1 Derivation of Point-to-Plane Minimization

Consider the Chen-Medioni (point-to-plane) framework for ICP. Assume we have a collection of points (p_i, q_i) with normals n_i. We want to determine the optimal rotation and translation to be applied to the first collection of points (i.e., the p_i) to bring them into alignment with the second (i.e., the q_i). Thus, we want to minimize the alignment error

$$\mathcal{E} = \sum_i [(Rp_i + t - q_i) \cdot n_i]^2 \tag{1}$$

with respect to the rotation R and translation t.

The rotation is a nonlinear function, incorporating sines and cosines of the rotation angles. If, however, we assume that incremental rotations will be small, it is possible to linearize the rotations, approximating $\cos\theta$ by 1 and $\sin\theta$ by θ. For example, in the case of rotation in x,

$$\begin{aligned} R_{x,\alpha} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \\ &\approx \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -\alpha \\ 0 & \alpha & 1 \end{pmatrix}. \end{aligned}$$

Thus, the full rotation may be approximated as

$$R \approx \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix}, \tag{2}$$

for rotations α, β, and γ around the x, y, and z axes, respectively.

Substituting Equation (2) into (1) we obtain

$$\begin{aligned} \mathcal{E} = \sum_i [\ & (p_{i,x} - \gamma p_{i,y} + \beta p_{i,z} + t_x - q_{i,x}) n_{i,x} + \\ & (\gamma p_{i,x} + p_{i,y} - \alpha p_{i,z} + t_y - q_{i,y}) n_{i,y} + \\ & (-\beta p_{i,x} + \alpha p_{i,y} + p_{i,z} + t_z - q_{i,z}) n_{i,z}\ ]^2, \end{aligned}$$

which may be rewritten as

$$\begin{aligned} \mathcal{E} = \sum_i [\ & (p_i - q_i) \cdot n_i + t \cdot n_i + \\ & \alpha(p_{i,y} n_{i,z} - p_{i,z} n_{i,y}) + \\ & \beta(p_{i,z} n_{i,x} - p_{i,x} n_{i,z}) + \\ & \gamma(p_{i,x} n_{i,y} - p_{i,y} n_{i,x})\ ]^2. \end{aligned}$$

Defining

$$c = p \times n$$

and

$$r = \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix},$$

the alignment error may be written as

$$\mathcal{E} \;=\; \sum_i \left[(p_i - q_i)\cdot n_i + t\cdot n_i + r\cdot c_i\right]^2.$$

We now minimize $\mathcal{E}$ with respect to α, β, γ, t_x, t_y, and t_z by setting the partial derivatives to zero:

$$\begin{aligned}
\frac{\partial \mathcal{E}}{\partial \alpha} &= \sum_i 2\, c_{i,x} \left[(p_i - q_i)\cdot n_i + t\cdot n_i + r\cdot c_i\right] &= 0\\
\frac{\partial \mathcal{E}}{\partial \beta} &= \sum_i 2\, c_{i,y} \left[(p_i - q_i)\cdot n_i + t\cdot n_i + r\cdot c_i\right] &= 0\\
\frac{\partial \mathcal{E}}{\partial \gamma} &= \sum_i 2\, c_{i,z} \left[(p_i - q_i)\cdot n_i + t\cdot n_i + r\cdot c_i\right] &= 0\\
\frac{\partial \mathcal{E}}{\partial t_x} &= \sum_i 2\, n_{i,x} \left[(p_i - q_i)\cdot n_i + t\cdot n_i + r\cdot c_i\right] &= 0\\
\frac{\partial \mathcal{E}}{\partial t_y} &= \sum_i 2\, n_{i,y} \left[(p_i - q_i)\cdot n_i + t\cdot n_i + r\cdot c_i\right] &= 0\\
\frac{\partial \mathcal{E}}{\partial t_z} &= \sum_i 2\, n_{i,z} \left[(p_i - q_i)\cdot n_i + t\cdot n_i + r\cdot c_i\right] &= 0
\end{aligned}$$

These equations may be collected and written in matrix form:

$$\sum_i \begin{pmatrix}
c_{i,x}c_{i,x} & c_{i,x}c_{i,y} & c_{i,x}c_{i,z} & c_{i,x}n_{i,x} & c_{i,x}n_{i,y} & c_{i,x}n_{i,z}\\
c_{i,y}c_{i,x} & c_{i,y}c_{i,y} & c_{i,y}c_{i,z} & c_{i,y}n_{i,x} & c_{i,y}n_{i,y} & c_{i,y}n_{i,z}\\
c_{i,z}c_{i,x} & c_{i,z}c_{i,y} & c_{i,z}c_{i,z} & c_{i,z}n_{i,x} & c_{i,z}n_{i,y} & c_{i,z}n_{i,z}\\
n_{i,x}c_{i,x} & n_{i,x}c_{i,y} & n_{i,x}c_{i,z} & n_{i,x}n_{i,x} & n_{i,x}n_{i,y} & n_{i,x}n_{i,z}\\
n_{i,y}c_{i,x} & n_{i,y}c_{i,y} & n_{i,y}c_{i,z} & n_{i,y}n_{i,x} & n_{i,y}n_{i,y} & n_{i,y}n_{i,z}\\
n_{i,z}c_{i,x} & n_{i,z}c_{i,y} & n_{i,z}c_{i,z} & n_{i,z}n_{i,x} & n_{i,z}n_{i,y} & n_{i,z}n_{i,z}
\end{pmatrix}
\begin{pmatrix} \alpha\\ \beta\\ \gamma\\ t_x\\ t_y\\ t_z \end{pmatrix}
\;=\; -\sum_i \begin{pmatrix}
c_{i,x}\,(p_i - q_i)\cdot n_i\\
c_{i,y}\,(p_i - q_i)\cdot n_i\\
c_{i,z}\,(p_i - q_i)\cdot n_i\\
n_{i,x}\,(p_i - q_i)\cdot n_i\\
n_{i,y}\,(p_i - q_i)\cdot n_i\\
n_{i,z}\,(p_i - q_i)\cdot n_i
\end{pmatrix}.$$

This is a linear matrix equation of the form $Cx = b$, where C is the 6×6 "covariance matrix" accumulated from the c_i and n_i, x is a 6×1 vector of unknowns, and b is a 6×1 vector that also depends on the data points. The equation may be solved using standard methods (A is symmetric, so Cholesky decomposition is the preferred algorithm), yielding the optimal incremental rotation and translation.

2 Analysis of Stability

The above 6×6 covariance matrix also encodes the increase in the alignment error when the transformation is moved away from its optimum:

$$d\mathcal{E} \;=\; \begin{pmatrix} d\alpha & d\beta & d\gamma & dt_x & dt_y & dt_z \end{pmatrix} \begin{pmatrix} & & \\ & C & \\ & & \end{pmatrix} \begin{pmatrix} d\alpha\\ d\beta\\ d\gamma\\ dt_x\\ dt_y\\ dt_z \end{pmatrix}.$$

The larger this increase the greater the stability of ICP, since the error landscape will have a deep, well-defined minimum. On the other hand, if there are incremental transformations that cause only a small increase in alignment error, ICP will be relatively unstable with respect to these degrees of freedom.

By expanding C in terms of its eigenvectors we may see directly the effect of various incremental transformations. If all eigenvalues of C are large, any transformation away from the minimum will result in a large increase in alignment error. If, on the other hand, one or more eigenvalues are small, the corresponding eigenvectors are transformations that do not increase error much, and therefore represent directions in transformation space along which the error landscape is shallow.

3 Applications of Eigenvalue Analysis

The most obvious application of the above analysis is to evaluate the stability of aligning two meshes together. This involves computing the matrix C, summed over the entire region of overlap, and finding its eigenvalues. Any small eigenvalues indicate a low-confidence alignment. This has implications on which pairings to use for global registration.

A second potential application involves looking at small patches on a single mesh, computing the eigenvalues of C over each patch, and thus determining the *potential* stability of ICP on each region. Applications of this might be:

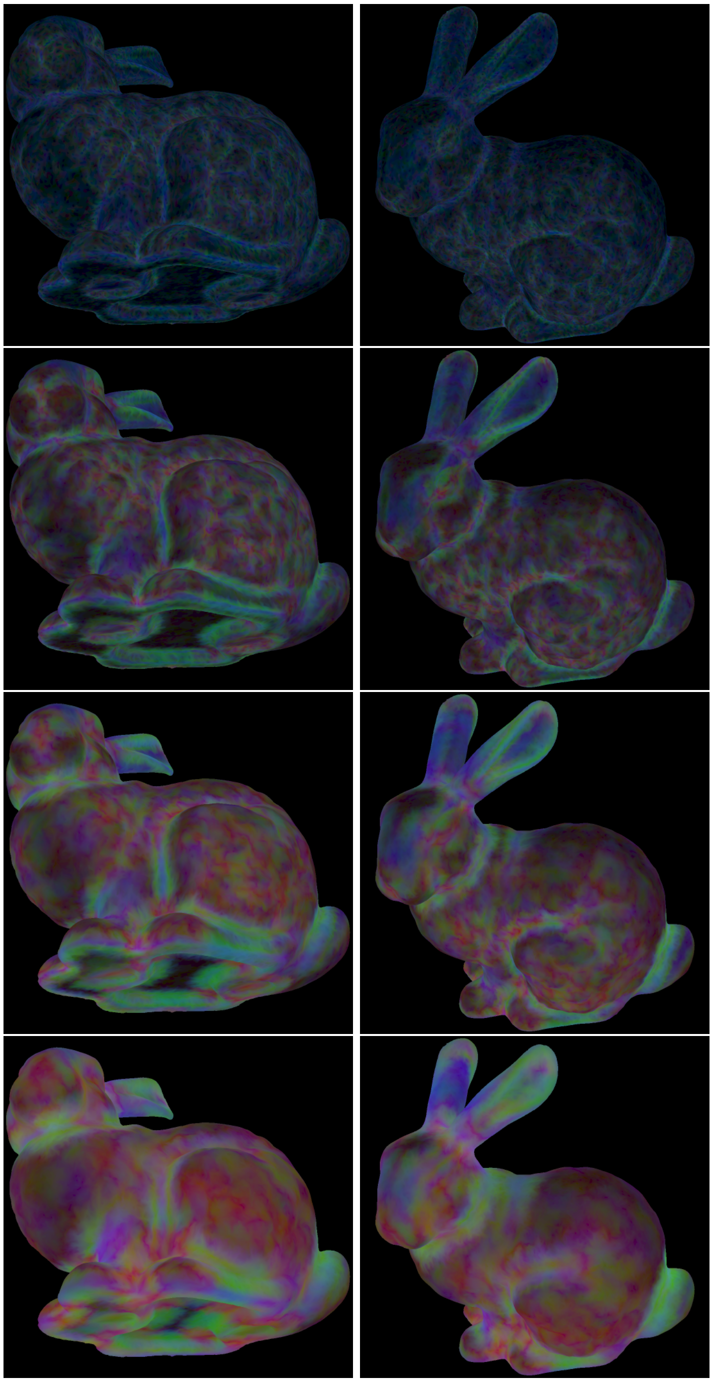
- Using the local stability to assign weights during ICP. This would help to prevent noise in mostly-flat regions from swamping the "signal" available near good features.

- Using local stability to determine the best places to compute and store surface signatures for structural indexing.

- Building a "dictionary" of local surface shapes and the number of small eigenvalues each produces. For example:

Planar patch	3 small eigenvalues	1 rotation and 2 translations unstable
Spherical patch	3 small eigenvalues	3 rotations unstable
Cylindrical patch	2 small eigenvalues	1 translation and 1 rotation unstable
Patch w. spherical bump	1 small eigenvalue	1 rotation unstable
Patch w. groove	1 small eigenvalue	1 translation unstable
Corner of a cube	0 small eigenvalues	No unstable components of alignment

  This table is probably not complete – it would be nice to come up with a systematic way of classifying the possibilities.



Neighborhood size: 1

Neighborhood size: 3

Neighborhood size: 6

Neighborhood size: 10

Figure 1: Bunny model color coded according to the magnitude of the three smallest eigenvalues on local neighborhoods. The four rows of the table correspond to evaluating C over neighborhoods of radius 1, 3, 6, and 10 edges. The color coding is such that black regions correspond to 3 small eigenvalues, blue to 2 small eigenvalues, green to 1 small eigenvalue, and red to no small eigenvalues.