, all points in both data sets are coplanar. This analysis is of interest for image registration using points and yields a simple solution to the rigid registration problem [3].

Up to Equation 4 we did not assume anything about the dimensionality of the points and a specific rotational operator. We concluded that the translation is given by:

$$t = \mu_r - R\mu_l \qquad \text{and} \quad \mu_l = \frac{1}{n}\sum_{i=1}^{n} p_{l,i} \quad , \quad \mu_r = \frac{1}{n}\sum_{i=1}^{n} p_{r,i}$$

Equation 4 yielded the following maximization problem:

$$\max_R \left(\sum_{i=1}^{n} p'_{r,i} \cdot (Rp'_{l,i}) \right)$$

Given that both point sets are coplanar we seek an in plane rotation as our solution. Rotating the left point set in the plane by θ reduces the angles α_i by theta:

$$\max_\theta \left(f = \sum_{i=1}^{n} |p'_{r,i}||p'_{l,i}|cos(\alpha_i - \theta) \right)$$

To find the extrema of f we diffrentiate with respect to θ:

$$\begin{aligned}
\frac{df}{d\theta} &= \sum_{i=1}^{n} |p'_{r,i}||p'_{l,i}|sin(\alpha_i - \theta) &(5)\\
&= \sum_{i=1}^{n} |p'_{r,i}||p'_{l,i}| \left[\cos(\alpha_i)\sin(-\theta) + \sin(\alpha_i)\cos(-\theta)\right] &(6)\\
&= \cos(\theta)\sum_{i=1}^{n} |p'_{r,i}||p'_{l,i}| \sin(\alpha_i) - \sin(\theta)\sum_{i=1}^{n} |p'_{r,i}||p'_{l,i}| \cos(\alpha_i) &(7)\\
&= \cos(\theta)S - \sin(\theta)C &(8)
\end{aligned}$$

We got from Equation 5 to Equation 6 using the following identity:

$$\sin(\alpha + \beta) = \cos(\alpha)\sin(\beta) + \sin(\alpha)\cos(\beta)$$

The coefficients C and S are computed from the original data as follows:

$$\begin{aligned}
C &= \textstyle\sum_{i=1}^{n} |p'_{r,i}||p'_{l,i}| \cos(\alpha_i) = \sum_{i=1}^{n} p'_{r,i} \cdot p'_{l,i}\\
S &= \textstyle\sum_{i=1}^{n} |p'_{r,i}||p'_{l,i}| \sin(\alpha_i)\\
&= \textstyle\sum_{i=1}^{n} |[p'_{r,i};0]||[p'_{l,i};0]| \sin(\alpha_i)\\
&= \left(\textstyle\sum_{i=1}^{n} [p'_{r,i};0] \times [p'_{l,i};0]\right) \cdot [0,0,1]\\
&= \textstyle\sum_{i=1}^{n} (p'_{r,i}(x)p'_{l,i}(y) - p'_{r,i}(y)p'_{l,i}(x))
\end{aligned}$$

To find the extremum values of f Equation 8 is equated to zero and then squared yielding:

$$\begin{gathered}
\cos(\theta)^2 S^2 = \sin(\theta)^2 C^2\\
\Downarrow\\
\sin(\theta)^2 C^2 - (1 - \sin(\theta)^2) S^2 = 0\\
\Downarrow\\
\sin(\theta)^2 (C^2 + S^2) - S^2 = 0
\end{gathered} \qquad (9)$$

Solving the quadratic equation 9 yields:

$$\sin(\theta) = \pm\frac{S}{\sqrt{C^2 + S^2}}$$

similarly we get:

$$\cos(\theta) = \pm\frac{C}{\sqrt{C^2 + S^2}}$$

Substituting these solutions into Equation 8 we get:

$$\frac{SC}{\pm\sqrt{C^2 + S^2}} - \frac{SC}{\pm\sqrt{C^2 + S^2}} = 0$$

We now see that the extrema are achieved for $\pm\sqrt{C^2 + S^2}$ without regard to the sign of C or S. We can arbitrarily choose the pluses to get the maximum.

B Symmetric Matrices

Theorem B.1 *The eigenvalues of a real symmetric matrix are real.*

Proof Given the symmetric real matrix A we have:

$$Ax = \lambda x \tag{10}$$

Where λ is an eigenvalue and x the corresponding eigenvector. Until we prove the theorem we must assume that λ might be a complex number ($\lambda = a + ib$) and x might contain components which are complex too. Remembering that $\overline{\lambda x} = \overline{\lambda}\overline{x}$ and that $A = \overline{A} = A^T$ we take the conjugates of equation 10:
$\overline{Ax} = \overline{\lambda x}$ leads to $A\overline{x} = \overline{\lambda}\overline{x}$. Transpose this to get $\overline{x}^T A = \overline{x}^T\overline{\lambda}$ Now taking the dot product of the first equation with $\overline{x}$ and the last equation with x we get:
$\overline{x}^T Ax = \overline{x}^T\lambda x$ and $\overline{x}^T Ax = \overline{x}^T\overline{\lambda}x$
Which gives us: $\overline{x}^T\lambda x = \overline{x}^T\overline{\lambda}x$
Therefore $\lambda = \overline{\lambda}$ proving that λ is real ($a + bi = a - bi$ so the complex coefficient is equal to zero).

Theorem B.2 *The eigenvectors of a real symmetric matrix which correspond to different* [1] *eigenvalues are perpendicular.*

Proof Let λ_1 and λ_2 be two different eigenvalues and x_1 and x_2 the corresponding eigenvectors. This gives us the following two equations:

$$Ax_1 = \lambda_1 x_1$$
$$Ax_2 = \lambda_2 x_2$$

Taking the dot product with x_2 we get:

$$(\lambda_1 x_1)^T x_2 = (Ax_1)^T x_2 = x_1^T A^T x_2 = x_1^T A x_2 = x_1^T \lambda_2 x_2$$

The left side is $x_1^T\lambda_1 x_2$ and the right side is $x_1^T\lambda_2 x_2$. Since $\lambda_1 \neq \lambda_2$ this proves that $x_1^T x_2 = 0$. The eigenvectors are perpendicular.

Theorem B.3 *Let A be a real nxn symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ and corresponding orthonormal eigenvectors $x_1, x_2, \ldots, x_n$, as proven in B.2. Define $X_k = (x_1, x_2, \ldots, x_k)$ $(k = 1, 2, \ldots, n-1)$ and $X = (x_1, x_2, \ldots, x_n)$. Then if we assume that $\alpha \neq 0$, we have the following:*

[1] Proof that *all eigenvectors of a real symmetric matrix are orthogonal to each other can be found in [1]*

1.

$$\sup_{\alpha}\left\{\frac{\alpha^T A\alpha}{\alpha^T\alpha}\right\}=\lambda_1$$

and the supremum is attained if $\alpha = x_1$.

2.

$$\sup_{X_k^T\alpha=0}\left\{\frac{\alpha^T A\alpha}{\alpha^T\alpha}\right\}=\lambda_{k+1}$$

and the supreumum is attained if $\alpha = x_{k+1}$.

3.

$$\inf_{\alpha}\left\{\frac{\alpha^T A\alpha}{\alpha^T\alpha}\right\}=\lambda_n$$

and the infimum is attained if $\alpha = x_n$.

4. If $X_{n-k} = (x_{n-k+1}, x_{n-k+2}, \dots, x_n)$ *then*

$$\inf_{X_{n-k}^T\alpha=0}\left\{\frac{\alpha^T A\alpha}{\alpha^T\alpha}\right\}=\lambda_{n-k}$$

and the infimum is attained if $\alpha = x_{n-k}$.

Proof 1. Let $\alpha = Xy = y_1x_1 + y_2x_2 + \cdots + y_nx_n$ and

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}.$$

Then

$$\frac{\alpha^T A\alpha}{\alpha^T\alpha} = \frac{y^TX^TAXy}{y^Ty} = \frac{y^TX^TX\Lambda y}{y^Ty} = \frac{(\sum_i \lambda_i y_i^2)}{y^Ty} \le \frac{\lambda_1 y^Ty}{y^Ty} = \lambda_1$$

with equality when $y_1 = 1, y_2 = y_3 = \cdots = y_n = 0$, that is, when $\alpha = x_1$.

2. If $\alpha \perp x_1, x_2, \dots, x_k$, then $y_1 = y_2 = \cdots = y_k = 0$. The result then follows with the same argument as 1.

3. Same proof as 1 but with the inequality reversed.

4. Same proof as 2 but with inequality reversed.

B.1 Matrices of the form AA^T

Theorem B.4 *Matrices of the form AA^T have the following properties:*

1. *They are positive definite.*
2. *They have positive eigenvalues.*
3. *The Singular Value Decomposition of A yields the eigenvalues and eigenvectors of AA^T.*

Proof

1. A matrix B is positive definite if:

$$\forall x,\ x \neq 0 \quad x^T B x > 0$$

   Given a matrix B of the form AA^T we have:

$$x^T B x = x^T A A^T x = (A^T x)^T (A^T x) = \|A^T x\|^2 > 0$$

2. Given a matrix B of the form AA^T with eigenvalue λ and corresponding eigenvector x we have:

$$Bx = \lambda x$$

   Premultiplying by x^T we get:

$$x^T B x = \lambda x^T x = \lambda \|x\|^2$$

   As B is positive definite we have:

$$\begin{array}{rcl} \lambda \|x\|^2 & > & 0 \\ & \Downarrow & \\ \lambda & > & 0 \end{array}$$

3. The Singular Value Decomposition of the matrix A is given by

$$A_{m\times n} = U_{m\times n} \Sigma_{n\times n} V^T_{n\times n}$$

   where the columns of U are an orthonormal basis for the column space of A, the rows of V are an orthonormal basis for the row space of A and Σ is a diagonal matrix. We now look at the matrix AA^T:

$$AA^T = (U\Sigma V^T)(V\Sigma U^T) = U\Sigma^2 U^T$$

   Postmultipying this equation by U yields the following equation:

$$(AA^T)U = U\Sigma^2$$

   The eigenvectors of AA^T are the columns of U with corrosponding eigenvalues in Σ.

References

[1] Golub G.H., Van Loan C.F., *Matrix Computations, third edition*, Jhons Hopkins University Press,1996.

[2] Hajnal J.V., Hill D.L.G., Hawkes D.J, editors, *Medical Image Registration*, ,chapter 6, pages 117–139, CRC Press, 2001.

[3] Horn B.K.P., "Closed-form solution of absolute orientation using unit quaternions", Journal of the Optical Society of America, Vol. 4(4), pp 629–642, 1987.

[4] LaRose D., Bayouth J., Kanade T., "Evaluating 2D/3D registration accuracy", Computer Assisted Radiology and Surgery (CARS), pp.147–152, 2000.

[5] Strang G., *Introduction To Linear Algebra*, Wellesley-Cambridge Press, 1993.

[6] West J.B., "Predicting Error in Point-Based Registration", Phd. thesis, Graduate School of Vanderbilt University, Nashville Tennessee, 2000.