Image Registration
Lecture 18: Initialization

Prof. Charlene Tsai

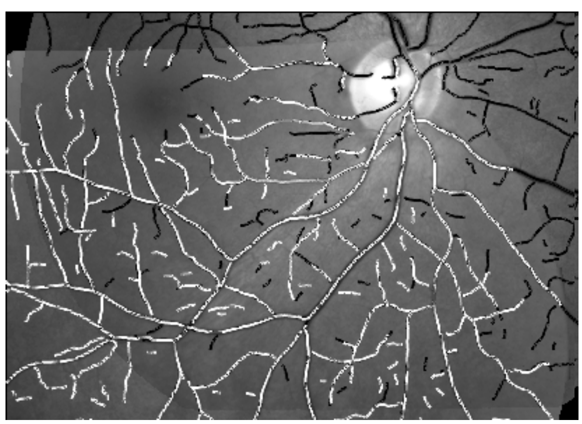
Overview - Lectures 18 & 19

- Lecture 18:
  - Importance of initialization
  - Simple methods:
    - Sampling of parameter space
    - Moment-based methods
  - Least-squares estimation of rotation in 3D
- Lecture 19:
  - Invariants
  - Interest points and correspondences
  - Robust estimation based on random-sampling

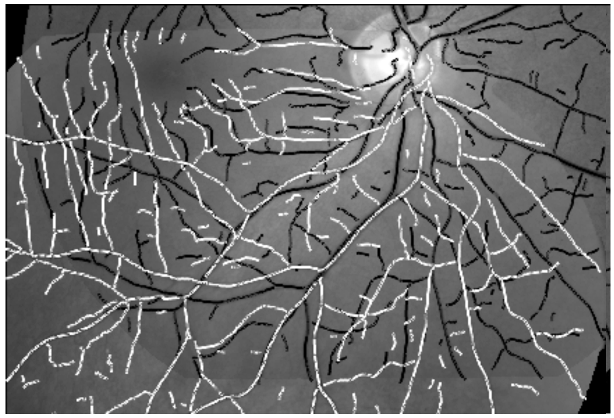
Motivation: Domain of Convergence

- Most core registration algorithms, such as intensity-based gradient descent and ICP, have a narrow domain of convergence ("capture range")
- We saw this in Lectures 6-10 on intensity-based registration
- The following page repeats an earlier example using feature-based registration

ICP - Correct Convergence



ICP - Incorrect Convergence



What Affects the Domain of Convergence?

- Structural complexity:
  - Lots of similar blood vessels in retinal image
- Smoothness of objective function
  - Small hitches in the objective function can cause the minimization to become “stuck”
- Flexibility of search technique
  - More flexible, adaptive searches have broader domains of convergence
- Complexity of parameter space
  - More complicated transformations are more difficult to minimize

Improving Domain of Convergence

- Techniques
  - Multiresolution
  - Region-growing
  - Increasing order models
  - Adaptive search techniques
  - Multiple initial estimates

Implications for Initialization

- Without these techniques, initialization must be extremely good!
- With them, initialization is still important
- Many of the initialization techniques we'll describe can or should be used in conjunction with one (or more) of the foregoing methods
- In addressing a registration problem you need to think about:
  - Complexity of the problem (data, transformation, etc)
  - Capabilities of techniques available
  - Methods of initialization available
- These all interact in designing the solution

Onto Initialization Techniques…

- Prior information / manual initialization
- Sampling of parameter space
- Moments
  - And Fourier methods
- Invariants
- Random-sampling of correspondences

9

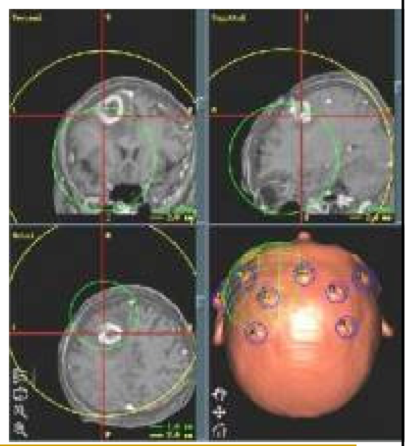
Prior Information / Manual Initialization

- Registration starts from an approximately known position
  - Industrial inspection is a common example
- The goal of registration is then high-precision "docking" of the part against a model
- One must be careful to analyze the domain of convergence and the initial uncertainty in position to determine when this will work
- In some applications, it is sufficient to manually "drag" one image on top of another
  - This is really a translation-based initialization

10

Fiducials

- Often used in image-guided surgery
- Easily-detected markers (beads) are placed pre-operatively
- Initial registration is based on the correspondence of these fiducials
  - This can be done manually or automatically
- This initializes an automatic procedure for fine-resolution alignment of all image data



11

Sampling of Parameter Space

- Applied at coarse resolutions and low-order transformations (esp. translation only)
- Initialize by sampling a range of transformations established a priori
- Sample spacing is based on the convergence properties of the algorithm:
  - Wider spacing of samples is possible when the algorithm has a broader domain of convergence
- Assumes the registration algorithm is fast enough to allow testing of all samples
  - Imagine how **many** samples you could test if the registration algorithm required only a few milliseconds!
  - Imagine how **few** samples you could test if the registration algorithm required hours!

12

Alignment Based on Moments

- Used for aligning "point sets" such as range data sets
- Applied before establishing correspondence
- Based on first moments (center of mass) and second moments and low order transformations.
- The details are described the next few slides
- Related techniques can be applied to images, often by matching Fourier transformations:
  - Translations in the spatial domain become phase shifts in the frequency domain
  - Rotations in the spatial domain become rotations in the frequency domain
  - Scaling in the spatial domain corresponds to inverse scaling in the frequency domain

13

Setting Up the Problem

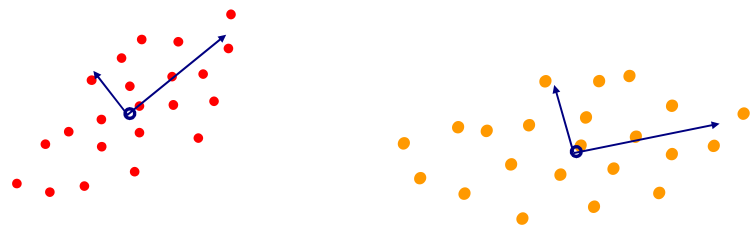
- Given two point sets

$$\mathcal{P} = \{\mathbf{p}_i\} \quad \text{and} \quad \mathcal{Q} = \{\mathbf{q}_i\}$$

- Our goal is to find the similarity transformation that best aligns the moments of the two point sets.
- In particular we want to find
  - The rotation matrix **R**,
  - The scale factor *s*, and
  - The translation vector **t**

14

In Pictures



Moments

- The first moment is the center of mass

$$\mathbf{p}_c = \sum_{\mathbf{p}_i \in \mathcal{P}} \mathbf{p}_i \;/\; |\,\mathcal{P}\,| \qquad \text{and} \qquad \mathbf{q}_c = \sum_{\mathbf{q}_j \in \mathcal{Q}} \mathbf{q}_j \;/\; |\,\mathcal{Q}$$

Second Moments

- Scatter matrix

$$\mathbf{S}_p = \sum_{\mathbf{p}_i \in \mathcal{P}} (\mathbf{p}_i - \mathbf{p}_c)(\mathbf{p}_i - \mathbf{p}_c)^T \quad \text{and} \quad \mathbf{S}_q = \sum_{\mathbf{q}_j \in \mathcal{Q}} (\mathbf{q}_j - \mathbf{q}_c)(\mathbf{q}_j - \mathbf{q}_c)^T$$

- Eigenvalue / eigenvector decomposition ($\mathbf{S}_p$)

$$\mathbf{S}_p = \mathbf{V}_p \operatorname{diag}(\lambda_1, \ldots, \lambda_n) \mathbf{V}_p^T$$

- where λ_i are the eigenvalues and the columns of $\mathbf{V}_p$ hold the corresponding eigenvectors:

17

Effect of Rotation, Translation and Scale

- Rotated points:

$$\mathbf{p}'_i = s\mathbf{R}\mathbf{p}_i + \mathbf{t}$$

- Center of the transformed data is the same as the rotated, scaled and translated center before transformation:

$$\begin{aligned}\mathbf{p}'_c &= \frac{1}{|\mathcal{P}|} \sum \mathbf{p}'_i \\ &= \frac{1}{|\mathcal{P}|} \sum (s\mathbf{R}\mathbf{p}_i + \mathbf{t}) \\ &= s\mathbf{R} \frac{1}{|\mathcal{P}|} \sum \mathbf{p}_i + \mathbf{t} \\ &= s\mathbf{R}\mathbf{p}_c + \mathbf{t}\end{aligned}$$

18

Scatter Matrix

- Rotated, scaled scatter matrix:

$$\begin{aligned}\mathbf{S}'_p &= \sum(\mathbf{p}'_i - \mathbf{p}'_c)(\mathbf{p}'_i - \mathbf{p}'_c)^T \\ &= \sum(s\mathbf{R}\mathbf{p}_i - s\mathbf{R}\mathbf{p}_c)(s\mathbf{R}\mathbf{p}_i - s\mathbf{R}\mathbf{p}_c)^T \\ &= (s\mathbf{R})\sum(\mathbf{p}_i - \mathbf{p}_c)(\mathbf{p}_i - \mathbf{p}_c)^T(s\mathbf{R})^T \\ &= s^2\mathbf{R}\mathbf{S}_p\mathbf{R}^T\end{aligned}$$

19

Eigenvalues and Eigenvectors

- Given an eigenvector $\mathbf{v}_j$ of $\mathbf{S}_p$ and its rotation $\mathbf{v}_j' = \mathbf{R}\mathbf{v}_j$:

$$\begin{aligned}\lambda_j\mathbf{v}_j &= \mathbf{S}\mathbf{v}_j \\ &= \mathbf{S}\mathbf{R}^T\mathbf{R}\mathbf{v}_j\end{aligned}$$

- Multiply both sides by $s^2\mathbf{R}$, and manipulate:

$$s^2 R\lambda_j v_j = s^2 RSR^T Rv_j$$

$$s^2\lambda_j v_j' = S_p' v_j'$$

20

(cont)

- So,
  - Eigenvectors are rotated eigenvectors
  - Eigenvalues are multiplied by the square of the scale
    - Std deviations, the squareroots of the eigenvalues, increase only by the scale

Using This in Registration

- Goal: compute similarity transformation that best aligns the first and second moments
- Assume the 2nd moments (eigenvalues) are distinct
- Procedure:
  - Center the data in each coordinate system
  - Compute the scatter matrices and their eigenvalues and eigenvectors

$$\mathbf{V}_p = \begin{pmatrix} \mathbf{v}_{p,1} & \cdots & \mathbf{v}_{p,k} \end{pmatrix} \qquad \mathbf{V}_q = \begin{pmatrix} \mathbf{v}_{q,1} & \cdots & \mathbf{v}_{q,k} \end{pmatrix}$$

$$\{\lambda_{p,1}, \ldots, \lambda_{p,k}\} \qquad \{\lambda_{q,1}, \ldots, \lambda_{q,k}\}$$

  - Using this, compute
    - Scaling and rotation
    - Then, translation

Rotation

- The rotation matrix should align the eigenvectors in order:

$$\mathbf{v}_{q,1} = \mathbf{R}\mathbf{v}_{p,1}$$
$$\cdots \quad \cdots$$
$$\mathbf{v}_{q,k} = \mathbf{R}\mathbf{v}_{p,k}$$

- Manipulating:

$$\begin{aligned}\begin{pmatrix}\mathbf{v}_{q,1} & \cdots & \mathbf{v}_{q,k}\end{pmatrix} &= \begin{pmatrix}\mathbf{R}\mathbf{v}_{p,1} & \cdots & \mathbf{R}\mathbf{v}_{p,k}\end{pmatrix} \\ &= \mathbf{R}\begin{pmatrix}\mathbf{v}_{p,1} & \cdots & \mathbf{v}_{p,k}\end{pmatrix} \\ \mathbf{V}_q &= \mathbf{R}\mathbf{V}_p\end{aligned}$$

- As a result: $\mathbf{R} = \mathbf{V}_q \mathbf{V}_p^T$

23

Scale

- Ratios of corresponding eigenvalues should determine the scale:

$$s^2 = \lambda_{qj} / \lambda_{pj}$$

- But, because of noise and modeling error the ratios will not be exact.
- Hence, we switch to a least-squares formulation

$$E(s^2) = \sum_j (\lambda_{q,j} - s^2 \lambda_{p,j})^2$$

24

(cont)

- Taking the derivative with respect to s^2, which we are treating as the variable here, we have:

$$\frac{\partial E}{\partial s^2} = (-2)\sum_j (\lambda_{q,j} - s^2\lambda_{p,j})\lambda_{p,j}$$

- Setting the result to 0 and solving yields

$$s^2 = \frac{\sum_j \lambda_{q,j}\lambda_{p,j}}{\sum_j \lambda_{p,j}^2}$$

Translation

- Once we have the rotation and scale we can compute the translation based on the centers of mass:

$$\mathbf{t} = \mathbf{q}_c - s\mathbf{R}\mathbf{p}_c$$

Comments

- Calculations are straightforward, non-iterative and do not require correspondences:
  - Moments, first
  - Rotation and scale separately
  - Translation
- Assumes the viewpoints of the two data sets coincide approximately
  - Can fail miserably for significantly differing viewpoints

Part 2: Estimating Rotations In 3D

- Goal: Given moving and fixed feature sets and correspondences between them,

$$\mathcal{C} = \{(\mathbf{g}_k, \mathbf{f}_k)\}$$

  estimate the rigid transformation (no scaling at this point) between them
- New challenge: orthonormal matrices in 3d have 9 parameters and only 3 degrees of freedoms (DoF)
  - The orthonormality of the rows and columns eliminates 6 DoF
- This is one example of constrained optimization, where the number of parameters and the degrees of freedom don't match

Rotation Matrices in 3D

- Formed in many ways.
- Here we'll consider rotations about 3 orthogonal axes:

$$\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix}$$

$$\mathbf{R}_y = \begin{pmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{pmatrix}$$

$$\mathbf{R}_z = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

29

Composing

- Composing rotations about the 3 axes, with rotation about z, then y, then x, yields

$$\mathbf{R} = \mathbf{R}_x\mathbf{R}_y\mathbf{R}_z$$

$$= \begin{pmatrix} \cos\alpha\cos\theta & -\cos\alpha\sin\theta & \sin\alpha \\ \cos\theta\sin\alpha\sin\phi + \cos\phi\sin\theta & -\sin\alpha\sin\phi\sin\theta + \cos\phi\cos\theta & -\cos\alpha\sin\phi \\ -\cos\phi\cos\theta\sin\alpha + \sin\phi\sin\theta & \cos\phi\sin\alpha\sin\theta + \cos\theta\sin\phi & \cos\alpha\cos\phi \end{pmatrix}$$

- Notes:
  - It is an easy, though tedious exercise to show that R is orthonormal
  - Changing the order of composition changes the resulting matrix R
- Most importantly, it appears that estimating the parameters (angles) of the matrix will be extremely difficult in this form.

30

Options

- Different formulation:
  - Quaternions are popular
  - Angle-axis
- Approximations:
  - Route we will take here
  - Perhaps better than quaternions for error projector matrices that yield non-Euclidean distances

Small-Angle Approximation

- First-order (small angle) Taylor series approximations:

$$\sin\theta = \theta - \frac{\theta^3}{3} + \frac{\theta^5}{5} - \dots \approx \theta \qquad \cos\theta = 1 - \frac{\theta^2}{2} + \frac{\theta^4}{4} - \dots \approx 1$$

- Apply to R:

$$\mathbf{R} \approx \begin{pmatrix} 1 & -\theta & \alpha \\ \alpha\phi + \theta & 1 - \alpha\phi\theta & -\phi \\ -\alpha + \phi\theta & \alpha\theta + \phi & 1 \end{pmatrix}$$

Small-Angle Approximation

- Eliminate 2nd order and higher combined terms:

$$\mathbf{R} \approx \begin{pmatrix} 1 & -\theta & \alpha \\ \theta & 1 & -\phi \\ -\alpha & \phi & 1 \end{pmatrix} = \mathbf{I} - \begin{pmatrix} 0 & -\theta & \alpha \\ \theta & 0 & -\phi \\ -\alpha & \phi & 0 \end{pmatrix}$$

- Discussion:
  - Simple, but no longer quite orthonormal
  - Can be viewed as the identity minus a skew-symmetric matrix, which is closely related to robotics formulations.

33

Least-Squares Estimation

- Rigid transformation distance error:

$$\mathbf{Rg} + \mathbf{t} - \mathbf{f} = \begin{pmatrix} 1 & -\theta & \alpha \\ \theta & 1 & -\phi \\ -\alpha & \phi & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \mathbf{t} - \mathbf{f}$$

$$= \begin{pmatrix} 0 & z & -y & 1 & 0 & 0 \\ -z & 0 & x & 0 & 1 & 0 \\ y & -x & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \phi \\ \alpha \\ \theta \\ t_x \\ t_y \\ t_z \end{pmatrix} - (\mathbf{f} - \mathbf{g})$$

$$= \mathbf{Xa} - \mathbf{r}$$

- This has the same form as our other distance error terms, with **X** and **r** depending on the data and **a** being the unknown parameter vector.

34

Least-Squares Estimation

- Using the error projector we can estimate the parameter vector **a** in just the same way as we did for affine estimation
  - Note: we are not estimating the scale term here, which makes the problem easier. This is ok in cases, such as range data, where the scale is known.
- What we need to worry about is undoing the effects of the small-angle approximation. In particular we need to
  - Make the estimated **R** orthonormal
  - Iteratively update **R**

Making the Estimate Orthonormal

- Inserting the estimated parameters into R

$$\hat{\mathbf{R}} = \begin{pmatrix} 1 & -\theta & \alpha \\ \theta & 1 & -\phi \\ -\alpha & \phi & 1 \end{pmatrix}$$

- The matrix is NOT orthonormal.
- Two solutions:
  - Put the estimated parameters (angles) back into the original matrix (with all of the sines and cosines)
  - Find the closest orthonormal matrix to R.
    - This is the option we apply. It is very simple.

The Closest Orthonormal Matrix

- It can be proved that the closest orthonormal matrix, in the Frobenius-norm sense is found by computing the SVD and setting the singular values to 1
- In other words, with

$$\hat{\mathbf{R}} = \mathbf{U}\mathbf{W}\mathbf{V}^T$$

- The closest orthonormal matrix is

$$\hat{\mathbf{R}} \approx \mathbf{U}\mathbf{V}^T$$

37

Iteratively Estimating R and t

- Given are initial estimates of **R** and **t**, the correspondences $\{(\mathbf{g}_k, \mathbf{f}_k)\}$ and the error projectors $\mathbf{P}_k$
- Do
  - Apply the current estimate to each original moving image feature:

    $$\mathbf{g}'_k = \mathbf{R}\mathbf{g}_k + \mathbf{t}$$

  - Estimate rotation and translations, as just described above, based on the correspondences $\{(\mathbf{g}_k', \mathbf{f}_k)\}$.
  - Convert the rotation to an orthonormal matrix. Call the results $\Delta\mathbf{R}$ and $\Delta\mathbf{t}$.
  - Update the estimates to **R** and **t**. In particular, because the transformation is now,

    $$\Delta\mathbf{R}(\mathbf{R}\mathbf{g}_k + \mathbf{t}) + \Delta\mathbf{t}$$

    the new estimates are

    $$\mathbf{R} = \Delta\mathbf{R} \cdot \mathbf{R} \qquad \text{and} \qquad \mathbf{t} = \Delta\mathbf{R} \cdot \mathbf{t} + \Delta\mathbf{t}$$

- Until $\Delta\mathbf{R}$ and $\Delta\mathbf{t}$ are sufficiently small (only a few iterations)

38

Summary and Discussion

- Small angle approximation leads to simple form of constraint that can be easily incorporated into a least-squares formulation
- Resulting matrix must be made orthonormal using the SVD
- Estimation, for a fixed set of correspondences, becomes an iterative process

39

Looking Ahead to Lecture 19

- Initialization based on
  - Matching of interest points
  - Random-sampling robust estimation
  - Related papers are available online.

40