

IMA101 - Introduction to image registration

Loic Le Folgoc

Slides by Pietro Gori

Enseignant-chercheur

Equipe IMAGES - Télécom Paris

contact: loic.lefolgoc@telecom-paris.fr



Plan

- 1 Introduction
- 2 Geometric global transformations
- 3 Image warping and interpolations
- 4 Intensity-based registration

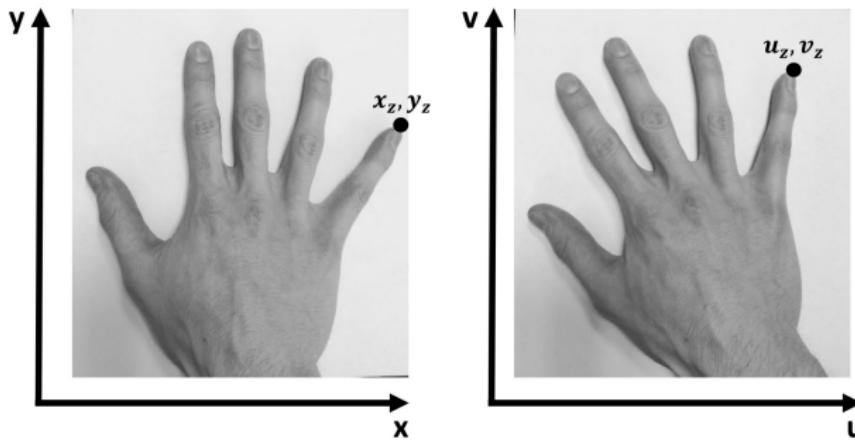
Summary

- 1 Introduction
- 2 Geometric global transformations
- 3 Image warping and interpolations
- 4 Intensity-based registration

Image registration

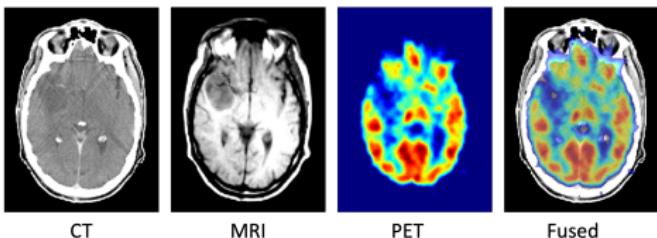
Definition

Geometric: find the *optimal* parameters of a *geometric transformation* to spatially align two different images of the same object. It establishes *spatial correspondence* between the pixels of the source (or moving) image with the ones of the target image



Medical Image Registration - Applications

Multi-modal image fusion



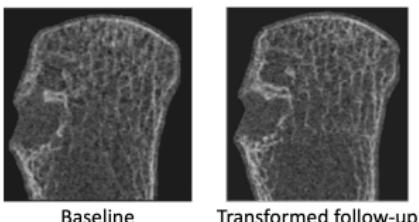
CT

MRI

PET

Fused

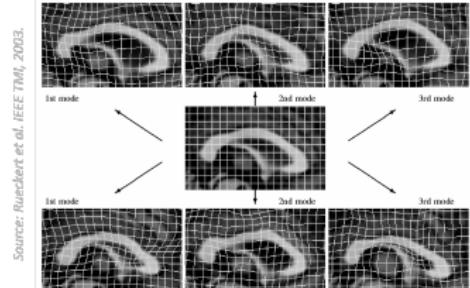
Longitudinal comparison



Baseline

Transformed follow-up

Population modelling and statistical atlases

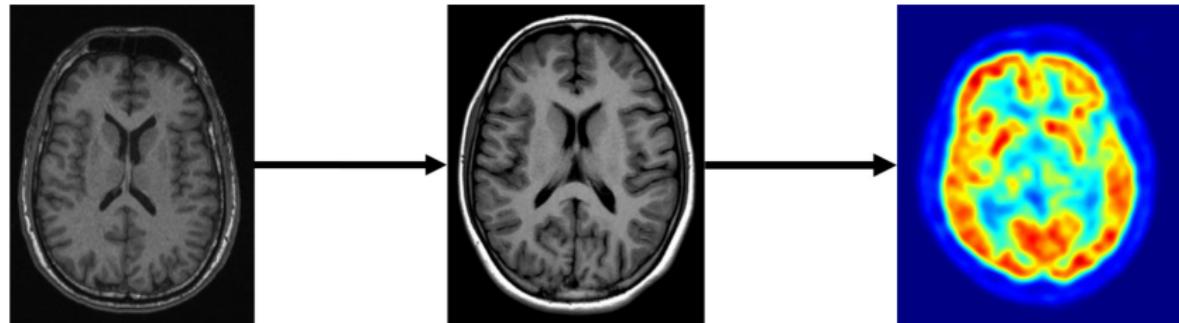


Source: Rueckert et al. IEEE TMI, 2003.



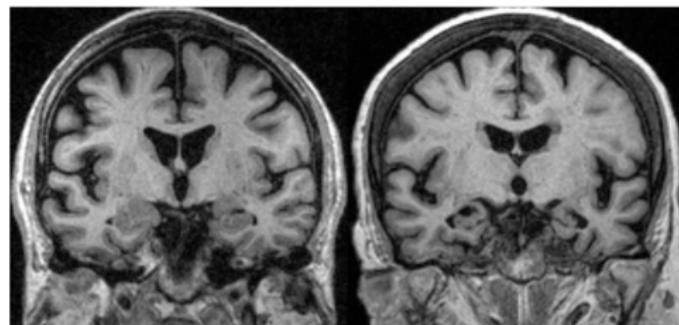
From Brunet et al. BMC Medical Imaging, 2020.

Medical Image Registration - Applications

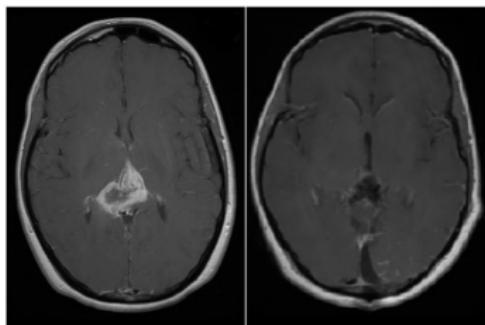


**Same modality
Different subjects**

**Different modalities
Same subject**

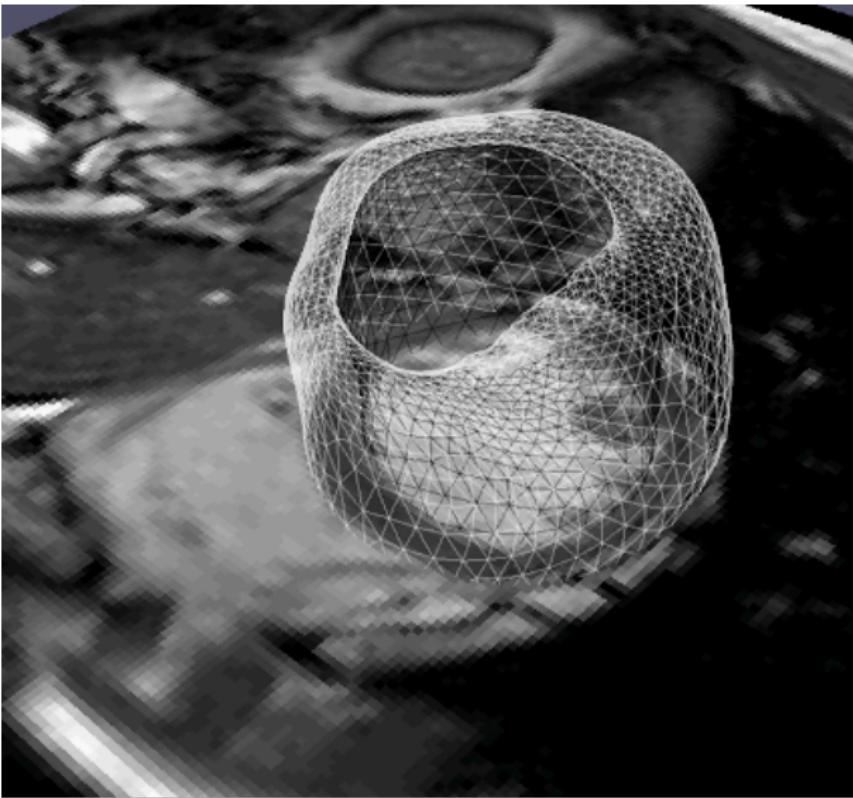


Longitudinal study

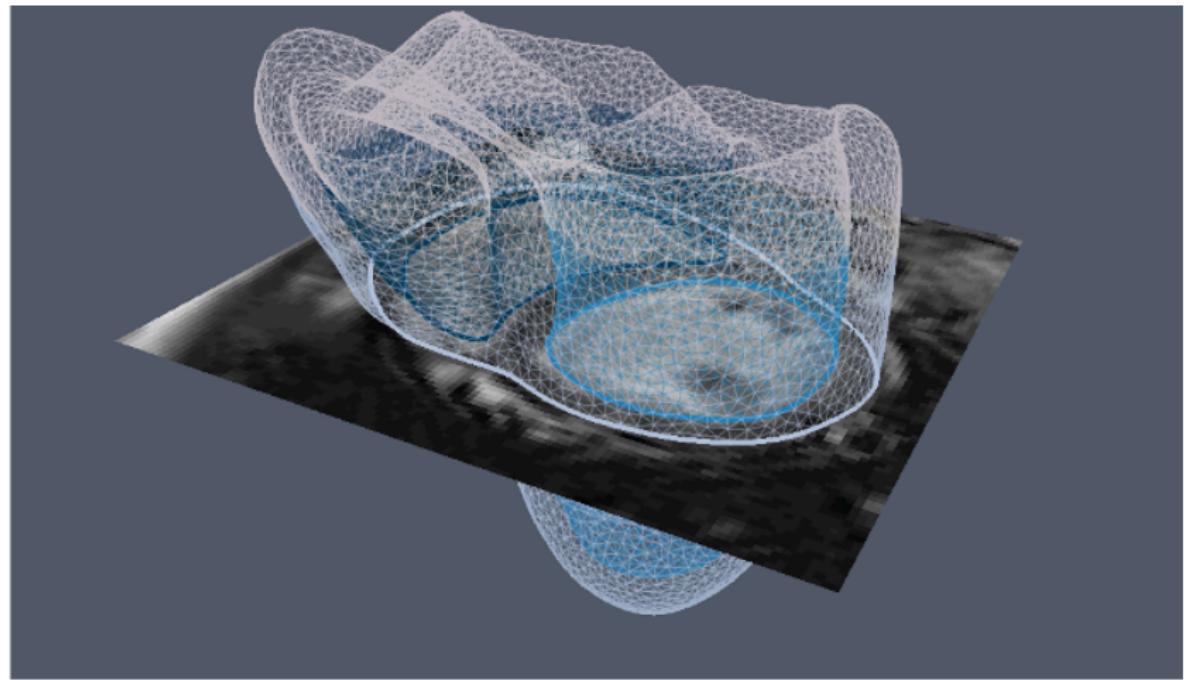


Pre and postoperative

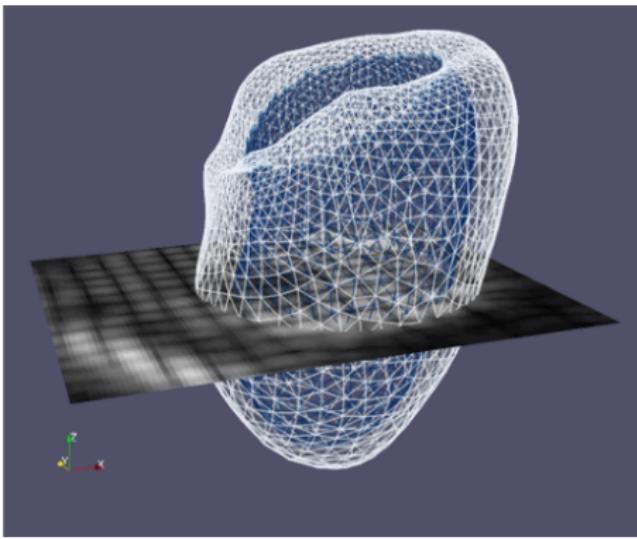
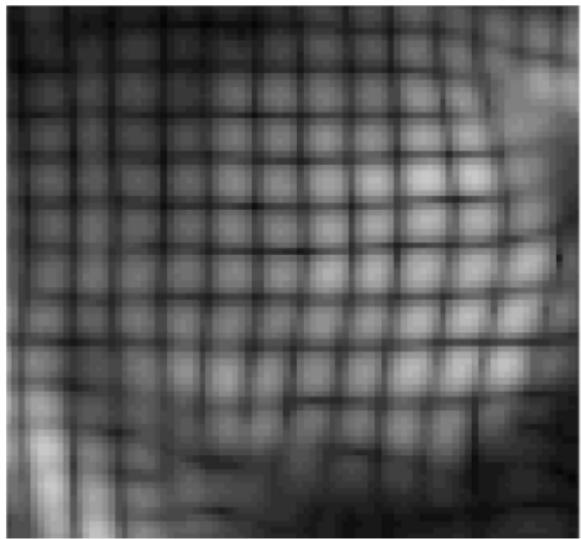
Medical Image Registration - Applications



Medical Image Registration - Applications



Medical Image Registration - Applications



Other Applications

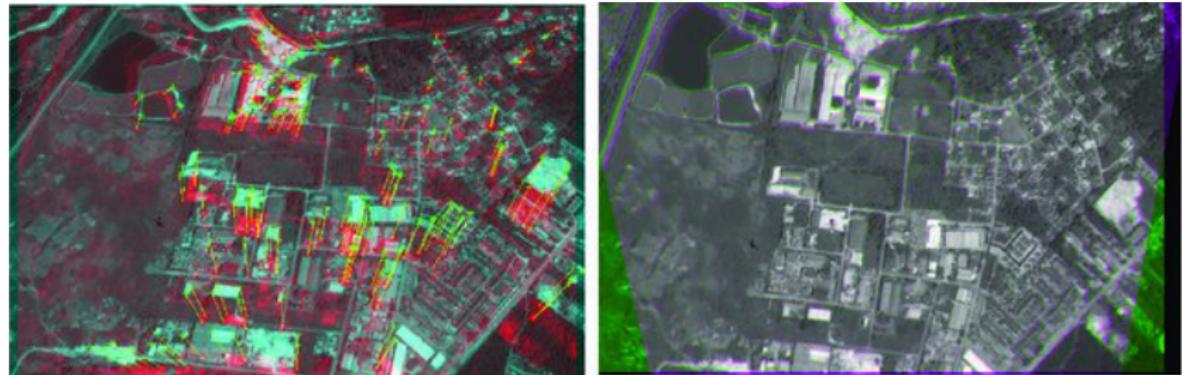


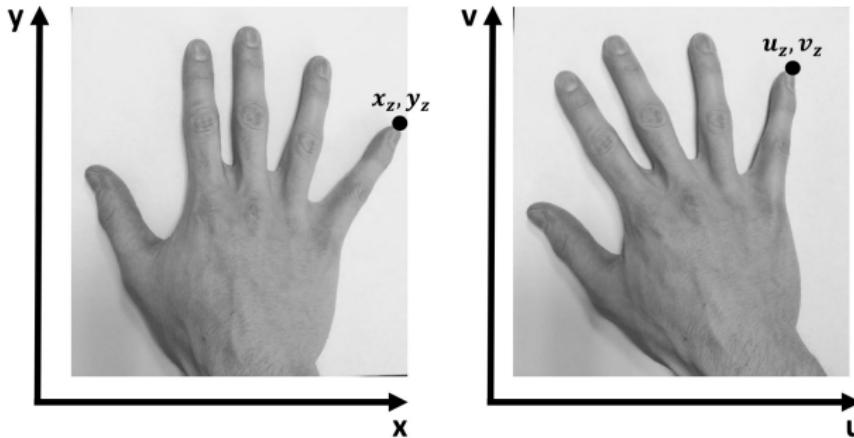
Figure 1: Remote sensing example - From 'Geometric feature descriptor and dissimilarity-based registration of remotely sensed imagery' PLoS One, 2018

Image registration components

- **Dimensionality:** 2D/2D, 3D/3D, 2D/3D
- **Transformation** (linear / non-linear)
- **Similarity metric** (e.g. intensities, landmarks, edges, surfaces)
- **Optimization procedure**
- **Interaction** (automatic / semi-automatic / interactive)
- **Modalities** (mono-modal / multi-modal)
- **Subjects** (intra-subject / inter-subject / atlas construction)

Introduction

- Let I and J be the source and target images. They show the same anatomical object, most of the time with a different field of view and resolution (sampling)
- $I(x, y)$ and $J(u, v)$ represent the intensity values of the pixels located onto two regular grids: $\{x, y\} \in \Omega_I$ and $\{u, v\} \in \Omega_J$
- For the same subject, the same anatomical point z can be in position x_z, y_z in I and in u_z, v_z in J



Mathematical definition

- Both I and J are functions:

$$\begin{array}{ccc} I(x, y) : & \Omega_I \subseteq \mathbb{R}^2 & \rightarrow & \mathbb{R} \\ & (x, y) & \rightarrow & I(x, y) \end{array}$$

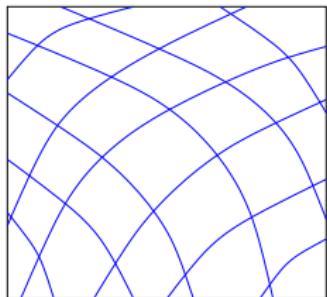
- We look for a geometric transformation \mathbf{T} , which is a 2D warping parametric function that belongs to a certain family Γ :

$$\begin{array}{ccc} \mathbf{T}_\phi(x, y) : & \mathbb{R}^2 & \rightarrow & \mathbb{R}^2 \\ & (x, y; \phi) & \rightarrow & \mathbf{T}_\phi(x, y) \end{array}$$

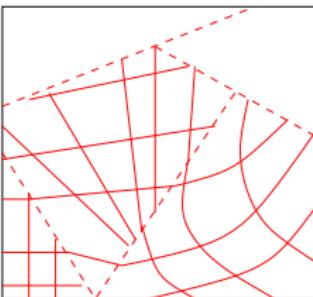
- ϕ is the vector of parameters of \mathbf{T} . We look for a transformation that maps (x_z, y_z) to (u_z, v_z)

Mathematical definition

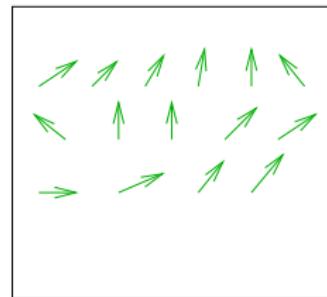
- Most of the time, I and J are simply seen as matrices whose coordinates (x, y) and (u, v) are thus integer-valued (number of line and column)
- The values of the intensities of the pixels can be real numbers \mathbb{R} (better for computations) or integer, usually in the range $[0, 255]$ for a gray-scale image
- There are several kind of transformations:



modèle global



modèle par morceaux
(régional)



modèle local

Summary

- 1 Introduction
- 2 Geometric global transformations
- 3 Image warping and interpolations
- 4 Intensity-based registration

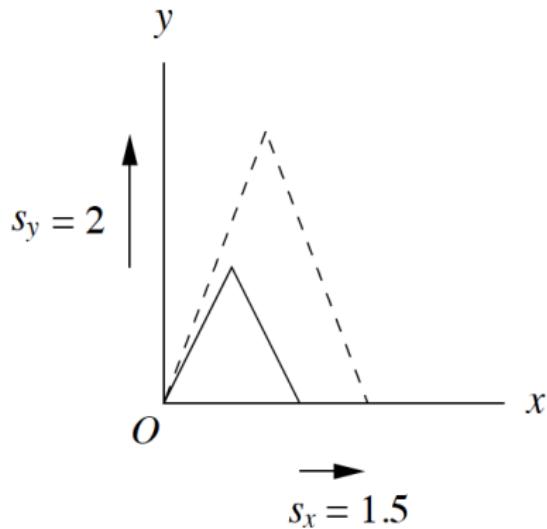
Global transformations

- Global transformations can be defined with matrices
- Application: $\mathbf{T}_\phi(x, y) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = A\mathbf{x}$
- Inverse (if invertible): $\mathbf{T}_\phi^{-1}(x, y) = A^{-1}\mathbf{x}$
- Composition: $\mathbf{T}_1(\mathbf{T}_2(x, y)) = (\mathbf{T}_1 \circ \mathbf{T}_2)(x, y) = A_1 A_2 \mathbf{x}$
- Note: order of transformation is important : $A_1 A_2$ is not equal to $A_2 A_1$ in general

Global transformations

- Global means that the transformation is the same for any points p
- **Scaling** - multiply each coordinate by a scalar

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (1)$$



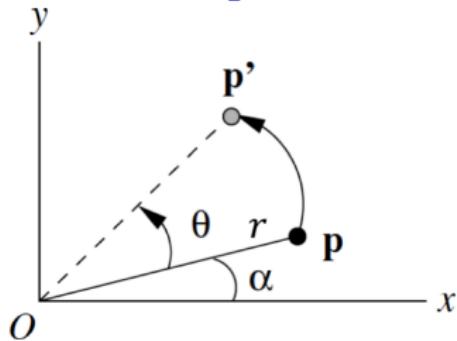
Global transformations

- **Rotation** - WRT origin. Let $p = [x \ y]^T$ and $p' = [u \ v]^T$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r \cos(\alpha) \\ r \sin(\alpha) \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} r \cos(\alpha + \theta) \\ r \sin(\alpha + \theta) \end{bmatrix} = \begin{bmatrix} r(\cos(\alpha) \cos(\theta) - \sin(\alpha) \sin(\theta)) \\ r(\sin(\alpha) \cos(\theta) + \cos(\alpha) \sin(\theta)) \end{bmatrix}$$

$$= \begin{bmatrix} x \cos(\theta) - y \sin(\theta) \\ y \cos(\theta) + x \sin(\theta) \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



- $R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$

- $\det(R) = 1$

- $R^{-1} = R^T$

Global transformations

- **Reflection**

- Horizontal (Y-axis)

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} r \cos(\pi - \alpha) \\ r \sin(\pi - \alpha) \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

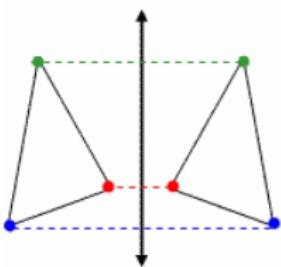
- Vertical (X-axis)

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} r \cos(-\alpha) \\ r \sin(-\alpha) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- $\det(R) = -1$

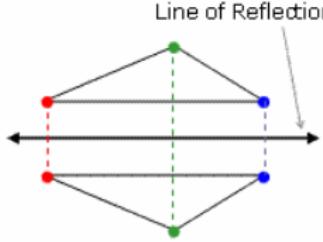
- $R^{-1} = R^T$

Line of Reflection



Horizontal Reflection
(flips across)

Line of Reflection



Vertical Reflection
(flips up/down)

Global transformations

- **Shear** - Transvection in french

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 & \lambda_x \\ \lambda_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & \tan(\phi) \\ \tan(\psi) & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

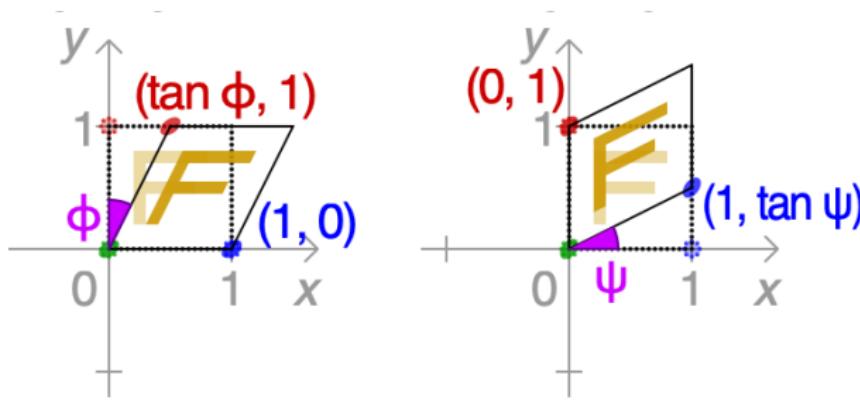


Figure 2: Shear in x and y direction. Image taken from Wikipedia.

2D Linear transformations

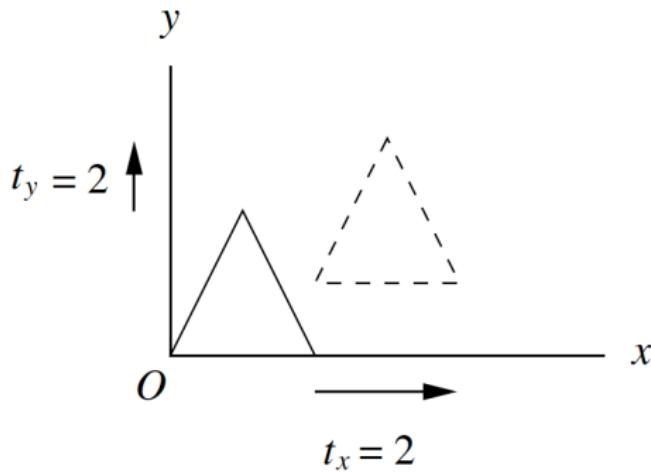
$$\begin{bmatrix} u \\ v \end{bmatrix} = \underbrace{\begin{bmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} 1 & \lambda_x \\ \lambda_y & 1 \end{bmatrix}}_{\begin{bmatrix} a & b \\ c & d \end{bmatrix}} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Linear transformations are combinations of:
 - scaling
 - rotation
 - reflection
 - shear
- Properties of linear transformations:
 - origin is always transformed to origin
 - parallel lines remain parallel
 - ratios are preserved
 - lines remain lines

Translation

- It does not have a fixed point → no matrix multiplication

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \end{bmatrix} \quad (2)$$



Homogeneous coordinates - 2D affine transformation

- Instead than 2D matrices we use 3D matrices.
- **Affine transformation:** combination of linear transformations and translations
- Let $p = [x \ y]^T$ and $p' = [u \ v]^T$, we obtain $p' = \mathbf{T}p$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- $\mathbf{T} = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{0}^t & 1 \end{bmatrix}$, where A is the 4 dof linear component and \mathbf{t} the 2 dof translation

2D affine transformation

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Properties of affine transformations:

- origin is *not* always transformed to origin
- parallel lines remain parallel
- ratios are preserved
- lines remain lines

2D Projective transformations (homographies)

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} a & b & g \\ c & d & h \\ e & f & 1 \end{bmatrix}}_{\mathbf{T}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- $\mathbf{T} = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^t & 1 \end{bmatrix}$, where A is the 4 dof affine component, \mathbf{t} the 2 dof translation and \mathbf{v} the 2 dof elation component
- If we consider only \mathbf{v} , so $\mathbf{t} = \mathbf{0}$ and $A = \mathcal{I}$:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ e & f & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ ex + fy + 1 \end{bmatrix}$$

$$u = \frac{x}{ex + fy + 1} \quad v = \frac{y}{ex + fy + 1}$$

2D Projective transformations (homographies)

$$u = \frac{x}{ex + fy + 1} \quad v = \frac{y}{ex + fy + 1}$$

- **Elation:** points are scaled by a scaling factor which is a linear combination of x and y . Points can be mapped to (resp. from) infinite to (resp. from) a finite scalar value. (Ex. if you set $f=0$ and $e=1$, then the point $[\infty, \infty]$ is mapped to $[1, 1]$)
- Properties of projective transformations:
 - origin is *not* always transformed to origin
 - parallel lines do *not* necessarily remain parallel
 - ratios, length and angle are *not* preserved
 - lines remain lines

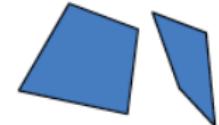
Examples

A square transforms to:



Projective
8dof

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}$$



Affine
6dof

$$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



Similarity
4dof

$$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



Euclidean
3dof

$$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



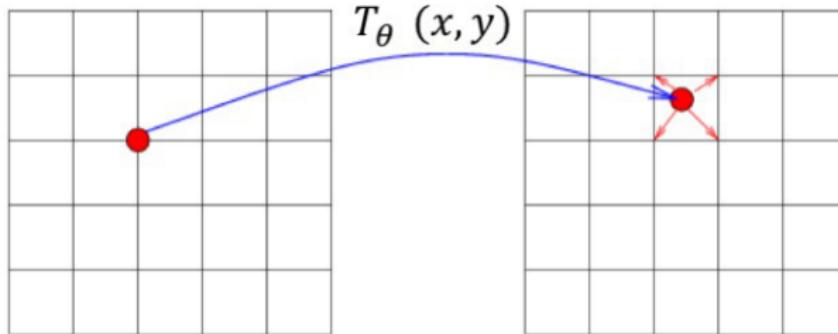
Summary

- 1 Introduction
- 2 Geometric global transformations
- 3 Image warping and interpolations
- 4 Intensity-based registration

Forward warping

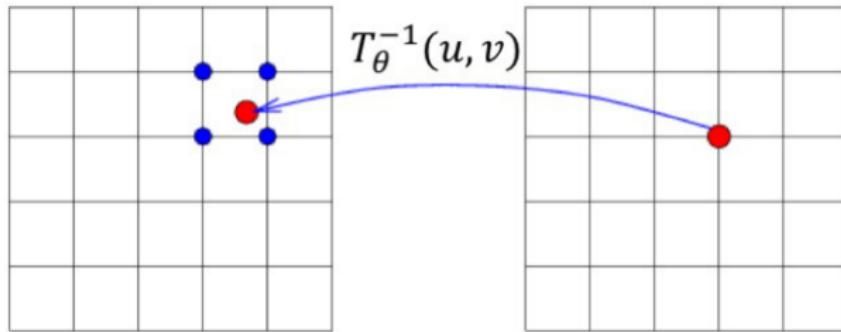
$$\begin{array}{ccc} \mathbf{T}_\phi(x, y) : & \mathbb{R}^2 & \rightarrow & \mathbb{R}^2 \\ & (x, y; \phi) & \rightarrow & \mathbf{T}_\phi(x, y) \end{array}$$

- Ideally, if Ω_I and Ω_J were continuous domains, we would simply map (x, y) to $(u, v) = \mathbf{T}_\phi(x, y)$ and then compare $I(x, y)$ with $J(u, v)$
- However, Ω_I and Ω_J are regular grids ! What if $\mathbf{T}_\phi(x, y)$ is not on the grid Ω_J ? \rightarrow Splatting: add weighted contribution to neighbor pixels.

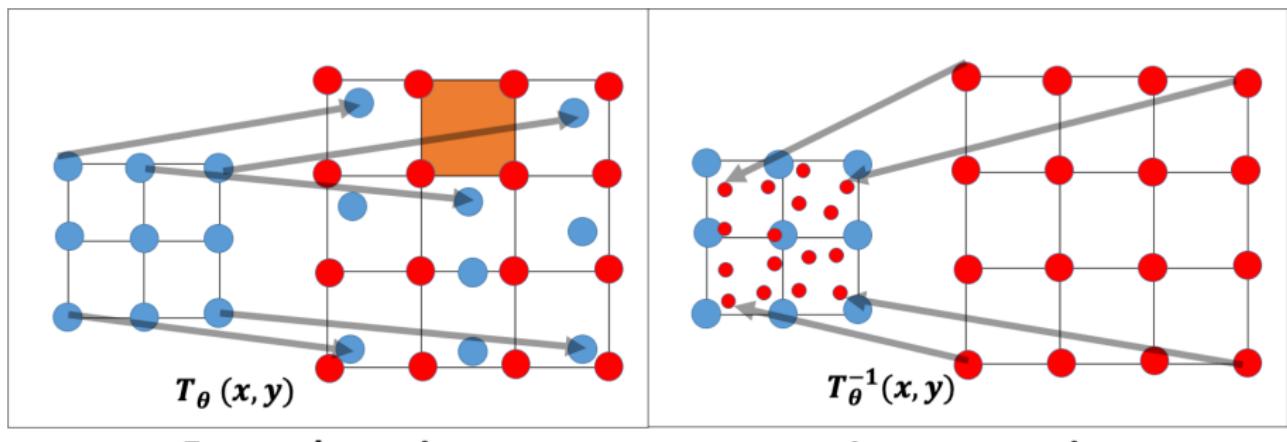


Inverse warping

- Find the pixel intensities for the deformed image I_T starting from Ω_J :
 $(x, y) = \mathbf{T}_\phi^{-1}(u, v)$
- Assign to $I_T(u, v)$ the pixel intensity in $I(x, y)$
- What if $\mathbf{T}_\phi^{-1}(u, v)$ is not on Ω_I ? → Interpolation!



Forward versus Inverse warping



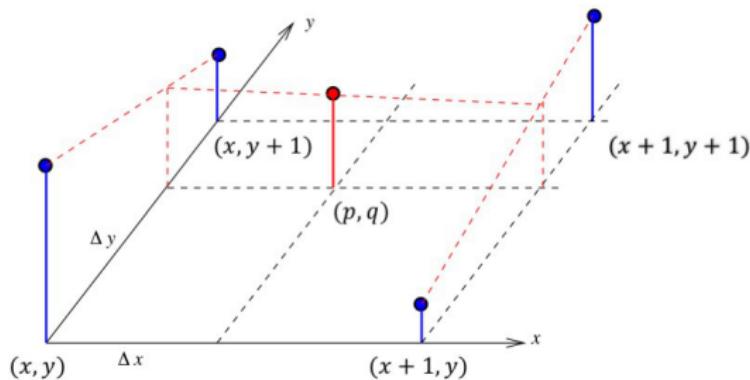
Forward warping

Inverse warping

In the case of forward warping, holes can occur in the warped image (shaded in orange). Using the inverse warping, we avoid this issue. However, we need to be able to define the inverse transformation !

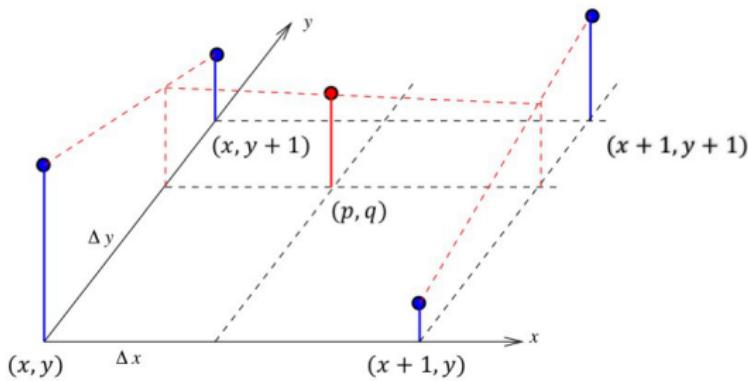
Interpolation

- **Goal:** estimate the intensity value on points not located onto the regular grids
 - nearest neighbor
 - bilinear
 - cubic
 - lanczos
 - ...



Interpolation

- **Nearest neighbor:** $J(u, v) = I(\text{round}(x), \text{round}(y))$
- **Bilinear:** $\frac{f(x, q) - f(x, y)}{\Delta y} = \frac{f(x, y+1) - f(x, q)}{1 - \Delta y} \rightarrow f(x, q) = (1 - \Delta y)f(x, y) + f(x, y+1)\Delta y$. Similarly,
 $f(x+1, q) = (1 - \Delta y)f(x+1, y) + f(x+1, y+1)\Delta y$. Then,
 $f(p, q) = f(x, q)(1 - \Delta x) + f(x+1, q)\Delta x$



Interpolation examples

Every time we deform an image, we need to interpolate it. For instance, this is the result on Lena after 10 rotations of 36 degrees:

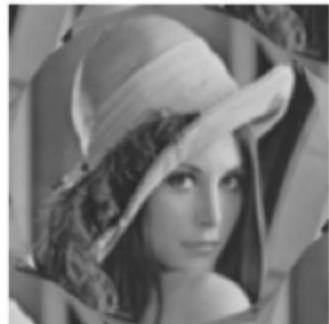
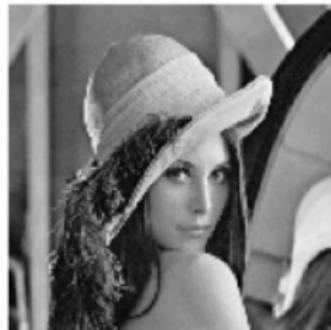


Figure 3: Original image - Nearest Neighbour - Bilinear

Source : <http://bigwww.epfl.ch/demo/jaffine/index.html> (Michael Unser)

Deformation algorithm

Recipe:

- Given a source image I and a global transformation \mathbf{T} , compute the forward warping of I . The min and max values of the x and y coordinates of the resulting deformed image $I_{\mathbf{T}}$ will give the bounding box
- Given the bounding box of $I_{\mathbf{T}}$, create a new grid within it with, for instance, the same shape of Ω_J to allow direct comparison between $I_{\mathbf{T}}$ and J
- Use the inverse warping and interpolation to compute the intensity values at the grid points of the warped image $I_{\mathbf{T}}$ (we avoid holes)
- Be careful! During the inverse warping, points that are mapped outside Ω_I are rejected.

Recap - Matrix inversion

- The inverse of a square (invertible) matrix \mathbf{T} can be computed as the ratio between the adjoint of \mathbf{T} and its determinant:
$$\mathbf{T}^{-1} = \text{adj}(\mathbf{T}) / \det(\mathbf{T})$$
- The adjoint $\text{adj}(\mathbf{T})$ of \mathbf{T} is the transpose of its cofactor matrix

- Given $\mathbf{T} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$ its cofactor matrix \mathbf{C} is:

$$\mathbf{C} = \begin{pmatrix} +\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} & +\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \\ -\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} & +\begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} \\ +\begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & +\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix},$$

Transformation parameters

- Given a transformation \mathbf{T} defined by a set of parameter θ and two images I and J , how do we estimate θ ?

Transformation parameters

- Given a transformation \mathbf{T} defined by a set of parameter θ and two images I and J , how do we estimate θ ?
- By minimizing a cost function:

$$\theta^* = \arg \min_{\theta} d(I_T, J) \quad (3)$$

- The similarity measure d might be based on the pixel intensities and/or on corresponding geometric objects such as control points (i.e. landmarks), curves or surfaces

Summary

- 1 Introduction
- 2 Geometric global transformations
- 3 Image warping and interpolations
- 4 Intensity-based registration

Intensity-based registration - similarity measures

Same modality

- Sum of squared intensity differences (SSD). Best measure when I and J only differ by Gaussian noise. Very sensitive to “outliers” pixels, namely pixels whose intensity difference is very large compared to others.

$$d(I_{\mathbf{T}}, J) = \sum_u \sum_v (J(u, v) - I(\mathbf{T}_{\phi}^{-1}(u, v)))^2 = (J(u, v) - I_{\mathbf{T}}(u, v))^2 \quad (4)$$

- Normalized Cross-Correlation. Assumption is that there is a linear relationship between the intensity of the images

$$d(I_{\mathbf{T}}, J) = \frac{\sum_u \sum_v (J(u, v) - \bar{J})(I_{\mathbf{T}}(u, v) - \bar{I}_{\mathbf{T}})}{\sqrt{\sum_u \sum_v (J(u, v) - \bar{J})^2 \sum_u \sum_v (I_{\mathbf{T}}(u, v) - \bar{I}_{\mathbf{T}})^2}} \quad (5)$$

Multi modality

- Mutual information. We first need to define the joint histogram between I_T and J . The value in (a, b) is equal to the number of locations (u, v) that have intensity a in $I_T(u, v)$ and intensity b in $J(u, v)$. For example, a joint histogram which has the value of 2 in the position (4,3) means that we have found two locations (u, v) where the intensity of the first image was 4 ($I_T(u, v) = 4$) and the intensity of the second was 3 ($J(u, v) = 3$). By dividing by the total number of pixels N , we obtain a joint probability density function (pdf) $p_{I_T, J}$.
- The sum over the rows or columns gives the marginal pdf of J (p_J) and of I_T (p_{I_T}) respectively

Intensity-based registration - similarity measures

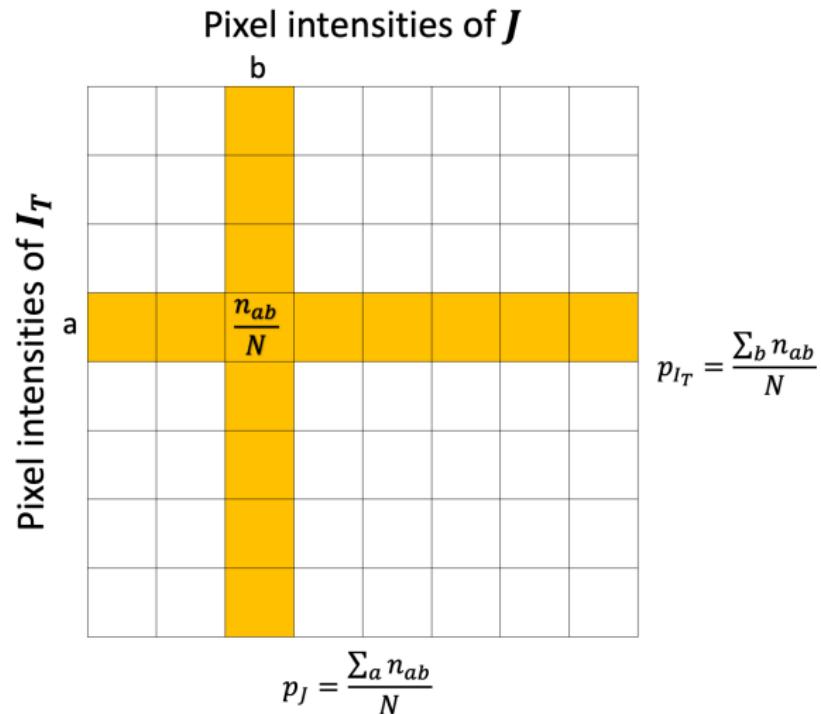


Figure 4: Normalized joint histogram example. n_{ab} indicates the number of locations where the intensity of I_T is equal to a and the intensity of J is equal to b

Intensity-based registration - similarity measures

- We suppose that the pixels of I_T and J take only 8 different intensity values, or that we can group them into 8 bins
- What's the difference between the two normalized histograms ?
Which one represents a perfect alignment ?

7	0	0	0	0	0	0	0	0.05
6	0	0	0	0	0	0	0.1	0
5	0	0	0	0	0	0.15	0	0
4	0	0	0	0	0.1	0	0	0
3	0	0	0	0.1	0	0	0	0
2	0	0	0.1	0	0	0	0	0
1	0	0.2	0	0	0	0	0	0
0	0.2	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7

7	0	0	0	0	0	0.01	0.03	0.05
6	0	0	0.04	0.01	0.1	0.05	0.01	0.05
5	0	0	0.02	0.01	0.05	0.01	0.01	0.05
4	0	0	0.02	0	0.01	0.04	0.01	0.01
3	0	0.01	0.2	0.01	0.03	0.01	0.01	0.01
2	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0
1	0.01	0.02	0	0	0	0	0	0
0	0.02	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7

Intensity-based registration - similarity measures

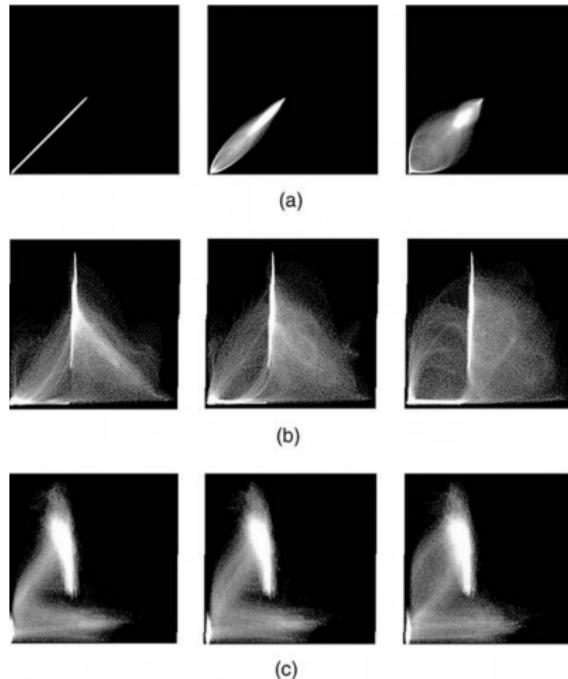


Figure 5: Joint histogram of a) same modality (IRM) b) different modality (MR-CT) c) different modality (MR-PET). First column, images are aligned. 2nd and 3rd columns images are translated. Taken from [2].

Intensity -based registration - similarity measures

- The definition of joint entropy is:

$$H(I_T, J) = - \sum_a \sum_b p_{I_T, J}(a, b) \log(p_{I_T, J}(a, b)) \quad (6)$$

- where a and b are defined within the range of intensities in I_T and J respectively
- The individual entropies of I_T and J are:
 $H(I_T) = - \sum_a p_{I_T}(a) \log(p_{I_T}(a))$ and
 $H(J) = - \sum_b p_J(b) \log(p_J(b))$ respectively.
- We know that $H(I_T, J) \leq H(I_T) + H(J)$ and the more similar the distributions p_{I_T} and p_J , the lower the joint entropy compared to the sum of individual entropies

Intensity-based registration - similarity measures

- The definition of Mutual information is:

$$M(I_{\mathbf{T}}, J) = H(I_{\mathbf{T}}) + H(J) - H(I_{\mathbf{T}}, J) \quad (7)$$

- It results:

$$M(I_{\mathbf{T}}, J) = \sum_a \sum_b p_{I_{\mathbf{T}}, J}(a, b) \log \frac{p_{I_{\mathbf{T}}, J}(a, b)}{p_{I_{\mathbf{T}}}(a)p_J(b)} \quad (8)$$

- It can be seen as the Kullback–Leibler divergence between $p_{I_{\mathbf{T}}, J}$ and $p_{I_{\mathbf{T}}} \otimes p_J$. It measures the cost for considering $I_{\mathbf{T}}$ and J as independent random variables, when in reality they are not.

Intensity-based registration - similarity measures

$$M(I_T, J) = H(I_T) + H(J) - H(I_T, J) = H(J) - H(J|I_T) \quad (9)$$

- where the conditional entropy

$$H(J|I_T) = -\sum_a \sum_b p_{I_T, J}(a, b) \log p_{J|I_T}(b|a).$$

- Mutual information measures the amount of uncertainty about J minus the uncertainty about J when I_T is known, that is to say, how much we reduce the uncertainty about J after observing I_T . It is maximized when the two images are aligned. [8]
- Maximizing M means finding a transformation \mathbf{T} that makes I_T the best predictor for J . Or, equivalently, knowing the intensity $I_T(u, v)$ allows us to perfectly predict $J(u, v)$.

Intensity-based registration - optimization procedure

- Pixel-based similarity measures need an **iterative** approach where an initial estimate of the transformation is gradually refined using the gradient and, depending on the method, also the Hessian of the similarity measure with respect to the parameters θ
- Possible algorithms: gradient descent, Gauss-Newton, Newton-Raphson, Levenberg-Marquardt, etc.
- Problem of the “local minima” → stochastic optimization, line search, trust region, multi-resolution (first low resolution and then higher resolution)
- Validation** → visual inspection, alignment of manually segmented objects, value of similarity measure

Intensity-based registration

- The goal is to minimize a similarity measure d between a transformed source image I_T and a target image J with respect to the parameters θ of the transformation T
- If we choose the SSD as similarity measure, it results:

$$\theta^* = \arg \min_{\theta} \sum_u \sum_v (I(\mathbf{T}_\theta^{-1}(u, v)) - J(u, v))^2 = \\ \sum_u \sum_v (I_T(u, v; \theta) - J(u, v))^2 \quad (10)$$

- This is a non-linear optimization procedure even if the transformation T is linear in θ because the pixel intensities are (in general) not (linearly) related to the pixel coordinates (u, v)

Intensity-based registration - Lucas-Kanade Algorithm

- Probably the first image registration algorithm was the Lucas-Kanade one (1981)
- We start with an initial guess for the parameters θ and we look for the best increment to the parameters $\Delta\theta$, by minimizing:

$$\Delta\theta^* = \arg \min_{\Delta\theta} \sum_u \sum_v (I_T(u, v; \theta + \Delta\theta) - J(u, v))^2 \quad (11)$$

- After that, the parameters are updated as: $\theta^* = \theta + \Delta\theta^*$
- These two steps are iterated until convergence (typically $||\Delta\theta|| < \epsilon$)

Intensity-based registration - Lucas-Kanade Algorithm

- We first linearize the non-linear expression in Eq.11 by performing a first order Taylor expansion on $I_{\mathbf{T}}(u, v; \theta + \Delta\theta)$, obtaining:

$$\sum_u \sum_v (I_{\mathbf{T}}(u, v; \theta) + \nabla I_{\mathbf{T}}(u, v; \theta)^T \frac{\partial \mathbf{T}}{\partial \theta}(u, v; \theta) \Delta\theta - J(u, v))^2 \quad (12)$$

- Reminder: the first order Taylor expansion of a composite scalar function is:

$$f(g(x + h)) \approx f(g(x)) + f'(g(x))g'(x)h \quad (13)$$

Intensity-based registration - Lucas-Kanade Algorithm

$$\sum_u \sum_v (I_{\mathbf{T}}(u, v; \theta) + \nabla I_{\mathbf{T}}(u, v; \theta)^T \frac{\partial \mathbf{T}}{\partial \theta}(u, v; \theta) \Delta \theta - J(u, v))^2 \quad (14)$$

- $\nabla I_{\mathbf{T}}(u, v; \theta) = \left(\frac{\partial I_{\mathbf{T}}(u, v; \theta)}{\partial u}, \frac{\partial I_{\mathbf{T}}(u, v; \theta)}{\partial v} \right)^T$ is a $[2 \times 1]$ column vector and is the *gradient* of the image I evaluated at $\mathbf{T}_{\phi}^{-1}(u, v)$. This means computing the gradient of ∇I in the coordinate frame of I and then warp it back onto the coordinate frame of J using the current estimate of \mathbf{T}

Reminder - Image gradient

- The image gradient can be computed as the convolution of the original image I with one filter for the x direction and one for the y direction
- $\nabla I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})^T$ where $\frac{\partial I}{\partial x} = G_x * I$ and $\frac{\partial I}{\partial y} = G_y * I$
- Common choices for G_x and G_y are:

- **Sobel:** $G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$ and $G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

- **Scharr:** $G_x = \begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix}$ and $G_y = \begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix}$

Intensity-based registration - Lucas-Kanade Algorithm

$$\sum_u \sum_v (I_{\mathbf{T}}(u, v; \theta) + \nabla I_{\mathbf{T}}(u, v; \theta)^T \frac{\partial \mathbf{T}}{\partial \theta}(u, v; \theta) \Delta \theta - J(u, v))^2 \quad (15)$$

- $\frac{\partial \mathbf{T}}{\partial \theta}$ is a $[2 \times d]$ matrix where d is the number of parameters θ and is the *Jacobian* of the transformation. Let $\mathbf{T}(u, v; \theta) = (T_u(u, v; \theta), T_v(u, v; \theta))^T$ be a 2D column vector then:

$$\frac{\partial \mathbf{T}}{\partial \theta} = \begin{bmatrix} \frac{\partial T_u}{\partial \theta_1} & \frac{\partial T_u}{\partial \theta_2} & \dots & \frac{\partial T_u}{\partial \theta_d} \\ \frac{\partial T_v}{\partial \theta_1} & \frac{\partial T_v}{\partial \theta_2} & \dots & \frac{\partial T_v}{\partial \theta_d} \end{bmatrix} \quad (16)$$

- We follow the notational convention that the partial derivatives with respect to a column vector are laid out as a row vector. This convention has the advantage that the chain rule results in a matrix multiplication as in Eq.15

Intensity-based registration - Lucas-Kanade Algorithm

- By substituting the linearization in the original cost function Eq.11, we obtain:

$$\Delta\theta^* = \arg \min_{\Delta\theta} \sum_u \sum_v (I_{\mathbf{T}}(u, v; \theta) + \nabla I_{\mathbf{T}}(u, v; \theta)^T \frac{\partial \mathbf{T}}{\partial \theta}(u, v; \theta) \Delta\theta - J(u, v))^2 \quad (17)$$

- The partial derivative with respect to $\Delta\theta$ is:

$$2 \sum_u \sum_v \left(\nabla I_{\mathbf{T}}(u, v; \theta)^T \frac{\partial \mathbf{T}}{\partial \theta}(u, v; \theta) \right)^T (I_{\mathbf{T}}(u, v; \theta) + \nabla I_{\mathbf{T}}(u, v; \theta)^T \frac{\partial \mathbf{T}}{\partial \theta}(u, v; \theta) \Delta\theta - J(u, v)) \quad (18)$$

Intensity-based registration - Lucas-Kanade Algorithm

- Setting equal to zero the previous Eq., we obtain:

$$\Delta\theta = H^{-1} \sum_u \sum_v \left(\nabla I_{\mathbf{T}}(u, v; \theta)^T \frac{\partial \mathbf{T}}{\partial \theta}(u, v; \theta) \right)^T (J(u, v) - I_{\mathbf{T}}(u, v; \theta)) \quad (19)$$

- where H is a $[d \times d]$ matrix and is an approximation of the Hessian matrix. This is a *Gauss-Newton gradient descent algorithm*.

$$H = \sum_u \sum_v \left(\nabla I_{\mathbf{T}}(u, v; \theta)^T \frac{\partial \mathbf{T}}{\partial \theta}(u, v; \theta) \right)^T \left(\nabla I_{\mathbf{T}}(u, v; \theta)^T \frac{\partial \mathbf{T}}{\partial \theta}(u, v; \theta) \right) \quad (20)$$

- Please note that if we approximate H with an identity function, we obtain the *steepest descent parameter updates*

Intensity-based registration - Lucas-Kanade Algorithm

- Let d the number of parameters for the transformation T and n the number of pixels in J , the total computational cost of each iteration is $O(nd^2 + d^3)$
- The two most expensive steps are: computing the Hessian matrix ($O(nd^2)$) and inverting it ($O(d^3)$).
- The Lucas-Kanade Algorithm is one possible solution but other approaches exist. See [9] for more details.

Intensity-based registration - Lucas-Kanade Algorithm

Algorithm 1 Lucas-Kanade Algorithm

```
1: Get  $(r, c)$  the number of rows and columns of  $J$ , initialize  $\theta_0$ , maximum  
number of iterations  $K$  and iteration index  $k = 0$   
2: while  $\|\Delta\theta\|_2 \geq \epsilon$  and  $\|J(u, v) - I_T(u, v; \theta)\|_2 \geq \tau$  and  $k < K$  do  
3:   Initialize  $\Delta\theta = (0, \dots, 0)^T$ ,  $H = ((0, 0), (0, 0))$   
4:   for  $u = 1$  to  $r$  do  
5:     for  $v = 1$  to  $c$  do  
6:       Compute  $I_T(u, v; \theta_k)$  and  $\nabla I_T(u, v; \theta_k)$   
7:       Evaluate the Jacobian  $\frac{\partial \mathbf{T}}{\partial \theta}(u, v; \theta)$   
8:        $\Delta\theta += \left( \nabla I_T(u, v; \theta_k)^T \frac{\partial \mathbf{T}}{\partial \theta}(u, v; \theta_k) \right)^T (J(u, v) - I_T(u, v; \theta_k))$   
9:        $H += \left( \nabla I_T(u, v; \theta_k)^T \frac{\partial \mathbf{T}}{\partial \theta} \right)^T \left( \nabla I_T(u, v; \theta_k)^T \frac{\partial \mathbf{T}}{\partial \theta} \right)$   
10:      end for  
11:    end for  
12:     $\Delta\theta = H^{-1} \Delta\theta$   
13:     $k = k + 1$   
14:     $\theta_k = \theta_{k-1} + \Delta\theta$   
15: end while
```

Registration - General variational formulation

- More often, we have a local (non-rigid) deformation model for \mathbf{T} , that is richer but enables a priori physically implausible deformations
- We define the optimal transform \mathbf{T}^* as the minimizer of an energy that is the sum of two terms: an image-based data fidelity term $d(I_{\mathbf{T}}, J)$ as before, and a regularization term $\mathcal{R}(\mathbf{T})$ that penalizes physically implausible deformations

$$\mathbf{T}^* = \arg \min_{\mathbf{T}} d(I_{\mathbf{T}}, J) + \mathcal{R}(\mathbf{T}) \quad (21)$$

- $\mathcal{R}(\mathbf{T})$ typically penalizes high values for the first-order or higher-order derivatives of \mathbf{T}

References

- ① J. Modersitzki (2004). *Numerical methods for image registration*. Oxford university press
- ② J. V. Hajnal, D. L.G. Hill, D. J. Hawkes (2001). *Medical image registration*. CRC press
- ③ G. Wolberg (1990). Digital Image Warping. IEEE
- ④ The Matrix Cookbook
- ⑤ S. Umeyama (1991). Least-squares estimation of transformation parameters.... IEEE TPAMI
- ⑥ S. Durrleman et al. (2014). Morphometry of anatomical shape complexes NeuroImage.
- ⑦ J. Ashburner (2007). A fast diffeomorphic image registration algorithm. NeuroImage
- ⑧ J. P. W. Pluim (2003). Mutual information based registration of medical images: a survey. IEEE TMI
- ⑨ S. Baker, I. Matthews (2004). Lucas-Kanade 20 Years On: A Unifying Framework. IJCV