Image Registration

Andreas Husch[1]

University of Luxembourg

WS2022

[1] andreas.husch@uni.lu

Organisational things

- MOOC - Coursera / deeplearning.ai
  - `https://de.coursera.org/learn/ai-for-medical-prognosis/home/week/1`
  - Does it still work for students for free?

Organisational things

- MOOC - Coursera / deeplearning.ai
  - `https://de.coursera.org/learn/ai-for-medical-prognosis/home/week/1`
  - Does it still work for students for free?
- Student job opportunities at the Interventional Neuroscience group

Organisational things

- Student job opportunities at the Interventional Neuroscience group

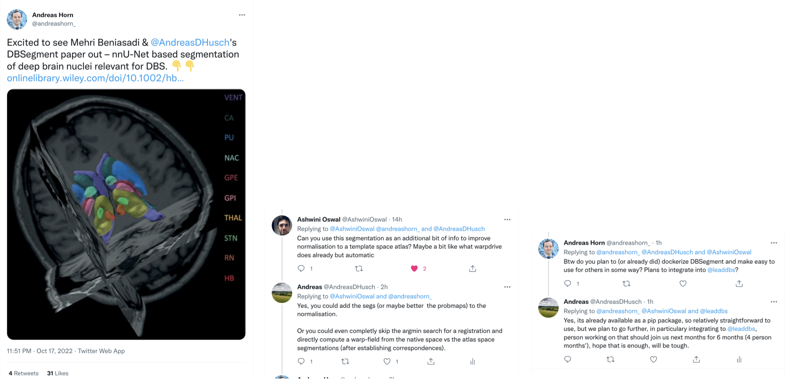


Figure 1: One student project, paper: `https://onlinelibrary.wiley.com/doi/10.1002/hbm.26097`

Acknowlegements

- Peter Gemmar
- Florian Bernard `https://lovc.cs.uni-bonn.de/`

Overview I

- Introduction
- Taxonomie of Registrations
- Point-based Registration
- Surface-based Registration
- Image-based Registration

*Cf. [5, 1, 6].

Introduction I

Registration: Informal: Finding a geometric transformation that best *aligns* an Image (or an object in space) with an other image (or object in space)

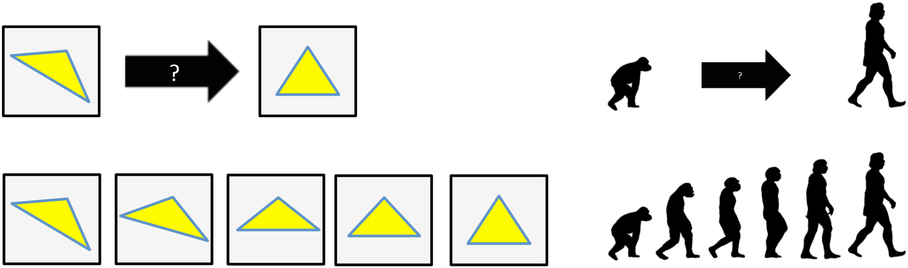


Figure 2: Registration problems and intermediate steps towards transforming an image to target image

Introduction II

- (Image-)registration is mostly only a *part* of a greater system (a pipeline)

In practice, often concatenation of a whole series of variously determined transformations.

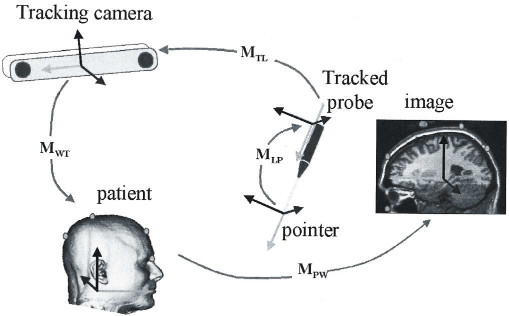


Figure 3: Practical application: concatenation of various registration processes as a basis for a medical system. Figure from [5]

Taxonomie

- What ”types” of registrations do exist?

[2] In the original nine categories, here compressed to eight according to Fitzpatrick [5]

Taxonomie

- What ”types” of registrations do exist?
- Classification of registration methods according to Maintz and Viergever [4] with eight main categories [2]:

[2] In the original nine categories, here compressed to eight according to Fitzpatrick [5]

Taxonomie

- What "types" of registrations do exist?
- Classification of registration methods according to Maintz and Viergever [4] with eight main categories [2]:
  1. **dimensionality**.

[2] In the original nine categories, here compressed to eight according to Fitzpatrick [5]

Taxonomie

- What ”types” of registrations do exist?
- Classification of registration methods according to Maintz and Viergever [4] with eight main categories [2]:
  1. **dimensionality**.
  2. **registration basis** (point-based, surface-based, intensity-based).

[2] In the original nine categories, here compressed to eight according to Fitzpatrick [5]

Taxonomie

- What ”types” of registrations do exist?
- Classification of registration methods according to Maintz and Viergever [4] with eight main categories [2]:
  1. **dimensionality**.
  2. **registration basis** (point-based, surface-based, intensity-based).
  3. **geometric transformation** (type of mathematical mapping, number of degrees of freedom. *object searched for!*).

[2] In the original nine categories, here compressed to eight according to Fitzpatrick [5]

Taxonomie

- What ”types” of registrations do exist?
- Classification of registration methods according to Maintz and Viergever [4] with eight main categories [2]:
    1. **dimensionality**.
    2. **registration basis** (point-based, surface-based, intensity-based).
    3. **geometric transformation** (type of mathematical mapping, number of degrees of freedom. *object searched for!*).
    4. **interaction** (manual, semi-automatic, interactive, automatic).

[2] In the original nine categories, here compressed to eight according to Fitzpatrick [5]

Taxonomie

- What "types" of registrations do exist?
- Classification of registration methods according to Maintz and Viergever [4] with eight main categories [2]:
  1. **dimensionality**.
  2. **registration basis** (point-based, surface-based, intensity-based).
  3. **geometric transformation** (type of mathematical mapping, number of degrees of freedom. *object searched for!*).
  4. **interaction** (manual, semi-automatic, interactive, automatic).
  5. **optimization method** (type of geom. transformation determination).

[2] In the original nine categories, here compressed to eight according to Fitzpatrick [5]

Taxonomie

- What "types" of registrations do exist?
- Classification of registration methods according to Maintz and Viergever [4] with eight main categories [2]:
  1. **dimensionality**.
  2. **registration basis** (point-based, surface-based, intensity-based).
  3. **geometric transformation** (type of mathematical mapping, number of degrees of freedom. *object searched for!*).
  4. **interaction** (manual, semi-automatic, interactive, automatic).
  5. **optimization method** (type of geom. transformation determination).
  6. **modalities** (registration of two same modalities (intra-modal/mono-modal) usually different from registration of two different modalities (inter-modal/multi-modal)).

[2] In the original nine categories, here compressed to eight according to Fitzpatrick [5]

Taxonomie

- What ”types” of registrations do exist?
- Classification of registration methods according to Maintz and Viergever [4] with eight main categories [2]:
  1. **dimensionality**.
  2. **registration basis** (point-based, surface-based, intensity-based).
  3. **geometric transformation** (type of mathematical mapping, number of degrees of freedom. *object searched for!*).
  4. **interaction** (manual, semi-automatic, interactive, automatic).
  5. **optimization method** (type of geom. transformation determination).
  6. **modalities** (registration of two same modalities (intra-modal/mono-modal) usually different from registration of two different modalities (inter-modal/multi-modal)).
  7. **subject** (registration of two images of the same patient (intra-patient), two different patients (inter-patient) or with one atlas).

[2] In the original nine categories, here compressed to eight according to Fitzpatrick [5]

Taxonomie

- What "types" of registrations do exist?
- Classification of registration methods according to Maintz and Viergever [4] with eight main categories [2]:
  1. **dimensionality**.
  2. **registration basis** (point-based, surface-based, intensity-based).
  3. **geometric transformation** (type of mathematical mapping, number of degrees of freedom. *object searched for!*).
  4. **interaction** (manual, semi-automatic, interactive, <u>automatic</u>).
  5. **optimization method** (type of geom. transformation determination).
  6. **modalities** (registration of two same modalities (intra-modal/mono-modal) usually different from registration of two different modalities (inter-modal/multi-modal)).
  7. **subject** (registration of two images of the same patient (intra-patient), two different patients (inter-patient) or with one atlas).
  8. **object** (anatomical area being registered, e.g. head, spine or heart).

[2] In the original nine categories, here compressed to eight according to Fitzpatrick [5]

Geometric transformations I

- A transformation T is a mapping that assigns to each point $\mathbf{x}$ of a point cloud X a transformed point $\mathbf{x}'$:

$$\mathbf{x}' = T(\mathbf{x}) \tag{1}$$

Geometric transformations II

- **rigid transformations:**

Properties:

- Distances between all points from X correspond to distances between all points from X' (distances are preserved).
- Straight lines remain straight lines.
- Angles between two straight lines are preserved.

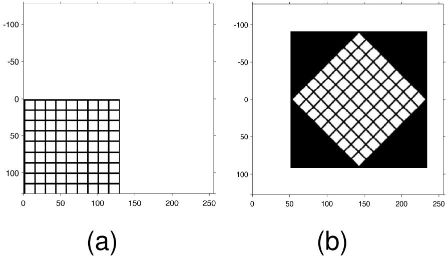


Figure 4: a) Untransformed lattice. (b) Grid after rigid transformation (rotation by angle $-\frac{pi}{4}$ and subsequent translation by $(50, 0)^T$).

Geometric transformations III

- Rigid transformations in 2D: Parameters: Rotation angle ϕ and translation vector $\mathbf{t} = (t_x, t_y)^T$ (total 3 degrees of freedom).

$$\mathbf{x}' = T(\mathbf{x}) = R_\phi \mathbf{x} + \mathbf{t} \tag{2}$$

$$\text{mit} \;\; R_\phi = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \tag{3}$$

Rigid 2D transformation with homogeneous coordinates (*post-multiplication*):

$$\mathbf{x}' = T(\mathbf{x}) = R_\phi^h \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} ; \;\; \text{mit} \;\; R_\phi^h = \begin{pmatrix} \cos\phi & -\sin\phi & t_x \\ \sin\phi & \cos\phi & t_y \\ 0 & 0 & 1 \end{pmatrix} \tag{4}$$

Geometric transformations IV

Matlab functions *packetform()* and *imtransform()* use homogeneous coordinates with *pre-multiplication*:

$$\mathbf{x}' = T(\mathbf{x}) = R_\phi^h \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} \tag{5}$$

$$\Leftrightarrow \quad (T(\mathbf{x}))^T = (R_\phi^h \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix})^T \tag{6}$$

$$\Leftrightarrow \quad (\mathbf{x}')^T = (T(\mathbf{x}))^T = \begin{pmatrix} x_1 & x_2 & 1 \end{pmatrix} (R_\phi^h)^T \tag{7}$$

$$\text{mit} \ (R_\phi^h)^T = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ t_x & t_y & 1 \end{pmatrix} \tag{8}$$

Geometric transformations V

- Rigid transformations in 3D:
  Parameters: Rotation angle ϕ_x, ϕ_y, ϕ_z (Euler angles) and translation vector $\mathbf{t} = (t_x, t_y, t_z)^T$ (total 6 degrees of freedom).

$$\mathbf{x}' = T(\mathbf{x}) = R\mathbf{x} + \mathbf{t} \qquad \text{(analog zu 2D)} \tag{9}$$

$$R = R_z R_y R_x \tag{10}$$

$$R_z = \begin{pmatrix} \cos\phi_z & -\sin\phi_z & 0 \\ \sin\phi_z & \cos\phi_z & 0 \\ 0 & 0 & 1 \end{pmatrix};\ R_y = \begin{pmatrix} \cos\phi_y & 0 & \sin\phi_y \\ 0 & 1 & 0 \\ -\sin\phi_y & 0 & \cos\phi_y \end{pmatrix}; \tag{11}$$

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_x & -\sin\phi_x \\ 0 & \sin\phi_x & \cos\phi_x \end{pmatrix} \tag{12}$$

Geometric transformations VI

- Remark:
  - Here the rotation is first around the x-axis, then around the y-axis and finally around the z-axis.
  - The use of homogeneous coordinates is also possible in 3D.
  - Homogeneous coordinates only allow for a more compact notation. Although they do not formally bring any further advantages, they are often used in practice since 4×4 matrix multiplications are natively supported by GPUs. 4×4
- Further variants for applying a rotation [5]:
  - Using Quaternions (extension of the complex numbers in the form $a + bi + cj + dk$)
  - One-time rotation around a rotation axis ω with angle θ_ω.

Geometric transformations VII

- **Lineare transformations** are all transformations, that can be computed using a Matrix multiplication (non-homogeneous transformation matrix) .

$$\mathbf{x}' = T(\mathbf{x}) = A\mathbf{x} \tag{13}$$

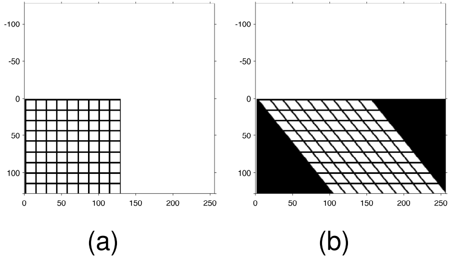


Figure 5: (a) Original grid (b) Grid after linear transformation.

Geometric transformations VIII

- Two main differences to a rigid transformation:
  - The matrix *A* Xb be an arbitrary (2×2 (2D) bzw. 3×3 (3D) Matrix sein.
  - A linear transformation does *not* include *translation*.
- linear transformations feature 4 d.o.f. in 2D and 9 d.o.f. in 3D.

Geometric transformations IX

- In contrast to the special case of the linear transformation, the **affine transformation** also involves a translation:

$$\mathbf{x}' = T(\mathbf{x}) = A\mathbf{x} + \mathbf{t} \tag{14}$$

Properties of affine (and linear) transformations:

- distances are not necessarily preserved (due to scaling and shearing).
- Straight lines remain straight lines.
- Parallelism is preserved, but angles between non-parallel lines may change.

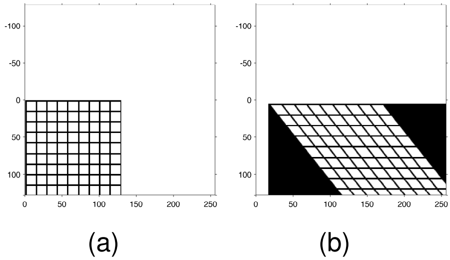


Figure 6: (a) Original grid. (b) Grid after affine transformation.

Geometric transformations X

- In case of **projective transformations** the parallelism of straight lines is no longer preserved.

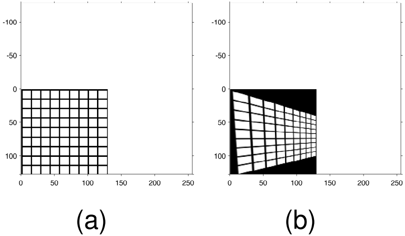


Figure 7: (a) Original grid. (b) Grid after projective transformation.

- A projective transformation can be defined as:

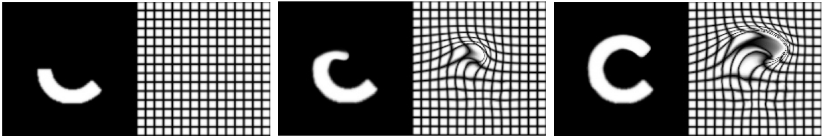
$$\mathbf{x}' = T(\mathbf{x}) = \frac{A\mathbf{x} + \mathbf{t}}{< \mathbf{p}, \mathbf{x} > + \alpha}. \quad (15)$$

Geometric transformations XI

- Projective transformations can be expressed in 3D with fully populated 4×4 matrix in homogeneous coordinates [5] (giving a total of 16 degrees of freedom in 3D).
- Perspective transformations are a subset of projective transformations.
  - Projection of spatial data onto a plane (3D to 2D).
  - Takes place, for example, in photography, X-rays, or when displaying 3D scenes on a (2D) monitor.

Geometric transformations XII

- **Curved/deformable transformations** allow transformations where straight lines are *not* preserved.



(a) (b) (c)

Figure 8: (a) Untransformed image with untransformed grid. (b) Partially transformed image with partially transformed grid. (C) Fully transformed image with fully transformed grid. Figures from [1].

Geometric transformations XIII

- Deformable Transformations:
  - In the literature often referred to as *non-linear* transformations. Strictly speaking, however, *non-linear* transformations include other transformations which are not deformable.
  - diffeomorphic transformations T are deformable transformations which satisfy the following properties:
    - Mapping T is bijective (total, injective, surjective).
      Colloquially: there is complete list of pairs between X and $X' = T(X)$.
      (Bijectivity ensures that an inverse mapping exists).
    - T and the inverse T^{-1} are continuously differentiable (T and T^{-1} do not produce jumps / are smooth).

Geometric transformations XIV

- Definition of deformable transformations:
  - A simple deformable transformation can be modeleds as a polynomial.
  - Using *B-Spline*-segments.

Geometric transformations XV

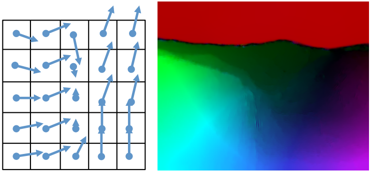
- Definition of deformable transformations in practise:
  - Non-analytic description of the transformation using *warp fields* (warps)(cf. *Lookup-tables*, cf. *images as functions*).

Geometric transformations XVI

- A warp field W defines for each point $\mathbf{x}$ in X a *displacement* vector $\mathbf{d}$.
- For a 3D image $f(x,y,z) : \mathbb{N}^3 \to \mathbb{N}$ the warp field is a mapping $W(x,y,z) : \mathbb{N}^3 \to \mathbb{R}^3$

$$\mathbf{x}' = T(\mathbf{x}) = \mathbf{x} + W(\mathbf{x}). \tag{16}$$

- We can consider a warp itself as a vector valued image



(a) (b)

Figure 9: (a) 2D warp field for an image of size 5×5. (b) Example of warp field represented as RGB image from ([3]).

Geometric transformations XVII

- Connection of the transformation types:
  Comparing the complexity of the transformation types (in terms of degrees of freedom):

  rigid $\prec$ linear $\prec$ affine $\prec$ projective $\overset{\text{in general}}{\prec}$ deformable

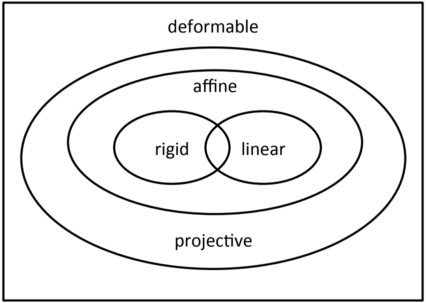
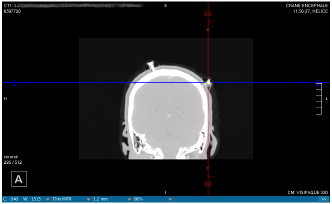


Figure 10: Venn Diagramm of discussed transformation types.

Point-based registration using homologous points I

point-based/landmark-based registration: Alignment of two point clouds with each other.



Point-based registration using homologous points II

- Input: Point-Clouds X und X'
  - Anatomical landmarks (anterior commissure, AC).
  - External markers (*Fiducials*), e.g. screws.
- Unkown: Transformation T
- Variants:
  1. Alignment of two point clouds of the same size with homologous landmarks (exact assignment of a landmark from X and X').
  2. Alignment of two point clouds with different number of points.

Point-based registration using homologous points III

- Theoretically, for affine transformations, 3 homologous points in 3D are sufficient to define a solution.
- Practically, however, it is useful to use more points so that errors in the definition of the landmarks/setting of the markers have less influence on the result.

Point-based registration using homologous points IV

- Squared error $e(T)$ of point-based registration for N homologous point pairs.

$$e(T) = \sum_{i=1}^{N} |\mathbf{x}_i - T(\mathbf{x}_i)|^2 = \sum_{i=1}^{N} |\mathbf{x}_i - \mathbf{x}'_i|^2 = \tag{17}$$

$$\sum_{i=1}^{N} (D_e(\mathbf{x}_i, \mathbf{x}'_i))^2 = \sum_{i=1}^{N} \sum_{d=1}^{D} (x_{i,d} - x'_{i,d})^2. \tag{18}$$

(19)

D is the dimensionality of the points ($D = 2$ or $D = 3$). e is a function of T. X and X' are considered as constant parameters here.

Point-based registration using homologous points V

- Objective: Find transformation T such that error e is minimal.
- Approach: Minimize the error $e(T) = \sum_{i=1}^{N} |\mathbf{x}_i - T(\mathbf{x}_i)|^2$ to find T:

$$T^* = \arg\min_T e(T) = \arg\min_T \left\{ \sum_i^N |\mathbf{x}_i - T(\mathbf{x}_i)|^2 \right\} \tag{20}$$

$\rightarrow$ registration is an optimization problem.

- solution depends on the type of transformation T.
  - rigid
  - linear
  - affine
  - projective
  - deformable

Point-based registration - Linear transformation I

- Simple case: linear transformation T ($T \in \mathbb{R}^{3\times 3}$).
- 1st possibility: solution by means of *analysis*
  . Determination of T using derivatives of the error function $e(T)$.
  - $e'(T) = 0$
    (at T there is a minimum, maximum or saddle point),
  - $e''(T) > 0$
    ($e(T)$ is left curved at T).
  - T is a vector quantity:
    - Instead of the first derivative, the gradient is set equal to zero: $\nabla e = \mathbf{0}$.
    - Instead of the second derivative, check if the Hessian matrix $H_e(T) = \left(\frac{\partial^2 e}{\partial T_i \partial T_j}(x)\right)_{i,j=1,\dots,9}$ is positive definite (positive eigenvalues only).

Point-based registration - Linear transformation II

- 2. possibility: solution by linear algebra:
- . T can be determined by solving N linear system of equations (LES):

$$\forall i \in \{1, ..., N\}: \quad T\mathbf{x}_i = \mathbf{x}'_i \tag{21}$$

$$\text{mit } T = \begin{pmatrix} T_{1,1} & T_{1,2} & T_{1,3} \\ T_{2,1} & T_{2,2} & T_{2,3} \\ T_{3,1} & T_{3,2} & T_{3,3} \end{pmatrix}. \tag{22}$$

- The N LESs in Eq. 21 can be combined into a single system as follows:

$$\underbrace{\begin{pmatrix} T_{1,1} & T_{1,2} & T_{1,3} \\ T_{2,1} & T_{2,2} & T_{2,3} \\ T_{3,1} & T_{3,2} & T_{3,3} \end{pmatrix}}_{?} \underbrace{\begin{pmatrix} x_{1,1} & x_{2,1} & \dots & x_{N,1} \\ x_{1,2} & x_{2,2} & \dots & x_{N,2} \\ x_{1,3} & x_{2,3} & \dots & x_{N,3} \end{pmatrix}}_{\text{bekannte Punkte X}} = \underbrace{\begin{pmatrix} x'_{1,1} & x'_{2,1} & \dots & x'_{N,1} \\ x'_{1,2} & x'_{2,2} & \dots & x'_{N,2} \\ x'_{1,3} & x'_{2,3} & \dots & x'_{N,3} \end{pmatrix}}_{\text{bekannte Punkte X'}} \tag{23}$$

Point-based registration - Linear transformation III

$$\Leftrightarrow TX = X' \qquad \text{(transponieren)} \tag{24}$$

$$\Leftrightarrow (TX)^T = (X')^T \tag{25}$$

$$\Leftrightarrow X^T T^T = (X')^T \tag{26}$$

$$\Leftrightarrow X^T T^T = (X')^T \tag{27}$$

$$\Leftrightarrow \underbrace{\begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ \vdots & \vdots & \vdots \\ x_{N,1} & x_{N,2} & x_{N,3} \end{pmatrix}}_{A =_{def} X^T} \begin{pmatrix} \underbrace{\begin{matrix} T_{1,1} \\ T_{1,2} \\ T_{1,3} \end{matrix}}_{\mathbf{y}_1} & \underbrace{\begin{matrix} T_{2,1} \\ T_{2,2} \\ T_{2,3} \end{matrix}}_{\mathbf{y}_2} & \underbrace{\begin{matrix} T_{3,1} \\ T_{3,2} \\ T_{3,3} \end{matrix}}_{\mathbf{y}_3} \end{pmatrix} = \begin{pmatrix} \underbrace{\begin{matrix} x'_{1,1} \\ \vdots \\ x'_{N,1} \end{matrix}}_{B_1} & \underbrace{\begin{matrix} x'_{1,2} \\ \vdots \\ x'_{N,2} \end{matrix}}_{B_2} & \underbrace{\begin{matrix} x'_{1,3} \\ \vdots \\ x'_{N,3} \end{matrix}}_{B_3} \end{pmatrix} \tag{28}$$

$\rightarrow$ Known form $A\mathbf{y}_j = B_j$ for LGSs with $A, B \in \mathbb{R}^{N \times 3}$

- For each $j \in \{1, 2, 3\}$ there is exactly one system of equations to solve (the j-th column of T is determined in each case).

Point-based registration - Linear transformation IV

- For $N \leq 3$ the LES is underdetermined and infinitely many solutions exist (at least 3 points are needed).
- For $N = 3$ no or exactly one solution exists:

$$A\mathbf{y} = B \tag{29}$$

$$\Leftrightarrow \quad \mathbf{y} = A^{-1}B \tag{30}$$

- For $N > 3$ the LGS is overdetermined and no exact solution exists. $\rightarrow$ **least-squares** approach to error minimization.

Point-based registration - Linear transformation V

- Justification for the non-existence of an exact solution for $N > 3$ (self-study) Comparison to base transformation in PCA:
  - There are exactly three basis vectors $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ available (columns in A).
  - By $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ a three-dimensional hyperplane is spanned.

$$A = \begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{pmatrix}. \tag{31}$$

  - $\mathbf{y} = (y_1, y_2, y_3)^T$ form the coefficients by which the columns are multiplied:

$$\mathbf{a}_1 y_1 + \mathbf{a}_2 y_2 + \mathbf{a}_3 y_3. \tag{32}$$

  - Thus, any choice of $\mathbf{y}$ can reach only the points lying on the plane spanned by A.
  - Usually, the 3 ($< N$) basis vectors are not enough to represent the N-element column vector $\mathbf{b}$ (N basis vectors are needed). In other words, $\mathbf{b}$ usually lies outside the plane spanned by A.

Point-based registration - Linear transformation VI

- **least-squares** approach (method of least squares, minimization of the squared error) by means of the pseudoinverses, since A^{-1} does not exist:

$$
\begin{aligned}
& & Ay &= B & (33)\\
&\Rightarrow & y &= A^{+}B & (34)\\
&\Leftrightarrow & y &= (A^{T}A)^{-1}A^{T}B & (35)
\end{aligned}
$$

- Keyword: Normal equation. Cf. also linear regression (mathematically exactly the same!)

Point-based registration - Linear transformation VII

- This approach can *not* directly be used for other transformation types, since additional requirements are imposed here.
- Affine transformation:

$$T^{*} = \arg\min_{T} e(T) = \arg\min_{T} \left\{ \sum_{i}^{N} |\mathbf{x}_i - (T(\mathbf{x}_i) + \mathbf{t})|^2 \right\}. \tag{36}$$

Point-based registration - Linear transformation VIII

- Rigid Transformation:

$$T^* = \arg\min_T e(T) = \arg\min_T \left\{ \sum_i^N |\mathbf{x}_i - (T(\mathbf{x}_i) + \mathbf{t})|^2 \right\}. \qquad (37)$$

  - T must be rotation matrix only (columns orthonormal and $\det(T) = 1$).
  - Also known as *Procrustes analysis*.
  - Can be solved by *Singular Value Decomposition (SVD)* (generalization of eigenvalue decomposition for non-square matrices).

Point-based registration - Linear transformation IX

- Weighted variants:
  - Same weight in x-, y- und z-direction (isotropic, w_i is scalar):

$$T^* = \arg\min_T e(T) = \arg\min_T \left\{ \sum_i^N w_i |\mathbf{x}_i - (T(\mathbf{x}_i) + \mathbf{t})|^2 \right\}. \qquad (38)$$

    Can also be solved in the rigid case by means of SVD.
  - Different weights along the x, y, and z directions (anisotropic, each W_i is a matrix):

$$T^* = \arg\min_T e(T) = \arg\min_T \left\{ \sum_i^N |W_i(\mathbf{x}_i - (T(\mathbf{x}_i) + \mathbf{t}))|^2 \right\}. \qquad (39)$$

    Iterative solution.

Surface-based registration I

- Usually, no homologous points can be identified directly. $\rightarrow$ Alignment of two point clouds without known homologous points.
- General idea: transform into point-based registration problem by determining points on the surface.

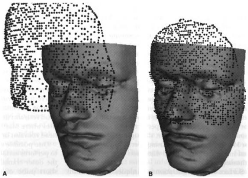


Figure 11: Figure from [5]

- Typically, a surface pointcloud X' derived from an image dataset and a "sampled" pointcloud X with $|X'| >> |X|$

  In practice, for example, X' with 20000 elements and X with 200

ICP-Algorihm

Idea of *Iterative Closest Point* algorithm for two point clouds X and X' *without* homologous points:

1. Assign to each point from X the *closest* point from X' (thus reducing the problem to the known homologous case).
2. Apply point-based registration for homologous points.
3. *Repeat* until convergence is reached.

Image-based registration - overview I

- In **image-based/voxel-based** registration, image intensities are compared rather than points.
- *image*-based registration is more complex than the *point*-based registration variants.
- If matching points can be determined in two images, the images can be efficiently registered using the methods discussed so far.
  - Anatomical landmarks (anterior commissure, AC).
  - External markers (*fiducials*), e.g. screws. Automatic detection of these markers is the task of medical imaging.

Image-based registration - overview II

- The goal of **image-based registration** is to align two images as best as possible.
- For this purpose, the image *intensities* are compared with each other.
- The static reference image is called *fixed image*.
- The image which is transformed to the *fixed image* is called *moving image*.
- input: images $I_{fixed}(\mathbf{x}), I_{moving}(\mathbf{x})$.
- output: transformation T, mapping all voxel coordinates $\mathbf{x}$ from I_{moving} to voxel coordinates $\mathbf{x}'$ in I_{fixed}.

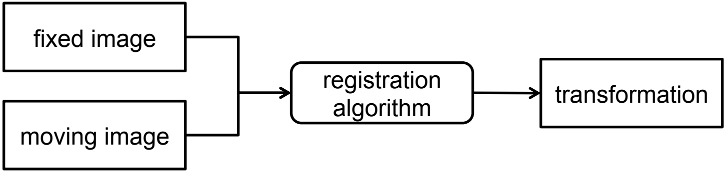
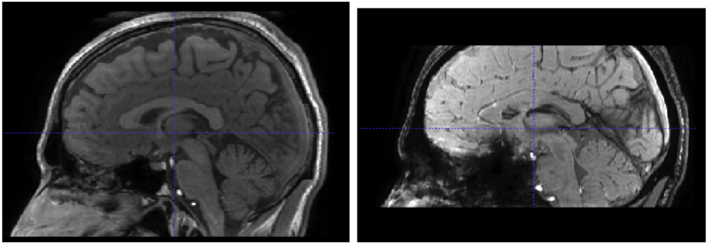


Image-based registration: modalities, subject, and examples of use. I

- Inter-modality-intra-patient registration:
  - Alignment of different images of the same patient with each other.
- Necessary for multimodal approaches (all images must be in the same coordinate system).
  - For MRI or CT images usually using a rigid transformation:
    - anatomy is the same
    - only translation and rotation expected (e.g. due to different positioning of the patient in the scanner).
  - For other image modalities where perspective distortions occur, perspective transformation is necessary, e.g.
    - X-ray images,
    - photography,
    - etc.

Image-based registration: modalities, subject, and examples of use. II



(a) (b)

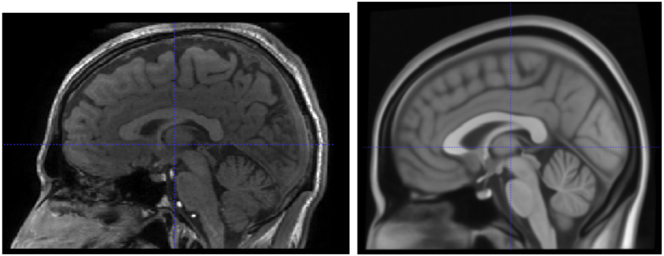
Figure 12: (a) T1 MRI of one patient. (b) SWAN MRI of the same patient.

Image-based registration: modalities, subject, and examples of use. III

- Intra-modality-inter-patient registration:
  - Nessesary for:
    - Representation of patient images in a standardized coordinate system.
    - Creation of an atlas (registration of many images and subsequent averaging).
  - Rigid Transformation? Affine Transformation? Deformable transformation? $\rightarrow$ Type of transformation depends on application purpose.
  - Deformable transformation allows best possible matching of both images.

Image-based registration: modalities, subject, and examples of use. IV

- Here: Alignment of a patient image (T1 MRI) with a brain atlas (T1 MRI).

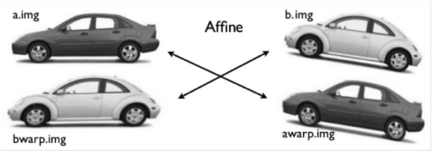


(a) (b)

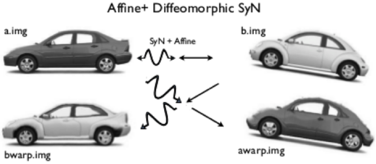
Figure 13: (a) T1-MRI of a patient. (b) T1-MNI152 brain atlas.

Image-based registration: modalities, subject, and examples of use. V

- comparison of an affine transformation with an affine+diffeomorphic transformation:



(a)



(b)

Figure 14: (a) Affine registration. (b) Affine and diffeomorphic registration. Figures from [1].

Image-based registration: modalities, subject, and examples of use. VI

- Diffeomorphic transformation of faces:

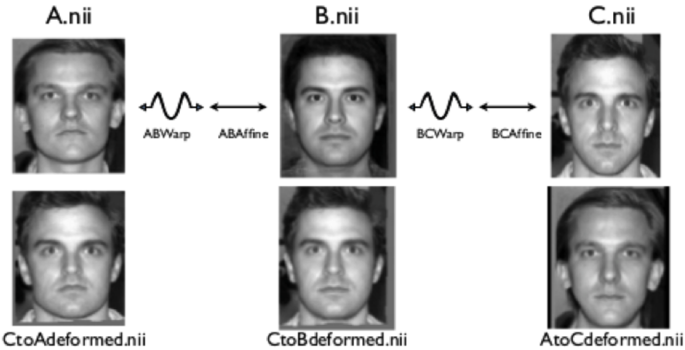


Figure 15: Affine and diffeomorphic registration. Figure from [1].

Image-based registration: algorithm

- Overview of an image-based registration algorithm:

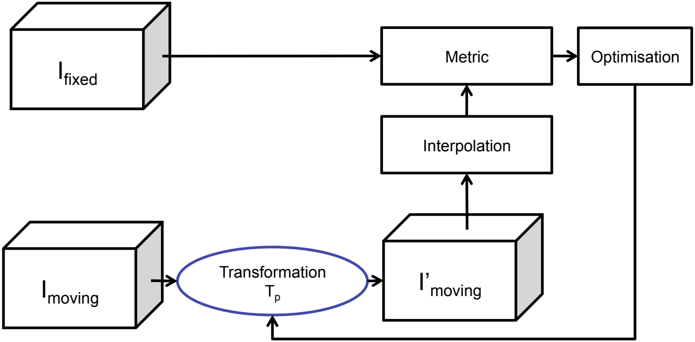


Image-based registration: transformation types I

- The **transformation types** in image-based registration correspond to the transformation types of point-based registration (rigid, affine, linear, projective, deformable).
- However, all voxels $\mathbf{x}_i$ transformed by $\mathbf{x}'_i = T(\mathbf{x}_i)$ are considered as input 'points', with the goal that $I_{fixed}(\mathbf{x}_i)$ and $I_{moving}(T(\mathbf{x}_i))$ match as well as possible for all voxels.

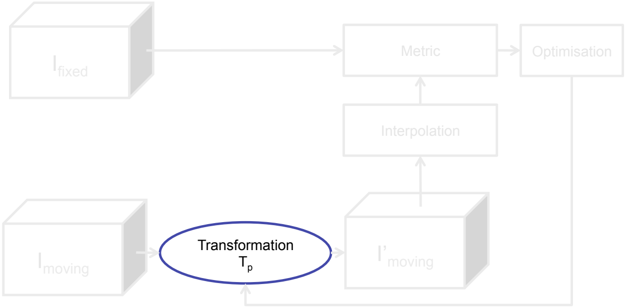


Image-based registration: metrics I

- **metrics** / **error measures** are needed to evaluate the agreement or disagreement between $I_{fixed}(\mathbf{x}_i)$ and $I_{moving}(T(\mathbf{x}_i))$.

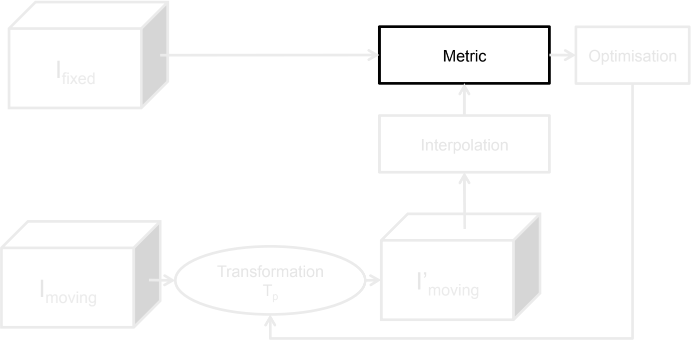


Image-based registration: metrics II

- Using the error measure, image-based registration can be defined as an optimization problem:

$$T^* = \arg\min \left\{ \underbrace{f(T)}_{f(T)=e(T)+s(T)} \right\} \tag{40}$$

$$= \arg\min \left\{ \underbrace{e(T)}_{\text{data term}} + \underbrace{s(T)}_{\text{regularisation term}} \right\} \tag{41}$$

  - The *data term* indicates the similarity of the images $I_{fixed}(\mathbf{x})$ and $I_{moving}(T(\mathbf{x}))$:

$$e(T) =_{def} D(I_{moving}(T(\mathbf{x})), I_{fixed}(\mathbf{x})). \tag{42}$$

  - The *regularization term* $s(T)$ is used to ensure that the simplest possible transformation T is found (avoiding *overfitting*).
    - Reduction of local minima during optimization.
    - Degradation of error measure for unwanted configurations.
    - Simple approach: constraining parameters to certain interval.

Image-based registration: metrics III

- Digression: regularization to avoid *overfitting*:

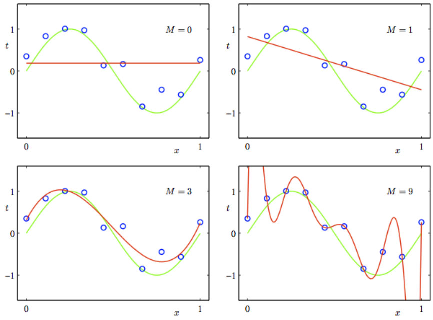


Figure 16: illustration of *overfitting*. The blue points are noisy measurement data generated by the green curve. The points should be approximated as well as possible by a polynomial. M indicates the degree of the polynomial used. In this case, the regularization term could be defined as $s(M) = \alpha M$ to obtain a polynomial of the lowest possible degree (α describes the weighting of the *regularization term* relative to the *data term*). To ensure that the result of the optimization is not a trivial (sub-optimal) solution (e.g., due to $M = 0$), both the *data term* and the *regularization term* must be appropriately chosen. Figure from [2].

Image-based registration: metrics IV

- Remark:
  - By applying the image transformation to I_{moving} there are image areas of $I_{fixed}(\mathbf{x})$ and $I_{moving}(T(\mathbf{x}))$ that no longer overlap. The distance measures are applied only to the overlapping areas.
  - images I_{fixed}, I_{moving} are considered as column vectors $\mathbf{a}, \mathbf{b}$ in the following (facilitates formalization).

Image-based registration: metrics V

- Scalar product (SP):

$$D_{SP}(\mathbf{a},\mathbf{b}) = -|\mathbf{a}^T\mathbf{b}| \quad (43)$$

- Repetition:
  - Scalar product for angle determination between two vectors.
  - The smaller the angle between **a** and **b**, the larger the amount of the scalar product.
  - Since the distance is to be minimized (because of $\arg\min$ in formula 41), the negative scalar product is used.
- Simple measure used for *intra*-modality registrations.

Image-based registration: metrics VI

- Sum of Squares Difference (SSD):

$$D_{SSD}(\mathbf{a},\mathbf{b}) = \frac{1}{N}\sum_{i}^{N}|a_i - b_i|^2 \tag{44}$$

$$= \frac{1}{N}|\mathbf{a}-\mathbf{b}|^2 \tag{45}$$

$$= \frac{1}{N}(\mathbf{a}-\mathbf{b})^T(\mathbf{a}-\mathbf{b}). \tag{46}$$

  - Simple measure used for **intra**-modality registrations. Why for intra?

Image-based registration: metrics VII

- Correlatioin coefficient (CC):

$$D_{CC}(\mathbf{a},\mathbf{b}) = -\frac{\sum_i^N \left((a_i - \bar{a})(b_i - \bar{b})\right)}{\sqrt{\sum_i^N (a_i - \bar{a})^2 \sum_i^N (b_i - \bar{b})^2}} \tag{47}$$

$$= -\frac{(\mathbf{a} - \bar{\mathbf{a}})^T (\mathbf{b} - \bar{\mathbf{b}})}{|\mathbf{a} - \bar{\mathbf{a}}||\mathbf{b} - \bar{\mathbf{b}}|}. \tag{48}$$

- Corresponds to the (known) normalized cross correlation.
- A high correlation corresponds to a high agreement.
- Since minimization problem is considered, negative correlation is used.
- Difference from SSD:
  - Invariant to intensity value scaling ($s\mathbf{a}$, s is scaling parameter).
  - Invariant to addition of a constant intensity offset ($\mathbf{a} + \mathbf{1}c$, $\mathbf{1}c$ is offset vector).

Image-based registration: metrics VIII

- Correlatioin coefficient (CC):
  - Üblicherweise für *intra*-modality Registrierungen verwendbar.
  - *Nur bedingt* auch für *inter*-modality Registrierungen verwendbar (gdw. Unterschied zwischen den Modalitäten überwiegend aus Skalierung und konstantem Offset besteht).

Image-based registration: metrics IX

- Normalisierte Mutual-Information (NMI):

$$D_{NMI}(\mathbf{a},\mathbf{b}) = -\frac{H(\mathbf{a})+H(\mathbf{b})}{H(\mathbf{a},\mathbf{b})}. \tag{49}$$

  - entropy $H(\mathbf{a}) = -\sum_s P_{\mathbf{a}}(s)\log_2 P_{\mathbf{a}}(s) \quad (H(\mathbf{b})$ analog) Measure of included randomness in **a** (larger values $\rightarrow$ more randomness). (Shannon, 1948)
  - s are the occurring intensity values in the image: $s = \{a_i \mid 1 \leq i \leq N\}$.
  - $P_{\mathbf{a}}(s)$ is the probability density (*PDF*) which is approximated by the normalized histogram in the discrete case.
  - joint entropy $H(\mathbf{a},\mathbf{b}) = -\sum_{s,t} P_{\mathbf{a},\mathbf{b}}(s,t)\log_2 P_{\mathbf{a},\mathbf{b}}(s,t)$
  - Joint frequency density $P_{\mathbf{a},\mathbf{b}}(s,t)$ is approximated with normalized 2D histogram.
  - Since minimization problem is considered, negative NMI is used. (Viola & Wells, 1993)
  - Can be used for *inter*-modality registrations, therefore *outstanding importance* in practice.

Digression: Multidimensional Histograms I

A **histogram** is a graphical representation of the frequency of data.

- For this purpose, the range of values is divided into discrete ranges (*bins*) or intervals.
- For each of these ranges, it is then counted how many of the occurring values lie in the corresponding range.
- An N-dimensional histogram is a function $h : \mathbb{R}^N \rightarrow \mathbb{N}$ that assigns a number (frequency value/accumulator) to an N-dimensional point.

Digression: Multidimensional Histograms II

- 1D histograms are used when the *value range* is one-dimensional (e.g. signal, gray scale image, unimodal MRI image, ...).

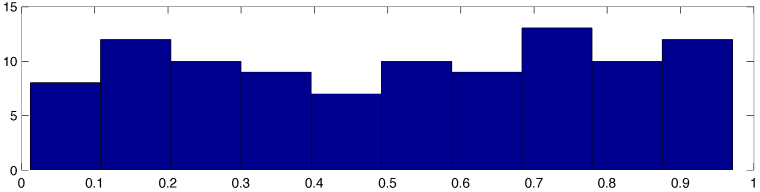
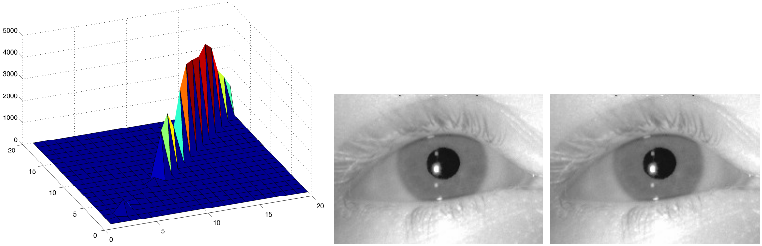


Figure 17: 1-dimensional histogram of a 100-element random vector with values from the range [0; 1] (generated in MATLAB using $x = rand(1, 100)$).

Digression: Multidimensional Histograms III

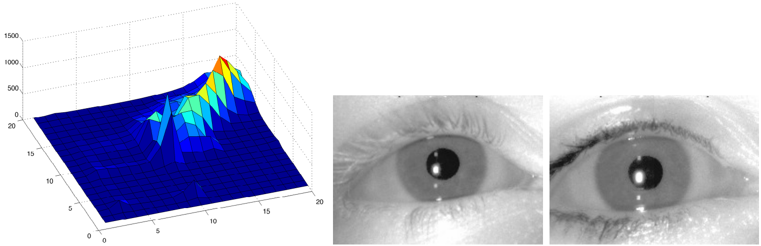
- 2D histograms are used when the *value range* is two-dimensional (e.g. a joint histogram of two gray-scale images *joint histogram*, is used for image registration).



(a) (b) (c)

Figure 18: (a) Two-dimensional histogram with 20 *bins*, the first dimension being the image in (b), the second dimension being the image in (c). Since (b) and (c) each represent the same images, so all elements in the accumulator are on the main diagonal.

Digression: Multidimensional Histograms IV



(a) (b) (c)

Figure 19: (a) Two-dimensional histogram with 20 *bins*, the first dimension is the image in (b), the second dimension the image in (c). The two images (b) and (c) are different here, so the pairs of values of the accumulator are no longer on a main diagonal.

$\Rightarrow$ 2D histograms of two images can be used to compare the gray values/intensities in the images.

Digression: Multidimensional Histograms V

- 2D histograms to compare the intensity values of two images:
  - The spatial arrangement of the occurring values is ignored.
    - Whether this is desirable depends on the specific application.
    - If spatial information is needed, the histograms of image patches/superpixels can be considered instead and spatial information about the arrangement of these patches can be integrated.
  - Equal images (in the sense of occurring intensity values) provide only points on the main diagonal in the 2D histogram (and vice versa).

Digression: Multidimensional Histograms VI

- 3D histograms are used when the *value range* is three-dimensional (e.g. RGB image).

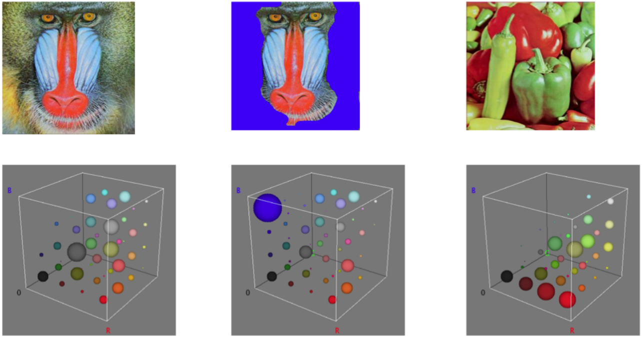


Figure 20: RGB images and their 3D histograms displayed as 3D *Bubble Chart*. The images are taken from slides of the lecture in Visual Information Retrieval by Kai Uwe Barthel, HTW Berlin.

Digression: Multidimensional Histograms VII

- ND Histograms are used when the *value range* is N-dimensional.

Image-based registration: interpolation I

- **interpolation**: Transformed indices $\mathbf{x}' = T(\mathbf{x})$ are usually not integer. Since images cannot be indexed with decimal numbers, interpolation must be applied.

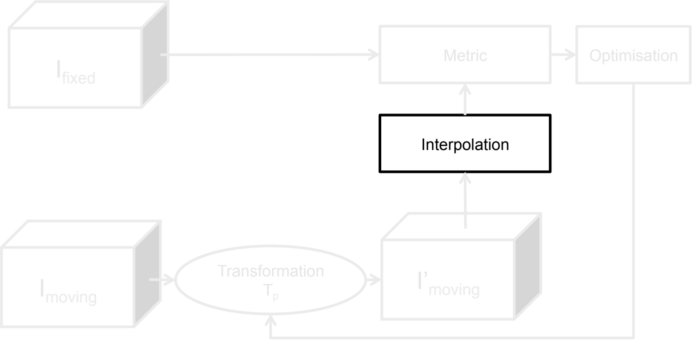


Image-based registration: interpolation II

- *Nearest-Neighbor* Interpolation: The intensity value in the closest voxel is used:

$$\mathbf{x}' = \mathrm{round}(T(\mathbf{x})). \tag{50}$$

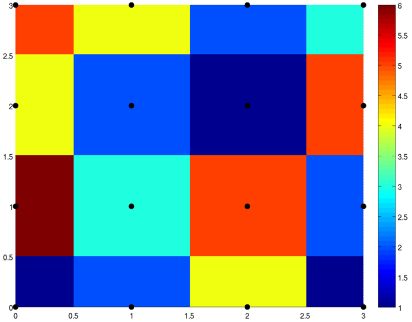


Figure 21: Example of the result of a *nearest-neighbor* interpolation. Figure from Wikipedia.

Image-based registration: interpolation III

- Linear interpolation: The intensity value is averaged proportionally from the neighboring voxels.
  - In 2D: bilineare Interpolation.
  - In 3D: trilineare Interpolation.

Image-based registration: interpolation IV

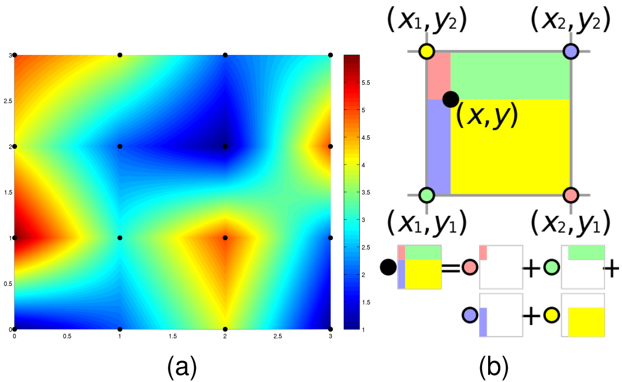


Figure 22: (a) Example of the result of bilinear interpolation. (b) Schematic representation of bilinear interpolation as a mixture of the intensity values of the 4-neighborhood. Figures from Wikipedia (http://en.wikipedia.org/wiki/File:Bilinear_interpolation_visualization.svg).

Image-based registration: interpolation V

- Cubic interpolation: The intensity value is determined by means of a cubic function (e.g. splines).

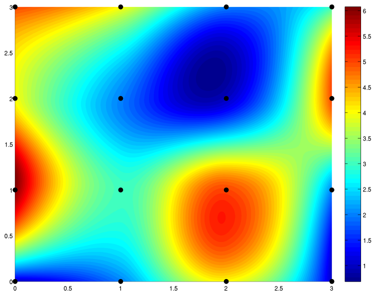
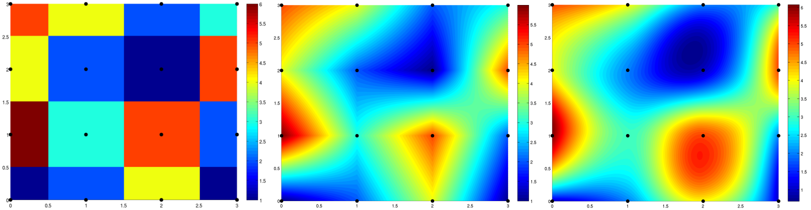


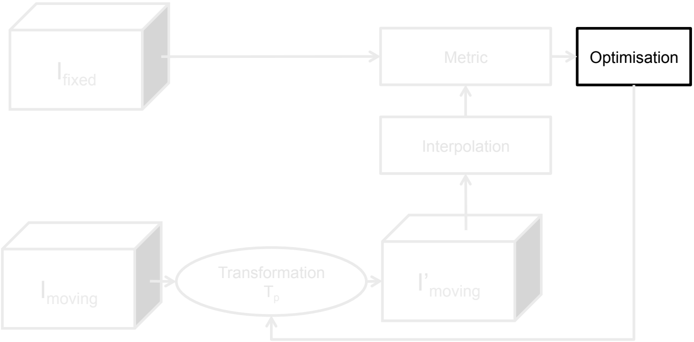
Figure 23: Example of the result of a bicubic interpolation. Figure from Wikipedia



(a) (b) (c)

Figure 24: Comparison of NN, bilinear and cubic interpolation. Figures from Wikipedia

Image-based registration: optimization / search I



- no analytical solution
- search procedure is needed to find minimum T^*:

$$T^* = \arg\min\{e(T) + s(T)\}. \tag{51}$$

Image-based registration: optimization / search II

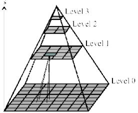
- Many optimization variants depending on the choice of $e(T)$ and $s(T)$.
- Search space is usually very large:
  - Dimensionality of the search space corresponds to the number of degrees of freedom of T (depending on the transformation type).
- General approach: Iterative methods for solving non-linear optimization problems (e.g. gradient methods).
- To accelerate convergence and increase robustness, a *multi-scale* approach is often used (e.g. Gaussian image pyramid). 

Image-based registration: optimization / search III

- **Gradient based method**:
  - Iterative method (which you all know from Deep Learning ;-)).
  - *gradient-descend*: Gradient descent method for minimization using error function f.
  - Basic idea: Determine the searched value in the next iteration using the gradient of $\nabla f(T)$:

$$T^{n+1} = T^n + \beta \nabla f(T^n). \tag{52}$$

    where β is the step size:
    - Constant.
    - decreasing with n.
    - *Line Search*: find best value for β along gradient direction T^n.
    - *Inexact Line Search*: find approximation for best value for β along gradient direction T^n.
  - Requirement: Sufficiently good initialization, since it cannot be guaranteed that a global maximum is found.

Image-based registration: optimization / search IV

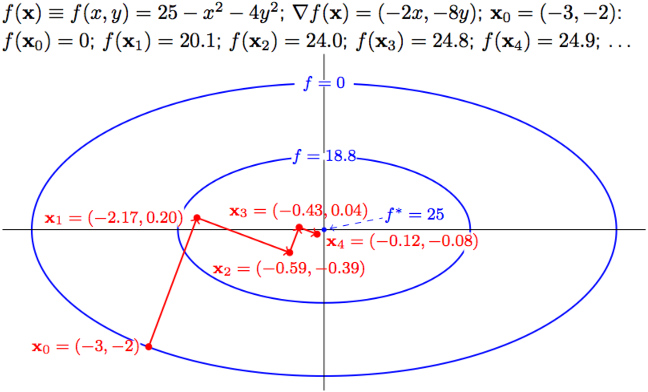


Figure 25: Schematic of the gradient method in 2D for optimizing the nonlinear function $f(x, y)$. Figure by Georgy Gimel'farb, University of Auckland.

Overview I

- Introduction
- Categorization of registrations
- Transformation types
- Point-based registration
- Surface-based registration
- Image-based registration

*The slides on image registration are based on [5, 1, 6].

Bibliography

AVANTS, BRIAN B, NICK TUSTISON and GANG SONG: *Advanced Normalization Tools (ANTS).*
Insight J, 2009.

BISHOP, CHRISTOPHER M: *Pattern Recognition and Machine Learning (Information Science and Statistics).*
Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

BROX, THOMAS, ANDRÉS BRUHN, NILS PAPENBERG and JOACHIM WEICKERT: *High accuracy optical flow estimation based on a theory for warping.*
pages 25–36, 2004.

MAINTZ, JB and MAX A VIERGEVER: *A survey of medical image registration.*
Medical image analysis, 2(1):1–36, 1998.

SONKA, M and J M FITZPATRICK: *Handbook of Medical Imaging, Volume 2: Medical image processing and analysis.*
Optics & Photonics News, 2002.

TOENNIES, KLAUS D: *Guide to Medical Image Analysis.*
Methods and Algorithms. Springer, February 2012.