



Computer Aided Medical Procedures II

Feature-Based Registration

PCA and ICP

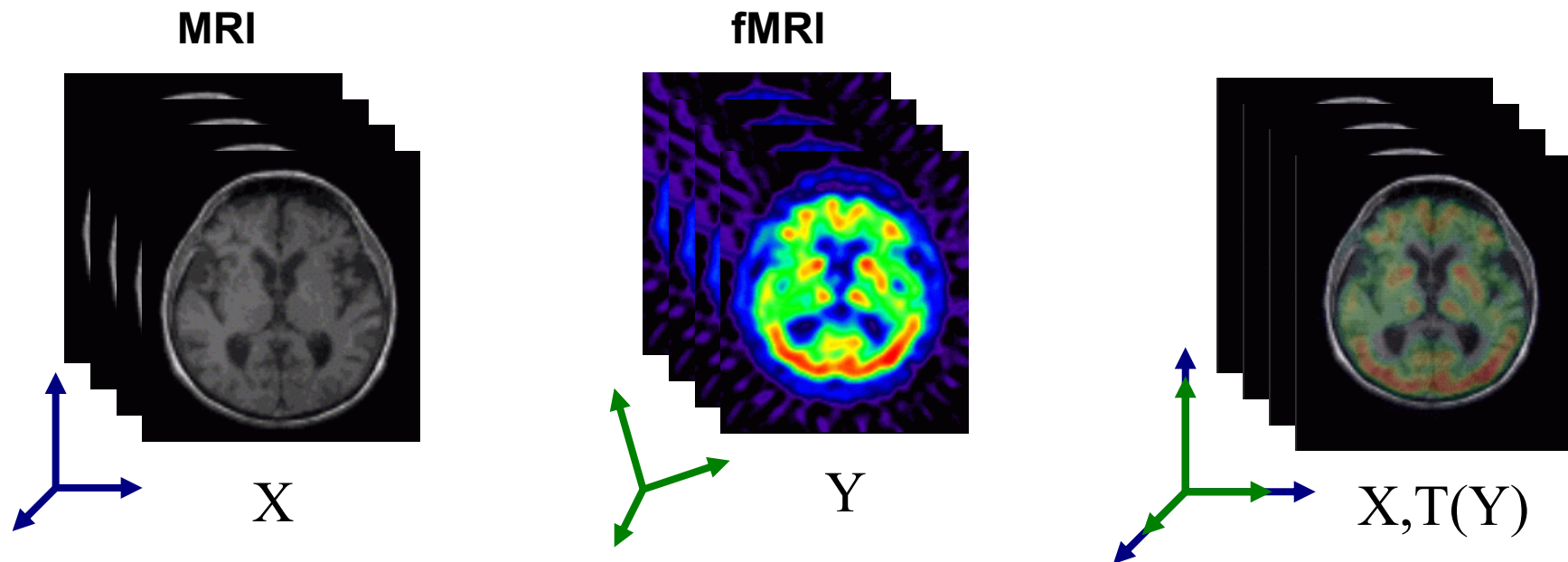
Martin Groher & Nassir Navab



Outline

- Short recall: Point-Based Registration
 - Point-Based Registration without correspondences: Principal Components Analysis (PCA)
 - Point-Based Registration with initial estimate: Iterative Closest Point (ICP)
-

Registration

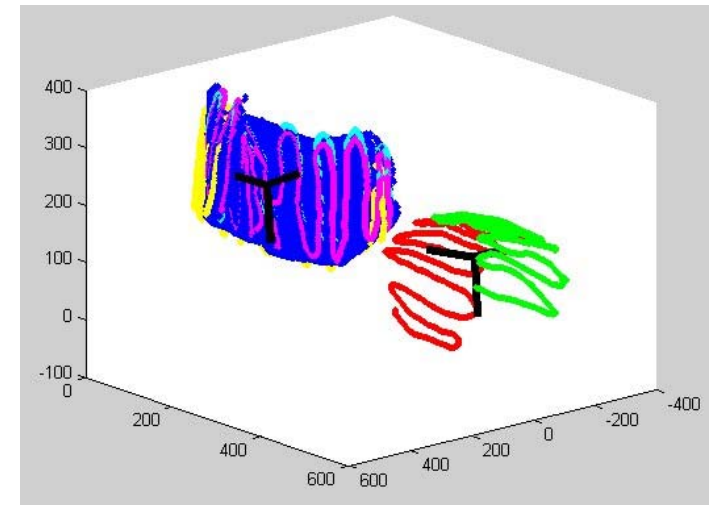


- Define Transformation T on one of the images
- Compare X , $T(Y)$ using either features or the full image content
- Refine T and repeat, until convergence criteria reached.

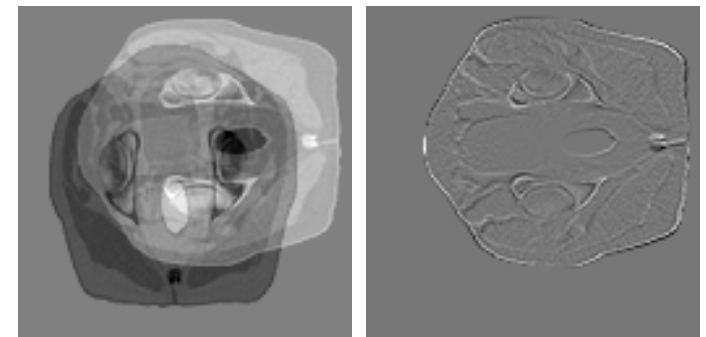
Registration Overview

- Feature-based methods
 - Point/line/curve/surface extraction
 - Registration of extracted entities only
 - Problem becomes one of **correspondences** and **distance**:
 - Corresponding Data available: Umeyama
 - No corresponding data available: PCA and ICP

- Intensity-based methods
 - Registration on pixel intensities only
 - Problem becomes one of **interpolation** and **similarity**:
 - How to get the right values to compare: Image Warping
 - How to compare the values in a meaningful way: Similarity Measures



Registration of point sets, see exercise

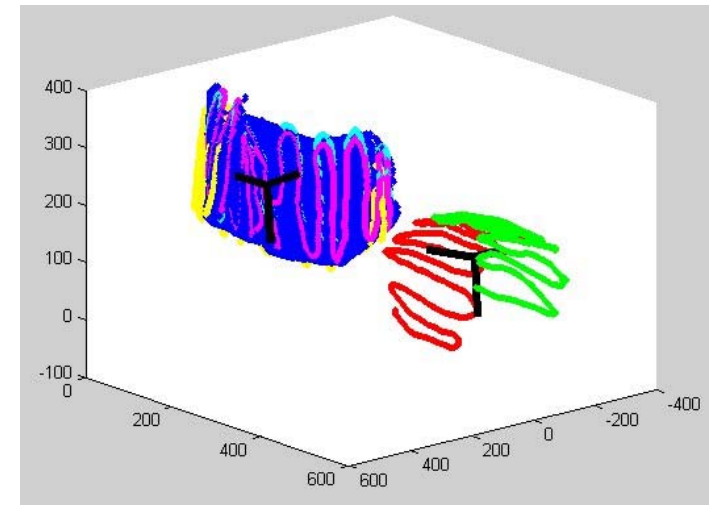


Registration of 2 CT slices, difference images

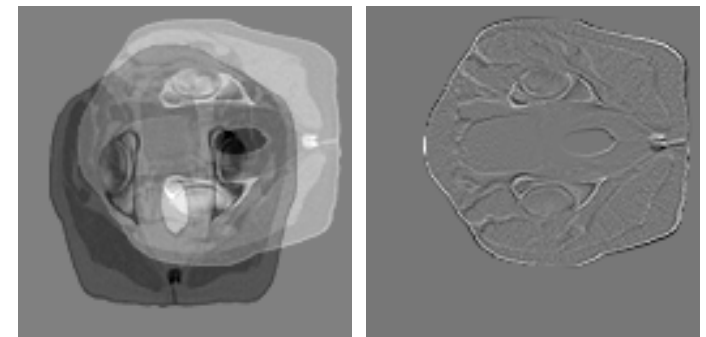
Registration Overview

- Feature-based methods
 - Point/line/curve/surface extraction
 - Registration of extracted entities only
 - Problem becomes one of **correspondences** and **distance**:
 - Corresponding Data available: Umeyama
 - No corresponding data available: PCA and ICP

- Intensity-based methods
 - Registration on pixel intensities only
 - Problem becomes one of **interpolation** and **similarity**:
 - How to get the right values to compare: Image Warping
 - How to compare the values in a meaningful way: Similarity Measures



Registration of point sets, see exercise



Registration of 2 CT slices, difference images

Corresponding 3D Point Sets – Problem

- Given: Two corresponding point sets $\{x_i\}, \{y_i\}$ in 3D
- Task: Find the rigid motion H between the point sets:

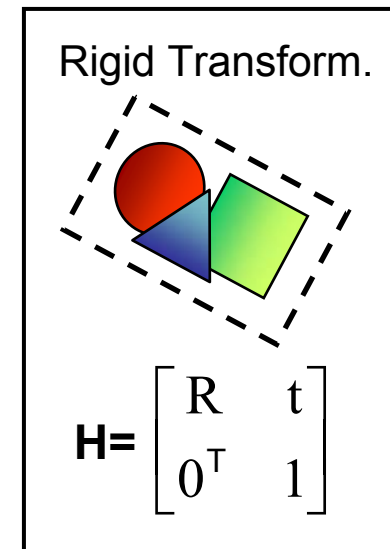
$$\forall_i : y_i = H x_i$$

$$Y_{4 \times N} = H_{4 \times 4} X_{4 \times N}$$

where H is a rigid 6-DOF transformation
→ known as *Procrustes Problem*

- In inhomogeneous coordinates:

$$\forall_i : y_i = R x_i + t$$



Corresponding 3D Point Sets – Umeyama Method

- Demean the point sets:

$$\begin{aligned}x'_i &= x_i - \bar{x} & y'_i &= y_i - \bar{y} \\ \forall_i : y'_i &= R x'_i & Y'_{3 \times N} &= R X'_{3 \times N}\end{aligned}$$

- Compute rotation with SVD of correlation matrix:

$$\begin{aligned}svd(X'Y'^T) &= UDV^T \\ R &= VU^T\end{aligned}$$

- Compute translation:

$$t = \bar{y} - R\bar{x}$$

Corresponding 3D Point Sets (2)

- Singular Value Decomposition:

$$V^T V = I$$

$$A_{m \times n} = U_{m \times m} D_{m \times n} V_{n \times n}^T \quad U^T U = I$$

- Special case for symmetric matrix:

$$X X^T = U D U^T$$

- Applied to our rotation problem:

$$Y = R X \rightarrow Y^T = X^T R^T \rightarrow X Y^T = X X^T R^T = U D \underbrace{U^T R^T}_{V^T}$$

$$\rightarrow R = V U^T$$

Corresponding 3D Point Sets - Summary

- Good method if correspondences are known
 - If no corresponding points are available (only two point clouds) method fails
 - combination PCA-ICP

 - PCA: use center of masses and principal components of point clouds for initial estimate of R, t

 - ICP: iterative solution to registration problem from initial estimate
-

3D Point Sets w/o Correspondences - PCA

- Demean the point sets:

$$\begin{aligned} x'_i &= x_i - \bar{x} & y'_i &= y_i - \bar{y} \\ \forall_i : y'_i &= R x'_i & Y'_{3 \times N} &= R X'_{3 \times N} \end{aligned}$$

- Compute principal components of both point sets (SVD of covariance matrices):

$$\begin{aligned} X'X'^T &= \overbrace{U_1}^{R_X} D_1 U_1^T & Y'Y'^T &= \overbrace{U_2}^{R_Y} D_2 U_2^T \\ && \rightarrow R &= R_Y R_X^T \end{aligned}$$

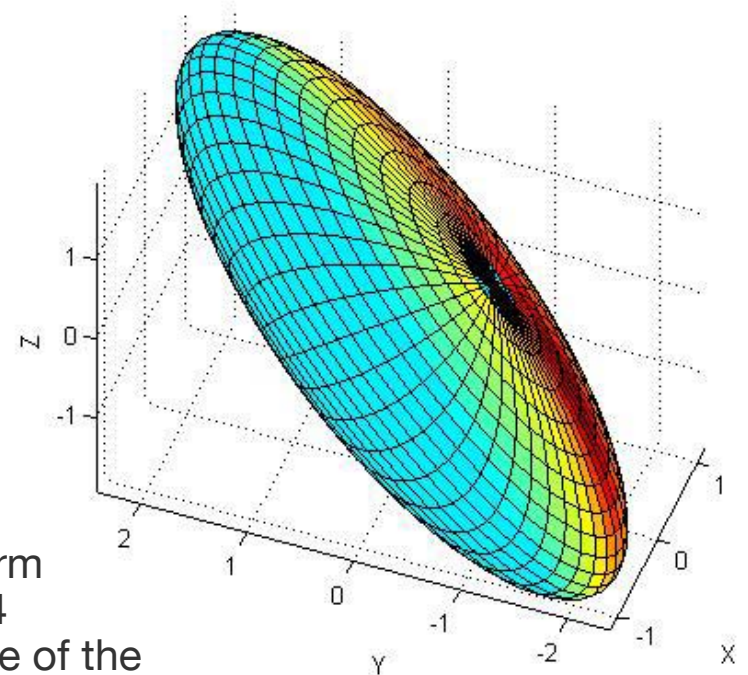
Different from
Umeyama method

- Compute translation:

$$t = \bar{y} - R\bar{x}$$

Registration based on PCA - Summary

- Pro:
 - Analytical solution: Fast
 - No correspondences needed
- Con:
 - Registration not unique!
Principal components with their magnitudes form ellipsoid. An interpretation as rotations leaves 4 directions of the axes without altering the shape of the ellipsoid
 - Registration not very accurate
→ can be used as initialization for another registration algorithm, e.g. Iterative Closest Point (ICP)



Point Based Registration - Summary

- Accurate if corresponding information available
- If no correspondences are given only rough initial estimate via PCA
- How to improve the estimate?

→ iterative methods

→ robust methods

→ using of initial estimate



Iterative Closest Point

[Besl, Mccay `92], [Zhang `92]

Iterative Closest Point

Algorithm Outline:

INPUT: Two spatial entities (point sets, surfaces, curves) E_1, E_2

OUTPUT: Transformation R, t , such that $E_2 = RE_1 + t$

INIT: $\hat{E}_1 := E_1$

1. $\forall p \in E_2$: Determine the closest points in $\hat{E}_1 \rightarrow$ Match $M_{E_1 E_2}$
2. Update the Matching $\hat{M}_{E_1 E_2}$ using a statistical analysis to get rid of outliers \rightarrow Match $\hat{M}_{E_1 E_2}$
3. Assume the points in $\hat{M}_{E_1 E_2}$ correspond and estimate the motion R, t using some linear analytical method, e.g. the Umeyama method
4. Apply the motion R, t to all points in E_1 and repeat until convergence

1. $\forall p \in E_2$: Determine the closest points in $E_1 \rightarrow$ Match $M_{E_1 E_2}$

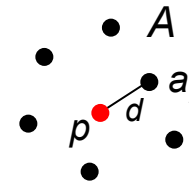
- Euclidian distance between point $p_1=(x_1, y_1, z_1)$ and $p_2=(x_2, y_2, z_2)$

$$d(p_1, p_2) = \|p_1 - p_2\| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

- Point to basic geometric entity distance

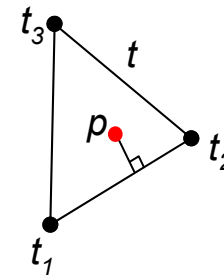
- point set $A=\{a_i\}, i=1, \dots, N_A$

$$d(p, A) = \min_{I \in \{1, \dots, N_A\}} d(p, a_i)$$



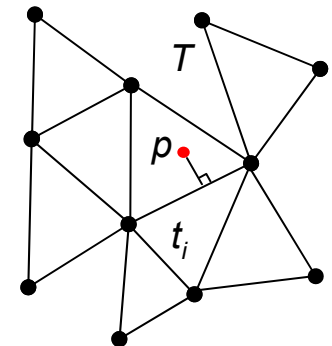
- triangle t defined by t_1, t_2 , and t_3

$$d(p, t) = \min_{u+v+w=1} \|ut_1 + vt_2 + wt_3 - p\|, u, v, w \in [0, 1]$$



- triangle set $T=\{t_i\}, i=1, \dots, N_T$

$$d(p, T) = \min_{I \in \{1, \dots, N_T\}} d(p, t_i)$$



2. Update the Matching $M_{E_1 E_2}$ using a statistical analysis to get rid of outliers → Match $\hat{M}_{E_1 E_2}$

- Determine the mean and sample deviation of all distances:

$$\mu = \frac{1}{N} \sum_{i=1}^N d_i \quad \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (d_i - \mu)^2}$$

- Depending on these values, reset the distance threshold for outlier detection

$$D_{max}^I = \mu + 3\sigma;$$

$$\text{else if } \mu < 3D$$

$$D_{max}^I = \mu + 2\sigma;$$

$$\text{else if } \mu < 6D$$

$$D_{max}^I = \mu + \sigma;$$

else

$$D_{max}^I = m_{median \text{ of all distances}}$$

endif

Reg good

Reg still good

Reg not too bad

Reg really bad

- Reject all outliers from the previously acquired set $M_{E_1 E_2}$

2. Update the Matching $M_{E_1 E_2}$ using a statistical analysis to get rid of outliers \rightarrow Match $\hat{M}_{E_1 E_2}$

- How to choose the distance threshold D ?
 \rightarrow heuristics, i.e. play around 😊
- The value of D is crucial:
 - D very small: maybe no initial transformation estimate (all point correspondences are discarded as outliers)
 - D very big: maybe convergence against a wrong transformation (too many outliers considered as good matches)
 - Find better solution: Depends on the shape of the input data (curve, surface, point cloud, etc.)

3. Assume the points in $\hat{M}_{E_1 E_2}$ correspond and estimate the motion R, t using some linear analytical method, e.g. the Umeyama method

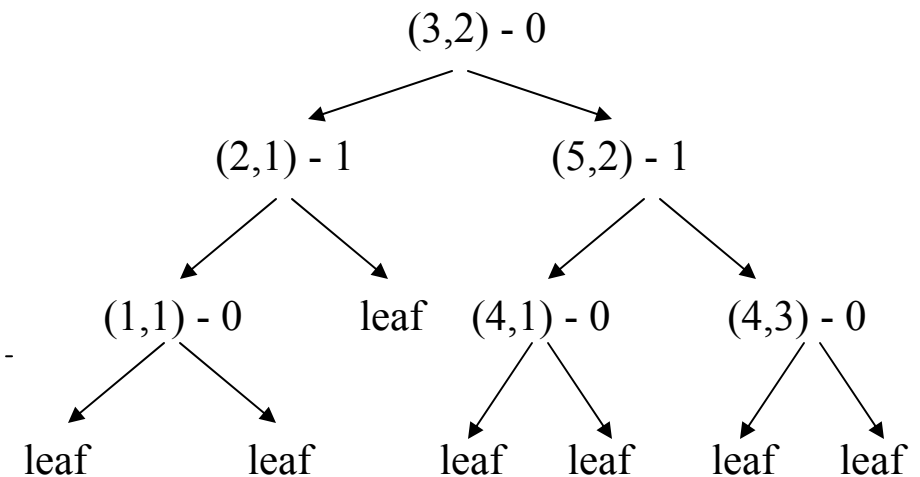
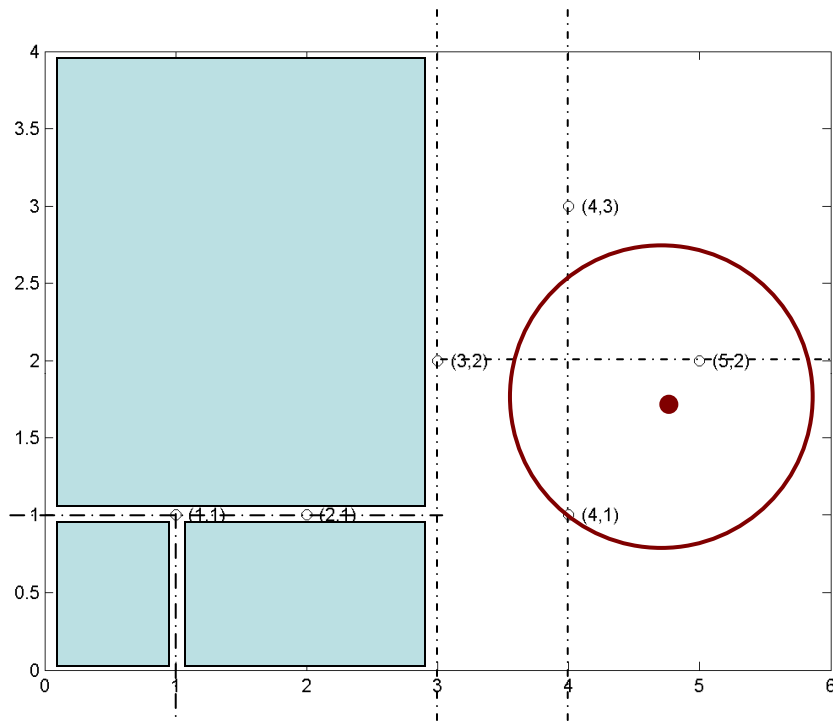
- We take any analytical solution to the pose problem
- As before, Umeyama method could be taken
- Other methods also possible
- This makes the ICP algorithm modular:
Just use a suitable estimation algorithm and you can do 2D-2D, 2D-3D, 3D-3D registration depending on the input data

4. Apply the motion R, t to all points in E_1 and repeat until convergence

- When to converge?
- As in many iterative optimization schemes, the difference in cost function evaluations is deciding on termination
- t_{old} and t_{new} are translations from previous and current iteration.
- α_{old} and α_{new} are axes of rotation from previous and current iteration.
- If $((t_{old} - t_{new}) < \epsilon \ \&\& \ \sqrt{(\alpha_{old} - \alpha_{new})^T (\alpha_{old} - \alpha_{new})} < \epsilon)$ terminate.

Practical Considerations: KD-Trees for Efficient Point Set Structuring

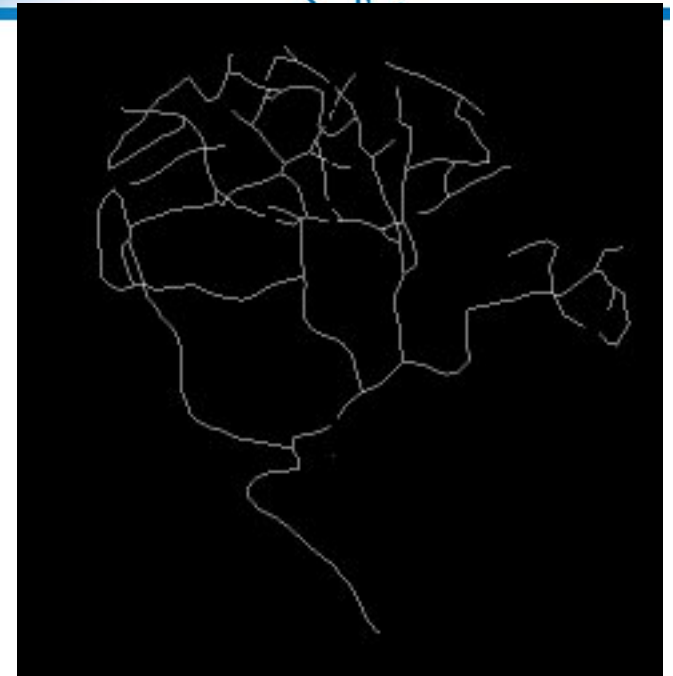
- Successively separate the space in each dimension



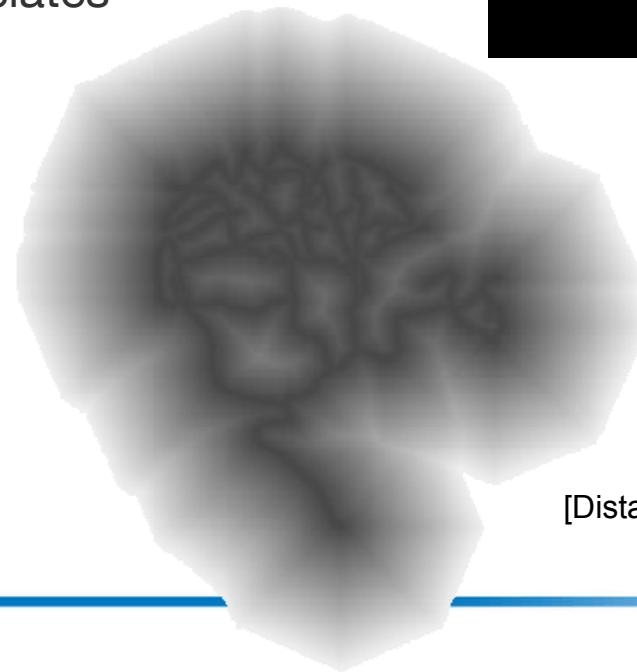
→ Querying for closest point
becomes a local problem!

Practical Considerations: Distance Transforms

- Use distance map for getting closest points
- Computing closest points becomes a problem of map-lookup (interpolation)
- Sampling problem now correlates with the grid sampling in the distance map

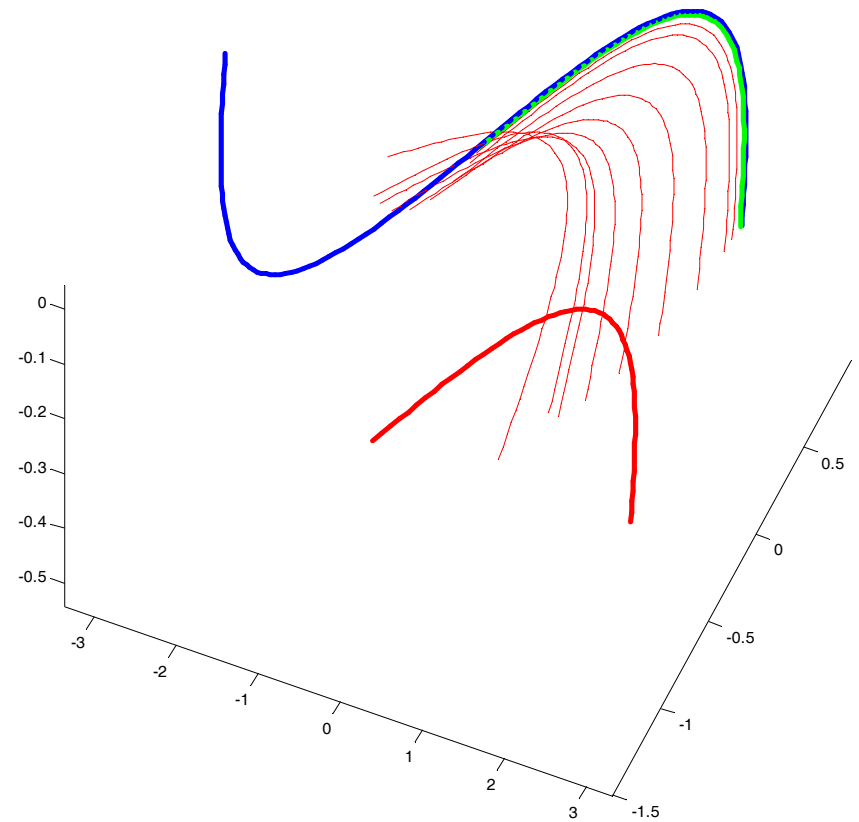
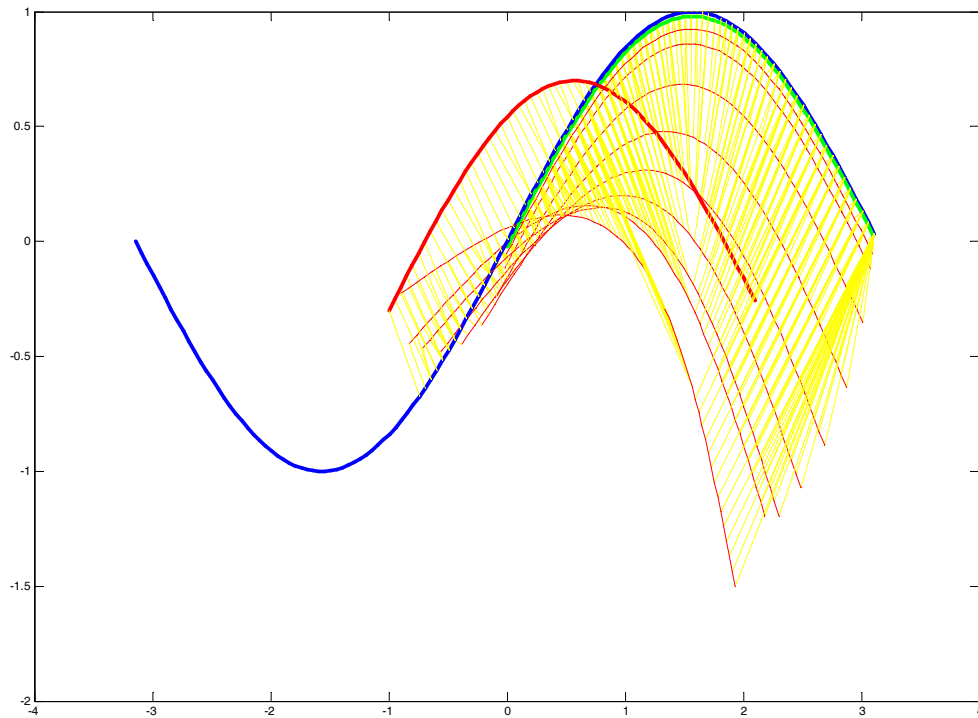


[curves]

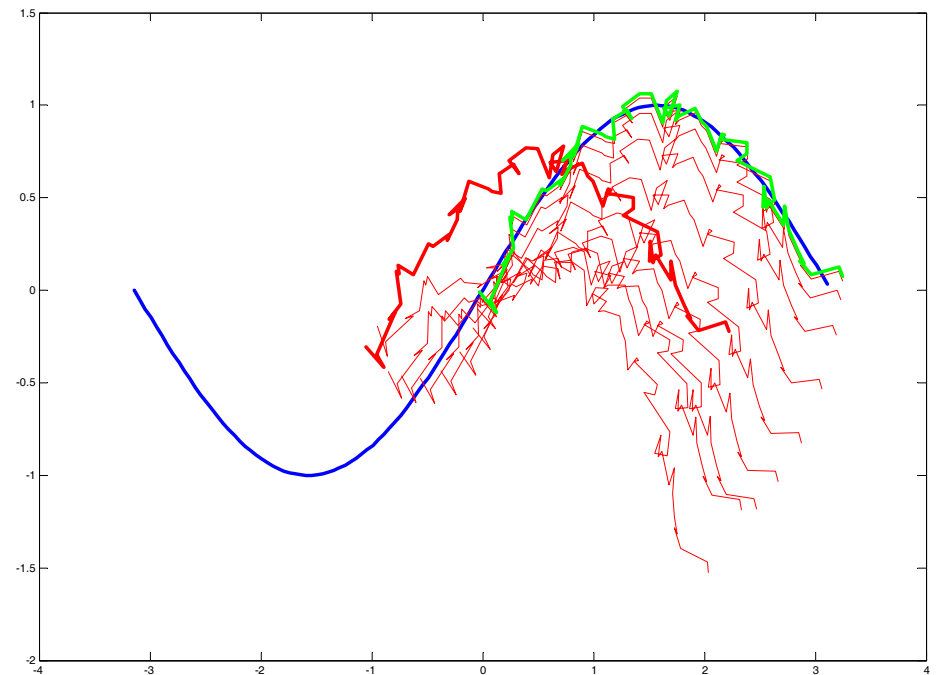
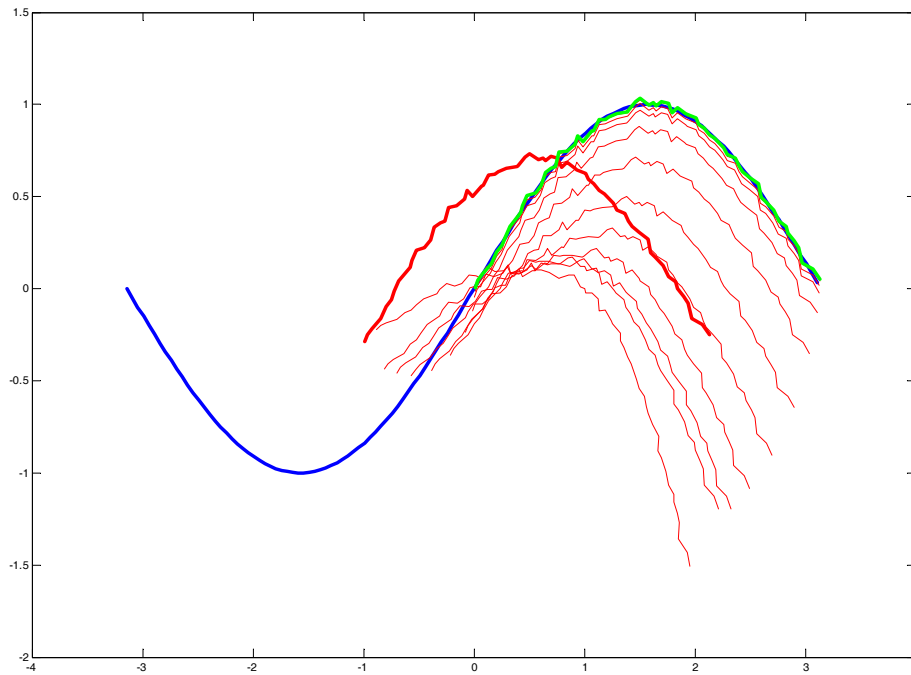


[Distance transform of curves]

Iterative Closest Point: Example

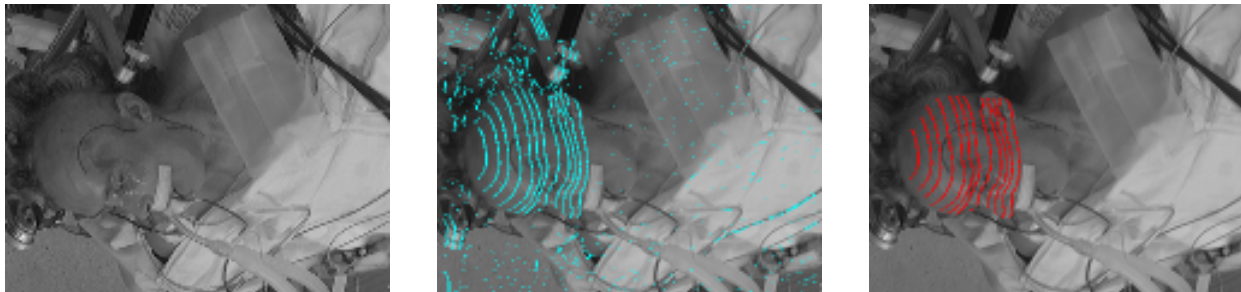


Iterative Closest Point: Robustness



Application: Intra-operative navigation

- 3D surface of the patient's skin is obtained with laser scanner



- ICP aligns it with the skin surface from a preoperative data set, e.g. Magnetic Resonance (MR) or Computed Tomography (CT).



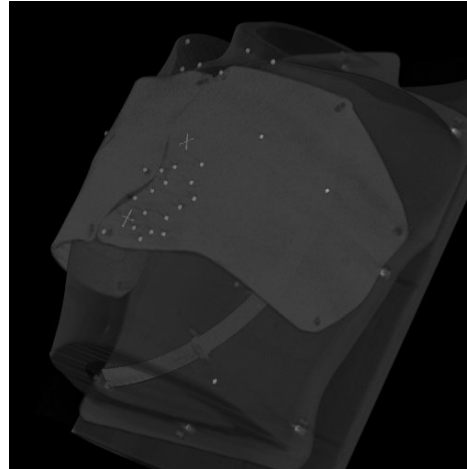
References on Feature-Based Registration

- [1] Hajnal, Hill, Hawkes – *Medical Image Registration*
- [2] Maintz, Viergever - *An Overview of Medical Image Registration Methods*
- [3] Z. Zhang - *Iterative Point Matching for registration of free form curves*, March 1992
- [4] Paul J. Besl and Neil D. McKay - *A Method for Registration of 3-D Shapes*, 1992
- [5] A. Moore - *An introductory tutorial on kd-trees*, Technical Report No. 209, Computer Laboratory, University of Cambridge, 1991.
http://www.ri.cmu.edu/pubs/pub_2818.html
- [6] Lindsay I. Smith, *A Tutorial on Principal Components Analysis*, Feb 2002

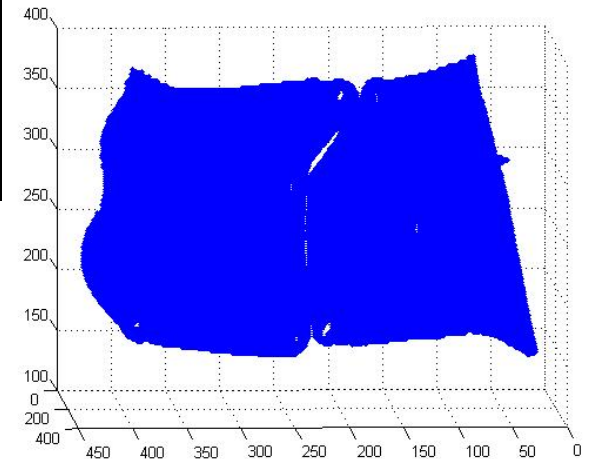
Some slides are from presentations by Hesam Najafi.

Assignment

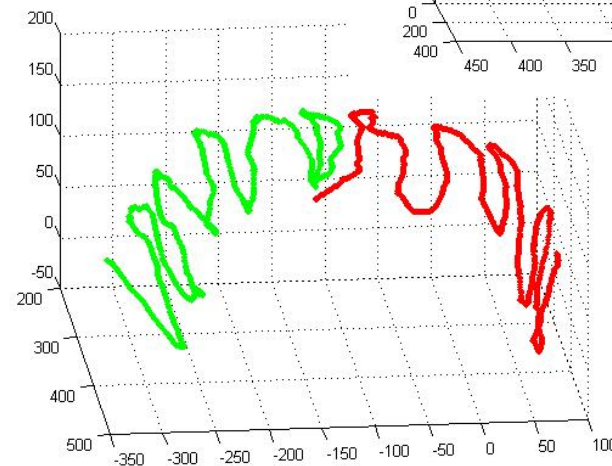
1. CT scan of phantom data



2. Extracted surface represented as point cloud



3. Tracking data from surface



4. Match surface and tracking data

Optimization Methods, 2D-3D Registration, and Validation

Martin Groher

agenda

1. Non-linear Optimization Algorithms

Best Neighbor

Powell-Brent

Simplex

Gradient Descent

2. 2D/3D Registration Methods

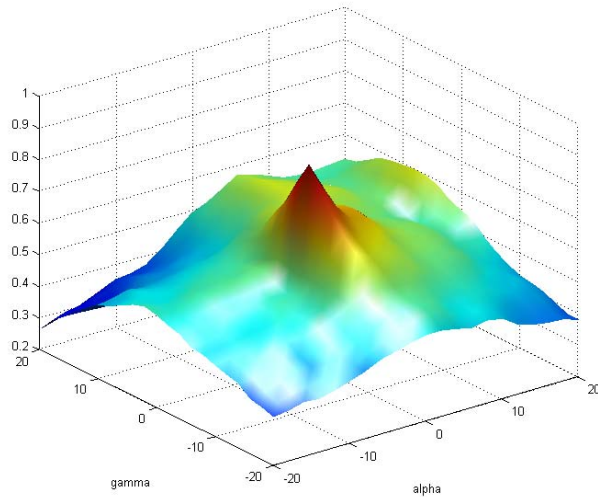
X-Ray Projection Physics

DRR Generation Methods

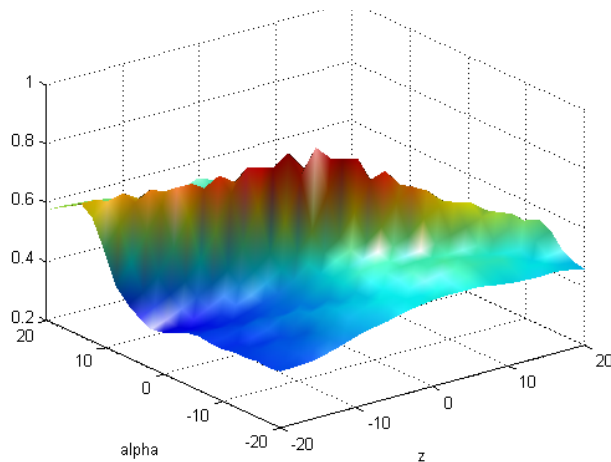
Adapted Similarity Measures

3. Validating Registration & Conclusion

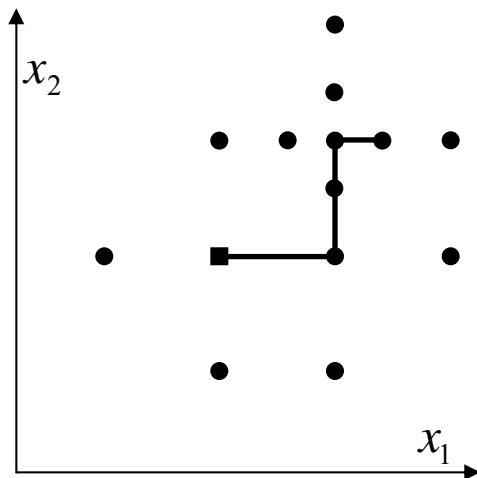
Non-linear Optimization



- Function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ to be minimized
- Direct search: Simplex, Best Neighbor, Powell-Brent
- Gradient information available: Gradient Descent, Steepest Descent, Newton



Best Neighbor Optimizer

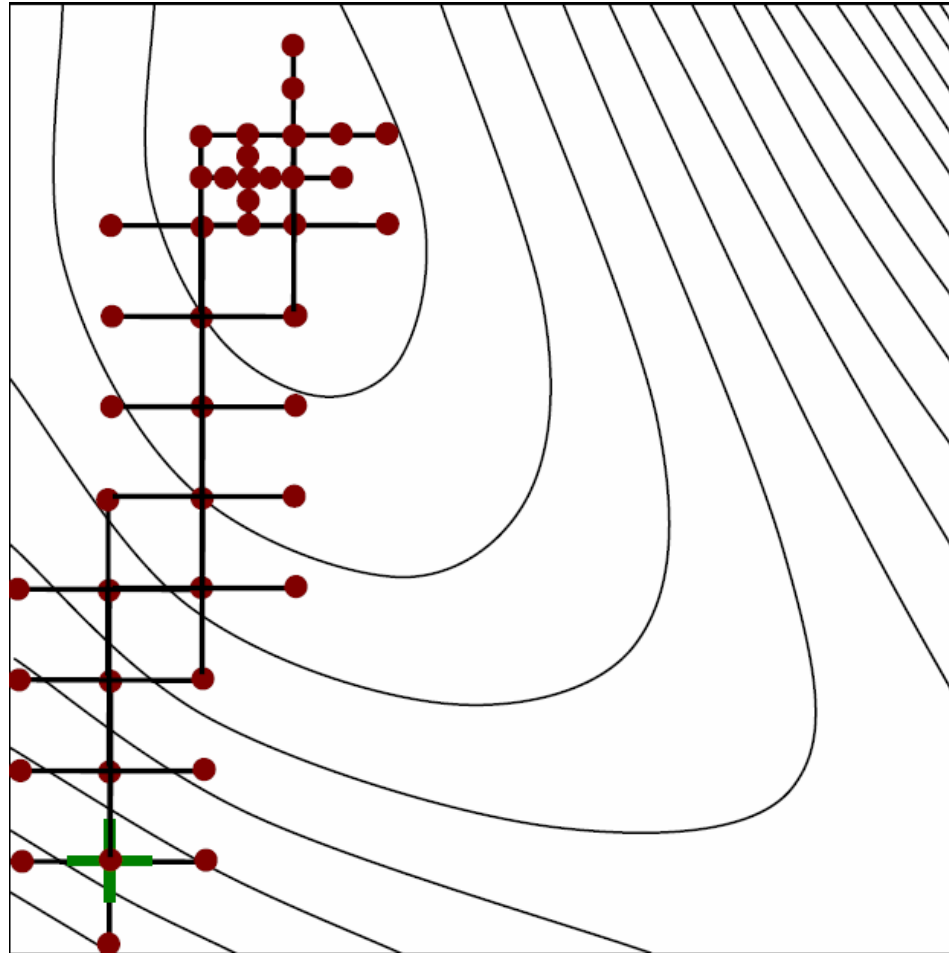


1. Evaluate all $2n$ neighbors of the current location
2. If better than the current value, set best neighbor as next location, repeat 1.
3. Otherwise half the step size and try again
4. Abort if stepsize or cost function difference below limit

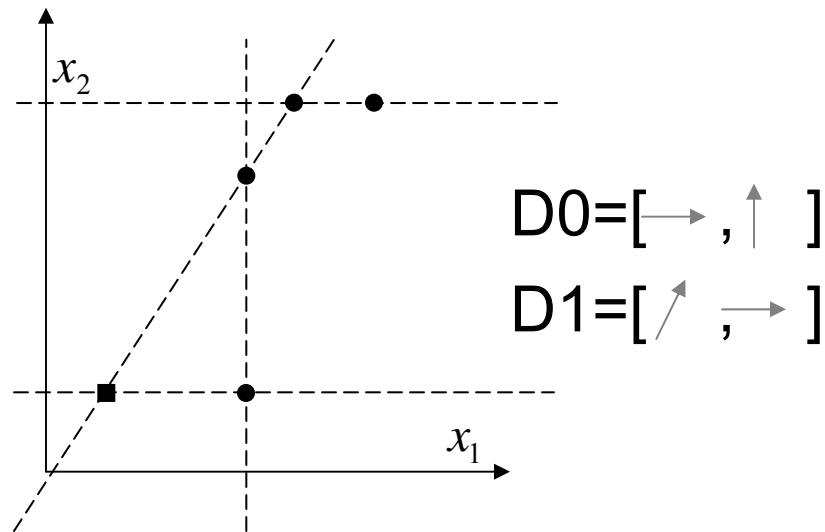
Works very stable – only locations are chosen that have been evaluated

Best Neighbor Optimizer (2)

Applied to two parameters of a registration problem



Powell-Brent Optimizer



Each Iteration:

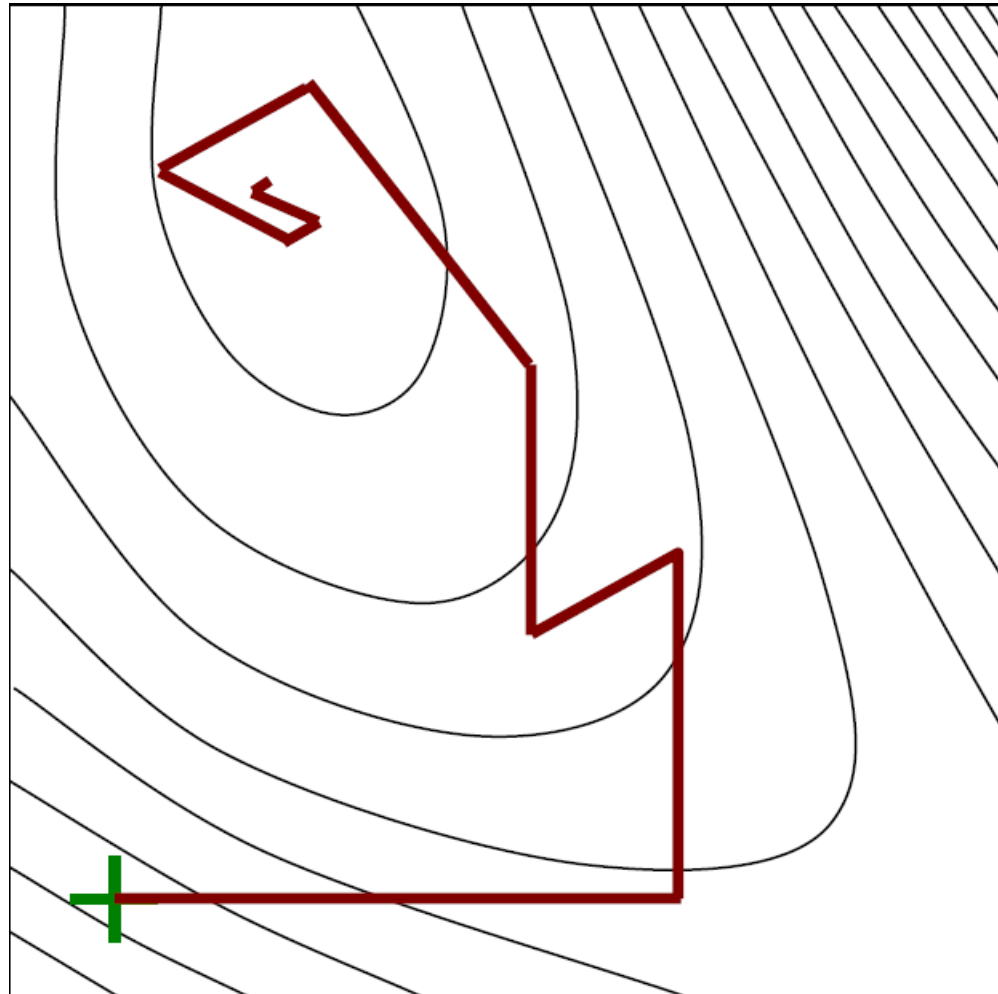
- n successive line minimizations
- Adaptation of the direction set

Advantages:

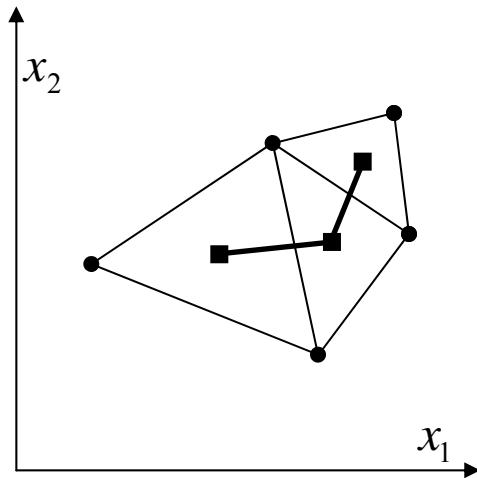
- No gradient information needed
- Few cost function evaluations

Powell-Brent Optimizer (2)

Applied to two
parameters of a
registration
problem



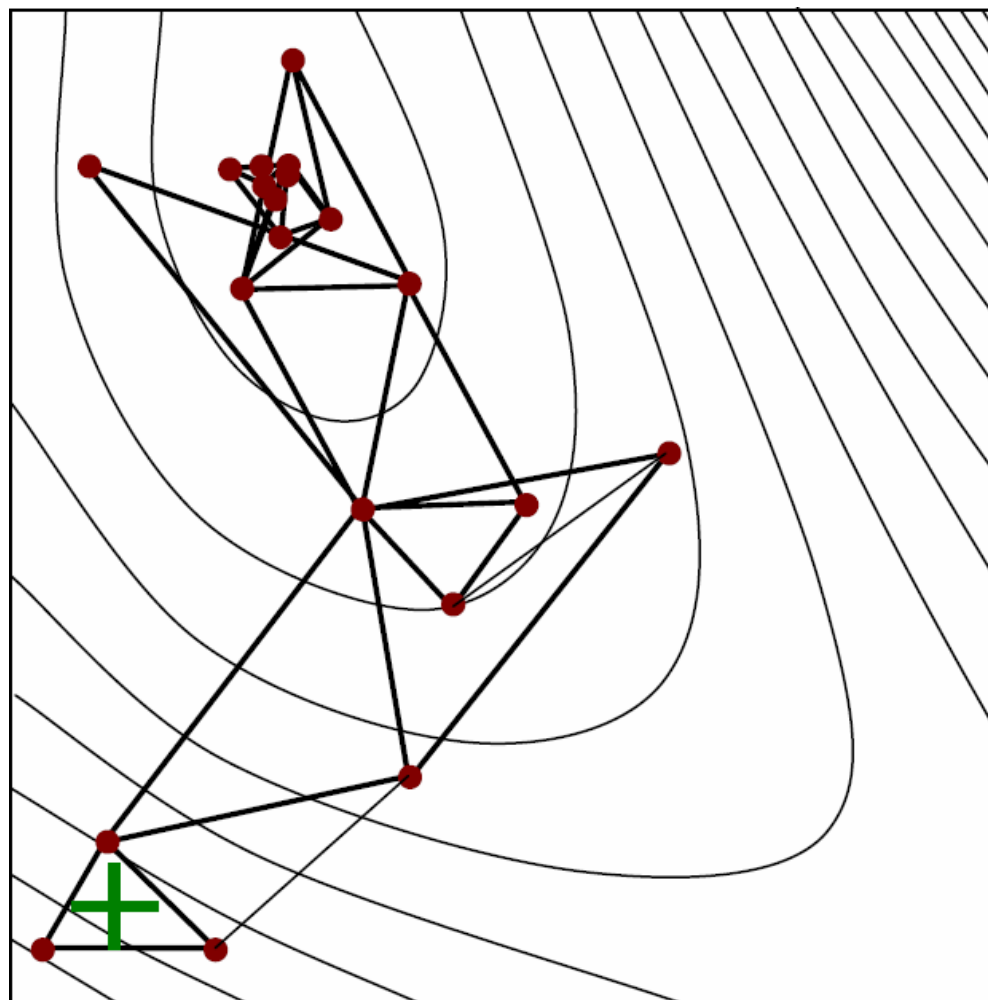
Simplex Optimizer



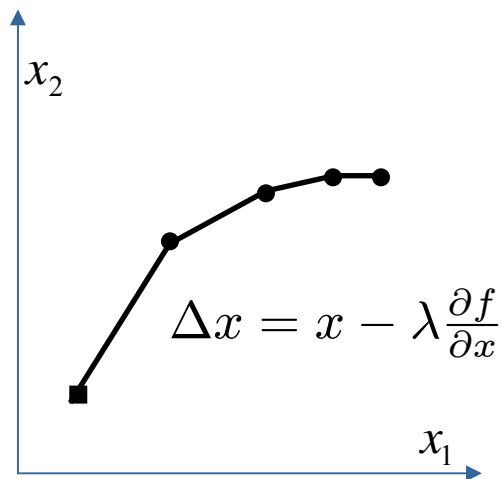
- Simplex: Minimal geometric shape in n dimensions, $n+1$ vertices
- Position and shape of Simplex is altered during optimization
- Very few cost function evaluations
- Have a look at [2]

Simplex Optimizer (2)

Applied to two parameters of a registration problem



Gradient Descent Optimizer



- Update equation for each optimizer

$$x_{new} \leftarrow x + \lambda \Delta x$$

→ why not use gradient in Δx computation?

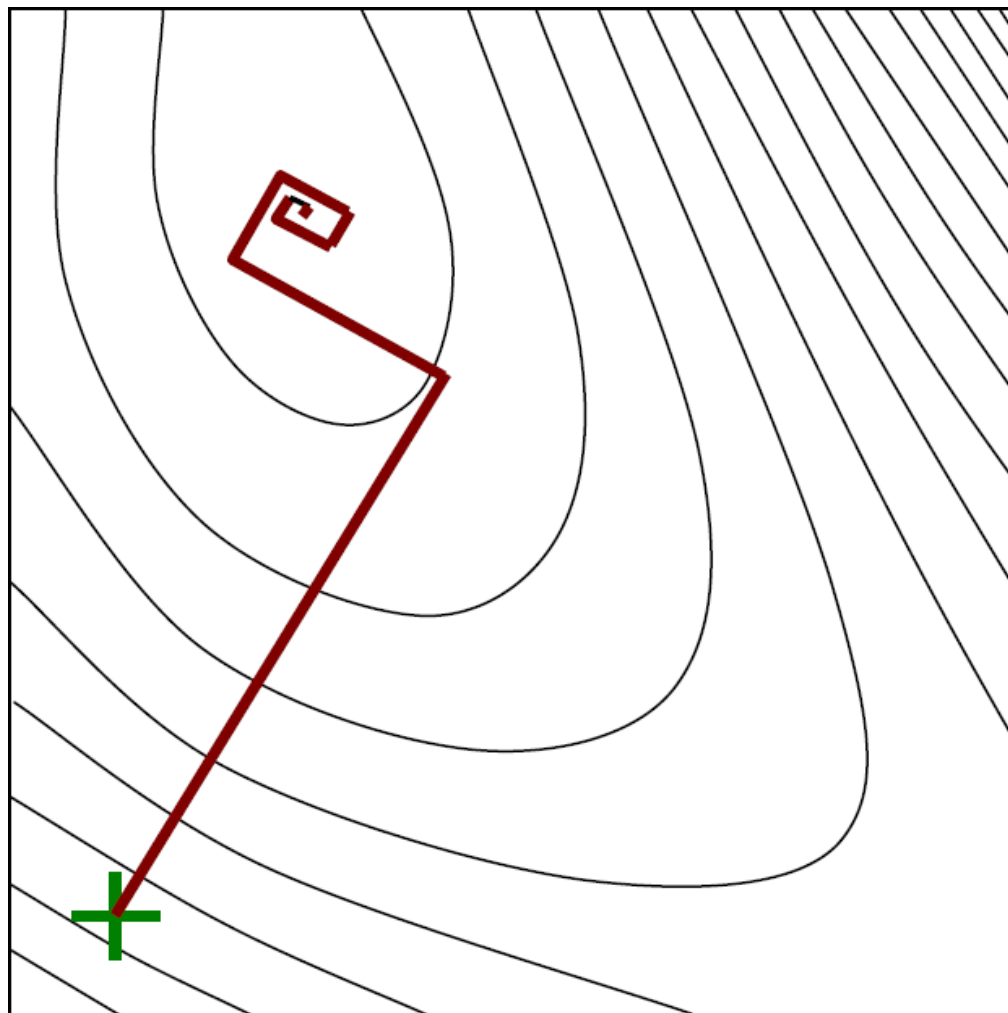
- Steepest Descent - compute Δx from 1st derivative:

$$\Delta x = - \frac{\partial f}{\partial x}$$

- Gradient delivers direction of steepest ascent → fast convergence
- How to determine λ ? Line search!
- Gradient Descent Optimizer are slow due to line search for λ and gradient computation
- Even faster (quadratic) convergence when involving higher-order derivatives: Newton

Gradient Descent Optimizer (2)

Applied to two parameters of a registration problem



agenda

1. Non-linear Optimization Algorithms

Best Neighbor

Powell-Brent

Simplex

Gradient Descent

2. 2D/3D Registration Methods

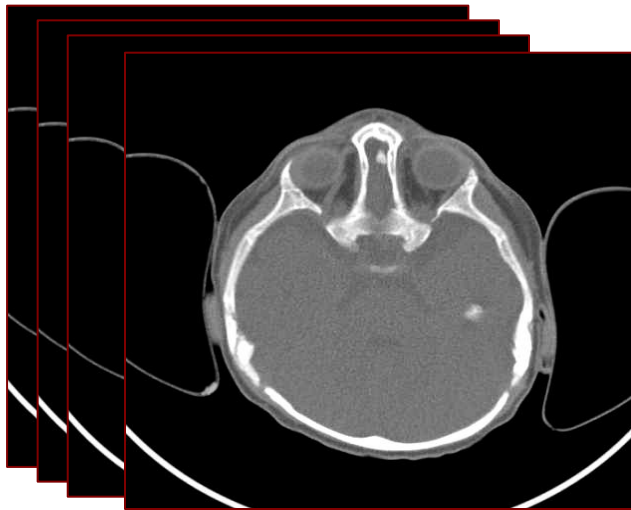
X-Ray Projection Physics

DRR Generation Methods

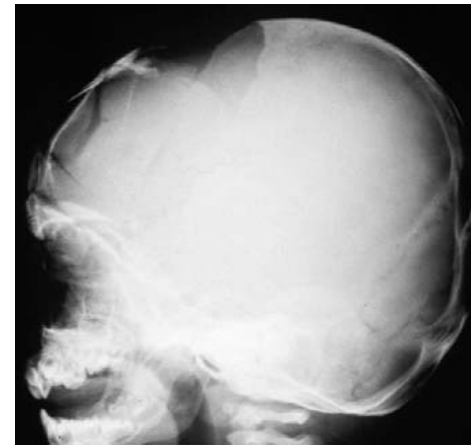
Adapted Similarity Measures

3. Validating Registration & Conclusion

2D-3D Registration



cross-sectional CT slices
→ 3D volume

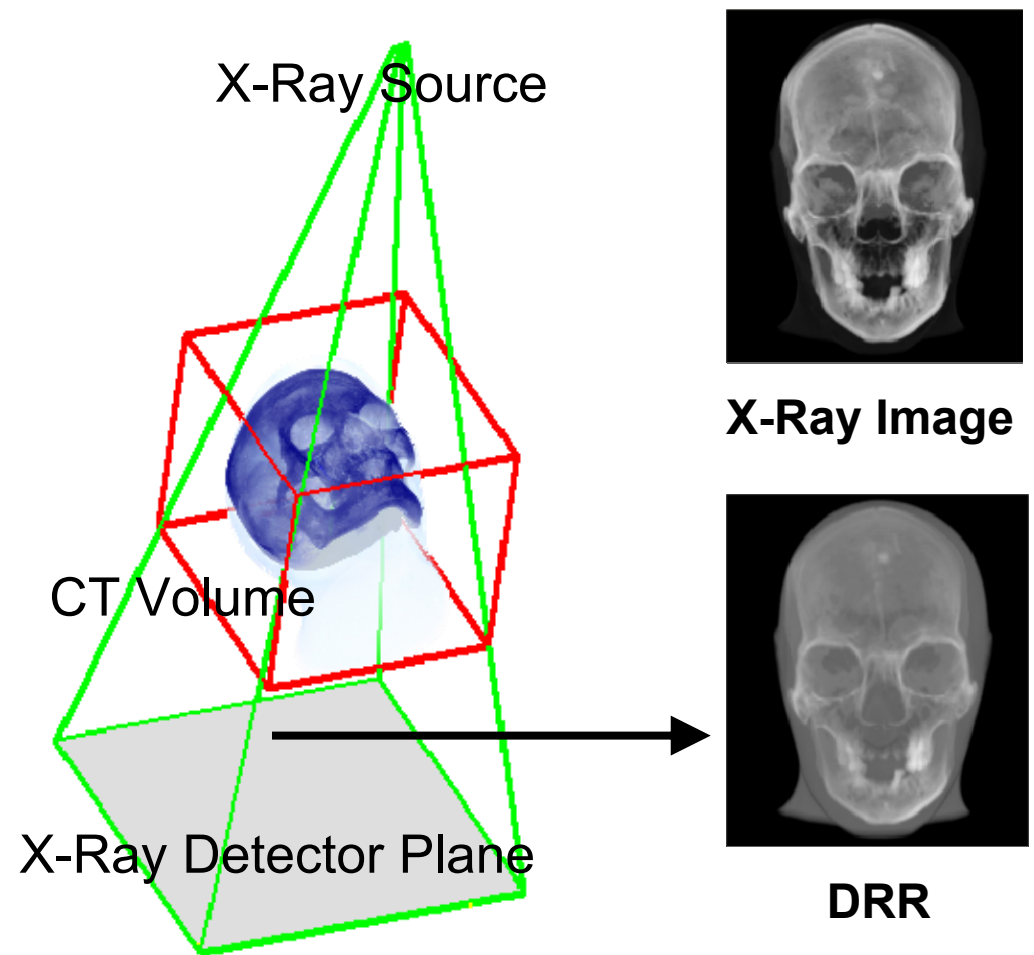


X-Ray data
→ 2D projective image

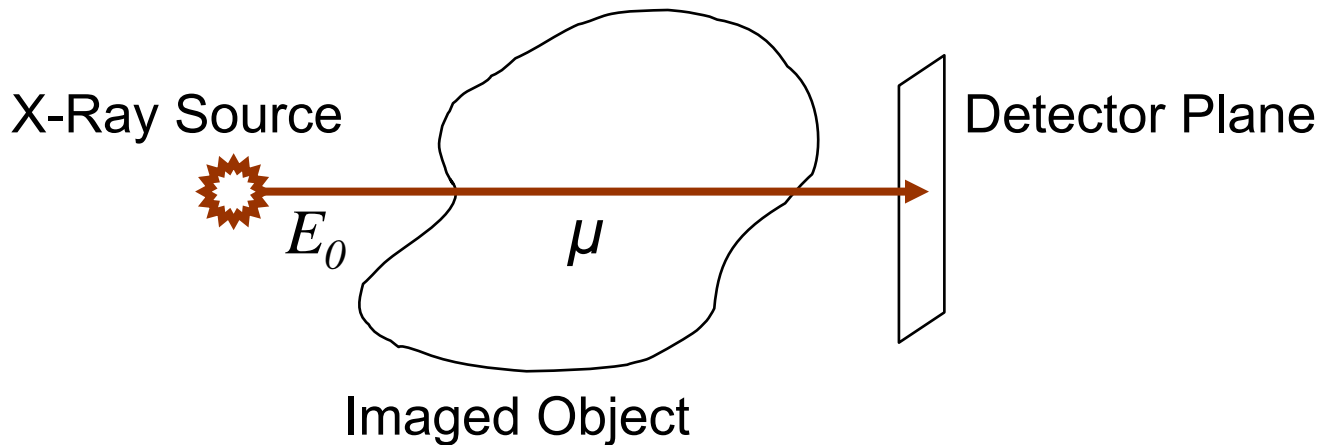
- How to register a 2D projective image with a 3D volume?

2D-3D Registration

- Image information is the same, X-Ray attenuation values
- Integrating along rays of the volume simulates X-Ray images
→ **Digitally Reconstructed Radiograph (DRR)**



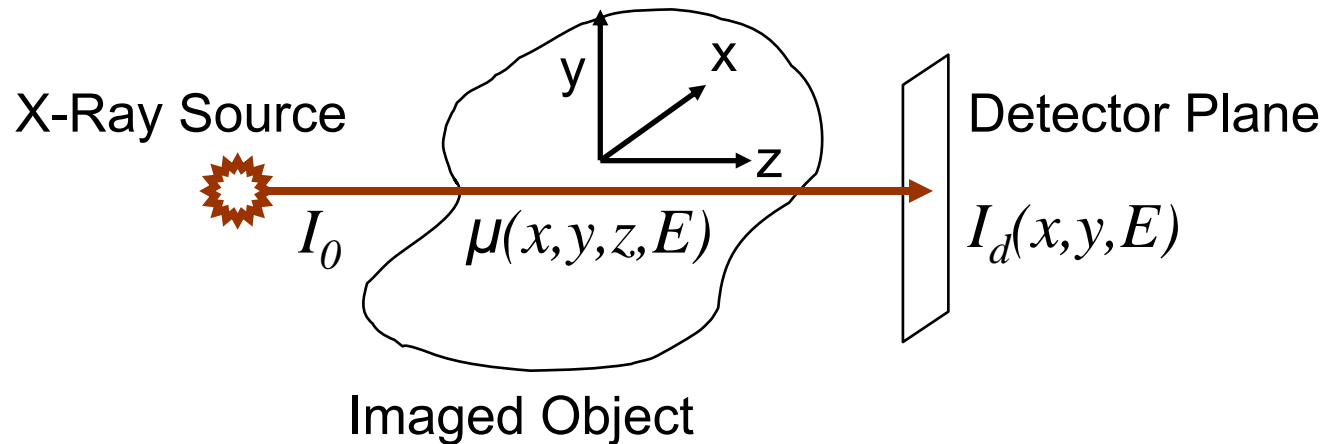
X-Ray Projection Physics



- Different materials interact with emitted energy of an X-ray beam in a different way
- If material interacts with energy, part of the energy is lost (attenuation) according to the material's radio density (measured with the attenuation coefficient μ)
- Attenuation can be formulated by Lambert-Beer law:

$$E = E_0 e^{-\int \mu x dx}$$

X-Ray Projection Physics



Intensity proportional to emitted energy:

$$\mathcal{I} = \frac{E}{A}$$

$$\mathcal{I}_d \propto I_0 e^{-\int \mu z dz}$$

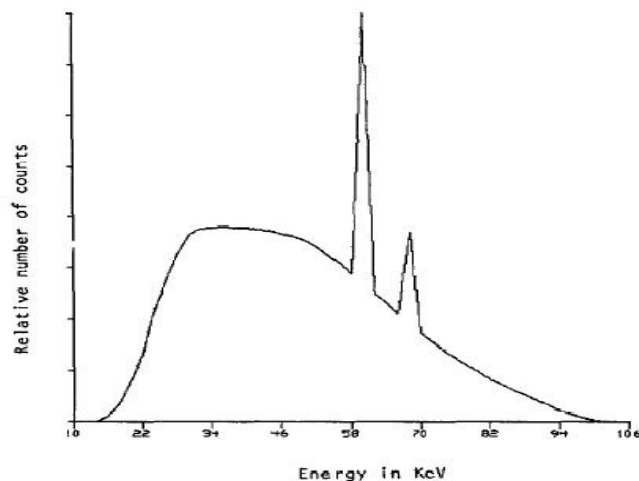
μ not constant, but dependant on position and energy:

$$\mathcal{I}_d(x, y, E) = I_0 e^{-\underbrace{\int \mu(x, y, z, E_0) dz}_{\text{computation from CT voxels!}}}$$

computation from CT voxels!

X-Ray Projection Physics

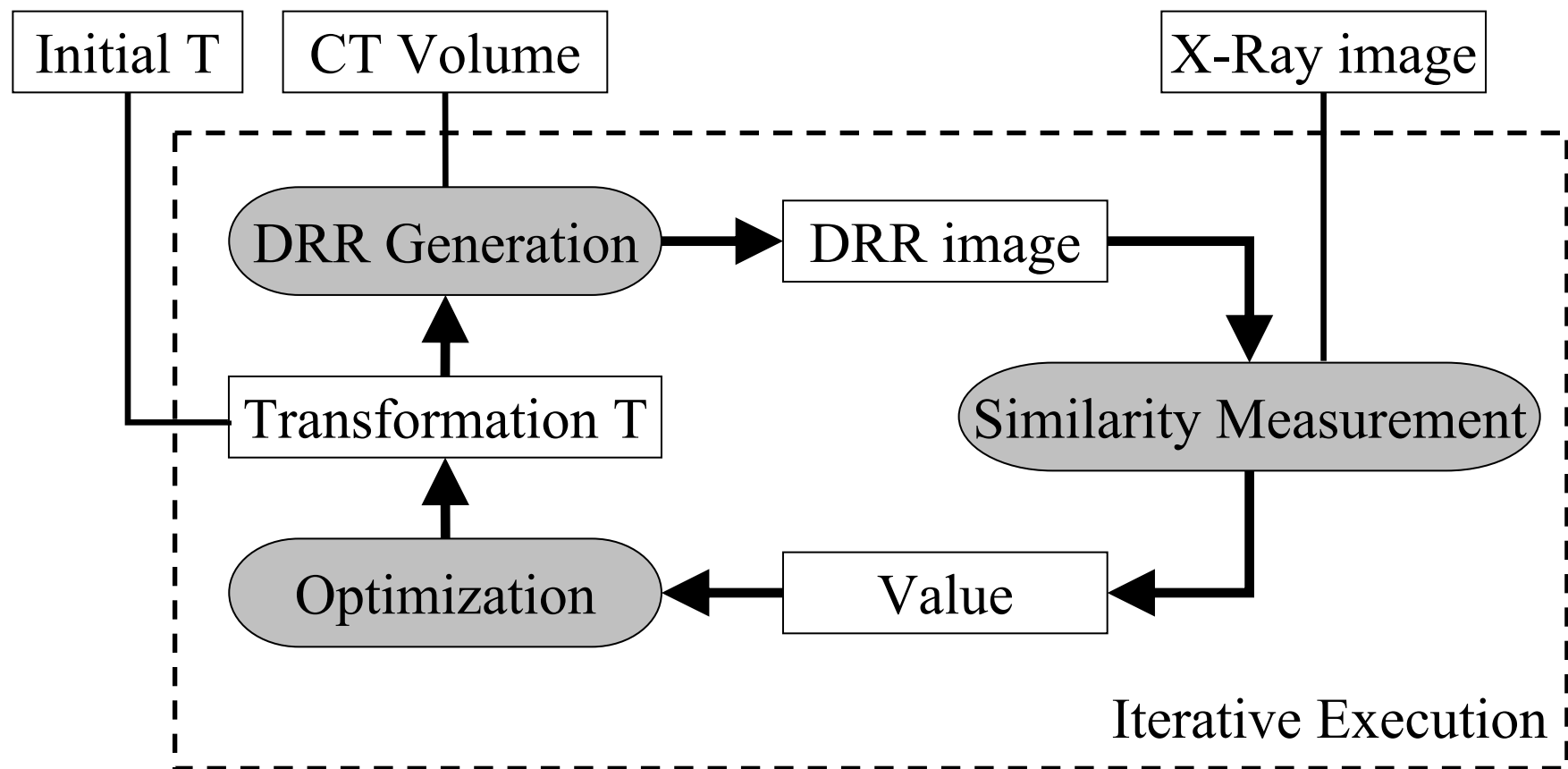
- E_0 represents an effective Energy of the X-Ray source \rightarrow approximation
- Both effective energies & characteristic spectrum differ for CT and X-Ray devices \rightarrow DRRs will always be slightly different from real X-Rays



correct equation:

$$I_d(x, y) = \int_0^{E_{\max}} I_0(E) e^{-\int \mu(x, y, z, E) dz} dE$$

2D-3D Registration – Iterative Scheme



2D-3D Registration - Challenges

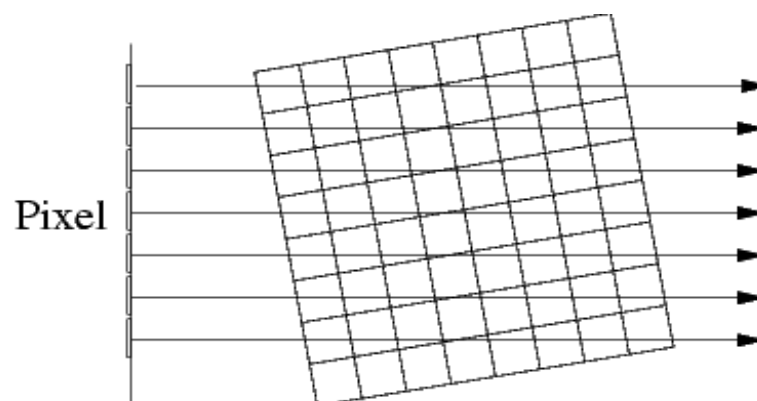
- **DRR Generation:**
 - Computationally expensive, make it as fast as possible
 - At the same time good quality is needed
 - **Similarity Measurement:**

Images originate from 2 different modalities – measure should be stable nevertheless. Structures that are present in only one image should not affect the measure too much
 - **Optimization Algorithm:**
 - Finding the best pose with as few DRRs as possible
 - Do not get stuck in local optima
-

DRRs – Volume Rendering

Ray Casting:

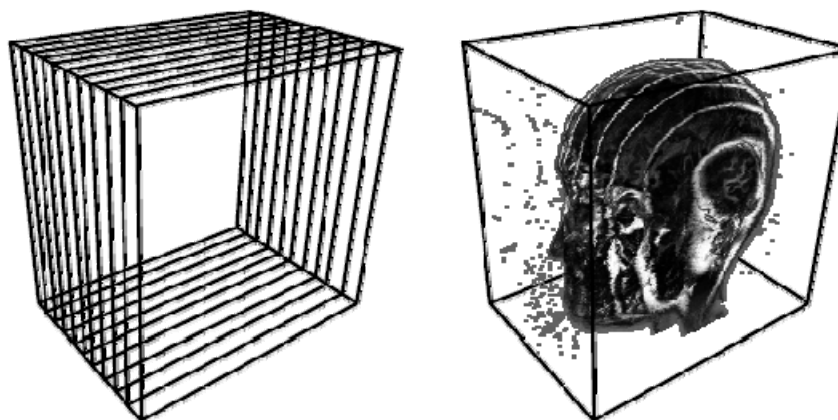
- Directly implements line integrals by summing up over voxels
- Too slow, no interactive frame rates



DRRs – Volume Rendering

2D Texture:

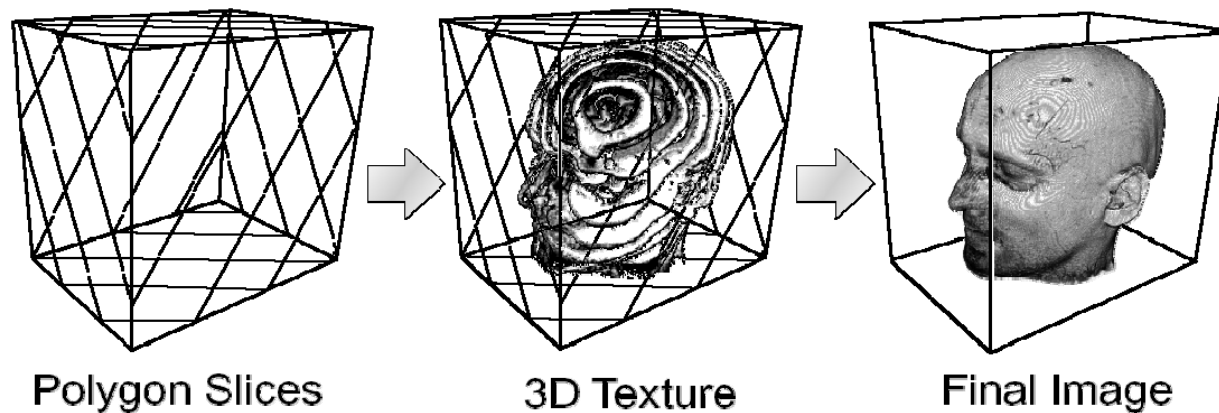
- Hardware-Accelerated Rendering
- GPU draws transparent slice textures, oriented axially
- 3 sets of textures necessary



DRRs – Volume Rendering

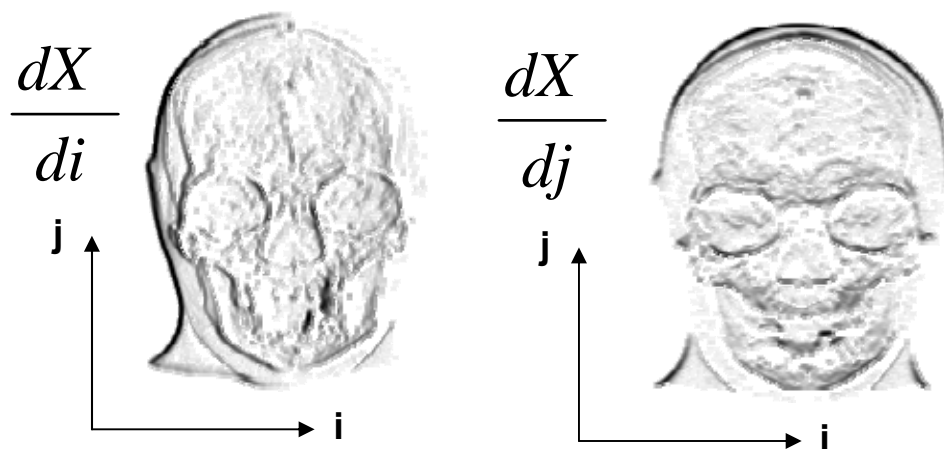
3D Texture:

- Hardware-Accelerated Rendering
- GPU draws transparent slices perpendicular to the viewpoint, automatically interpolated from a volume



Gradient Based Similarity Measures

- Good indicator for quality of alignment: correspondence of edges
- Similarity Measure that uses image gradients can be of advantage



Computing Gradients of discrete images

- Using forward difference operator

$$\frac{dX}{di}(i, j) = \Delta_i X(i, j) = X(i+1, j) - X(i, j)$$

$$\frac{dX}{dj}(i, j) = \Delta_j X(i, j) = X(i, j+1) - X(i, j)$$

- Using convolution kernel

$$\begin{aligned} \frac{dX}{di} &= X ** K_i \\ \frac{dX}{dj} &= X ** K_j \end{aligned} \quad K_i = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad K_j = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Computing Gradients of discrete images

- Better: Sobel filter

$$K_i = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad K_j = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

- Symmetric
 - Smooth, as using neighboring rows/columns
-

Gradient Correlation

- Mean of normalized cross correlation values of horizontal and vertical gradient images

$$GC = \frac{1}{2} NCC \left(\frac{dX}{di}, \frac{dY}{di} \right) + \frac{1}{2} NCC \left(\frac{dX}{dj}, \frac{dY}{dj} \right)$$

→ adds more freedom to the image dependency

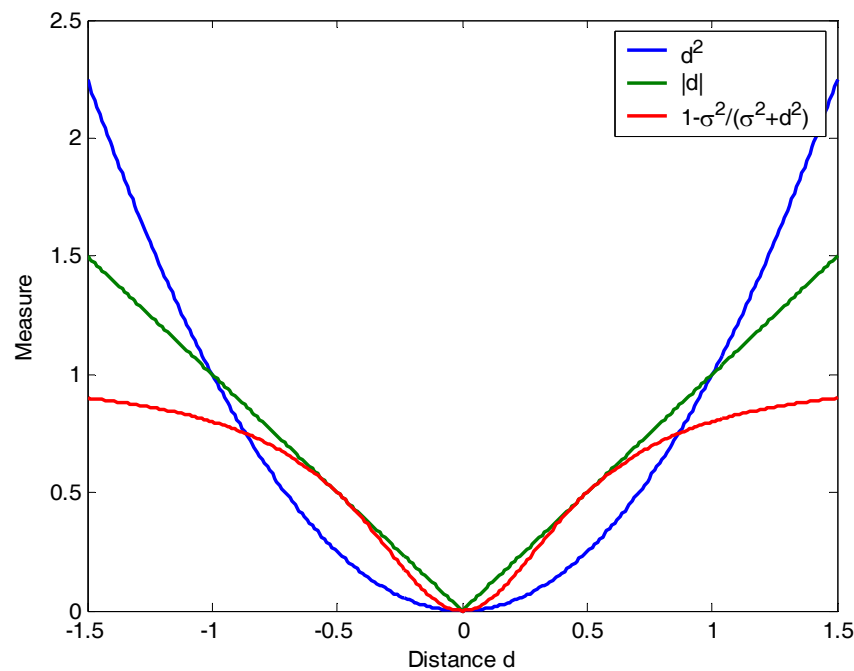
Gradient Difference

- Assesses content of horizontal and vertical gradient difference images
- Uses robust distance measure between gradient intensities

$$D_i = \frac{dX}{di} - \frac{dY}{di}; \quad D_j = \frac{dX}{dj} - \frac{dY}{dj}$$

$$GD = \sum_{i,j} \frac{\sigma^2}{\sigma^2 + D_i(i,j)^2} + \sum_{i,j} \frac{\sigma^2}{\sigma^2 + D_j(i,j)^2}$$

Robust Measures



$$1 - \frac{\sigma^2}{\sigma^2 + d^2}$$

If the error gets too large,
measure does not
increase anymore.
Sensibility selected with σ .

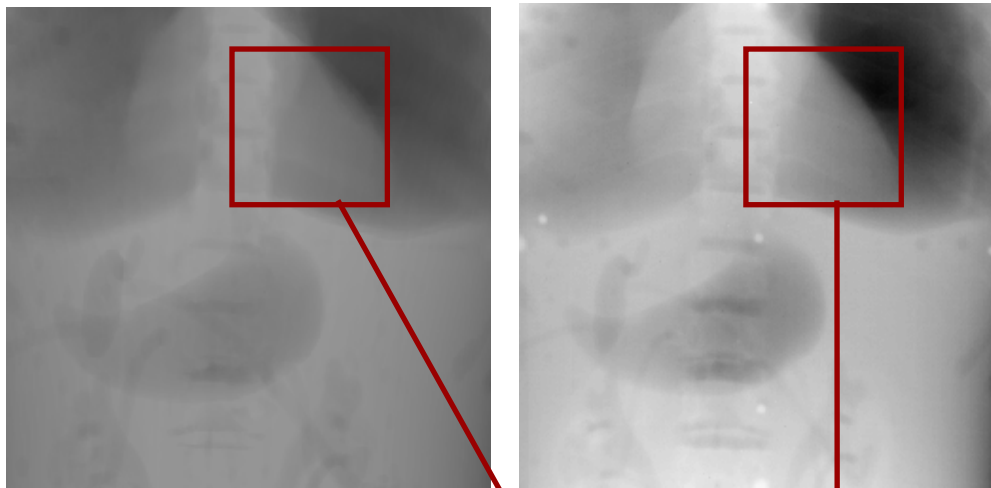
Pattern Intensity

$$PI = \sum_{i,j} \sum_{v,w} \frac{\sigma^2}{\sigma^2 + (D(i,j) - D(v,w))^2} \left[(i-v)^2 + (j-w)^2 \leq r^2 \right]$$

Assesses the amount of structures in a difference image.

→ using robust measure

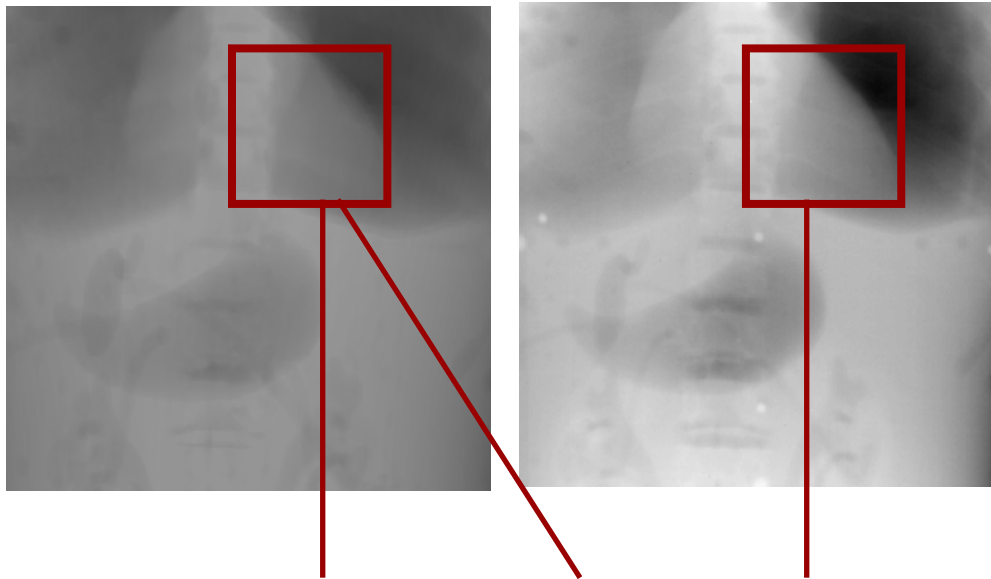
Local Normalized Correlation (LNC)



Much more precise than plain correlation, as globally varying (non-linear!) intensity changes do not matter anymore.

$$LNC = \frac{1}{n} \sum_{i,j} NCC(w_x(i,j), w_y(i,j))$$

Variance-Weighted LNC



$$VWC = \frac{\sum_{i,j} \sigma_{i,j}^2 NCC(w_x(i,j), w_y(i,j))}{\sum_{i,j} \sigma_{i,j}^2}$$

Further improvement -
weighting with the local
variance

Choice of image for variance
is important – wrong
structures in the other image
can be effectively ignored!

agenda

1. Non-linear Optimization Algorithms

Simplex

Best Neighbor

Gradient Descent

Powell-Brent

2. 2D/3D Registration Methods

X-Ray Projection Physics

DRR Generation Methods

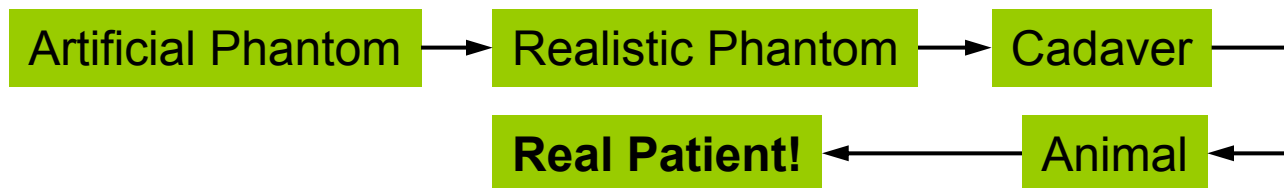
Adapted Similarity Measures

3. Validating Registration & Conclusion

Validating Registration

In particular intensity-based methods need to be verified – how?

- Very often: point-based methods
- Obtaining transformations with high precision from



- The closer to realism, the more effort!

Validating Registration

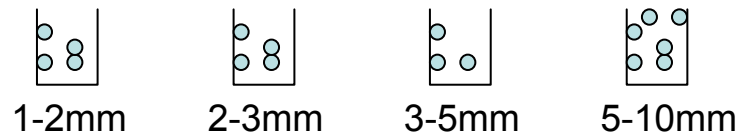
- General process
 1. Generate a **reference result**: transformation T_{ref}
 2. Run algorithm from **starting position** T_{start} (usually randomly generated) to generate a transformation T_{reg}
 3. Evaluate **Error** between T_{ref} and T_{reg}
 4. Iterate 1.-3. for a certain number of times (*Monte-Carlo Simulation*)
- Important results:
 - Accuracy: how small is your error?
 - Capture Range: how far can your starting position differ from reference result for a successful registration?

Validating Registration (2)

- How to generate a reference result T_{ref} ?
 - Ground Truth: Registration parameters are previously chosen, and input data is *synthetically* generated with them (difficult part is to *simulate* input data from registration parameters, easier for feature-based registration algorithms)
 - Gold Standard: Compute registration parameters with an algorithm that is known to yield a very good result (e.g. Maximum Likelihood estimator) → *Gold Standard algorithm*
-

Validating Registration (3)

- How to generate a starting position?
 - Usually generated by randomly perturbing registration parameters (e.g. angles and translation for rigid registration) → *position sample*
 - Control the starting positions: create random samples within a certain range (where Euclidean distance to desired result is smaller than a certain threshold)
 - Put random samples into “*range bins*” to analyze the capture range.



- Method only works for a reasonably small number of registration parameters

Validating Registration (4)

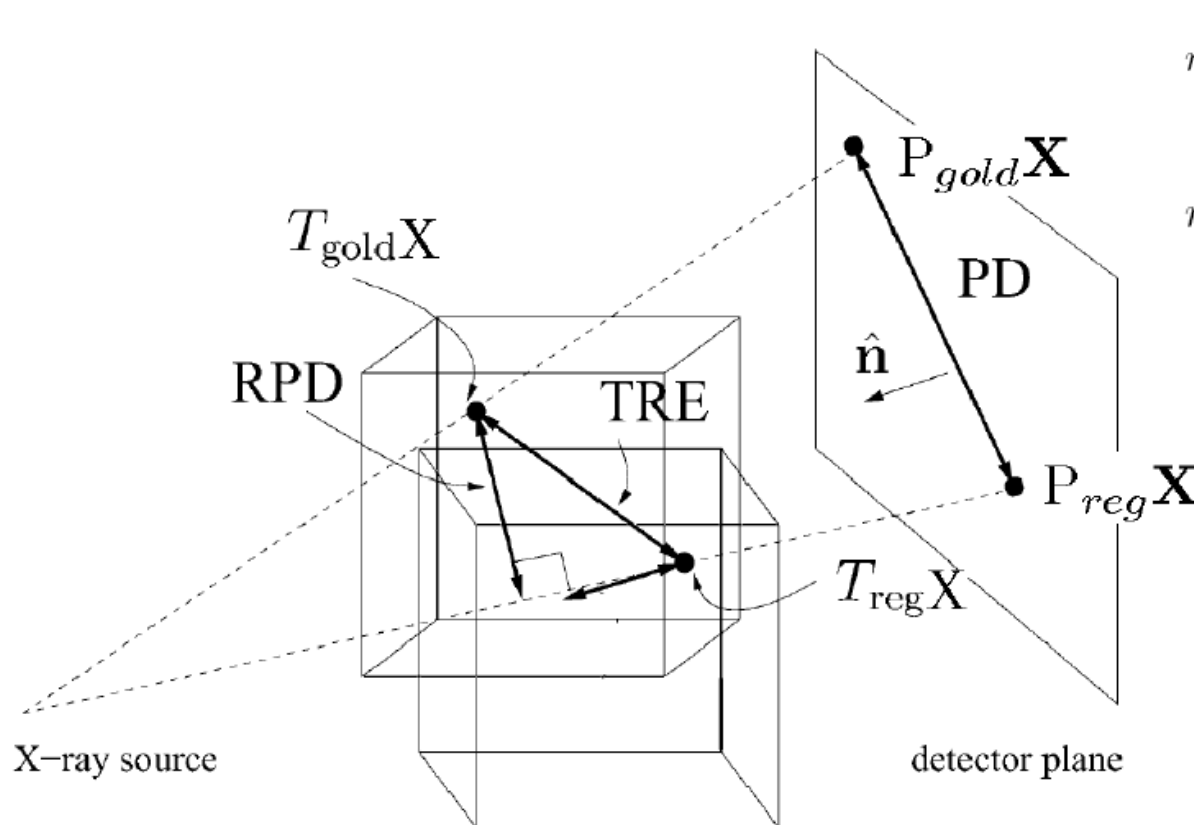
- How to define an error?
 - Target Registration Error (TRE)
 - Determine the position of anatomical targets in the data and measure the Euclidean distance
 - If anatomical targets are not identifiable: Sample data with a (regular) grid of point locations $\{X_i\}$ and measure Euclidean distance at all the sampling points: *Mean Target Registration Error (mTRE)*

$$\frac{1}{n} \sum_{i=1}^n d(T_{reg} X_i, T_{ref} X_i)$$

- Fiducial Registration Error (FRE)
 - Use fiducial markers for calculating the averaged Euclidean distance
- Depending on the registration problem (3D-3D, 3D-2D, ...)

Validating Registration (4)

- How to define an error: Example for 2D-3D Registration



$$mTRE = \frac{1}{n} \sum_{i=1}^n d(\mathbf{T}_{reg} \hat{\mathbf{X}}_i, \mathbf{T}_{gold} \hat{\mathbf{X}}_i)$$

$$mRPD = \frac{1}{n} \sum_{i=1}^n d_L(L_i, T_{gold} \hat{\mathbf{X}}_i)$$

$$mPD = \frac{1}{n} \sum_{i=1}^n d(\mathbf{P}_{reg} \hat{\mathbf{X}}_i, \mathbf{P}_{gold} \hat{\mathbf{X}}_i)$$

Validating Registration (5)

- Evaluating accuracy and capture range
 - The accuracy of a registration can be measured with TRE, mTRE, FRE
 - For the capture range, a *successful* registration must be defined (e.g. all results that yield a $TRE < 5\text{mm}$)
 - The capture range is the maximal TRE of T_{start} and T_{ref} where the registration was still successful
-

Conclusion on Medical Image Registration

- Methods strongly depend on underlying image data
- Choice of transformation, features, similarity measures etc. is crucial
- A lot of computational power necessary for many methods
- Close interaction with physicians necessary

→ **Very rich and interesting field of research
in computer science!**

References on Optimization and 2D-3D Registration

- [1] **W. Press et al.:** *Numerical Recipes in C*, Second Edition, CRC Press, Inc., 1992.
- [2] **Java Applet for Simplex Optimization:**
http://www.chem.uoa.gr/applets/AppletSimplex/Appl_Simplex2.html
- [3] **J. Hajnal, D. Hill, D. Hawkes:** *Medical Image Registration*. CRC Press, 2001
- [4] **J. Maintz, M. Viergever:** *An Overview of Medical Image Registration Methods*
- [5] **G. P. Penney, J. Weese, J. A. Little, P. Desmedt, D. L. Hill, and D. J. Hawkes:** *A Comparison of Similarity Measures for Use in 2D-3D Medical Image Registration*. LNCS, 1496 (2001).
- [6] **Khamene et al** *Automatic registration of portal images and volumetric CT for patient positioning in radiation therapy*. *Medical Image Analysis* 10 (2006), 96–112.
- [7] **D.B. Russakoff, T. Rohlfing, D. Rueckert, R. Shahidi, D. Kim and C.R. Maurer:** *Fast calculation of digitally reconstructed radiographs using light fields*. *SPIE 2003 Proceedings*.
- [8] **van de Kraats et al**, Standardized evaluation methodology for 2-D-3-D registration. *IEEE Trans. on Medical Imaging (TMI)* 24, 9 (2005), 1177–1189.