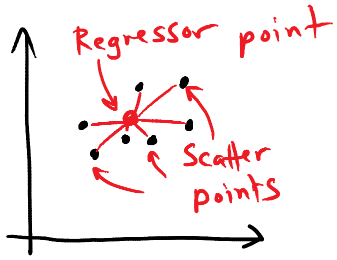
Registration by Clustering

L 34

Goal: To investigate a different way of quantifying the dispersion/compactness of the joint intensity scatter plot.

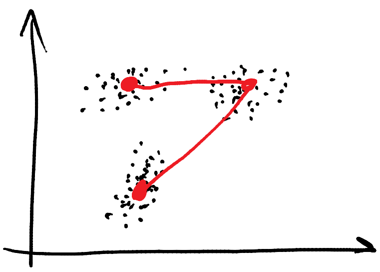
Rather than having to classify pixels into bins, it would be good to derive the cost directly from the points in the joint intensity scatter plot (JISP).
Recall that aligned images result in tighter clustering in the JISP (less dispersion).

Suppose we add a regressor point to the JISP.



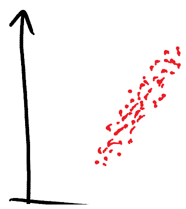
Then the cost could be computed by the **sum of the squared distances** of the JISP points to that regressor point.

In this way, we model a cluster of points using a point regressor. We can add multiple point regressors (PRs), one for each cluster.



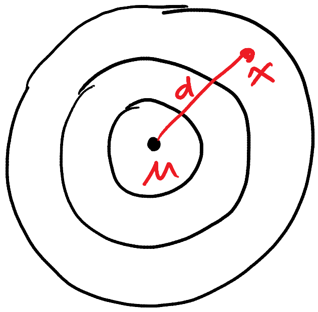
Each JISP point is **assigned** to its nearest PR, and the total cost is the sum of the squares of the distances from each JISP point to its nearest PR.

Some clusters of points are elongated:
We can model this behaviour using
Generalized Euclidean Distance (GED).



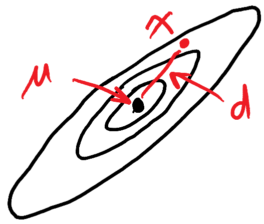
We can model this behaviour using
Generalized Euclidean Distance (GED).

Euclidean Distance.



$$d = (x-\mu)^T(x-\mu)$$

GED

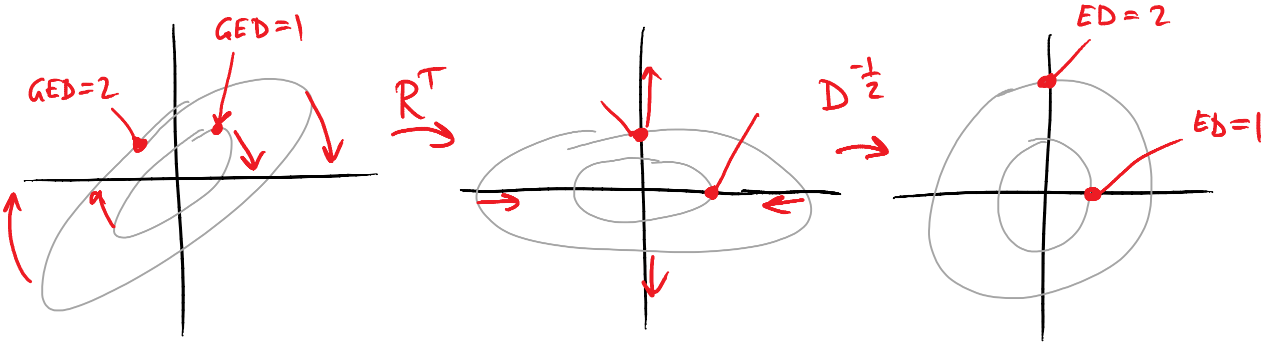


$$d = (x-\mu)^T\Sigma^{-1}(x-\mu)$$

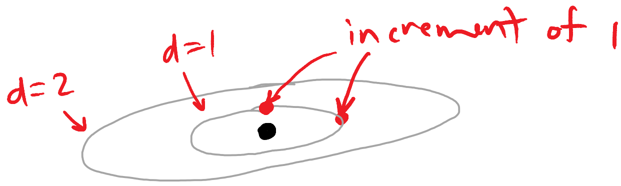
Σ is a covariance matrix, composed of scalings and a rotation. Notice that Euclidean distance is a special case of GED (as the name implies).

$$\begin{aligned} d &= (x-\mu)^T\Sigma^{-1}(x-\mu) \\ &= (x-\mu)^T R D^{-1} R^T (x-\mu) \\ &= \left[(x-\mu)^T R D^{-\frac{1}{2}}\right]\left[D^{-\frac{1}{2}} R^T (x-\mu)\right] \\ &= \left[D^{-\frac{1}{2}} R^T (x-\mu)\right]^T\left[D^{-\frac{1}{2}} R^T (x-\mu)\right] = \|D^{-\frac{1}{2}} R^T (x-\mu)\|_2^2 \end{aligned}$$

since Σ is symmetric positive definite



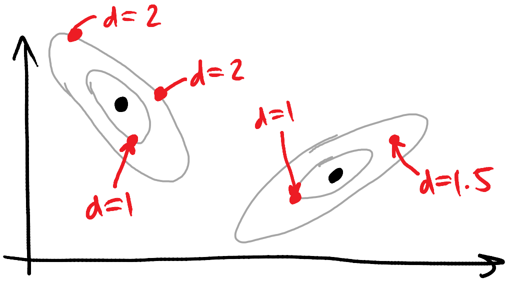
In GED, displacement along the minor axis increments distance more than displacement along the major axis.



The total cost is

$$\sum_{i \in JISP} (\text{GED to assigned regressor})^2$$

eg.



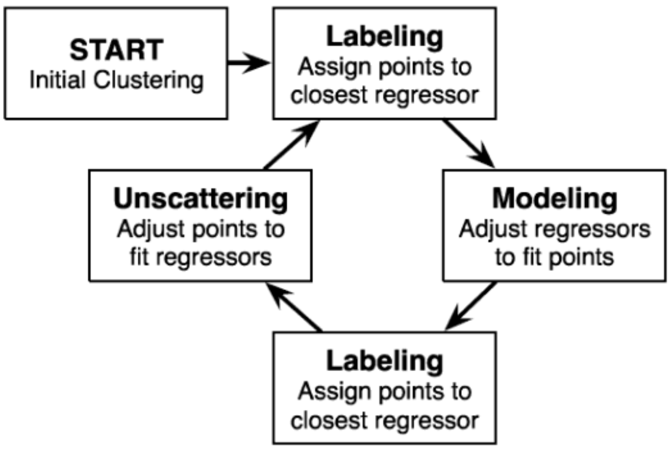
$$\text{Cost} = 2^2 + 2^2 + 1^2 + 1^2 + 1.5^2$$
$$= 12.25$$

If one image is moved slightly (eg. 0.01°), the JISP points move slightly, and the cost changes only slightly.

==> continuous

There are 3 processes involved in registration using JISP regressors:

Labelling
Modelling
Unscattering



These 3 parts are iterated inside a fixed-point interation framework.

Labelling

This is simply assigning each JISP point to its nearest regressor.

Modelling

Each set of points is approximated by a single regressor. The point of modelling is to adjust that regressor to better fit the points assigned to it.

Optimal location: **centroid of the cluster (centre of mass)**

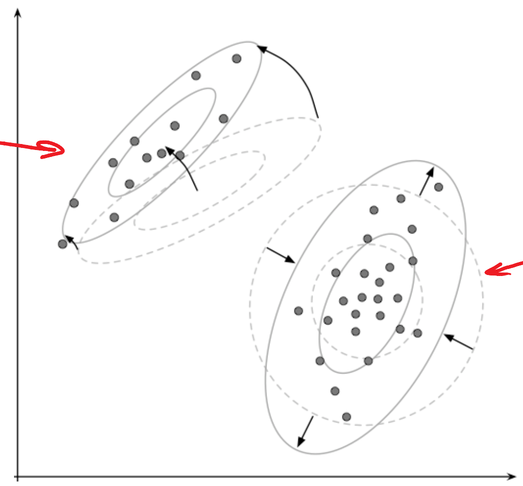
i.e. Before After

This gives the min. GED.

Optimal covariance:

$$\Sigma = \sum_i (x-\mu)(x-\mu)^T = \sum_i$$

Both μ and Σ are adjusted



Only Σ needs to be adjusted

Unscattering

We adjust the motion parameters to bring the scatter points closer (on average) to their regressors.

Suppose we are registering f and g.

Let $p_f = [x_f \ y_f \ \theta_f]^T$ and $p_g = [x_g \ y_g \ \theta_g]^T$

Let $p_f = [x_f \; y_f \; \theta_f]'$ and $p_g = [x_g \; y_g \; \theta_g]'$

be the 2D rigid-body motion parameters for f & g.

Then our full set of motion parameters is

$$p = \begin{bmatrix} p_f \\ p_g \end{bmatrix} \quad \text{(6×1 column vector)}$$

As usual, we linearize. For pixel i

$$f_i(p_f) \cong f_i + \frac{\partial f_i}{\partial x_f} x_f + \frac{\partial f_i}{\partial y_f} y_f + \frac{\partial f_i}{\partial \theta_f} \theta_f$$

$$g_i(p_g) \cong g_i + \frac{\partial g_i}{\partial x_g} x_g + \frac{\partial g_i}{\partial y_g} y_g + \frac{\partial g_i}{\partial \theta_g} \theta_g$$

Then, the i^{th} scatter point is

$$s_i(p) = \begin{bmatrix} f_i(p_f) \\ g_i(p_g) \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} f_i \\ g_i \end{bmatrix}}_{s_i} + \underbrace{\begin{bmatrix} \frac{\partial f_i}{\partial x_f} & \frac{\partial f_i}{\partial y_f} & \frac{\partial f_i}{\partial \theta_f} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial g_i}{\partial x_g} & \frac{\partial g_i}{\partial y_g} & \frac{\partial g_i}{\partial \theta_g} \end{bmatrix}}_{L_i} \underbrace{\begin{bmatrix} p_f \\ p_g \end{bmatrix}}_{p}$$

$$= s_i + L_i p$$

Hence, our cost function can be written

$$C(p) = \sum_i \Big\| \underbrace{D_i^{-\frac{1}{2}} R_i^T (s_i + L_i p - \mu_i)}_{\text{2-vector}} \Big\|^2$$

Instead of representing our cost as a sum of 2-vectors squared, we can stack all the 2-vectors into one giant column vector with 2×(# pixels) elements.

squared, we can stack all the 2-vectors into one giant column vector with $2*(\#\text{pixels})$ elements.

$$\left\| \begin{bmatrix} D_1^{-\frac{1}{2}} R_1^T L_1 \\ \vdots \\ D_N^{-\frac{1}{2}} R_N^T L_N \end{bmatrix} p + \begin{bmatrix} D_1^{-\frac{1}{2}} R_1^T (s_1 - \mu_1) \\ \vdots \\ D_N^{-\frac{1}{2}} R_N^T (s_N - \mu_N) \end{bmatrix} \right\|^2 = \| \bar{L} p - \bar{d} \|^2$$

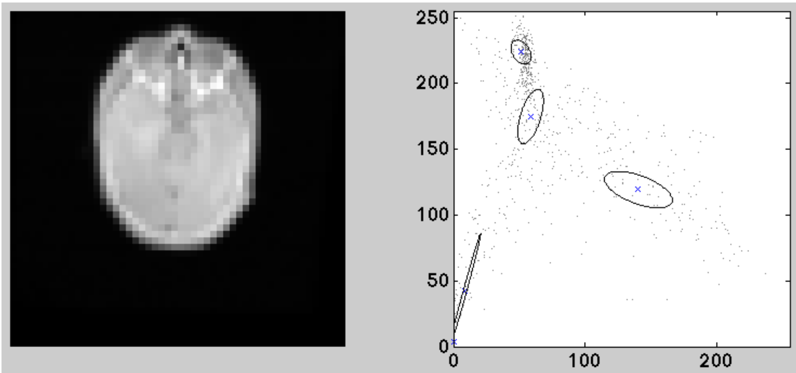
Thus, we want to solve

$$\min_p \| \bar{L} p - \bar{d} \|^2 .$$

This is a simple linear least-squares problem,

$$p = \underbrace{(\bar{L}^T \bar{L})}_{6\times 6 \text{ matrix}}{}^{-1} \bar{L}^T \bar{d}$$

(video)



Here is a slightly different implementation that uses line segments instead of GED.

(video)