Image Registration
Lecture 4: First Examples

Prof. Charlene Tsai

Outline

- Example
- Intensity-based registration
- SSD error function
- Image mapping
- Function minimization:
  - Gradient descent
  - Derivative calculation
- Algorithm
- Results and discussion

2

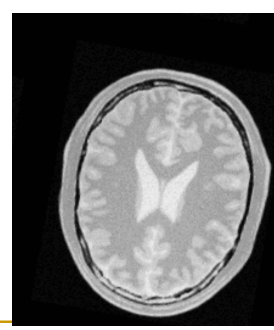
Reading Material

- Paper:
  - Hill, Batchelor, Holden and Hawkes, Medical Image Registration, *Physics of Medicine and Biology* **46** (2001) R1-R45.
- Available electronically from my website
- Excellent introduction to registration problem, but heavily slanted toward medical applications using similarity transformations

Running Example

- MRI image registration, similarity transformation (rotated by 10 degrees, with a translation of 17mm and 13mm)

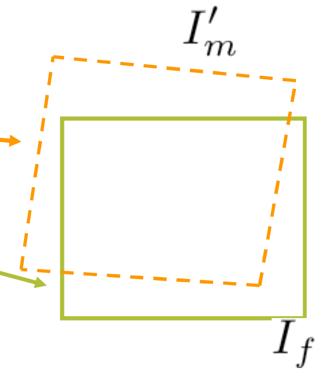
I_m



I_f

Intensity-Based Registration

- Use the intensities more or less directly
- Compare intensities between
  - Mapped (transformed) version of the moving image $\mathbf{I}_m$ (based on an estimated transformation) and
  - Fixed image $\mathbf{I}_f$
- Need:
  - Pixel-by-pixel error measure
  - Mapping technique
  - Minimization technique



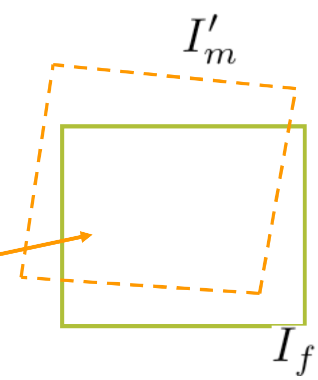
5

Example Error Measure: SSD

$$\sum_{\mathbf{p} \in \Omega} \left[I_f(\mathbf{p}) - I'_m(\mathbf{p}) \right]^2$$

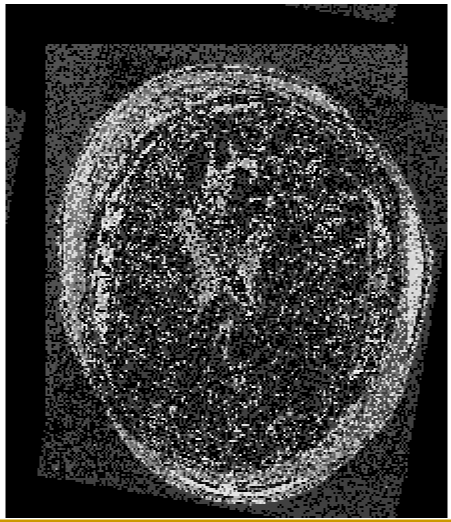
Ω Region of intersection between images

$\mathbf{p}$ Pixel location within region



6

SSD Example: Initial Alignment



7

SSD: Sum of Squared Errors

- Advantages:
  - Simple to compute
  - Differentiable
  - Optimal for Gaussian error distributions
- Disadvantages:
  - Doesn't allow varying "gain" between the images, which may be caused by different illuminations or different camera settings
  - Biased by large errors in intensity
    - E.g. caused by contrast agent injection

8

Working in the Parameters

- Remember:

$$I'_m(\mathbf{p}) = I_m(\mathbf{T}^{-1}(\mathbf{p};\mathbf{\Theta}))$$

- This means that to evaluate the effect of a transformation estimate what we really want to evaluate is

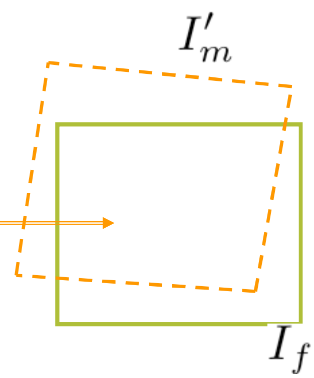
$$\sum_{\mathbf{p}\in\Omega}\left[I_f(\mathbf{p}) - I_m(\mathbf{T}^{-1}(\mathbf{p};\mathbf{\Theta}))\right]^2$$

9

Aside: The Role of the Region

- Observe: the region over which the transformation is evaluated depends on the parameters:

$$\sum_{\mathbf{p}\in\Omega}\left[I_f(\mathbf{p}) - I_m(\mathbf{T}^{-1}(\mathbf{p};\mathbf{\Theta}))\right]^2$$

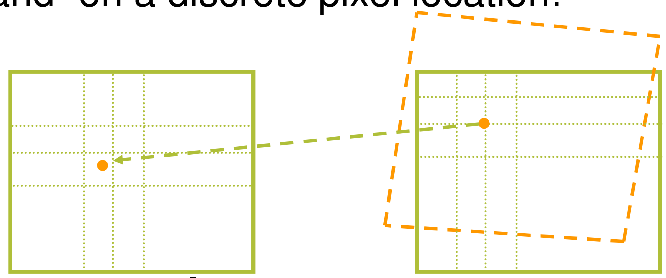


- This can cause problems in practice:
  - A transformation resulting in no overlap leads to 0 error!

10

Evaluating the Objective Function

- Pixel-by-pixel evaluation within the region
- Apply the inverse mapping at each pixel
- Problem: inverse mapping of pixel does not "land" on a discrete pixel location!

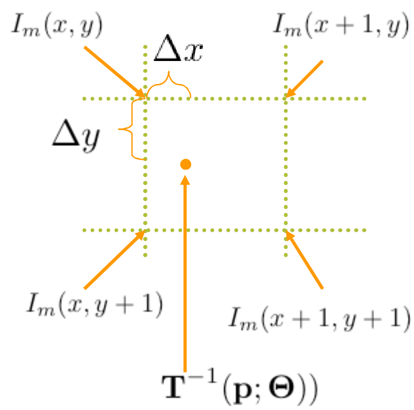


Many Interpolation Options

- Nearest neighbor
- Bilinear (or trilinear in 3d)
- Spline
- Sinc

Bilinear Interpolation in Moving Image

- Weighted average of 4 surrounding pixels
  - 8 surrounding pixels in 3d
- Weight proportional to distance in x and in y



13

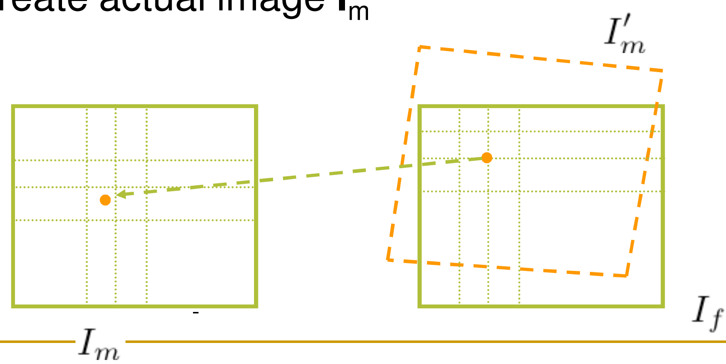
Bilinear: Resulting Intensity

$$\begin{aligned}\mathbf{T}^{-1}(\mathbf{p};\mathbf{\Theta}) = & (1-\Delta x)(1-\Delta y)\ I_m(x,y) \\ & + \Delta x(1-\Delta y)\ I_m(x+1,y) \\ & + (1-\Delta x)\Delta y\ I_m(x,y+1) \\ & + \Delta x \Delta y\ I_m(x+1,y+1)\end{aligned}$$

14

Two Options In Practice

- Create intensity, pixel-by-pixel, but don't create an explicit image $\mathbf{I}_m$'
- Create actual image $\mathbf{I}_m$'



Resetting the Stage

- We have:
  - Formulated the SSD objective function
  - Discussed how to evaluate it
- Next step is how to minimize it with respect to the transformation parameters

Before Proceeding

- We will estimate the parameters of the backward transformation
- Abusing notation, we will minimize the equation

$$\sum_{\mathbf{p} \in \Omega} \left[I_f(\mathbf{p}) - I_m(\mathbf{T}(\mathbf{p}; \mathbf{\Theta})) \right]^2$$

- It should be understood (implicitly) that this is the inverse transformation and the parameter values will be different

17

Thinking Abstractly: Function Minimization

- Function to minimize:

$$E(\mathbf{\Theta}) = \sum_{\mathbf{p} \in \Omega} \left[I_f(\mathbf{p}) - I_m(\mathbf{T}(\mathbf{p}; \mathbf{\Theta})) \right]^2$$

- Possibilities
  - Amoeba (simplex) methods - non-differential
  - Gradient / steepest descent ←
  - Linearization (leading to least-squares)
  - Newton's method
  - Many more …

18

Gradient / Steepest Descent

- Compute gradient of objective function (with respect to transformation parameters), evaluated at current parameter estimate

$$\nabla E(\mathbf{\Theta}_t) = \frac{\partial E}{\partial \mathbf{\Theta}}(\mathbf{\Theta}_t)$$

- Make tentative small change in parameters in the negative gradient direction

$$\mathbf{\Theta}_{t+1} = \mathbf{\Theta}_t - \eta \nabla E(\mathbf{\Theta}_t)$$

  - η is called the "learning rate"

(cont)

- Re-evaluate objective function and accept change if it is reduced (otherwise reduce the learning rate)
- Continue until no further changes are possible

Computing the Derivative

- Issue:
  - Images are discrete
  - Parameters are continuous
- Two methods
  - Numerical
  - Continuous (eventually numerical as well)
- Abstract definition of parameter vector:

$$\boldsymbol{\Theta} = \begin{pmatrix} \theta_1 & \theta_2 & \cdots & \theta_k \end{pmatrix}^T$$

21

Numerical Derivatives

- Form each partial derivative by taking a small step in each parameter, $i = 1,..,k$:

$$\frac{\partial E}{\partial \theta_i} \approx \frac{E(\theta_1, \ldots, \theta_i + \Delta\theta_i, \ldots, \theta_k) - E(\theta_1, \ldots, \theta_i, \ldots, \theta_k)}{\Delta\theta_i}$$

- Choice of step size can be difficult
- Requires k function evaluations to compute the derivative
- Sometimes this is the only thing you can do!

22

Continuous Computation of Derivative

- Apply chain rule:

$$\frac{\partial E}{\partial \boldsymbol{\Theta}} = \sum_{\mathbf{p}\in\Omega} -2\left[I_f(\mathbf{p}) - I_m(\mathbf{T}(\mathbf{p};\boldsymbol{\Theta}))\right] \frac{\partial I_m}{\partial \mathbf{T}} \frac{\partial \mathbf{T}}{\partial \boldsymbol{\Theta}}$$

Current error at pixel location $\Delta I(\mathbf{p})$

Intensity gradient in moving image

Change in transformation wrt change in parameters

23

Computing Image Derivatives

- Many ways.
  - Simplest is pixel differences.

$$I_x \equiv \frac{\partial I}{\partial x} \approx \frac{I(x+\Delta x, y) - I(x,y)}{\Delta x}$$

$$I_y \equiv \frac{\partial I}{\partial y} \approx \frac{I(x, y+\Delta y) - I(x,y)}{\Delta y}$$

  - More sophisticated methods account for image noise
- Computed at each pixel

24

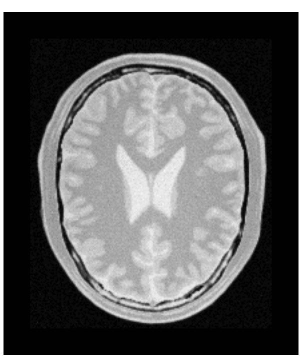
Derivative In Moving Image

- Equation

$$\frac{\partial I_m}{\partial \mathbf{T}}(\mathbf{p}) = \big(I_{mx}(\mathbf{T}(\mathbf{p};\boldsymbol{\Theta})) \quad I_{my}(\mathbf{T}(\mathbf{p};\boldsymbol{\Theta}))\big)$$

- In detail
  - Pre-compute derivatives in moving image $\mathbf{I}_m$
  - During minimization, map pixels back into moving image coordinate system and interpolate

Image Derivative Example



I_m

I_{mx}

I_{my}

dT/dθ

- Similarity transform:

$$\mathbf{T}(\mathbf{p};\boldsymbol{\Theta}) = \begin{pmatrix} ax - by + t_x \\ bx + ay + t_y \end{pmatrix}$$

  - Where

$$\boldsymbol{\Theta} = \begin{pmatrix} a & b & t_x & t_y \end{pmatrix}^T \qquad \mathbf{p} = (x, y)^T$$

- So derivative is 2x4 matrix (Jacobian):

$$\frac{\partial \mathbf{T}}{\partial \boldsymbol{\Theta}} = \begin{pmatrix} x & -y & 1 & 0 \\ y & x & 0 & 1 \end{pmatrix}$$

27

Putting It All Together

- At each pixel in overlap region:
  - Calculate intensity difference (scalar)
  - Multiply by 1x2 intensity gradient vector computed by mapping pixel location back to moving image
  - Multiply by 2x4 Jacobian of the transformation, evaluated at pixel location
  - Result is 1x4 gradient vector at each pixel
- Sum each component of vector over all pixels

28

Algorithm Outline

- Initialize transformation
- Repeat
  - Compute gradient
  - Make step in gradient direction
  - Update mapping equation
  - Remap image
- Until convergence

Initialization

- Since this is a minimization technique, an initial estimate is required,

$$\mathbf{\Theta}_0$$

- There are many ways to generate this estimate:
  - Identity transformation, e.g.

$$a = 1, b = 0, t_x = t_y = 0$$

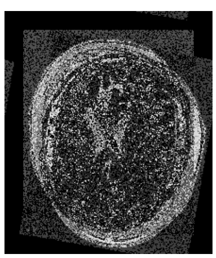
  - Prior information
  - Different technique
- Steepest descent only finds a local minimum of the objective function
- We'll revisit initialization in Lectures 16 and 17

Convergence

- Ideal is that gradient is **0.**
- In practice, algorithm is stopped when:
  - Step size becomes too small
  - Objective function change is sufficiently small
  - Maximum number of iterations is reached

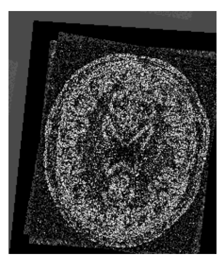
31

Example

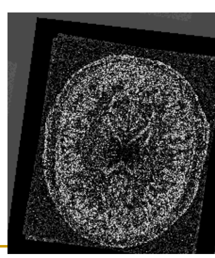


Initial errors

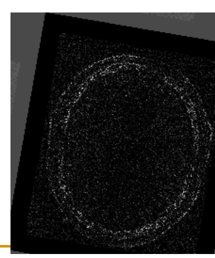
Iteration 100



Iteration 200



Iteration 300



Final: 498 iterations

32

Discussion

- Steepest descent is simple, but has limitations:
  - Local minima
  - Slow (linear) convergence

Summary

- Intensity-based registration is driven by direct functions of image intensity
- SSD is a common, though simple example
- Evaluating the SSD objective function (and most other intensity-based functions) requires image interpolation
- Gradient descent is a simple, commonly-used minimization technique
- Derivatives may be calculated using either numerical approximations or differentiation of the objective function.

Looking Ahead to Lecture 5

- Feature-based registration
- Topics:
  - Features
  - Correspondences
  - Least-squares estimation
  - ICP algorithm
  - Comparison to intensity-based registration

35